Volume 4

Cropping System Model Documentation

Volume 4

DSSAT v4.5
Cropping System Model
Documentation

Editors:

James W. Jones
Gerrit Hoogenboom
Kenneth J. Boote
Cheryl H. Porter

International Consortium for Agricultural Systems Applications
University of Hawaii

IBSNAT, The International Benchmark Sites Network for Agrotechnology Transfer, was a network consisting of the contractor (University of Hawaii), its subcontractors and many global collaborators. Together they created a network of national, regional, and international agricultural research for the transfer of agrotechnology among global partners in both developed and developing countries.

From 1982 to 1987, IBSNAT was a program of the U.S. Agency for International Development under a cost-reimbursement Contract, No. DAN-4054-C-00-2071-00, with the University of Hawaii. From 1987 to 1993, the contract was replaced with a Cooperative Agreement, No. DAN- 4054-A-00-7081-00, between the University of Hawaii and USAID.

In 1994, the International Consortium for Agricultural Systems Application or ICASA was established as a non-profit corporation in Honolulu, Hawaii. ICASA is governed by a Board of Directors consisting of former members of the IBSNAT network and international systems scientists. The ICASA network oversees the development of systems tools, including DSSAT.

Correct citation: J. W. Jones, G. Hoogenboom, P.W. Wilkens, C.H. Porter, and G.Y. Tsuji (Editors). 2010. Decision Support System for Agrotechnology Transfer Version 4.0. Volume 4. DSSAT v4.5: Crop Model Documentation. University of Hawaii, Honolulu, HI

Copyright © University of Hawaii 2010

All reported opinions, conclusions, and recommendations are those of the authors (contractors) and not those of the funding agency or the United States government.

LIBRARY OF CONGRESS 94-19296
ISBN 1-886684-08-1

Volume 4

Table of Contents

Chapter 1. Cropping System Model (CSM) Main program
Chapter 2. CROPGRO Crop Growth and Partitioning Model
Chapter 3. Other Crop Modules
Chapter 4. Weather Module
Chapter 5. Soil Water Balance Module (WATBAL)
Chapter 6. Inorganic Soil Nitrogen Module
Chapter 7. Inorganic Soil Phosphorus Module
Chapter 8. CENTURY-based Soil Organic Matter Module
Chapter 9. Ceres-based Soil Organic Matter Module
Chapter 10. Soil Dynamic Processes Module
Chapter 11. Soil-Plant-Atmosphere Module (SPAM)
Chapter 12. Management Operations Module
Chapter 13. Pest and Disease Damage Module
Chapter 14. Generic Plant Phosphorus Module

Appendix A Structure and Definition of Composite Variables
Appendix B CSM Module Interface Tables
Appendix C Module Structure

Volume 4
Chapter 1

Cropping System Model (CSM)
Main program

J. W. Jones
Gerrit Hoogenboom
Cheryl Porter

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
Washington State University, Prosser, WA
International Consortium for Agricultural Systems Applications

CSM Main Program

April 2010

C.H. Porter, J.W. Jones, G. Hoogenboom

Agricultural and Biological Engineering Department
Research Report No 2010-101
University of Florida, Gainesville, Florida

TABLE OF CONTENTS

Chapter 1 ..... **i**

1.0 Introduction **2**

2.0 Modular format **2**

3.0 Modes of operation **2**

4.0 CSM components **3**

4.1 Main program 3
4.2 Land unit module 4
4.3 Input module 4
4.4 PLANT module 4
4.5 Soil module 4
4.6 Soil plant atmosphere interface module 7
4.7 Weather module 7
4.8 Operations management module 8

5.0 References **8**

LIST OF TABLES

Table 1. Summary description of modules in the DSSAT-CSM 1
Table 2. CSM modes of operation 3
Table 3. List of crops currently included in CSM v4.5 5

LIST OF FIGURES

Figure 1. Overview of the components and modular structure of DSSAT-CSM. 3

1.0 Introduction

The DSSAT Cropping System Model (CSM) simulates growth and development of a crop over time, as well as the soil water, carbon and nitrogen processes and management practices. Figure 1 shows the main components of CSM. These include:

- A main driver program, which controls timing for each simulation,
- A Land unit module, which manages all simulation processes which affect a unit of land
- Primary modules that individually simulate the various processes that affect the land unit including weather, plant growth, soil processes, soil-plant-atmosphere interface and management practices.

Collectively, these components simulate the changes over time in the soil and plants that occur on a single land unit in response to weather and management practices. Unlike previous versions of DSSAT and its crop models, the DSSAT-CSM incorporates models of all crops within a single set of code. This design feature greatly simplifies the simulation of crop rotations since soil processes operate continuously, and different crops are planted, managed, and harvested according to cropping system information provided as inputs to the model.

2.0 Modular format

DSSAT-CSM was restructured from previous DSSAT crop models into a modular format, which is described in detail in Appendix C of this document.

As shown in Figure 1, each module has six operational steps, (run initialization, season initialization, rate calculations, integration, daily output, and summary output). The main program controls the timing of events: the start and stop of simulation, beginning and end of crop season, as well as daily time loops. This feature, an adaptation of van Kraalingen's (1991, 1995) work, allows each module to read its own inputs, initialize itself, compute rates, integrate its own state variables, and write outputs completely independently from the operation of other modules. Table 1 lists the primary and sub modules that are currently used in the cropping system model and summarizes their functions.

Only a few "interface" variables are communicated to and from each module. This allows one to "unplug" a module and replace with a different one as long as it communicates the same variables to the rest of the modules, even if the parameters, state variables, and module input files are different. The concept of “interface variables” is critical to the modular approach used in DSSAT-CSM. Tables of interface variables are included in Appendix B in this document.

Sub modules operate exactly like primary modules in that each will usually perform two or more of the six steps (run initialization, seasonal initialization, rate calculations, integration, daily output and seasonal summary). There can be additional levels of sub modules, each behaving the same way. For example, the CERES-Maize sub module could have a phenology sub module. One could unplug this phenology module, and introduce a new one, if desired, without changing the rest of the CERES-Maize module. Any module or sub module can also have other subroutines as needed; there are no technical restrictions about how simple or complex a module should be.

There are two ways of interfacing crop growth routines in the DSSAT-CSM. A new plant growth routine can be introduced by interfacing it with the Plant module. This is the approach that was used to introduce the CERES maize, millet, sorghum, and rice models into DSSAT-CSM. These routines operated as stand-alone crop models in DSSAT v3.5 and previous versions of DSSAT.

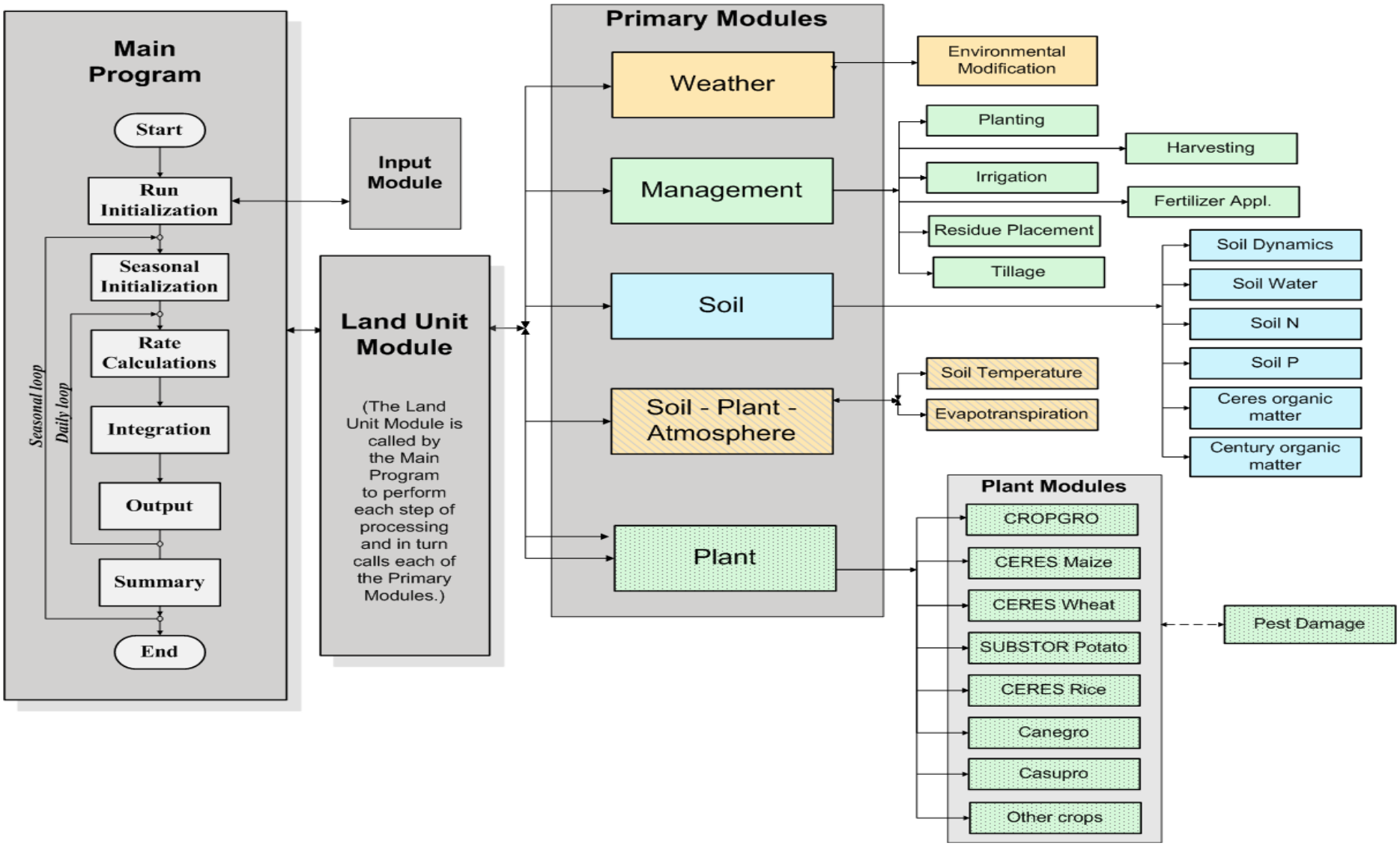


Figure 1. Overview of the components and modular structure of DSSAT-CSM.

Table 1. Summary description of modules in the DSSAT-CSM

Primary Modules	Sub Modules	Behavior
Main Program (CSM)	Controls time loops, determines which modules to call based on user input switches, and controls print timing for all modules. Calls the input module to read FILEX, soil file, and cultivar file and write the appropriate output information to a temporary input file (DSSAT45.INP).	
Land Unit	Provides a single interface between cropping system behavior and applications that control the use of the cropping system. It serves as a collection point for all components that interact on a homogenous area of land.	
Weather	Reads or generates daily weather parameters used by the model. Adjusts daily values if required, and computes hourly values	
Soil	Soil Dynamics Module	Computes soil structure characteristics by layer. Reads soil parameters and initial conditions and modifies to tillage and addition of organic matter.
	Soil Water Module	Computes soil water processes including snow accumulation and melt, runoff, infiltration, saturated flow and water table depth. Volumetric soil water content is updated daily for all soil layers. Tipping bucket approach is used.
	Soil Organic Matter Modules	Compute transformations of organic matter including mineralization and immobilization of inorganic nutrients. Two organic matter modules are available in CSM: Century (Parton) and Ceres (Godwin).
	Inorganic N Module	Inorganic soil N processes are computed including nitrification, denitrification, ammonia volatilization, urea hydrolysis, and leaching.
	Inorganic P Module	Inorganic soil P processes are computed for soluble, labile, active and stable pools of phosphorus.
Soil – Plant – Atmosphere Module (SPAM)	Root water uptake module	Computes potential root water uptake
	Potential evapotranspiration	Computes potential evapotranspiration
	Soil and flood pool evaporation	Computes actual evaporation
	Plant transpiration	Computes plant transpiraion
	Soil Temperature	Computes soil temperature by layer.
Individual Plant Growth Modules	CROPGRO Crop Template Module	Computes crop growth processes including phenology, photosynthesis, plant nitrogen and carbon demand, growth partitioning, and pest and disease damage for crops modeled using the CROPGRO model crop Template (soybean, peanut, dry bean, chickpea, cowpea, faba bean, tomato, Macuna, Brachiaria, Bahiagrass).
	CERES-Maize	Modules that simulate growth and yield for individual species. Each is a separate module that simulates phenology, daily growth and partitioning, plant nitrogen and carbon demands, senescence of plant material, etc.
	CERES-Sweetcorn	
	CERES-Wheat / Barley	
	CERES-Rice	
	CERES-Sorghum	
	CERES-Millet	
	CANEGRO - sugarcane	
	CASUPRO - sugarcane	
	MZIXM – maize IXIM	
	Cropsim – cassava, wheat, barley	
	SUBSTOR-Potato	
	Aroids (taro & tanier)	
Management Operations Module	Planting	Planting date based on input value or simulated using an input planting window and soil, weather conditions.
	Harvesting	Harvest date, based on maturity, read-in value or on a harvesting window along with soil, weather conditions.
	Irrigation	Daily irrigation, input or automatic applications based on soil water depletion.
	Fertilizer	Fertilizer additions, based on read-in values or automatic conditions.
	Residue	Application of organic material (plant, animal) applications

The second way to introduce a new crop is through the use of a crop template approach. This can be implemented through the CROPGRO module and allows users to modify values in a species crop template file a minimum of code changes. The CROPGRO development team has used this approach in creating models for different species, including faba bean (Boote et al., 2002), brachiaria grass (Giraldo et al., 1998), tomato (Scholberg et al., 1997), chickpea (Singh and Virmani, 1994) and velvet bean (Hartkamp et al., 2002), for example. A major advantage of this approach is that working with the crop template is less prone to errors as few changes to the model code are required. The major disadvantage to this method is that crops with very different life cycles from that described by the CROPGRO approach may not be adequately modeled.

The reorganization of the model code into a modular structure resulted in some differences in computation for some modules relative to DSSAT v3.5 models, even where no changes to algorithm were made. This is particularly evident in the soil water module. Previously, soil water content in each soil layer was updated sequentially throughout a day of simulation by each process affecting the soil profile. Drainage, root water uptake, soil evaporation, and plant transpiration each modified the soil water content sequentially and preferentially. With the modular structure, the various soil process rates are calculated using the value of soil water content at the end of the previous day. After all rates are calculated throughout the model, the rates are used to update the state variables. This change in order of calculations became important for some experiments, particularly where drought stresses limited plant growth.

3.0 Modes of operation

Different types of applications are accomplished in DSSAT-CSM by using different modes of operation to call the Land unit module on a daily basis. The run mode is specified by the command line arguments when the model is called. If the model is run using the DSSAT shell, these command lines are transparent to the user. However, if the model is run as a stand-alone program, the use of command line arguments becomes important. Table 2 lists the various modes of operation for CSM.

The first five run modes listed (batch, sensitivity, sequence, spatial, and seasonal) are available from the DSSAT v4 shell. All modes are available when the model is run from the command line.

Table 2. CSM modes of operation

Run mode	Description	Command line arguments
Batch	Experiments and treatments are listed in a batch file	B *batchfilename* (1)
Sensitivity analysis	Screen user interface to interactively modify input parameters	E *FileX TN* (2,3)
Sequence analysis	Soil processes are continuous, crop sequence is listed in batch file	Q *batchfilename* (1)
Spatial analysis	Simulates one or more crops over space.	S *batchfilename* (1)
Seasonal analysis	Multiple years run with the same initial conditions	N *batchfilename* (1)
Interactive	Screen user interface for interactive selection of experiment and treatment	None
Run all treatments	All treatments of specified experiment are run	A *FileX* (2)
Debug	The input module is not called to read FILEX, soils files or cultivar file. Instead, the temporary input file (*inpfile*) is read.	D *inpfile* (4)
(1) *batchfilename* – name of batch file; typically "DSSBatch.v45"		
(2) *FileX* – name of experiment file, e.g., UFGA7801.SBX		
(3) *TN* – treatment number to be run		
(4) *inpfile* – DSSATv4.5 temporary input file, typically "DSSAT45.INP"		

4.0 CSM components

A brief overview of the main program and each of the primary modules is included herein. The main program and the Land unit module are discussed in this chapter. Detailed technical documentation for the primary and secondary modules is presented in subsequent chapters of this document.

4.1 Main program

The main program reads information from the DSSAT experiment file (FILEX) that describes a particular experiment or situation to be simulated (Hunt et al., 2001) and sets a number of variables for controlling a simulation run. Figure 1 presents the interaction between the main program and the Land unit module. The main program initiates the simulation by setting the DYNAMIC variable for initialization of the run. The main program calls the input module, which reads FILEX, the soils file, and the cultivar file and writes the appropriate information for this run to a temporary input file (DSSAT45.INP) for use by the modules. The Land unit module is then called for run initialization.

The main program then starts a crop season time loop and calls the Land unit module for initializing variables that must be set at the start of each season. After initialization of the seasonal loop, the main program starts a daily loop and calls the Land unit module three times in sequence, first to compute rates, secondly to integrate, and finally to report daily outputs. After a crop season is completed, it calls the Land unit module to produce season-end variables and to create summary output files.

The main program controls operation of the various modes of simulation and provides the timing

and simulation control variables to all modules. Much of this simulation control information is contained in the composite variables; CONTROL and ISWITCH (see Appendix A for a list of composite variables that are passed among the DSSAT-CSM primary modules).

4.2 Land unit module

The Land unit module calls each of the primary cropping system modules as shown in Figure 1. At the start of each new crop season, it obtains management information from the DSSAT input file. The Land unit and primary modules link to sub modules, and thus are used to aggregate processes and information describing successive components of the cropping system. For example, the Soil module has six sub modules that integrate soil dynamics, soil water, soil N, soil P, and two modules for soil organic matter processes. In turn, these sub modules are also comprised of sub modules.

4.3 Input module

The input module reads the experiment file, soil file and cultivar file and prepares the input required for a single season of simulation. A temporary file, FILEIO (DSSAT45.INP) is created which is accessed by all other modules in CSM to read the required input data.

The input module also contains routines that allow interactive sensitivity analysis of various input parameters, when the 'E' run mode is used. The input module is bypassed completely when the 'D' run mode is used.

4.4 PLANT module

All plant growth sub-modules are called by the CSM PLANT module. With the release of DSSAT v4.5, additional crops modules have been added and there are multiple crop models available for maize, sugarcane, wheat, and barley. The crops currently simulated by CSM are listed in Table 3.

4.5 Soil module

The soil in the land unit is represented as a one-dimensional profile; it is homogenous horizontally and consists of a number of vertical soil layers. The Soil module integrates information from three sub modules: soil water, soil carbon and nitrogen, and soil dynamics. The soil dynamics module is designed to read in soil parameters for the land unit and to modify them based on tillage, long-term changes in soil carbon, or other field operations. Descriptions of the other sub modules called by the Soil module are given below.

4.5.1 Soil water sub-module

The soil water balance model developed for CERES-Wheat by Ritchie and Otter (1985) was adapted for use by all of the DSSAT v3.5 crop models (Jones and Ritchie, 1991; Jones, 1993; Ritchie, 1998). This one-dimensional model computes the daily changes in soil water content by soil layer due to infiltration of rainfall and irrigation, vertical drainage, unsaturated flow, soil evaporation, and root water uptake processes. In DSSAT-CSM v4.0 and later, soil evaporation, plant transpiration, and root water uptake processes were separated out into a soil-plant-atmosphere module (SPAM) to create more flexibility for expanding and maintaining the model. In addition, paddy management components were added so that lowland flooded fields could be simulated. The module is described in detail in Chapter 6 of this document.

Table 3. List of crops currently included in CSM v4.5

Model name	Model Reference ID	Crops simulated
CROPGRO	SBGRO045	soybean (*Glycine max* L. Merr.)
	PNGRO045	peanut (*Arachis hypogaea* L.)
	BNGRO045	dry bean (*Phaseolus vulgaris* L.)
	CHGRO045	chickpea (*Cicer arietinum* L.)
	CPGRO045	cowpea (*Vigna savi*)
	VBGRO045	velvet bean (*Mucuna pruriens*)
	FBGRO045	faba bean (*Vicia faba* L.)
	TMGRO045	tomato (*Lycopersicon esculentum* Mill.) (Scholberg et al., 1997; Boote et al., 1998a,b)
	COGRO045	cotton (*Gossypium* L.)
	CBGRO045	cabbage (*Brassica oleracea* L.)
	PRGRO045	bell pepper (*Capsicum annuum*)
	GBGRO045	green bean (*Phaseolus vulgaris*)
	BHGRO045	bahia (*Paspalum notatum* Flueggé)
	BRGRO045	brachiaria (*Brachiaria* spp.)
	BMGRO045	bermuda (*Cynodon dactylon*)
CERES – maize	MZCER045	maize (*Zea mays* L.)
IXIM	MZIXM045	maize (*Zea mays*)
CERES – sweetcorn	SWCER045	sweetcorn (*Zea mays*)
CERES – wheat/barley	WHCER045	wheat (*Triticum aestivum* L.)
	BACER045	Barley (*Hordeum vulgare)*
CERES – rice	RICER045	Rice (*Oriza sativa* L.)
CERES – sorghum	SGCER045	Sorghum (*Sorghum bicolor* (L.) Moench)
CERES – millet	MLCER045	Millet (*Pennisetum glaucum)*
CANEGRO	SCCAN045	Sugarcane (*Saccarum officinarum)*
CASUPRO	SCCSP045	Sugarcane (*Saccarum officinarum)*
Cropsim	CSCRP045	cassava (*Manihot esculenta)*
	CSCRP045	Wheat (*Triticum aestivum)*
	CSCRP045	barley *(Hordeum vulgare)*
SUBSTOR	PTSUB045	Potato (*Solanum tuberosum)*
Aroids	TRARO045	Taro (*Colocasia esculenta)*
	TNARO045	Tanier (*Xanthosoma spp.*)

4.5.2 Soil organic matter modules and nitrogen balance sub module

The DSSAT cropping system model has two options to simulate soil organic matter (SOM) processes. The original SOM model in DSSAT v3.5 (Godwin and Jones, 1991; Godwin and Singh, 1998), based on the PAPRAN model of Seligman and van Keulen (1981), was converted into a modular structure and retained in the new DSSAT-CSM. This is the default method for use by DSSAT-CSM.

Additionally, a SOM module developed by Gijsman et al. (2002), based on the CENTURY model (Parton *et al.*, 1988, 1994), is included in DSSAT-CSM. This CENTURY-based module was added to facilitate simulation of soil organic sequestration potential for different crop rotations over long time periods after initializing soil C and other variables only once at the start of the simulation.

Most of the interface input variables to the soil carbon and nitrogen balance modules are soil properties and variables computed in the soil water and soil temperature sub modules. Transport of N through the soil to deeper layers is based on water flux values obtained from the soil water module. The only interface variable from the Plant module is the array of plant mass being senesced and abscised daily. The output variables sent to other modules are ammonium and nitrate nitrogen in each soil layer.

4.5.2.1 CENTURY-based soil carbon and nitrogen balance sub module

The CENTURY model is more appropriate for use in low input agricultural systems, for example those that use green manure where the surface layer is crucial. Gijsman et al. (2002) showed that this new component greatly improved the accuracy of simulating the long-term changes in soil carbon in the Rothamsted bare fallow experiment. The main differences between the CENTURY-based module and the CERES-based soil N module are:

(i) The CENTURY-based module divides the SOM into more fractions, each of which has variable C:N and C:P ratios and can mineralize or immobilize nutrients,
(ii) it has a residue layer on top of the soil which decomposes, and
(iii) decomposition rates are texture dependent.

The CENTURY-based module distinguishes three types of SOM: [1] easily decomposable (microbial) SOM1, [2] recalcitrant SOM2, which contains lignin and cell walls, and [3] an almost inert SOM3. At initialization of the simulation, the fractional ratio of these three pools is set, with SOM1 of only about 2% of total SOM, while SOM2 and SOM3 vary with the management history of the soil (grassland or cultivated) and the degree of depletion. The improved SOM module also allows one to perform more realistic simulations on carbon sequestration, *i.e.* the accumulation or depletion of soil organic C under different management systems.

Organic phosphorus and nitrogen components are maintained in each of the SOM and FOM pools. Immobilization and mineralization of N and P are computed for each soil layer and returned to the inorganic N and P modules.

4.5.2.2 Revisions to the CERES-based soil carbon and nitrogen balance sub module

Because the CERES-based soil carbon and nitrogen balance routines are well-documented (Godwin, 1991, 1998), only the major changes to the module are described herein.

Senescence. Senesced plant matter is now added to the soil surface (for leaf, stem and shell) or to the soil layers (root, nodule) on a daily basis.

4.5.3 Soil nitrogen balance sub module

Because the CERES-based inorganic nitrogen balance routines are well-documented (Godwin, 1991, 1998), only the major changes to the module are described herein. As with the other modules in DSSAT-CSM, NTRANS was reorganized into a modular format. This resulted in a change in the order of calculations, which in some cases can alter computed results.

Flooded N routines. Flooded N processes have been added to the NTRANS routine. Other than reorganization into the CSM modular format, these routines are identical to those in the CERES-Rice model.

Ammonia volatilization. An ammonia volatilization routine was added to NTRANS. This routine was previously used in the DSSAT v3.5 CERES-Rice model.

Nitrification. The previous nitrification routine was replaced with that from the CERES-Rice model based on the work of Gilmour (1984).

4.5.4 Soil phosphorus balance sub module

An inorganic soil phosphorus module has been added to CSM with DSSAT v4.5. This routine had its origins in a model described by Daroub et al. (2003) and is described in detail in Chapter 8 of this document. Three inorganic soil phosphorus pools are maintained: labile, active and stable; with first order kinetics describing transformation between the pools. Interaction with organic matter occurs via immobilization or mineralization of P. Soluble phosphorus is assumed to be a proportion of labile P, depending on soil type and chemistry. A volume of soil which is in direct contact with roots, i.e., within a species-dependant radius around roots, is maintained to compute the fraction of soluble phosphorus which is available for uptake by the plant. Fertilizer can be placed in banded or hill applications so that the P is available in the plant root zones.

4.6 Soil plant atmosphere interface module

This module computes daily soil evaporation and plant transpiration. The current version was originally developed by Ritchie (1972) and was used in all of the DSSAT v3.5 crop models as part of the soil water balance. This module brings together soil, plant and atmosphere inputs and computes light interception by the canopy, potential evapotranspiration, as well as actual soil evaporation and plant transpiration. It also computes the root water uptake of each soil layer. The daily weather values as well as all soil properties and current soil water content, by layer, are required as input. In addition, leaf area index (LAI) and root length density for each layer are needed.

4.7 Weather module

The main function of the weather module is to read or generate daily weather data. It reads in daily weather values (maximum and minimum air temperatures, solar radiation and precipitation, relative humidity and wind speed when available), from the daily weather file. Hourly weather values are computed for use by some modules that require them. This module generates daily weather data using the WGEN (Richardson, 1981, 1985) or SIMMETEO (Geng et al., 1986, 1988) weather generators. It also can modify daily weather variables for studying climate change or

simulating experiments in which solar radiation, rainfall, maximum and minimum temperatures, day length, and/or atmospheric CO_2 concentrations were set at constant values or increased/decreased relative to their read-in values. Based on the inputs provided from the experiment file, the Weather module knows whether to just read in daily values or to generate or modify them (using the Environmental Modification sub module).

4.8 Operations management module

The management module determines when field operations are performed. These operations are planting, harvesting, applying inorganic fertilizer, irrigating, tilling and applying crop residue and organic material. These operations are specified by users in the standard "experiment" input file (Hunt et al., 2001). Users specify whether any or all of the operations are to be automatic or fixed based on input dates or days from planting.

These management options allow users a great deal of flexibility for simulating experiments that were conducted in the past for model evaluation and improvement and for simulating optional management systems for different applications. The management file also provides scope to define multiple crops and management strategies for crop rotations and sequencing.

5.0 References

Boote, K.J., M.I. Mínguez, and F. Sau. 2002. Adapting the CROPGRO-legume model to simulate growth of faba bean. Agron. J. 94:743-756.

Daroub, S.H., A. Gerakis, J.T. Ritchie, D.K. Friesen, J. Ryan. 2003. Development of a soil-plant phosphorus simulation model for calcareous weathered tropical soils. Ag. Systems 76:1157-1181.

Geng, S., Penning de Vries, F.W.T., Supit, I., 1986. A simple method for generating daily rainfall data. Agric. For. Meteorol. 36, 363-376.

Geng S., Auburn, J., Brandstetter, E., Li, B., 1988. A program to simulate meteorological variables. Documentation for SIMMETEO. Agronomy Report No. 204. University of California, Davis Crop Extension, Davis, California.

Gilmour, J. T. 1984. The effects of soil properties on nitrification and nitrification inhibition. Soil Sci. Soc. Amer. J. 48:1262-1266.

Giraldo, L. M., L. J. Lizcano, A. J. Gijsman, B. Rivera, and L. H. Franco. 1998. Adaptation of the DSSAT model for simulation of *Brachiaria decumbens* production. Pasturas tropicales 20: 2-12.

Gijsman A.J., Hoogenboom G., Parton W.J. and Kerridge P.C. 2002. Modifying DSSAT for low-input agricultural systems, using a soil organic matter - residue module from CENTURY. Agron. J. 94(3):462-474.

Godwin, D.C., and C.A. Jones. 1991. Nitrogen dynamics in soil-plant systems. p. 287-321. *In* J. Hanks and J.T. Ritchie (eds.) Modeling plant and soil systems. Agronomy monograph no. 31. ASA, CSSA, and SSSA, Madison, WI.

Godwin, D.C., and U. Singh. 1998. Nitrogen balance and crop response to nitrogen in upland and lowland cropping systems. p. 55-77. *In* G.Y. Tsuji , G. Hoogenboom, and P.K. Thornton (eds.) Understanding options for agricultural production. System approaches for sustainable agricultural development. Kluwer Academic Publishers, Dordrecht, Netherlands.

Hartkamp, A.D., G. Hoogenboom, and J.W. White. 2002. Adaptation of the CROPGRO growth model to velvet bean as a green manure cover crop: I. Model development. *Field Crops Research* 78(1):27-40.

Hunt, L.A., J.W. White, and G. Hoogenboom. 2001. Agronomic data: Advances in documentation and protocols for exchange and use. *Agricultural Systems* 70:477-492.

Jones, J. W., and J. T. Ritchie. 1991.Crop growth models. *In:* G. J. Hoffman, T. A. Howell, and K. H. Solomon 9eds.), Management of Farm Irrigation Systems. Am. Soc. Agr. Engr. Pp. 63-89.

Jones, J. W. 1993. Decision support systems for agricultural development. *In:* F. Penning de Vries, P. Teng, and K. Metselaar, Systems Approaches for Agricultural Development. Kluwer Academic Press, Boston. pp. 459-471.

Kraalingen, D. W. G. van, 1991. The FSE system for crop simulation. Simulation reports CABO-TT ; nr. 23. Centre for Agrobiological Research and Dept. of Theoretical Production Ecology, Agricultural University, Wageningen, the Netherlands.

Kraalingen, D.W.G. van. 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis, No. 1, C.T. de Wit Graduate School for Production Ecology and Resource Conservation, Wageningen University, The Netherlands, p. 58.

Parton, W.J., J.W.B. Stewart, and C.V. Cole. 1988. Dynamics of C, N, P and S in grassland soils: a model. Biogeochemistry 5:109-131.

Parton, W.J., D.S. Ojima, C.V. Cole, and D.S. Schimel. 1994. A general model for soil organic matter dynamics: sensitivity to litter chemistry, texture and management. p. 147-167. *In* R.B. Bryant and R.W. Arnold (eds.) Quantitative modeling of soil forming processes. Special Publication 39. SSSA, Madison, WI.

Richardson, C.W. 1981. Stochastic simulation of daily precipitation, temperature, and solar radiation. Water Resources Research 17(1): 182-90.

Richardson, C.W. 1985. Weather simulation for crop management models. Transactions of the ASAE 28(5): 1602-6.

Ritchie, J.T. 1972. Model for predicting evaporation from a row crop with incomplete cover. Water Resources Research 8:1204-1213.

Ritchie, J.T. 1998. Soil water balance and plant stress. p. 41-54. *In*: [G. Y. Tsuji, G. Hoogenboom, and P. K. Thornton, Editors]. Understanding Options for Agricultural Production. Kluwer Academic Publishers, Dordrecht, the Netherlands.

Ritchie, J.T., and S. Otter. 1985. Description and performance of CERES-Wheat: A User-oriented wheat yield model. p. 159-175. In ARS Wheat Yield Projhect. ARS-38. Natl Tech. Info. Serv., Springfield, Missouri.

Scholberg, J. M. S., K. J. Boote, J. W. Jones, and B. L. McNeal. 1997. Adaptation of the CROPGRO model to simulate the growth of field-grown tomato. pp. 133-151. *In* M. J. Kropff et al. (eds.). Systems Approaches for Sustainable Agricultural Development: Applications of Systems Approaches at the Field Level. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Seligman, N.C., and H. Van Keulen. 1981. PAPRAN: a simulation model of annual pasture production limited by rainfall and nitrogen. p. 192-221. *In* M.J. Frissel and J.A. Van Veen (eds.) Simulation of nitrogen behaviour of soil-plant systems. Centrum voor Landbouwpublikaties en Landbouwdocumentatie (PUDOC), Wageningen, Netherlands.

Singh, P. and S. M. Virmani. 1994. Modeling growth and yield of chickpea (*Cicer arietinum* L.). Field Crops Res. 46: 1-29.

Volume 4
Chapter 2

CROPGRO
Crop Growth and Partitioning Module

K.J. Boote
J.W. Jones
G. Hoogenboom
W.D. Batchelor
C.H. Porter

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
Iowa State University, Ames, IA
International Consortium for Agricultural Systems Applications

CROPGRO
PLANT GROWTH AND PARTITIONING MODULE

March 31, 2004

K.J. Boote, J.W. Jones, G. Hoogenboom, W.D. Batchelor, C.H. Porter

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
Iowa State University, Ames, IA
International Consortium for Agricultural Systems Applications

CROPGRO PLANT GROWTH MODULE
TABLE OF CONTENTS

1.0 CROPGRO Module Overview 4

2.0 CROPGRO Module Structure 8

3.0 Plant Growth and Partitioning (CROPGRO) 12
3.1 Run Initialization 12
3.2 Seasonal Initialization 13
3.3 Rate Calculations 13
3.4 Emergence 13
3.5 Integration 13
3.6 Output 21
3.7 Final 21

4.0 Initial Composition (INCOMP) 22
4.1 Run Initialization 22
4.2 Seasonal Initialization 22

5.0 Maintenance Respiration (RESPIR) 23

6.0 Nutrient Demand (DEMAND, IPDMND and SDCOMP) 24
6.1 Run Initialization (Subroutine IPDMND) 25
6.2 Seasonal Initialization 25
6.3 Emergence 26
6.4 Rate/Integration 26
6.5 Seed Composition (SDCOMP) 38

7.0 Nitrogen uptake (NUPTAK) 41
7.1 Run Initialization 41
7.2 Seasonal Initialization 41
7.3 Rate/Integration 41

8.0 Nitrogen Mobilization (MOBIL) 43
8.1 Seasonal Initialization 43
8.2 Rate/Integration 43

9.0 Nitrogen Fixation (NFIX) 44
9.1 Run Initialization 44
9.2 Seasonal Initialization 44
9.3 Rate/Integration 44

10.0 Pod Growth (PODS and PODCOMP) 47
10.1 Run Initialization 47
10.2 Seasonal Initialization 47
10.3 Emergence 47
10.4 Rate/Integration 48
10.5 Seed Composition (PODCOMP) 56

11.0 Pod Detachment (PODDET) 60
11.1 Run Initialization 60
11.2 Emergence 60
11.3 Rate/Integration 60

12.0 Vegetative Growth and Partitioning (VEGGR and CANOPY) 61
12.1 Run Initialization 61
12.2 Seasonal Initialization 61
12.3 Emergence 61
12.4 Rate/Integration 61
12.5 Canopy Expansion (CANOPY) 65

13.0 Leaf Senescence (SENES) 69
13.1 Run Initialization 69
13.2 Seasonal Initialization 69
13.3 Rate/Integration 69

14.0 Freeze Damage (FREEZE) 71

15.0 Root Growth (ROOTS, IPROOT and INROOT) 71
15.1 Run Initialization (IPROOT) 71
15.2 Seasonal Initialization 71
15.3 Emergence (INROOT) 71
15.4 Rate/Integration 72

16.0 Integration of Growth Variables (GROW, IPGROW and STRESS) 75
16.1 Run Initialization (IPGROW) 75
16.2 Seasonal Initialization 76
16.3 Emergence 76
16.4 Rate/Integration 77

17.0 References 84

Cover photograph: Ground nut or potato bean: Apios americana: pods. Courtesy of H. David Thurston, Professor Emeritus - Department of Plant Pathology, Cornell University.

LIST OF TABLES

Table 1– CROPGRO Module - Subroutine Calling Order 11
Table 2 – Input files read by CROPGRO Module Routines 12
Table 3 – Input Parameters Read by INCOMP Routine 23
Table 4 – Input Parameters for DEMAND Routines 25
Table 5 – TMPFAC calculations 28
Table 6 – Input Parameters Read by NFIX Routine 44
Table 7 – Input Parameters Read by PODS Routine 47
Table 8 – Example of shell age and physiological age computation. 49
Table 9 – Input Parameters Read by PODCOMP Routine 56
Table 10 – Input Parameters Read by PODDET Routine 60
Table 11 – Input Parameters Read by VEGGR Routine 61
Table 12 – Input Parameters Read by CANOPY Routine 65
Table 13 – Input Parameters Read by SENES Routine 69
Table 14 – Input Parameters Read by ROOTS Routine 71
Table 15 – Input Parameters Read by GROW Routines 75
Table 16– Interface for CROPGRO Module 86
Table 17 – Interface for INCOMP Routine 87
Table 18 – Interface for RESPIR Routine 87
Table 19 – Interface for DEMAND Routines 88
Table 20 – Interface for NUPTAK Routine 90
Table 21 – Interface for MOBIL Routine 90
Table 22 – Interface for NFIX Routine 91
Table 23 – Interface for PODS Routines 92
Table 24 – Interface for PODDET 94
Table 25 – Interface for VEGGR Routine 95
Table 26 – Interface for SENES Routine 97
Table 27 – Interface for FREEZE Routine 97
Table 28 – Interface for ROOTS Routine 98
Table 29 – Interface for GROW Routines 99

LIST OF FIGURES

Figure 1 – Sources and Sinks of C and N as modeled in CROPGRO Module 7
Figure 2 – Components of CROPGRO Module 9
Figure 3 – Schematic of the flow of C (PGAVL) and N (NAVL) 14
Figure 4 – Effect of temperature on seed growth rate 28
Figure 5 – Relative partitioning to leaf, stem and roots prior to first flowering 33
Figure 6 – Effect of PAR on specific leaf area (F) using variable PARSLA. 34
Figure 7 – Relative effect of temperature on specific leaf area 35
Figure 8– Effect of temperature on specific leaf area using variable TPHFAC 35
Figure 9 – Leaf Area Expansion vs. VSTAGE during Early Growth Stage 36
Figure 10 – Seed composition as a function of temperature. 40
Figure 11 – Effect of temperature on pod setting using quadratic curve. 51
Figure 12 – Reduction in seed growth rate (XRSD) due to limited N supply 59
Figure 13 – Effect of Temperature on Canopy Expansion 67
Figure 14 – Leaf area index vs. effective shading of growing point. 67
Figure 15 – Effect of PAR on Canopy Expansion 68
Figure 16 – Effect of VSTAGE on Canopy Variables 68
Figure 17 – VSTAGE vs. Senescence 69
Figure 18 – VSTAGE vs. root growth rate 72

1.0 CROPGRO Module Overview

The CROPGRO module is called by the Land Unit Module of the DSSAT4 Cropping System Model (CSM) to compute plant growth and development processes on a daily basis. Within each day, carbon and nitrogen balance concepts are utilized to describe the tissue growth of different plant components and the storage of excess C and N not used in tissue growth. Demands for C and N for tissue growth are compared with supplies of these substrate to determine daily growth and C and N composition of each growing tissue. During a day, growth may be limited by limited sink strength (or potential growth rates of growing tissue) or by C or N availability. After all processes have been considered in a day, growth rate variables are computed for each component, N and C use rates are calculated, and any excess of C or N is calculated. These rates are used to perform the seasonal mass balances for C, N, and dry weights.

The overall mathematical structure of this crop growth model is based on the differential equation representation of mass balances presented by Wilkerson et al. (1985) for the original SOYGRO soybean model. These equations express the daily net growth rates by each of the components in the model as the sum of new growth minus any abortion, repartitioning of mass, and pest damage that occurs in a day. The equations contain rate variables and state variables. The rate variables describe the rates of change, or equivalently, the daily change, in weights, nitrogen, or carbon storage in the different plant components.

In the original soybean model, equations were developed for the accumulation of mass and nitrogen in various plant tissues. In this new generic crop growth model, daily rate equations were added to account for the storage and use of carbon in excess of what can be used to grow tissue mass in a day. The main subroutine of the plant growth module, CROPGRO, controls the process calculations of rate variables to determine the available carbon based on daily photosynthesis minus maintenance respiration, the available nitrogen based on N uptake, mobilization, and N fixation, and the net tissue growth rate of each plant component. Subroutine GROW integrates these process rate variables using a daily time step to update the state variables of the crop after each day's growth.

Boote et al. (1998) discusses the theoretical concepts used in the CROPGRO model in some detail. For additional information, that publication is recommended.

Units for Rate and State Variables. Rate variables for most of the plant components have units of mass per m^2 per day. State variables describe the simulated plant component dry weights on a given day, and usually have units of mass per unit area.

The plant, which is simulated by the model, is assumed to be composed of leaves, stems, roots, shells, seed and nodules. State variables in the model describe each of these types of tissue. Tissue dry weight, nitrogen content (N), carbon reserves (C, non-structural carbohydrates), leaf area and number of leaves on the main stem (V-Stage) are the state variables that describe leaves. Dry weights have units g[plant material]/m^2. Stems, shells, and roots also have tissue dry weights, N, and C contents. Tissue dry weight in each plant component includes the weights of N and C that may be stored in it. A proportion of the vegetative weight that is N and C may be mobilized from the vegetative for use in growth, whereas the tissue in other plant components can not be mobilized. It remains in the component in which it grew until the end of the season or until it is aborted from the plant.

Seeds are described by their dry weight and N content. In addition, numbers of seed and shells are state variables that depend on addition and abortion rates. In GROW, the overall numbers or weights

of seed and shells are updated, whereas in PODS the individual age classes of seeds and shells are updated. This separation of the age classes or cohorts for seed and shells from the overall plant state variables is somewhat arbitrary because the sum of the weights of seed over all age classes (calculated in PODS) must equal the overall weight of seed (computed in GROW). However, this separation allows for a clear presentation of the overall weight, N and C balances for the crop (in GROW) with the details of how it is distributed among age classes to be handled elsewhere. The overall equations for growth presented below are independent of whether the seeds and shells have cohorts. Roots also are broken down into different groups with weights and lengths of roots maintained for each soil layer.

For purposes of computing photosynthesis, C stored in the plant, C used for respiration, and C used for mass growth, the model uses the carbohydrate form (CH_2O). Mass of plant components, however, has units of g[tissue]. Tissue growth rates are based on conversion efficiency multiplied by the mass of CH_2O used in growth. This efficiency accounts for tissue synthesis and for respiration costs for synthesis as explained in the section on growth model equations (subroutine GROW).

Nitrogen may be taken up from the soil, mobilized from plant tissue, and assimilated through symbiotic N fixation. After initial assimilation and reduction, N in the plant is assumed to be in the form of protein groups, with a constant proportion of N (16%). As N is assimilated from the different sources, CH_2O is subtracted to account for the energy costs of synthesizing the protein groups. Calculations are also made to convert between fraction of N and fraction of protein using the assumption that protein is 16% N.

C and N Flow and Utilization. Figure 1 is a schematic of the overall C and N flow paths in the plant. A priority scheme is assumed to control the flow of C and N during a day. Total CH_2O available for growth in a day is equal to the day's photosynthesis plus mobilized CH_2O from C reserves in leaves, stems, shells, and roots minus maintenance respiration. Photosynthesis is computed first, maintenance respiration is subtracted and potential amount of C mined in a day (CMINEP) is added to compute the maximum CH_2O available (PGAVL) for all growth, N reduction, N fixation, and growth respiration processes during the day.

Seeds have the highest priority for all C and N in determinate crops, but in indeterminate crops, this priority is limited to a maximum daily reproductive partitioning coefficient (XFRUIT, g[fruit growth] / g[total plant growth]). For example, peanut varieties may have partitioning coefficients of 0.6 to 0.9, meaning that 40 to 10 percent of the day's C would go to vegetative growth, regardless of the sink strength of seed and shells. Potential growth rates are computed, based on available C from the current day's photosynthesis as well as mobilized C.

Nitrogen assimilation is limited to that needed for maximum tissue concentrations for potential growth, as defined by available C. If additional N is needed, based on the demand calculations, then N mobilized from existing tissue as protein groups is computed next, and C is subtracted from the available C (PGAVL) to account for the energy costs in resynthesis of protein. Soil N uptake is computed and C is subtracted from available C to account for the energy costs of conversion of NO_3 and NH_4 to protein groups. If still more N is needed, N fixation is computed, as described in Section 10.3 and as limited by nodule mass and specific activity. Either C or N can limit growth during any day.

Users may optionally choose not to simulate growth with the soil N balance and N fixation. In this case, N mobilization will occur and additional N will be "fixed" as needed, using the amount of C

needed to account for the cost of N fixation. Another option allows the soil N routines to be used, with the assumption that N fixation will occur as needed, after NO_3 and NH_4 are taken up from the soil.

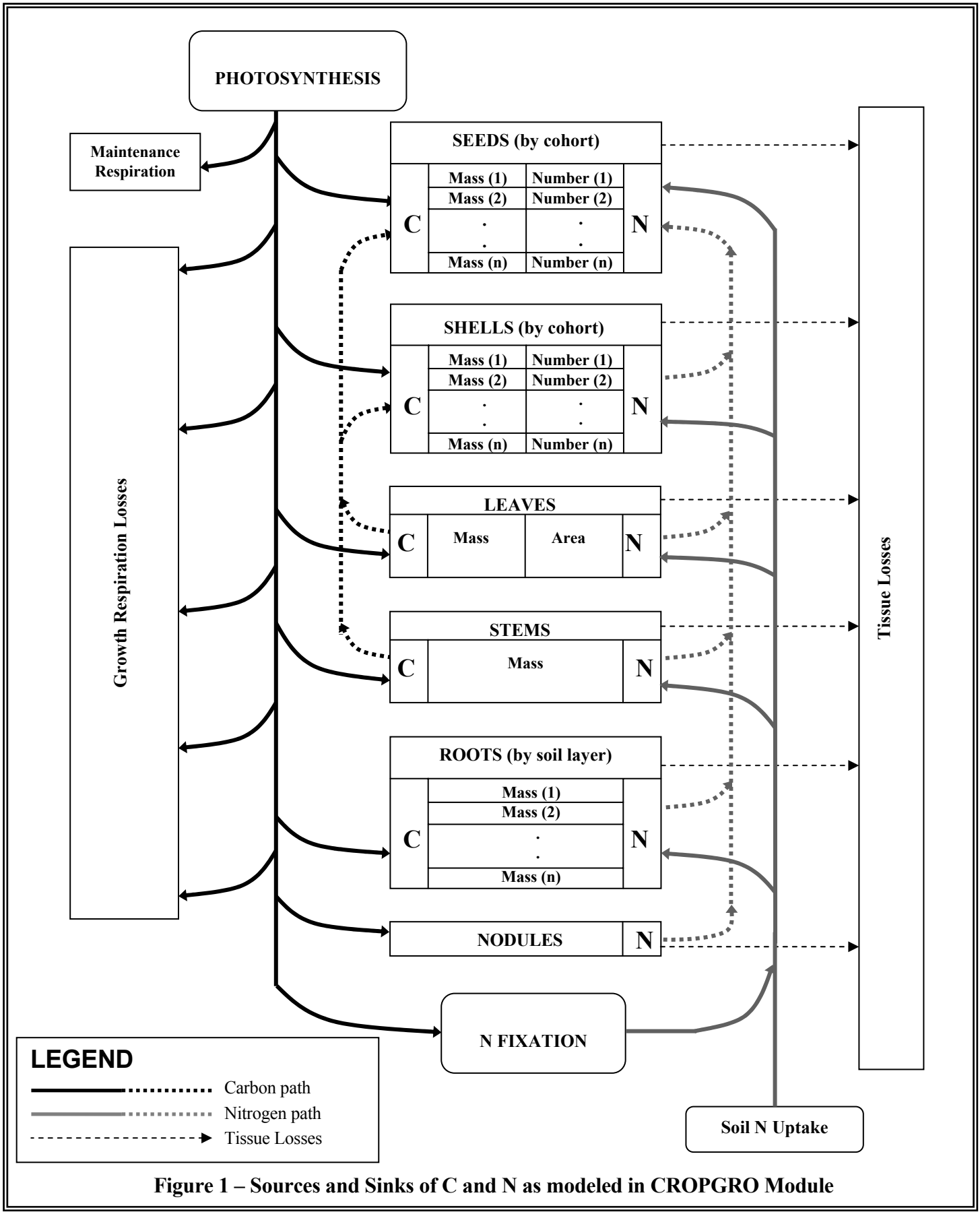


Figure 1 – Sources and Sinks of C and N as modeled in CROPGRO Module

2.0 CROPGRO Module Structure

The plant growth module is called by the CROPGRO main program. The FORTRAN code is organized in a modular format as described in "Module Structure in DSSAT v4.0 Documentation and Source Code Listing" (Porter et al., 1998).

The module is called a total of six times:
1) once at the beginning of simulation for input of parameters from files (DYNAMIC = RUNINIT),
2) once at the beginning of each seasonal simulation for initialization (DYNAMIC = SEASINIT),
3) daily for rate calculations (DYNAMIC = RATE),
4) daily for integration (DYNAMIC = INTEGR),
5) daily for output (DYNAMIC = OUTPUT) and
6) once at the end of each season of simulation to close output files (DYNAMIC = FINAL).

The CROPGRO module differs from other modules described in the module structure paper in that rate and integration calculations could not be easily separated when the modular structure was incorporated into the model. Instead, the rate calculation and integration calculations are combined into a single section. Had the original code been written to a modular format, these functions may well have been more easily separated.

An additional section has been used in the CROPGRO module, which is not seen in other CSM modules. This is for initialization upon plant emergence (DYNAMIC = EMERG). The GROW, ROOTS, DEMAND, PODS, VEGGR, and PODDET routines all contain a section for initialization on the day of plant emergence.

Figure 2 presents the components of the CROPGRO module. As shown, the main CROPGRO routine controls the flow of processing within the module and calls 17 other routines.

1) Subroutine IPPLNT is called to read data from input files for use by the CROPGRO module.

2) The vegetative and reproductive development module (PHENOL) is called first to update the stages of growth for the plant. Then routines are called to compute C and N assimilation and their allocation to growing tissues or reserve pools. This module is discussed in detail in a separate document (Jones, J. W., et al., 2000).

3) RESPIR is called to calculate maintenance respiration and net photosynthate available.

4) The pest and disease damage module (PEST) is called to calculated daily pest damage. This module is discussed in detail in a separate document (Batchelor, et al., 2000).

5) INCOMP computes initial composition and respiration requirements for leaf, stem, root, nodule, shell and seed components.

6) DEMAND is called to compute the potential growth rates of all plant organs for the day. The purpose of this subroutine is to compute the N required for growing tissue at the potential growth rates allowed by the available C and the environment (i.e., temperature, water states). N demand is used to limit N uptake from the soil as well as N fixation.

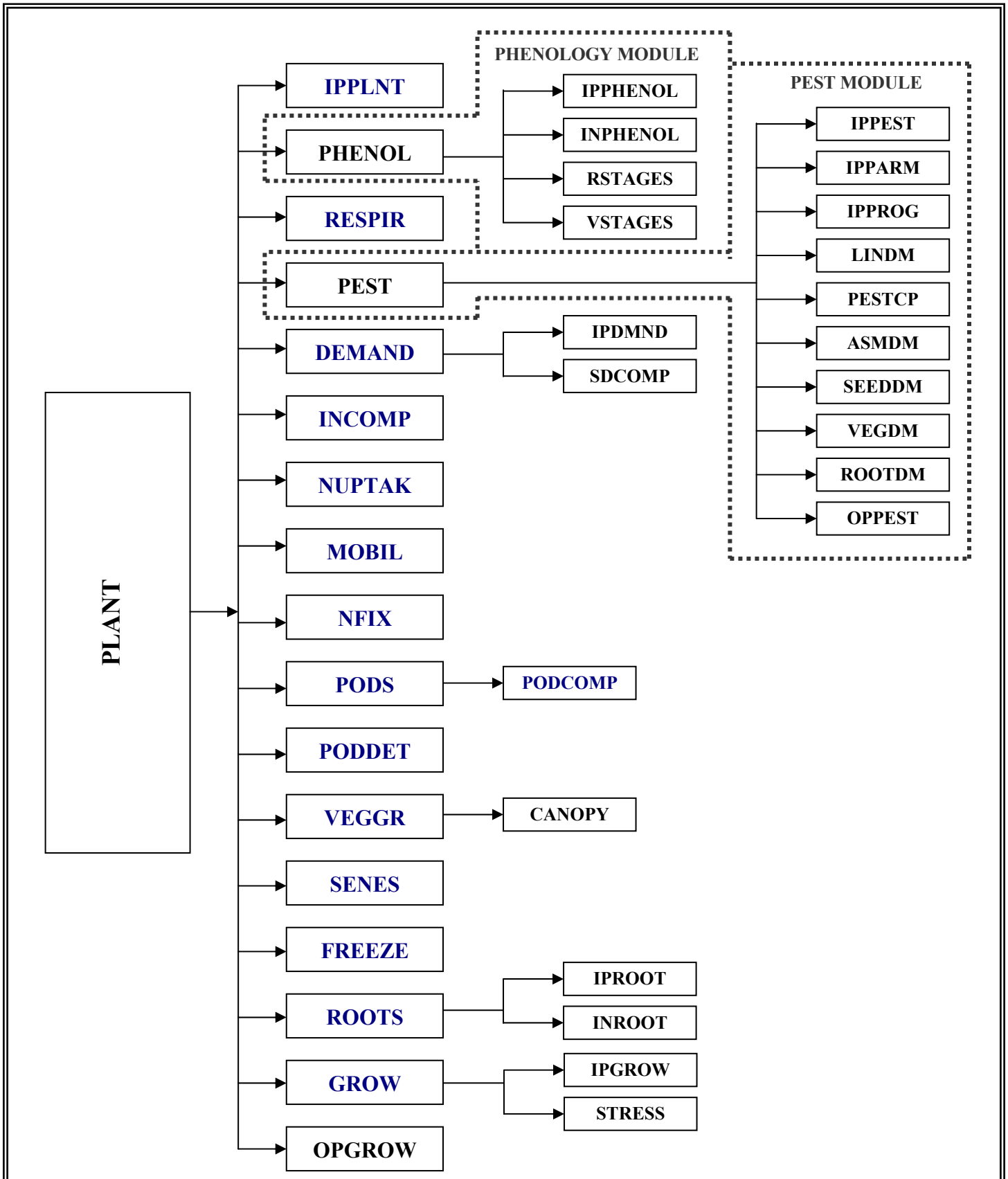


Figure 2 – Components of CROPGRO Module

7) The soil N uptake routine (NUPTAK) is called to compute supply of N from the soil. Soil N uptake may increase vegetative N concentrations if N mining has depleted N concentrations below seasonal-varying values.

8) The nitrogen mobilization subroutine, MOBIL, is called to compute N supply from existing tissue.

9) If there is sufficient N supply from the soil and from mobilization, there would be no N fixation. If N supply from mobilization and uptake is not sufficient to meet new growth demands, then N fixation will occur, if there is carbon available and if the nodules have sufficient capacity (Subroutine NFIX). There is a possibility that there will still be a shortage of N after N fixation because of insufficient capacity. In this case, C will be in excess supply relative to N and some C will be stored in the plant and become available on subsequent days.

10) The PODS subroutine is called to compute actual growth of seeds and shells as well as initiation of new fruit. Final seed composition is also determined here with a call to PODCOMP and is based on actual C and N supply limitation.

11) An option to compute pod and seed abscission losses for peanut is included in the subroutine PODDET.

12) Any remaining C and N after reproductive growth is used for vegetative growth (VEGGR). This subroutine controls partitioning among vegetative tissues (leaf, stem, root) and accumulates excess CH_2O if N is deficient for vegetative growth.

13) Leaf senescence is calculated in the SENES subroutine. It can be caused by N mining, low light, water stress, and plant development (aging).

14) If freezing temperature occurs, subroutine FREEZE is called to determine if the plants die or just lose their leaves.

15) ROOTS is called to compute root depth extension, length proliferation, and senescence in individual soil layers. These calculations depend on root mass increase passed from the VEGGR subroutine.

16) The GROW subroutine is called to integrate the processes and update the overall plant state variables.

17) OPGROW writes daily output to the Plantgro.OUT file.

Many of the subroutines called by the CROPGRO module can be considered to be independent modules, as they perform two or more processing functions such as input of parameters, initialization of variables, rate calculations, integration of state variables, output of data or final processing. The components of the CROPGRO module are summarized in Table 1 including the order in which routines are called from the CROPGRO main module for each of the modular processing functions. For instance, in the 'Seasonal Initialization' section, 13 routines are called by the main routine of the CROPGRO Module.

Each routine will be discussed separately in this document beginning with the main CROPGRO module in Section 3.0. Subsequent sections will be discussed in the order in which they are called from the main CROPGRO module (as listed in Table 1). The pest and disease damage module (PEST) and the phenology module (PHENOL) are described in detail in separate documents (Batchelor, 2000 and Jones, J. W., 2000).

Appendix Q lists variables that are used in the various components of the CROPGRO module, as well as the associated units and definitions. Appendices B through P present listings of the FORTRAN code for the main module and the additional routines.

Table 1– CROPGRO Module - Subroutine Calling Order

Subroutines called from CROPGRO	Phase in which Routine is Called						Section number this document	Description
	Run Initialization	Seasonal Initialization	Rate / Integration	Emergence **	Output	Final		
IPPLNT	1						4.0	Read input data for CROPGRO Module
PHENOL	2	1	1, 3 *				--	Calculates phenological development
RESPIR			4				6.0	Calculates maintenance respiration and net photosynthate available
PEST	3	2	2, 5 *		1	1	--	Computes daily pest damage rates
DEMAND	4	3	6	3			7.0	Calculates potential demand for C and N
INCOMP	5	4					5.0	Initializes parameters for composition of tissues at the beginning of each run.
NUPTAK	6	6	7				8.0	Determines N uptake
MOBIL		7	8				9.0	Calculates the mobilization of N
NFIX	7	8	9				10.0	Calculates nitrogen fixation
PODS	8	9	10	4			11.0	Computes seed and shell growth
PODDET	9		11	6			12.0	Computes pod detachment rates
VEGGR	10	10	12	5			13.0	Calculates vegetative partitioning as a function of V-stage and limits growth prior to VSINK.
SENES	11	11	13				14.0	Calculates leaf senescence due to natural aging, drought light stress, and physiological maturity.
FREEZE			14				15.0	Calculates freeze damage
ROOTS	12	12	15	2			16.0	Calculates root length extension, proliferation and senescence
GROW	13	5	16	1			17.0	Integrates all growth related variables
OPGROW		13			2	2		Generates GROWTH.OUT file (daily output)

* Separate calls are made for rate and integration for these modules or subroutines.

** Calls to subroutines for emergence calculations are made between the 3rd (PHENOL) and 4th (RESPIR) subroutine calls for Rate/Integration.

3.0 Plant Growth and Partitioning (CROPGRO)

The CROPGRO Module code is divided into the seven phases of computation: 'Run Initialization', 'Seasonal Initialization', 'Rate Calculations' 'Emergence', 'Integration', 'Output', and 'Final'. Although the plant growth routines have not been separated into distinct 'Rate' and 'Integration' sections, the pest damage and phenology modules, which are called by the plant module, do have distinct 'Rate' and 'Integration' sections. Therefore, these divisions have included in the plant module.

Table 16 presents a list of variables that represent the interface for the CROPGRO module.

3.1 Run Initialization

Within the 'Run Initialization' section of code, thirteen subroutines are called in turn to read the required input data for those routines. Many of these input variables are read by multiple routines, but they are considered to be local variables to each routine, as they are not passed as arguments between routines.

Table 2 presents the various routines within the CROPGRO module that read input data and the files that are read by each. Each of these input routines will be discussed in later sections.

Table 2 – Input files read by CROPGRO Module Routines

Subroutine Name:	Input File Name				
	FILEIO[1]	FILEC[2]	FILEE[3]	FILEP[4]	FILET[5]
IPPLNT	X	X			
PHENOL (IPPHENOL)	X	X	X		
PEST (IPPEST, IPPARM, IPPROG)	X			X	X
DEMAND (IPDMND)	X	X	X		
INCOMP		X	X		
NUPTAK		X			
NFIX	X	X			
PODS/PODCOMP	X	X	X		
PODDET		X			
VEGGR		X			
CANOPY		X	X		
SENES		X			
ROOTS (IPROOT)	X	X			
GROW (IPGROW)	X	X	X		

Description of input files:

[1] FILEIO – Input file created for CROPGRO by pre-processor program – general information about the simulation (e.g., DSSAT40.INP)

[2] FILEC – Species information file (e.g., SBGRO030.SPE)

[3] FILEE – Ecotype information file (e.g., SBGRO030.ECO)

[4] FILEP – Pest data file (e.g., SBGRO030.PST)

[5] FILET – Time series data file (e.g., UFGA7801.SBT)

3.2 Seasonal Initialization

Several variables calculated in the main routine of the CROPGRO Module are initialized at the beginning of this section. Calls to twelve subroutines are then made to perform initialization calculations for variables which "belong" to those routines. The PEST routine is called only if ISWDIS = 'Y' (indicating that pest damage is to be simulated). The NFIX routine is called only if ISWNIT = 'Y' (nitrogen cycle is to be simulated) and ISWSYM = 'Y' (nitrogen fixation is to be simulated). The ROOTS subroutine is called only if soil water processes are being simulated (ISWWAT = 'Y').

Subroutine OPGROW is also called to generate headings for the GROWTH.OUT file.

3.3 Rate Calculations

The Phenology and Pest modules are called within the rate calculation section to calculate rates of change of phenological and pest damage processes.

The remainder of the routines in the CROPGRO Module contain a combined 'Rate / Integration' section, as no attempt was made to "retrofit" the code to differentiate between the two functions, although they are all called from the 'Integration' section of the code.

3.4 Emergence

The CROPGRO Module calls six routines with DYNAMIC = EMERG, to perform initialization on the day of plant emergence (DAS = NVEG0). This initialization differs from the initialization at the beginning of a season of simulation in that the values are not determined or assigned until the time that the plants emerge or, for transplants, are planted. The routines called for initialization upon emergence are: GROW, ROOTS, DEMAND, PODS, PODDET, and VEGGR. The VEGGR and PODS routines in turn call subroutines CANOPY and PODCOMP, respectively, for emergence initialization.

The specific 'Emergence' calculations are described in more detail in the discussions on individual subroutines.

3.5 Integration

At the beginning of the integration section, the phenology module is called to compute progress toward emergence, which occurs on day NVEG0. Until simulated time reaches NVEG0, other plant growth processes are not calculated. If emergence has occurred today, (DAS = NVEG0), then calls are made to the appropriate routines to perform initialization for the day of emergence.

The remainder of the integration section is performed subsequent to emergence and combines calls to subroutines for a combination of rate calculations and integration of state variables, mixed with calculations within the main routine of the CROPGRO module for updating the available carbon and nitrogen supplies (PGAVL and NAVL).

Figure 3 is a schematic of the C and N calculation sequence during a day. As calculations are made for the different processes, three variables are used to account for the masses of C and N remaining for growth and respiration. The variable CMINEP is the potential mass of C that can be mined from long-term reserves in a day. The variable PGAVL is the mass of C

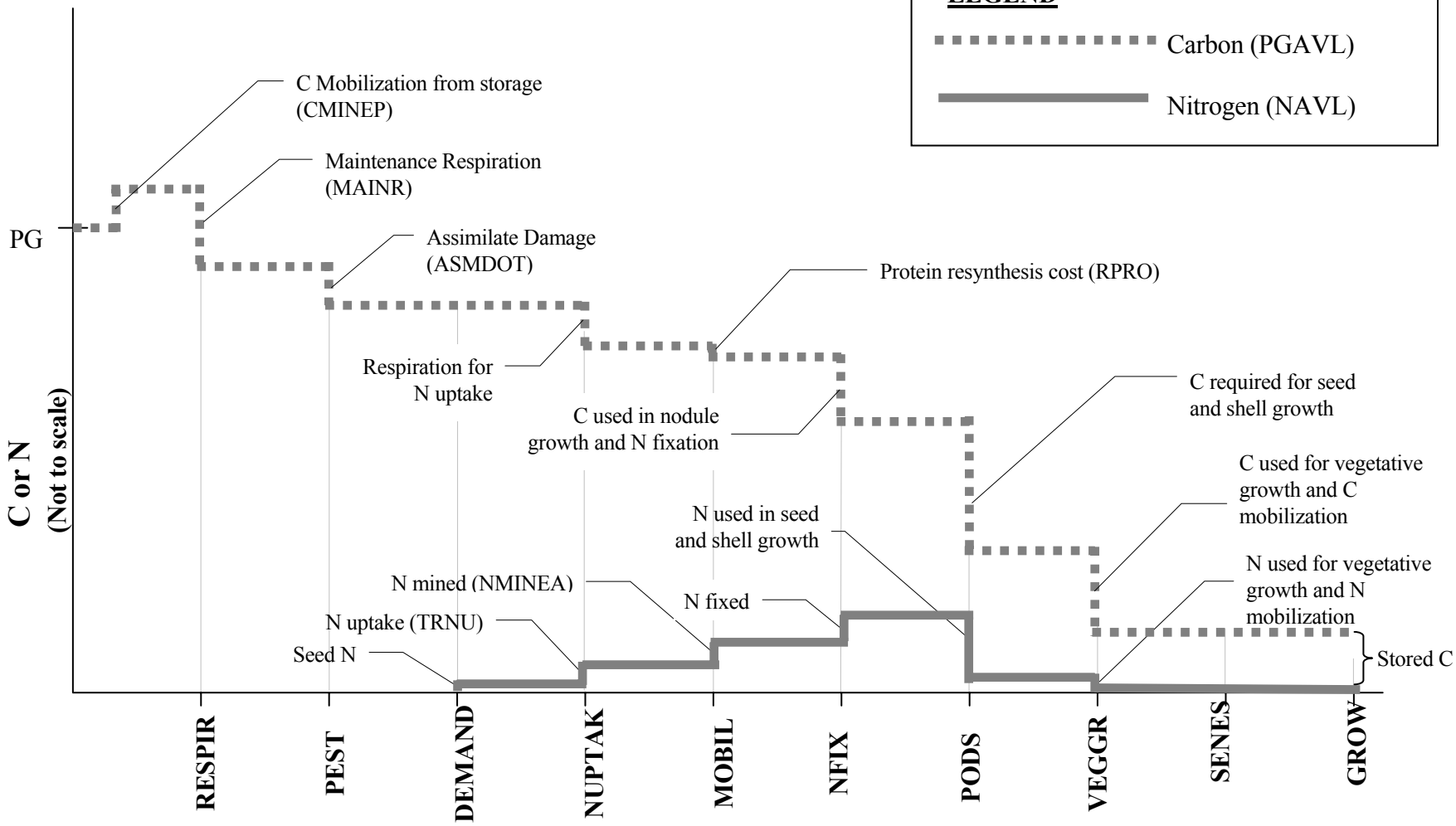


Figure 3 – Schematic of the flow of C (PGAVL) and N (NAVL)
The calculations for each process are shown in the order that they are computed through subroutine calls to appropriate modules.

($g[CH_2O]/m^2$) remaining after accounting for its use in various respiration and growth process calculations. The value of PGAVL changes as C is used in the various processes for synthesizing protein and growing tissue. It includes the day's gross photosynthesis plus mined C minus maintenance respiration. Normally, the amount of PGAVL is fully depleted for growth each day, except for a programmed amount that is stored as C reserves in vegetative tissues and which increases from flowering to beginning of rapid seed growth. Other exceptions include C storage if N is highly deficient or if insufficient pod and seed cohorts are growing.

The variable NAVL is the cumulative mass of N assimilated and not yet used in the sequence of growth calculations.

State variables are maintained for C storage in leaves, stems, roots, and shells, as WCRLF, WCRST, WCRRT, and WCRSH, respectively, with units of $g[CH_2O]/m^2$[ground area]. N state variables are WTNLF, WTNST, WTNRT, WTNSH, and WTNSD, for leaf, stem, root, shell, and seed, respectively.

At the start of the day, when CROPGRO is called, C reserves may be positive, and the values of PGAVL and NAVL are set to 0.0.

Potential C mobilization (CMINEP) is computed by multiplying the C stored in vegetative tissue by a maximum C mobilization fraction per day (CMOBMX). After plants begin their seed filling phase, the rate of C mobilization is increased in proportion to relative progress between first seed and maturity (DXR57). Mined C is computed by:

```
CMINEP = CMOBMX * (DTX + DXR57) * (WCRST + WCRRT + WCRSH + WCRLF)
```

The total C available for growth and respiration during the day (PGAVL) is the sum of the current day's photosynthesis and mined C:

```
PGAVL = PG + CMINEP
```

In the schematic in Figure 3, PGAVL is maximum at this point in the calculation sequence.

Maintenance respiration (MAINR) is then calculated in subroutine RESPIR. If maintenance respiration is greater than the day's photosynthesis plus mined C, then PGAVL is set to 0.0, meaning that no new growth will occur today. If maintenance respiration is less than PGAVL, then maintenance respiration is subtracted from PGAVL:

```
IF (MAINR .GT. PGAVL) THEN
   PGAVL  = 0.0
ELSE
   PGAVL = PGAVL - MAINR
ENDIF
```

Next, C losses to pest damage are accounted for. These losses are computed in the PEST damage module, and are expressed as C loss (ASMDOT, in units of $g[CH_2O]/m^2$-d). If assimilative damage has occurred today, then ASMDOT is subtracted from PGAVL using the following equations:

```
IF (ASMDOT .GT. 0.0) THEN
   PGAVL = PGAVL - ASMDOT
   PGAVL = MAX(0.0,PGAVL)
ENDIF
```

Next the DEMAND subroutine is called. This routine computes the C and N demands for reproductive and vegetative growth as well as respiration costs for the growth to occur. In DEMAND, calculations are made based on growth that will occur with today's PGAVL if seed and shells grow at their potential growth rates, and if the remaining C, if any, is entirely used for new vegetative tissue growth. The N demands are then used to set a limit on the amount of N that can be taken from the soil.

The first source of N available for tissue growth is from the seed itself, early in the season. The variable SDNPL is the total N available from the seed in addition to the N already in leaf, stem and roots at emergence, and it is assumed to become available during the first seven thermal days after emergence. Thus, NAVL is first computed in CROPGRO as:

```
NAVL = MAX(SDNPL * DTX / 7. , 0.0)
```

Next, N uptake is computed from the NUPTAK subroutine, if the switch for soil nitrogen balance is turned on (ISWNIT= 'Y'). Nitrogen uptake may be limited by the concentration of N in the soil, by root length density, or by N demand, as described in Section 8.3. In CROPGRO, the respiration C required for converting the N taken up from the soil into protein is then calculated and subtracted from PGAVL. Total N uptake, TRNU g[N]/m^2-d,N is composed of NO_3 (TRNO3U, g[NO_3-N]/m^2-d) and as NH_4 (TRNH4U, g[NH_4]/m^2-d). These N values are converted to g[protein] by dividing by 0.16, and respiration is calculated by multiplying by RNO3C and RNH4C, the g[CH_2O] required per g[protein] converted, assuming N as NO_3 or NH_4, respectively.

```
RSPNO3 = TRNO3U/0.16 * RNO3C
RSPNH4 = TRNH4U/0.16 * RNH4C
```

These respiration costs are then subtracted from PGAVL. In the equations below, if PGAVL is not sufficient for the respiration requirements, PGAVL is set to 0.0 and no growth occurs. In DEMAND, the demand for N is based on available C, so a shortage at this point should not occur.

```
IF (PGAVL .LT. (RSPNO3+RSPNH4)) THEN
   PGAVL = 0.0
ELSE
   PGAVL = PGAVL - (RSPNO3 + RSPNH4)
ENDIF
```

There should always be sufficient PGAVL for reducing the N uptake to protein because N uptake limits were based on the availability of C. The remaining PGAVL is calculated by the last equation above. In the calculation sequence, N uptake is computed after N mobilization, and NAVL is increased by the uptake amount, or:

```
NAVL = NAVL + TRNU
```

The next source of N for new growth is from N mobilized from vegetative tissue. Subroutine MOBIL is called to compute mined N (NMINEA, g[N]/m^2). Respiration cost for re-synthesizing protein from mined N (RPRO, g[CH_2O]/g[protein mined]) is much lower than that for N uptake. The following statements reduce PGAVL for this respiration cost. Mined N is computed as g[N]/m^2, so it is divided by 0.16 to convert it to g[protein]/m^2 and multiplied by the respiration cost, RPRO, to calculate the amount of PGAVL used.

```
IF (PGAVL .GT. NMINEA/0.16*RPRO) THEN
   PGAVL = PGAVL - NMINEA/0.16*RPRO
ELSE
   PGAVL = 0.0
ENDIF
```

Total N available for today's new growth is then initialized as:

```
NAVL = NAVL + NMINEA
```

thus, NAVL is increased while PGAVL is decreased.

This sequence will have the effect of allowing mobilization and soil N uptake to supply all of the plant needs if sufficient soil N and roots are available, thereby eliminating C allocation to nodules for fixing N. N fixation during a day is limited by the mass of nodules and its potential specific activity, the mass of CH_2O allocated to nodules, and environmental conditions surrounding the nodules.

After the amounts of N uptake and mining are computed, there still may be a shortage of N needed for tissue growth. In this situation, some N may be fixed, depending on the availability of C for the N fixation process, as well as the mass of nodules and soil conditions, as explained in the chapter on N fixation (NFIX).

How much C should be allocated to the nodules? Photosynthate C can reach the nodules within three to six hours. In reality, the allocation of C to nodules is an instantaneous process whereas we have to decide what proportion of the day's photosynthesis or total available C can be used for nodule growth and/or N fixation. The previous discussions specify the amount of C to be used in reducing N and the remaining amount that can be used for tissue growth, if sufficient N has been reduced. However, the amount of N that is available from uptake and mobilization may be less than that required to fully use available C. We assume that the C remaining after reducing N from the soil and existing tissue will be optimally used by the plant; that is, that C will be allocated to the nodules to fix just enough N to meet the remaining N needs, and the rest of the C will be used for tissue growth. Thus, C allocation to nodules can be computed based on this assumption. This is done in two steps, first calculating CH_2O allocated to nodules for fixing N needed for reproductive growth (CTONODR, g[CH_2O]/m^2-d), then adding that needed for fixing N for vegetative growth. Note, however, that actual N fixation may be restricted, even if sufficient C is allocated to nodules, because of limited nodule mass or soil environmental conditions.

If the demand for N for reproductive growth is larger than N uptake and mobilized in a day, then the amount that needs to be fixed is (NDMREP - NAVL), and the C required to fix this amount of N for reproductive growth (CTONODR) is:

```
CTONODR = MAX(0.0, (NDMREP-NAVL)*RFIXN/0.16)
```

The amounts of C (CAVVEG) and N (NAVLV) available for vegetative growth are computed by:

```
CAVVEG = MAX(0.,(PGAVL - CDMREP))
NAVLV = MAX(0.,(NAVL - NDMREP))
```

The C needed for fixing the unmet N requirements for vegetative growth (CNDFX) can be computed by:

```
CNDFX = MAX(0.,(RFIXN/0.16)*(NDMVEG-NAVLV))
```

If there is any C available for vegetative growth (CAVVEG > 0), and if the N needed for vegetative tissue growth is not met by N uptake and mobilization (CNDFX > 0), then the protein composition of growing vegetative tissue (PROVEG) is computed:

```
IF (CAVVEG .GT. 0.0 .AND. CNDFX .GT. 0.0) THEN
   PROVEG = PROLFI * FRLF + PRORTI * FRRT + PROSTI * FRSTM
```

Then, the amount of C to allocate to nodules to fix N needed for vegetative growth (CTONOD) is computed. The following equation was derived by assuming that all C would be used for vegetative tissue synthesis, and that a proportion of PGAVL would be used to fix N needed to meet the requirements for new vegetative tissue N composition.

```
   CTONOD = CAVVEG - (CAVVEG + (NAVLV*RFIXN/0.16)) *
         AGRVG / (AGRVG + PROVEG*RFIXN)
ENDIF
```

During vegetative growth (DAS < NR2), a certain fraction of C reserved for roots will be used for nodule growth. In the following code, FRCNOD is the fixed fraction of the C which is allocated to roots and is read from FILEC in the IPPLANT routine. CNODMN is the minimum carbon to be allocated to nodule growth.

```
IF (DAS .LT. NR2) THEN
  CNODMN = CAVVEG * FRRT * FRCNOD
ELSE
  CNODMN = 0.0
END IF
```

The total C to allocate to nodules (CTONOD, g[CH_2O]/m^2-d) can be computed by adding the C needed for reproductive growth to the above equation:

```
CTONOD = MIN(CNODMN + MAX(0.0, CTONOD), CAVVEG) + CTONODR
```

This approach is based on the hypothesis that the plants will allocate C optimally to grow tissue, including allocating what is necessary for fixing the N needed for growth. It should be noted that even though this amount of C is allocated for nodule use, it may not be used if nodule mass is small or if nodule activity is low because of environmental conditions. In such cases, some C will remain as surplus after growth, and will thus add to C reserves. If the plant is sufficiently rich in C reserves, then we also assume that any of the mined C can be used for

fixing N. During reproductive growth phases, N deficits will lead to a relative excess in C, which can then be used to fix N if nodule mass is sufficient.

The PGAVL actually used by the nodules for fixing N and for growth (CNOD) is not known until after the NFIX subroutine is called.

NFIX is called to compute N fixation and nodule growth, if the soil N switch (ISWNIT) and the symbiosis switch (ISWSYM) are equal to 'Y', and if plants have developed sufficiently. It is assumed in the model that the plants must develop to a specified vegetative stage (TTFIX) after emergence before nodule growth and N fixation will occur. This parameter is an input value, which may be 0.0 if the species is capable of fixing N as soon as it emerges. This subroutine computes the amount of N fixed during the day (NFIXN, g[N]/m^2-d).

If the soil N balance switch is on but the symbiosis switch (ISWSYM) is U, meaning unlimited N fixation potential depending on only the amount of C available, then the routine NFIX is not called. Rather, the amount of N fixation is computed as the demand for N not met by N mobilization plus N uptake (NAVL in the equation below), or:

```
NFIXN = MAX(0.0,NDMREP + NDMVEG - NAVL)
```

and the C required for fixing this N (CNOD) is:

```
CNOD = RFIXN * NFIXN/0.16
```

These last two equations are also used if the soil N routines are not used, and C drives the entire processes. In this case, NAVL at this point would be equal to the mobilized N, NFIXN would just be equal to N demand minus mobilized N, and the C required is calculated by the last equation, which is equivalent to the way that SOYGRO V5.42 computed C costs for acquiring N. This set of logic also allows the model to simulate non-nodulating or non-N-fixing crops if ISWSYM is not equal to either Y or U. In this case, mobilization and N uptake are computed, but N fixation is set to 0.0.

Next, the C and N temporary pools are updated. The C actually used in N fixation is subtracted from PGAVL, and NAVL is increased by the amount of N fixed.

```
IF (PGAVL .GT. CNOD) THEN
   PGAVL = PGAVL - CNOD
ELSE
   PGAVL = 0.0
ENDIF
NAVL = NAVL + NFIXN
```

At this point in the sequence, all N that will be available for both vegetative and reproductive growth is available from uptake plus mobilization plus fixation. And, C available for tissue growth has been computed, consisting of today's PG as well as mined C that remains after using some for respiration costs for protein synthesis and maintenance. In Figure 3, these values are shown schematically just before the tissue growth actually occurs in routines PODS and VEGGR.

Actual growth rates of seeds and shells are computed in the PODS subroutine. Small pods may

abort under conditions of limited carbon supply or pods may abscise after seeds mature using a routine that Batchelor (1994) developed for peanut. This subroutine (PODDET) is called when variable DETACH = 'Y' and, depending on parameters in the crop species file, it may cause cohorts of pods to abscise.

In the schematic, considerable amounts of both C and N are used in PODS to grow seeds and shells. Only a small amount of N may be left for vegetative growth. Calculations are made of C and N remaining after using some of each for seed and shell growth in PODS, first by computing the C required for seed (CGRSD) and shell (CGRSH) growth:

```
CGRSD = WSDDTN * AGRSD1
CGRSH = WSHDTN * AGRSH1
```

Then, PGAVL and NAVL left over after pod growth are computed,

```
IF (PGAVL .GT. (CGRSD + CGRSH)) THEN
   PGAVL = PGAVL - CGRSD - CGRSH
ELSE
   PGAVL = 0.0
ENDIF
NAVL   = NAVL - (NGRSD + NGRSH)
```

Most annual seed-producing plants accumulate increasing amounts of carbohydrates in vegetative tissues (primarily stems) during the transition from flowering (NR1) until rapid seed growth begins. A fraction of the PGAVL used for vegetative growth (CADPR1) is set aside for CH_2O storage (CSAVEV). The amount that is stored increases with progress from flowering through seed fill (FRACDN) and decreases the PG available for vegetative growth.

```
CSAVEV = CADPR1 * PGAVL * FRACDN
PGAVL = PGAVL - CSAVEV
```

The fractional value (CADPR1) can be calibrated to mimic the peak carbohydrate concentration found in stems prior to rapid seed fill. Carbohydrate mobilization occurs continuously and will normally deplete the C storage pool after vegetative growth ends and as seeds grow.

VEGGR is then called to compute vegetative growth for leaves (WLDOTN), stems (WSDOTN) and roots (WRDOTN). C required for leaf, stem and root growth, excluding the C required for synthesis of protein which has already been computed, is subtracted from PGAVL:

```
PGAVL = PGAVL - AGRVG*(WLDOTN + WSDOTN + WRDOTN)
```

Then, N requirements for vegetative growth are subtracted from NAVL:

```
NAVL = NAVL - (NGRLF + NGRST + NGRRT)
```

In the schematic, NAVL goes to 0.0, whereas PGAVL may be positive. This remaining PGAVL is added to reserve C pools for later use. The amounts of C and N that can be added to reserves are computed in VEGGR each day. Finally, these amounts are subtracted from NAVL

and PGAVL, after which the values of these variables are 0.0.

```
NAVL = NAVL - (NADLF + NADST + NADRT)
PGAVL = PGAVL - (CADST + CADLF) * PCH2O
```

CADST and CADLF are the amounts of CH_2O reserves added to stem and leaf, respectively.

The SENES subroutine is called to compute the senescence of vegetative tissue. The FREEZE subroutine is called if the temperature drops below a tolerance temperature. These routines do not affect the current day's photosynthesis or N uptake processes.

Subroutine ROOTS is called to compute root depth increase, proliferation of root length, and senescence.

Next, an average respiration cost, RPROAV (g[CH_2O]/g[protein]) is used to estimate growth respiration requirements for each type of tissue (seed, shells, and vegetation). Then, the total C required per g of new tissue growth, including C required for the protein in tissue, is computed for seed (AGRSD2), shell (AGRSH2) and vegetative tissue (AGRVG2):

```
AGRSD2 = SDPRO*RPROAV + PMINSD*RMIN + PLIGSD*RLIG + POASD*ROA
       + (PLIPSD*RLIP + PCARSD*RCH2O)*(1. - SDPROR)

AGRSH2 = PROSHI*RPROAV + PLIPSH*RLIP + PLIGSH*RLIG + POASH*ROA
       + PMINSH*RMIN + PCARSH*RCH2O

AGRVG2 = AGRVG + (FRLF*PROLFI+FRRT*PRORTI+FRSTM*PROSTI)*RPROAV
```

After all growth rates, C and N mining rates, vegetative and reproductive senescence rates, and pest damage rates are computed, they are passed to a subroutine called GROW, which integrates all processes and updates the values of the state variables at the end of the day.

3.6 Output

Subroutine OPGROW is called daily from the 'Output' section to report daily values to the PlantGro.OUT file. The PEST module is called daily to generate the PEST.OUT file.

3.7 Final

In the 'Final' section of the CROPGRO module, PEST and OPGROW are called to close output files after all seasonal simulations are complete.

4.0 Initial Composition (INCOMP)

The INCOMP routine is called to initialize parameters for composition, which vary with genotype at the beginning of each run. The routine reads input data that is required for these calculations in the 'Run Initialization' section of code, and performs the initialization computations in the 'Seasonal Initialization' section. Calling arguments for the INCOMP routine are listed in Table 17.

4.1 Run Initialization

In order to initialize composition variables, the INCOMP routine must read parameters from FILEC and FILEE. These input parameters, which are listed in Table 3 include proportions of leaf, stem, root, seed, shell and nodule tissue that is composed of protein, carbohydrate, lignin, lipid, mineral, and organic acids, as well as the respiration required for synthesizing these components. Definitions of these parameters and other variables used in the INCOMP routine are listed in Appendices Q and G.

4.2 Seasonal Initialization

The INCOMP routine computes initial values of the CH_2O requirements for new tissue growth for leaf, stem, root, nodule, shell and seed. The model estimates initial tissue composition of six types of compounds: protein, lipid, lignin, carbohydrate-cellulose, organic acids and minerals (Wilkerson et al., 1983 and Jones et al., 1989). Proportions of these compounds in each type of plant tissue were read from the species file (e.g., PLIPLF is proportion of lipids in new leaf tissue). Also, respiration cost for each tissue component (e.g., RLIP is the respiration required for synthesizing lipid structure) were obtained from the species file. The following statement computes the respiration cost for new leaf growth (AGRLF) based on these components:

```
AGRLF  = PLIPLF*RLIP + PLIGLF*RLIG + POALF*ROA + PMINLF*RMIN + PCARLF*RCH2O
```

(leaf synthesis cost) = (lipid respiration) + (lignin respiration) + (organic acid respiration) + (mineral respiration) + (carbohydrate respiration)

Similar computations are done for stem, root, nodule, shell and seed tissue to obtain AGRSTM, AGRRT, AGRNOD, AGRSH1, and AGRSD1. AGRVG is the summed respiration cost for all vegetative tissue including leaf, stem and root, where FRLF, FRRT and FRSTM are the partitioning of new growth to leaf, root and stem, respectively.

```
AGRVG  = AGRLF * FRLF + AGRRT * FRRT + AGRSTM * FRSTM
```

These calculations exclude the respiration cost of protein synthesis. The protein components are added into the respiration costs in variables AGRVG2, AGRSH2 and AGRSD2 for vegetative tissue, shell and seed, respectively. The calculation for AGRVG2 is shown below:

```
AGRVG2 = AGRVG + (FRLF*PROLFI+FRRT*PRORTI+FRSTM*PROSTI)*RNO3C
```

(proportion of protein in new vegetative tissue) (protein synthesis cost)

The variable SDPROR is used to adjust lipid and carbohydrate proportions when the seed

protein composition of the simulated cultivar differs from the protein composition of the standard cultivar. This variable is used by the INCOMP and GROW subroutines in the computation of respiration requirements for seed.

```
SDPROR = (SDPRO - SDPROS) / (SDLIP + PCARSD )
```

Table 3 – Input Parameters Read by INCOMP Routine

FILEC Parameters (e.g., SBGRO030.SPE):			
PCARLF	PLIGST	PMINSH	PRORTI
PCARNO	PLIPLF	PMINST	PROSHI
PCARRT	PLIPNO	POALF	PROSTI
PCARSD	PLIPRT	POANO	RCH2O
PCARSH	PLIPSH	POART	RLIG
PCARST	PLIPST	POASD	RLIP
PLIGLF	PMINLF	POASH	RMIN
PLIGNO	PMINNO	POAST	RNO3C
PLIGRT	PMINRT	PROLFF	ROA
PLIGSD	PMINSD	PROLFI	SDPROS
PLIGSH			
FILEE Parameters (e.g., SBGRO030.ECO):			
ECOTYP	SDLIP	SDPRO	

5.0 Maintenance Respiration (RESPIR)

The RESPIR subroutine is called once from the CROPGRO 'Rate/Integration' section to compute maintenance respiration (MAINR), used to account for tissue turnover and its re-synthesis. Maintenance respiration is comprised of two components, one is a function of total plant mass (WTMAIN) and the second is a function of plant activity represented by photosynthesis (PG). Both components have coefficients (RO and RP) that are dependent on temperature. The second part of the maintenance respiration assumes a proportion (RP) of the day's photosynthesis is respired in the maintenance process:

```
MAINR = RO * WTMAIN + RP * PG
```

Table 18 lists calling arguments used in the RESPIR routine.

6.0 Nutrient Demand (DEMAND, IPDMND and SDCOMP)

The daily allocation of C and N in the model is based on source-sink concepts. All references to C will imply units of g[CH_2O]/m^2 and references to N imply units of g[N]/m^2 unless otherwise stated.

Supply of C is from the current day's photosynthesis process and from mobilized C from reserves. Photosynthesis depends on canopy size, environmental conditions (light and temperature), soil water availability, and leaf characteristics, including leaf N concentration. Mobilized C depends on the availability of C stored in vegetative tissue as well as the availability of N for growing new tissue and the potential growth rates of the tissue.

Supply of N for new growth in the model is from NO_3 and NH_4 uptake from the soil, mobilized N from vegetative tissue, and symbiotic N fixation. N uptake from the soil depends on concentration and distribution of NO_3 and NH_4 in the soil, root length density and its distribution, and on the "demand" for N by the plant. Therefore, plant demand for N is a critical variable in determining the availability and use of N in the plant. In the model, N mobilization is computed first, followed by root uptake of N. N uptake from the soil can replenish the N mined from vegetative tissue up to a seasonally varying maximum concentration. If these sources of N satisfy the demand for N, then growth proceeds at its potential rate, using all of the day's available C, and no N fixation occurs. In such cases, all of the available C will be used with the N that is assimilated from the soil and mined from vegetative tissue for growth, leaving no C for N fixation. If there is a shortage of N, relative to that needed for potential growth, then some C is made available to nodules for N fixation. It is assumed that the plant will allocate sufficient C to nodules to meet the N demand for tissue growth in a day. Actual N fixation then depends on the mass of C allocated to nodules, nodule mass, the specific activity of nodules, and soil temperature and water status.

New plant tissue growth thus depends on the availability of both C and N substrate, the concentrations of both elements in the tissue products, and on the energy for tissue synthesis. The availability of N in the model also depends on the demand or sink strength of growing tissue and its N content. This DEMAND subroutine computes the potential tissue growth rates of different plant tissues as well as the C and N requirements for the new growth, assuming that growth will be at the potential rate. These "demands" for growth, C and N are passed back to the main subroutine, CROPGRO, where the N supply processes are considered while balancing the supply of substrate (C and N) with the actual growth.

Demand for plant tissue growth can be expressed as the number of growing organs multiplied by the rate of growth of each organ, summed for the entire plant. N demand can be expressed as this growth rate multiplied by the N concentration. C demand is computed by multiplying tissue growth rate by the overall C costs for conversion of photosynthate to final tissue product, which includes energy costs for synthesis as well as C for the tissue itself. This is generally the approach used in this model. However, some additional details are needed, and one significant deviation from this concept was necessary for this model. These details will be explained below.

Priorities for C and N Allocation. In this model, vegetative tissue growth is assumed to be limited by sink strength only until a specified number of main stem leaves (VSSINK) have appeared, after which time vegetative tissue growth depends on supply of substrate. Thus after "VSSINK" leaves have appeared, the plant is assumed to have the capacity to grow vegetative branches as well as main stem leaves and stems to utilize the C and N available for growth. This assumption allowed us to avoid

having to keep up with the number of vegetative branches, but it caused us to have to make some adjustment to the calculation of demand for C and N from the simple concept explained above.

Otherwise, the concept presented above was used. Seeds, then shells, in the model have first priority for both C and N, up to a limit of XFRUIT fraction of the day's available C. For determinate crops, XFRUIT is 1.0, and all C and N can go to fruit growth if there are sufficient numbers of seeds and shells growing and growth conditions are favorable. In subroutine DEMAND, C and N demands for seed and shell growth are computed first, assuming that there will be sufficient N to fully supply the growth needs. However, in the calculations of fruit demands, the availability of C is taken into account and the growth and N demands are limited to the available C that has already been computed (PGAVL). After fruit demands have been computed, if there is any C remaining, this C is then used to compute the vegetative growth that it could support if adequate supply of N is available. Thus, vegetative demands for C at this point are based on supply of C, and demands for tissue growth and for N depend on the fractions of leaves, stems, and roots growing, their N concentrations, and the C costs for synthesizing the tissue. Appendix Q lists all of the variables computed by subroutine DEMAND as well as other major variables that are needed in their computations. Variables and definitions are also listed at the end of the FORTRAN code in Appendix D.

The calling arguments used to invoke the DEMAND routine are listed in Table 19.

6.1 Run Initialization (Subroutine IPDMND)

The 'Run Initialization' section of DEMAND consists of a single call to the IPDMND subroutine, which reads all input parameters required for this routine. Parameters read are listed in Table 4 with definitions and units supplied in Appendices Q and D.

Table 4 – Input Parameters for DEMAND Routines

FILEIO Parameters (e.g. DSSAT40.INP):				
ECONO	SIZELF	SLAVAR	XFRUIT	
FILEC Parameters (e.g., SBGRO030.SPE):				
CARMIN	NVSMOB	RCH2O	SLAMIN	XSLATM(I)
FINREF	PLIGSD	RLIG	SLAPAR	XTRFAC(I)
FNSDT	PMINSD	RLIP	SLAREF	XVGROW(I)
FRLFF	POASD	RMIN	SLOSUM	XXFTEM(I)
FRLFMX	PROLFF	RNO3C	SRMAX	YLEAF(I)
FRSTMF	PROLFI	ROA	TURSLA	YSLATM(I)
LIPOPT	PRORTF	RPRO	TYPSDT	YSTEM(I)
LIPTB	PRORTI	SHLAG	VSSINK	YTRFAC(I)
NMOBMX	PROSTF	SIZREF	XFRMAX	YVREF(I)
NRCVR	PROSTI	SLAMAX	XLEAF(I)	YXFTEM(I)
FILEE Parameters (e.g., SBGRO030.ECO):				
ECOTYP	SDLIP	THRESH	LNGSH	SDPRO

6.2 Seasonal Initialization

'Seasonal Initialization' in the DEMAND routine includes setting initial values for variables at the beginning of each seasonal simulation, including initialization at zero for variables that must be calculated new for each new day's simulation. The variety-specific leaf parameters are set up under the supposition that the modeled crop is designed for a standard reference cultivar,

but that other cultivars may have higher or lower specific leaf area (SLA), or larger or smaller early leaves set. Variable DUMFAC is the ratio of the cultivar-specific value of specific leaf area (SLAVAR) to the crop species standard (SLAREF). The ratio, DUMFAC, is used to set the starting points for other parameters affecting specific leaf area (SLA) as it responds to irradiance from low values (SLAMAX) to high values (SLAMIN).

```
DUMFAC = SLAVAR / SLAREF
F      = DUMFAC * FINREF
FVEG   = DUMFAC * SLAMAX
SLAMN  = DUMFAC * SLAMIN
SLAMX  = DUMFAC * SLAMAX
GROMAX = 0.0
```

Likewise, SIZRAT is the ratio of variety-dependent leaf size (SIZELF) to the species-dependent leaf size (SIZREF). SIZELF and the ratio, thus allow the "sink-limited" rate of exponential leaf area growth for the first "VSSINK" leaves (YVGROW) to vary with variety.

```
SIZRAT = SIZELF / SIZREF

DO I = 1,6
  YVGROW(I) = SIZRAT * YVREF(I)
ENDDO
```

Initial partitioning of total plant dry matter to leaves, stems and roots (FRLF, FRSTM and FRRT) are set using a VSTAGE value of 0.0 and a look-up function. This function is used to compute vegetative partitioning based on VSTAGE and is discussed in detail in Section 7.4

```
FRLF = TABEX(YLEAF,XLEAF,0.0,8)
FRSTM = TABEX(YSTEM,XLEAF,0.0,8)
FRRT = 1.0 - FRLF - FRSTM
```

6.3 Emergence

The 'Emergence' section of the DEMAND code is used to initialize variables on the day of plant emergence. These include seed demand for growth variables (XFRT, ADDSHL, TURXFR, GDMSD, CDMSD, NDMSD, GDMSDR, CDMSDR, NDMSDR, CDMREP), photo-thermal time variables for cohort growth (PHTIM and PNTIM) and maximum fractions of N required for new tissue growth (FNINSD, FNINL, FNINS, FNINR).

6.4 Rate/Integration

To calculate C requirements, it is important to know how much of the N is supplied by mined N, as it costs less energy for tissue growing with mined N as compared with N uptake that must be reduced to protein groups. Thus, the maximum N mined is computed, based on the assumption that there is some maximum rate of N mining, and that this varies vith development. In the equations below, the potential N mining rate (NMOBR) is a fraction of the maximum value from emergence to first seed (before NR5), then it changes toward its maximum value after seeds start growing (at NR5), although the maximum rate is not achieved until XPOD approaches 1.0. , Potential N mobilized in a day (NMINEP) is computed as the potential N mining rate (NMOBR) multiplied by the stored N in leaves, stems, roots, and shells, as shown below:

```
      NMOBR  = NVSMOB * NMOBMX * TDUMX
      IF (DAS .GT. NR5) THEN
         NMOBR = NMOBMX * TDUMX2 * (1.0+0.5*(1.0-SWFAC))
     &      * (1.0+0.3*(1.0-NSTRES)) * (NVSMOB+(1.-NVSMOB)
     &      * MAX(XPOD,DXR57**2.))
      ENDIF
      NMINEP = NMOBR * (WNRLF + WNRST + WNRRT + WNRSH)
```

Note that NVSMOB is the fractional component of NMOBMX occurring during vegetative growth, whereas (1 – NVSMOB) is the complementary added fraction increasing during seed fill as XPOD gradually increases.

When DEMAND is first called from subroutine CROPGRO, the mass of C available for growth within the day has already been computed as PGAVL. The top sections of DEMAND compute the demands for reproductive tissue growth, the C and N for this growth, and other factors needed for the reproductive component. The first section actually deals with the physiological ages of the different cohorts of flowers (PNTIM) and pods (PHTIM). See Appendix Q for definitions of these and other variables used below. Also, see the PODS and PODCOMP descriptions for a more detailed discussion of how reproductive growth is handled in the model. These physiological times, in photothermal days, are set on each day that new flowers or pods appear so that the physiological age of any cohort can be computed. The daily photothermal time accumulator, TDUMX, is computed in the phenology module, and the photothermal time for a cohort of flowers is established as:

```
      IF (DAS .EQ. NR1) THEN
        PNTIM(1) = 0.0
      ELSE
        PNTIM(DAS - NR1 + 1) = PNTIM(DAS - NR1) + TDUMX
      ENDIF
```

Similarly, the photothermal time when a cohort of pods formed is established as:

```
      IF (DAS .LE. NR2) THEN
        PHTIM(1) = 0.0
      ELSE
        PHTIM(DAS - NR2 + 1) = PHTIM(DAS - NR2) + TDUMX
      ENDIF
```

Pegs (gynophores) first appear on the plant at the R2 stage, designated by the variable NR2 in the model. For peanuts, this stage is the time of first peg, whereas for soybean and dry bean, NR2 is skipped, going directly to the NR3 stage, which is the day when 50% of the plants have a beginning pod size greater than 0.5 cm in length. In actuality, there may be a minor lag in the model depending on time to grow from zero to the mass required for a 0.5 cm pod.

One can compute the physiological age of a group of pods on day DAS for the group of pods formed "NPP" days after the first pod by the following:

```
   PAGE = PHTIM(DAS-NR2+1) - PHTIM(NPP)
```

Physiological ages of pods are used to determine the initiation of seeds and their potential growth rates.

Next in DEMAND, potential seed growth rate for the day is computed based on the day's temperature, taking into account any damage by pests to seed that would reduce their individual potential growth rates. First, the temperature growth factor is computed, using hourly temperatures (TGRO(I)) in an equation defined by TYPSDT and FNSDT.

```
      TMPFAC = 0.
      TMPFCS = 0.
      DO I = 1,24
        TMPFAC =CURV(TYPSDT,FNSDT(1),FNSDT(2),FNSDT(3),FNSDT(4),
     &       TGRO(I))
        TMPFCS = TMPFCS + TMPFAC
      ENDDO
      TMPFAC = TMPFCS / 24.
```

TYPSDT defines the type of curve (i.e., quadratic, linear, etc.), while the values of FNSDT define points on the curve. The returned value of TMPFAC is a value between zero and one. Table 5 shows the returned values of CURV with actual calculated values being dependant on the type of curve and current temperature values.

Table 5 – TMPFAC calculations

Temperature Value	TMPFAC Value
TEMP < FNSDT(1)	TMPFAC = 0.0
FNSDT(1) < TEMP < FNSDT(2)	0.0 < TMPFAC < 1.0
FNSDT(2) ≤ TEMP ≤ FNSDT(3)	TMPFAC = 1.0
FNSDT(3) < TEMP < FNSDT(4)	0.0 < TMPFAC < 1.0
TEMP > FNSDT(4)	TMPFAC = 0.0

Figure 4 shows this relationship which is based on a quadratic curve with FNSDT values of 6, 21, 23.5, and 41 °C. For this day of simulation, temperatures vary from a low of 21.9 to a high of 31.1 °C. For temperatures above 23.5 °C, the calculated TMPFAC is below 1.0, with an average daily TMPFAC of 0.940. Likewise, calculated TMPFAC would be less than 1.0 for times when instantaneous temperature was less than 21°C.

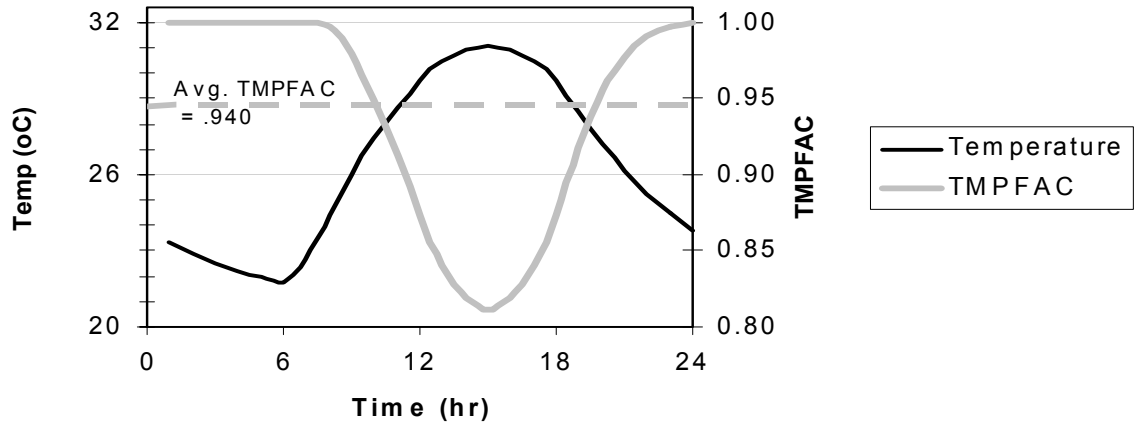


Figure 4 – Effect of temperature on seed growth rate

A water stress factor, TURADD, is used to affect reproductive growth and pod addition. Stress is defined to increase growth and addition.

```
      TURADD = TABEX (YTRFAC,XTRFAC,TURFAC,4)
```

The potential growth rate per seed (SDGR g/seed-d) is computed by multiplying the maximum cultivar-dependent seed growth rate (SDVAR) by the temperature factor, the puncture damage factor (not currently used in CROPGRO), a day length factor (DRPP), and a water stress factor (TURADD). (The latter is presently of value only for chickpea.) SRMAX is the maximum fractional decrease in seed growth rate for long daylengths, and the DRPP is the reduction in the rate of development on the current day due to non-optimum day length.

```
      SDGR = SDVAR * TMPFAC * REDPUN * (1.-(1.-DRPP)*SRMAX) *
     &       (1 + TURADD)
```

This potential seed growth rate is assumed to be the same for all seed that are still growing. Each age group of pods must be considered, however, to add up the total seed growth potential, taking into account pods that do not yet have seed (those younger than LAGSD photothermal days in age) and those that have mature seed. It is assumed that individual seed stop growing when a threshing fraction is reached, or when seed mass exceeds a specified percentage of seed plus pod mass, designated by THRESH, the maximum threshing percentage. Thus, a "do-loop" is used to examine each cohort of pods, first determining if the cohort is old enough to have seeds, then to ensure that it has not yet filled the shell cavity. A maximum amount of remaining growth, SDMAX, is computed for each cohort, taking into account any shell mass loss due to pest damage. Finally, the growth demand over all seed cohorts is accumulated as GDMSD. This is done with the code below.

```
DO NPP = 1, DAS - NR2
   PAGE = PHTIM(DAS - NR2 + 1) - PHTIM(NPP)
   IF (PAGE .GE. LAGSD) THEN
      REDSHL = 0
      IF (SDDES(NPP).GT.0) THEN
         REDSHL = WTSHE(NPP)*SDDES(NPP)/(SDDES(NPP)+SDNO(NPP))
      ENDIF
      SDMAX = (WTSHE(NPP)-REDSHL)*THRESH/(100.-THRESH)-WTSD(NPP)
      SDMAX = MAX(0.0,SDMAX)
      GDMSD  = GDMSD  + MIN(SDGR*SDNO(NPP)*REDPUN,SDMAX)
   ENDIF
ENDDO
```

Subroutine SDCOMP is called to compute seed composition for oil and protein. SDCOMP returns carbohydrate requirements for seed growth (AGRSD1 and AGRSD2), fraction of N required for seed growth (FNINSD), and potential carbohydrate and lipid composition of seed (POTCAR and POTLIP).

After the total seed growth demand is computed, demands for N and C are computed. FNINSD is the fraction seed N. N demand for the growing seed is thus:

```
      NDMSD  = FNINSD * GDMSD
```

If mined N is sufficient to meet the N needs of the seed, then NDMSDR, the N to be used from

mobilized sources, is set equal to the demand, NDMSD. Otherwise, NDMSDR is equal to the maximum N mining rate, NMINEP. Then, growth using mined N, GDMSDR, is computed so that C cost for this tissue synthesis can be calculated separately from the cost for tissue that is grown with a new source of N.

```
      IF (NDMSD .GT. NMINEP) THEN
        NDMSDR = NMINEP
      ELSE
        NDMSDR = NDMSD
      ENDIF
      GDMSDR = NDMSDR/FNINSD
      CDMSDR = GDMSDR * (AGRSD1 + FNINSD*6.25 * RPRO)
```

If there is any remaining growth demand, for which the mined N is not sufficient, then C required for this growth is `AGRSD2*(GDMSD-GDMSDR)`, and the total C required for seed growth is:

```
      CDMSD = (MAX(0.0,(GDMSD - GDMSDR))) * AGRSD2 + CDMSDR
```

At this point in the calculations, potential seed growth, and the demands for C and N, do not take into account the limitation of C. This will be computed after the demands for shell growth are computed.

Potential Shell Growth, C, and N Demands. Shells are divided into classes that determine their potential growth rates. When shells are less than LNGPEG photothermal days in age, they are assumed not to grow in mass. After they are LNGPEG p-t-d in age and until they are LNGSH p-t-d in age, they are assumed to grow at a rate that depends on temperature and daylength. Potential dry weight growth of shells (GRRAT1 g/shell/d) is first computed assuming that each growing shell that is not damaged grows at its maximum rate:

```
      GRRAT1 = SHVAR * TMPFAC * (1.- (1.-DRPP) * SRMAX)
     & * (1.0 + TURADD)
```

These potential growth calculations are carried out inside a loop to examine each cohort to see how old it is, and accumulate growth demand over all cohorts. In the code below, cohorts are checked to determine if their physiological age is younger than LNGPEG, between LNGPEG and LNGSH or older than LNGSH photothermal days. Between LNGPEG and LNGSH p-t-d of age, shells are assumed to grow at a constant rate of GRRAT1, whereas they grow at a fraction (SHLAG) of this rate before they reach LNGSH p-t-d age, and do not grow at all after LNGSH p-t-d age.

```
      DO NPP = 1,DAS - NR2
         NAGE = DAS - NR2 + 1 -NPP
         PAGE = PHTIM(DAS - NR2 + 1) - PHTIM(NPP)
         IF (PAGE .LE. LNGSH .AND. SHELN(NPP) .GE. 0.001 .AND.
&           GRRAT1 .GE. 0.001) THEN
            IF (PAGE .GE. LNGPEG) THEN
               ADDSHL = GRRAT1 * SHELN(NPP)
            ELSE
               ADDSHL = GRRAT1 * SHELN(NPP) * SHLAG
            ENDIF
         ENDIF
```

```
      GDMSH  = GDMSH + ADDSHL
ENDDO
CDMSH = GDMSH * AGRSH2
```

The total growth demand for shells is accumulated in the above loop, and now the C costs for this growth can be computed as (CDMSH):

```
CDMSH = GDMSH * AGRSH2
```

before taking into account the limitation of available C in the current day.

Adjusting Seed and Shell Demands Due to Limited C. Now, we calculate how much C can be used for fruit growth so that we can determine how much is potentially available for vegetative growth. The cultivar-specific parameter, XFRUIT, specifies the maximum partitioning of C into reproductive growth, thus (1-XFRUIT) * PGAVL of C will be available for vegetative growth. However, partitioning to fruit varies with environmental conditions, particularly with temperature and water stress, so factors for modifying XFRUIT are computed. The temperature factor (TEMXFR) is based on a table look-up function, called TABEX, in which a pair of x-y values for temperature (XXFTEM) and the factor (YXFTEM) are provided and a value is returned based on the air temperature (TGRO(I)). Hourly values of this function are computed and a 24-h average is used to adjust partitioning.

```
TEMXFR = 0.
DO I = 1,24
  TEMXFR = TEMXFR + TABEX(YXFTEM,XXFTEM,TGRO(I),6)
ENDDO
TEMXFR = TEMXFR/24.
```

Partitioning to pods is increased under drought stress conditions depending on XFRMAX, an input parameter, and TURFAC, the water stress variable. XFRT is thus computed from the standard partitioning, XFRUIT, and the temperature and water stress factors. It is restricted to a minimum of 0.0 and a maximum of 1.0.

```
TURXFR = XFRMAX * (1. - TURFAC)
TURXFR = MIN(TURXFR,1.0)
TURXFR = MAX(TURXFR,0.0)

XFRT = XFRUIT * TEMXFR + XFRUIT * TURXFR
XFRT = MIN(XFRT,1.0)
XFRT = MAX(XFRT,0.0)
```

After partitioning is computed, we can calculate how much C is available for reproductive growth (CAVTOT). CDMREP is the total C needed for potential reproductive growth.

```
CAVTOT = PGAVL * XFRT
CDMREP = CDMSH + CDMSD
```

If C demand for growth is greater than C available for growth, C demand and growth demands are reduced to balance C supply with demand. The two "IF" statements below determine whether all C is needed for seed growth. If it is, calculations of C, N and growth demands for seed are adjusted.

```
   IF (CDMREP .GT. CAVTOT) THEN
      IF (CDMSD .GT. CAVTOT) THEN
         CDMSH = 0.0
         GDMSH = 0.0
         CDMSD = CAVTOT
         IF (CDMSDR .GT. CAVTOT) THEN
            CDMSDR = CAVTOT
         ENDIF
         GDMSD = (MAX(0.0,(CDMSD-CDMSDR)))/AGRSD2 +
&             CDMSDR/(AGRSD1+FNINSD*6.25*RPRO)
         NDMSDR = GDMSDR * FNINSD
```

In the next section, not enough C is available for shells to grow at their potential rates. Thus, C, N and growth demands for shells are computed based on supply of C to shells.

```
      ELSE
         CDMSH = CAVTOT - CDMSD
         GDMSH = CDMSH/AGRSH2
```

The C allocated for seed and shell growth are summed to compute C demand for reproductive tissue (CDMREP).

```
      CDMREP = CDMSD + CDMSH
   ENDIF
```

Next, N demands for seed, shell, and total reproductive growth are computed.

```
      NDMSD  = GDMSD * FNINSD
      NDMSH  = GDMSH * FNINSH
      NDMREP = NDMSD + NDMSH
```

The total C available for growth in the current day was PGAVL. Now, we have computed the demand for reproductive growth, and that remaining for vegetative growth is PGAVL - CDMREP.

Vegetative Growth, C and N demands. Demands for vegetative tissue growth and N depend on the C remaining for vegetative growth, and on the costs of growing the tissue. A check is made to see if new vegetative tissue can be grown, using PGAVL. If so, C demand for vegetative growth (CDMVEG) is computed. Note in the equations below that at least (1-XFRT) fraction of the C is available for vegetative tissue plus any C remaining from reproductive growth will be used for vegetative tissue growth. This is important in computing the demand for N.

```
   CDMVEG = MAX(0.0,(1.-XFRT)*PGAVL)
   NDMVEG = 0.0
   CDMVEG = (PGAVL * XFRT - CDMREP) + CDMVEG
```

Partitioning among leaves, stems and roots varies with stage of growth, so these partitioning factors are computed using a table look-up function. Figure 5 shows the relative partitioning between leaf, stem and root components used for soybean in CROPGRO, prior to NR1, or first

flower growth stage.

```
FRLF  = TABEX(YLEAF,XLEAF,VSTAGE,8)
FRSTM = TABEX(YSTEM,XLEAF,VSTAGE,8)
```

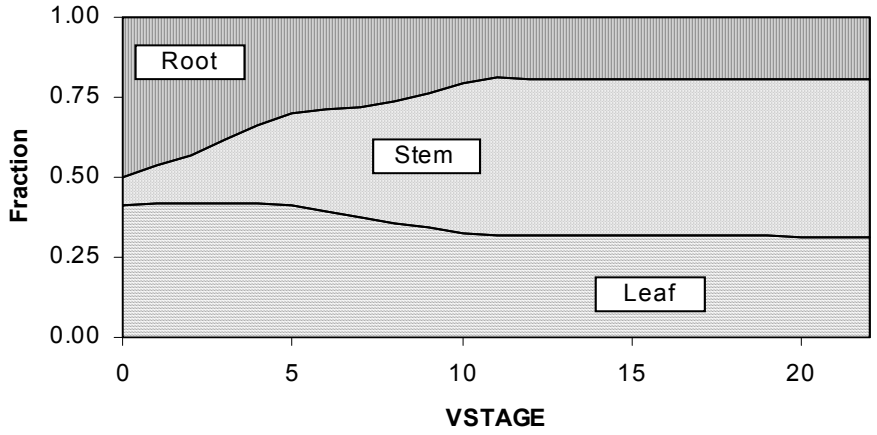


Figure 5 – Relative partitioning to leaf, stem and roots prior to first flowering

The partitioning varies in a linear projection from what it is at first flowering (NR1) to a set of partitioning values at the time of the last leaf appearance (NDLEAF). The "final" partitioning values are defined by FRSTMF and FRLFF, which are input from the species file. This transition is accomplished in the code by remembering the partitioning to leaf (FRLFM) and stem (FRSTMM) at flowering, as well as the amounts to change (YY and XX for leaves and stems, respectively).

```
FRLFM  = TABEX(YLEAF,XLEAF,VSTAGE,8)
FRSTMM = TABEX(YSTEM,XLEAF,VSTAGE,8)
YY = FRLFM - FRLFF
XX = FRSTMM - FRSTMF
```

Then, partitioning between vegetative tissues depends on development as expressed by FRACDN, the relative development between NR1 and NDLEAF, using the following:

```
      FRLF = FRLFM - YY * FRACDN
      FRSTM = FRSTMM - XX * FRACDN
      IF ( DAS .GE. NDLEAF) THEN
         FRLF = FRLFF
         FRSTM = FRSTMF
      ENDIF
   ENDIF
   FRRT = 1 - FRLF - FRSTM
```

When the plants are young (VSTAGE is less than VSSINK), then leaf area is assumed to follow a genetically determined pattern of expansive growth vs. thermal time. The specific leaf

area (SLA) is a combined effect of the environmentally determined SLA and the cumulative mass allocated to leaves. After this point, it is assumed that the plants can branch and then the leaf area expansion is source limited. Relationships for these two cases are given below.

After the VSTAGE = VSSINK is reached, leaf area expansion depends on the dry weight growth, and on an empirically computed SLA. A standard SLA is input as species (SLAREF) and cultivar (SLAVAR) parameters with a species maximum (SLAMX) in low light and minimum (SLAMN) in high light. This value is modified by functions that depend on PAR and temperature. The relationships for the irradiance effects were developed from the data of Cox (Peanut Sci. 5:27, 1978). Figure 6, Figure 7, and Figure 8 and show values for modifying FVEG under different PAR and temperature values. This approach results in thin leaves when light is low, and thick leaves under high light and/or low temperature conditions. The parameters used in the PARSLA equation were for soybean and are SLAMN = 250 cm^2/g, SLAMX = 950 cm^2/g, and SLAPAR = -0.048.

Light, temperature and water stress jointly modify FFVEG with the following code, with water stress acting via the variable TURFAC:

```
TPHFAC = 0.
DO I = 1,24
   TPHFAC = TPHFAC + TABEX (YSLATM,XSLATM,TGRO(I),5)
ENDDO
TPHFAC = TPHFAC/24.
PARSLA = (SLAMN+(SLAMX-SLAMN)*EXP(SLAPAR*PAR))/SLAMX

FFVEG = FVEG * TPHFAC * PARSLA * MAX(0.1, (1.0 - (1.0 - TURFAC)*TURSLA))
F = FFVEG
```

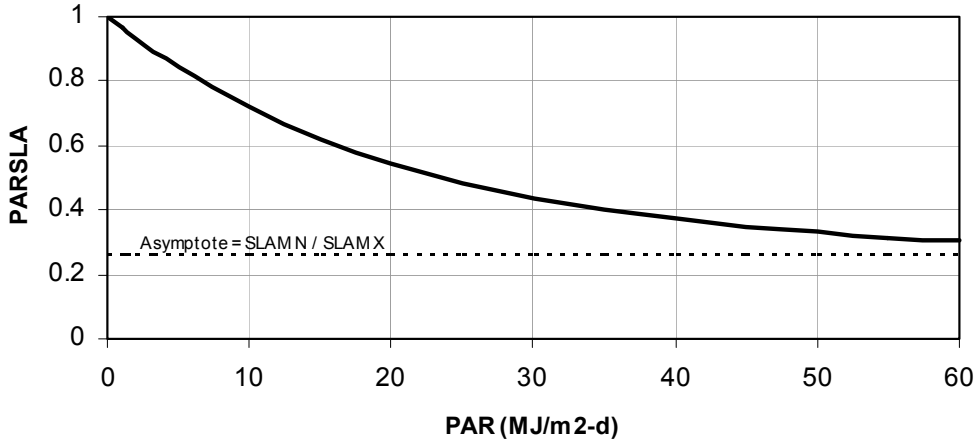


Figure 6 – Effect of PAR on specific leaf area (F) using variable PARSLA.
As PAR increases, PARSLA decreases, F decreases proportionally, indicating thicker leaves.

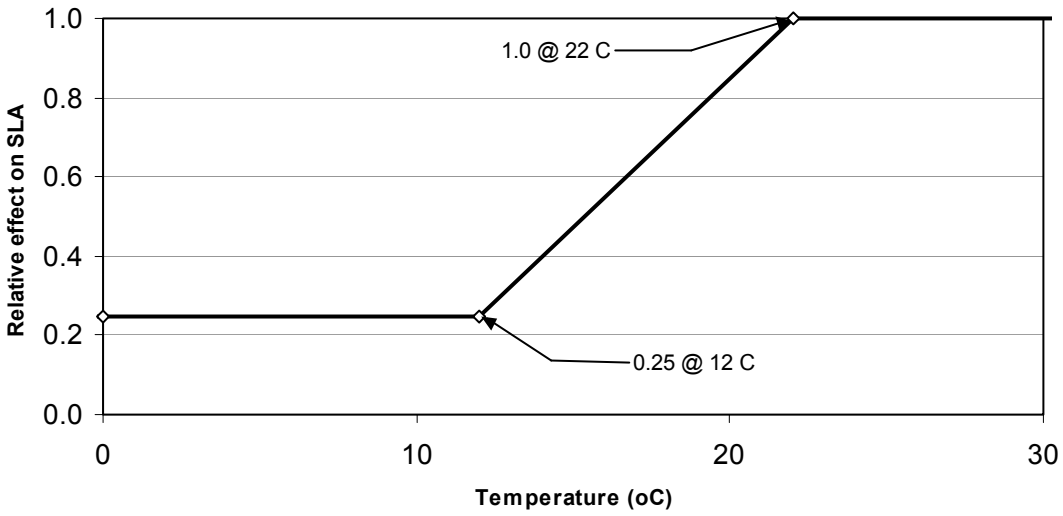


Figure 7 – Relative effect of temperature on specific leaf area
Lower temperatures will reduce TPHFAC below 1.0 resulting in proportionately lower specific leaf area indicating thicker leaves.

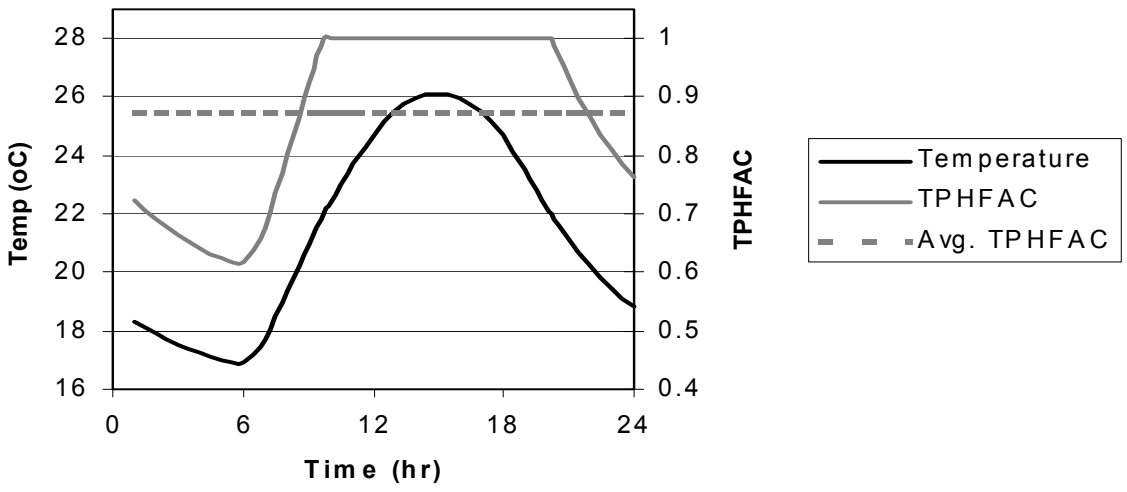


Figure 8– Effect of temperature on specific leaf area using variable TPHFAC

After flowering, the SLA values are allowed to decrease in proportion to development toward the end of leaf expansion, NDLEAF. FRACDN is the fraction of development between flowering and last leaf, and any leaf expansion during this time will occur as thicker leaves, using the following equation:

```
IF (XFRT*FRACDN .GE. 0.05) F = FFVEG * (1.0 - XFRT * FRACDN)
IF (XFRUIT .EQ. 1. .AND. DAS .GE. NDLEAF) F = 0.0
```

In this equation, XFRT is the maximum partitioning to reproductive growth. In determinate plants, this is 1.0, and the F will go to 0.0 at the time of the last leaf appearance. This will cause any additional leaf mass growth to thicken leaves instead of growing new leaf area.

The potential area of individual plants vs. VSTAGE is an input in FILEC (e.g., *.SPE file) for each crop. Figure 9 shows the exponential curve used for soybean until VSTAGE is 5.0 for a reference soybean variety. Values from this curve are linearly interpolated on any day using a Table look-up function (TABEX). Values are modified for different varieties based on a coefficient that specifies the maximum leaf size for each variety (SIZELF). GROMAX is the leaf area that will be produced for each VSTAGE, and the rate of leaf expansion per plant (GAINNW) is computed by taking the value today and subtracting the value determined the previous day. This per plant value is multiplied by the plant population to obtain canopy leaf area expansion:

```
IF (VSTAGE .LT. VSSINK) THEN
   GROYES = GROMAX
   GROMAX = TABEX(YVGROW,XVGROW,VSTAGE,6) * SIZELF / SIZERF
   GAINNW = (GROMAX - GROYES) * PLTPOP
```

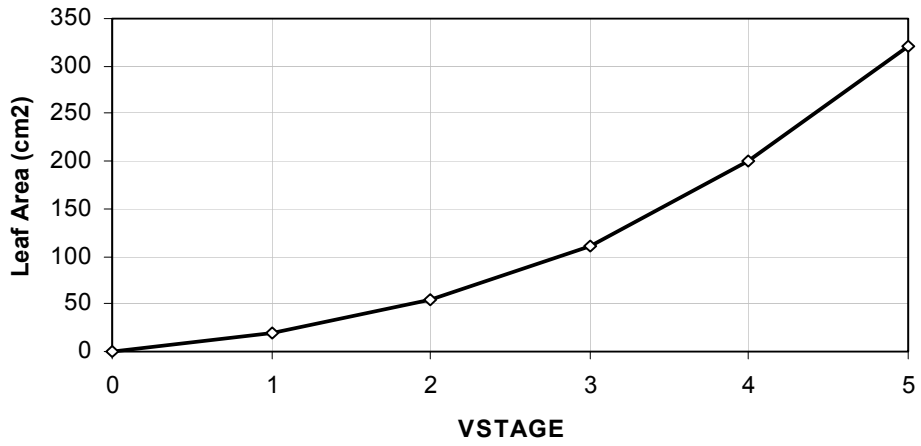


Figure 9 – Leaf Area Expansion vs. VSTAGE during Early Growth Stage

GAINWT represents an increase in leaf mass and is based on the potential expansion (GAINNW) divided by the leaf area per mass (F).

```
GAINWT = GAINNW/F
```

When VSTAGE is less than VSSINK, the partitioning of dry matter to leaves (FRLF) is set by the potential leaf area per V-stage and the environmentally determined specific leaf area. FRRT and FRSTM are then computed by leftover C. When an upper limit of dry matter goes to leaves (FRLFMX), leaf area expansion is restricted so that the specific leaf area of new growth (F) is maintained as computed and minimal amounts of C are partitioned to stem (FRSTM) and root (FRRT). This is accomplished by the following code:

```
      FRLF = (AGRLF*GAINWT)/(CDMVEG + 0.0001)
      IF (FRLF .GT. FRLFMX) THEN
         GAINWT = (CDMVEG/AGRLF) * FRLFMX
         GAINNW = GAINWT * F
         FRLF = FRLFMX
      ENDIF

      FRSTM = (1. - FRLF) * FRSTM / (FRSTM + FRRT)
      FRRT  = 1. - FRLF - FRSTM
   ENDIF
```

The VSTAGE is greater than VSSINK for most of the season, during which the model is source driven and leaf area growth depends on photosynthesis (PG), partitioning to leaf not subject to leaf area constraints (FRLF) and environmentally determined specific leaf area (SLA).

Next, the model computes the CH_2O cost per mass of vegetative tissue, excluding cost for protein (AGRVG), and the total CH_2O cost per mass of vegetative tissue (AGRVG2).

```
      AGRVG = AGRLF * FRLF + AGRRT * FRRT + AGRSTM * FRSTM
      AGRVG2 = AGRVG + (FRLF*PROLFI+FRRT*PRORTI+FRSTM*PROSTI)*RPROAV
```

N demand for new vegetative tissue growth (NDMVEG) is computed next.

```
      NDMVEG = (CDMVEG/AGRVG2) * (FRLF*FNINL+FRSTM*FNINS+FRRT*FNINR)
```

N demand for all new tissue, including reproductive and vegetative growth, is then computed:

```
      NDMNEW = NDMREP + NDMVEG
```

A check is then made to see if any C is left over after reproductive growth. If all C is used for reproductive growth, none is available to reduce N for adding it to existing tissue. If C is available (CNOLD>0), it may be used in reducing N to re-fill old tissue. It is assumed that only N taken from the soil can be used to re-fill N that was previously mined from vegetative tissues. We further assume that the vegetative tissue N (NVSTL, NVSTS, and NVSTR) can be re-filled to the maximum N content (FNINL, FNINS, and FNINR).

```
   IF (DAS .GT. NVEG0 .AND. DAS .LT. NR7 .AND. CNOLD .GT. 0.0) THEN
      NVSTL = FNINL
      NVSTS = FNINS
      NVSTR = FNINR
```

After seeds start to grow, the maximum tissue N contents decline with progress toward

physiological maturity. Variable FRNLFT represents a quadratic function of the progress from NR5 to NR7 (DXR57) and is used to decrease the tissue N concentrations below the maximums. Variable NCRVR is the minimum fraction applied to N concentrations at physiological maturity (DXR57 = 1.0).

```
      IF(DXR57 .GT. 0.0) THEN
         FRNLFT = NRCVR + (1 - NRCVR)*(1-DXR57**2)
         NVSTL = PROLFF*0.16 + (FNINL-PROLFF*0.16) * FRNLFT
         NVSTS = PROSTF*0.16 + (FNINS-PROSTF*0.16) * FRNLFT
         NVSTR = PRORTF*0.16 + (FNINR-PRORTF*0.16) * FRNLFT
      ENDIF
```

C must be available for this to occur. This N demand is referred to as NDMOLD. The fraction of N in each plant tissue is subtracted from its time-dependent maximum value and multiplied by tissue weight to compute N that could be added to existing tissue (NDMOLD). NDMOLD is limited to the amount of N that could be reduced by the available C.

```
      NDMOLD = (WTLF  - WCRLF) * MAX(0.0,(NVSTL - PCNL /100.))
&            + (STMWT - WCRST) * MAX(0.0,(NVSTS - PCNST/100.))
&            + (RTWT  - WCRRT) * MAX(0.0,(NVSTR - PCNRT/100.))

      IF (NDMOLD .GT. (CNOLD/RNO3C*0.16)) THEN
         NDMOLD = CNOLD/RNO3C*0.16
      ENDIF
   ENDIF
```

Total Growth, C and N Demands. Total N required is computed as the sum of reproductive and vegetative demands for N, including any going to fill old vegetative tissue. This value (NDMTOT) will be used in later calculations, controlled from CROPGRO, to compute N uptake, N mobilization, and N fixation by the plant on the current day.

```
     NDMTOT = NDMREP + NDMVEG + NDMOLD
     CDMTOT = CDMREP + CDMVEG + NDMOLD*RNO3C/0.16
```

The last statement computes the total C demand for all vegetative and reproductive growth. At this point, the C is balanced, in that no more is required than will be available for growth during the day. We do not know for sure how the C will be used, because N may be limited, and some C could be stored instead of being used for growth. Finally, at this point, we have not balanced N, but we have only computed how much is needed for optimum growth, assuming that C will be used for respiration as N is acquired. Limitations in N uptake, N mining, and N fixation can create a shortage of N in the plant relative to the supply of C, and thus relative to the amount of N demand that was calculated. In such cases, partitioning will be changed to roots, concentrations of N in vegetative tissue will begin to drop, and C will build up in the plant.

6.5 Seed Composition (SDCOMP)

The SDCOMP subroutine is called once from DEMAND during 'Rate/Integration' and is used to compute seed composition for oil and protein.

The lipid percentage is assumed to increase with temperature as a quadratic-plateau (Piper, 1993). The lipid content is computed based on user-supplied input of temperature above which

lipid composition is at a maximum (LIPOPT) and temperature below which lipid composition is zero (LIPTB). LIPTEM is a factor that reduces the maximum seed lipid concentration to a temperature related function, and is 1.0 when the average temperature (TAVG) is above LIPOPT, 0.0 when TAVG is below LIPTB and computed by a quadratic function when TAVG is between LIPOPT and LIPTEM.

```
IF (TAVG .GE. LIPOPT) THEN
   LIPTEM= 1.0
ELSEIF ((TAVG .LT. LIPOPT) .AND. (TAVG .GT. LIPTB)) THEN
   LIPTEM = 1.0 - ((LIPOPT - TAVG) / (LIPOPT - LIPTB))**2
ELSE
   LIPTEM= 0.0
ENDIF
```

The potential lipid composition (POTLIP) is computed to be:

```
POTLIP = SDLIP * LIPTEM
```

where SDLIP is the maximum genotypic lipid proportion in seed under optimum conditions.

Data across a wide range of temperatures indicate that the sum of lipid and protein proportions in seed has a linear relationship with temperature (Wolf et al., 1982 and Sato and Ikeda, 1979). This is depicted in Figure 10 and computed with the following code:

```
GENSUM = (SDPRO*100.0) +
&    SDLIP*100.0 * (1.0 - (((LIPOPT - 25.) / (LIPOPT - LIPTB))**2))
SUMTEM = 1.0 + SLOSUM * (TAVG - 25.0)
PSUMSD = GENSUM * SUMTEM / 100.0
POTPRO = PSUMSD - POTLIP
```

where GENSUM = protein plus lipid composition in seed at 25 °C (%)
SDPRO = Seed protein fraction at 25 °C (g[protein] / g[seed])
SUMTEM = Linear relationship between temperature and protein plus lipid compostion, normalized to 25°C
SLOSUM = rate of change of protein plus lipid composition with temperature, normalized to value at 25°C
PSUMD = fraction of protein plus lipid composition in seed
POTPRO = potential protein fraction in seed

The remaining percentage of seed composition is checked so it does not fall below a minimum value (CARMIN):

```
POTCAR = 1.0 - POTLIP - POTPRO
IF (POTCAR .LT. CARMIN) THEN
  POTCAR =  CARMIN
ENDIF
```

If the value of POTCAR is adjusted, POTPRO and POTLIP are also adjusted so that the sum of the values does not exceed one.

For the remainder pool, the fractions for minerals (PMINSD), organic acids (POASD), and

lignin (PLIGSD) are assumed to be constant, while the carbohydrate component is allowed to vary.

```
POTCAR = POTCAR - PMINSD - POASD - PLIGSD
```

After the seed composition is determined, the respiratory loss in grams glucose per gram seed growth (AGRSD1) is calculated. Energy requirements for synthesis of seed components in units of glucose cost per gram product are RMIN for mineral, RLIG for lignin, ROA for organic acids, RLIP for lipids, and RCH2O for carbohydrates. AGRSD2 includes the cost of synthesizing protein. The protein component (POTPRO) is converted to N fraction (FNINSD).

```
 AGRSD1 = PMINSD*RMIN + PLIGSD*RLIG + POASD*ROA +
&     POTLIP*RLIP + POTCAR*RCH2O
 AGRSD2 = AGRSD1 + RNO3C*POTPRO
 FNINSD = POTPRO / 6.25
```

The model then returns to the DEMAND subroutine where the C and N demand for the other plant parts are determined.

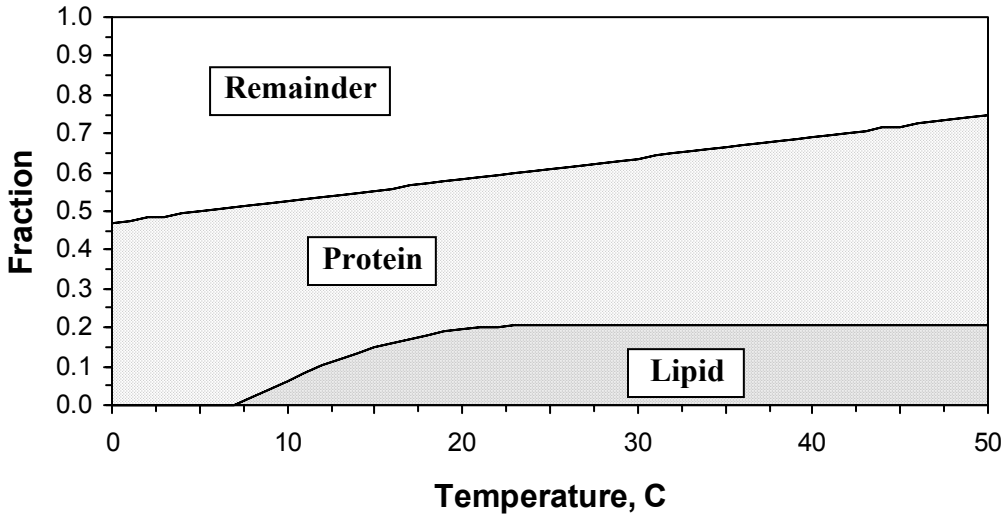


Figure 10 – Seed composition as a function of temperature.

7.0 Nitrogen uptake (NUPTAK)

If the N balance switch is on, NUPTAK is called by CROPGRO three times for: (1) 'Run Initialization', (2) 'Seasonal Initialization', and (3) 'Rate/Integration'. The function of the routine is to calculate root uptake of N after the total N demand for growth (NDMTOT) has been determined in DEMAND. NUPTAK calculates the potential availability of both NO_3 and NH_4 for all soil layers in which roots are present. The uptake of each ion is affected by soil N concentrations, soil water content, and root length density. If total potential root uptake of N is smaller than N demand, the entire potential N is absorbed. If the potential root uptake is greater than demand, the root uptake of N is reduced to the level of demand. This subroutine is based on the N-uptake subroutine developed for the CERES models (Godwin and Jones, 1991).

Table 20 lists input and output arguments for the NUPTAK routine.

7.1 Run Initialization

The 'Run Initialization' section of NUPTAK is used to read values of RTNO3 and RTNH4 from FILEC. These variables represent the nitrate and ammonium uptake per unit length of root in mg [N] per cm [root].

7.2 Seasonal Initialization

Total nitrogen uptake variables are set to zero.

7.3 Rate/Integration

On each day of the daily time loop, NUPTAK begins with initialization of variables. ANDEM is total plant nitrogen demand minus mobilized N that can be used for seed growth.

```
ANDEM = (NDMTOT - NDMSDR) * 10.0
IF (ANDEM .GT. 0.0) THEN
```

If plant N demand is zero or less than zero, then no need for nitrogen uptake from the soil and the routine returns to CROPGRO. If ANDEM is greater than zero, then N uptake is calculated.

Potential N Uptake. The N uptake potential is computed for each soil layer. To determine potential N uptake, a potential N availability factor is determined for NO_3 and NH_4 as well as a soil water factor for each layer. The potential N availability factors (FNO3, FNH4) are calculated based on the NO_3 and NH_4 concentrations in each soil layer.

```
FNH4 = 1.0 - EXP(-0.08 * NH4(L))
FNO3 = 1.0 - EXP(-0.08 * NO3(L))
```

These factors are restricted to values between 0.04 and 1.0. The soil water factor (SMDFR) is calculated as a function of available soil water. Potential N uptake is assumed to decrease at soil water contents either below or above the drained upper limit.

```
SMDFR = (SW(L) - LL(L)) / ESW(L)
IF (SW(L) .GT. DUL(L)) THEN
  SMDFR = 1.0 - (SW(L) - DUL(L)) / (SAT(L) - DUL(L))
ENDIF
```

Next, the integrated effect of root length density (RLV), soil water factor (SMDFR squared), and thickness of the soil layer (DLAYR) is calculated (RFAC).

```
RFAC = RLV(L) * SMDFR * SMDFR * DLAYR(L) * 100
```

The potential uptake of NO_3 (RNO3U) and NH_4 (RNH4U) is then calculated using the user-supplied values of RTNO3 and RTNH4, with the constraints imposed by the availability of water, the root length density, and ionic concentration.

```
RNO3U(L) = RFAC * FNO3 * RTNO3
RNH4U(L) = RFAC * FNH4 * RTNH4
```

Potential N uptake from the whole profile (TRNU) is the sum of RNO3U and RNH4U for all soil layers in which roots occur.

Calculated N Uptake. The total crop N demand (ANDEM) is compared to the potential N uptake (TRNU) to determine whether all or only part of the potential N will be absorbed. If the demand is greater than the potential N uptake, all of the N will be taken up. Otherwise, a N uptake factor (NUF) is calculated and used to reduce the N uptake from each layer (UNO3(L), UNH4(L)) to the level of demand.

```
NUF = ANDEM/TRNU
UNO3(L) = RNO3U(L) * NUF
UNH4(L) = RNH4U(L) * NUF
```

The calculated uptake of NO_3 and NH_4 from the whole profile (TRNO3U, TRNH4U) is then converted to g/m^2 and passed back to DEMAND where the respiration cost of converting this N to protein is calculated and subtracted from PGAVL.

8.0 Nitrogen Mobilization (MOBIL)

The MOBIL routine calculates nitrogen mobilization and is called twice from CROPGRO for 'Seasonal Initialization' and 'Rate/Integration'. Table 21 lists calling arguments for this routine.

8.1 Seasonal Initialization

Mobilization variables are set to zero at the beginning of each season of simulation.

8.2 Rate/Integration

Potential N mining NMINEP was computed in the DEMAND routine. This routine computes the actual N mining rate, NMINEA, which can be less if the total N demand (NDMNEW - TRNU) is lower than the potential mining rate.

```
IF (TRNU .LT. NDMNEW .AND. NMINEP .GT. 0.0) THEN
         NMINEA = NDMNEW - TRNU
         IF (NMINEA .GT. NMINEP) NMINEA = NMINEP
```

The reduced N mining rate is computed as NMINER, a fraction of the stage dependant mining rate, NMOBR. Actual mined N from leaf, stem, root, shell (NRUSLF, NRUSST, NRUSRT, NRUSSH) are calculated as well as the CH_2O used in the re-synthesis of protein (CNMINE):

```
         NMINER = NMINEA/NMINEP * NMOBR
         NRUSLF = NMINER * WNRLF
         NRUSST = NMINER * WNRST
         NRUSRT = NMINER * WNRRT
         NRUSSH = NMINER * WNRSH
         CNMINE = NMINEA / 0.16 * RPRO
      ENDIF
```

9.0 Nitrogen Fixation (NFIX)

Subroutine NFIX calculates nitrogen fixation by nodules and is called if both the ISWNIT (nitrogen simulation) and ISWSYM (nitrogen fixation) switches are set to 'Y'. The routine contains sections for data 'Run Initialization', 'Seasonal Initialization' and 'Rate/Integration' and is thus called three times from the CROPGRO routine. Table 22 lists input and output arguments used by NFIX.

9.1 Run Initialization

Table 6 lists input parameters that are read from FILEIO and FILEC by the NFIX routine. Variable definitions and units can be found in Appendices Q and I. The input data includes curve information for determining the effect of temperature, canopy age, dry soil and wet soil on N fixation and nodule growth rates.

Table 6 – Input Parameters Read by NFIX Routine

FILEIO Parameters (e.g. DSSAT40.INP):			
	EFINOC	EFNFIX	
FILEC Parameters (e.g., SBGRO030.SPE):			
CNODCR	FNFXW	PRONOD	TYPFXD
DWNODI	FNNGT	RFIXN	TYPFXT
FNFXA	NDTHMX	SNACTM	TYPFXW
FNFXD	NODRGM	TYPFXA	TYPNGT
FNFXT			

9.2 Seasonal Initialization

Several variables are set to zero at the beginning of each season of simulation in the 'Seasonal Initialization' section.

9.3 Rate/Integration

The N_2 fixation component in CROPGRO has been changed considerably since the earlier versions in 1992-1993. There is a thermal-time requirement for initiating first-nodule mass. When N uptake is sufficient, nodule growth is slow, receiving a minimum fraction (bypass flow) of the total assimilate that is allocated to roots. When N uptake is deficient (less than N demand) for growth of new tissues, carbohydrates are used for N_2-fixation to the extent of the nodule mass and the species-defined nodule specific activity. If nodule mass is insufficient, then assimilates are used for nodule growth at a rate that is dependent on soil temperature and species-defined nodule relative growth rate. The N_2-fixation rate is further influenced by soil temperature, soil water deficit, soil aeration (flooding), and plant reproductive age. The N_2-fixation process is decreased as the plant water stress factor (TURFAC) is decreased below a species-dependent value.

The first call to the 'Rate/Integration' section of NFIX occurs when plant development has reached a specified physiological age (VSTAGE greater than TTFIX). On this first call, the nodule mass is initialized based on user-supplied initial nodule mass (DWNODI) and the current plant population (PLTPOP).

Next, three factors (ACSTF, ACSTG, and FLDSUM) are summed over the layers of the soil profile. These factors affect the N fixation rates and/or nodule growth rates. All three are

computed using the CURV function, which returns a factor between 0.0 and 1.0 based on four critical curve definition points. ACSTF is a soil temperature factor that affects potential and actual N fixation rates. ACSTG is a soil temperature factor that affects nodule growth rate. FLDSUM is related to excess soil moisture and affects N fixation rates.

```
      ACSTF = ACSTF + DLAYR(I) * FLAYR *
     &        CURV(TYPFXT,FNFXT(1),FNFXT(2),FNFXT(3),FNFXT(4),ST(I))

      ACSTG = ACSTG + DLAYR(I) * FLAYR *
     &        CURV(TYPNGT,FNNGT(1),FNNGT(2),FNNGT(3),FNNGT(4),ST(I))

      EPORS = MAX(SAT(I) -SW(I),0.0)
      FLDSUM = FLDSUM + DLAYR(I) * FLAYR *
     &        CURV(TYPFXW,FNFXW(1),FNFXW(2),FNFXW(3),FNFXW(4),EPORS)
```

These factors are then divided by DNOD, the depth of the nodule zone, to yield the nodule growth and fixation factors, TNFIX, TNGRO, and FLDACT:

```
      TNFIX  = ACSTF  / DNOD
      TNGRO  = ACSTG  / DNOD
      FLDACT = FLDSUM / DNOD
```

Other factors that are computed are SWFACT, a drought factor based on TURFAC (water stress factor), and NFXAGE, which determines the effect of canopy age on N growth rates:

```
      SWFACT = CURV(TYPFXD,FNFXD(1),FNFXD(2),FNFXD(3),FNFXD(4),TURFAC)
      NFXAGE = CURV(TYPFXA,FNFXA(1),FNFXA(2),FNFXA(3),FNFXA(4),DXR57)
```

A running average of the last 8 days of water deficit factor (SWFACT) is kept. Both the 8 day average and the current day's value of SWFACT are used to reduce the N fixation rate.

CLEFT is a running value of CH_2O left over as the various processes occur in the following order:

1) Set aside a minimum (CNODMN) for nodule growth,
2) Fix N_2,
3) Grow nodule mass with any CH_2O not used.

The CH_2O left for nodule growth after N fixation is:

```
      CLEFT = CTONOD - CNODMN
```

The specific nodule activity (SNACT) is computed taking into account the maximum activity of the nodules (SNACTM) and strain effects (EFNFIX) for N-fixation plus the amount CNODMN (except in the case where N2-fixation was limited by water deficit).

The nodule death rate, RNDTH is computed as a function of the soil water deficit factor (SWFACT), soil water excess factor (FLDACT) and carbon deficit factor (CNFACT):

```
      RNDTH = NDTHMX * MAX((1.-FLDACT),(1.-SWFACT),(1.-CNFACT))
      NDTH = MIN(1.0,RNDTH) * DWNOD
```

NDTHMX is the maximum nodule death rate for flooded or dry soil conditions. NDTH is the nodule death rate per unit area.

If physiological maturity has not been reached (DAS < NR7), potential N fixation is computed

```
PNFIXN = MIN((CLEFT * 0.16 / RFIXN), (DWNOD * SNACT)) * TNFIX
```

RFIXN is defined as the respiration required for biological N fixation in g[CH_2O] / g[protein]. The 0.16 factor converts from g[protein] to g[N]. TNFIX is the soil temperature factor affecting N fixation. So, the potential rate of N fixation is the nodule activity rate, limited by available carbon and soil temperature.

Actual N-fixation, NFIXN, is computed with the soil water drought factors (SWFACT and SWMEM8) and soil water excess factor (FLDACT).

```
NFIXN = PNFIXN * MIN(SWFACT, SWMEM8, FLDACT)
```

Normally, CH_2O not used for N-fixation (because of limited nodule mass or activity), can be used to grow more nodule mass, thereby growing the mass needed to overcome this limitation. In the case of plant water deficit that decreases N-fixation specific activity, we do not want the extra CH_2O to be used for nodule growth. For this purpose we compute actual N-fixation (NFIXN) limited by water deficit and potential N-fixation with no water deficit (PNFIXN) and their respective CH_2O costs, CUSFIX and PCSFIX.

```
PCSFIX = (PNFIXN / 0.16) * RFIXN
CUSFIX = (NFIXN  / 0.16) * RFIXN
```

The amount of CH_2O not used for potential N-fixation (CNOFIX) is set aside and is not used for nodule growth in the case of water stress.

```
CNOFIX = PCSFIX - CUSFIX
```

Actually 10% of it can be used for nodule growth plus the amount CNODMN.

```
CLEFT = MAX(0.0,CLEFT - CUSFIX- 0.9*CNOFIX) + CNODMN
```

The maximum nodule relative growth rate is reduced by EFINOC, the inoculation effectiveness (or rhizobium density factor) and EFNFIX, the strain efficiency to obtain the effective nodule relative growth rate (NODRGR).

```
IF (DAS .LT. NR7) THEN
   NODRGR = NODRGM  * EFNFIX * EFINOC
ELSE
   NODRGR = 0.0
ENDIF
```

The new nodule growth rate (NODGR) is limited by either supply or demand for C.

```
NODGR = MIN(CLEFT/AGRNOD,DWNOD*NODRGR)
&        * TNGRO * MIN(SWFACT,FLDACT) * NFXAGE
```

The C required for new nodule growth today and the C used in N-Fixation and nodule growth (including respiration costs) today are calculated as:

```
CNODGR = NODGR * AGRNOD
CNOD = CUSFIX + CNODGR
```

10.0 Pod Growth (PODS and PODCOMP)

The PODS routines are used to compute seed and shell growth and to add new flowers, shells and seeds. PODS is called four times from CROPGRO for 'Run Initialization', 'Seasonal Initialization', 'Emergence' and 'Rate/Integration'. Table 23 lists input and output arguments for PODS.

PODS calls PODCOMP to compute modifications to seed composition at each time step. PODCOMP is called three times by PODS: 'Run Initialization', 'Emergence', and 'Rate/Integration'.

10.1 Run Initialization

Table 7 summarizes the variables which are read by the PODS routine. Definitions for these variables can be found in Appendices Q and L.

Table 7 – Input Parameters Read by PODS Routine

FILEIO Parameters (e.g. DSSAT40.INP):			
CROP	ISWWAT	SDPDVR	WTPSD
ECONO	PODUR	SFDUR	
FILEC Parameters (e.g., SBGRO030.SPE):			
DSWBAR	PROSHI	TYPPDT	XSWFAC(I)
FNPDT(I)	RFLWAB	XMPAGE	YSWBAR(I)
PROLFF	SETMAX	XSWBAR(I)	YSWFAC(I)
PROSHF	SHLAG		
FILEE Parameters (e.g., SBGRO030.ECO):			
ECOTYP	LNGSH	THRESH	

In addition, the 'Run Initialization' section contains calculations that are performed once only per simulation. MNESPM (number of days from end of pod set to physiological maturity), LNGPEG (number of days between start of peg and shell formation) and LAGSD (number of days between start of shell and seed formation in a pod) are calculated based on PHTHRS values from the phenology module. SDVAR and SHVAR, maximum seed and shell growth rates, are also calculated.

10.2 Seasonal Initialization

Several variables are set to zero at the beginning of each season of simulation.

10.3 Emergence

Initialization of variables upon plant emergence is done in both PODS and PODCOMP in the 'Emergence' sections.

10.4 Rate/Integration

This subroutine computes the growth for each age class or cohort of seeds and shells, based on previously computed demands for growth, and on the availability of both C and N. The amount of C (PGAVL) and N (NAVL) available for growth today are used to determine actual fruit growth. Seeds are given first priority for the available C and N for growth over vegetative growth. For determinate crops and cultivars, the variable XFRUIT for the cultivar would be set equal to 1.0. For indeterminate crops, the variable XFRUIT can be less than 1.0, which will have the effect of allowing seed to have priority for only XFRUIT fraction of the day's growth, thereby allowing a fraction (1.0 - XFRUIT) of the growth to be vegetative. The genetic input, XFRUIT, may be modified by environmental conditions, as explained in DEMAND, and the resulting value (XFRT) is actually used in this subroutine. C available for fruit growth (PGAVLR) is computed by:

```
PGAVLR = PGAVL * XFRT
```

Nitrogen stress, CNSTRES, is calculated as an eight day rolling average, which allows for a slow response to change in nitrogen stress. The parameter, NSTRES, is an instantaneous daily variable that changed suddenly with fertilization or large variations in photosynthesis versus N uptake. CNSTRES was used to impact several pod and seed addition processes.

```
CNSTRES = 0.875 * CNSTRES + 0.125 * NSTRES
```

PODS also computes flower, pod, and seed initiation rates depending on environmental conditions and on the C and N status of the plant. It maintains the numbers and ages of these plant structures as they grow in dry weight. Cohort state variables are updated within this subroutine, and the total crop, seed, and shell growth rates are computed for updating the total crop variables in GROW.

Representation of Flowers, Pods, and Seeds. The numbers of pods, seeds and flowers that are initiated within a given day are stored in arrays that allow us to know not only the numbers of pods, etc., on the plants, but also the real time ages of the pods and their physiological ages. Starting at flowering (R1), the variables SHELN(I), SDNO(I), and FLWN(I) maintain the numbers of shells (or pods), seeds, and flowers, respectively, that were initiated on day I and are still on the plants. These and other variables are defined in Appendices Q and L. This approach does not contain information as to whether the fruiting structures are on the main stem or on branches. For example, on day I, the number of pods that are initiated is put into the variable SHELN(I). On later days, some pods from this group may abort, thereby decreasing the value of SHELN(I), but no more will be added to that group. On day (I+1), new pods are entered into the variable SHELN(I+1).

The age, in real days, of the shells in any group or cohort is computed by subtracting the index number (I) from the current time from the first pod (NR2). An example of these variables is given in Table 8 for shell number (SHELN(I)) and the ages of shells (PHTIM(I)). In the table, first pod (NR2) occurred on day 45. On day 49, 16 pods are added. There were 10 pods remaining that were initiated on the first day of pod formation, and these pods are now (on day 49) 5 days old, in real time. The physiological ages, in photothermal time (p-t-d), of the pods can be computed for any group using the PHTIM vector. On day DAS = 49, or I = 5, pods have been forming for 5 days, and the p-t-d age of pods formed on the first day is (3.0 - 0.0) = 3.0. The p-t-d occurring in each day (the variable TDUMX in the code) is based on

temperature and daylength and is computed in the phenology module.

The p-t-d ages of seeds are also computed using the PHTIM variable. The physiological ages of flower cohorts are computed similarly, using a separate variable, PNTIM, which starts at flowering, NR1, instead of first pod. Thus, age distribution and weights of each age group of fruit can be studied with the model, although these detailed variables are not saved in the model outputs. A write statement could be added to PODS.FOR to examine the age structures of fruit for any day or each day, if desired.

Table 8 – Example of shell age and physiological age computation.

Days after start of simulation (DAS)	45 (NR2)	46	47	48	49
Days after first pod (I) I = DAS – NR2 + 1	1	2	3	4	5
Number of shells (SHELN(I))	10	7	12	15	16
Shell age on day DAS = 49 (days)	5	4	3	2	1
Cumulative Physiological Age of cohorts (PHTIM(I)) (p-t-d)	0.0	0.7	1.4	2.1	3.0
Physiological Age of cohort I on day DAS = 49 (p-t-d)	3.0	2.3	1.6	0.9	0.0

In addition to numbers of pods and seeds, the weights of shells and seeds are stored in the variables WTSHE(I) and WTSD(I), respectively, for shells and seeds that are still on the plants "today" after they were formed on day I after NR2. The weights of seeds and shells of different ages are thus stored in the model. The growth of these seeds and shells are based on the available C and N and on the sink strengths of these growing structures.

Seed Growth. Either C or N may limit seed growth on a day. The growth demand for seed, GDMSD g[tissue]/m^2-d, was computed in DEMAND. RSD is the ratio of supply to demand for seed growth and is calculated to be the minimum of the C and N ratios.

The C required for seed growth is GDMSD * AGRSD1 g[CH_2O]/m^2-d, where AGRSD1 is the C cost for producing one g of seed tissue, excluding the cost for protein. The ratio of supply to demand for seed C is:

```
CRSD = MIN(PGAVLR / (GDMSD*AGRSD1), 1.0)
```

Similarly, the demand for N for this growth can be computed and a ratio of supply of N (NAVL) to demand is:

```
NREQ = FNINSD * MIN(PGAVLR/AGRSD1, GDMSD)
NRSD = NAVL / NREQ
```

where FNINSD is the fraction N required for seed. An overall ratio of supply to demand for seed growth (RSD) is computed in PODS as the minimum of these two ratios, with an upper limit of 1.0. This ratio is used to grow each cohort of seed, assuming that a shortage of either C or N will affect all age groups of seed similarly (see PODCOMP).

```
RSD = MIN(CRSD, NRSD, 1.0)
```

The above RSD is a simplified calculation of the supply to demand ratio, which is modified by the PODCOMP routine, which is called at this point in the code. PODCOMP is described in detail in Section 11.5 and basically allows seed protein to decline if N is limiting, so NRSD is not as dominant in setting RSD or limiting seed growth.

After the first pod occurs on the crop, each age group of pods is checked, in a loop, to determine whether the pod is old enough to have seed growing (LAGSD). The age of each cohort is computed as PAGE:

```
PAGE = PHTIM(YRDOY - YRNR2 + 1) - PHTIM(NPP)
IF (PAGE .GE. LAGSD) THEN
```

If the pods are old enough for seed growth, a check is made to determine if the seed are fully grown. Fully grown seed are those in age groups that have reached the maximum threshing percentage (THRESH), which is weight of seed divided by weight of seed plus shell for the cohort. It is assumed that pods are mature at the point in time when the seed fill the cavity, as defined by this THRESH variable, and will no longer grow. A variable, SDMAX, is computed to determine the remaining growth that can occur in a cohort. If the cohort is already at its maximum threshing percentage, then SDMAX will be 0.0, and no seed growth will occur in the cohort. If pests have destroyed some seed in a cohort, then a variable REDSHL is computed to take into account the weight of shells without seed in the calculations. SDMAX for cohort NPP is computed by:

```
SDMAX = (WTSHE(NPP) - REDSHL)*THRESH/(100. - THRESH) - WTSD(NPP)
```

Actual growth of the seed in the cohort is computed next. SDGR, the potential growth rate of each seed, multiplied by the number of seed in the cohort, SDNO(NPP), gives the potential growth for the cohort if it can grow. By taking the minimum of this value and the SDMAX, potential growth is computed. Multiplying RSD by potential growth provides the final amount of growth for the cohort, and seed weight of the cohort is updated by:

```
WTSD(NPP) = WTSD(NPP)+RSD*MIN(SDGR*SDNO(NPP)*REDPUN,SDMAX)
```

Seed growth and N used for seed growth are then summed over all cohorts to obtain total crop growth increments, WSDDTN and NGRSD, respectively.

```
WSDDTN = WSDDTN + RSD * MIN(SDGR*SDNO(NPP)*REDPUN,SDMAX)
NGRSD = NGRSD+ANINSD*RSD*MIN(SDGR*SDNO(NPP)*REDPUN,SDMAX)
```

The C and N remaining for shell or vegetative growth are computed as PGLEFT and NLEFT, respectively, and PGNPOD and NAVPOD are set equal to these variables for later use:

```
PGLEFT = MAX(0.0,(PGAVLR - WSDDTN*AGRSD3))
NLEFT  = MAX(0.0,(NAVL - NGRSD))
PGNPOD = PGLEFT
NAVPOD = NLEFT
```

Shell Growth and Environmental Factors. After the seed growth section in PODS, temperature and water stress factors affecting shell growth or initiation are computed. The first factor, TEMPOD, modifies a maximum rate of flower and shell addition under non-optimal

(colder or warmer) temperatures. An hourly loop is used to compute a value of the function for each hour, and it is averaged over the 24-h period. The function CURV is used to compute each hour's value of TEMPOD and accumulate it for the day:

```
      TEMPOD = 0.
      DO I = 1,24
         TEMPOD = TEMPOD +
     &       CURV(TYPPDT,FNPDT(1),FNPDT(2),FNPDT(3),FNPDT(4),TGRO(I))
      ENDDO
      TEMPOD = TEMPOD/24.
```

where TYPPDT is the curve type (piecewise linear, quadratic, sine curve, or inverse linear), the FNPDT are the minimum, first and second optimal values, and maximum temperature values over which the function goes from 0.0 to 1.0, stays at 1.0, then decreases to 0.0. Figure 11 is an example of the function used for soybean pod setting, using a quadratic curve.

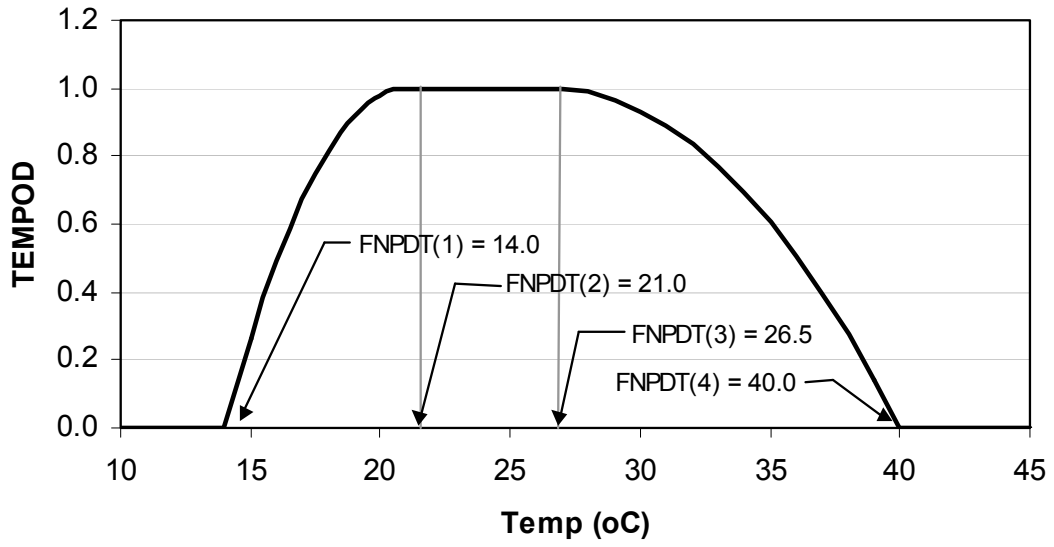


Figure 11 – Effect of temperature on pod setting using quadratic curve.

Average soil water in the top depth of soil which can affect flower and pod addition (DSWBAR) is computed next. ACTSW is the total water in the defined zone, and POTSW is the potential amount that it can hold at the drained upper limit. SWBAR is computed as the ratio of these two values and is bounded by 0.0 and 1.0. Two water stress factors are computed: one based on this average topsoil water content and the other based on the SWFAC variable, which is the ratio of actual transpiration to transpiration demand computed in the water balance component of the model. A table look-up function, TABEX, is used to specify the amounts of SWADD1 and SWADD2 for calculated values of SWBAR and SWFAC, respectively. The concept of the soil water stress factor, SWADD1, is that peanut pods develop in soil, and if it is dry, initiation of pods may be impaired. For other crops such as soybean, SWADD2 is used. The crop parameter file contains values for the table look-up function for

each crop. Therefore, it is possible make either function inactive by putting values of 1.0 in each data slot, depending on the crop. Potential growth per shell (GRRAT1, g[tissue]/shell/d) was computed in DEMAND. Actual growth of shells depends on their physiological ages, this potential growth rate per shell, and the C and N remaining for growth.

Shells are divided into classes that determine their potential growth rates. While shells are less than LNGPEG photothermal days in age, they are assumed (for peanut) to grow at a slow rate (SHLAG * GRRAT1). For soybean and bean, LNGPEG = 0. After pods are LNGPEG p-t-d in age and until they are LNGSH p-t-d in age, they are assumed to grow at the GRRAT1 rate which depends on temperature and daylength. Inside a loop, growth rate for each cohort is calculated. For cohorts between LNPEG and LNGSH p-t-d in age, the following equation is used to compute shell growth for the cohort:

```
ADDSHL = MIN(PGLEFT/AGRSH1,GRRAT1 * SHELN(NPP),
 &              NLEFT/(FNINSH*CNSTRES**0.5))
```

This selects the most limiting factor for growth of the cohort.

For each cohort, a ratio of growth to potential growth, SUPDAY, is computed. This ratio is used to compute a running average for each cohort NPP in SUPDE(NPP). Also, a running average value of TEMPOD is computed for each cohort as AVTEM(NPP) for use in seed setting.

```
SUPDE(NPP) = (SUPDE(NPP) * (NAGE - 1) + SUPDAY)/NAGE
AVTEM(NPP) = (AVTEM(NPP) * (NAGE - 1) + TEMPOD)/NAGE
```

where NAGE is the age of the cohort.

Inside the cohort loop, overall shell growth for the crop is accumulated as WSHDTN, and the overall N used in shell growth is computed by NGRSH:

```
WSHDTN = WSHDTN + ADDSHL
NGRSH = NGRSH + ADDSHL * PROSHI * 0.16 * CNSTRES**0.5
```

Also, inside the cohort loop and after each cohort, PGLEFT and NLEFT are adjusted to remove the C and N required for the last cohort, starting with the oldest. In this way, older pods have priority for growth over younger ones, in contrast to seed (which equally share any deficit). This feature is important in determining shell abortion also in that the supply to demand ratio for younger cohorts will be lower than for older ones, which means that younger shells will have more of a tendency to abort than older ones when C or N is limiting.

```
PGLEFT = MAX(0.0,(PGLEFT - ADDSHL * AGRSH1))
NLEFT  = MAX(0.0,(NLEFT - ADDSHL * (FNINSH*CNSTRES**0.5)))
```

The shell cohort weights are updated in the same do loop by adding the computed ADDSHL for the cohort and subtracting mined protein from the shells (SHMINE):

```
WTSHE(NPP) = WTSHE(NPP) + ADDSHL - MAX(SHMINE,0.0)
```

where SHMINE is the protein mined from each cohort. Total protein mass mined can be

computed as N mined from shells, NRUSSH, divided by 0.16. This amount is assumed to be mined from shells that have not reached their threshing percentages. WTSHM is the weight of shells that have reached their maximum threshing percentage, computed at the top of the PODS subroutine, so (SHELWT - WTSHM) is the total shell weight from which N mining is occurring. The amount to mine from a particular cohort is proportional to the weight of shell in that cohort to the total shell weight that can be mined, or:

```
SHMINE = NRUSSH/0.16 * WTSHE(NPP)/(SHELWT - WTSHM)
```

After this point in PODS, flower, pod and seed initiation rates and abortions are computed. These calculations do not involve C and N balances, so any C and N remaining after shell growth will be used in vegetative growth in VEGGR.

Seed Setting. Shells grow in cohorts for LAGSD photothermal days before seeds can be initiated in each shell. In a do loop, each cohort of shells is checked to determine if the shells have reached this age during the current day, or if PAGE is between LAGSD and LAGSD + TDUMX. If so, the number of seed that are set (SDNO(NPP)) depends on the number of seed per pod (SDPDVR), the running average supply to demand ratio (SUPDE(NPP)) and temperature factor (AVTEM(NPP)), and a photoperiod factor (DRPP):

```
      IF (SUPDE(NPP) .GE. SETMAX) THEN
         SHRAT = 1.0
      ELSE
         SHRAT = SUPDE(NPP)/SETMAX
      ENDIF
      SDNO(NPP) = MIN(SHRAT, AVTEM(NPP) * DRPP) * SHELN(NPP)
&           * SDPDVR + SDNO(NPP)
```

where SETMAX is a fraction such that if SUPDE(NPP) falls below this value, shell abortion will occur. Therefore, seed setting depends on nutritional as well as environmental factors. The weight of shells aborted is next computed by multiplying the number of shells aborted by their average weights, and the shell weight of the cohort is updated:

```
      WTABR = 0.0
      START = SHELN(NPP)
      SHELN(NPP) = SHELN(NPP) * MIN(SHRAT, AVTEM(NPP) * DRPP)
      IF (START .GT. 0.) WTABR = (START-SHELN(NPP))*WTSHE(NPP)/START
      WTSHE(NPP) = WTSHE(NPP) - WTABR
      WTABRT = WTABRT + WTABR
```

Flower Development and Pod addition. After flowering, the numbers of flowers initiated on each day are kept in the variable FLWN(I), where I is the days after first flower. Flowers may abort each day, depending on photothermal time, and on water stress. The fraction to abort each day is computed by:

```
      AFLW = RFLWAB * (1.0 - TDUMX) * (1.0 - SWADD2)
```

where RFLWAB is a maximum relative rate of abortion under adverse conditions. In a do loop that examines each flower cohort, the population of flowers in the cohort NPP is reduced by the AFLW abortion fraction:

```
      FLWN(NPP) = FLWN(NPP) * (1.0 - AFLW)
```

Then, the physiological age of the cohort is computed as PNAGE. If the physiological age of the cohort is old enough for forming pods, i.e., at least PHTHRS(6) p-t-d in age, then flower numbers are converted into pods or are aborted.

```
      PNAGE = PNTIM(N-NR1+1) - PNTIM(NPP)
      IF (PNAGE .GE. PHTHRS(6)) THEN
```

The variable FLWFRC determines the fraction of flowers in a cohort to convert in a day, to allow for the flowers to be converted to pods over one full p-t-d. FLWRDY is then computed as the number of flowers ready to form pods after they have developed sufficient length of time. FLWRDY is computed and then the remaining flowers in the cohort are updated:

```
         FLWRDY = FLWFRC*FLWN(NPP) + FLWRDY
         FLWN(NPP) = (1.0-FLWFRC)*FLWN(NPP)
```

Pod addition then depends on the numbers of flowers developing to the time that they can form pods, and on environmental and crop conditions. First, a calculation is made to compute an imaginary pod load that the current day's photosynthesis would support, PMAX. This calculation basically takes into account the canopy size. For example, for small canopies, PGAVLR would be small, and thus pod addition would be small. It also accounts for other factors that would reduce the day's overall photosynthesis, such as low light levels, on pod setting. This imaginary pod load is then divided by the minimum time that the canopy takes to reach a final load of pods, PODUR, to compute a maximum pod addition rate, PMAX pods/m^2-d, which is the maximum pod addition rate under optimal temperature and photoperiod:

```
      PMAX = PGAVLR/(SDVAR*AGRSD1*SDPDVR)*(1./PODUR)
```

Next, FLADD is computed, which is the number that can be formed from flowers modified by environmental conditions:

```
      FLADD = FLWRDY * TEMPOD *(DRPP**1.3)* MIN(SWADD1,SWADD2) * XFRT
```

Canopy N status is presumed to affect pod addition. RNITP is leaf N concentration (for photosynthesis purposes) excluding CH_2O storage. RNITPD is an attempt to scale pod addition back as a function of available N concentration above minimum N (PROLFF).

```
      RNITPD = (RNITP*0.01-PROLFF*0.16)/(FNINL-PROLFF*0.16)
      RNITPD = MIN(1.1,RNITPD)
      RNITPD = MAX(0.1,RNITPD)
```

Then, PODADD is computed by adjusting PMAX by temperature, photoperiod, water stress, and stage of growth factors. ACCAGE is the proportion of development from last flower to physiological maturity, which limits pod addition after this time.

```
      PODADD = PMAX * TEMPOD * (DRPP**1.3) *
&            MIN(SWADD1,SWADD2,RNITPD) * MAX((1.0 - ACCAGE),0.0)
```

Finally, actual pod addition, which is calculated as the number of shells in the cohort SHELN(YRDOY-YRNR2+1), is determined by the most limiting factor in pod addition: PODADD (from PMAX), FLADD, C remaining after seed growth, or N remaining after seed growth. Thus, seed load on the plant will slow pod addition through its use of either C or N, which ever becomes limiting first. The C limitation is computed as the number of pods that can be grown with the C remaining after seed growth, or PGNPOD/(SHMAXG*AGRSH1), where the SHMAXG is the maximum per shell growth rate (g[shell tissue]/shell-d) and the AGRSH1 is the C cost for shell tissue (g[CH_2O]/g[shell tissue]). The limitation on pod addition by N availability is calculated similarly, and new shells are initiated as:

```
SHELN(YRDOY - YRNR2 + 1) = Minimum { PODADD
                                   { PGNPOD/(SHMAXG*AGRSH1)
                                   { FLADD
                                   { NAVPOD/(SHMAXG*(FNINSH*CNSTRES**0.5)))
```

The code involving CUMSIG, TRIGGR, and NDSET is designed to minimize resurgence of pod set after the normal window of pod addition is past (NDSET) and especially for situations where stress (drought, etc.) reduced pod set and the stress is now relieved.

CUMSIG is a 5-day rolling average of CRSD which is an indictor of the daily ratio of PGAVLR to seed demand for CH_2O. If a full pod load has been achieved for 5 days (CUMSIG < 0.98), then TRIGGR is re-set from 0 to 1. Next, if the crop is past NDSET and TRIGGR=1, pod addition is stopped. NDSET is the "normal" time by which a pod load (full number) should be achieved. However, if CUMSIG > 1.0 and TRIGGR is not equal to 1 the pods still are added. For indeterminant peanut, NDSET should be almost near maturity date.

CUMSIG is also used to influence seed composition. After a full pod load is achieved (CUMSIG < 0.98), seed protein is allowed to float up or down depending on deficiency of CH_2O versus N.

Flower addition is assumed to depend on canopy size as expressed by PMAX, and by the C and N remaining after seed growth. FLWADD is calculated similar to the PODADD, but flowers are assumed to form at twice the rate as pods. Also, the hypothetical cost of flower addition is assumed to be only one tenth of the cost of daily pod growth. The day's new flowers are entered into the variable FLWN(N-NR1+1), as computed below,

```
FLWN(YRDOY - YRNR1 + 1) = Minimum { FLWADD
                                  { PGNPOD/(SHMAXG*0.1*AGRSH1),
                                  { NAVPOD/(SHMAXG*0.1*(FNINSH*CNSTRES**0.5))
```

Calculation of Crop Totals for Seeds and Shells. In the last section of PODS, overall crop fruit variables are computed. A do loop is used to add over all cohorts seed number (SEEDNO) and pod number (PODNO). In addition, each cohort is checked to see if seeds have completely filled the shell cavity, or have reached their maximum shelling percentage, to compute overall crop percent pod maturity, PCTMAT. Daily growth of seed and shells are passed to CROPGRO as WSDDTN and WSHDTN, respectively, and N used for the seed and shell growth are passed as NGRSD and NGRSH, respectively.

10.5 Seed Composition (PODCOMP)

The PODCOMP routine accounts for variations in C and N supply and allows the seed protein concentration to be modified from the value determined in the SDCOMP subroutine.

10.5.1 Run Initialization

Table 9 lists input parameters which are read by the PODCOMP routine from FILEC (e.g., SBGRO030.SPE). These parameters include the fractional composition of seed (PLIGSD, PMINSD, POASD) and respiration required for synthesizing these components (RLIG, RLIP, RMIN). Definitions and units for these input parameters and other variables used in the PODS and PODCOMP routines can be found in Appendices Q and L.

Table 9 – Input Parameters Read by PODCOMP Routine

PLIGSD	PROMAX	RLIG	ROA
PMINSD	PROMIN	RLIP	THETA
POASD	RCH2O	RMIN	

10.5.2 Emergence

On the day of emergence, the following variables are initialized to 1.0:

- CUMSIG - 5 day running average of the ratio of supply to demand for carbon
- RATION – Ratio of C supply to N supply
- RATIOC – Ratio of N supply to C supply

10.5.3 Rate/Integration

Seed protein is constrained by the maximum and minimum N fractions allowed (PROMAX and PROMIN), which extend on either side of genetic potential N fraction (FNINSD, which is set from SDPRO). N is initially not limiting to seed growth, but becomes progressively more limiting as protein concentration declines and approaches the minimum N (PROMIN). This subroutine determines the supply: demand ratio for seed growth (RSD), actual seed N fraction (ANINSD), and actual glucose cost to produce a gram of seed (AGRSD3).

The subroutine first checks whether any seed growth has occurred for the given day, indicated by either PGAVLR (C available for reproductive growth) or GDMSD (seed growth demand) being greater than zero. When no seed growth has occurred, the seed composition variables are not changed.

When seed growth occurs, the subroutine proceeds to compute seed composition modification. The supply demand ratios for C (CRSD) and N (NRSD) are determined.

```
CRSD2 = PGAVLR/(GDMSD*AGRSD1)
CRSD  = MIN(CRSD2,1.0)
NREQ  = FNINSD*(MIN(PGAVLR/AGRSD1,GDMSD))
NRSD  = NAVL/NREQ
```

CRSD is limited to a maximum of 1.0, while CRSD2 has no maximum value. NREQ represents N demand for today's seed growth with no change in N concentration.

The possible seed N concentration, PNINSD, is determined from the "total" N (NAVL) relative to the available C (PGAVLR / AGRSD1) or seed growth demand (GDMSD).

```
PNINSD = NAVL/(MIN(PGAVLR/AGRSD1,GDMSD))
```

The reader should note that PNINSD, NREQ, and NRSD are calculated considering the C supply and potential seed growth rate. Thus, when C supply is high relative to N supply (NAVL), then PNINSD will be low and NRSD less than 1. When C supply is low, then PNINSD will be high (above FNINSD) and NRSD greater than 1. When CRSD is less than 1, NRSD can be either greater or less than 1, depending on N available (NAVL).

A five-day rolling average of CRSD (CUMSIG) is used to determine when a full seed load has been achieved, i.e. C supply (PGAVLR) is equal or less than seed growth demand. A full seed load is defined to occur (triggers threshold) when CUMSIG has a value less than 0.98.

```
CUMSIG = 0.8 * CUMSIG + 0.2 * CRSD
```

This function (CUMSIG< 0.98) is used to set cases of good C supply days when CRSD2 is greater than one yet the crop had previously achieved a full seed load.

The model then considers four cases for modifying seed composition.

1. Do not have a full seed load and carbon supply is not limiting seed growth (i.e. CRSD2 > 1.0 and CUMSIG > 0.98). Do not change composition.
2. Have a full seed load and neither N or C supply is limiting seed growth (i.e. CRSD2 > 1.0 and CUMSIG < .98). Protein concentration is allowed to increase or decrease depending on the relative amounts of C and N.
3. C supply is limiting seed growth (i.e. CRSD2 < 1.0). Decrease growth rate per seed and allow protein concentration to rise: however do not allow protein concentration to increase above PROMAX.
4. N supply is limiting seed growth (i.e. NRSD < 1.0). Progressively decrease seed growth to partially maintain protein concentration while decreasing the protein concentration; however do not allow protein to decrease below PROMIN.

Case 1 – Seed addition phase. During the seed addition phase an increasing fraction of the plant available C and N is partitioned to the seed as individual seeds enter linear growth. In the CROPGRO model, C and N do not limit individual seed growth rate until a full seed load has occurred. During this phase CRSD2 and NRSD (C and N ratios of supply to demand) are above 1. Thus seed composition should not be changed from the potential composition calculated in SDCOMP until there is a full seed load. Actual seed N (ANINSD) is set to FNINSD (maximum seed N) prior to seed set to allow N to go to vegetative parts when carbon is not limiting. Not changing seed protein prior to a full seed load has a negligible effect on the final seed composition at harvest because seed dry matter gain during this phase is small.

Case 2 – C and N Supply Greater than Demand. After a full seed load occurs

(CUMSIG < 0.98) and C and N supply are greater than demand, the protein concentration is allowed to increase or decrease depending on the relative amounts of C and N available. The seed N is modified as a ratio of N to C supply normalized to FNINSD. First ANINSD is set equal to FNINSD. NAVL is checked to be greater than zero. The ratio of N supply to C supply (RATION) is the determined as:

```
RATION = (NAVL/(PGAVLR/AGRSD1))/FNINSD
```

RATIOC is the reciprocal of RATION. When RATION is greater or equal to 1.0 then

```
ANINSD = MIN(FNINSD * (1.0 + (RATION - 1.0)/3.), PROMAX)
```

Otherwise,

```
ANINSD = MAX(FNINSD * (1.0 - (RATIOC - 1.0)/3.), PROMIN)
```

The functions assume a slope of one-third of possible change from FNINSD to PROMAX or PROMIN.

Case 3 - Carbon Limitation (CRSD < 1.0), but N not limiting (NRSD≥1.0). The N concentration in legume seed has been observed to increase with low C supply from late season water stress, low CO_2, or low light during reproductive growth (Pate, 1985; Ross, 1988). When C supply limits seed growth rate (CRSD < 1) and more N is available (NAVL) than required (possible from N mobilization), then potential seed N (PNINSD) may be greater than seed N concentration computed by SDCOMP (FNINSD) and the seed N concentration is allowed to increase. Actual seed N (ANINSD) is set equal to PNINSD but not allowed to be greater than PROMAX.

```
ANINSD = MIN(PNINSD, PROMAX)
```

Case 4 - Nitrogen Limitation. When the N supply is less than demand (i.e. NRSD < 1.0), the N supply is insufficient to grow seed at FNINSD (PNINSD always less than FNINSD). Soybean respond to limited N supply by decreasing the N concentration in the seed prior to significant decreases in the seed growth rate (Hayati et al., 1995; Tolley-Henry et al, 1992; Pate, 1985; Streeter, 1978). The seed growth is decreased using a non-rectangular hyperbola function for the relationship between PNINSD and reduction in seed growth for limiting N supply (XRSD).

```
     XRSD = ((PNINSD/PROMIN + 1.0 - ((PNINSD/PROMIN + 1.0)**2
&           - 4*THETA*PNINSD/PROMIN*1.0)**0.5)/(2*THETA)) /
&            ((FNINSD/PROMIN + 1.0 - ((FNINSD/PROMIN + 1.0)**2
&           - 4*THETA*FNINSD/PROMIN*1.0)**0.5)/(2*THETA))
```

Figure 12 illustrates the relationship between PNINSD, FNINSD and XRSD. As PNINSD declines, N stress increasingly limits seed growth; nevertheless, the amount of seed growth is considerably greater than if constant seed N concentration was required. When N limits seed growth rate, the seed N concentration is slightly increased because less C is required for seed growth. ANINSD is recalculated with seed growth reduced by XRSD and the value will be slightly higher than PNINSD. The reader should note that the carbon supply is already considered in the NRSD and PNINSD equations and

by definition the new ANINSD will be greater than PROMIN.

```
ANINSD = NAVL/(XRSD*MIN(PGAVLR/AGRSD1,GDMSD))
```

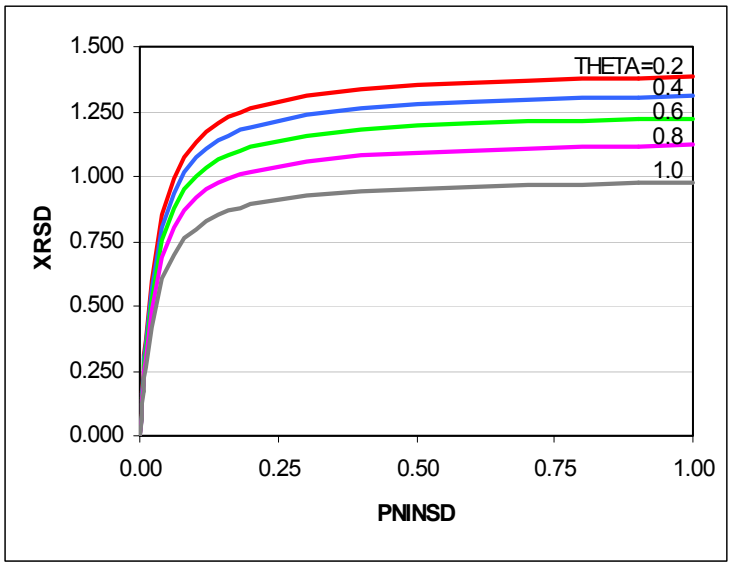


Figure 12 – Reduction in seed growth rate (XRSD) due to limited N supply

Adjusting Other Seed Composition Components. When the seed N fraction is changed by C and N supply, the other seed fractions need to be modified so that the total of the seed fractions sum to one. The lipid and carbohydrate fractions do not appear to change equally with the protein fraction. The model assumes that for a one percent change in protein, the lipid changes 0.33 (DTLIP) and the carbohydrate fraction changes 0.67 (DTCAR). The lipids and carbohydrate fractions (POTLIP and POTCAR) are then adjusted by DTLIP and DTCAR.

```
DTLIP = 0.33*(FNINSD - ANINSD)*6.25
DTCAR = (FNINSD - ANINSD)*6.25 - DTLIP
POTLIP = POTLIP + DTLIP
POTCAR = POTCAR + DTCAR
```

Next the actual glucose cost to produce a gram of seed (AGRSD3) is determined with the new composition.

```
      AGRSD3 = PNINSD*RMIN + PLIGSD*RLIG + POASD*ROA +
&           POTLIP*RLIP + POTCAR*RCH2O
```

Last, C limitations to seed growth are determined considering the shift in protein concentration.

```
DMSDN = NAVL / ANINSD
DMSDC = PGAVLR / AGRSD3
RSD = MIN(MIN(DMSDN, DMSDC)/GDMSD, 1.0)
```

After the seed composition, AGRSD3 and RSD are determined, the PODCOMP subroutine returns to PODS. If C and N supplies are approximately in the same proportion as required for seed growth, then, all C and N would be used in a day's time, and none would remain as "unutilized" reserves. (In the CROPGRO model, N supply will not normally be in excess because N supply is based on C availability as well as potential growth.) Therefore, we can assume that C will remain, if any, after growth takes place during a day.

11.0 Pod Detachment (PODDET)

PODDET is called from CROPGRO when the variable DETACH is set to 'Y' and is used to compute pod detachment rates. The routine is called for input of variables ('Run Initialization' section), initialization on the day of plant emergence ('Emergence' section) and for pod detachment calculations ('Rate/Integration' section). Table 24 lists input and output arguments for the routine.

11.1 Run Initialization

Input variables, listed in Table 10 below, are read from FILEC.

Table 10 – Input Parameters Read by PODDET Routine

DWC	TB	TM
PR1DET	TO1	XP1DET
PR2DET	TO2	XP2DET

11.2 Emergence

On the day of plant emergence, PODDET is called to initialize variables to zero.

11.3 Rate/Integration

The reader is referred to the paper by Batchelor (1994) for description of the PODDET routine.

12.0 Vegetative Growth and Partitioning (VEGGR and CANOPY)

The VEGGR subroutine plays an important role in the model calculations for vegetative growth. In the sequence of calculations each day, VEGGR is called after the daily supply of C (PHOTO or ETPHOT) and N (NUPTAK, MOBIL, NFIX) have been computed, after the demand for growth C and N have been computed (in DEMAND), and after seeds and shells have been allowed to grow with first priority (in PODS). For indeterminate plants, some C is partitioned to vegetative growth each day, even if the seed load is large.

When VEGGR is called, the remaining C (PGAVL) and N (NAVL) that can be used for vegetative growth are provided. This subroutine then computes the vegetative tissue growth and its partitioning into leaves, stems, and roots. It also determines how much C and N remain after vegetative growth, if any, that will be stored in the plant for later use. Canopy height and width are computed in subroutine CANOPY.

Because the proportions of leaf, stem, and root tissue growth change with time in a growing season, the efficiency of tissue synthesis from C also varies with time. In VEGGR, the C requirement for the growth of vegetative tissue is computed each day, depending on how much leaf, stem, and root tissue grow in a day, and the N composition of each tissue. The cost for synthesizing vegetative tissue (AGRVG) is expressed as g[CH_2O]/g[tissue], and is the inverse of growth efficiency that is used in some crop growth models.

Table 25 lists input and output arguments that are included in the calling statement for the VEGGR routine.

12.1 Run Initialization

Table 11 lists data, all of which is read from FILEC for use by the VEGGR routine. In addition, the CANOPY subroutine is called to input data as discussed in Section 13.5.

Table 11 – Input Parameters Read by VEGGR Routine		
ATOP	PROLFG	PRORTI
CADSTF	PROLFI	PROSTG
CMOBMX	PRORTG	PROSTI

12.2 Seasonal Initialization

A number of variables are initialized at the beginning of each season of simulation.

12.3 Emergence

The normal "growth" fractions of N for growing leaf, root and stem tissue (FNINLG, FNINRG, FNINSG) are calculated based on the input parameters PROLFG, PRORTG and PROSTG. CANOPY is called to establish the initial canopy height and width.

12.4 Rate/Integration

Dry Weight Growth. The supply of N available for vegetative growth is calculated (SUPPN). If N is limiting, a N stress term is computed (NSTRES) by dividing total supply of N from N fixation, soil N uptake, and N mobilization (NFIXN + TRNU + NMINEA) by total N demand for new growth (NDMNEW) and multiplying this ratio by 0.7. If NSTRES is less than 1.0, then N was short by at least 30%, and the root-shoot partitioning is modified. The parameter ATOP

is the maximum shift in partitioning from top growth to root growth. Water stress, through the factor TURFAC, can also modify root-shoot partitioning (FRRT), and shift depends on which stress, N or water, is most severe on a day using the following equations:

```
FRRT  = ATOP * (1.0 - (MIN(TURFAC,NSTRES)))*(1.0-FRRT) + FRRT
```

A cumulative turgor factor (CUMTUR) is computed which keeps a modified 20-day rolling average of TURFAC, a drought stress factor. This factor is used to shift partitioning between leaf and stem toward leaf growth, especially after drought is released. Leaf and stem partitioning (FRLF, FRSTM) are then calculated.

```
CUMTUR = 0.95*CUMTUR + 0.05*TURFAC
FRLF   = (1.0 + 0.6*(1.0-CUMTUR))*(1.-FRRT)*FRLF/(FRLF + FRSTM)
FRLF   = MIN(FRLF,0.90*(1-FRRT))
FRSTM  = 1.0 - FRRT - FRLF
```

Leaf and stem partitioning is limited to a maximum of 98% of the total vegetative partitioning.

```
FRLF  = MIN(FRLF,FRLF*0.98/(MAX(0.001,FRLF+FRSTM)))
FRSTM = MIN(FRSTM,FRSTM*0.98/(MAX(0.001,FRLF+FRSTM)))
FRRT  = 1.0 - FRLF - FRSTM
```

If the partitioning coefficients are modified, then the cost for synthesis, AGRVGR, is re-computed as:

```
AGRVG = AGRLF*FRLF + AGRRT*FRRT + AGRSTM*FRSTM
```

Then, potential vegetative tissue growth rate (VGRDEM) is computed, using the revised AGRVG, and potential growth of leaves, stems, and roots are computed:

```
VGRDEM = PGAVL / AGRVG
WLDOTN = FRLF  * VGRDEM
WSDOTN = FRSTM * VGRDEM
WRDOTN = FRRT  * VGRDEM
```

New vegetative tissue growth is reduced if the supply of N is low enough to cause tissue N to fall below a <u>minimum</u> growth N content. Initially, N demand for vegetative tissue growth is computed by multiplying the tissue growth rates for leaves, stems and roots by the <u>maximum</u> N contents:

```
NGRLF  = WLDOTN * FNINL
NGRST  = WSDOTN * FNINS
NGRRT  = WRDOTN * FNINR
NGRVEG = NGRLF + NGRST + NGRRT
```

Similarly, the minimum N requirement for tissue growth (NGRVGG) is computed.

```
NGRLFG = WLDOTN * FNINLG
NGRSTG = WSDOTN * FNINSG
NGRRTG = WRDOTN * FNINRG
NGRVGG = NGRLFG + NGRSTG + NGRRTG
```

If N available (NAVL) is less than the minimum normally required for tissue growth, then tissue growth is reduced and conversion costs (AGRVG) are re-calculated based on the reduced growth rates:

```
      IF (NAVL .LT. NGRVGG) THEN
         IF (NGRVGG .GT. 0.0) THEN
            NRATIO = NAVL / NGRVGG
            WLDOTN = WLDOTN * NRATIO
            WSDOTN = WSDOTN * NRATIO
            WRDOTN = WRDOTN * NRATIO
            NGRLF  = NGRLFG * NRATIO
            NGRST  = NGRSTG * NRATIO
            NGRRT  = NGRRTG * NRATIO
            AGRVG = AGRLF * FRLF * (1.0 - (PROLFG - PROLFI)/(1.0 -
&                 PROLFI))+ AGRRT * FRRT * (1.0 - (PRORTG - PRORTI)/(1.0 -
&                 PRORTI)) + AGRSTM * FRSTM * (1.0 - (PROSTG - PROSTI)/
&                 (1.0 - PROSTI))
         ENDIF
```

If available N is greater than the minimum required for tissue growth, but less than the maximum, then N concentrations are reduced, but leaf expansion occurs as normal.

```
      ELSE
         IF (NGRVEG .GT. 0.0 .AND. NAVL .LT. NGRVEG) THEN
            NGRLF = MIN(NAVL * NGRLF / NGRVEG,NGRLF)
            NGRST = MIN(NAVL * NGRST / NGRVEG,NGRST)
            NGRRT = MIN(NAVL * NGRRT / NGRVEG,NGRRT)
         ENDIF
```

Protein fractions of new tissue growth are computed and respiration costs are re-computed.

Remaining C and N After Growth. Either C or N could be left over after allocating these substrate for tissue growth rates for a day. The remaining C, if any, can be computed by:

```
      PGLEFT = MAX(0.0,PGAVL - ((WLDOTN+WSDOTN+WRDOTN)*AGRVG))
```

The variable, EXCESS, is a factor used to reduce photosynthesis tomorrow, whenever there is assimilate left over today because of inadequate sink strength (pod and seed) and especially because there is inadequate N to grow seed and vegetative tissue. The function is a square root factor that is less "reducing" when the ratio, PGLEFT/PG, is small, but becomes dominant as that ratio increases. The values, 1.20 and 0.20, prevent any reduction in photosynthesis if PGLEFT is less than 20% of today's PG.

```
      IF (PG .GT. 0.0001 .AND. PGLEFT .GT. 0.00001) THEN
         EXCESS =  (1.20 - MIN(1.0, MAX(PGLEFT/PG,0.20)) )**0.5
      ELSE
         EXCESS = 1.00
      ENDIF
```

If PGLEFT is greater than the potential C mining rate, then no mined C would have been needed during the day, and additional C would be added to reserves, in the amount of

(PGLEFT - CMINEP). If positive, it will be added to C reserves in stem and leaf tissue. The proportion of the PGLEFT to add to stems and leaves depends on the mass of each and a factor, CADSTF that allows stems to store more per unit mass than leaves. The variable LSTR, computed below, is the ratio of excess C to be added to leaves in a day relative to the amount to be stored in stems. It is first computed from CADSTF, an input parameter, then adjusted by the ratio of leaf mass to leaf plus stem mass. Before flowering (DAS < NR1), LSTR is adjusted to allow a greater proportion of the C to be added to leaves. The code in VEGGR that does this is:

```
IF (DAS .LT. NR1) THEN
   LSTR = (1.-0.6*CADSTF)/(0.6*CADSTF)
ELSE
   LSTR = (1.-CADSTF)/CADSTF
ENDIF
IF (STMWT+WTLF .GT. 0.0) THEN
   LSTR = LSTR * WTLF/(STMWT+WTLF*LSTR)
ENDIF
```

The variables CADLF and CADST are then computed as mass of CH_2O per m^2 to leaves and stems, respectively.

```
IF (PGLEFT .GE. CMINEP) THEN
   CADLF = (PGLEFT-CMINEP)/PCH2O * LSTR
   CADST = (PGLEFT-CMINEP) * (1. - LSTR) / PCH2O
```

PCH2O is the respiration cost for storing C for later use. It was assumed that each vegetative plant tissue has some C when it grows, in leaves, stems, roots, and shells. When C from the current day's photosynthesis is stored in existing tissue (PGLEFT - CMINEP), it is stored only in stems and leaves. Any of the stored C may be mined for use as needed.

If PGLEFT is less than CMINEP, this means that C was mined to meet the growth demands, and actual mined C is computed (CMINEA) as well as how much and it was derived from stems, leaves, roots, and shells, or:

```
ELSE
   IF (CMINEP .GT. 0) THEN
      CMINEA = CMINEP - PGLEFT
      CRUSLF = CMINEA / CMINEP * CMOBMX * WCRLF * (DTX + DXR57)
      CRUSST = CMINEA / CMINEP * CMOBMX * WCRST * (DTX + DXR57)
      CRUSRT = CMINEA / CMINEP * CMOBMX * WCRRT * (DTX + DXR57)
      CRUSSH = CMINEA / CMINEP * CMOBMX * WCRSH * (DTX + DXR57)
   ENDIF
ENDIF
```

Similarly, N remaining is computed by subtracting the N used for leaf, stems and roots from the N available for vegetative growth:

```
   NLEFT  = MAX(0.0,NAVL  -  (NGRLF  + NGRST  + NGRRT))
```

This amount will be added to existing tissue. However, the amount that can be added to existing tissue is restricted to the amount that can be added to old tissue, computed as N

demand for old tissue, or NDMOLD. Therefore, the amount of N to add to existing tissue is computed by:

```
IF (NLEFT .GT. NDMOLD) THEN
   NLEAK  = NLEFT  - NDMOLD
   TNLEAK = TNLEAK + NLEAK
   NLEFT  = NLEFT  - NLEAK
ELSE
   NLEAK = 0.0
ENDIF
```

If N remains, it is allocated to leaf, stem, and root tissue in proportion to the growth rates of these tissue types. N pools are not maintained explicitly. A value of N concentration at leaf senescence is a parameter, and the "stored" N can be computed at any time by subtracting this senescence N value from the current N composition.

12.5 Canopy Expansion (CANOPY)

12.5.1 Run Initialization

Table 12 lists input data that is read from FILEC and FILEE by the CANOPY routine.

Table 12 – Input Parameters Read by CANOPY Routine		
FILEC Parameters (e.g., SBGRO030.SPE) :		
KCAN	XVSHT	YVSHT
XHWPAR	YHWPAR	YVSWH
XHWTEM	YHWTEM	
FILEE Parameters (e.g., SBGRO030.ECO):		
ECOTYP	RHGHT	RWIDTH

12.5.2 Emergence

Upon plant emergence, initial canopy height and width are linearly interpolated from user supplied arrays (XVSHT and YVSHT in species file) based on VSTAGE.

12.5.3 Rate/Integration

At the end of VEGGR, the subroutine CANOPY is called to compute the rate of canopy envelope expansion, and to update canopy height (CANHT) and canopy width (CANWH). These canopy dimensions are required by the hedge row photosynthesis model (ETPHOT) that is optionally available in CROPGRO.

Temperature and PAR functions are used to modify canopy expansion. The temperature function, HWTEM, is illustrated in Figure 13. This modifier is used to directly influence canopy height and width via effect on internode length. Temperature effects also occur via temperature effect on rate of node appearance (RVSTGE). Next, a modifier for PAR is calculated which includes the effects of shading on canopy expansion. In computing the value of PARNOD, the effective PAR after shading, it is assumed that the upper 30% of the canopy shades the growing point. This reduced PAR is used only on canopy height expansion. Unreduced values of PAR are used to calculate width expansion. Figure 14 shows the relationship between PARNOD and leaf area index (XHLAI).

```
      PARNOD = PAR * EXP(-KCAN*(0.3*XLAI))
```

PAR and PARNOD values are used as indices for a look-up table to establish the relative width and height expansions, respectively. This relationship is shown in Figure 15.

```
      HPAR = TABEX(YHWPAR,XHWPAR,PARNOD,8)
      WPAR = TABEX(YHWPAR,XHWPAR,PAR,8)
```

Tables of internode length vs. node position on the main stem are used to compute the absolute rates of canopy expansion. Figure 16 shows the values used for soybean, based on calibration of height and width relative to V-stage over time for several Florida and Iowa data sets (Boote, unpublished). After computing the temperature (HWTEM) and PAR (HWPAR) rate modifiers, the rate of increase in canopy height is computed by:

```
      RCANHT= RVSTGE * TABEX(YVSHT,XVSHT,VSTAGE,10) * HWTEM *
     &  TURFAC * HWPAR * RHGHT
```

and canopy height is updated by the equation:

```
      CANHT = CANHT + RCANHT
```

Similarly, canopy width is updated after calculating a daily rate of expansion, or:

```
      RCANWH= RVSTGE * TABEX(YVSWH,XVSHT,VSTAGE,10) * HWTEM *
     &  TURFAC * HWPAR * RWIDTH
      CANWH = CANWH + RCANWH
```

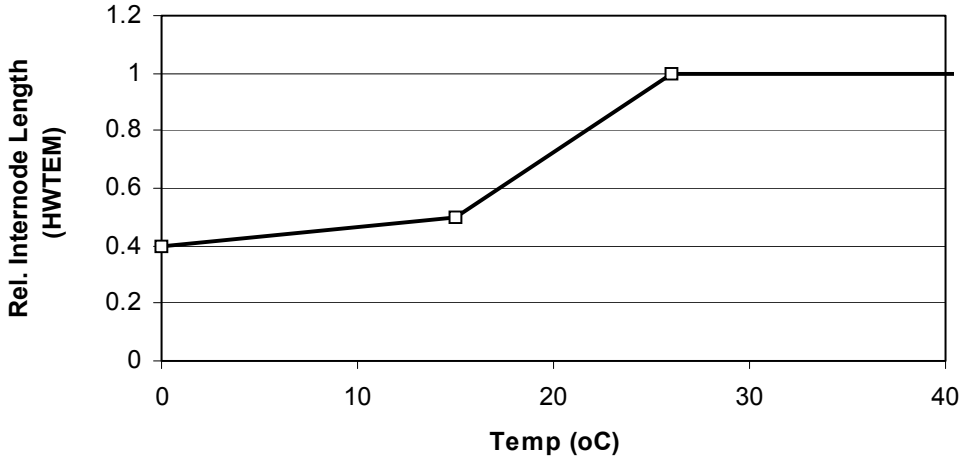


Figure 13 – Effect of Temperature on Canopy Expansion (relative internode length)

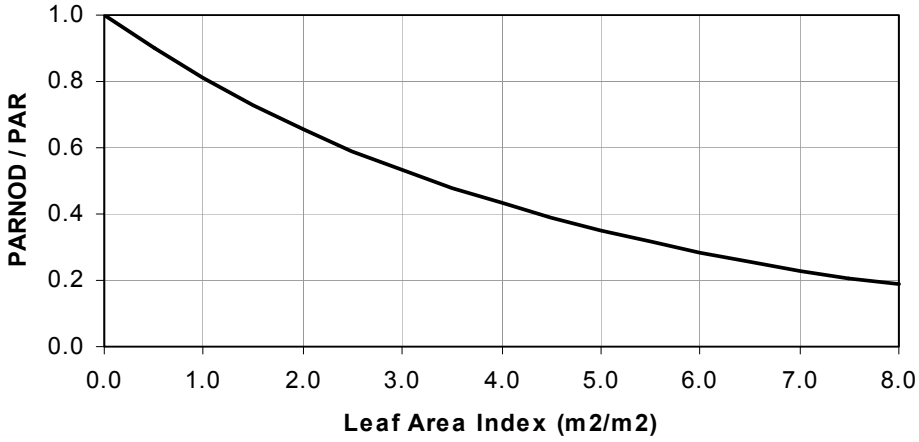


Figure 14 – Leaf area index vs. effective shading of growing point.

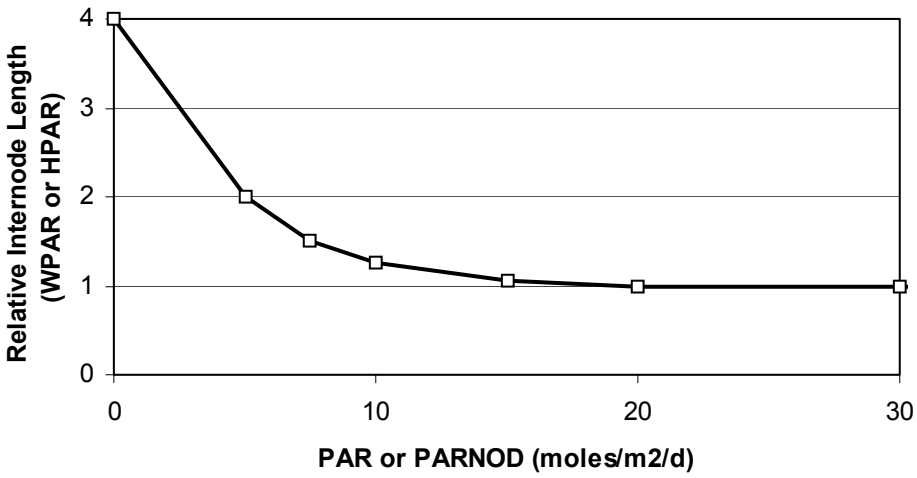


Figure 15 – Effect of PAR on Canopy Expansion

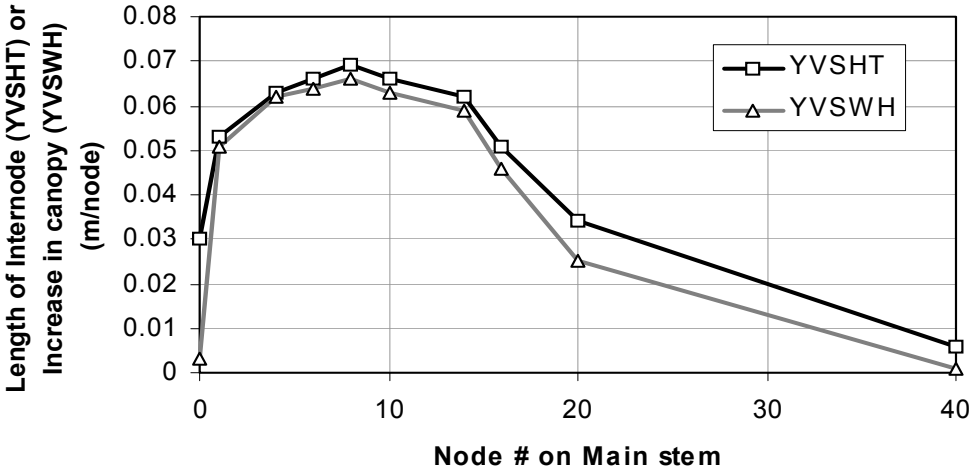


Figure 16 – Effect of VSTAGE on Canopy Variables

13.0 Leaf Senescence (SENES)

The SENES routine is called to compute leaf senescence due to natural aging, drought stress, light stress, and physiological maturity. Table 26 lists variables that occur as input and output arguments in the calling statement for SENES. The routine is called three times from the CROPGRO main module for 'Run Initialization' of data, 'Seasonal Initialization' of variables, and 'Rate/Integration' calculations.

13.1 Run Initialization

Table 13 lists input parameters which are read in the 'Run Initialization' section of the SENES routine. All parameters are read from FILEC.

Table 13 – Input Parameters Read by SENES Routine

ICMP	SENDAY	SENRTE
KCAN	SENMAX(I)	TCMP
PORPT	SENRT2	XSENMX(I)

13.2 Seasonal Initialization

The four output variables, SLDOT, SSDOT, SLNDOT and SSNDOT are set to zero at the beginning of each season of simulation. These variables represent total daily leaf and petiole senescence and senescences due to water stress.

13.3 Rate/Integration

Before stage NR7 has been reached (when 50% of plants first have yellowing or maturing pods), natural senescence is calculated based on a table of values indexed by VSTAGE. This relationship is shown in Figure 17, represented by variable SENPOR.

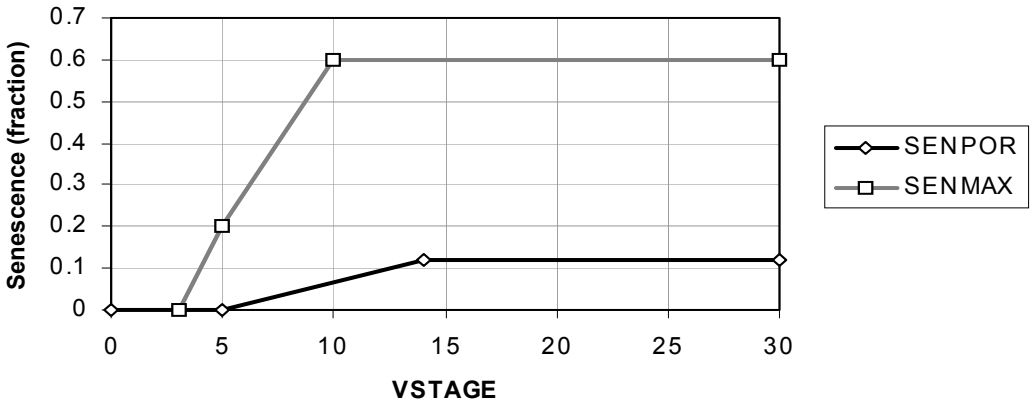


Figure 17 – VSTAGE vs. Senescence

The portion of leaf left after natural senescence (PORLFT) is calculated to be:

```
PORLFT = 1.0 - TABEX(SENPOR,XSTAGE,VSTAGE,4)
```

If the existing leaf weight is greater than the portion left, daily leaf senescence (SLDOT) is calculated to be the difference.

```
SLDOT = WTLF * ( 1.0 - RHOL) - WLPOS * PORLFT
```

Senescence due to N mobilization from leaves is added to the daily leaf senescence.

```
SENWT = SENRTE * NRUSLF / 0.16
SENWT = MIN(WTLF,SENWT)
SLDOT = SLDOT + SENWT
SLDOT = MIN(WTLF,SLDOT)
```

Next, senescence due to low light conditions in the lower canopy is computed. The leaf area index at which light compensation is reached (LCMP) is calculated. This is the threshold for leaf senescence due to crop canopy cover. Leaf area index in excess of LCMP is senesced over TCMP thermal days, where LTSEN is the leaf area index senesced in $m^2/m^2/d$.

```
LCMP = -(1/KCAN)*ALOG(ICMP/PAR)
LTSEN = DTX * (XLAI - LCMP)/TCMP
LTSEN = MAX(0.0,LTSEN)
```

ICMP (light compensation point) and TCMP (time constant for senescence) are input from FILEC. Leaf area loss is converted to mass by the following:

```
SLDOT = SLDOT + LTSEN * 10000./SLAAD
```

Additional senescence due to water stress, WSLOSS, is calculated based on the water stress factor, RATTP, which is calculated in the soil water balance module and represents soil water stress conditions 4 days ago.

```
WSLOSS = SENDAY * (1. - RATTP) * WTLF
```

Figure 17, in addition to presenting the natural senescence rate, includes a maximum senescence rate, SENMAX, which is used as an upper limit on senescence due to water stress.

```
PORLFT = 1.0 -TABEX(SENMAX,XSENMX,VSTAGE,4)
WSLOSS = MIN(WSLOSS, WTLF - WLPOS*PORLFT)
WSLOSS = MAX(WSLOSS, 0.0)
```

Petiole senescence is calculated based on a ratio of petiole mass to leaf mass (PORPT) which was input from FILEC.

After NR7, leaf senescence is calculated based on a factor, SENRT2, also input from FILEC.

```
SLDOT = WTLF * SENRT2
```

14.0 Freeze Damage (FREEZE)

The FREEZE routine is called once from the CROPGRO module during the 'Integration' section, if the minimum daily temperature (TMIN) is less than a user-supplied variable, FREEZ1. The plant will lose all its leaves (but continue phenological development) if the temperature goes below FREEZ1 and stops growth entirely if the temperature goes below FREEZ2, another user-supplied parameter. WLFDOT is returned as leaf weight loss due to freezing. Table 27 lists the calling arguments used to invoke the FREEZE routine.

15.0 Root Growth (ROOTS, IPROOT and INROOT)

The ROOTS routine is called by CROPGRO to calculate root growth, extension, respiration and senescence. The routine is called four times by the CROPGRO main module for 'Run Initialization', 'Seasonal Initialization', 'Emergence' and 'Rate/Integration'. Table 28 lists variables that occur as input or output arguments in the calling statement for ROOTS.

15.1 Run Initialization (IPROOT)

Table 14 lists input parameters that are read by the ROOTS routine from FILEIO and FILEC. Definitions for these parameters are listed in Appendices Q and N.

Plant extractable soil water by layer (ESW) is calculated as soil parameters are read in for each soil layer:

Table 14 – Input Parameters Read by ROOTS Routine

	FILEIO Parameters (e.g., DSSAT40.INP)		
ISWWAT	LL(L)	SAT(L)	WR(L)
	FILECC Parameters (e.g., SBGRO030.SPE)		
PORMIN	RTDEPI	RTSEN	YRTFAC(I)
RFAC1	RTEXF	XRTFAC(I)	DS
RLDSM	RTSDF		

15.2 Seasonal Initialization

'Seasonal Initialization' consists of setting the root senescence variable (SRDOT) to zero and determining an initial value of root growth rate (RFAC2). The initial growth rate is established based on a look-up table indexed to VSTAGE, with the initial VSTAGE set to zero. This look-up function will be discussed further in Section 16.4.

```
RFAC2 = TABEX(YRTFAC,XRTFAC,0.0,4)
```

15.3 Emergence (INROOT)

The 'Emergence' section of ROOTS consists of a single call to INROOT to perform initialization of root variables on the day of plant emergence. The initial root density over the soil profile (RLINIT) is calculated based on the initial weight of the seed or seedling (WTNEW), initial partitioning to roots (FRRT), root length per unit weight (RFAC1), and the rooting depth (RTDEP). RLINIT is then proportioned throughout the rooting depth to obtain

root length density (RLV) by layer.

```
DO L = 1,NLAYR
   DEP = MIN(RTDEP - CUMDEP, DLAYR(L))
   RLINIT = WTNEW * FRRT * PLTPOP * RFAC1 * DEP / ( RTDEP *10000)
   CUMDEP = CUMDEP + DEP
   RLV(L) = RLINIT / DLAYR(L)
   IF (CUMDEP .GE. RTDEP) GO TO 300
ENDDO
```

15.4 Rate/Integration

The root depth growth rate (RFAC2, cm / Physiol. Day) is calculated daily using TABEX, a linear interpolation "look-up" routine. Root growth rate is dependent upon the number of nodes in the main stem (VSTAGE) as shown in Figure 18. This Figure uses root growth rate data from the dry bean species file, which indicates a decrease in root growth with increasing VSTAGE. Soybean root growth rates are generally assumed to be constant.

```
RFAC2 = TABEX(YRTFAC, XRTFAC, VSTAGE, 4)
```

New root growth density (RLNEW) is calculated based on WRDOTN, growth rate of new root tissue, calculated in VEGGR and on RFAC1, the user-supplied root density coefficient (cm [root] / g[root]) found in FILEC. The carbon demand for this new root growth is defined by CGRRT. AGRRT is the mass of C required for new root growth (g[CH_2O]/g[root]).

```
RLNEW = WRDOTN * RFAC1 / 10000.
CGRRT = AGRRT * WRDOTN
```

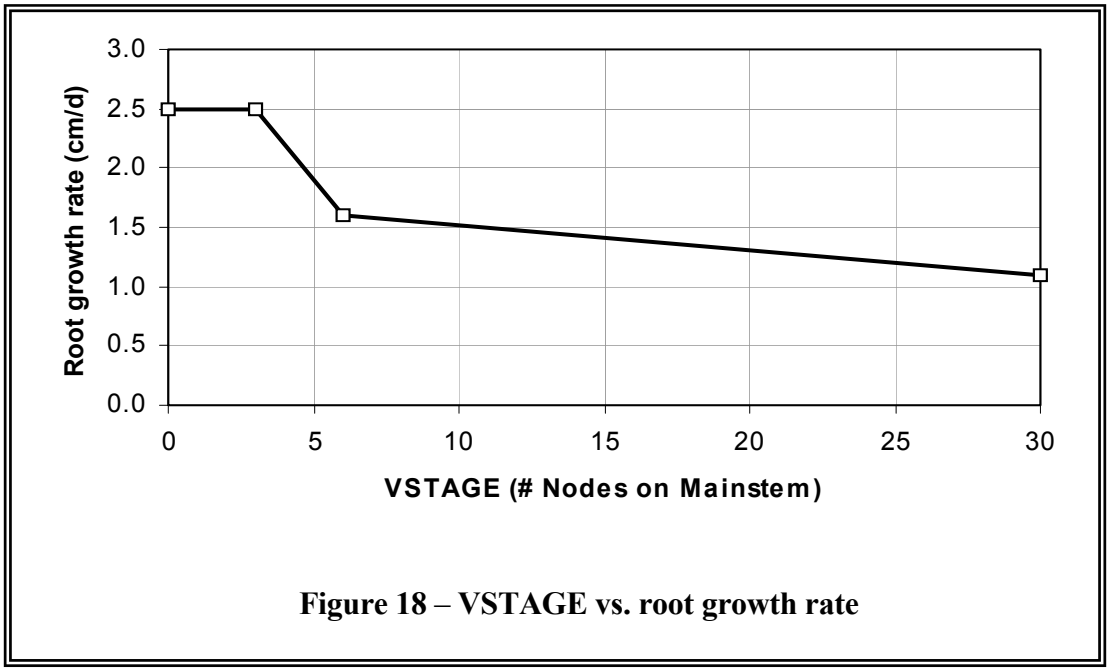


Figure 18 – VSTAGE vs. root growth rate

Soil water stress factors that affect root growth are SWEXF, excess water stress factor and SWDF, soil water deficit factor. Both are 0.0 for maximum stress and 1.0 for unstressed conditions. These are calculated in a loop through the soil layers.

```
      SWEXF = (SAT(L) - SW(L)) / PORMIN
      SWDF  = (SW(L) - LL(L)) / (0.25*ESW(L))
```

where SW(L) is the current soil water content of layer L, SAT(L) is the volumetric water content at saturation, LL(L) is the volumetric water content at lower limit. PORMIN is the minimum pore space required for supplying oxygen to roots for optimal growth and function.

The fraction of roots that survive (RTSURV) is calculated based on these stress factors.

```
      RTSURV = MIN(1.0,(1.-RTSDF*(1.-SWDF)),(1.-RTEXF*(1.-SWEXF)))
```

RTSDF is the maximum fraction of root length senesced in a given layer per physiological day when water content falls below 25% of extractable soil water and RTEXF is the fraction of root death per day under oxygen depleted, saturated soil.

The root length density (RLV) for each layer is then adjusted by RTSURV, if the value is greater than the minimum density below which water stress-induced senescence is not allowed (RLDSM).

```
      IF (RLV(L) .GT. RLDSM) THEN
         RLV(L) = RLV(L) * RTSURV
      ENDIF
```

Next, a combined weighting factor for root growth distribution, RLDF(L), is calculated for the current soil layer based on a user-supplied weighting factor (WR(L), from the soil input file) and the water stress factors, SWDF and SWEXF for respective layers.

```
      RLDF(L) = WR(L) * DLAYR(L) * MIN(SWDF,SWEXF)
```

A cumulative weighting factor, TRLDF, is kept over the rooting depth. For the deepest soil layer within the rooting depth, the rooting depth is extended based on the calculated value of RFAC2 and layer-dependent variables, SWDF and SWEXF, except that root depth is accelerated up to 25% when under whole plant water stress (SWFAC<1.0). This acceleration under plant water deficit is only overcome when available soil water content in the extending zone is less than 25%.

```
      RTDEP = RTDEP + DTX * RFAC2 * MIN(SWDF,SWEXF) *
&        (1. + 0.25 * (1. - MAX(SWFAC,0.40)))
```

Another loop through the soil layers which contain roots is used to distribute the new root growth. RRLF(L) is the proportion of new root growth which is allocated to layer L. New root growth, RLGRW(L), is calculated based on this factor.

```
      IF (TRLDF .LT. 0.00001) THEN
         RRLF(L) = 1.0
      ELSE
         RRLF(L) = RLDF(L)/TRLDF
      ENDIF
      RLGRW(L) = RRLF(L) * RLNEW / DLAYR(L)
```

Root senescence (RLSEN) is assumed to occur as a fraction of roots which are senesced per

physiological day.

```
RLSEN(L) = RLV(L) * RTSEN * DTX
```

Respiration costs are calculated for new root growth. These root respiration costs are computed for output purposes, so that soil CO_2 efflux from roots and nodules can be output.

Root length density for the current soil layer, RLV(L), is updated based on senescence and new growth. Total soil profile values for root growth (TRLGRW), root senescence (TRLSEN) and the total root length (TRLV) are accumulated.

```
RLV(L) = RLV(L) - RLSEN(L) + RLGRW(L)
TRLGRW = TRLGRW + RLGRW(L) * DLAYR(L)
TRLSEN = TRLSEN + RLSEN(L) * DLAYR(L)
TRLV = TRLV + RLV(L) * DLAYR(L)
```

RFAC3, the current ratio of root length to root weight (cm[root]/g[root]) is calculated from yesterday's values of total root length and root weight.

```
RFAC3 = TRTDY * 10000.0 / RTWT
```

This is used to compute the mass of senesced roots today:

```
SRDOT = TRLSEN / RFAC3 * 10000.
```

16.0 Integration of Growth Variables (GROW, IPGROW and STRESS)

The GROW routine is called from CROPGRO four times: once each for 'Run Initialization', 'Seasonal Initialization', 'Emergence' and 'Rate/Integration'. Table 29 lists input and output arguments for the GROW routines.

16.1 Run Initialization (IPGROW)

The 'Run Initialization' section of GROW, a call is made to IPGROW to read all input data required for this routine from FILEIO, FILEC and FILEE. These input parameters are listed in Table 15. Definitions for these parameters are listed in Appendices Q and F.

Table 15 – Input Parameters Read by GROW Routines

FILEIO Parameters (e.g., DSSAT40.INP):			
ECONO	ISWNIT	PLTPOP	SDWTPL
IDETO	ISWSYM	ROWSPC	WTPSD
IHARI	PLME		
FILECC Parameters (e.g., SBGRO030.SPE):			
ALPHL	PLIGNO	PMINNO	PROLFF
ALPHR	PLIGRT	PMINRT	PROLFI
ALPHS	PLIGSD	PMINSD	PRONOD
ALPHSH	PLIGSH	PMINSH	PRORTF
PCARLF	PLIGST	PMINST	PRORTI
PCARNO	PLIPLF	POALF	PROSHF
PCARRT	PLIPNO	POANO	PROSTF
PCARSD	PLIPRT	POART	PROSTI
PCARSH	PLIPSH	POASD	RMIN
PCARST	PLIPST	POASH	WTFSD
PLIGLF	PMINLF	POAST	XPODF
FILEE Parameters (e.g., SBGRO030.ECO):			
ECOTYP	SDLIP	SDPRO	

In addition, some once-only initialization calculations are done in the 'Run Initialization' section of GROW. Based on the input value of ROWSPC (spacing between plant rows) and PLTPOP (plant population), the value of BETN (spacing between plants in a row) is calculated for use in the GROW subroutine.

Respiration production for plant components are computed based on input parameters. For leaf respiration, the following equation is used:

```
      CPFLF  = 30./44.*(PLIPLF*1.720 + PLIGLF*0.659 + POALF*(-0.011)
     &         + PMINLF*RMIN + PCARLF*0.170)
```

Where PLIPLF = Proportion of leaf that is lipid,
PLIGLF = Proportion of leaf that is lignin,
POALF = Proportion of leaf tissue that is organic acid,
PMINLF = Proportion of leaf tissue that is mineral,
PCARLF = Proportion of leaf tissue that is carbohydrate,
RMIN = Respiration required for synthesizing mineral structure.

All variables on the right hand side of the equation are input parameters. Constants in the

equation represent respiration costs for leaf growth. Similar calculations are done for stem, root, nodule, shell and seed respiration requirements.

The minimum percentage of leaf N composition after N mining is computed to be:

```
PCNMIN = PROLFF * 16.0
```

16.2 Seasonal Initialization

In the 'Seasonal Initialization' section, numerous variables are set to zero at the beginning of each seasonal simulation. Specific leaf area (SLA) is initialized to the specific leaf area of new tissue growth (F) with was initialized in the DEMAND subroutine.

16.3 Emergence

The 'Emergence' section of the GROW routine is called once from the CROPGRO module to initialize variables on the day of plant emergence. Initial seedling or transplant weights (WTNEW (g/plant), TOTWT (g/m^2)) are calculated from input parameters. The initial mass of leaf, stem and root (WTLF, STMWT and RTWT) are then distributed based on initial partitioning (FRLF, FRSTM), which were computed in the DEMAND subroutine.

```
WTLF  = FRLF * TOTWT
STMWT = FRSTM * TOTWT
TOPWT = WTLF + STMWT
RTWT  = TOTWT - TOPWT
```

Carbohydrate reserves in leaf, stem and root (WCRLF, WCRST, and WCRRT) are computed as a fraction of the plant component mass. The initial fractional values (ALPHL, ALPHS, and ALPHR) are read from FILEC.

Plant N is set to the maximum value, initially. For leaf this is computed as:

```
WTNLF  = WLFI * PROLFI * 0.16
```

Where WTNLF = Mass of N in leaves (g[leaf N] / m2[ground])
WLFI = Initial leaf mass (g[leaf]/m2[ground])
PROLFI = Maximum protein composition in leaves during growth with luxurious supply of N (g[protein] / g[leaf tissue])

Units of protein N are converted to elemental N by the 0.16 factor. Similar calculations are done for stem and root.

SDNPL represents the N in the seed portion of the plant at planting, while SEEDNI represents the total N in the plant on planting day. These are computed by:

```
 SDNPL  = WTPSD * SDPRO * 0.16 * 0.75 * PLTPOP -
&  (WTNLF + WTNST + WTNRT)
 SDNPL  = MAX(SDNPL,0.0)
 SEEDNI = WTNLF + WTNST + WTNRT + SDNPL
```

WTPSD is the maximum weight per seed under non-limiting substrate (g / seed). This value is reduced by 25%. SDPRO represents the mass of protein as a fraction of seed weight. Units are

converted to g[N]/m2 and the previously computed masses of leaf, stem and root are subtracted. SEEDNI adds back the leaf, stem and root masses.

Initial leaf area for all leaves (AREALF) and healthy leaves (AREAH) are computed as:

```
AREALF = WTLF * F
AREAH  = AREALF
```

The specific leaf index with and without leaf carbon reserves are computed as:

```
SLA    = AREALF / WTLF
SLAAD  = AREALF / (WTLF - WCRLF)
```

Leaf are index values are converted from leaf area values for all leaves and healthy leaves:

```
XLAI   = AREALF / 10000.
XHLAI  = XLAI
```

16.4 Rate/Integration

Subroutine GROW is called from CROPGRO after all rate variables have been computed by the different process subroutines. GROW integrates the rate variables to compute updated values of all of the overall crop state variables.

Dry Weight Growth. New tissue growth (GROWTH) and the required respiration for this growth is defined by the following equations at the beginning of the GROW routine:

```
GROWTH = WLDOTN + WSDOTN + WRDOTN + WSHDTN + WSDDTN + NODGR

GRWRES = WLDOTN*CPFLF + WSDOTN*CPFSTM + WRDOTN*CPFRT +
&          NODGR*CPFNOD + WSHDTN*CPFSH1 + WSDDTN*CPFSD1 +
&          30./44*(TRNO3U/0.16 * 1.798 + TRNH4U/0.16 * 0.462 +
&          NFIXN/0.16 * 1.798)
```

New growth rates, senescence of tissue, and N and C mobilization rate variables were computed in a series of subroutines called from CROPGRO. In GROW, the net rates of growth (positive or negative) for all plant weight state variables are computed by summing the component process rate variables. The net leaf growth rate, WLDOT g[leaf]/m^2-d, is equal to the new leaf growth (WLDOTN), minus weight losses due to senescence (SLDOT), freezing (WLFDOT), N mining (NRUSLF), C mobilization (CRUSLF), and insect damage (WLIDOT). Any C (CADLF) or N (NADLF) that is added to leaf storage during the day is also added to the net leaf growth rate after adjusting for losses due to senescence, freezing and insect damage.

```
WLDOT = WLDOTN - SLDOT - WLIDOT - WLFDOT - NRUSLF/0.16 - CRUSLF
IF (WTLF .GT. 0.0) THEN
   WLDOT = WLDOT + (CADLF+NADLF/0.16) *
&   (1. - MIN(1.0,(SLDOT+WLIDOT+WLFDOT)/WTLF))
ENDIF
```

where the variables used in this and other equations are given in Appendices Q and F. The 0.16 factor converts N to protein, which is assumed to be the form of N storage in the leaf. The

terms in the right side of the equation were computed in various subroutines called from CROPGRO, including the daily supply and demand of C and N for growth, growth rates, senescence, and remaining N and C. Similar net growth rate equations are in GROW.FOR for stems (WSDOT), roots (WRDOT), shells (WSHDOT) and seeds (WSDDOT). Net pod growth rate (WPDOT) is simply the addition of shell and seed growth rates, or

```
WPDOT = WSHDOT + WSDDOT
```

Nodule growth rate (WNDOT) is computed by adding new growth and subtracting mass of nodules that died:

```
WNDOT  = NODGR - NDTH
```

Net growth rate for the entire crop, on a unit area basis, is:

```
WDOT = WLDOT + WSDOT + WRDOT + WPDOT + WNDOT
```

The six-day running ratio of new pod or seed growth to total new growth (XPOD) is computed by:

```
XPOD = 0.17 * (WSDDTN+WSHDTN)/GROWTH + 0.83 * XPOD
```

for pod growth (XPODF = 'PD') or by:

```
XPOD = 0.17 * (WSDDTN)/GROWTH + 0.83 * XPOD
```

for seed growth (XPODF = 'SD'). The six-day running average XPOD is used to regulate N mobilization and other physiological processes. For non-seed producing species like tomato, XPODF is set to 'PD'.

Integration of the net rates is accomplished by simply adding today's net growth, which may be positive or negative, to the values of the state variables from the day before. This is equivalent to an Euler integration method in which the integration time step is one day. The equations for updating the weight state variables are:

```
TOTWT  = TOTWT  + WDOT
TOPWT  = TOPWT  + WSDOT + WLDOT + WPDOT
WTLF   = WTLF   + WLDOT
STMWT  = STMWT  + WSDOT
SDWT   = SDWT   + WSDDOT
SHELWT = SHELWT + WSHDOT
RTWT   = RTWT   + WRDOT
PODWT  = PODWT  + WPDOT
DWNOD  = DWNOD  + WNDOT
TGROW  = TGROW  + GROWTH
```

The weights of tissue components include the weights of N (as protein) and C (as carbohydrate). Thus, if C or N is removed from the leaves or other tissue, the state variables representing weight of C or N in the leaf is reduced and the leaf weight (WTLF) also has to be reduced. Using this structure, for example, the weight of each leaf constituent (C, N, and tissue excluding C and N) can be computed at any time.

Cumulative leaf and stem growth (CLW and CLS) are computed for use in estimating the percentage losses. Also, values of the above-ground weight at flowering and seed weight at maturity are saved for output:

```
      IF (YRDOY .EQ. YRNR1) THEN
         CANWAA = TOPWT
      ELSEIF (YRDOY .EQ. YRNR8) THEN
         SDWTAM = SDWT
      ENDIF
```

The weight of the crop to be used in the maintenance respiration calculations is computed as WTMAIN. The mass of seed respiring was assumed to be equal to the shell mass, which approximately mimics the concept that storage products (lipid, protein, starch) do not require maintenance respiration..

```
      WSDMAN = MIN(SDWT,SHELWT)
      WTMAIN = TOTWT - SDWT + WSDMAN
```

After updating the weight state variables, the weights are checked to determine if any dropped to 0.0, and if they have dropped to this level to end the simulation. Subroutine STRESS is called to print out an end of simulation message to the OVERVIEW.OUT file.

Carbohydrate Content. Net changes in the carbohydrate content of each plant component are computed similar to the net growth rate equations described above. Carbohydrate in the plant components is maintained as mass of carbohydrate (CH_2O) per unit area. For example the net CH_2O addition for leaves (WRCLDT) is computed by:

```
      WRCLDT = ALPHL * WLDOTN - CRUSLF - RHOL*(SLNDOT+WLIDOT+WLFDOT)
      IF (WTLF .GT. 0.0) THEN
         WRCLDT = WRCLDT + CADLF *
&                 (1. - MIN(1.0,(SLDOT+WLIDOT+WLFDOT)/WTLF))
      ENDIF
```

where ALPHL*WLDOTN is the CH_2O added in new leaf tissue growth for the day. ALPHL is the proportion of new leaf growth that is non-structural carbohydrates, which can be mobilized for use as time progresses. CRUSLF is the amount of CH_2O mobilized from the leaves during the day. The variable RHOL is the fraction of CH_2O in the leaves on a day. All leaves that are lost from the plant result in loss of the CH_2O reserves contained in the tissue. Excess CH_2O is added to the plant as expressed by CADLF. CH_2O lost to senescence, pest damage and freezing are subtracted from this term.

Similar equations are in the code for net rates of addition of stem, root, and shell CH_2O reserves. Carbohydrate that is added to seed is assumed to be unavailable for mining, thus, such a pool is not maintained for seed.

The following equations update the weights of carbohydrate pools in each plant component:

```
      WCRLF = WCRLF + WRCLDT
      WCRST = WCRST + WRCSDT
      WCRRT = WCRRT + WRCRDT
```

```
      WCRSH = WCRSH + WRCSHD
```

Finally, in the carbohydrate section of GROW, the fractions of CH_2O in leaves, stems, roots, and shells are computed as variables RHOL, RHOS, RHOR, and RHOSH, respectively.

Nitrogen Variables. Net changes in the N content of each plant component are computed similar to the net growth rate equations described above. N in the plant components is maintained as mass of elemental N per m^2 of ground area. Protein weight can be computed by dividing the mass of N by 0.16 under the assumption that protein is 16% N. For example the net N addition for leaves (NLDOT) is computed by:

```
      NLDOT = NGRLF - (SLNDOT + WLIDOT + WLFDOT) * (PCNL/100) - NRUSLF
&               - (SLDOT - SLNDOT) * PROLFF * 0.16
      IF (WTLF .GT. 0.0) THEN
         NLDOT = NLDOT + NADLF *
&               (1. - MIN(1.0,(SLDOT + WLIDOT + WLFDOT) / WTLF))
      ENDIF
```

where NGRLF is the N added by new leaf tissue growth for the day. This equation demonstrates that leaf tissue that is lost from the plant may cause different amounts of N to be carried with it. The variable PCNL is the percentage of N in the existing leaves on a day (WTNLF / WTLF * 100.). Leaves that are lost from the plant by insect damage, frost damage, or water stress result in loss of all of the N contained in the tissue. Leaf weight that is lost as N is mined (SLDOT-SLNDOT) is assumed to be at the lowest leaf protein content, PROLFF, 16% of which is N mass. NRUSLF is the N mining rate. NADLF is the excess N that is added to leaves after tissue growth. This value is adjusted to account for N losses with senescence, pest damage and freezing.

Similar equations compute the net changes in weights of N for stems, roots, shells, and seeds. Then, weight of N in each component is computed by:

```
      WTNLF = WTNLF + NLDOT
      WTNST = WTNST + NSDOT
      WTNRT = WTNRT + NRDOT
      WTNSH = WTNSH + NSHDOT
      WTNSD = WTNSD + NSDDOT
```

and total plant N content (WTNTOT) and above-ground N content (WTNCAN) are updated as:

```
      WTNTOT = WTNLF + WTNST + WTNRT + WTNSH + WTNSD + WTNNOD
      WTNCAN = WTNLF + WTNST + WTNSH + WTNSD
```

Canopy N at flowering is saved as CANNAA for output purposes.

The next two sections in GROW compute the cumulative N added to each plant component during the season, and the cumulative N losses from each plant component during the season. These variables are used in NBAL.FOR to check if the total plant N is equal to the cumulative N used by the plant for growth minus the cumulative N loss during the season.

For photosynthesis purposes only, the percentages of N in leaf is computed by dividing the N weights by component weights minus CH_2O stored in the components. Thus, the percentage N in leaves and other components is not influenced by CH_2O reserves stored in the components. For example, percentage N in leaves is computed by:

```
RNITP = WTNLF / (WTLF - WCRLF) * 100.0
```

Finally, the weights of N that is above the minimum concentrations are computed to determine the potential N available for mining. For example, the N available for mining from leaves, WNRLF g[N]/m^2-d, is computed by:

```
WNRLF = MAX (WTNLF - PROLFF * 0.16 * (WTLF-WCRLF), 0.0)
```

Surface and soil residue contributions. Carbon, nitrogen and lignin in tissue losses are accumulated for addition to surface and soil residue. Senescence and freeze values are added to soil residue, while pest damage components are assumed to be lost from the system. Surface losses are computed from senesced and frozen leaf, stem and shell, while soil losses include roots and nodules.

Senesced and frozen tissue losses are combined for leaf (SLDOT, WLFDOT), stem (SSDOT), and shell (WTABRT) for addition to surface residue. Values are converted from biomass to C with a factor of 0.40.

```
SENSURFC = (SLDOT + WLFDOT + SSDOT + WTABRT) * 0.40
```

Surface nitrogen includes nitrogen losses previously computed minus the pest damage components.

```
 SENSURFN = (NLOFF - WLIDOT * PCNL / 100.) +      !Leaf
&           (NSOFF - WSIDOT * PCNST / 100.) +     !Stem
&           (NSHOFF - WSHIDT * PCNSH / 100.)      !Shell
```

Contribution of lignin to surface litter from senesced and frozen plant matter are calculated as:

```
 SENSURFL = (SLDOT + WLFDOT) * PLIGLF + SSDOT * PLIGST +
&    WTABRT * PLIGSH
```

Similarly, soil carbon, nitrogen and lignin are computed from losses of roots and nodules

```
SENSOILC = (SRDOT + NDTH) * 0.40
SENSOILN = NROFF - WRIDOT * PCNRT / 100. + NNOFF
SENSOILL = SRDOT * PLIGRT + NDTH * PLIGNO
```

Leaf Area Calculations. The rate of leaf expansion, ALFDOT in units of cm^2[leaf]/m^2[ground]-d, is equal to the addition of area from new leaf growth minus various leaf area losses. New leaf area growth is computed by multiplying the new leaf weight growth by the specific leaf area of new leaves (F), computed in the vegetative growth subroutine, VEGGR. Since abscised leaves are assumed to still contain CH_2O reserves, their average specific leaf area is SLA, computed by dividing leaf area by total leaf weight. Total leaf dry

weight lost is:

```
(SLDOT+WLIDOT+WLFDOT+NRUSLF/0.16).
```

Therefore, the equation for leaf area growth is:

```
      ALFDOT = WLDOTN*F - (SLDOT+WLIDOT+WLFDOT+NRUSLF/0.16)*SLA
      IF (WTLF .GT. 0.0) THEN

         ALFDOT = ALFDOT + SLA * (NADLF/0.16) *
     &    (1. - MIN(1.0,(SLDOT+WLIDOT+WLFDOT)/WTLF))
      ENDIF
```

and leaf area is updated by:

```
      AREALF = AREALF + ALFDOT
```

Leaf area index is computed by converting the units of AREALF from cm^2[leaf]/m^2[ground] to m^2[leaf]/m^2[ground] basis, or:

```
      XLAI   = AREALF / 10000
```

and specific leaf area is computed by dividing area by leaf weight, or:

```
      SLA    = AREALF / WTLF
```

The specific leaf area of leaf tissue is also computed as area of leaves divided by mass of leaves excluding stored CH_2O (SLAAD).

The maximum leaf area index per simulation is maintained as LAIMX.

Pest Damage Calculations. Area of healthy leaves (AREAH), and healthy leaf area index (XHLAI) are computed by:

```
      AREAH  = AREALF - 2*DISLA
      XHLAI  = AREAH / 10000
```

Daily pest damage to seed is computed in a pest module as number of destroyed seed (SDIDOT). Seed number is updated by:

```
      SEEDNO = SEEDNO - SDIDOT
```

Plant population and spacing between plants must be recomputed if pests destroy some of the plants. Plant population (PLTPOP) is updated by subtracting the number of plants damaged, NPLTD, or by reducing the population by a percentage damage, PPLTD, depending on the type of damage input to the model (Batchelor, et al, 1993). After updating PLTPOP, the between plant spacing is computed as:

```
      BETN = 1.0/(ROWSPC*PLTPOP).
```

Discussion. At the end of GROW, all crop state variables have been updated and program control is returned to CROPGRO. The state variables are the same set originally used in SOYGRO (Wilkerson et al., 1985) with additions to account for the storage and mobilization of non-structural carbohydrates. Therefore, the growth equations are also similar, with extensions to account for C storage and use. However, the process subroutines that compute the rate variables have been modified considerably from the original SOYGRO. The original version did not include C reserves. It also did not include N fixation or soil N uptake, although it did include N in the plant and its mobilization during seed growth.

17.0 References

Batchelor, W.D., J.W. Jones, K.J. Boote, and G. Hoogenboom. 1994. Carbon-based model to predict peanut pod detachment. Transactions of the ASAE 37(5):1639-1646.

Batchelor, W.D., J.W. Jones, K.J. Boote, C.H. Porter. 2000. Pest and Disease Damage Module, in DSSAT v.4.0 Documentation and Source Code Listing. Research Report Number 2000-103.

Boote, K.J., J.W. Jones and G.Hoogenboom. 1998. "Simulation of Crop Growth: CROPGRO Model", Chapter 18 in *Agricultural Systems Modeling and Simulation*, ed. R.M. Peart and R.B. Curry. Marcel Dekker, Inc., New York.

Godwin, D.C., and C.A. Jones. 1991. Nitrogen dynamics in soil-plant systems. p. 287-321. In J. Hanks and J.T. Ritchie (eds.) Modeling plant and soil systems. Agronomy monograph no. 31. ASA, CSSA, and SSSA, Madison, WI.

Hayati, R., D.B. Egli, and S.J. Crafts-Brandner. 1995. Carbon and N supply during seed filling and leaf senescence in soybean. Crop Sci. 35:1063-1069.

Jones, J.W., J. White, K. Boote, G. Hoogenboom, C.H. Porter. 2000. Phenology Module in DSSAT V.4.0 Documentation and Source Code Listing. Research Report Number 2000-102.

Jones, J.W., K.J. Boote, S.S. Jagtap, G. Hoogenboom and G.G. Wilkerson. 1989. SOYGRO V5.42. Soybean crop growth simulation model. User's guide. Florida Agr. Exp. Station Jouirnal No. 83.4. University of Florida, Gainesville.

Pate, J.S. 1985. Partitioning of carbon and nitrogen in N2-fixing grain legumes. pp715-725. In World Soybean Rearch Conference III: Proceedings. R. Shibles Ed. Westview Press, Boulder and London.

Porter, C.H. and J.W.Jones. 1998. Modular Structure in DSSAT Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Ross., I.A. 1988. Effects of moisture stress on the oil and protein components of soybean seeds. Aust. J. Agric. Res. 39:163-170.

Sato, K. and T. Ikeda. 1979. The growth responses of soybean to photoperiod and temperature. IV. The effect of temperature during the ripening period on the yield and characters of seeds. Japan. J. Crop. Sci. 48:283-290.

Streeter, J.G. 1978. Effect of N starvation of soybean plants at various stages of growth on seed yield and N concentration of plant parts at maturity. Agron. J. 70:74-76.

Tolley-Henry, L.T., C. D. Raper, Jr., and J.W. Rideout. 1992. Onset of and recovery from nitrogen stress during reproductive growth of soybean. Int. J. Plant Sci. 153:178-185.

Wilkerson, G.G., J.W. Jones, K.J. Boote, K.T. Ingram, and J.W. Mishoe. 1983. Modeling soybean growth for crop management. Trans. ASAE 26:63-73.

Wilkerson, G.G., J.W. Jones, K.J. Boote, and J.W. Mishoe. 1985. SOYGRO V5.0: Soybean crop growth and yield model. Technical Documentation. Agricultural Engineering Dept., University of Florida, Gainesville. 253 pp.

Wolf, R.B., J.F. Cavins, R. Kleiman, and L.T. Black. 1982. Effect of temperature on soybean seed constituents: oil, protein, moisture, fatty acids, amino acids, and sugars. J. Amer. Oil Chem. Soc. 59:230-232.

Table 16– Interface for CROPGRO Module			
Variable Name		**Definition**	**Units**
Input Arguments	CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	EOP	Potential plant transpiration rate	mm/d
	HARVFRAC	Two-element array containing fractions of (1) yield harvested and (2) by-product harvested	fraction
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]
	PAR	Daily photosynthetically active radiation or photon flux density	moles[quanta]/m2-d
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	TAVG	Average daily temperature	°C
	TDAY	Average temperature during daylight hours	°C
	TGRO(I)	Hourly canopy temperature	°C
	TGROAV	Average daily canopy temperature	°C
	TMIN	Minimum daily temperature	°C
	TRWUP	Potential daily root water uptake over soil profile	cm/d
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD
Output Arguments	CANHT	Canopy height	m
	EORATIO	Ratio of increase in potential evapotranspiration with increase in LAI (up to LAI=6.0) for use with FAO-56 Penman reference potential evapotranspiration.	--
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	KSEVAP	Light extinction coefficient used for computation of soil evaporation	--
	KTRANS	Light extinction coefficient used for computation of plant transpiration	--
	MDATE	Harvest maturity date	YYYYDDD
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	PORMIN	Minimum pore space required for supplying oxygen to roots for optimal growth and function	cm3/cm3
	RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
	RWUMX	Maximum water uptake per unit root length, constrained by soil water	cm3[water] / cm [root]
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	STGDOY(I)	Day when plant stage I occurred	YYYYDDD
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	XLAI	Leaf area (one side) per unit of ground area	m2[leaf] / m2[ground]

Table 17 – Interface for INCOMP Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
ECONO	Ecotype code - used to match ECOTYP in .ECO file	--
FILECC	Path plus filename for species file (*.spe)	--
FILEGC	Pathname plus filename for ECO file	--
FRLF	Fraction of vegetative tissue growth that goes to leaves on a day	g[leaf] / g[veg]
FRRT	Fraction of vegetative tissue growth that goes to roots on a day	g[root] / g[veg]
FRSTM	Fraction of vegetative tissue growth that goes to stems on a day	g[stem] / g[veg]
Output Arguments:		
AGRLF	Mass of CH_2O required for new leaf growth	g[CH_2O] / g[leaf]
AGRNOD	CH2O requirement for nodule growth	g[CH2O] / g[nodule]
AGRRT	Mass of CH_2O required for new root growth	g[CH_2O] / g[root]
AGRSD1	CH_2O requirement for seed growth, excluding cost for protein content	g[CH_2O] / g[seed]
AGRSD2	CH_2O requirement for seed growth, including cost for protein content	g[CH_2O] / g[seed]
AGRSH1	CH_2O required for shell growth, excluding cost for protein content	g[CH_2O] / g[shell]
AGRSH2	CH_2O requirement for shell growth, including cost for protein content	g[CH_2O] / g[shell]
AGRSTM	Mass of CH_2O required for new stem growth	g[CH_2O] / g[stem]
AGRVG	Mass of CH_2O required for vegetative tissue growth including stoichiometry and respiration	g[CH_2O] / g[tissue]
AGRVG2	Total mass of CH_2O required for vegetative tissue growth	g[CH_2O] / g[tissue]
SDPROR	Ratio to adjust lipid and carbohydrate proportions when seed protein differs from protein composition of standard cultivar (SDPROS)	--

Table 18 – Interface for RESPIR Routine		
Variable	**Definition**	**Units**
Input Arguments:		
PG	Daily gross photosynthesis	g[CH_2O] / m^2/ d
R30C2	Respiration coefficient that depends on total plant mass, value at 30C	g[CH_2O] used / g[CH_2O] fixed / hr
RES30C	Respiration coefficient that depends on gross photosynthesis, value at 30C	g [CH_2O] /g [DW]/hr
TGRO(I)	Hourly air temperature	°C
WTMAIN	Mass of tissue assumed to require maintenance	g[tissue] / m^2
Input/Output Arguments:		
RO	Respiration coefficient that depends on total plant mass	g[CH_2O] / g[tissue]
RP	Proportion of the day's photosynthesis which is respired in the maintenance process	--
Output Arguments:		
MAINR	Maintenance respiration	g[CH_2O]/m^2-d

Table 19 – Interface for DEMAND Routines		
Variable	**Definition**	**Units**
Input Arguments:		
AGRLF	Mass of CH_2O required for new leaf growth	g[CH_2O] / g[leaf]
AGRRT	Mass of CH_2O required for new root growth	g[CH_2O] / g[root]
AGRSH2	CH_2O requirement for shell growth, including cost for protein content	g[CH_2O] / g[shell]
AGRSTM	Mass of CH_2O required for new stem growth	g[CH_2O] / g[stem]
CROP	Crop identification code	--
DRPP	Photoperiod days which occur in a real day	photoperiod days / day
DXR57	Relative time between first seed (NR5) and physiological maturity (NR7)	--
FILECC	Path plus filename for species file (*.spe)	--
FILEGC	Pathname plus filename for ECO file	--
FILEIO	Filename for input file (e.g., DSSAT40.INP)	--
FNINSH	Maximum fraction of N for growing shell tissue	g[N] / g[shell]
FRACDN	Relative time between flowering (NR1) and last leaf appearance (NDLEAF)	--
LAGSD	Time required between shell growth and seed growth, per cohort	Photo-thermal days
LNGPEG	Time between start of peg and rapid shell formation (for peanuts only). Defines slow growth period.	Photo-thermal days
NDLEAF	Day when leaf expansion ceased	days
NR1	Day when 50% of plants have at least one flower	days
NR2	Day when 50% of plants have one peg (peanuts only)	days
NR5	Day when 50% of plants have pods with beginning seeds	days
NR7	Day when 50% of plants first have yellowing or maturing pods	days
NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
NVEG0	Day of emergence	days
PAR	Daily photosynthetically active radiation or photon flux density	MJ / m^2- d
PCNL	Percentage of N in leaf tissue	100 g[N] / g[leaf]
PCNRT	Percent N in root tissue	100 g[N] / g[root]
PCNST	Percent N in stem tissue	100 g[N] / g[stem]
PGAVL	Total available CH_2O available for growth & respiration	g[CH_2O] / m^2
PLTPOP	Plant population	# plants / m^2
PUNCSD	Cumulative puncture damage to seed (not yet implemented)	
PUNCTR	Cumulative puncture damage (not yet implemented)	
RPROAV	Respiration required for protein synthesis, average based on sources of N	g[CH_2O] / g[protein]
RTWT	Dry mass of root tissue, including C and N	g[root] / m^2 [ground]
SDDES(J)	Mass of seeds destroyed in cohort J by pests	g/ m^2/day
SDNO(J)	Number of seeds for cohort J	#/ m^2
SDVAR	Maximum cultivar-dependent seed growth rate, per seed	g / seed / d
SHELN(J)	Number of shells for cohort J	#/ m^2
SHVAR	Shell growth rate during its rapid growth phase, per shell	g / shell / d
STMWT	Dry mass of stem tissue, including C and N	g[stem] / m^2 [ground
SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress	--
TAVG	Average daily temperature	°C
TDUMX	Photo-thermal time that occurs in a real day based on early reproductive development temperature function	photo-thermal days / day
TDUMX2	Photo-thermal time that occurs in a real day based on late reproductive development temperature function	photo-thermal days / day
TGRO(I)	Hourly air temperature	°C
TURFAC	Water stress factor for expansion (0 - 1)	--
VSTAGE	Number of nodes on main stem of plant	--

Table 19 (continued) – Interface for DEMAND Routines		
Variable	**Definition**	**Units**
Input Arguments:		
WCRLF	Mass of CH_2O reserves in leaves	g[leaf CH_2O] / m^2 [ground]
WCRRT	Mass of CH_2O reserves in roots	g[root CH_2O] / m^2 [ground]
WCRST	Mass of CH_2O reserves in stems	g[stem CH_2O] / m^2 [ground]
WNRLF	N available for mobilization from leaves above lower limit of mining	g[N] / m^2
WNRRT	N available for mobilization from roots above lower limit of mining	g[N] / m^2
WNRSH	N available for mobilization from shells above lower limit of mining	g[N] / m^2
WNRST	N available for mobilization from stems above lower limit of mining	g[N] / m^2
WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m^2 [ground]
WTSD(J)	Seed mass for cohort J	g/ m^2
WTSHE(J)	Shell mass for cohort J	g/ m^2
XPOD	Growth partitioning to pods which slows node appearance	fraction
YRDOY	Current day of simulation	YYDDD
YRSIM	Start of simulation date	YYDDD
Output Arguments:		
AGRSD1	CH_2O requirement for seed growth, excluding cost for protein content	g[CH_2O] / g[seed]
AGRSD2	CH_2O requirement for seed growth, including cost for protein content	g[CH_2O] / g[seed]
AGRVG	Mass of CH_2O required for vegetative tissue growth including stoichiometry and respiration	g[CH_2O] / g[tissue]
AGRVG2	Total mass of CH_2O required for vegetative tissue growth	g[CH_2O] / g[tissue]
CDMREP	Total CH_2O needed for potential reproductive growth	g[CH_2O] / m^2/ d
F	Specific leaf area of new leaf tissue growth, including N	cm^2 [leaf] / g[leaf]
FNINL	Maximum fraction of N for growing leaf tissue	g[N] / g[leaf]
FNINR	Maximum fraction of N for growing root tissue	g[N] / g[root]
FNINS	Maximum fraction of N for growing stem tissue	g[N] / g[stem]
FNINSD	Maximum fraction of N for growing seed tissue based on temperature	g[N] / g[seed]
FRLF	Fraction of vegetative tissue growth that goes to leaves on a day	g[leaf] / g[veg]
FRRT	Fraction of vegetative tissue growth that goes to roots on a day	g[root] / g[veg]
FRSTM	Fraction of vegetative tissue growth that goes to stems on a day	g[stem] / g[veg]
GDMSD	Seed growth demand based on temperature and photoperiod	g[seed] / m^2/ d
GRRAT1	Maximum growth per individual shell	g / shell / d
NDMNEW	Total N demand for new growth	g[N] / m^2/ d
NDMOLD	N demand for old tissue	g[N] / m^2/ d
NDMREP	Total N needed for potential reproductive growth	g[N] / m^2/ d
NDMSDR	Amount of Mobilized N which can be used for seed growth	g[N] / m^2/ d
NDMTOT	Total N demand	g[N] / m^2/ d
NDMVEG	N required for vegetative growth if all PGAVL is used as computed	g[N] / m^2/ d
NMINEP	Potential N mobilization from storage	g[N] / m^2/ d
NMOBR	Stage dependent N mining rate	--
PHTIM(I)	Cumulative photothermal time ages of seeds and shells	--
PNTIM(I)	Photothermal days from first flower when flowers in age group I formed	p-t-d
POTCAR	Potential carbohydrate composition of seed based on temperature	fraction
POTLIP	Potential lipid composition of seed based on temperature	fraction
SDGR	Potential growth rate per seed	g / seed / d
TURADD	Water stress factor (TURFAC) effect on reproductive growth and pod addition. Stress is defined to INCREASE growth and addition.	--
XFRT	Current day's partitioning to reproductive growth (0-1)	g[fruit] / g[plant]

Table 20 – Interface for NUPTAK Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
BD(L)	Bulk density, soil layer L (g [soil] / cm3 [soil])	g [soil] / cm3 [soil]
DLAYR(L)	Thickness of soil layer L (cm)	cm
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L (cm3[water]/cm3[soil])	cm3[water]/cm3[soil]
FILECC	Path plus filename for species file (*.spe)	
LL(L)	Volumetric soil water content in soil layer L at lower limit (cm3 [water] / cm3 [soil])	cm3 [water] / cm3 [soil]
NDMSDR	Amount of Mobilized N which can be used for seed growth (g[N] / m2 / d)	g[N] / m2 / d
NDMTOT	Total N demand (g[N] / m2 / d)	g[N] / m2 / d
NDMTOT	Total N demand (g[N] / m2 / d)	g[N] / m2 / d
NH4(L)	Ammonium N in soil layer L (µg[N] / g[soil])	µg[N] / g[soil]
NLAYR	Actual number of soil layers	
NO3(L)	Nitrate in soil layer L (µg[N] / g[soil])	µg[N] / g[soil]
RLV(L)	Root length density for soil layer L (cm[root] / cm3[soil])	cm[root] / cm3[soil]
SAT(L)	Volumetric soil water content in layer L at saturation (cm3 [water] / cm3 [soil])	cm3 [water] / cm3 [soil]
SW(L)	Volumetric soil water content in layer L (cm3 [water] / cm3 [soil])	cm3 [water] / cm3 [soil]
Output Arguments:		
TRNH4U	Total N uptake in ammonium form in a day (kg[N] / ha / d)	kg[N] / ha / d
TRNO3U	Total N uptake in nitrate form in a day (kg[N] / ha / d)	kg[N] / ha / d
TRNU	Total N uptake in a day (kg[N] / ha / d)	kg[N] / ha / d
UNH4(L)	Rate of root uptake of NH4 in soil layer L (kg [N] / ha - d)	kg [N] / ha - d
UNO3(L)	Rate of root uptake of NO3 in soil layer L (kg [N] / ha -d)	kg [N] / ha -d

Table 21 – Interface for MOBIL Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
NDMNEW	Total N demand for new growth	g[N] / m^2 / d
NMINEP	Potential N mobilization from storage	g[N] / m^2/ d
NMOBR	Stage dependent N mining rate	--
RPRO	Respiration required for resynthesizing protein from mobilized protein	g[CH_2O] / g[protein]
TRNU	Soil nitrogen uptake	g[N] / m^2/ d
WNRLF	N available for mobilization from leaves above lower limit of mining	g[N] / m^2
WNRRT	N available for mobilization from roots above lower limit of mining	g[N] / m^2
WNRSH	N available for mobilization from shells above lower limit of mining	g[N] / m^2
WNRST	N available for mobilization from stems above lower limit of mining	g[N] / m^2
Output Arguments:		
NMINEA	Actual Nitrogen mined from existing tissue	g[N] / m^2/ d
NRUSLF	N actually mobilized from leaves in a day	g[N]/ m^2-d
NRUSRT	N actually mobilized from roots in a day	g[N]/ m^2-d
NRUSSH	N actually mobilized from shells in a day	g[N]/ m^2-d
NRUSST	N actually mobilized from stems in a day	g[N]/ m^2-d

Table 22 – Interface for NFIX Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
AGRNOD	CH_2O requirement for nodule growth	g[CH_2O] / g[nodule]
CNODMN	Minimum CH_2O reserved for nodule growth	g[CH_2O] / m^2/ d
CTONOD	CH_2O to allocate to nodules to fix N needed for reproductive and vegetative growth	g[CH_2O] / m^2/ d
DLAYR(L)	Soil Depth in layer L	cm
DXR57	Relative time between first seed (NR5) and physiological maturity (NR7)	--
FILECC	Pathname plus filename for SPE file	
FILEIO	Filename for INP file	
NLAYR	Number of soil layers	
NR7	Day when 50% of plants first have yellowing or maturing pods	days
PLTPOP	Plant population	# plants / m^2
SAT(L)	Saturated Soil water content in layer L	cm^3/ cm^3
ST(L)	Soil temperature by soil layer	ºC
SW(L)	Soil Water Content in layer L	cm^3/ cm^3
TURFAC	Water stress factor for expansion	--
YRDOY	Current day of simulation	YYDDD
YRSIM	Beginning date of simulation	YYDDD
Output Arguments:		
CNOD	CH_2O used in N-Fixation and nodule growth (including respiration costs) today	g[CH_2O] / m^2/ d
DWNOD	Nodule mass	g[nodule] / m^2
DWNODA	Nodule mass	g[nodule] / m^2
NDTH	Nodule death rate	g[nodule] / m^2/ d
NFIXN	Amount of N fixed during the day	g[N] / m^2/ d
NODGR	New nodule growth	g[nod] / m^2/ d
SENNOD	Daily senesced matter from nodules in soil layer L (kg[dry matter]/ha)	kg[dry matter]/ha
WTNFX	Cumulative weight of N fixed	g[N] / m^2

Table 23 – Interface for PODS Routines		
Variable Name	**Definition**	**Units**
Input Arguments:		
AGRSD1	CH_2O requirement for seed growth, excluding cost for protein content	g[CH_2O] / g[seed]
AGRSH1	CH_2O required for shell growth, excluding cost for protein content	g[CH_2O] / g[shell]
DLAYR(L)	Soil Depth in layer L	cm
DRPP	Photoperiod days which occur in a real day	photoperiod days / day
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L	cm^3 [H_2O] / cm^3 [soil]
FILECC	Path plus filename for species file (*.spe)	--
FILEGC	Pathname plus filename for ECO file	--
FILEIO	Filename for input file (e.g., DSSAT40.INP)	--
FNINL	Maximum fraction of N for growing leaf tissue	g[N] / g[leaf]
FNINSD	Maximum fraction of N for growing seed tissue based on temperature	g[N] / g[seed]
FNINSH	Maximum fraction of N for growing shell tissue	g[N] / g[shell]
GDMSD	Seed growth demand based on temperature and photoperiod	g[seed] / m^2 / d
GRRAT1	Maximum growth per individual shell	g / shell / d
ISWWAT	Water simulation control switch (Y or N)	
LL(L)	Volumetric soil water content in soil layer L at lower limit	cm^3/ cm^3
NAVL	Total mass of nitrogen available for growth	g[N] / m^2/ d
NDSET	Normal time by which a pod load (full number) should be achieved with no water stress	days
NLAYR	Actual number of soil layers	--
NRUSSH	N actually mobilized from shells in a day	g[N]/ m^2-d
NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
PGAVL	Total available CH_2O available for growth & respiration	g[CH_2O] / m^2
PHTHRS(I)	Threshold time that must accumulate in phase I for the next stage to occur	thermal or photothermal days
PHTIM(I)	Cumulative photothermal time ages of seeds and shells	--
PNTIM(I)	Photothermal days from first flower when flowers in age group I formed	p-t-d
PUNCSD	Cumulative puncture damage to seed (not yet implemented)	
PUNCTR	Cumulative puncture damage (not yet implemented)	
RNITP	True nitrogen concentration in leaf tissue for photosynthesis reduction.	%
SDDES(J)	Mass of seeds destroyed in cohort J by pests	g/ m^2/day
SDGR	Potential growth rate per seed	g / seed / d
SHELWT	Total mass of all shells	g / m^2
SW(L)	Volumetric soil water content in layer L	cm^3 [water] / cm^3 [soil]
SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress	--
TDUMX	Photo-thermal time that occurs in a real day based on early reproductive development temperature function	photo-thermal days / day
TGRO(I)	Hourly air temperature	°C
TURADD	Water stress factor (TURFAC) effect on reproductive growth and pod addition. Stress is defined to INCREASE growth and addition.	--
XFRT	Current day's partitioning to reproductive growth (0-1)	g[fruit] / g[plant]
YRDOY	Current day of simulation	YYDDD
YRNR1	Day when 50% of plants have at least one flower	YYDDD
YRNR2	Day when 50% of plants have one peg (peanuts only)	YYDDD
YRSIM	Start of simulation date	YYDDD

Table 23 (continued) – Interface for PODS Routines		
Variable Name	**Definition**	**Units**
Output Arguments:		
AGRSD3	CH_2O requirement for seed growth, with reduced N content	g[CH_2O] / g[seed]
FLWN(J)	Number of flowers added for cohort J	# / m^2
LAGSD	Time required between shell growth and seed growth, per cohort	Photo-thermal days
LNGPEG	Time between start of peg and rapid shell formation (for peanuts only). Defines slow growth period.	Photo-thermal days
NGRSD	Rate of N accumulation in new seeds	g[N] / m^2/ d
NGRSH	Rate of N accumulation in new shells	g[N] / m^2/ d
PCTMAT	Fraction of pods that are mature (seed are 90% of final size)	--
PODNO	Total number of pods	#/ m^2
POTCAR	Potential carbohydrate composition of seed based on temperature	fraction
POTLIP	Potential lipid composition of seed based on temperature	fraction
SDNO(J)	Number of seeds for cohort J	#/ m^2
SDVAR	Maximum cultivar-dependent seed growth rate, per seed	g / seed / d
SEEDNO	Total number of seeds	#/ m^2
SHELN(J)	Number of shells for cohort J	#/ m^2
SHVAR	Shell growth rate during its rapid growth phase, per shell	g / shell / d
WSDDTN	New seed growth today	g[seed] / m^2/ d
WSHDTN	New shell growth today	g[shell] / m^2/ d
WTABRT	Weight of shells aborted on a day	g[shell] / m^2/ d
WTSD(J)	Seed mass for cohort J	g/ m^2
WTSHE(J)	Shell mass for cohort J	g/ m^2
WTSHMT	Cohorts that reach THRESH today	g/ m^2

Table 24 – Interface for PODDET		
Variable Name	**Definition**	**Units**
Input Arguments:		
FILECC	Path plus filename for species file (*.spe)	--
TGRO(I)	Hourly air temperature	ºC
WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m^2 [ground]
YRDOY	Current day of simulation	YYDDD
YRNR2	Day when 50% of plants have one peg (peanuts only)	YYDDD
Output Arguments:		
PODWTD	Mass of detached pods	g[pods] / m^2 [ground]
SDNO(J)	Number of seeds for cohort J	#/ m^2
SHELN(J)	Number of shells for cohort J	#/ m^2
SWIDOT	Daily seed mass damage	g/ m^2/day
WSHIDT	Weight of shell tissue consumed by pests today	g[shell]/ m^2-d
WTSD(J)	Seed mass for cohort J	g/ m^2
WTSHE(J)	Shell mass for cohort J	g/ m^2

Table 25 – Interface for VEGGR Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
AGRLF	Mass of CH_2O required for new leaf growth	g[CH_2O] / g[leaf]
AGRRT	Mass of CH_2O required for new root growth	g[CH_2O] / g[root]
AGRSTM	Mass of CH_2O required for new stem growth	g[CH_2O] / g[stem]
CMINEP	Potential CH_2O mobilization from storage	g[CH_2O] / m^2 / d
CSAVEV	Fraction of PG for VEG that is stored as CH_2O	--
DTX	Thermal time that occurs in a real day based on vegetative development temperature function	thermal days / day
DXR57	Relative time between first seed (NR5) and physiological maturity (NR7)	--
ECONO	Ecotype code - used to match ECOTYP in .ECO file	--
FILECC	Path plus filename for species file (*.spe)	--
FILEGC	Pathname plus filename for ECO file	--
FNINL	Maximum fraction of N for growing leaf tissue	g[N] / g[leaf]
FNINR	Maximum fraction of N for growing root tissue	g[N] / g[root]
FNINS	Maximum fraction of N for growing stem tissue	g[N] / g[stem]
NAVL	Total mass of nitrogen available for growth	g[N] / m^2/ d
NDMNEW	Total N demand for new growth	g[N] / m^2/ d
NDMOLD	N demand for old tissue	g[N] / m^2/ d
NFIXN	Amount of N fixed during the day	g[N] / m^2/ d
NMINEA	Actual Nitrogen mined from existing tissue	g[N] / m^2/ d
NR1	Day when 50% of plants have at least one flower	days
PAR	Daily photosynthetically active radiation or photon flux density	MJ / m^2- d
PCH2O	Respiration loss due to storage/mobilization of CH_2O	g[CH_2O] / g[CH_2O]
PG	Daily gross photosynthesis	g[CH_2O] / m^2/ d
PGAVL	Total available CH_2O available for growth & respiration	g[CH_2O] / m^2
ROWSPC	Row spacing	m
RVSTGE	Rate of VSTAGE change	nodes/day
STMWT	Dry mass of stem tissue, including C and N	g[stem] / m^2 [ground
TGRO(I)	Hourly air temperature	°C
TRNU	Total N uptake in a day	g[N] / m^2/ d
TURFAC	Water stress factor for expansion (0 - 1)	--
VSTAGE	Number of nodes on main stem of plant	--
WCRLF	Mass of CH_2O reserves in leaves	g[leaf CH_2O] / m^2 [ground]
WCRRT	Mass of CH_2O reserves in roots	g[root CH_2O] / m^2 [ground]
WCRSH	Mass of CH_2O reserves in shells	g[shell CH_2O] / m^2 [ground]
WCRST	Mass of CH_2O reserves in stems	g[stem CH_2O] / m^2 [ground]
WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m^2 [ground]
XLAI	Leaf area (one side) per unit of ground area	m2[leaf] / m^2 [ground]
YRDOY	Current day of simulation	YYYYDDD
YREMRG	Day of emergence	YYYYDDD
YRSIM	Start of simulation date	YYYYDDD

Table 25 (continued) – Interface for VEGGR Routine		
Variable Name	**Definition**	**Units**
Input/Output Arguments:		
AGRVG	Mass of CH_2O required for vegetative tissue growth including stoichiometry and respiration	g[CH_2O] / g[tissue]
FRLF	Fraction of vegetative tissue growth that goes to leaves on a day	g[leaf] / g[veg]
FRRT	Fraction of vegetative tissue growth that goes to roots on a day	g[root] / g[veg]
FRSTM	Fraction of vegetative tissue growth that goes to stems on a day	g[stem] / g[veg]
Output Arguments:		
CADLF	Mass of CH_2O added to leaf reserves after growth	g[CH_2O] / m^2 / d
CADST	Mass of CH_2O added to stems	g[CH_2O] / m^2/ d
CANHT	Canopy height	m
CANWH	Canopy width normal to row	m
CMINEA	Actual carbon mined from vegetative tissue	g[CH_2O] / m^2/ d
CRUSLF	C mobilized from leaf tissue in a day	g[CH_2O] / m^2/ d
CRUSRT	C mobilized from root tissue in a day	g[CH_2O] / m^2/ d
CRUSSH	C mobilized from shell tissue in a day	g[CH_2O] / m^2/ d
CRUSST	C mobilized from stem tissue in a day	g[CH_2O] / m^2/ d
EXCESS	Factor based on excess PG used to affect tomorrow's PG calculation	--
NADLF	N added to leaf N reserves	g[N] / m^2/ d
NADRT	N added to root N reserves	g[N] / m^2/ d
NADST	N added to stem N reserves	g[N] / m^2/ d
NGRLF	Maximum N demand for leaf growth	g[leaf N] / m^2 [ground] / d
NGRRT	Maximum N demand for root growth	g[root N] / m^2 [ground] / d
NGRST	Maximum N demand for stem growth	g[stem N] / m^2 [ground] / d
NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
TNLEAK	Total nitrogen leak	g[N] / m^2/ d
WLDOTN	Dry weight growth rate of new leaf tissue including N but not C reserves	g[leaf] / m^2 [ground]-d
WRDOTN	Dry weight growth rate of new root tissue including N but not C reserves	g[root] / m^2 [ground]-d
WSDOTN	Dry weight growth rate of new stem tissue including N but not C reserves	g[stem] / m^2 [ground]-d

Table 26 – Interface for SENES Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
CLW	Cumulative leaf growth	g[leaf]/m2
DTX	Thermal time that occurs in a real day based on vegetative development temperature function	thermal days / day
FILECC	Path plus filename for species file (*.spe)	--
NR7	Day when 50% of plants first have yellowing or maturing pods	days
NRUSLF	N actually mobilized from leaves in a day	g[N]/ m^2-d
PAR	Daily photosynthetically active radiation or photon flux density	MJ / m^2- d
RHOL	Fraction of leaf which is carbohydrate	g [CH20] / g[leaf]
SLAAD	Specific leaf area, excluding weight of C stored in leaves	cm^2 [leaf] / g[leaf]
STMWT	Dry mass of stem tissue, including C and N	g[stem] / m^2 [ground
SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress	
VSTAGE	Number of nodes on main stem of plant	--
WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m^2 [ground]
XLAI	Leaf area (one side) per unit of ground area	m2[leaf] / m^2 [ground]
YRDOY	Current day of simulation	YYDDD
YRSIM	Start of simulation date	YYDDD
Output Arguments:		
SLDOT	Defoliation due to daily leaf senescence	g/ m^2/day
SLNDOT	Leaf senescence due to water stress	g/ m^2/day
SSDOT	Daily senescence of petioles	g / m^2/ d
SSNDOT	Petiole senescence due to water stress	g/ m^2/day

Table 27 – Interface for FREEZE Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
FREEZ2	Temperature below which plant growth stops completely.	°C
IDETO	Switch for printing OVERVIEW.OUT file	--
NOUTDO	Logical unit for OVERVIEW.OUT file	--
NRUSLF	N actually mobilized from leaves in a day	g[N]/ m^2-d
SLDOT	Defoliation due to daily leaf senescence	g/ m^2/day
TMIN	Minimum daily temperature	°C
WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m^2 [ground]
YRDOY	Current day of simulation	YYDDD
YRPLT	Planting date	YYDDD
Input/Output Arguments:		
MDATE	Date of harvest maturity	YYDDD
Output Arguments:		
WLFDOT	Leaf weight losses due to freezing	g[leaf]/ m^2-d

Table 28 – Interface for ROOTS Routine		
Variable Name	**Definition**	**Units**
Input Arguments:		
AGRRT	Mass of CH_2O required for new root growth	g[CH_2O] / g[root]
CROP	Crop identification code	--
DLAYR(L)	Thickness of soil layer L	cm
DS(L)	Cumulative depth in soil layer L	cm
DTX	Thermal time that occurs in a real day based on vegetative development temperature function	thermal days / day
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L	cm3[water]/cm3[soil]
FILECC	Path plus filename for species file (*.spe)	--
FRRT	Fraction of vegetative tissue growth that goes to roots on a day	g[root] / g[veg]
ISWWAT	Water simulation control switch (Y or N)	
LL(L)	Volumetric soil water content in soil layer L at lower limit	cm3 [water] / cm3 [soil]
NLAYR	Actual number of soil layers	
PG	Daily gross photosynthesis	g[CH_2O] / m^2/ d
PLTPOP	Plant population	# plants / m^2
RO	Respiration coefficient that depends on total plant mass	g[CH_2O] / g[tissue]
RP	Proportion of the day's photosynthesis which is respired in the maintenance process	--
RTWT	Dry mass of root tissue, including C and N	g[root] / m^2 [ground]
SAT(L)	Volumetric soil water content in layer L at saturation	cm3 [water] / cm3 [soil]
SW(L)	Volumetric soil water content in layer L	cm^3 [water] / cm^3 [soil]
SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress	--
VSTAGE	Number of nodes on main stem of plant	--
WR(L)	Root hospitality factor, used to compute root distribution	
WRDOTN	Dry weight growth rate of new root tissue including N but not C reserves	g[root] / m^2 [ground]-d
WTNEW	Initial mass of seedling or seed	g / plant
Output Arguments:		
RLV(L)	Root length density for soil layer L	cm root / cm^3soil
RTDEP	Root depth	cm
SATFAC	Root length weighted soil water excess stress factor (0 = no stress; 1 = saturated stress)	
SENRT(L)	Daily senesced matter from roots in soil layer L	kg[dry matter]/ha
SRDOT	Daily root senescence	g / m^2/ d

Table 29 – Interface for GROW Routines		
Variable Name	**Definition**	**Units**
Input Arguments:		
AGEFAC	Reduction in photosynthesis, correlated to a decrease in percentage of Nitrogen in leaves (0-1)	fraction
CADLF	Mass of CH_2O added to leaf reserves after growth	g[CH_2O] / m^2/ d
CADST	Mass of CH_2O added to stems	g[CH_2O] / m^2/ d
CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	
CRUSLF	C mobilized from leaf tissue in a day	g[CH_2O] / m^2/ d
CRUSRT	C mobilized from root tissue in a day	g[CH_2O] / m^2/ d
CRUSSH	C mobilized from shell tissue in a day	g[CH_2O] / m^2/ d
CRUSST	C mobilized from stem tissue in a day	g[CH_2O] / m^2/ d
DISLA	Diseased leaf area	cm2/ m^2/d
F	Specific leaf area of new leaf tissue growth, including N	cm^2 [leaf] / g[leaf]
FILECC	Path plus filename for species file (*.spe)	--
FILEGC	Pathname plus filename for ECO file	--
FRLF	Fraction of vegetative tissue growth that goes to leaves on a day	g[leaf] / g[veg]
FRSTM	Fraction of vegetative tissue growth that goes to stems on a day	g[stem] / g[veg]
ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
NADLF	N added to leaf N reserves	g[N] / m^2/ d
NADRT	N added to root N reserves	g[N] / m^2/ d
NADST	N added to stem N reserves	g[N] / m^2/ d
NDTH	Nodule death rate	g[nodule] / m^2/ d
NFIXN	Amount of N fixed during the day	g[N] / m^2/ d
NGRLF	Maximum N demand for leaf growth	g[leaf N] / m^2 [ground] / d
NGRRT	Maximum N demand for root growth	g[root N] / m^2 [ground] / d
NGRSD	Rate of N accumulation in new seeds	g[N] / m^2/ d
NGRSH	Rate of N accumulation in new shells	g[N] / m^2/ d
NGRST	Maximum N demand for stem growth	g[stem N] / m^2 [ground] / d
NMINEA	Actual Nitrogen mined from existing tissue	g[N] / m^2/ d
NODGR	New nodule growth	g[nod] / m^2/ d
NOUTDO	Logical unit for OVERVIEW.OUT file	--
NPLTD	Number of plants destroyed	#/ m^2/d
NRUSLF	N actually mobilized from leaves in a day	g[N]/ m^2-d
NRUSRT	N actually mobilized from roots in a day	g[N]/ m^2-d
NRUSSH	N actually mobilized from shells in a day	g[N]/ m^2-d
NRUSST	N actually mobilized from stems in a day	g[N]/ m^2-d
POTCAR	Potential carbohydrate composition of seed based on temperature	fraction
POTLIP	Potential lipid composition of seed based on temperature	fraction
PPLTD	Percent plants destroyed	%/ m^2/d
SDIDOT	Number of seeds destroyed on the current day	#/ m^2/d
SDPROR	Ratio to adjust lipid and carbohydrate proportions when seed protein differs from protein composition of standard cultivar (SDPROS)	--
SENNOD(L)	Daily senesced matter from nodules in soil layer L	kg[dry matter]/ha
SENRT(L)	Daily senesced matter from roots in soil layer L	kg[dry matter]/ha
SLDOT	Defoliation due to daily leaf senescence	g/ m^2/day
SLNDOT	Leaf senescence due to water stress	g/ m^2/day

Table 29 (continued) – Interface for GROW Routines		
Variable Name	**Definition**	**Units**
Input Arguments (continued):		
SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
SRDOT	Daily root senescence	g / m^2/ d
SSDOT	Daily senescence of petioles	g / m^2/ d
SSNDOT	Petiole senescence due to water stress	g/ m^2/day
TRNH4U	Total N uptake in ammonium form in a day	g[N] / m^2/ d
TRNO3U	Total N uptake in nitrate form in a day	g[N] / m^2/ d
TRNU	Total N uptake in a day	g[N] / m^2/ d
TURFAC	Water stress factor for expansion (0 - 1)	--
WLDOTN	Dry weight growth rate of new leaf tissue including N but not C reserves	g[leaf] / m^2 [ground]-d
WLIDOT	Daily pest or freeze damage to leaf mass	g/ m^2/day
WRDOTN	Dry weight growth rate of new root tissue including N but not C reserves	g[root] / m^2 [ground]-d
WRIDOT	Daily pest damage to root mass	g/ m^2/day
WSDDTN	New seed growth today	g[seed] / m^2/ d
WSDOTN	Dry weight growth rate of new stem tissue including N but not C reserves	g[stem] / m^2 [ground]-d
WSHDTN	New shell growth today	g[shell] / m^2/ d
WSIDOT	Daily pest damage to stem mass	g/ m^2/day
WTABRT	Weight of shells aborted on a day	g[shell] / m^2/ d
WTSHMT	Cohorts that reach THRESH today	g/ m^2
YRNR1	Day when 50% of plants have at least one flower	YYDDD
MDATE	Date of harvest maturity	YYDDD
YRPLT	Planting date	YYDDD
Input / Output Arguments:		
SWIDOT	Daily seed mass damage	g/ m^2/day
WLFDOT	Leaf weight losses due to freezing	g[leaf]/ m^2-d
WSHIDT	Weight of shell tissue consumed by pests today	g[shell]/ m^2-d
WTNFX	Cumulative weight of N fixed	g[N] / m^2
XHLAI	Leaf area index	m2[leaf] / m^2 [ground]
Output Arguments:		
AREALF	Area of leaves (one side) per unit ground area	cm2[leaf] / m^2 [ground]
BETN	Spacing between plants in a row	m / plant
CANNAA	Weight of N in total plant at flowering	g[N] / m2
CANWAA	Canopy weight at flowering stage	g[plant] / m^2
CLW	Cumulative leaf growth	g[leaf]/ m^2-d
CSW	Cumulative stem growth	g[stem]/ m^2-d
DWNOD	Current nodule mass	g[nodule] / m^2
DWNODA	Cumulative nodule growth	g[nodule] / m^2
GROWTH	Total new growth of plant tissue on a day	g[tissue] / m^2/ d
GRWRES	Growth respiration	g[CH_2O]/ m^2-d
LAIMX	Maximum leaf area index this season	m2[leaf] / m^2 [ground]
PCCSD	Percentage of carbohydrate in seed tissue	100 g[C] / g[seed]
PCLSD	Percentage of lipid in seed tissue	100 g[lipid] / g[seed]
PCNL	Percentage of N in leaf tissue	100 g[N] / g[leaf]
PCNRT	Percent N in root tissue	100 g[N] / g[root]

Table 29 (continued) – Interface for GROW Routines		
Variable Name	**Definition**	**Units**
Output Arguments (continued):		
PCNSD	Percentage of N in seed tissue	100 g[N] / g[seed]
PCNSH	Percentage of N in shell tissue	100 g[N] / g[shell]
PCNST	Percent N in stem tissue	100 g[N] / g[stem]
PLTPOP	Plant population	# plants / m^2
PLIGLF	Proportion of leaf tissue that is lignin	fraction
PLIGNO	Proportion of nodule tissue that is lignin	fraction
PLIGRT	Proportion of root tissue that is lignin	fraction
PLIGSD	Proportion of seed tissue that is lignin	fraction
PLIGSH	Proportion of shell tissue that is lignin	fraction
PLIGST	Proportion of stem tissue that is lignin	fraction
PODWT	Dry mass of seeds plus shells, including C and N	g[pods] / m^2 [ground]
PUNCSD	Cumulative puncture damage to seed (not yet implemented)	
PUNCTR	Cumulative puncture damage (not yet implemented)	
RHOL	Fraction of leaf which is carbohydrate	g [CH20] / g[leaf]
RHOS	Fraction of stem which is carbohydrate	g [CH_2O] / g[stem]
RNITP	True nitrogen concentration in leaf tissue for photosynthesis reduction.	%
ROWSPC	Row spacing	m
RTWT	Dry mass of root tissue, including C and N	g[root] / m^2 [ground]
SDNPL	Seed N	g[N] / m^2
SDRATE	Seeding rate, mass of seed sown	g[seed] / m^2 [ground]
SDWT	Dry mass of seed tissue, including C and N	g[seed] / m^2 [ground]
SDWTAM	Seed weight at maturity	g / m^2
SEEDNI	Seed or transplant N at planting	g[N] / m^2
SEEDNO	Total number of seeds	#/ m^2
SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
SHELWT	Total mass of all shells	g / m^2
SLA	Specific leaf area	cm^2 [leaf] / m^2 [ground]
SLAAD	Specific leaf area, excluding weight of C stored in leaves	cm^2 [leaf] / g[leaf]
STMWT	Dry mass of stem tissue, including C and N	g[stem] / m^2 [ground
TOPWT	Total weight of above-ground portion of crop, including pods	g[tissue] / m^2
TOTWT	Total weight of crop	g[tissue] / m^2
WCRLF	Mass of CH_2O reserves in leaves	g[leaf CH_2O] / m^2 [ground]
WCRRT	Mass of CH_2O reserves in roots	g[root CH_2O] / m^2 [ground]
WCRSH	Mass of CH_2O reserves in shells	g[shell CH_2O] / m^2 [ground]
WCRST	Mass of CH_2O reserves in stems	g[stem CH_2O] / m^2 [ground]
WNRLF	N available for mobilization from leaves above lower limit of mining	g[N] / m^2
WNRRT	N available for mobilization from roots above lower limit of mining	g[N] / m^2
WNRSH	N available for mobilization from shells above lower limit of mining	g[N] / m^2

PHENOLOGY SUB-MODULE FOR CROPGRO

March 31, 2004

J.W. Jones, K. Boote, J. White, G. Hoogenboom, C.H. Porter

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
Iowa State University, Ames, IA
International Consortium for Agricultural Systems Applications

Phenology Module
TABLE OF CONTENTS

1.0 Phenology Model Overview 1

2.0 Phenology Module Structure 3
2.1 Subroutine PHENOL 3
2.2 Subroutine IPPHENOL 8
2.3 Subroutine INPHENOL 9
2.4 Subroutine VSTAGES 9
2.5 Subroutine RSTAGES 10

3.0 Theory and Equations 14

4.0 References 20

LIST OF TABLES

Table 1 - Development stages 2
Table 2 – Interface Variables for Phenology Module 5
Table 3 - Phases of Crop Development 15
Table 4 - Temperature Function Values 17
Table 5 - Cultivar and Ecotype parameters 19

LIST OF FIGURES

Figure 1 - Phenology Module Structure 4
Figure 2 - Relative sensitivity of three soybean cultivars to daylength. 16
Figure 3 - Relative effect of temperature on developmental rate of soybean, for each of the three periods of growth 17

1.0 Phenology Model Overview

Crop models require accurate prediction of timing of developmental stages such as flowering or physiological maturity in order to control partitioning among organs and to determine the duration of growth and dry matter accumulation. The importance of phenology is also apparent in field experiments since timing of flowering and maturity are often critical determinants of productivity (White and Izquierdo, 1991). Laing et al. (1984) showed that much of the apparent differences in yield potential among grain legumes could be explained by differences in time to maturity.

Simple crop models and those developed for use over a limited range of environments sometimes use phenological data as inputs, but most simulation models predict phenology primarily based on temperature data over the growing season and, if the crop is photoperiod sensitive, on daylength. Inherent in this approach is the assumption that the developmental processes underlying phenology are relatively insensitive to the assimilate, nutrient, or water balances of the plant. This contrasts with growth processes where these factors are of primary importance.

In SOYGRO, PNUTGRO, and BEANGRO models, time to a given developmental event was predicted as a function of temperature (hourly values), daylength, and cultivar specific parameters (Jones et al., 1989; Boote et al., 1989; Hoogenboom et al., 1991). These events, called developmental stages, include emergence, first flower appearance, and physiological maturity. In describing how the models predict events, the first occurrence of an event is called a stage, while the interval between stages is called a phase. Jones et al. (1991) reviewed the history of the phenology model in SOYGRO and discussed other approaches for modeling phenology in soybean. Grimm et al. (1993, 1994) described the most recent results for soybean, much of which was instrumental in the phenology module in the current version of CROPGRO.

The vegetative and reproductive stages used to describe development of crops simulated with the CROPGRO generic model code are based on those used in the earlier SOYGRO V5.42, BEANGRO V1.01 and PNUTGRO V1.02 models (Table 1). In addition to the reproductive stages defined in Table 1, the models maintain the actual V-stages, which is the number of leaf nodes on the main axis of the plants. To avoid confusion with V-stages, the days when vegetative stages occur, as defined in Table 1, start with the letter N to designate a day when the event occurred, such as the first true leaf. The variable VSTAGE is used in the model to describe the average number of mainstem leaves on the plants.

The stages defined in Table 1 were originally based on the descriptions of stages for soybean (Fehr and Caviness, 1977) and for peanut (Boote, 1982). Only the stages with variable names identified are explicitly included in the models. The actual stages for dry beans are defined similarly to the system proposed by Schoonhoven and Pastor-Corrales (1987), but it is important to note that the code names for stages are not the same.

The phenology subroutine of the earlier models was completely rewritten for the CROPGRO structure. In this generic model, the development rates are computed in the subroutine PHENOL, and the model includes the effects of temperature, day length, soil water, and

nitrogen. A second subroutine, RSTAGES, is called from PHENOL to accumulate development for each phase and to determine the date when each stage occurs. V-stages are updated in the subroutine VSTAGES.

Table 1 - Development stages

Defines development stages used for crops (soybean, peanut, and dry bean) in the CROPGRO generic model code. Stage names are based on the stage definitions for soybean by Fehr and Caviness (1971) and for peanut by Boote (1982). Variable names are the names of variables used in the computer code to define the day of occurrence of a particular stage.

STAGE	VARIABLE NAME	STAGE DESCRIPTION
Vegetative:		
VE	NVEG0	First day with 50 Percent of plants with some part visible at soil surface
	NVEG1	First day with 50 Percent of plants with completely unrolled leaf at unifoliate node
V1		First day with 50 Percent of plants with completely unrolled leaf at first node above the unifoliate node
V2		First day with 50 Percent of plants with 2 leaves above the unifoliate on the main stem
V(n)		First day with 50 Percent of plants with n leaves above the unifoliate on the main stem
	NDVST	Day on which last main stem node formed
	NDLEAF	Day when leaf expansion ceased (often on branches)
Reproductive:		
R00	JPEND	Day when juvenile phase ends and plants first become sensitive to photoperiod
R0	NR0	Day when floral induction occurs
R1	NR1	Day when 50 Percent of plants have at least one flower at any node on the plant
R2	NR2	Day when 50 Percent of plants have one peg at any node (for peanut only)
R3+	NR3	Day when 50 Percent of plants have at least one pod formed (at least 0.5 cm in length) and ready to grow
R4+		Day when 50 Percent of plants have at least one fully expanded pod
R5+	NR5	Day when 50 Percent of plants have pods with seeds beginning to grow
R6+		Day when 50 Percent of plants have at least one pod containing a full-sized green seed
	NDSET	Day when last pod can form
R7	NR7	Day when 50 Percent of plants first have at least one pod that is yellowing, physiological maturity
R8	NR8	Day when 50 Percent of plants have at least 95 Percent of pods brown, harvest maturity

+Intentionally deviates from Fehr and Caviness (1971) to designate first occurrences of reproductive trait rather than looking at top 4 nodes with fully expanded leaves on the plant.

2.0 Phenology Module Structure

The phenology module consists of the following components:

PHENOL – Main routine of module, which calls other components.
IPPHENOL – Data input routine
INPHENOL – Variable initialization routine
VSTAGES – Vegetative growth stages routine
RSTAGES – Reproductive growth stages routine

Figure 1 illustrates the structure of the phenology module. Each component of the module is discussed in the following sections. Table 2 lists input and output data associated with the Phenology Module.

Appendix A provides a complete listing of all variables used in the Phenology module with their definitions, units and data type. Appendix B lists the FORTRAN code for the five components of the Phenology module.

2.1 Subroutine PHENOL

PHENOL, the main component of the phenology module, is called four times from the main CROPGRO program: once at the beginning of each simulation for input of data (RUNINIT); once at the beginning of each season of simulation for initialization of variables (SEASINIT); twice each day during the daily simulation loop for rate calculations (RATE) and integration (INTEGR). CROPGRO does not include a separate output file for phenological data and therefore, the module components 'OUTPUT' and 'FINAL' are not included in this module.

2.1.1 Run Initialization

Subroutine IPPHENOL is called to read data from input files. Subroutine INPHENOL is called to perform initialization of phenology variables, which are to be calculated once only at the beginning of simulation.

2.1.2 Seasonal Initialization

During the seasonal initialization section of the routine, initial values for phenology variables are defined for the beginning of each seasonal simulation. Subroutines VSTAGES and RSTAGES are called to perform initialization of variables.

2.1.3 Rate Calculations

In the rate calculation section of PHENOL, the average temperature (TSDEP) and soil water content (SWFEM) in the top 10 cm of soil are computed for the emergence phase (NVEG0 > DAS). The average temperature in the top 10 cm is used to drive development during the planting to emergence phase of development. The minimum time for emergence depends on an input development threshold for germination plus a time for plant to reach the surface that depends on planting depth (described in RSTAGES).

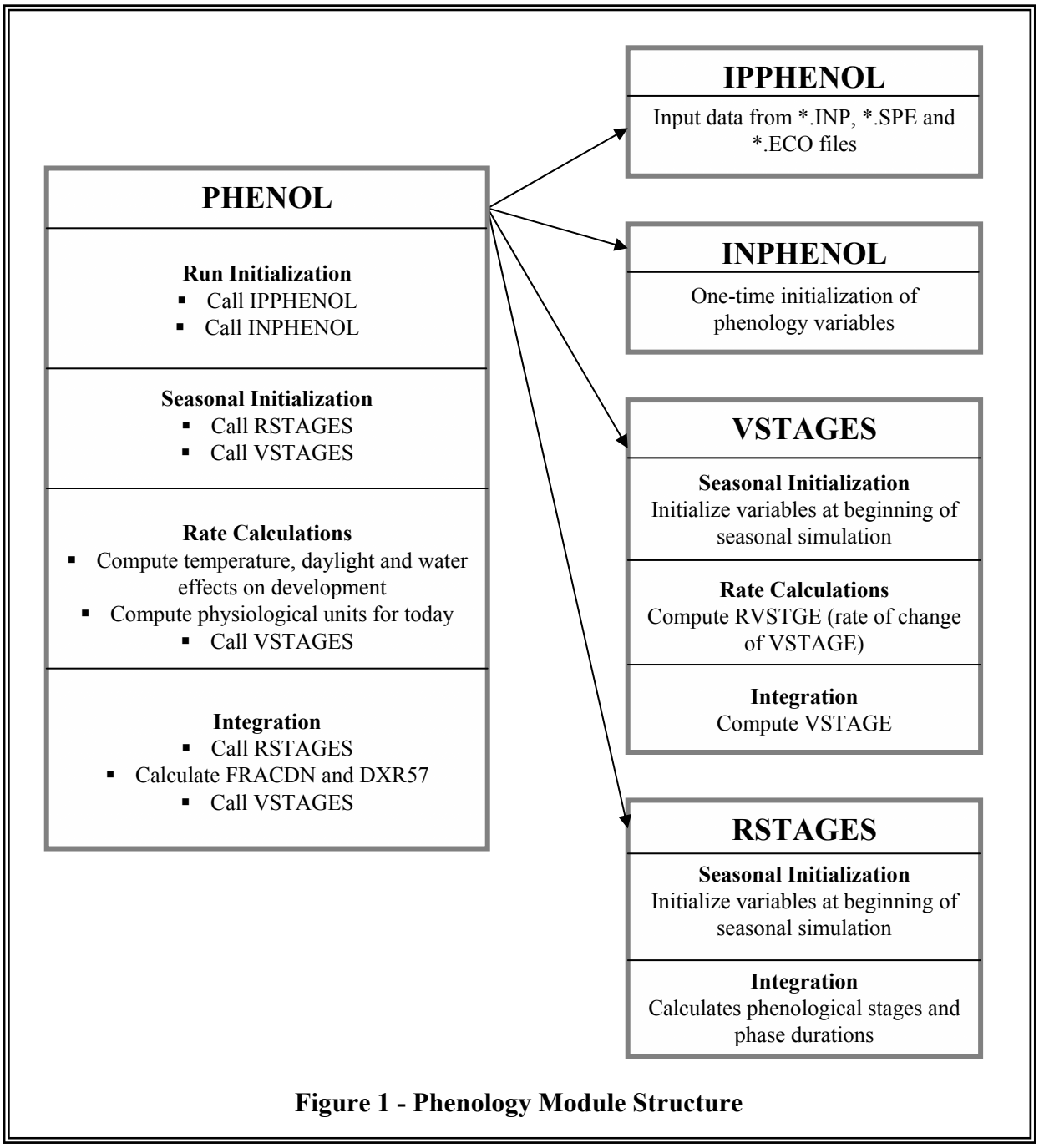


Figure 1 - Phenology Module Structure

Table 2 – Interface Variables for Phenology Module			
Variable Name		**Definition**	**Units**
Input Variables	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress	--
	TGRO(I)	Hourly canopy temperature	°C
	TMIN	Minimum daily temperature	°C
	TURFAC	Water stress factor for expansion (0 - 1)	--
	XPOD	Growth partitioning to pods which slows node appearance	fraction
	YRPLT	Planting date	YYYYDDD
Output Variables	DRPP	Photoperiod days which occur in a real day	p-t-d
	DTX	Thermal time that occurs in a real day based on vegetative development temperature function	t-d / day
	DXR57	Relative time between first seed (NR5) and physiological maturity (NR7)	fraction
	FRACDN	Relative time between flowering (NR1) and last leaf appearance (NDLEAF)	--
	MDATE	Harvest maturity date	YYYYDDD
	NDLEAF	Day when leaf expansion ceased	days
	NDSET	ormal time by which a pod load (full number) should be achieved with no water stress	days
	NR1	Day when 50% of plants have at least one flower	days
	NR2	Day when 50% of plants have one fruit (pod or peg)	days
	NR5	Day when 50% of plants have pods with beginning seeds	days
	NR7	Day when 50% of plants first have yellowing or maturing pods	days
	NVEG0	Day of emergence	days
	PHTHRS(I)	Threshold time that must accumulate in phase I for the next stage to occur	t-d or p-t-d
	RSTAGE	Number of RSTAGES that have occurred.	--
	RVSTGE	Rate of addition of nodes on main stem of plant	nodes/day
	STGDOY(I)	Day when stage I occurred	YYYYDDD
	TDUMX	Photo-thermal time that occurs in a real day based on early reproductive development temperature function	photo-thermal days / day
	TDUMX2	Photo-thermal time that occurs in a real day based on late reproductive development temperature function	photo-thermal days / day
	VSTAGE	Average number of nodes on main stem	nodes
	YREMRG	Day of emergence	YYYYDDD
	YRNR1	Day when 50% of plants have at least one flower	YYYYDDD
	YRNR2	Day when 50% of plants have one fruit (pod or peg)	YYYYDDD
	YRNR3	Day when 50% of plants have at least one beginning pod	YYYYDDD
	YRNR5	Day when 50% of plants have pods with beginning seeds	YYYYDDD
	YRNR7	Day when 50% of plants first have yellowing or maturing pods	YYYYDDD

SWFEM is a measure of how close soil water is to the drained upper limit, and is thus affected by the soil being too dry, (i.e., SW(I) is considerably less than DUL(I)) or too wet (i.e., SW(I) is near saturation). At the lower limit and saturated soil water contents (LL(I) and SAT(I)), this function becomes 0.0, meaning maximum stress if the soil is extremely dry or extremely wet. It is computed as a relative value, such that if SW(I) is equal to DUL(I) over the 10 cm depth, then SWFEM is equal to 1.0. The equations for each layer are:

```
IF (SW(I) .LE. DUL(I)) THEN
  SWFEM = SWFEM + DTRY*(MAX(SW(I) - LL(I),0.0))/(DUL(I) - LL(I))
ELSE
  SWFEM = SWFEM + DTRY*(MAX(SAT(I) - SW(I),0.0))/(SAT(I) -DUL(I))
ENDIF
```

where DTRY is the depth of layer I used in the calculation. After accumulating SWFEM over all layers in the top 10 cm of soil, the averages are calculated by dividing these variables by 10.

The SWFEM values are then used to determine FSW(1), the relative rate modifier for water stress for the first development phase, or the emergence phase:

```
FSW(1) = CURV('LIN',0.0,20.0,100.,1000.,SWFEM)
```

where 'LIN' specifies that a piece-wise linear curve type is to be used. The base soil water is 0.0 and the first optimal value is 20 percent, meaning that as long as SWFEM is between 20 and 100 percent, the development progress is not affected. This relationship was used to allow extreme soil water values (too dry or too wet) to delay or stop development progress, but under normal circumstances it would not affect development progress.

Average daily soil temperature over the top 10 cm of soil (TSDEP) is also computed to adjust the daily rate of development prior to emergence under non-optimal temperatures. The temperature curve for vegetative development, K = TSELC(1), is used for this emergence phase. The statement for the temperature function is:

```
FT(1) = CURV(CTMP(1),TB(K),TO1(K),TO2(K),TM(K),TSDEP)
```

where CTMP(1) refers to the type of curve used for phase 1, TB(K) is the base temperature for this function, TO1(K) and TO2(K) are the two optimal temperatures defining this function, and TM(K) is the temperature at which the function drops to 0.0. FT(1) thus depends on TSDEP and varies between 0.0 and 1.0.

If the soil water balance is not used in the simulation, i.e., if ISWWAT is not equal to 'Y', then FT(1) depends on air temperature since soil temperature is not computed, and FSW(1) is 1.0 to indicate no water stress effect on development.

Next, the development progress during a day is computed for all other phases, phases J = 2 through NPHS. The temperature sensitivity is set, K = TSELC(J). Then hourly values of the temperature function are computed, FTHR, and averaged over 24 hours to get FT(J). Before first flower (R1), the effect of daylength on development progress for phase J during a day, FUDAY(J), is computed by:

```
FUDAY(J) = CURV(DLTYP(J),1.0,CSDVAR,CLDVAR,THVAR,DAYL)
```

where DLTYP(J) is the type of function specified in the crop species file (See Section 3.0 - Table 3) for phase J (INL for inverse linear). CSDVAR is the daylength threshold below which developmental progress is a maximum for a cultivar (see Section 3.0 – Figure 2). CLDVAR is the value of daylength at which developmental progress is a minimum. The value of CLDVAR is computed from CSDVAR and a photoperiod sensitivity coefficient in the cultivar file, PPSEN, which is the slope of the curve shown in Figure 2 (Section 3.0). The THVAR parameter is specified in the ecotype file and is the minimum rate of developmental progress under long days at optimal temperature. Usually there is not enough information to estimate THVAR for a cultivar, and its value is set to 0.0. DAYL is daylength in hours.

Phases after flowering may have a different sensitivity to photoperiod than phases before flowering. For example, Grimm et al. (1993) found that soybean was more sensitive to daylength after flowering. Therefore a variable in the Ecotype file is used to specify increased (or decreased) photoperiod sensitivity. For those phases, the effect of daylength on development is computed by:

```
FUDAY(J) = CURV(DLTYP(J),1.0,CSDVRR,CLDVRR,THVAR,DAYL)
```

where CSDVRR and CLDVRR are the adjusted critical daylength thresholds after flowering.

Water and N stress effects on development are computed for each phase J by:

```
FSW(J)   = 1. + (1. - SWFAC)   * WSENP(J)
FNSTR(J) = 1. + (1. - NSTRES) * NSENP(J)
```

where WSENP(J) and NSENP(J) scale the effects of these stresses. SWFAC is the effect of water stress on photosynthesis, with 1.0 meaning no stress and 0.0 meaning maximum stress or no photosynthesis. When SWFAC is 0.0 (maximum stress), a value of WSENP(J) of -1.0 would slow development progress to 0.0, whereas a value of WSENP(J) = +1.0 would speed up development progress by a factor of 2.0. The values for N are similar. A value for the effects of phosphorus (FPSTR(J)) was also defined, but set to 1.0 because the P balance is not yet implemented.

If transplants are used (PLME = 'T') and today is the planting date (YRPLT = YRDOY), then the temperature function is defined based on the temperature sensitivity for phase 2, TSELC(2), and ATEMP, the temperature of the transplant environment:

```
FT(2) = CURV(CTMP(2),TB(K),TO1(K),TO2(K),TM(K),ATEMP)
```

The physiological accumulation for phase 2, PHZACC(2), is computed as the product of the temperature function, FT(2), by SDAGE, the age of the transplant.

```
PHZACC(2) = FT(2) * SDAGE
```

SDAGE and ATEMP are user supplied variables read from the input file in the initialization section.

A factor which accounts for the effects of minimum temperature on the rate of development from emergence to flowering (ZMODTE) is then calculated and applied to the temperature functions for phases 4 and 5. (Grimm et al. 1993, 1994.; Piper 1995.)

Variables are then computed that combine temperature and daylength effects for use in different parts of the model. DTX is the thermal days for vegetative development occurring during the day, based on air temperature; TNTFAC is thermal days for early reproductive development and TNTFC2 is thermal days for late reproductive development that occur during the current day. DRPP is the relative effect of daylength on development, without considering temperature. TDUMX and TDUMX2 are photothermal days for early and late reproductive developmental phases, respectively, and include both photoperiod and temperature effects.

VSTAGES is called to calculate the rate of change of V-stages for today (RVSTGE).

2.1.4 Integration Calculations

In the integration section of the module, RSTAGES is called to update developmental progress for each active phase and to determine if new stages occur. Several special accumulators are updated. FRACDN is the relative time from flowering to last leaf, and DXR57 is the relative developmental progress from first seed to physiological maturity.

VSTAGES is called to compute today's value of VSTAGE, the number of leaf nodes on the main stem.

2.2 Subroutine IPPHENOL

Subroutine IPPHENOL is called by PHENOL once during run initialization. This

subroutine opens FILEIO (IBSNATxx.INP), FILEC (*.SPE), and FILEE (*.ECO). Variables read from the FILEIO include:

- Simulation control variables - ISIMI (start of simulation code) and ISWWAT (water balance simulation code).
- Crop identification code (CROP).
- Planting details – PLME (planting method), SDEPTH (planting depth), SDAGE (transplant age), and ATEMP (temperature of transplant environment).
- Soil characteristics information – DS(L) (soil layer depth), LL(L) (water content at lower limit), DUL(L) (water content at upper limit), SAT(L) (water content at saturated conditions).
- Cultivar information – ECONO (ecotype code), CSDVAR (critical daylength above which development rate remains at minimum value prior to flowering), PPSEN (sensitivity to photoperiod), PH2T5 (time from end of juvenile phase to first flowering under optimal conditions), PHTHRS(I) (threshold time that must accumulate in phase I for next stage to occur).

The crop file is opened to read:

- EVMODC (modifier of rate of vegetative node appearance for the first few nodes),
- Temperature function parameters (TB, TO1, TO2, and TM) that define daily temperature distribution, and
- Phenological parameters – NPRIOR(I) (phase of growth at which phase I accumulator can start), DLTYP(I) (type of curve used for daylength function for phase I), CTMP(I) (type of curve used for temperature function for phase I), TSELC(I) (number of temperature curve to be used for phase I development rate), WSENP(I) (sensitivity of phase I to water stress), NSENP(I) (sensitivity of phase I to nitrogen stress), and PSENP(I) (sensitivity of phase I to phosphorus stress).

The species file is opened to read ecotype parameters: ECOTYP (ecotype code), ECONAM (ecotype name), IVRGRP and IVRTEM (maturity and temperature classifications), THVAR (minimum relative rate of reproductive development), PHTHRS (threshold timing for phases), PM06 (proportion of time between first seed and first pod), PM09 (proportion of tie between first seed and physiological maturity), TRIFOL (maximum rate of v-stage formation), R1PPO (sensitivity to photoperiod), OPTBI (temperature below which growth rate is slowed from emergence to flowering), SLOBI (sensitivity of growth rate to minimum temperatures from emergence to flowering).

2.3 Subroutine INPHENOL

INPHENOL is called once by PHENOL during the run initialization phase. Variables: MNEMV1 (minimum days from emergence to vegetative growth stage 1), MNFLLL (minimum days from start of flowering to last leaf appearance) and MNFLHM (minimum days from flowering to harvest maturity) are computed in this section. Values are assigned to STNAME, the names of reproductive phases based on Crop type.

2.4 Subroutine VSTAGES

VSTAGES is called three times by PHENOL, once each for seasonal initialization, rate

calculations and integration.

2.4.1 Seasonal Initialization

Variables VSTAGE and RVSTGE are initialized.

2.4.2 Rate Calculations

RVSTGE, the rate of change of VSTAGE (the number of nodes on the plant), is computed based on yesterday's values of VSTGED, the duration of time for new node development and on VSTAGE and VSTAGP, the number of nodes on the plant for the last two days.

2.4.3 Integration

VSTAGE is updated if emergence (NVEG0) has occurred and if the final leaf (NDVST) on the main stem has not yet occurred. Between NVEG0 and NVEG1, VSTAGE is simply equal to the cumulative progress toward NVEG1, computed as PHZACC(2) divided by the minimum time from NVEG0 to NVEG1 (MNEMV1). Then, it is assumed that the plants can produce the first few nodes (EVMODC) at a faster rate than later nodes, although this feature can be eliminated by setting EVMODC = 0 in the crop species file. EVMOD varies between 1.0 and 2.0, meaning that development of the first five nodes could be as much as twice as fast as later nodes. This variable is used primarily to simulate peanut growth. VSTAGE is computed by:

```
VSTAGE = VSTAGE + DTX * TRIFOL * EVMOD * TURFAC * (1.0 - XPOD)
```

where TRIFOL is the rate of appearance of leaves on the main stem in nodes per thermal day, TURFAC modifies this rate under water stress conditions, and XPOD is the growth partitioning to pods on the current day which slows node appearance as more growth is allocated to reproductive tissue. Finally, the rate of V-stage addition is computed for use in the canopy height and width calculations.

2.5 Subroutine RSTAGES

2.5.1 Seasonal Initialization

RSTAGES is called to initialize variables that are calculated within this routine once at the start of each season of simulation.

2.5.2 Integration

This subroutine determines which developmental phases are active, computes the developmental progress for those phases, and causes stages to occur when the developmental progress of each active phase reaches its threshold. The subroutine is called once during the integration portion of the PHENOL code. RSTAGES is organized into sections that compute all necessary information for each phase.

Each section is basically the same. When a stage occurs, it may only require a portion of the current day, so that developmental progress toward the next stage may also begin on the same day. The variable REM(J) is the proportion of the

current day that is left after stage J occurs, and it is used to initiate progress in phases that begin when stage J occurs. Each day this variable is initialized to 1.0 for each stage.

The first section of RSTAGES sets the initial accumulators for each phase for transplants if this is planting day (YRPLT = YRDOY). The physiological accumulator calculated in the rate section for transplants is allocated to phases 2 through 6.

The next section of RSTAGES deals with the planting to emergence phase. The FORTRAN statements for this entire section are given below.

```
IF (NVEG0 .GE. NVALP0) THEN
    PHTEM = PHTHRS(1) + SDEPTH * 0.6
    PROG(1) = FT(1) * FUDAY(1) * MIN(FSW(1),FNSTR(1),FPSTR(1))
    PHZACC(1) = PHZACC(1) + PROG(1)
    IF ((PHZACC(1) .GE. PHTEM) .OR. (ISIMI .EQ. 'E')) THEN
          NVEG0 = DAS
          NVALPH(2) = NVEG0
          YREMRG    = YRDOY
          STGDOY(1) = YRDOY
          IF (ISIMI .NE. 'E') THEN
                REM(2) = (PHZACC(1) - PHTEM)/(PROG(1) + 0.00001)
          ENDIF
    ENDIF
ENDIF
```

First, a check is made to determine if the ending stage of the phase has already occurred, emergence (NVEG0) in this case. NVEG0 is the number of days from the beginning day of simulation that emergence occurs and NVALP0 is a large number used to initialize stages. At the start of the season, NVEG0 is set equal to NVALP0 to indicate that the time of this stage has not occurred. When it does occur and NVEG0 is set to DAS (days after start of simulation), then it will be smaller than NVALP0 and this section of code will not be used again.

The second statement in this section computes the threshold for the emergence phase (PHTEM, in thermal days) as explained above, taking into account the planting depth. Data from Stucky (1976) for three planting depths and three temperatures were used to estimate the total duration of germination plus emergence to be 0.6 * SDEPTH + PHTHRS(1) thermal days, where SDEPTH is planting depth (cm). This step is not used for other phases.

Then, the overall developmental progress toward emergence is computed as PROG(1), by multiplying the temperature factor (FT(1)) by the daylength factor (FUDAY(1)) and by the minimum of the stress factors for water, N, and P. This relationship was used to allow the most limiting stress to modify the basic progress rate as determined by temperature and daylength. Note, however, that the actual effect of any of these stresses is completely defined in the input file (the

crop species file such as SBGRO980.SPE). Sensitivities to water (WSENP(J)), N (NSENP(J)), and P (PSENP(J)) can be set to 0.0 in this file to prevent any effect of these stresses on any developmental phase. Values for WSENP(1) can be negative, showing that water stress can slow development for phase 1. The code was written in this way to allow complete flexibility for tailoring the phenology model for different grain legume crops using input files as opposed to code changes.

The next statement in the above section of code computes the cumulative developmental progress in this phase, PHZACC(1) by adding today's developmental progress. Then, the next statement checks to see if the threshold for emergence, PHTEM, has been reached or exceeded today. If the variable ISIMI is set to E, then the simulation begins on emergence day, i.e. emergence is forced to occur on the first day of simulation. If not, control is passed from this section and it is repeated on the next day. If, however, PHZACC(1) has exceeded PHTEM (or ISIMI = 'E'), then emergence occurs today, and the next four statements set stage definition variables. NVEG0 is set to DAS, the current day of simulation and STGDOY(1) is set to YRDOY, which is a variable that defines the year and day of year when emergence occurred.

The variable RSTAGE counts the number of stages that have actually occurred during the season.

The NVALPH(2) and YREMRG variables specify the day that stage 2 (emergence) occurred for use in determining which phases are active. The REM(2) is the remainder of the day that can be used for developmental progress by any phase that begins when stage 2, emergence, occurs. It divides the remaining thermal time that has accumulated above the threshold by the progress during the day. This number will always be a fraction of a day.

The sections for the other stages are the same as this with two exceptions. As stated above, the threshold for emergence is computed, whereas for all other phases the threshold is simply PHTHRS(J). The second exception involves an IF statement to determine whether the reference stage for the phase has occurred. For example, the following statement is used to determine whether the flower induction stage has occurred:

```
IF(DAS .GE. NVALPH(NPRIOR(5))) .AND. (NR1 .GE. NVALP0) THEN
```

If it has not occurred, the section for determining progress to first flower would be skipped. The variable NVALPH(J) is the day that stage J occurs, and NPRIOR(I) is the reference stage for phase I. Phase 5 is the phase for progress from flower induction to first flower, and NPRIOR(5) is set to stage 5 (flower induction) in the species file (see Section 3.0 – Table 3). In this example therefore, DAS is compared with NVALPH(5), which is the day that stage 5 (floral induction) occurred. If DAS is smaller than NVALPH(5), then floral

induction has not yet occurred, and it is not time to start accumulating developmental progress toward flowering. If DAS is greater than NVALPH(5), then floral induction has occurred and a check is made to see if flowering has occurred before accumulating progress in this section. This allows one to define the reference stages in a very general way for different crops.

On each day of simulation, RSTAGES considers whether each phase is active in the ways described above. There may be more than one phase active at the same time. For example, after flowering progress toward the first seed occurs as does progress toward the last mainstem node appearance. These phases, stages, reference stages, types of accumulators and other crop specific information have been completely specified for three crops, soybean, peanut, and dry bean. Other legume crops could be implemented using the code by careful definition of these parameters for those crops.

3.0 Theory and Equations

The basic approach used in the Phenology Module of the CROPGRO model is to assume that a developmental stage occurs when developmental progress in a phase reaches a threshold level specified for that stage. Progress is estimated by integrating, or summing, a developmental rate over time. The rate of progress, (dP/dt) toward a given stage is calculated as the product of a maximal rate of development (MR), a rate modifier varying with temperature (f(T)) and, if appropriate, a modifier for photoperiod (g(P)):

$$dP/dt = MR * f(T) * g(P)$$

By defining MR = 1.0, and restricting f(T) and g(P) to a range of 0.0 to 1.0, the thresholds take on values that are equal to the minimum number of calendar days it would take to reach a given stage. Since the minimum number of days is obtained only if the rate modifiers are equal to 1.0, the thresholds can be thought of as the time required under optimal conditions for development. Numerically, this approach is identical to accumulating "heat units" or "photothermal time". We feel that it is physiologically more accurate to use and perhaps easier to conceptualize "developmental progress", such as differentiation of flower buds or formation of seed, than to refer to "accumulation of heat units". In the model, we refer to development progress as thermal days (t-d) when progress does not depend on daylength and as photothermal days (p-t-d) when developmental progress depends on daylength as well as temperature.

Phases in the models are described in Table 3. Each of these 13 phases has an accumulator that starts computing accumulated photothermal or thermal days at a reference stage, defined in the table. A reference stage is the stage that must occur before the accumulator for a particular phase starts. A phase is said to be active if its reference stage has occurred and the accumulator for that phase is increasing. A phase is complete when a threshold for the accumulator is reached, at which time the stage defining the end of the phase occurs. After this time, the accumulator for that phase is no longer active. The daily developmental progress is calculated in the variable PROG(I) for phase I, and the accumulator for phase I is PHZACC(I). The reference stage for phase I is stored in the variable NPRIOR(I), and the thresholds for the completion of phases are stored in the variable PHTHRS(I) for phase I.

Table 3 - Phases of Crop Development

Table for defining the phases for each crop model implemented in CROPGRO. Reference Stage numbers are the stages at the beginning of each phase. The Photoperiod Curve defines the type of response used for modifying the rate of development with daylength (NON means that the phase is not sensitive to daylength). Temperature Curve specifies the type of function used for temperature response for this crop. Temperature Sensitivity specifies which temperature curve to use for a phase (1=vegetative, 2=early reproductive, and 3=late reproductive functions). Water Sensitivity refers to the sensitivity of the phase development rate to water stress (if less than 0.0, delay in development occurs). The Variable Names used in the code are listed over each column.

Var. Name/ Phase No.	NPRIOR Ref Stage	DLTYP Photo Curve	CTMP Temp Curve	TSELC Temp Sens	WSENP Water Sens	Phase Description
1	1	NON	LIN	1	-0.20	Planting (STG 1) to Emergence (STG 2) Phase
2	2	NON	LIN	1	-0.20	Emergence (STG 2) to V1 (STG 3) Phase
3	2	NON	LIN	1	-0.40	Emergence (STG 2) to end Juvenile (STG 4) Phase
4	4	INL	LIN	2	-0.40	End Juvenile (STG 4) to Floral Ind. (STG 5) Phase
5	5	INL	LIN	2	-0.40	Floral Ind. (STG 5) to 1st Flower (STG 6) Phase
6	6	INL	LIN	2	-0.40	1st Flower (STG 6) to 1st Peg (STG 7) Phase
7	6	INL	LIN	2	-0.40	1st Flower (STG 6) to 1st Pod (STG 8) Phase
8	6	INL	LIN	2	-0.40	1st Flower (STG 6) to 1st Seed (STG 9) Phase
9	9	INL	LIN	3	0.20	1st Seed (STG 9) to Last Seed (STG 10) Phase
10	9	INL	LIN	3	0.20	1st Seed (STG 9) Phys. Maturity (STG 11) Phase
11	11	NON	NON	1	0.00	Phys. Mat (STG 11) to Harvest Mat (STG 12) Phase
12	6	INL	LIN	2	-0.60	1st Flwr (STG 6) to Last VSTAGE (STG 13) Phase
13	6	INL	LIN	2	-0.90	1st Flower (STG 6) to Last Leaf (STG 14) Phase

Each of the 13 phases also has temperature (FT) and daylength (FUDAY) functions that are depicted in the above equation as f(T) and g(P), respectively. This structure allows complete flexibility for defining which phases are dependent on temperature alone, or on temperature and daylength. The information in Table 3 is defined for each crop in a crop species parameter file designated by the extension PE. For example, the file SBGRO980.SPE contains the crop species parameters for soybean, including the phenology parameters shown in Table 3. Each day, the FT and FUDAY functions are computed in PHENOL for each phase. The overall rate of progress for each phase is computed using the variable PROG(I), only for those phases that are actively developing, where PROG(I) is the daily progress toward a stage. The cumulative developmental progress in an active phase toward the next stage is computed in the variable PHZACC(I) for phase I. As an example, phase 10 in Table 3 is for the interval from first seed to physiological maturity. The reference stage is stage 9 (first seed or NR5). Phase (10) is sensitive to photoperiod using an inverse linear function (INL), it is sensitive to temperature using a linear function (LIN), it uses the temperature curve defined for late reproductive development (TSELC = 3), and it has a +0.2 sensitivity to water stress. The positive sign means that development will be hastened under water stress for this phase, as much as 20% for severe stresses.

Grimm et al. (1993, 1994) found that rates of progress toward all reproductive stages in soybean were linearly related to night length. In this model, rate of progress toward any stage is related to daylength, thus the linear function used by Grimm et al. was transformed into what is referred to as an inverse linear function. This function varies from 1.0 for short days to a minimum value

as day length increases. The minimum value could be 0.0 or it could be some positive value to allow some progress to occur even under very long days. An example of this function is presented in Figure 2.

The information for the temperature function is also contained in the species parameter file (See Table 4 for values of the temperature functions used for soybean in the model). These temperature values are assumed to be identical for all cultivars within a species. Figure 3 shows the effect of temperature on development rate for soybean, for each of the three periods of growth. For each period of growth, four temperature values define piece-wise linear functions, using a base temperature (TB), two values that define the range of temperatures for most rapid progress (TO1 and TO2), and a maximum temperature above which development ceases (TM). In Table 4, the VEGETATIVE temperature values define the function for vegetative development, affecting emergence and juvenile phases as well as the rate of appearance of leaf nodes on the main stem. The EARLY REP values define the sensitivity to temperature during early reproductive growth, and are used to affect developmental progress toward flowering and first seed. The LATE REP values are for late reproductive development, or the time between first seed and physiological maturity. The temperature function for soybean was specified to be LIN in the species file (Table 3), which causes the function to be piece-wise linear. However, the four temperature values in Table 4 for each growth period could be used for quadratic or sinusoidal functions, using the CURV function, if specified in Table 3 for each phase.

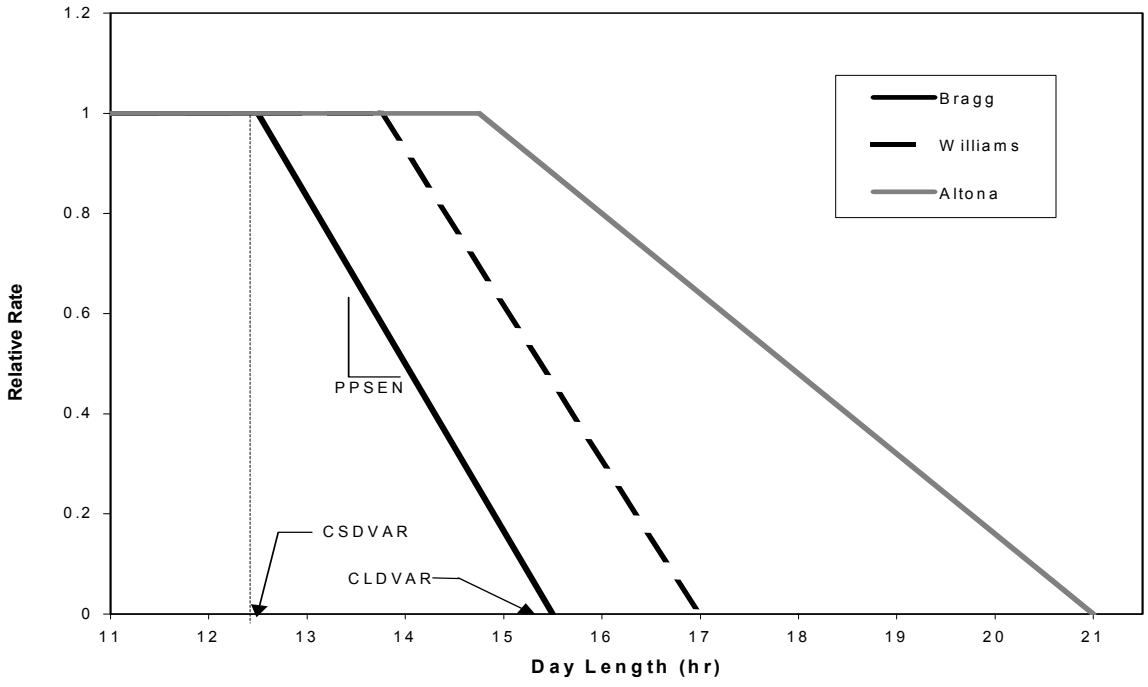


Figure 2 - Relative sensitivity of three soybean cultivars to daylength.
CSDVAR and CLDVAR are the values of daylength at which developmental progress is insensitive, and minimum, respectively. PPSEN is the photoperiod sensitivity coefficient.

Table 4 - Temperature Function Values					
Values of temperature used to define the functions for soybean for modifying development rates by temperatures; for vegetative phases (VEGD), early reproductive phases (REPDA) and late reproductive phases (REPDB). These values are used in the CURV function, which computes a function value between 0.0 and 1.0. These values are input to the model from the species parameter file (SBGRO980.SPE, PNGRO980.SPE, or BNGRO980.SPE). Values are based on Grimm et al. (1993, 1994).					
	Temperature Function Values				**Sensitivity**
PHASES	**TB**	**TO1**	**TO2**	**TM**	**TSELC**
VEGETATIVE	7.0	28.0	35.0	45.0	1
EARLY REP	6.0	26.0	30.0	45.0	2
LATE REP	-48.0	26.0	34.0	45.0	3

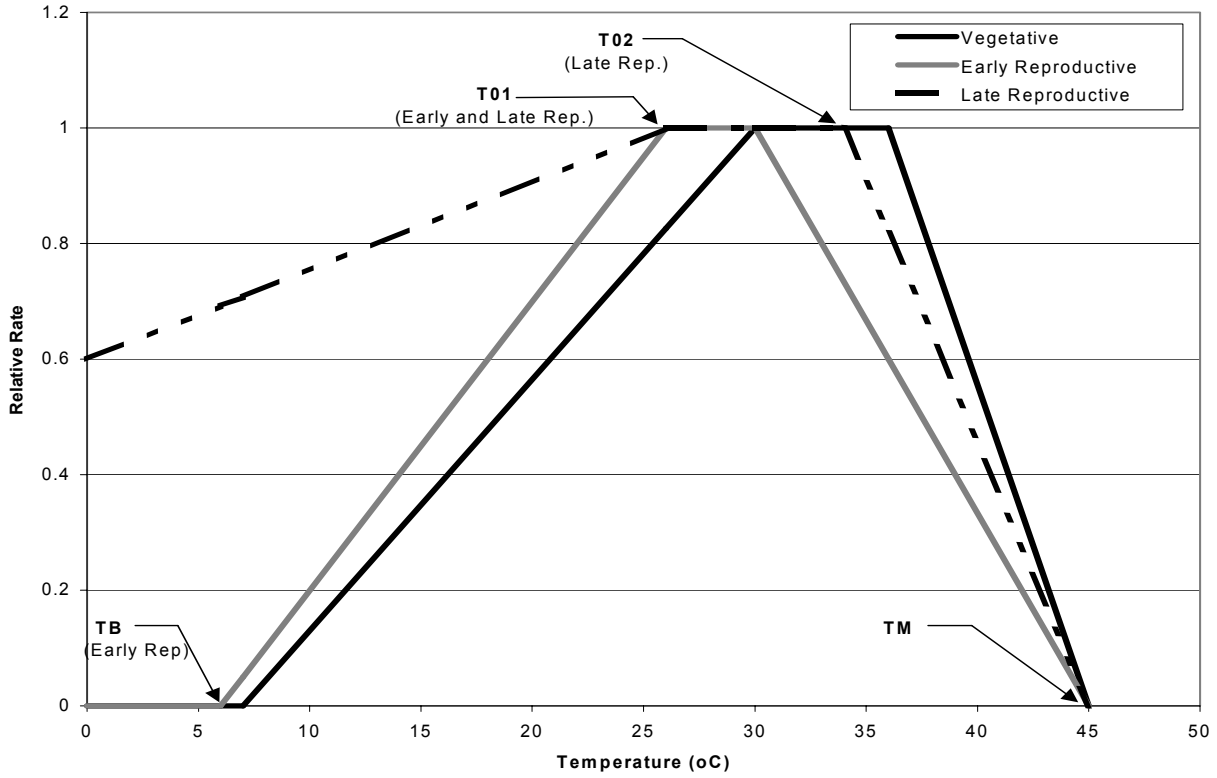


Figure 3 - Relative effect of temperature on developmental rate of soybean, for each of the three periods of growth

There is considerable variability among cultivars for the thresholds for different phases and also for photoperiod sensitivities. Similar to the previous SOYGRO, PNUTGRO, and BEANGRO models, this generic model has a cultivar parameter file for each crop type. It also has an ecotype file. Cultivar parameters are those that are expected to vary the most among cultivars and would need to be estimated for new cultivars. The parameters in the ecotype file are intended for use by many cultivars that have similar growth habits, and users would not be expected to vary these under normal circumstances. Some of the parameters in the ecotype file were in the genetics file for the earlier crop models, but were moved here to reduce the number of parameters that users would routinely have to work with.

Table 5 presents the phenology parameters that vary with cultivar. Most of the cultivar parameters that relate to phenology are thresholds for different phases to occur. For example, PH2T5, PHTHRS(7), PHTHRS(8), and PHTHRS(10) define the durations of phases from emergence to first flower, first flower to first pod, first flower to first seed, and first seed to physiological maturity, respectively. Note that PH2T5 defines the minimum amount of time from emergence to first flower, both of which are relatively easily observed in the field. Internally, the model computes the progress of plants from emergence to V1 stage (phase 2), emergence to end of juvenile phase (phase 3), floral induction (phase 4), and floral expression (phase 5). Coefficients for defining how to break the emergence to flower phase into these other phases are obtained in the ecotype files for each species, and normally would not be modified when estimating genetic coefficients for a cultivar within a species.

In addition to the thresholds, photoperiod sensitivity information is contained in the cultivar file. Figure 2 shows the meaning of CSDVAR and PPSEN schematically for several cultivars of soybean, using the inverse linear function of CURV. These values were obtained by Grimm et al. (1993) by analyzing time to flower by cultivars planted across a wide range of latitudes.

The rate of appearance of leaves on the main stem is defined by TRIFOL in the ecotype file. In addition, this file defines the other phase durations that were not in the cultivar file, those that are usually constant across cultivars.

The implementation of the phenology model in CROPGRO is very general. It was designed so that it could be easily adapted to simulate other grain legumes, or even other crops. Each phase could be dependent on any combination of temperature, daylength, water stress and other stresses. In addition, the order of the phases could be defined completely from the input files. This makes it easy to modify the functions, the values of the parameters, and the sensitivities to stresses by phase for each crop implemented by this code. However, the dry weight growth portion of the CROPGRO model depends on the developmental stage in several places. Therefore, users should not modify the numbers of phases in the species parameter file. Otherwise, the types of functions used for each temperature, daylength and stress sensitivities may be modified to adapt the code for other crops. In addition, the cultivar and ecotype files can be modified as needed for the current three crops or for new crops that are implemented.

Table 5 - Cultivar and Ecotype parameters

Cultivar and Ecotype parameters that relate to phenological development for the crops modeled with CROPGRO code.

VARIABLE	DEFINITION	UNITS
CSDVAR	Critical day length below which reproductive development proceeds unaffected by daylength, and above which development rate is reduced in proportion to hours above CSDVAR	h
LNGSH	Time required for shell growth	p-t-d
PHTHRS(1)	Time required for seed germination in thermal days	t-d
PHTHRS(2)	Time required from emergence to first true leaf (V1)	t-d
PHTHRS(3)	Time required from emergence to end of juvenile phase	t-d
PHTHRS(4)	Time required for floral induction, photothermal days, equal to the minimum number of days for floral induction under optimal temp and daylengths	p-t-d
PHTHRS(6)[+]	Time from first flower to first peg greater than 0.5 cm, in photothermal days, equal to the minimum duration under optimal temperature and photoperiod	p-t-d
PHTHRS(8)	Time from first flower to first seed in photothermal days, which is equal to the minimum duration under optimal temperature and photoperiod	p-t-d
PHTHRS(10)	Time from first seed to physiological maturity in photothermal days, which is equal to the minimum duration under optimal temperature and photoperiod	p-t-d
PHTHRS(11)	Time between physiological and harvest maturity dates	d
PHTHRS(12)	Time from first flower to last leaf on main stem	p-t-d
PHTHRS(13)	Time from first flower to end of leaf growth in photothermal days, equal to the minimum duration under optimal temperature and photoperiod	p-t-d
PH2T5	Time from end of juvenile phase to first flower in photothermal days, which is equal to the minimum duration under optimal temperature and photoperiod	p-t-d
PM06	Proportion of time between first peg and first seed for first pod. If PM06 is 0.0, first peg and first pod occur at the same time	-
PM09	Proportion of time between first seed and physiological maturity that the last seed can be formed	-
PPSEN	Slope of the relative rate of development for day lengths above CSDVAR, or sensitivity to photoperiod	1/h
R1PPO	Increased (+) or decreased (-) sensitivity to photoperiod, expressed as change in critical daylength	h
THVAR	Minimum rate of reproductive development under long days and optimal temperature	-
TRIFOL	Rate of appearance of leaves on the mainstem, leaves per thermal day	1/(t-d)

[+]For soybean and dry bean, this is also the time from first flower to first pod greater than 0.5 cm in length, and PM06 in the Ecotype file is 0.0 for this case.

4.0 References

Acock B., and Reynolds J.F., 1989. The rationale for adopting a modular generic structure for crop simulators. Acta Horticulturae 248:391-396.

Boote, K. J. 1982. Growth stages of peanut (Arachis hypogaea L.). Peanut Sci. 9:35-40.

Boote, K. J., J. W. Jones, G. Hoogenboom, G. G. Wilkerson, and S. S. Jagtap. 1989. PNUTGRO V1.02. Peanut crop growth simulation model: User's guide. Florida Exp. Sta. Journal No. 8420. Agronomy Department, and Agricultural Engineering Department, University of Florida, Gainesville, FL 32611.

Fehr, W. R., and C. E. Caviness. 1977. Stages of soybean development. Iowa State University Cooperative Extension Service Special Report 80.

Grimm, S. S., J. W. Jones, K. J. Boote, and J. D. Hesketh. 1993. Parameter estimation for predicting flowering dates of soybean cultivars. Crop Science 33: 137-144.

Grimm, S. S., J. W. Jones, K. J. Boote, and D. C. Herzog. 1994. Modeling the occurrence of reproductive stages after flowering for four soybean cultivars. Agronomy Journal 86: 31-38.

Hoogenboom, G., J. W. White, J. W. Jones, and K. J. Boote. 1991. BEANGRO V1301. Dry bean crop growth simulation model: User's guide. Fla. Exp. Station Journal No. N-00379. Agricultural Engineering Department, University of Florida, Gainesville, FL 32611.

Jones, J. W., K. J. Boote, G. Hoogenboom, S. S. Jagtap, and G. G. Wilkerson. 1989. SOYGRO V5.42. Soybean crop simulation model: User's guide. Fla. Exp. Sta. Journal No. 8304. Agricultural Engineering Department, University of Florida, Gainesville, FL 32611.

Jones, J. W., K. J. Boote, S. S. Jagtap, and J. W. Mishoe. 1991. Soybean development. In: Modeling plant and soil systems. Agronomy Monograph No. 31. ASA-CSSA-SSSA, Madison, WI.

Kenig A., and J.W.Jones, 1997. Model structure for dynamic crop-greenhouse simulations. In: Seginer I., J.W. Jones, P. Gutman, and C.E. Vallejos (eds), Optimal environmental control for indeterminate greenhouse crops. Final Report, BARD Research Project IS-1995-91RC, Agricultural Engineering Department, Technion, Haifa, Israel. Chap. II-4.

Kraalingen, D.W.G. van, 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis Report no. 1. AB/DLO, PE, Wageningen.

Laing, D.R.L., P.G. Jones and J.H.C. Davis. 1984. Common Bean (Phaseolus vulgaris L.). p. 305-351. In Goldsworthy, P.R. and N.M. Fisher. The physiology of tropical field crops.

Penning de Vries, F. W. T., D. M. Jansen, H. F. M. ten Berge, and A. Bakema. 1989. Simulation of ecophysiological processes of several annual crops. Pudoc, Wageningen, the Netherlands.

Piper; E.L., M.A.Smit, K.J. Boote, and J.W. Jones. 1996. The role of daily minimum temperature in modulating the development rate to flowering in soybean. Field Crops Research 47 (1996) 211-220.

Reynolds, J.F. and B. Acock. 1997. Modularity and genericness in plant and ecosystem models. In Ecological Modeling. 94 (1997) 7-16.

Schoonhoven and Pastor-Corrales. 1987. Standard system for the evaluation of bean germplasm. CIAT, Cali, Colombia.

Stucky, D. J. 1976. Effect of planting depth, temperature, and cultivars on emergence and yield of double cropped soybeans. Agronomy Journal 68: 291-294.

White, J., and Izquierdo. 1991. Physiology of yield potential and stress tolerance. p. 287-382. In A. van Schoonhoven and O. Voysest (eds.). Common beans: research for crop improvement.

Volume4
Chapter 3

Other Crop Modules

1. Ceres-Maize
2. IXIM - Maize
3. CSCER Wheat and Barley
4. CSCRP Wheat, Barley, Cassava
5. Ceres-Millet
6. Ceres-Sorghum
7. Ceres-Rice
8. Substor-Potato
9. Canegro – Sugarcane
10. Casupro – Sugarcane
11. ARGRO – Taro, tanier
12. SUOIL – sunflower

UNDER CONSTRUCTION

A Simulation Model for Potato Growth and Development: SUBSTOR-Potato Version 2.0

Timothy S. Griffin
Cooperative Extension and Department of Plant, Soil and Environmental Sciences
University of Maine, Orono ME 04469

Bradley S. Johnson
Department of Crop and Soil Science
Michigan State University
East Lansing, MI 48824

Joe T. Ritchie
Department of Crop and Soil Science
Michigan State University
East Lansing, MI 48824

IBSNAT Research Report Series 02

Published for distribution by the IBSNAT Project:

Michigan State University
Department of Crop and Soil Sciences
East Lansing, Michigan 48824-1325

and

University of Hawaii
College of Tropical Agriculture & Human Resources
Department of Agronomy and Soil Science
Honolulu, Hawaii 96822

Support for the IBSNAT Project is provided through a cooperative agreement (DAN-4054-A-00-7081-00) between the U.S. Agency for International Development and the University of Hawaii. All reported opinions, conclusions, and recommendations are those of the authors and not those of the funding agency or the United States government.

Correct citation: IBSNAT. 1993. Research Report Series 02. A Simulation Model for Potato Growth and Development: SUBSTOR - Potato Version 2.0. Dept. of Agronomy and Soil Science, College of Tropical Agriculture and Human Resources, Univ. of Hawaii, Honolulu, HI.

IBSNAT Research Report Series 02—05/93 (500)

Introduction

The accumulation and partitioning of biomass and the phenological development of a potato (*Solanum tuberosum* L.) crop are influenced by many factors. Individually or in combination, the most important are the environmental variables temperature (Snyder and Ewing, 1989; Prange et al., 1990), photoperiod or daylength (Ewing, 1981; Wheeler and Tibbetts, 1986) and intercepted radiation (MacKerron and Waister, 1985). The simulation of potato growth across diverse environments and different cultivars must take each of these variables into account. Numerous efforts have been made to simulate potato growth, ranging from simple regression or correlative models based on temperature (e.g., Iritani, 1963; Manrique and Bartholomew, 1991) to the mechanistic, organ-level model of Ng and Loomis (1984). A common feature of these models is that they are location- (and usually cultivar-) specific. By limiting the application of the model to a specific geographical region, the number of environmental inputs required to run the model is effectively reduced. Temperature response functions (i.e., thermal time or growing degree days (GDD)) are most commonly used to simulate both growth and development. Manrique and Bartholomew (1991) demonstrated that changes in biomass partitioning of cvs. Kennebec and Desiree in Hawaii were strongly related to a single environmental factor; minimum daily temperature. The model of Ingram and McCloud (1984), for cv. Sebago in Florida, depends almost entirely on temperature response to determine potential growth rate, but is rather unique in using different functions for the growth of roots, tops, and tubers. Hartz and Moore (1978), and Sands et al. (1979) use temperature as the primary factor in estimating potential biomass accumulation and adjust these estimates based on intercepted radiation. MacKerron and Waister (1985) were able to accurately predict the growth of cv. Maris Piper in Scotland using total intercepted radiation, not temperature, as the primary factor, presumably because daily temperatures in this region are near the optimum for potato growth (i.e., 17 to 20°C) during tuber initiation and bulking.

The effect of photoperiod is ignored in most potato models, although tuber initiation is sensitive to photoperiod and wide differences in photoperiod response between cultivars have been demonstrated (e.g., Ben Khedher and Ewing, 1985; Snyder and Ewing 1989). There are two potential reasons for this omission. First, the aforementioned geographic and cultivar specificity makes it unnecessary to quantify the effects of different photoperiods (vs. predictable seasonal shifts in photoperiod) on each cultivar. And second, unlike temperature response where cardinal values are reasonably well documented, the quantification of photoperiod effects has been less rigorous. For most cultivars, a threshold photoperiod beyond which tuber initiation is adversely affected has not been firmly established.

This paper presents the development and performance of a new potato model, SUBSTOR-Potato Version 2.0, which is intended to be used over a wide geographical range and for different cultivars. SUBSTOR-Potato was developed as a CERES-type crop model, and thus uses capacity type models of soil water and soil N dynamics that are used in other CERES-type models (e.g., Jones and Kiniry, 1986). The effects of soil water and plant N deficit are simulated in SUBSTOR-Potato and are used to modify rates of growth and phenological development, but the details of these calculations are not presented in this paper. Instead, this paper focuses on the effects of temperature, photoperiod, and light interception on development, and biomass accumulation and partitioning by potato. The input files for soil, climatic, and cultural data required as input to run the model are similar to those for other CERES-type models and are described in IBSNAT Technical Report 5 (IBSNAT, 1986), Thornton et al. (1991), and Ritchie et al. (1992). The following discussion includes: I. prediction of phenology; II. prediction of biomass accumulation and partitioning; and III. model performance.

Model Development

Relative Temperature Functions (subroutine THTIME)

SUBSTOR-Potato uses zero-to-one relative temperature functions based on mean daily air temperature (XTEMP; variable descriptions in Table 1) or soil temperature (ST(LO)) to simulate the response of different plant organs and processes over a wide temperature range. The relative temperature factors (RTF) increase from zero at some base temperature (TB; from 2 to 5°C), to a plateau value of one, then decrease to zero at temperatures of 33 to 35°C, depending on the function. Manrique and Hodges (1989) demonstrated that this type of function was preferable to a linear temperature function because it accounted for the obvious detrimental effect of high temperatures on potato growth and development.

The relative temperature factors for vine growth (RTFVINE) and tuber and root growth (RTFSOIL) are illustrated in Figure 1. The RTFSOIL function (Equation 1) is adapted from the seed piece substrate availability function of Ingram and McCloud (1984) and is used in the model for substrate mobilization from the seed piece, root growth and tuber growth. The RTFVINE function (Equation 2) is generalized using cardinal values from numerous literature sources (e.g., Yamaguchi et al., 1964; Epstein, 1966; Marinus and Bodleander, 1975; Moorby and Milthorpe, 1975; Prange et al., 1990). This function is used to calculate daily leaf expansion and vegetative biomass accumulation. We agree with the Ingram and

Table 1. Summary of simulated and state variables in SUBSTOR-Potato.

Variable name	Units	Description
Subroutine PHENOL (Phenological development)		
CTII	none	Cumulative tuber induction index (TII)
GRORT	g plant^{-1} d^{-1}	Daily root growth
PHPER	hr	Length of photoperiod
RDLFTI	none	Relative daylength factor for tuber initiation
RTFTI	none	Relative temperature factor for tuber initiation
SEEDAV	g plant^{-1}	Available seed reserve, 0.8*seed weight at planting
ST(L0)	degree C	Mean daily soil temperature (surface)
TEMPM	degree C	Weighted mean daily air temperature; 0.75*TEMPMN+0.25*TEMPMX
TEMPMN	degree C	Minimum daily air temperature (input)
TEMPMX	degree C	Maximum daily air temperature (input)
TII	none	Tuber induction index; strength of induction to tuberize
Subroutine THTIME (Calculation of relative thermal time)		
RTFSOIL	none	Relative temperature factor, for tuber & root growth
RTFVINE	none	Relative temperature factor, for vine growth
XTEMP	degree C	Mean daily air temperature
Subroutine GROSUB (Carbon assimilation and partitioning)		
CARBO	g plant^{-1} d^{-1}	Actual daily carbon assimilation
DDEADLF	g plant^{-1}	Weight of daily leaf loss from senescence
DEVEFF	none	Developmental effect, for partitioning during transition from ISTAGE1 to ISTAGE 2

GROLF	g plant^{-1} d^{-1}	Daily leaf growth
GROPLNT	g plant^{-1} d^{-1}	Daily total plant growth
GRORT	g plant^{-1} d^{-1}	Daily root growth
GROSTM	g plant^{-1} d^{-1}	Daily stem growth
GROTUB	g plant^{-1} d^{-1}	Daily tuber growth
LAI	$m^2 m^{-2}$	Leaf area index
LALWR	270 cm^2 g^{-1}	Leaf area to leaf weight
PAR	MJ m^{-2}	Photosynthetically active radiation
PCARB	g plant^{-1} d^{-1}	Potential carbon assimilation from photosynthesis
PLA	cm^2 plant^{-1}	Plant leaf area
PLAG	cm^2 plant^{-1}	Daily plant leaf area growth
PLAS	cm^2 plant^{-1}	Daily plant leaf area senesced, due to stress
PRFT	none	Photosynthetic reduction factor, for temperature
PTUBGR	g plant^{-1} d^{-1}	Maximum potential daily tuber growth
RLGR	none	Relative leaf growth rate
RVCHO	g plant^{-1}	Reserve soluble carbohydrate pool
TIND	none	Proportion of PTUBGR receiving first priority

Subroutine NFACTO (Nitrogen deficit factors)

NDEF1	none	Relative N deficit effect on photosynthesis
NDEF2	none	Relative N deficit effect on growth
NFAC	none	Leaf N concentration, relative to TCNP and TMNC
TANC	%	Actual N concentration in vines
TCNP	%	Critical vine N concentration
TMNC	%	Minimum vine N concentration, below which growth ceases

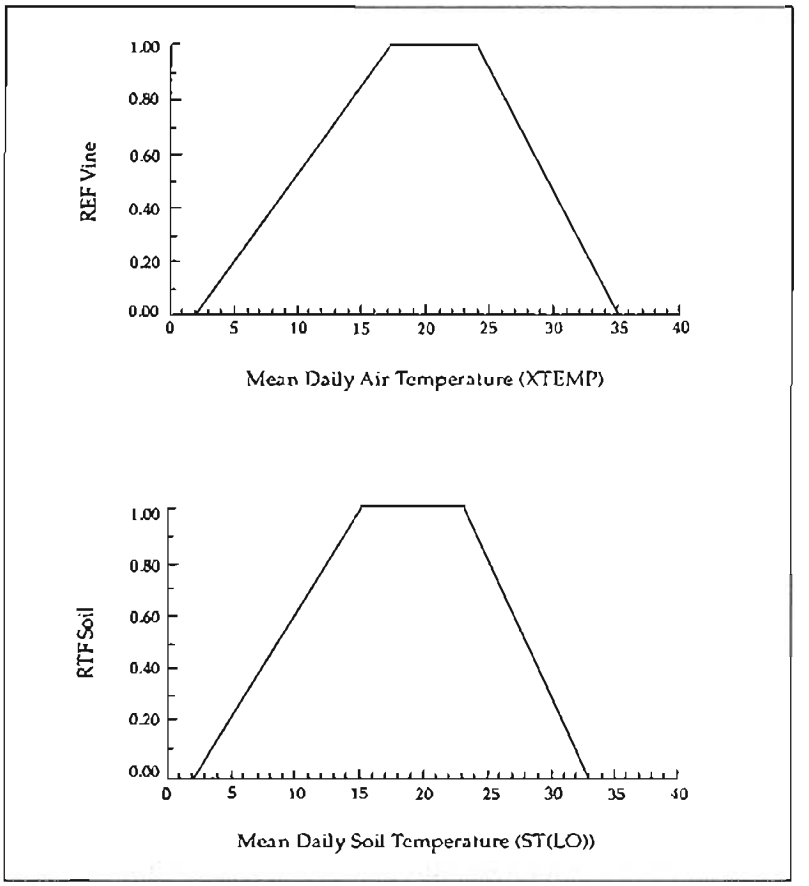


Figure 1. Relative thermal time functions for vine growth (RTFVINE) and for tuber and root growth (RTFSOIL).

McCloud (1984) conclusions that potato leaves, tubers and roots have different temperature response functions. However, there are insufficient data to distinguish separate temperature response functions for growth rate and development rate (Sands et al., 1979). Thus, RTFVINE is also used to calculate the rate of phenological development.

RTFSOIL = 0 if ST<=2 or ST>33 [1]
= 0.079 (ST-2) if 2<ST<=15
= 1 if 15<ST<=23
= 1 - 0.1 (ST-23) if 23<ST<=33

RTFVINE = 0 if XTEMP<=2 or XTEMP>35 [2]
= 0.0667 (XTEMP-2) if 2<XTEMP<=17
= 1 IF 17<XTEMP<=24
= 1 - 0.0909 (XTEMP-24) if 24<XTEMP<=35

Phenological Development (subroutine PHENOL)

Growth Stages

Potato growth is divided into five phenological stages (variable ISTAGE) within SUBSTOR-Potato, as follows:

ISTAGE 5: pre-planting
ISTAGE 6: planting to sprout germination
ISTAGE 7: sprout germination to emergence
ISTAGE 1: emergence to tuber initiation
ISTAGE 2: tuber initiation to maturity.

The variable XSTAGE is used to mark progression through each ISTAGE. It is calculated as a function of accumulated RTFVINE during ISTAGES 5, 6, 7 and 2 and as a function of RTFVINE and photoperiod during ISTAGE 1.

Sprout Germination and Emergence

In SUBSTOR-Potato, preference is given to observed emergence dates (IEMERG) to be included in the appropriate input file, rather than simulation of sprout germination and emergence. We based this decision on the paucity of reliable calibration data for pre-emergent development and, more importantly, the inherent difficulty in obtaining accurate assessments of seed piece physiological age that affects the rate at which potato reaches both of these phenological events (Greenwood et al., 1985a; Van der Zaag and Van Loon, 1987). If IEMERG is not input, germination of unsprouted seed and sprout elongation of both sprouted and unsprouted seed are simulated using RTFSOIL. For both unsprouted and sprouted seed, emergence occurs when cumulative sprout length (SPRLEN) > depth of planting (SDEPTH).

Tuber Initiation

Mechanistic simulation models for potato (e.g., Ng and Loomis, 1984) have attempted to model the timing, rate and duration of tuber initiation. Because of the multitude of factors affecting tuber initiation and the lack of understanding of how these factors affect initiation at the physiological level, we have taken an approach suggested by Sands et al. (1979) to estimate the timing of tuber initiation; that is, extrapolation of linear tuber bulking rate back to the time axis (zero tuber weight). This approach makes tuber initiation an instantaneous event, like emergence, rather than a distinct growth stage. Because the initial lag phase of tuber growth is not directly simulated in SUBSTOR-Potato, the estimated date of tuber initiation will, by necessity, be later than the observed date.

Tuber initiation effectively divides the post-emergence growing season into vegetative and tuber-bearing stages and an accurate estimation of when initiation occurs is critical. Plant

leaf area at initiation, and thus the plant's ability to intercept radiation during tuber bulking, strongly influences end-of-season tuber yield when nutrients and water are not limiting during bulking (Moorby and Milthorpe, 1975; MacKerron and Waister, 1985). In SUBSTOR-Potato, the timing of tuber initiation is a function of cultivar response to both temperature and photoperiod, with these responses modified by plant N status and soil water status. In developing the theoretical framework for predicting tuber initiation, we have relied heavily on the theory, put forth by Ewing (1981), Wheeler and Tibbetts (1986) and others, that tuber initiation by "early" cultivars is less sensitive to non-optimal conditions (i.e., high temperatures and/or long photoperiods) than initiation by "late" cultivars.

Researchers have established that (*i*) cultivars differ in the threshold temperature above which tuber initiation is inhibited (Ingram and McCloud, 1984; Ben Khedher and Ewing, 1985; Snyder and Ewing, 1989), and (*ii*) tuber initiation is influenced more strongly by daily minimum temperature than by daily mean or maximum temperature (Slater, 1968; Manrique and Bartholomew, 1991). We developed a dimensionless cultivar-specific relative temperature factor (RTFTI; range of 0 to 1) to simulate the effect of high temperatures on tuber initiation. This function is similar in shape to the RTFVINE function. Cultivars are assigned a coefficient for critical temperature (TC; Table 2), above which tuber initiation is inhibited to some degree. Cultivar TC corresponds roughly to "early" versus "late," with early cultivars having a higher value for TC. The RTFTI value above TC is

Table 2. Genetic coefficients in SUBSTOR-Potato for leaf expansion rate (G2), tuber growth rate (G3), determinacy (PD), and sensitivity of tuber initiation to photoperiod (P2) and temperature.

Cultivar	G2	G3	PD	P2	TC
Units	$cm^2 m^{-2} d^{-1}$	$gm^{-2} d^{-1}$			*C*
Segago	2000.0	22.5	0.7	0.8	15.0
Russet Burbank	2000.0	22.5	0.6	0.6	17.0
Katahdin	2000.0	25.0	0.7	0.6	19.0
Maris Piper	2000.0	25.0	0.8	0.4	17.0
Desiree	2000.0	25.0	0.9	0.6	17.0
Norchip	2000.0	25.0	1.0	0.4	17.0

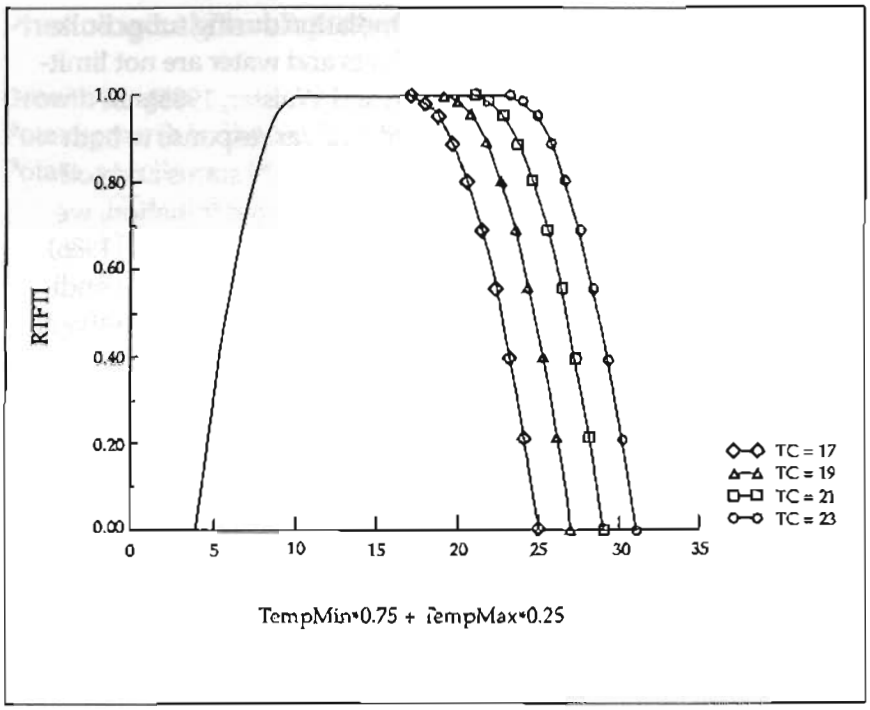


Figure 2. Relative thermal time function for effect of temperature on tuber initiation. (RTFTI is the relative temperature factor for tuber initiation; TC is the cultivar-specific critical temperature above which tuber initiation is inhibited. TEMPMIN and TEMPMAX are minimum and maximum daily air temperatures.)

calculated as:

$$RTFTI = 1.0-0.0156*(TEMPM-TC)^2, \quad TC<TEMPM<TC+8 \quad [3]$$

$$RTFTI = 0 \quad TEMPM>TC+8, \quad \text{where}$$

$$TEMPM = 0.75*TEMPMN+0.25*TEMPMX \quad [4]$$

where TEMPMN and TEMPMX are daily minimum and maximum temperatures, respectively. The result (Figure 2) is a family of identical, decreasing curves dependent on a single coefficient, TC.

The calculation of a relative daylength factor (RDLFTI) is similar to that for RTFTI. When the photoperiod is less than 12 hr, RDLFTI equals 1.0 for all cultivars. This common photoperiod insensitive phase was demonstrated by Rasco et al. (1980), who showed that different cultivars initiate tubers at about the same time under favorable (i.e., short) photoperiod. Differences in time to tuber initiation become apparent under long photoperiods, or as discussed above, under high temperatures.

For photoperiods greater than 12 hr, early and late cultivars are differentially sensitive to increasing photoperiod, with early cultivars being less sensitive than late cultivars (Ewing,

1981; Wheeler and Tibbetts, 1986; Snyder and Ewing, 1989). Thus, early cultivars should have a higher RDLFTI under long (16 to 20 hr) photoperiods. To accommodate these differences, each cultivar is assigned a dimensionless genetic coefficient (P2) indicating sensitivity of tuber initiation to photoperiod. This coefficient, effectively ranging from 0.2 to 0.8, defines the shape of the RDLFTI curve by:

$$RDLFTI = (1.0-P2)+0.00694*P2*(24.0-PHPER)^2 \quad [5]$$

where PHPER is photoperiod (in hrs). The resultant family of RDLFTI curves is shown in Figure 3. Examples of the upper- and lowermost RDLFTI curves are 'Norland' and *Andigena*, respectively (Wheeler and Tibbetts, 1986; Rasco et al., 1980).

A tuber induction index (TII) is calculated daily as:

$$TII = (RTFTI*RDLFTI)+0.5*(1.0-AMIN1(SWDF2,NDEF2)) \quad [6]$$

and is used as a measure of the relative strength of the induction to tuberize. AMIN1 is the FORTRAN command for selecting the minimum value from a list of numerical values. The modifiers SWDF2 and NDEF2 are factors for soil water and N stress, on expansion growth. These factors, through their effect on TII, hasten tuber initiation under stress conditions.

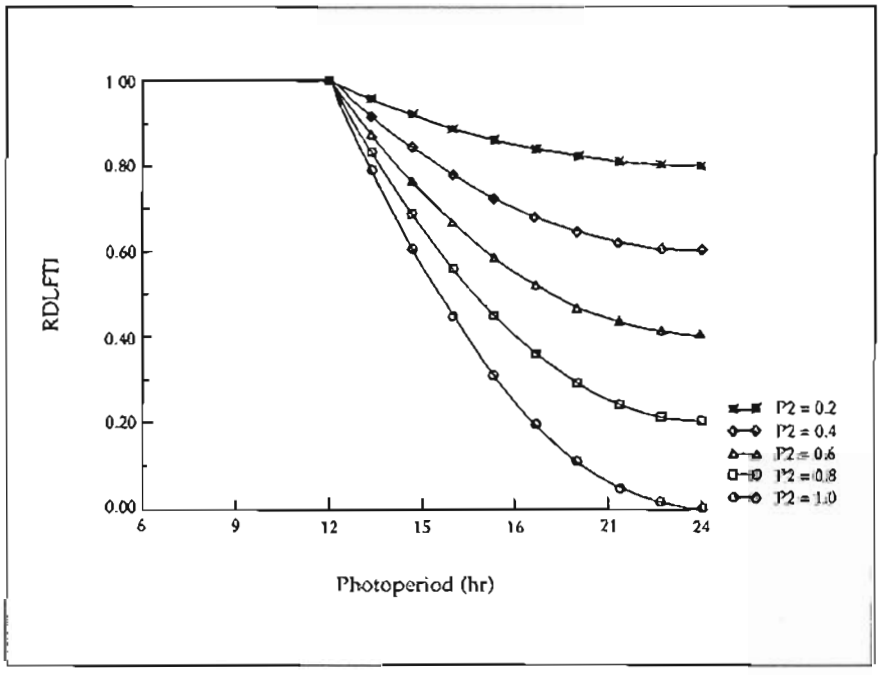


Figure 3. Function for relative effect of photoperiod on tuber initiation. (RDLFTI is the relative daylength factor for tuber initiation; P2 is the cultivar-specific genetic coefficient.)

The multiplicative function of RTFTI and RDLFTI is necessary because long photoperiods and high temperature act synergistically in inhibiting tuber initiation (Snyder and Ewing, 1989; Ewing et al., 1990).

From the daily calculation of TII (beginning at emergence), a cumulative tuber induction index (CTII) is calculated as:

$$CTII = \Sigma TII \qquad [7]$$

Available calibration data sets indicate that zero tuber weight, obtained by extrapolation of linear bulking back to the time axis, corresponds to a CTII value of approximately 20. Thus, this value is used to flag tuber initiation in SUBSTOR-Potato.

Biomass Accumulation and Partitioning (subroutine GROSUB)

Pre-emergent Growth (ISTAGE 7)

The seed piece represents the only carbon (C) source to support growth from sprout germination to emergence. Growth during this stage is simulated only if IEMERG is not supplied by the user. Maximum availability of seed piece C (SEEDAV) for growth is adapted from Ingram and McCloud (1984) and pre-emergent sprout growth (SPRWT) is a linear function of RTFSOIL. Daily root growth (GRORT) is assumed to equal daily SPRWT. If the sum of sprout and root growth exceeds SEEDAV, the growth of both components is reduced by the fraction SEEDAV/(SPRWT + GRORT).

Vegetative Growth (ISTAGE 1)

Post-emergent growth is supported by three C sources: seed reserves, current photosynthetic assimilate, and reserve carbohydrate. Immediately following emergence, growth is supported primarily by seed reserves. As in Ng and Loomis (1984), the availability of seed reserve decreases as plant leaf area (PLA) increases. This is accounted for in the model by calculating SEEDAV as a function of PLA up to 400 $cm^2 plant^{-1}$, after which SEEDAV is zero.

Potential photosynthetic C assimilation (PCARB) is calculated as:

$$PCARB = 3.5*PAR/PLANTS*(1.0*EXP(-0.55*LAI)) \qquad [8]$$

where PAR is photosynthetically active radiation (MJ m^{-2}), EXP is the FORTRAN command for an exponential function, and LAI is leaf area index ($m^2 m^{-2}$). The radiation use efficiency (RUE) of 3.5 g MJ^{-1} PAR is adapted from literature values (Sale, 1973; Allen and

Scott, 1980; Jeffries and MacKerron, 1989; Manrique et al., 1991), as is the extinction coefficient, *k*, of -0.55 (Allen and Scott, 1980). The above calculation estimates net C assimilation assuming that nutrients and soil water are non-limiting. Actual C assimilation (CARBO) is calculated by:

CARBO = PCARB*AMIN1(PRFT,SWDF1,NDEF1))+0.5*DDEADLF [9]

The unitless modifiers are for temperature (PRFT), soil water (SWDF1), and N stress (NDEF1) effects, on photosynthetic efficiency. One-half of the C in senesced leaves (DDEADLF) is translocated prior to abscission (Johnson et al., 1986). If CARBO is greater than daily growth demand, excess C enters a soluble carbohydrate pool (RVCHO), limited to 10 percent of haulm dry weight (Ng and Loomis, 1984).

The following priorities for C use are used in SUBSTOR-Potato. Photosynthetic assimilate is always used first. If additional C is needed to meet growth demand, either seed reserves or carbohydrate reserves may be used according to the criteria described above. The use of seed reserves and carbohydrate reserves are mutually exclusive in the model, because the carbohydrate reserve is allowed to accumulate when PLA > 400 cm^2 $plant^{-1}$, when seed reserve is no longer available.

During vegetative growth, potential leaf expansion (PLAG) is calculated first as:

PLAG = EXP(RLGR)*PLA-PLA [10]
RLGR = 0.50*RTFVINE [11]
GROLF = PLAG (1/LALWR) [12]

where RLGR is the relative leaf growth rate (Ingram and McCloud, 1984). This estimate of leaf expansion may be modified for N stress and soil water stress effects on expansion growth (NDEF2 and SWDF2), and is then used to calculate daily biomass addition to leaves (GROLF), based on a leaf area to leaf weight ratio (LAWLR) of 270 cm^2 g^{-1}. Daily stem growth (GROSTM) is assumed to equal GROLF, while partitioning of biomass for root growth (GRORT) changes with phenological stage. Growth of all organs are given equal priority during ISTAGE 1 (e.g., Munns and Pearson, 1974). Thus, a shortage of C for growth results in all organ growth potentials being reduced by the fraction CARBO/GROPLNT, where GROPLNT is the summation of GROLF, GROSTM, and GRORT.

Tuber-bulking (ISTAGE 2)

The initiation of tubers elicits several fundamental changes in the growth of the potato plant. First, as shown by Sale (1973), radiation use efficiency (RUE) *may* increase by 50

percent or more. Sale attributed this increase in RUE to the presence of the tubers, which represented a large, rapidly growing 'sink' for photosynthetic assimilate. Rather than attempting to dynamically simulate the effect of sink size on RUE, SUBSTOR-Potato calculates PCARB during tuber bulking using [8], with the RUE increased to 4.0 g MJ^{-1} PAR. Actual carbon assimilation (CARBO) is calculated as in [9].

The second change in growth after tuber initiation, involves biomass partitioning to competing organs or sinks. Unlike growth during ISTAGE 1, when the proportion of total assimilate partitioned to each organ remains relatively constant under stressed vs. non-limiting conditions, partitioning during ISTAGE 2 is a dynamic process potentially influenced by many factors. For example, partitioning to tubers is favored (i.e., a greater proportion of the total is allocated to tubers) by low temperature, short photoperiod, and low to moderate soil water or N level. Because these same factors hasten tuber initiation, Ewing (1981) suggested that the "tuberization stimulus" influences *both* initiation and partitioning of biomass after initiation. To make this partitioning response operational in SUBSTOR-Potato, we assume that tubers are given first priority on available assimilate (from both current photosynthesis and the soluble reserve carbohydrate pool), thereby eliminating the need to directly estimate partitioning coefficients to allocate C. When the tuber sink is small or conditions are non-limiting, this assumption is not critical because nearly all daily growth demands can be met. However, when stress reduces the amount of assimilate available or the tuber sink is very large, growth of vines and roots can be greatly reduced by imposing this priority. This was demonstrated experimentally by Munns and Pearson (1974), who found that drought during tuber bulking could cause a very rapid cessation of vine growth while tubers continued to grow.

Environmental and soil factors are used in SUBSTOR-Potato to modify potential tuber growth demand. The estimation of tuber growth demand is a two-step process. The first step is to estimate the proportion of maximum potential tuber growth that will receive first priority on assimilate (variable TIND). TIND is independent of the size of the tuber sink and is based on the concept that tuber sink strength is analogous to tuber induction strength. TIND is calculated as:

```
TIND = (ΣDTII/3)*(1/NFAC)*DEVEFF,     when NFAC>1     [13]
TIND = (ΣDTII/3)*DEVEFF,              when NFAC<1
```

where

```
DTII = RTFTI+0.5*((1.0-AMIN1(SWDF2,NDEF2,1.0))       [14]
DEVEFF = AMIN1((XSTAGE-2.0)*10*PD,1.0)
```

DTII estimates the daily tuber sink strength as a function of temperature, and nutrient and soil water status. A three-day moving average of DTII is used because the partitioning response to changing conditions is not instantaneous (Ewing, 1981), and serves to buffer against excessive fluctuations in partitioning. NFAC, which is used to calculate N deficit factors, is included only when greater than one, indicating the promotion of vegetative growth by excessive N. DEVEFF is an artificial variable that alters partitioning during the transition from vegetative to tuber-bearing stages. For example, for determinate cultivars (PD = 1), DEVEFF equals 1.0 when XSTAGE equals 2.1 (i.e., approximately one week after initiation).

Estimation of potential and actual tuber growth rate (PTUBGR and GROTUB) is the second step. PTUBGR is a function of maximum tuber growth rate (genetic coefficient G3) and temperature, in the form:

PTUBGR = G3*RFTSOIL/PLANTS [15]

Tuber growth, potential leaf expansion (PLAG), and leaf, stem, and root growth (GROLF, GROSTM, and GRORT) are then calculated as:

GROTUB = PTUBGR*AMIN1(SWDF2,NDEF2,1.0)*TIND [16]
PLAG = (G2*RFTVINE/PLANTS)*AMIN1(SWDF2,NDEF2,1.0)
GROLF = PLAG/LALWR
GROSTM = GROLF*0.75
GRORT = (GROLF+GROSTM)*0.2

G2 is a genetic coefficient for maximum leaf expansion rate, currently equal to 2000 $cm^2m^{-2}\,d^{-1}$ for all cultivars because of the lack of evidence to the contrary. After calculating potential growth, actual daily growth of each organ is determined within an hierarchical routine, where assimilated C becomes increasingly limiting in relation to potential growth. The structure of this routine is briefly illustrated below.

1. If CARBO > GROTUB, CARBO is updated as CARBO = CARBO-GROTUB to reflect the priority given to tuber growth.
2. If CARBO or CARBO+RVCHO > GROLF+GROSTM+GRORT, all organs grow at the estimated rate. The reserve carbohydrate pool is adjusted accordingly.
3. If CARBO+RVCHO < GROLF+GROSTM+GRORT, all reserve carbohydrate is used and the growth of leaves, stems and roots is reduced by a growth reduction factor (GRF)

   GRF = (CARBO+RVCHO)/(GROLF+GROSTM+GRORT) [17]

4. If CARBO+RVCHO < GROTUB, tuber growth is reduced by (CARBO+RVCHO)/GROTUB, and growth of all other organs is set to zero.

Stress Factors

Soil Water Deficit Factor (SWDF)

The effects of soil water deficit (SWD) on potato were reviewed by Van Loon (1981), and include reduced expansion growth and photosynthetic rates, increased allocation of biomass to tubers, and increased rate of phenological development. The manner in which these effects are simulated in SUBSTOR-Potato were discussed previously. Two generalized SWD factors are used in this model: SWDF1 and SWDF2 simulate the effects of SWD on photosynthesis and growth, respectively. SWDF2 is also used in various capacities as a modifier for developmental rates and partitioning of biomass.

Leaf and soil water potentials are not directly estimated in the model. Thus, SWDF1 and SWDF2 can be viewed as the relative deficiency between potential water uptake by roots and transpiration from the leaf surface. Both deficit factors are calculated as a ratio of total root water uptake potential (TRWU) and transpiration (EP1). TRWU is a function of rooting depth, root length density, and soil water content and distribution. The greater sensitivity of leaf expansion to SWD was demonstrated by Munns and Pearson (1974) and Jeffries and MacKerron (1987, 1989), attributable to the loss of leaf cell turgor under deficit conditions. The use of linear SWDFs is supported by Jeffries (1989), who found that leaf extension rate was directly proportional to leaf water potential and leaf turgor. In addition, Vos and Oyarzain (1987) observed a linear relationship between leaf water potential and photosynthetic rate.

Nitrogen Deficit Factor (NDEF; subroutine NFACTO)

The N concentration of potato vines is a much more sensitive indicator to transient changes in N fertility than is the concentration in either roots or tubers. Vine N concentration also generally declines with advancing maturity, even under N sufficient conditions. For these reasons, the estimation of N deficiency or excess in SUBSTOR-Potato is based solely on vine N concentration.

Critical N concentration in the vines or "tops" (TCNP) is the concentration required to maintain maximum growth and photosynthetic rates. Minimum N concentration (TMNC) is the concentration below which growth and photosynthesis cease. The values for TCNP are approximated from Saffigna and Keeney (1977) and Greenwood et al. (1985). In SUBSTOR-Potato, the N concentration in roots and tubers (that have a critical N concentration, TUBCNP, of 1.4 percent regardless of growth stage) is maintained at or slightly above

the respective critical values, reflecting the lack of fertility effects on these organs (e.g., Carter and Bosma, 1974; Saffigna and Keeney, 1977; Kleinkopf et al., 1981).

The extent of N deficiency (NFAC) is measured on a linear scale relative to the minimum and critical concentrations by:

$$NFAC = (TCNP-TANC)/(TCNP-TMNC) \quad [18]$$

where TANC is the actual N concentration in the vines. NFAC is then used to calculate NDEF1 and NDEF2, for photosynthesis and growth, respectively:

$$NDEF1 = NFAC \quad [19]$$
$$NDEF2 = 0.95*NFAC$$

Under conditions of excess N, NFAC is allowed to exceed 1.0 and is used to delay development of the plant.

Performance of Substor-Potato

The validation data set for SUBSTOR-Potato (Table 3) includes a wide range of geographical regions, cultivars, and management intensities (e.g., irrigation, N fertilization). Because of the diversity of these data, we have intentionally limited our presentation of model validation in several ways. First, we chose not to conduct a formal sensitivity analysis in our evaluation. The inclusion of diverse data in the validation set seemed to us to make such analyses redundant. For example, within the validation set climatic conditions range from cool temperature/short photoperiod to high temperature/short photoperiod (both from Hawaii; Manrique and Bartholomew, 1991) to cool temperature/long photoperiod (Scotland; Jeffries and MacKerron, 1987, 1989). Likewise, early, intermediate, and late cultivars (e.g., Norchip, Katahdin, and Russett Burbank) were included in the validation set.

The second limitation is the scope of validation specifically for plant stress factors. Although the validation set inherently addresses the effects of N and soil water stresses on development, growth and yield, we have not included validation statistics on plant-soil N (or water) balance. This decision was based on the fact that most data sets did not include sufficient soil and plant N analyses to adequately evaluate the nitrogen subroutines in the model.

Table 3. Description of validation data for SUBSTOR-Potato V2.0.

Location	Latitude	Cultivar(s)	Year(s)	Reference(s)
Murrumbridge, Australia	35.0 S	Sebago	1970, 1971	Sale (1973)
Olinda, HA (1097 m)	20.5 N	Katahdin, Desiree	1986	Manrique & Bartholomew (1991)
Hamakuapoko, HA (91 m)	20.6 N	Katahdin, Desiree	1986	Manrique & Bartholomew (1991)
Haleakala, HA (640 m)	23.0 N	Katahdin, Desiree	1986	Manrique & Bartholomew (1991)
Ithaca, NY		Katahdin	1980, 1981, 1982, 1985, 1986	Ewing et al. (1990)
Kimberly, ID	42.3 N	Russet Burbank	1978	Ritchie (unpublished)
Aberdeen, ID	43.0 N	Russet Burbank	1978	Ritchie (unpublished)
Entrican, MI	43.2 N	Russet Burbank	1985, 1986, 1987, 1988	Ritchie (unpublished)
Hermiston, OR	45.8 N	Russet Burbank	1988	English (unpublished)
Grand Forks, ND	47.9 N	Russet Burbank Norchip	1985, 1986, 1987	Ritchie (unpublished)
Invergowrie, Scotland	56.5 N	Maris Piper	1984, 1985, 1986, 1987	Jeffries & MacKerron (1987, 1989)

To demonstrate the performance of SUBSTOR-Potato, we have included two distinct approaches in the following discussion. First, the traditional approach showing the relationship between simulated and observed values using all available validation data. And second (where appropriate), simulated versus observed values for a single growing season at a specific location (i.e., seasonal growth pattern).

Simulation of Tuber Initiation Date

Tuber initiation (TI) represents the critical phenological event of potato during the growing season, and unlike grain crops for example, is the only distinct phenological event after emergence. Simulation of the timing of this event is made more difficult by the indeterminate growth habit of some potato cultivars. For these reasons, we put considerable effort into developing a framework for simulating this event that would be useful across environments and cultivars. Within our validation data set, a strong linear relationship was apparent between simulated and observed time to TI (Figure 4), demonstrating the utility of our approach to simulating TI. Despite unknown differences in seed piece physiological age, SUBSTOR-Potato accurately simulated TI under conditions highly conducive for initiation (30 to 40 DAP) and under conditions that delayed initiation until mid-season (80 to 90 DAP). The intercept (a) of the regression of simulated on observed values was 19.35,

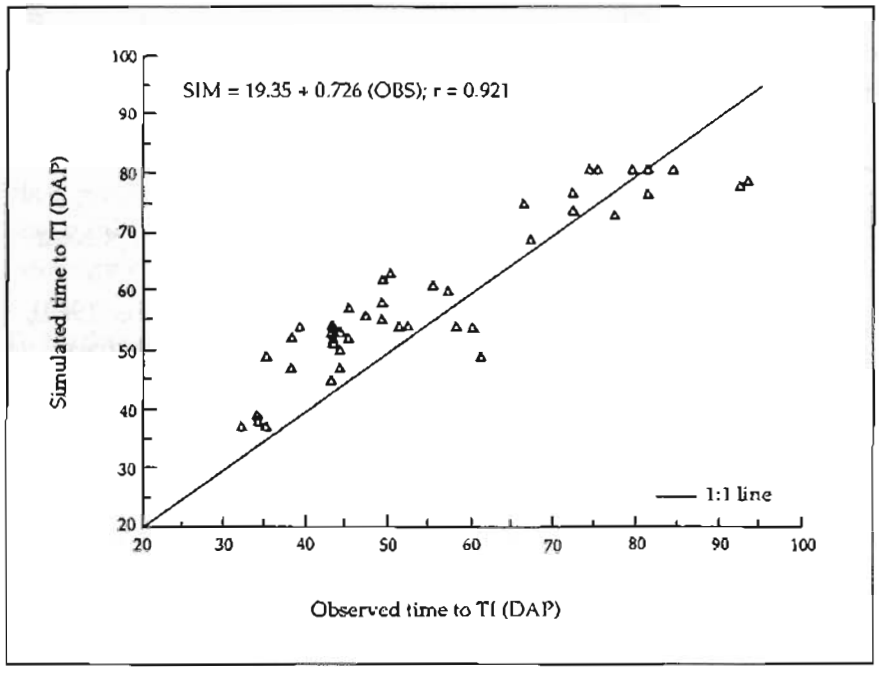


Figure 4. Simulated versus observed time to tuber initiation. n = 41.

indicating that the initiation occurred earlier than the model simulated. This was expected because the model does not simulate the discrete lag phase of tuber growth that occurs immediately after initiation.

The general approach we have taken to simulate phenological development, incorporating both temperature and photoperiod effects, is not unique to SUBSTOR-Potato or potato simulation models in general. Hammer et al. (1989) analyzed grain sorghum (Sorghum bicolor (L.) Moench.) development in diverse environments and developed functions for temperature and photoperiod response very similar to those reported here. They also recognized that cultivars (or cultivar groups) differ in their response to these factors. Russell and Stuber (1985) demonstrated that tassel initiation in maize (Zea mays L.) involved an interaction of temperature and photoperiod effects, and that the effect of photoperiod was minimal after tassel initiation. Similar relationships have been shown for the rate of development by soybean [Glycine max (L.) Merr.] (Major et al., 1975).
The integration of climatic effects, the extrapolation of linear tuber bulking to define tuber initiation, and the recognition of cultivar differences, which are features unique to this model, represent a new approach to modeling potato development. Compared to previous models, this approach may be preferable for several reasons. First, compared to simulation based solely on temperature, this approach takes into account the obvious effects of photoperiod on delaying or hastening TI. Second, although we realize that refinement of the cultivar-specific genetic coefficients (Table 2) are likely, this approach recognizes that cultivars are distinctly different in their response to climate. And third, compared to mechanistic models, this approach greatly simplifies the simulation of TI. This may change as the physiological basis and control factors for TI are clarified.

Simulation of Leaf Growth

Carbon assimilation by potato is directly related to the ability of the plant to intercept solar radiation, which is in turn a function of photosynthetically active leaf area. SUBSTOR-Potato does not simulate the development of individual leaves as in Ng and Loomis (1984), but rather the development of the entire canopy. The model's ability to simulate leaf area expansion or leaf biomass was evaluated in two ways. First, we compared simulated and observed maximum LAI values (Figure 5). Essentially, this evaluates the model equations for potential leaf growth given prevailing climatic and fertility conditions. Compared to observed values, the model performance was poorer than that for tuber initiation, with a correlation coefficient (r) of 0.47. For maximum LAI between 4 and 6, the model tended to overestimate leaf area. Most likely, this was due to the lack of disease and(or) insect defoliation subroutines within the model, which would serve to constrain leaf area. The potential for coupling pathogen submodels with CERES-type crop models was discussed by

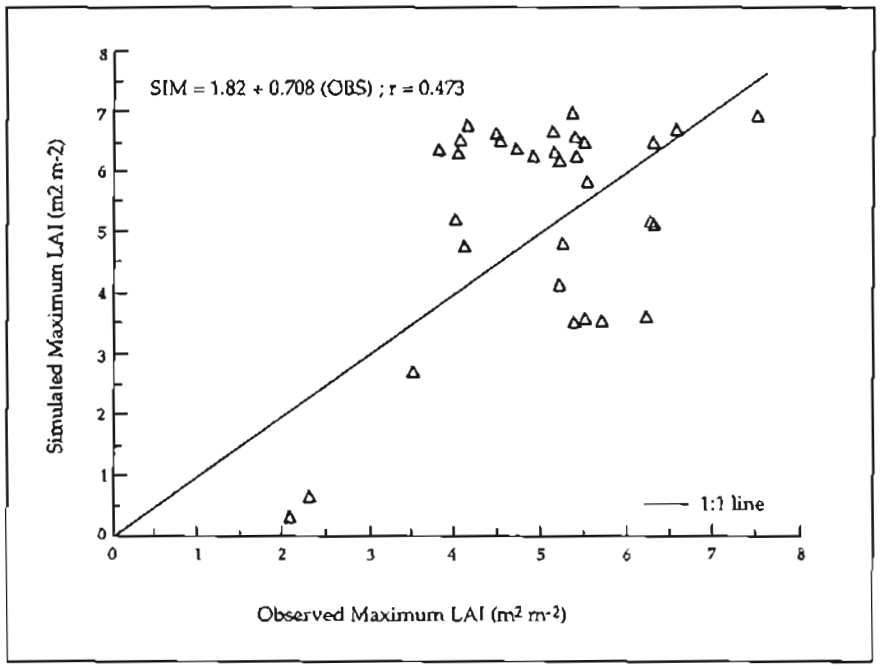


Figure 5. Simulated versus observed maximum leaf area index (LAI) during growing season. n = 32.

Rickman and Klepper (1991), and should represent a major goal of plant simulation modeling. At very low observed maximum LAI, the model underestimated leaf area, suggesting that the model may be overly sensitive to severe environmental conditions (e.g., drought, N stress). As pointed out by Greenwood et al. (1985b) and Jones and Kiniry (1986), crop growth simulation is more difficult under extreme conditions because small differences in initial soil water and soil N levels have disproportionally dramatic effects on simulated plant growth. The designation of tubers as having first priority on assimilated C may also play a role; under extreme conditions that result in less leaf area being produced, this priority may cause leaf growth to cease completely.

To further evaluate discrepancies in the simulation of leaf growth by the model, we present seasonal leaf growth patterns for three location-years: Oregon, 1988; Scotland, 1984; and North Dakota, 1987. These data represent very high, moderately high, and low tuber yield potentials. Simulated leaf growth under long photoperiod and irrigated conditions of Oregon was nearly identical to the observed leaf growth pattern (Figure 6A). This indicates that the functions to estimate leaf growth in the nearly complete absence of plant stress were appropriate. Under conditions slightly less favorable for canopy growth (Scotland), simulation of initial leaf development followed the growth patterns observed i the field (Figure 6B). However, observed leaf growth essentially ceased at about 40 g plant^{-1}, while

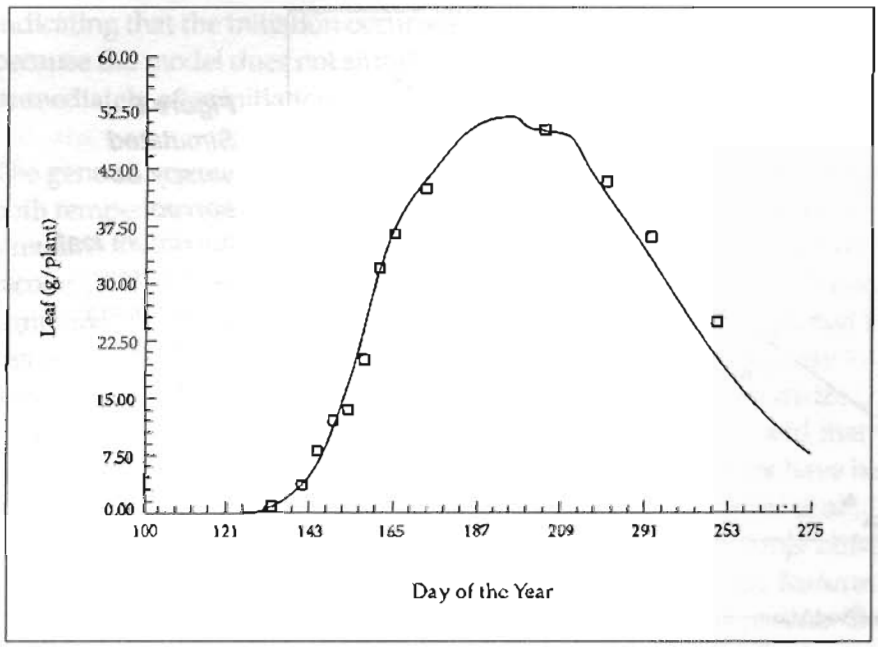


Figure 6A. Simulated (line) versus observed (symbols) seasonal leaf growth for Russet Burbank, Oregon 1988.

simulated growth continued to more than 50 g plant^{-1}. As mentioned previously, the model may continue to simulate leaf growth because there is no simulation of disease or

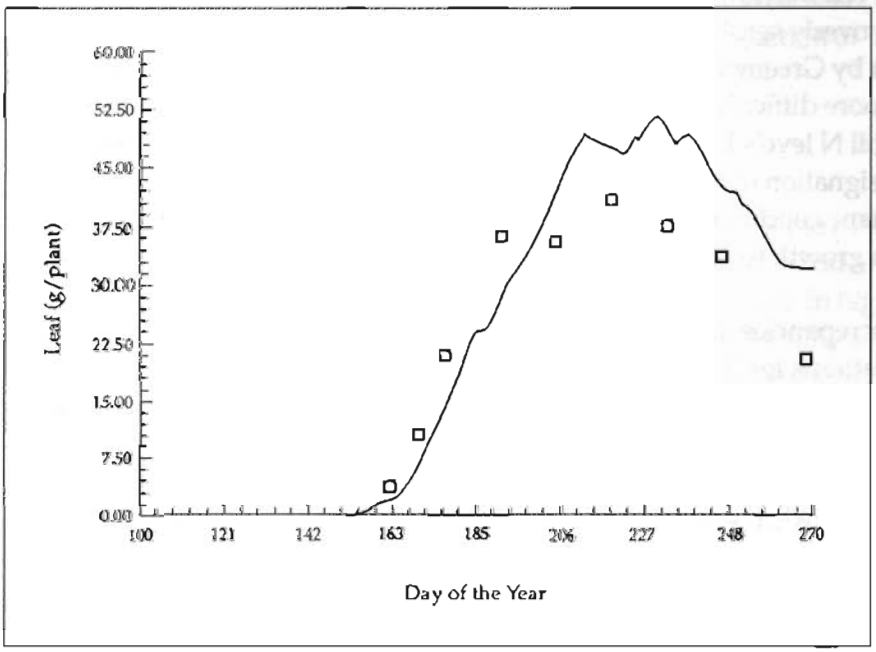


Figure 6B. Simulated (line) versus observed (symbols) seasonal leaf growth for Maris Piper, Scotland 1984.

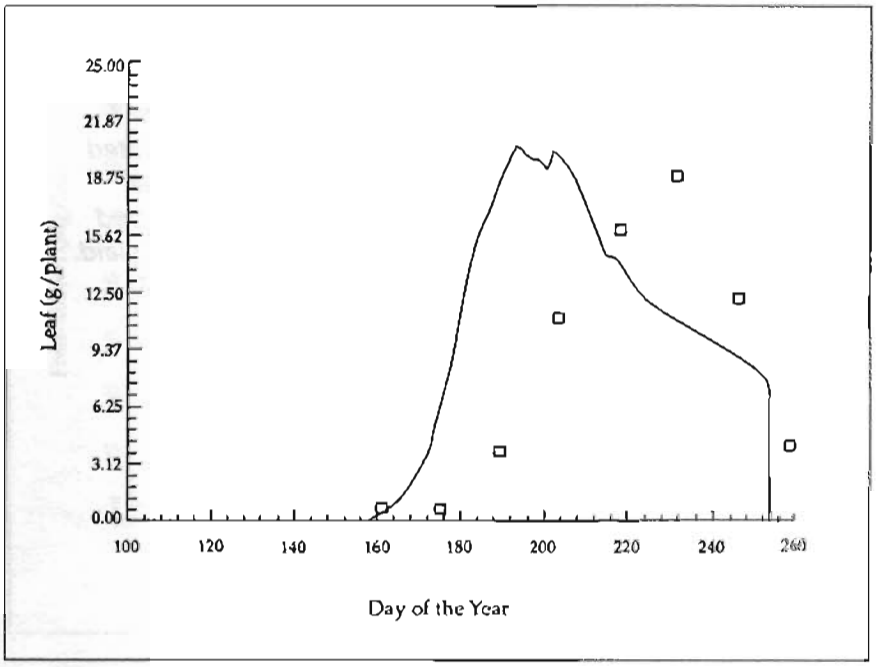


Figure 6C. Simulated (line) versus observed (symbols) seasonal leaf growth for Russet Burbank, North Dakota 1985.

insect effects in the model. Much smaller plants were observed in North Dakota (Figure 6C) due to the lack of irrigation. The model was successful in estimating the maximum canopy size (about 20 g plant^{-1}), but the time that maximum canopy size occurred was approximately 50 D prior to that observed in the field. The cessation of simulated leaf growth at Julian date 190 corresponds exactly to when tuber initiation occurred, providing evidence that the priority for assimilated C may be too rigid.

Simulation of Tuber Yield

Kiniry and Jones (1986) discuss the integration of numerous model processes in simulating maize grain yield within the CERES-Maize model. They noted that simulated grain yield was "affected by virtually every process simulated by the model." The same can be said of tuber yield simulation by SUBSTOR-Potato. Tuber yield is influenced by rate of development, intercepted radiation and use efficiency, biomass partitioning, fertility and soil water status, and other factors. Tuber yield also represents the variable of economic importance. Thus, accurate simulation of tuber yield is essential. We found that the simulation of tuber yield by our model was quite accurate (Figure 7), over a range of two or more than 20 Mg DM ha^{-1}. The intercept (0.40) and slope (0.958) and r value (0.897) of the predicted versus observed regression indicates that the model was not biased across this ten-fold range in

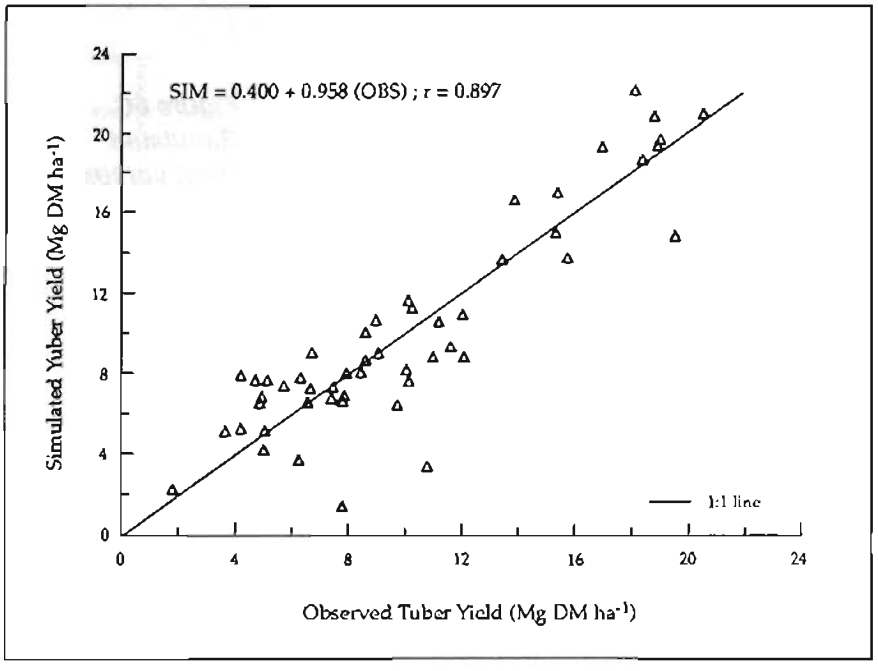


Figure 7. Simulated versus observed tuber yield. n = 54.

tuber yield. Except for two outlying data points where yield was greatly underestimated, there was no systematic under- or overestimation of tuber yield. Seasonal tuber growth, for

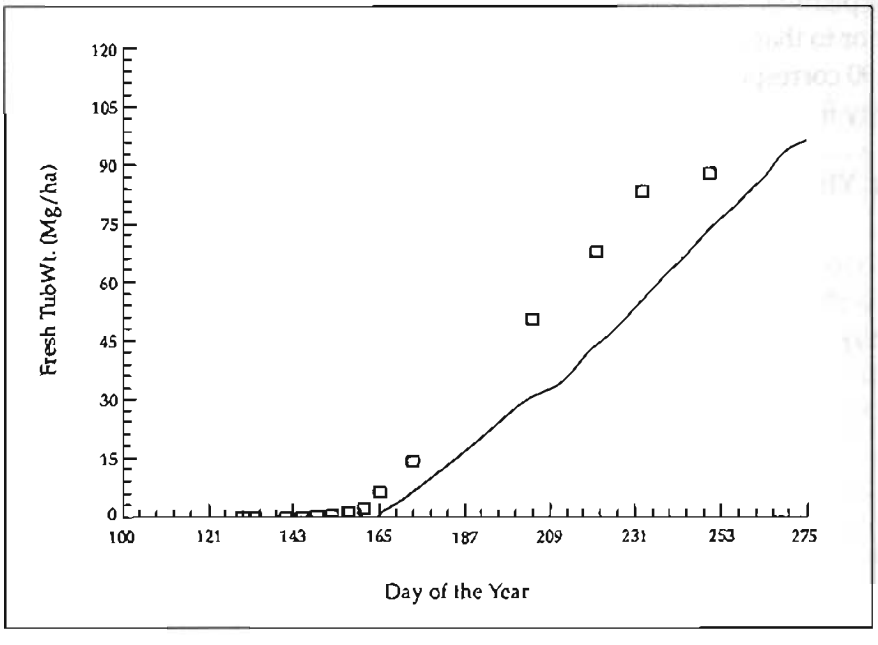


Figure 8A. Simulated (line) versus observed (symbols) seasonal tuber growth for Russet Burbank, Oregon 1988.

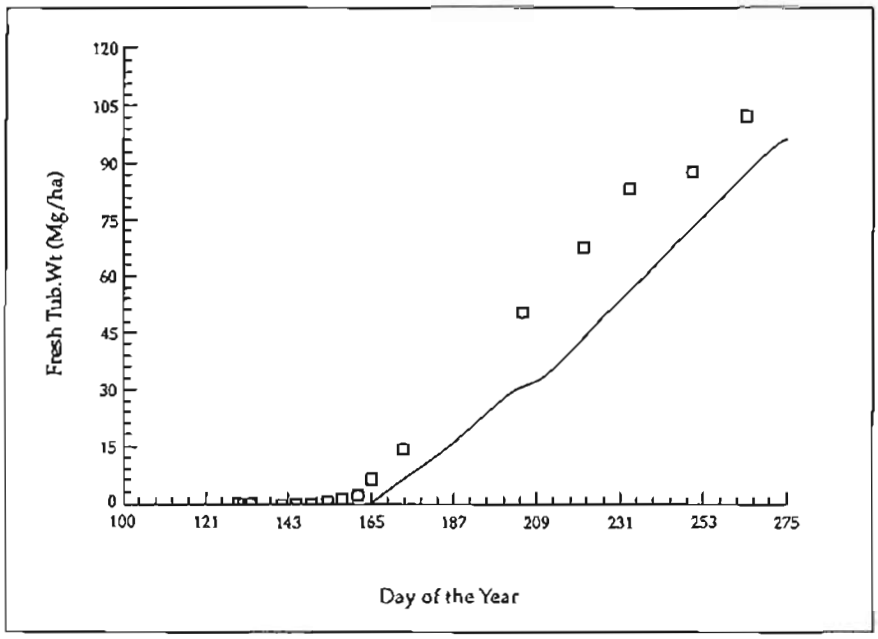


Figure 8B. Simulated (line) versus observed (symbols) seasonal tuber growth for Maris Piper, Scotland 1984.

the same locations described previously for leaf growth, also indicates that the rate of tuber growth was accurately simulated by SUBSTOR-Potato (Figure 8A,B,C). This is true even where the simulation of leaf growth was not successful (e.g., North Dakota data set).

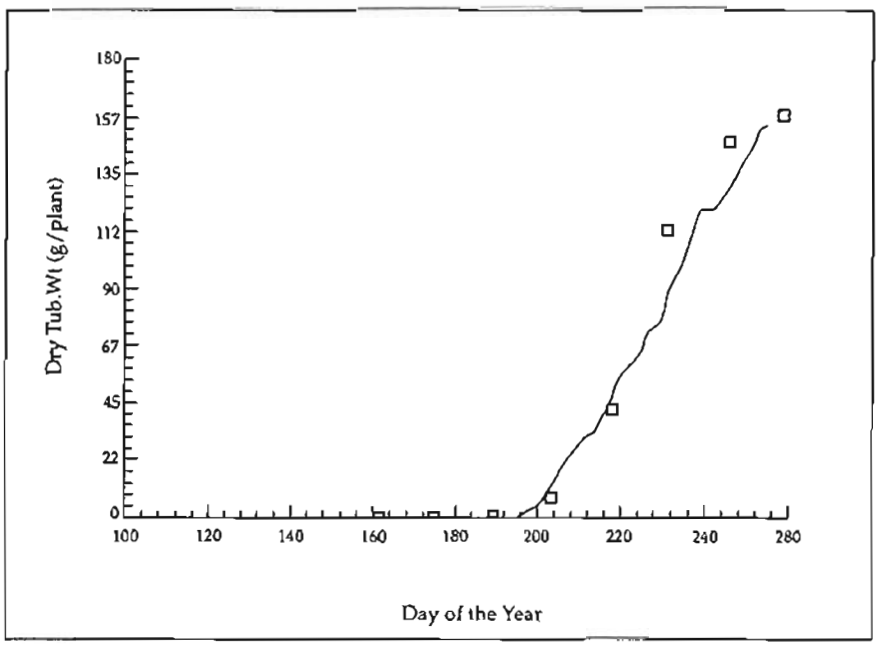


Figure 8C. Simulated (line) versus observed (symbols) seasonal tuber growth for Russet Burbank, North Dakota 1985.

Comparing the performance of our model with previous models for predicting tuber yield is difficult. Validation of many of the previous potato models consisted of seasonal growth within a single growing season or several growing seasons (e.g., Fishman et al., 1984; Ingram and McCloud, 1984; Ng and Loomis, 1984; Ewing et al., 1990). These models validate different approaches to modeling potato growth and development at the empirical or physiological levels, but give little indication of the response of cultivars to changing climate or management. Other models (Ng and Loomis, 1984; MacKerron, 1985; Greenwood et al., 1985a, b) conduct formal sensitivity analyses, making independent changes in key input variables to identify factors controlling growth and development. As discussed at the beginning of this section, we did not conduct a sensitivity analysis because of the diverse validation data set.

Jeffries et al. (1991), in their validation of a model originally developed by MacKerron and Waister (1985), compared simulated versus observed yield for three growing seasons with different levels of soil water stress. Their model accurately ($r^2 = 0.88$) simulated yield response to soil water stress, but was limited to a single cultivar (Maris Piper) at one location. Greenwood et al. (1985b) evaluated the performance of a model of potato growth and N status and found very good agreement between simulated and observed tuber yields. However, their validation data set included yield measurements from serial harvests, which may result in auto correlation from one harvest to the next. All experiments included were conducted on three different soils but under similar climatic conditions. Yield simulation by other CERES-type crop models, like CERES-Maize and SOYGRO, include validation data from different climates and cultivars. The validation of these models was similar to ours for SUBSTOR-Potato.

Simulation of End-of-season Biomass

End-of-season biomass (consisting of tubers, remaining vine biomass, and roots) is the summation of biomass accumulation and biomass loss due to senescence or root turnover. In our model, it is essentially a measurement at any time, which may be after desiccation of vine material. Accurate simulation of this variable is important for biomass and N cycling within the potato crop. We found that our model simulated end-of-season biomass about as well as tuber yield (Figure 9). This is not surprising because most of the biomass remaining at the end of the growing season is in the tubers. Except for two outlying, underestimated biomass yields, total end-of-season biomass tended to be slightly overestimated by the model. The overestimation of haulm biomass discussed earlier, due to the lack of pathogen- or pest-induced defoliation in the model, probably contributes to this overestimation.

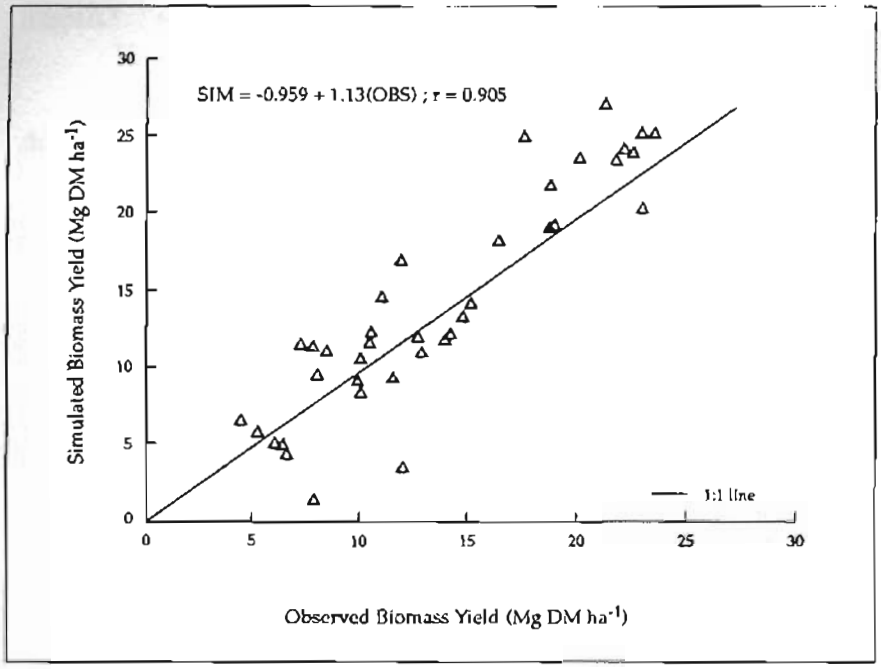


Figure 9. Simulated versus observed end-of-season biomass yield. n = 40.

Conclusions

In this paper, we have discussed the development and initial validation of a new simulation model for potato growth and development. This model simulates the phenological development of the potato crop, including a new approach to incorporating temperature and photoperiod effects on tuber initiation. It also simulates growth using a capacity model for carbon fixation constrained by radiation, high temperatures, nitrogen deficit, and soil water stress.

The performance of this model for numerous cultivars grown in diverse climates was similar to that of other CERES-type crop models, but comparison to previous potato models is difficult. Previous models of potato growth are generally specific to cultivar, environment, or both, and validation in some cases consists of only seasonal growth patterns. The SUBSTOR-Potato model validation presented may be considered preliminary, but the model has great potential for simulating potato growth and evaluating potential changes in management in many regions.

References

Allen, E.J., and R.K. Scott. 1980. An analysis of growth of the potato crop. *J. Agric. Sci.-Camb.* 94:583-606.

Ben Khedher, M., and E.E. Ewing. 1985. Growth analysis of eleven potato cultivars grown in the greenhouse under long photoperiods with and without heat stress. *Amer. Potato J.* 62:537-554.

Carter, J.M., and S.M. Bosma. 1974. Effect of fertilizer and irrigation on nitrate-nitrogen and total nitrogen in potato tubers. *Agron. J.* 66:263-266.

Epstein, E. 1966. Effect of temperature at different growth stages on growth and development of potato plants. *Agron. J.* 58:169-177.

Ewing, E.E. 1981. Heat stress and the tuberization stimulus. *Amer. Potato. J.* 58:31-49.

Ewing, E.E., W.D. Heym, E.J. Batutis, R.G. Snyder, M. Ben Khedher, K.P. Sandlan, and A.D. Turner. 1990. Modifications to the simulation model POTATO for use in New York. *Agric. Syst.* 33:173-192.

Fishman, S., H. Talpaz, M. Dinar, M. Levy, Y. Arazi, Y. Rozman, and S. Varshavsky. 1984. A phenomenological model of dry matter partitioning among potato organs for simulation of potato growth. Agric. syste. 14:159-169.

Greenwood, D.J., J.J. Neeteson, and A. Draycott. 1985a. Response of potatoes to N fertilizer: quantitative relations for components of growth. *Plant Soil* 85:163-183.

Greenwood, D.J., J.J. Neeteson, and A. Draycott. 1985b. Response of potatoes to N fertilizer: dynamic model. *Plant Soil* 85:185-203.

Hammer, G.L., R.L. Vanderlip, G. Gibson, L.J. Wade, R.G. Henzell, D.R. Younger, J. Warren, and A.B. Dale. 1989. Genotype-by-environment interaction in grain sorghum, II, effects of temperature and photoperiod on ontogeny. *Crop Sci.* 29:376-384.

Hartz, T.K., and F.D. Moore III. 1978. Prediction of potato yield using temperature and insolation data. *Amer. Potato J.* 55:431-436.

IBSNAT. 1986. *Technical Report 5: Documentation for IBSNAT crop model input and output files, Version 1.0.* Dept. of Agron. Soil Sci., College of Trop. Agric. and Human Resources, Univ. Hawaii.

Ingram, K.T., and D.E. McCloud. 1984. Simulation of potato crop growth and development. *Crop Sci.* 24:21-27

Iritani, W.M. 1963. The effect of summer temperature in Idaho on yield of Russett Burbank potatoes. *Amer. Potato J.* 40:47-52.

Jeffries, J.A. 1989. Water-stress and leaf growth in field growth of crops of potato (*Solanum tuberosum* L.). *J. Expt. Bot.* 40:1375-1381.

Jeffries, J.A., and D.K.L. MacKerron. 1987. Aspects of the physiological basis of cultivar differences in yield of potato under droughted and irrigated conditions. *Potato Res.* 30:201-207.

Jeffries, J.A., and D.K.L. MacKerron. 1989. Radiation interception and growth of irrigated and droughted potato (*Solanum tuberosum*). *Field Crop Res.* 22:101-112.

Johnson, K.B., S.B. Johnson, and P.S. Teng. 1986. Development of a simple potato growth model for use in crop-pest management. *Agric. Syst.* 19:189-209.

Jones, C.A., and J.R. Kiniry (eds.). 1986. *CERES-Maize: a simulation model of maize growth and development.* Texas A & M Univ. Press, College Station, TX.

Kleinkopf, G.E., D.T. Westermann, and R.B. Dwelle. 1981. Dry matter production and nitrogen utilization by six potato cultivars. *Agron. J.* 73:799-803.

MacKerron, D.K.L. 1985. A simple model of potato growth and yield. Part II. validation and external sensitive. Agric. Forest Meteor. 34:285-300.

MacKerron, D.K.L., and J.A. Jeffries. 1986. The influence of early moisture stress on tuber numbers in potato. *Potato Res.* 29:299-321.

MacKerron, D.K.L., and P.D. Waister. 1985. A simple model of potato growth and yield. Part I. Model development and sensitivity analysis. *Agric. Forest Meteor.* 34:241-252.

Major, D.J., D.R. Johnson, J.W. Tanner, and I.C. Anderson. 1975. Effects of daylength and temperature on soybean development. *Crop Sci.* 15:174-179.

Manrique, L.A., and D.P. Bartholomew. 1991. Growth and yield performance of potato grown at three elevations in Hawaii. II. Dry matter production and efficiency of partitioning. *Crop Sci.* 31:367-372.

Manrique, L.A., and T. Hodges. 1989. Estimation of tuber initiation in potatoes grown in tropical environments based on different methods of computing thermal time. *Amer. Potato. J.* 66:425-436.

Manrique, L.A., J.R. Kiniry, T. Hodges, and D.S. Axness. 1991. Dry matter production and radiation interception of potato. *Crop Sci.* 31:1044-1049.

Marinus, J., and K.B.A. Bodleander. 1975. Response of some potato varieties to temperature. *Potato Res.* 18:189-204.

Moorby, J., and F.L. Milthorpe. 1975. Potato. pp. 225-257 *In* L.J.Evans (ed.), Crop physiology. Cambridge Univ. Press, London.

Munns, R., and C.J. Pearson. 1974. Effect of water deficit on translocation of carbohydrate in *Solanum tuberosum*. *Aust. J. Pl. Phys.* 1:529-537.

Ng, E., and R.S. Loomis. 1984. Simulation of growth and yield of the potato crop. *Simulation Monographs* Pudoc, Wageningen.

Prange, R.K., K.B. McRae, D.J. Midmore, and R. Deng. 1990. Reduction in potato growth at high temperature: role of photosynthesis and dark respiration. *Amer. Potato J.* 67:357-369.

Rasco, E.T. Jr., R.L. Plaisted, and E.E. Ewing. 1980. Photoperiod response and earliness of *S. tuberosum* spp. *Andigena* after six cycles of recurrent selection for adaptation to long days. *Amer. Potato J.* 57:435-448.

Rickman, R.W., and B. Klepper. 1991. Environmentally driven cereal crop growth models. *Ann. Rev. Phytopath.* 29:361-380.

Ritchie, J.T., T.S. Griffin, and B.S. Johnson. 1992. SUBSTOR - a functional model of potato growth, development and yield. *Simulation Monographs* PUDOC, Wageningen, The Netherlands. *In press.*

Russell, W.K., and C.W. Stuber. 1985. Genotype x photoperiod and genotype x temperature interactions for maturity in maize. *Crop Sci.* 25:152-158.

Saffigna, P.G., and D.R. Keeney. 1977. Nitrogen and chloride uptake by irrigated Russet Burbank potatoes. *Agron. J.* 69:258-264.

Sale, P.J.M. 1973. Productivity of vegetable crops in a region of high solar input. II. yields and efficiencies of water use and energy. *Aust. J. Agric. Res.* 24:751-762.

Sands, P.J., C. Hackett, and H.A. Nix. 1979. A model of the development and bulking of potatoes (*Solanum tuberosum* L.). I. derivation from well-managed field crops. *Field Crop Res.* 2:309-331.

Slater, J.W. 1968. The effect of night temperature on tuber initiation of the potato. *Eur. Potato J.* 11:14-22.

Snyder, R.G., and E.E. Ewing. 1989. Interactive effects of temperature, photoperiod, and cultivar on tuberization of potato cuttings. *Hort. Sci.* 24:336-338.

Van der Zaag, D.E., and C.D. Van Loon. 1987. Effect of physiological age on growth vigour of seed potatoes of two cultivars. 5. Review of literature and integration of some experimental results. *Potato Res.* 30:451-472.

Van Loon, C.D. 1981. The effect of water stress on potato growth, development, and yield. *Amer. Potato J.* 58:51-69.

Vos, J., and P.J. Oyarzain. 1987. Photosynthesis and stomatal conductance of potato leaves - effects of leaf age, irradiance, and leaf water potential. *Photosyn. Res.* 11:253-264.

Wheeler, R.M., and T.W. Tibbetts. 1986. Utilization of potatoes for life support systems in space: I. Cultivar-photoperiod interactions. *Amer. Potato J.* 63:315-323.

Yamaguchi, M., H. Timm, and A.R. Spurr. 1964. Effect of soil temperature on growth and nutrition of potato plants and tuberization, composition, and periderm structure of tubers. *Amer. Soc. Hort. Sci. J.* 84:412-423.

DSSAT v4.5 - Canegro Sugarcane Plant Module

Scientific documentation

Singels[1], M. Jones[2] and M. van den Berg[3]

South African Sugarcane Research Institute

Mount Edgecombe, South Africa

[1]*abraham.singels@sugar.org.za*

[2]*matthew.jones@sugar.org.za*

[3]*maurits.vandenberg@sugar.org.za*

International Consortium for Sugarcane Modelling (ICSM)

http://sasri.sasa.org.za/misc/icsm.html

2008

Contents

1. Introduction 5

2. Overview of the model 5

2.1. Phenological phases 6

2.2. Canopy development 7

2.3. Biomass accumulation and partitioning 10

2.4. Root growth 12

2.5. Water stress 13

2.6. Lodging 13

3. Cultivar, ecotype and species parameters 14

4. Differences between the DSSAT4.5 and DSSAT3.5 versions of the Canegro model 19

5. Differences between DSSAT4.5 Canegro and the standalone SASRI Canegro 20

5.1. Potential evapotranspiration 20

5.2. Soil evaporation 21

5.3. Runoff 21

5.4. Root growth 21

5.5. Water stress in topsoil layers 21

5.6. Tiller senescence 21

6. Remaining coding issues that need attention 22

6.1. LAI calculation with the Canesim option for canopy development 22

6.2. Root growth 22

7. Model validation 22

7.1. Introduction 22

7.2. Validation data sets 22

7.3. Results 24

7.4. Discussion 30

7.5. Performance comparison of DSSAT v3.5 and v4.5 versions of Canegro 30

8. Conclusions 33

9. Acknowledgements 33

10. References 33

1. Introduction

The Canegro sugarcane model (Inman-Bamber, 1991, Singels and Bezuidenhout, 2002) simulates sugarcane crop growth and development from daily weather data, cultivar and soil properties, and management input data. It simulates:

- Canopy development at the tiller and leaf level
- Radiation capture from leaf area index
- The water balance using soil-plant-atmosphere continuum principles
- Biomass accumulation following a radiation use efficiency/respiration approach
- Biomass partitioning to different plant components, including stalk sucrose, using a source-sink approach and affected by physiological age, temperature and water stress.

The Canegro model can be regarded as one of the leading sugarcane crop growth models that has been used extensively in research and management. An early Canegro version (Inman-Bamber and Kiker, 1997) was included in version 3.5 (Tsuji *et al*., 1994) of the Decision Support System for Agrotechnology Transfer (DSSAT), Since then, amendments by different research groups resulted in different Canegro versions that were never integrated, nor incorporated into DSSAT. Simultaneously, DSSAT (version 4.0) adopted a modular structure (Jones *et al*., 2002), and many utilities were added.

The International Consortium for Sugarcane Modelling (ICSM), established by leading sugarcane industries and research groups in May 2006, recognised the above-mentioned problem and launched a project to incorporate an up-to-date version of Canegro into DSSAT v4. SASRI-Canegro was taken as the starting point. The project entailed:

- Restructuring of the Canegro code to the DSSAT v4 framework and the generic modules for management, soil, weather and the energy balance.
- Verification of model results against the current Canegro for a predetermined set of simulation runs.
- Sensitivity analysis of key processes such as canopy development, crop water uptake, biomass accumulation and partitioning, to changes in soil, weather, management and variety traits.
- Evaluation of the new DSSAT Canegro model with datasets from Zimbabwe, Australia, USA and South Africa.
- Documentation of code and concepts, as well as of the model evaluation experiments.
- Compiling a user manual.

This document is a description of model concepts and of the initial validation of the DSSAT 4.5 Canegro model. A companion document (Jones & Singels, 2008) provides guidance to the user on the setup, execution and interpretation of DSSAT v4.5 Canegro model runs and the calibration of the model for different cultivars.

2. Overview of the model

A schematic outline of the new modular structure of Canegro, and how it fits within the DSSAT v4 framework is given in Figure 1.1. Each submodule is further explained in subsequent sections.

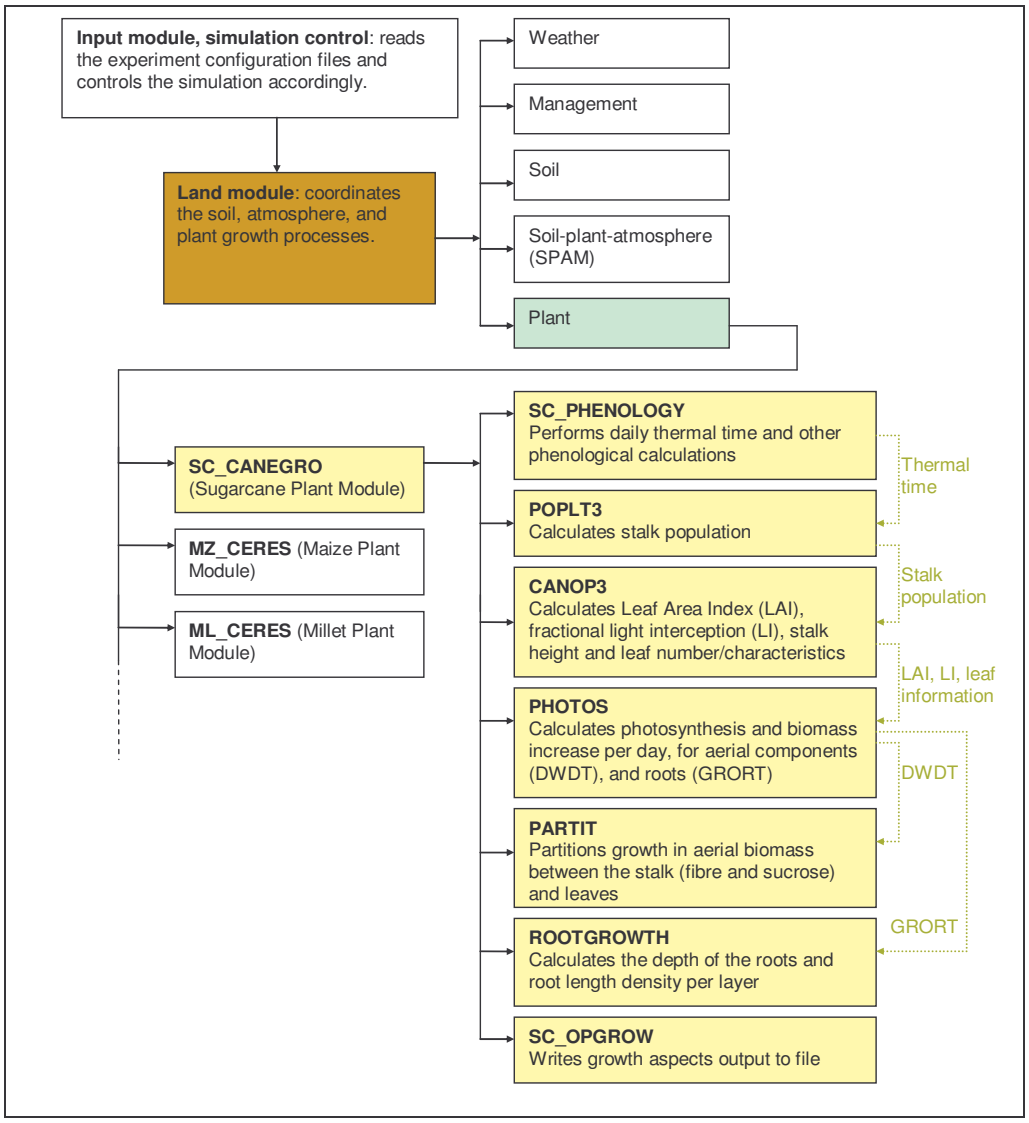


Figure 1.1. Diagrammatic representation of CANEGRO, as a module within the DSSAT Cropping System Model (CSM).

2.1. Phenological phases

The germination phase lasts from planting or ratooning to the emergence of primary tillers. Primary tiller emergence is simulated when a specified period of thermal time has accumulated from planting (TTPLNTEM) or ratooning (TTRATNEM) (Table 3.3). For the standard South African reference cultivar NCo376, these values are 428 and 203 ^{o}Cd respectively using a base temperature of 10 ^{o}C.

Start of stalk elongation is simulated when a cultivar-specific thermal time period (CHUPIBASE in Table 3.3) has elapsed since primary tiller emergence (NCo376 value 1050 ^{o}Cd with an associated base temperature of 10 ^{o}C.

Peak tiller population occurs when a cultivar-specific amount of thermal time has accumulated since emergence (NCo376 value 600 ^{o}Cd, base 10 ^{o}C) (TT_POPGROWTH in Table 3.3).

The tillering phase lasts from the emergence of primary tillers to the occurrence of peak tiller population, while the stalk elongation phase lasts from the start of stalk elongation up to harvest. The tiller

senescence phase commences after peak tiller population is reached and continues until harvest. Flowering is not simulated.

2.2. Canopy development

Two options of simulating the progression of fractional interception of radiation are provided in DSSAT-Canegro, namely (1) a simple thermal time-driven method at whole canopy level (Canesim) and (2) a method using leaf area index derived from thermal time-driven development of individual leaves and tillers (Canegro). These methods are described below.

2.2.1 Canesim canopy model

The Canesim canopy model for unstressed crops is fully described by Singels & Donaldson (2000). The model calculates the interception of radiation as a function of thermal time and row spacing. The coefficient that determines the shape of the canopy curve for sugarcane is the species parameter (Hillpar1 in Table 3.1). The function includes cultivar specific parameters for (1) the thermal time requirement to reach 50 % canopy cover at 1.4 m row spacing (*Tthalfo,* Table 3.3), (2) the base temperature for calculating thermal time and (3) the response in *Tthalfo* to a change in row spacing from the standard 1.4 m.

Canopy cover is reduced by water stress as a function of the duration of severe stress (net relative stress duration) and an ecotype-specific limit on the maximum reduction possible (CS_CNREDUC in Table 3.2, default value: 0.3). Net relative stress duration is the stress duration expressed as fraction of the period required to effect the maximum reduction. The latter is considered an ecotype parameter (CS_CNPERIOD in Table 3.2, default value: 21 days). Stress duration is calculated as the number of days that the growth stress factor (zero to unity) has been below 0.5 (indicating severe stress), minus the number of days that the growth stress factor exceeded 0.5 (indicating conditions favourable for recovery). Canopy cover is reduced when net relative stress duration is increasing from one day to the next and the canopy recovers when the net relative stress duration decreases from one day to the next. The model is partially based on the findings of Smit and Singels (2006).

DSSAT requires values for total leaf area index (TLAI) and green leaf area index (GLAI) to calculate evaporation from the soil and an FAO-56 evaporation coefficient respectively (the ratio between evaporation from fully canopied, unstressed sugarcane and evaporation from a reference crop). In the Canesim canopy model leaf area is not calculated. GLAI was therefore back calculated from Beer's law of radiation extinction in plant canopies.

$$FI = 1 - \exp(-Kc*GLAI)$$

where Kc is the extinction coefficient calculated by the Canegro model. TLAI was set equal to GLAI for the interim. This needs further refinement.

2.2.2 Canegro canopy model

The original model is fully described by Inman-Bamber (1991). It simulates the development of individual leaves and shoots. Shoot leaf area is then scaled up to a canopy level by multiplying leaf area per shoot by the number of shoots per unit area. Interception of photosynthetically active radiation (PAR) is then calculated according to Beer's law, as described in section 2.2.1.

2.2.2.1 Tiller development

Shoot population density is simulated using ecotype parameters POPCF1, POPCF2 and POPDECAY (see Table 3.2) and cultivar parameters MAX_POP and POPTT16 (in Table 3.3). The ecotype

parameters determine the unstressed rate of tiller production and senescence as a function of thermal time. The cultivar parameters determine the peak and mature tiller population.

Row spacing and water stress modify shoot appearance and senescence. Shoot production rate is inversely proportional to row spacing and therefore assumes a direct relationship with bud density in the ground. Shoot density is not allowed to exceed MAX_POP and final shoot density is fixed at POPTT16. Shoot cohorts are phased in (and out) to account for the range in shoot age at a given point in time (see Inman-Bamber, 1991). No distinction is made between plant and ratoon crops apart from the thermal time requirement for emergence of primary shoots (user specified - TTPLNTEM and TTRATNEM in Table 3.3).

2.2.2.2 Leaf development

Leaf emergence

The algorithm for leaf appearance in Canegro is based on the phyllochron concept, using a "broken stick" approach, as suggested by Inman-Bamber (1994a). The phyllochron interval (PI) is defined as the thermal time elapsed (^{o}Cd) between the emergence of subsequent leaves on a tiller, and is regarded as a cultivar-specific parameter. Two phyllochron values (PI1 and PI2, Table 3.3) apply to leaves below and above a cultivar-specific threshold (PSWITCH, Table 3.3) respectively. The concept is illustrated in Figure 2.1. Thermal time is calculated using a process-specific base temperature (TTBASELFEX, an ecotype parameter in Table 3.2, default value 10 ^{o}C).

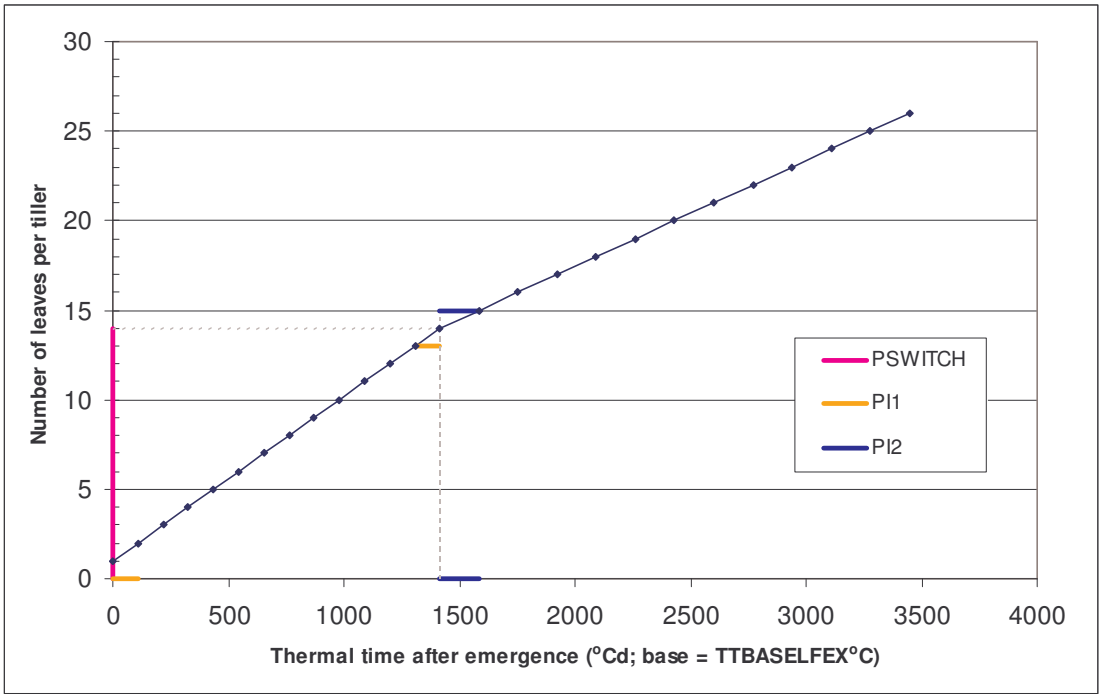


Figure 2.1. Illustration of 'broken stick' approach for leaf appearance as used in Canegro.

Values for PI1 and PI2 for a few South African varieties are indicated in Table 2.1.

Table 2.1 Indicative phyllochron intervals for a few South African varieties (Inman-Bamber, 1994a; Inman-Bamber and Kiker, 1997). The values were determined for a TTBASELFEX value of 10°C and PSWITCH value of 14 leaves.

Cultivar	PI1 (°Cd)	PI2 (°Cd)
NCo376	109	169
N12	118	200
N14	109	169
R570	119	119

2.2.2.3 Leaf senescence

The senescence of green leaves in Canegro is based on the assumption that a healthy, adequately-watered plant cannot have more than a variety-specific number of green leaves (LFMAX in Table 3.2). Once this has occurred, each new leaf formed is accompanied by the senescence of the oldest green leaf on the tiller. The rate of leaf senescence is accelerated under drought stress.

2.2.2.4 Leaf expansion

The basic variable used in Canegro to calculate leaf and plant dimensions is the plant elongation rate (PER, cm/day). PER can be visualised as the elongation rate of the whole shoot, including the longest leaf tip. PER depends on air temperature (Inman-Bamber, 1994b) as well as water availability. The equation used in Canegro is:

PER = SWDF2 * dPERdT * MAX(0., TMEAN-TBASEPER) * 24/10

where dPERdT is unstressed plant extension rate ($mm\ ^{o}C^{-1}\ h^{-1}$) and TBASEPER is the base temperature for plant elongation (^{o}C). Both are regarded as ecotype parameters (See Table 3.2). SWDF2 is the water stress factor for expansive growth (see section 2.5). The term 24/10 is needed to convert the units of PER to cm/day.

Increases in leaf length and leaf width are derived from PER, hence allowing for the calculation of leaf area expansion, which is done for individual leaves on a tiller. Leaves stop expanding once they reach a maximum allowable blade area. This value increases for subsequent leaves (Inman-Bamber and Kiker, 1997) until a specific number of leaves MXLFARNO has formed. After this, the maximum allowable blade area remains at a constant value MXLFAREA.

2.2.2.5 Leaf area index and light interception

The fraction of light intercepted by the crop is determined by the green leaf area index (GLAI). GLAI is calculated as the product of the mean area of green leaves per tiller and the tiller population.

The interception of photosynthetically-active radiation (IPAR) by the canopy is calculated by making use of Beer's law as described in section 2.2.1. The value of the extinction coefficient (Kc) changes during the development of the crop. In Canegro, this is calculated as a function of the total number of leaves per tiller (as determined in the model for primary tillers), as illustrated in Figure 2.2.

This function is determined by three parameters (EXTCFN, EXTCFST, and LFNMXEXT, as explained in Figure 2.2 and Table 3.2)

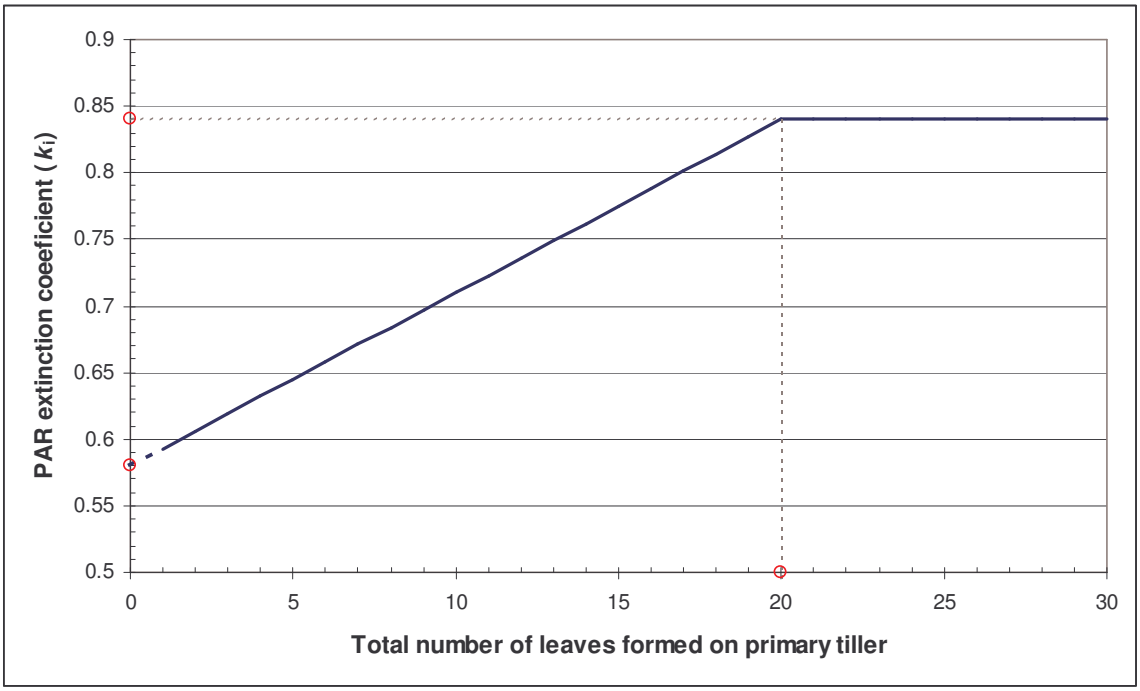


Figure 2.2. Determination of light extinction coefficient (for PAR) in Canegro (according to Inman Bamber, 1991). In this example, LFNMXEXT = 20; EXTCFST = 0.58; and EXTCFN = 0.84.

2.2.2.6 Stalk height and canopy height

Stalk height (SHGT, cm) is assumed to increase at a rate that is proportional to PER:

SHGT = SHGT + (PER * PERcoeff)

Canopy height (CANHEIGHT, cm) is calculated as:

CANHEIGHT = (SHGT + LMAX(LFN(1)) * CHTCoeff) / 100

where LMAX(LFN(1)) is the length of the longest leaf in the first tiller cohort group (cm) and PERcoeff and CHTcoeff are species parameters (Table 3.1).

2.3. Biomass accumulation and partitioning

The simulation of biomass accumulation and partitioning is fully described by Singels & Bezuidenhout (2002). Briefly, the model calculates daily increments in total biomass (dTOT/dt in t/ha/d) using photosynthetically-active radiation (PAR) conversion efficiency (PARCE in g/MJ) according to eq. 1 (derived from Inman-Bamber, 1991). The size of biomass increments depends on the amount of intercepted PAR (IPAR in MJ/ha), the size of the crop (TOT in t/ha) and the level of crop water stress (expressed by the SWDF1 factor as described by Ritchie et. al., 1986).

$$dTOT/dt = (1 - RespGcf)(PARCE.10^{-6} . IPAR - Rm .TOT) SWDF1$$

where RespGcf is the coefficient for growth respiration (considered a species parameter of 0.242 t/t; see Table 3.1) and Rm is the maintenance respiration rate.

PAR conversion efficiency and maintenance respiration depend on temperature, as described by Singels *et al.* (2005):

PARCE = PARCEmax (1 – exp (-0.08 (Tmean – Tbasephotos))

where PARCEmax is the theoretical maximum PAR-to-biomass conversion efficiency at an optimal temperature (considered a cultivar parameter, listed in Table 3.3) and Tbasephotos is the base temperature for photosynthesis (7^{o}C; considered a species parameter, listed in Table 3.1)

The fraction of biomass increments consumed by maintenance respiration (Rm) is calculated as an exponential function of temperature:

Rm = RespQ10 **((Tmean-10.0)/ 10.0) * Respcon

where RespQ10 is the fractional increase in respiration rate per 10^{o}C rise in temperature (Q10 coefficient, considered as a species parameter, see Table 3.1), and Respcon is the value of Rm at the reference temperature of 10^{o}C (species parameter). This equation was derived from results reported by Liu and Bull (2001).

Daily partitioning of assimilate between roots and aerial parts is simulated as a non-linear function of total biomass. A large fraction (Max_rootpf = 0.95, see Table 3.1) is partitioned to roots early in the life of the plant but this decreases rapidly as the plant ages. The maximum fraction of daily biomass increments that is partitioned to aerial parts (in a mature crop) is specified in the cultivar file (NCo376 value = 0.88, APFMX in Table 3.3).

A temperature-dependent fraction of aerial dry mass is partitioned to stalk (see Singels *et al.*, 2005) and is limited to a cultivar-specific maximum value (NCo376 value: 0.65, STKPFMAX in Table 3.3) when thermal time since emergence exceeds a cultivar-specific threshold (Nco376 value = 1050^{o}Cd, CHUPIBASE in Table 3.3).

The rate of dry matter partitioning to stalk is regarded as the source strength. Partitioning of stalk dry matter between sucrose and stalk structure is regulated by sink capacity for stalk structural growth and the source-to-sink ratio. Sink capacities for structural growth and sucrose storage are dictated by current growing conditions (temperature, water status), current stalk mass and cultivar characteristics (SUCA, TBFT in Table 3.3). The sucrose accumulation component of the model is based on a framework of sucrose distribution within stalks as it is affected by temperature and water stress (see Figure 2.3).

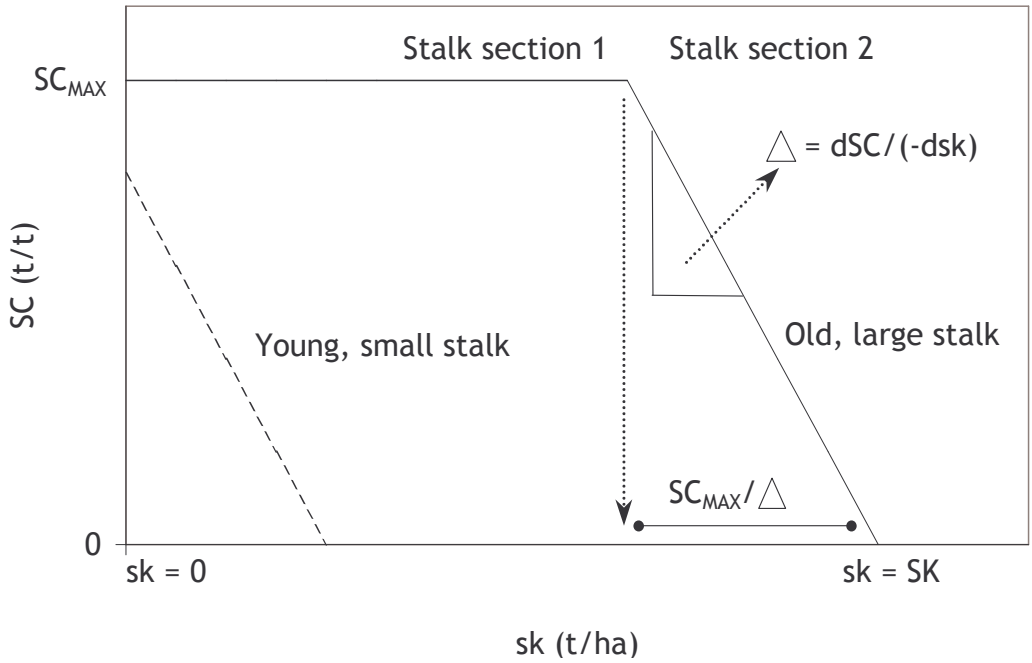


Figure 2.3. Schematic diagram of sucrose distribution within a single big stalk (SC - sucrose content, sk - cumulative stalk mass measured from the base (sk=0) to the top (sk=SK)) (from Singels & Bezuidenhout, 2002).

Sucrose accumulation response to temperature is determined by the FTCON species parameter (Table 3.1) and the TBFT cultivar parameter (Table 3.3) ; the higher the value of FTCON, the higher the sucrose partitioning response to temperature. High sucrose cultivars will tend to have higher TBFT values, reflecting their ability to partition more assimilate to sucrose at any given temperature. Sucrose partitioning response to water stress is captured in the SWDF2AMP ecotype parameter (Table 3.2).

2.4. Root growth

Root growth is expressed in terms of the extension of the rooting depth, as well as the increase of root mass and root length per soil layer. The latter, together with soil water status, determine the potential crop water uptake used to calculate the level of water stress (see section 2.5).

Root length density (RLV) is increased by converting daily biomass increments partitioned to roots (GRORT) to total root length per unit ground area (RLNEW, cm/cm^2), assuming a constant specific root length of RTcmpg (species parameter in Table 3.1, in cm/g):

$$RLNEW = GRORT * 0.0001 * RTcmpg$$

RLNEW is then apportioned to the rooted soil layers, according to the weight factors WR indicated in the soil file. It is assumed that 0.5 % of root length density (RLV) is lost each day (due to respiration and/or senescence). The value of RLV for a given layer is not allowed to be lower than the reference root length density (RLVo) for that layer. For plant crops, RLVo is initialised to a default value of 0.02 cm/cm^3 (RLVmin, a species parameter in Table 3.1) for all soil layers, while for ratoon crops it is set to 0.2 cm/cm^3 for the top soil layer and is reduced as a function of depth for deeper layers:

$$RLVo_i = 0.2 - 0.005 * [Cumdepth\ (i)]$$

where Cumdepth is the depth from the surface to the base of soil layer i (cm)

Daily potential increase in rooting depth is calculated assuming a 2.2 mm penetration per unit thermal time (base 16) for that day, until the maximum rooting depth is reached. Actual rooting depth increase is reduced below the potential when water stress occurs. Rooting depth is initialized at zero for a plant and "stand-alone" ratoon crop, while initial value for a ratoon crop following a preceding crop, is set to the final rooting depth of the preceding crop.

2.5. Water stress

The Canegro sugarcane model simulates the impact of water stress by regulating different plant processes (carbon assimilation, expansive growth) by different water stress factors ($SWDF_i$) according to an approach proposed by Singels & Bezuidenhout (2002). These stress factors, which represent the rate of the different processes (i) relative to the unstressed rate, are determined by potential water supply and demand using the approach of the CERES model (Jones & Kiniry, 1986) used in DSSAT 4.5 and shown below:

$$SWDF_i = f_i . WS_p/T_{max} \text{ [bound by } 0.0 \leq SWDF_i \leq 1.0]$$

where f_i is a process-specific parameter, usually with a value between 0.0 and 1.0; WS_p is potential root water uptake from the soil (i.e. potential supply) and T_{max} is maximum transpiration (i.e. demand). Currently in Canegro, f_2=0.5 (the inverse of the value of the RWUEP2 species parameter in Table 3.1) for the calculation of the water stress impact on expansive growth and f_1=1.0 (Inverse of RWUEP1 = 1.) for the calculation of the impact on carbon assimilation. This implies that expansive growth rate is reduced when WS_p drops below 2 times $T_{max.}$,and carbon assimilation is reduced when WS_p drops below T_{max}. It is assumed that once WS_p/T_{max} drops below Critsw (Species parameter in Table 3.1, value of 0.2 for NCo376) , photosynthesis can only gradually recover from water stress. The thermal time taken for full recovery is specified by HuRecover (specie parameter in Table 3.1, value of 150 ^oCd for NCo376.

T_{max} is calculated from the atmospheric evaporative demand (based on the FAO-56 short grass reference evaporation) modified by a crop coefficient (EORATIO species parameter with a value of 1.15, Table 3.1) and the canopy cover coefficient. The latter is calculated from leaf area index following Beer's law. .

Actual water uptake is calculated in the DSSAT water balance module following the CERES maize approach (Jones & Kiniry, 1986). The RWUMX species parameter, (Table 3.1) limits daily water transport by the sugarcane root system to a maximum value of 0.07 cm^3/cm

2.6. Lodging

The model simulates partial or full lodging of cane stalks when the mass of above-ground plant parts (fresh mass) plus the water retained on it, exceeds a cultivar-specific threshold (LG_AMBASE, the aerial mass at which lodging commences when other lodging factors such as water and wind are absent, see Table 3.3). The extent of lodging is proportional to the magnitude of the extent to which the threshold is exceeded. For NCo376, simulated lodging commences as soon as the above ground fresh plant plus water mass exceeds 220 t/ha and is completely lodged at 250 t/ha. Lodging thresholds are reduced by 7.5 t/ha when daily windrun exceeds 200 km/d or when runoff occurred, indicating a saturated topsoil. The effects of these two factors on lodging are additive. Lodging is simulated as an incremental process as each event occurs. Note: Mass of intercepted rainfall not calculated in DSSAT-Canegro v4.5

The model simulates the impact of lodged cane on the interception of radiation and on photosynthetic efficiency. These processes are reduced by 10 and 28% respectively for fully lodged cane (according to Singh *et al.*, 1999). Partially lodged cane has a proportional impact. The simulated impacts can be altered by adjusting parameters in Table 3.2 (LG_GP_REDUC, LG_FI_REDUC).

3. Cultivar, ecotype and species parameters

Although a validation workshop held in August 2007 provided some clarification regarding plant parameters, some uncertainty remains about the allocation of parameters to the species, ecotype and cultivar categories due to a lack of knowledge. The approach was to make more, rather than fewer, parameters available for adjustment by the end user and therefore the ecotype (Table 3.2) and cultivar (Table 3.3) files contain large numbers of parameters, compared to other crops. Cultivar parameters relate mainly to biomass partitioning, canopy (leaf and tiller) development and phenological phasing and these are expected to vary between cultivars. Ecotype parameters are expected to vary less with cultivars and are more difficult to adjust because data would not be readily available. Parameters for different tillering types are contained here. Species parameters (see Table 3.1) relate to photosynthesis, respiration, biomass partitioning, root growth, plant response to water stress and lodging.

Table 3.1. Species parameters for sugarcane.

Name	Value	Category	Description	Reference
Tbasephotos	7.0	Photosynthesis	Base temperature (^{o}C) for photosynthesis	Singels *et al.*, 2005
Critsw	0.2	Photosynthesis	Water stress threshold for prolonged impact from severe water stress on photosynthesis	
HuRecover	150	Photosynthesis	Thermal time required for photosynthesis to recover fully after a severe water stress event (^{o}Cd)	
RespQ10	1.68	Photosynthesis	Fractional increase in respiration rate per 10 ^{o}C rise in temperature (Q10 coefficient)	Inman-Bamber, 1991
RespGcf	0.242	Photosynthesis	Fraction of gross photosynthesis lost to growth respiration	Inman-Bamber, 1991
PCB	0.6	Biomass partitioning	Partitioning coefficient: extinction coefficient of fraction of dry mass increments allocated to above ground biomass	Parameter 'B' in Singels & Bezuidenhout, 2002
Max_rootpf	0.95	Biomass partitioning	Maximum partition fraction of daily mass increments to roots	
FTCON	0.32	Sucrose accumulation	Temperature response shape parameter	Singels & Bezuidenhout, 2002
SURCON	0.99	Sucrose accumulation	Sucrose partitioning parameter that determines the response time of shifts in partitioning between sucrose and fibre in the stalk due to environmental changes (varies between 0 and 1)	
RTcmpg	500	Root growth	Root length per mass of roots (cm/g)	
Wrk	-0.01	Root growth	Root length density extinction coefficient by depth	
RLVmin	0.02	Root growth	Minimum root length density in soil layers (cm/cm^3)	
SenesF	5	Canopy	Number of leaves per shoot senesced per 100 stress days	
Reset	5	Canopy	Rainfall required to reset stress day counter (mm)	
Percoeff	0.16	Canopy	Fraction of plant elongation attributable to stalk elongation	Inman-Bamber, 1991
CHTCoeff	0.864	Canopy - height	Coefficient determining canopy height as a function of stalk height and number of leaves (cm/cm)	
Hillpar1	2.453	Canesim canopy	Empirical function shape parameter	Singels & Donaldson (2000)
EORATIO	1.15	Water balance	Ratio of potential ET from fully canopied unstressed sugarcane canopy to grass reference ET (Kc from FAO-56)	Allen *et al.* (1998)
RWUEP1	1	Water balance	Soil water supply/potential evaporation ratio threshold below which evaporation and photosynthesis are limited	Jones & Kiniry (1986)
RWUEP2	2	Water balance	Soil water supply/potential evaporation ratio threshold below which expansive growth is limited	Jones & Kiniry (1986)
RWUMX	0.07	Water uptake	Maximum root water uptake per unit length of root (cm^3 water/cm RLV)	Jones & Kiniry (1986), van Antwerpen (1998)
LG_RATING	8	Lodging	Lodging score when crop is fully lodged	Singels 2007 (Pers comm.)
LG_CRIT_WIND	200	Lodging	Wind run (km/d) threshold for lodging	

Table 3.2. Ecotype parameters. Parameter names shown in italics are optional depending on choice of canopy model.

Name	Value	Category	Description	Reference	Guide for determination
DELTTMAX	0.07	Sucrose accumulation	Max. change in sucrose content per unit change in stalk mass in the unripenened section of the stalk (/t)	'DELTAMAX' in Singels & Bezuidenhout, 2002	Range between .06 and 0.08. High sucrose varieties will have high value. Calibrate against irrigated seasonal sucrose curve
SWDF2AMP	0.5	Sucrose accumulation	Sucrose partitioning sensitivity to water stress parameter	'FWCON' in Singels & Bezuidenhout, 2002	
CS_CNREDUC	0.3	Canopy - CANESIM	Maximum fractional canopy reduction due to water stress		
CS_CNPERIOD	21	Canopy - CANESIM	Water stress period required to effect maximum canopy reduction (days)		
Tthalfa	125	Canopy - CANESIM	Half canopy thermal time adjustment for row spacing (^{o}Cd/m)		
dPERdT	0.176	Canopy - height	Change in plant extension rate (mm/h) per unit change in effective temperature (^{o}C)	Originally from Inman-Bamber (1991), since adjusted	
EXTCFN	0.84	Canopy - light extinction	Maximum canopy light extinction coefficient	Inman-Bamber (1991)	See Figure 2.2
EXTCFST	0.58	Canopy - light extinction	Minimum canopy light extinction coefficient	Inman-Bamber (1991)	See Figure 2.2
LFNMXEXT	20	Canopy - light extinction	Leaf number (including dead leaves still attached) at which maximum light extinction occurs		
AREAMX_CF(1)	0	Canopy - leaves	Cultivar parameter for quadratic equation defining maximum leaf area	Inman-Bamber (1991)	Regress quadratic equation to leaf area (cm^2) vs leaf no data: $y= cf_1+cf_2*x+cf_3*x^2$
AREAMX_CF(2)	27.2	Canopy - leaves	Cultivar parameter for quadratic equation defining maximum leaf area	Inman-Bamber (1991)	
AREAMX_CF(3)	-20.8	Canopy - leaves	Cultivar parameter for quadratic equation defining maximum leaf area	Inman-Bamber (1991)	
WIDCOR	1	Canopy - leaves	Parameter affecting the width of leaves	Inman-Bamber (1991)	Used to determine area of expanding leaves
WMAX_CF(1)	-0.0345	Canopy - leaves	Cultivar parameter for quadratic equation defining max leaf width per leaf number	Inman-Bamber (1991)	Used to determine area of expanding leaves
WMAX_CF(2)	2.243	Canopy - leaves	Cultivar parameter for quadratic equation defining max leaf width per leaf number	Inman-Bamber (1991)	Used to determine area of expanding leaves
WMAX_CF(3)	7.75	Canopy - leaves	Cultivar parameter for quadratic equation defining max leaf width per leaf number	Inman-Bamber (1991)	Used to determine area of expanding leaves

Table 3.2. Ecotype parameters (continued)

Name	Value	Category	Description	Reference	Guide for determination
LMAX_CF(1)	-0.376	Canopy – leaves	Parameter for quadratic equation defining max leaf length per leaf number		
LMAX_CF(2)	12.2	Canopy – leaves	Parameter for quadratic equation defining max leaf length per leaf number		
LMAX_CF(3)	21.8	Canopy – leaves	Parameter for quadratic equation defining max leaf length per leaf number		
MAXLFLENGTH	100	Canopy – leaves	Absolute max leaf length (overrides LMAX_CF-calculated values)		
MAXLFWIDTH	3.5	Canopy – leaves	Absolute max leaf width (overrides LMAX_CF-calculated values)		
POPCF(1)	1.826	Tiller population	Stalk population coefficient, in ideal conditions (no stress), as function of thermal time		
POPCF(2)	-0.00201	Tiller population	Stalk population coefficient, in ideal conditions (no stress), as function of thermal time		
POPDECAY	0.004	Tiller population	Fraction of tillers above the future mature tiller population (at a thermal time of 1600 ^{o}Cd), that senesce per unit thermal time		Adjust by comparing simulated and observed tiller population. Varies from 0.003 to 0.005
TTBASEEM	10	Phenology	Base temperature (^{o}C) for emergence and start of stalk elongation		
TTBASELFEX	10	Phenology	Base temperature for leaf phenology (^{o}C)		
TTBASEPOP	16	Phenology	Base temperature for stalk phenology (^{o}C)		
TBASEPER	10.57	Phenology	Base temperature for plant extension (^{o}C)		
LG_AMRANGE	30	Lodging	Range in aerial mass from the start to the end of lodging (t/ha)		
LG_GP_REDUC	0.28	Lodging	Reduction in gross photosynthesis due to full lodging, as a fraction	Singh *et al.*, (1999)	
LG_FI_REDUC	0.1	Lodging	Reduction in fractional interception by the canopy due to full lodging		

Table 3.3. Cultivar parameters. Parameter names shown in italics are optional depending on choice of canopy model.

Name	Value	Category	Description	Reference	Rough guide for determination
PARCEmax	9.9	Biomass accumulation	Maximum (no stress) radiation conversion efficiency expressed as assimilate produced before respiration, per unit PAR. (g/MJ)	Singels *et al.*, 2005	
APFMX	0.88	Biomass accumulation	Maximum fraction of dry mass increments that can be allocated to aerial dry mass (t/t)	'ADMPFmax' in Singels & Bezuidenhout, 2002	
STKPFMAX	0.65	Biomass partitioning	Fraction of daily aerial dry mass increments partitioned to stalk at high temperatures in a mature crop (t/t on a dry mass basis)	Singels *et al.*, 2005 derived from Liu and Bull (2001)	
SUCA	0.58	Sucrose accumulation	Sucrose partitioning parameter: Maximum sucrose contents in the base of stalk (t/t)	'Scmax' in Singels & Bezuidenhout, 2002	
TBFT	25	Sucrose accumulation	Sucrose partitioning: Temperature at which partitioning of unstressed stalk mass increments to sucrose is 50% of the maximum value (^{o}C)	"T50" in Singels & Bezuidenhout, 2002	Range between 22 and 28. Early season high sucrose have high values. Calibrate against irrigated seasonal sucrose curve
Tthalfo	250	Canopy - CANESIM	Thermal time to half canopy (^{o}Cd)		Range between 200 and 300. Quick canopy variety will have low value. Calibrate against multiple data sets of canopy cover *vs* thermal time
Tbase	16	Canopy - CANESIM	Base temperature for canopy development (^{o}Cd)		Ranges between 14 and 18. Quick canopy variety will have low value. Calibrate against multiple data sets of canopy cover *vs* thermal time
LFMAX	12	Canopy – leaves	Maximum number of green leaves a healthy, adequately-watered plant will have after it is old enough to lose some leaves		
MXLFAREA	360	Canopy – leaves	Max leaf area assigned to all leaves above leaf number MXLFARNO (cm^2)	Inman-Bamber (1991)	Size of biggest leaf on primary stalk
MXLFARNO	14	Canopy – leaves	Leaf number above which leaf area is limited to MXLFAREA	Inman-Bamber (1991)	Leaf no of biggest leaf on primary stalk

Table 3.3. Cultivar parameters (continued)

Name	Value	Category	Description	Reference	Rough guide for determination
PI1	69	Leaf phenology	Phyllocron interval 1 (for leaf numbers below Pswitch, °C.d (base TTBASELFEX))	Inman-Bamber (1991)	Compare simulated and observed leaf number over thermal time
PI2	169	Leaf phenology	Phyllocron interval 2 (for leaf numbers above Pswitch, °C.d (base TTBASELFEX))	Inman-Bamber (1991)	Compare simulated and observed leaf number over thermal time
PSWITCH	18	Leaf phenology	Leaf number at which the phyllocron changes.	Inman-Bamber (1991)	Compare simulated and observed leaf number over thermal time
MAX_POP	30	Tiller phenology	Maximum tiller population (stalks/m2)		Obvious. Ranges from 20 to 80
POPTT16	13.3	Tiller phenology	Stalk population at/after 1600 degree days (/m2)		Obvious. Ranges from 7 to 15
TTPLNTEM	428	Phenology	Thermal time to emergence for a plant crop (degree C days, base TTBASEEM)		Obvious
TTRATNEM	203	Phenology	Thermal time to emergence for a ratoon crop (degree C days, base TTBASEEM)		Obvious
CHUPIBASE	1050	Phenology	Thermal time (baseTTBASEEM) from emergence to start of stalk growth	'TTskp' in Singels & Bezuidenhout, 2002	Obvious
TT_POPGROWTH	600	Phenology	Thermal time from emergence to peak tiller population (°Cd, base TTBASEPOP)		Obvious. Ranges from 400 to 800.
LG_AMBASE	220	Lodging	Aerial mass (fresh mass of stalks, leaves, and water attached to them) at which lodging starts; t/ha		Ranges from 180 to 300

4. Differences between the DSSAT4.5 and DSSAT3.5 versions of the Canegro model

In effect, the DSSAT v4.5 version of Canegro differs from the DSSAT v3.5 version primarily with regard to the calculation of biomass accumulation and partitioning.

The dependence of photosynthesis and growth respiration on temperature is now accounted for (following Liu & Bull, 2001). The equations used to simulate these relationships are fully described by Singels *et al*. (2005). The DSSAT v3.5 version assumed (1) a constant photosynthetically-active radiation use efficiency, independent of temperature, and (2) growth respiration rate as a constant proportion of photosynthesis rate after subtraction of maintenance respiration, also regardless of temperature.

In the DSSAT v3.5 version of Canegro, root and stalk mass were simulated as fractions of total dry matter and aerial dry matter respectively. Partitioning of daily increments was not simulated. Sucrose content was determined empirically from crop age, time of year and irrigation (fully described by O'Leary, 2000) and applied to only one cultivar (NCo376). The approach was based on states rather than rates and was therefore incapable of simulating the changes in partitioning ratios that occur in response to environmental changes, or to cultivar characteristics.

The DSSAT v4.5 version of Canegro simulates daily partitioning of biomass increments to roots and stalk as functions of total biomass. It simulates the partitioning of stalk mass increments to stored

sucrose and stalk structure using a source-sink approach. It also accounts for genotypic variation by allowing user adjustment of cultivar traits related to partitioning. See section 2.3 and Singels & Bezuidenhout (2002) for a more detailed description.

Biomass accumulation and partitioning of the DSSAT v4.5 version of the model is therefore more responsive to environmental and genotypic variation that that of the DSSAT v3.5 version. The model should provide more robust predictions of cane stalk and sucrose yields for situations where it has not been calibrated or applied before.

A slightly different method of calculating tiller senescence has also been introduced in the DSSAT v4.5 version. Originally, the model by Inman-Bamber (1991) used three empirical parameters to simulate the senescence of tillers after peak population has been reached. A new, more robust, method of simulating tiller senescence was introduced which uses only one parameter (POPDECAY). POPDECAY is defined as the tiller senescence rate per unit thermal time, expressed as the fraction of tillers above the future mature tiller population (at a thermal time of 1600 ^{o}Cd).

In terms of model implementation, the DSSAT v4.5 version of Canegro differs substantially from the DSSAT v3.5 version. The latter is a standalone model, while the DSSAT v4.5 version uses a modular approach (Cropping System Model (CSM) – Jones *et al*. (2003)), so only the plant growth and development aspects of the current SASRI Canegro model are reflected in the DSSAT v4.5 Canegro. All other model aspects are handled by common (to all crops in the DSSAT CSM) modules. This modular approach is beneficial in a number of ways, at the occasional cost of having to use common routines where Canegro-specific ones have been developed and are possibly more appropriate (see section 5 below).

5. Differences between DSSAT4.5 Canegro and the standalone SASRI Canegro

5.1. Potential evapotranspiration

The SASRI Canegro model uses a modified Penmon-Monteith algorithm (McGlinchey & Inman-Bamber, 1996) to calculate sugarcane reference evaporation directly, as opposed to the more conventional way of calculating grass reference evaporation and then applying a crop coefficient as prescribed by the FAO-56 method (Allen et. al., 1998).

The latter method is one of the options provided in DSSAT v4.5. In DSSAT v4.5, this option uses dew point as the measure of humidity, regardless of whether dew point or relative humidity is provided or not. If dew point temperature is not provided, DSSAT assumes that dew point temperature (Tdew) equals minimum temperature (Tmin), an assumption that may produce questionable estimates of reference evaporation for arid and semi-arid climates.

During our implementation of the DSSAT v4.5 Canegro model, the DSSAT CSM weather input module was modified such that it is now assumed that if a single value of humidity is provided in the weather input file, it represents maximum daily humidity, which is also assumed to occur at the coldest (Tmin) time of the day (DSSAT only allows for a single input relative humidity value per day). If dew point temperature is not provided as input, it is calculated using these humidity and temperature values following a method described by Campbell & Norman (1998).

DSSAT assumes that Tdew is equal to Tmin when relative humidity is not provided in the weather input file. In such a case, the FAO-56 method for calculating grass reference evaporation can be inaccurate when this is assumption is not valid. The Priestley-Taylor method can be specified in these situations. However, it is highly recommended that maximum relative humidity (and windspeed), or pre-determined Tdew, be included in the weather data input and that the FAO-56 method be specified.

In DSSAT v4.5, the grass reference calculated by the FAO-56 (or Priestley-Taylor) method is modified by a factor that is derived from sugarcane leaf area index (LAI) and a crop coefficient specified in the

species file (EORATIO). Sugarcane potential evapotranspiration will increase by this factor as LAI increases from 0 to 6. EORATIO was calibrated by comparing evapotranspiration for ten well-watered crops simulated by the DSSAT v4.5 and SASRI (standalone) versions of Canegro.

5.2. Soil evaporation

Potential soil evaporation (EOS) is now calculated identically in the two models. EOS in DSSAT was originally calculated as a function of healthy (green) leaf area index (XHLAI). This is not a problem in most crops, as senesced leaves usually fall off the plant. With sugarcane, however, the leaf area of dead leaves attached to the stalk affects light transmission through the canopy and will therefore impact on potential soil evaporation. Hence, a new measure of leaf area index was introduced into the DSSAT CSM, namely total LAI (green plus dead). This variable is used to calculate potential soil evaporation, while healthy LAI is still maintained for calculating potential transpiration.

5.3. Runoff

SASRI Canegro uses a variable runoff curve number approach to model runoff (Schmidt & Schulze, 1987), whereas DSSAT4.5 uses a constant runoff curve number specified in the soil input file. When the curve number in DSSAT4.5 was set to the average value calculated by SASRI Canegro, minimal differences in runoff between the models were noted.

5.4. Root growth

An inconsistency in the root partitioning code in SASRI Canegro was identified and fixed. For a plant crop, root length density (RLV) is initialised in SASRI Canegro to 0.02 cm/cm^3 in all layers in the soil profile. For a ratoon crop, RLV is initialised using an exponential function (decreasing with depth), resulting in values of RLV close to zero for layers below a depth of 40 cm. In the DSSAT 4.5 Canegro this function was kept, but RLV values of below 0.02 cm/cm^3 were set equal to 0.02 cm/cm^3, for consistency. This did not affect aboveground partitioning, because no mass balance was in place. The effect of this change was to reduce overall biomass increase allocated to root system, particularly later in the season. It seems to have had little effect on model output.

5.5. Water stress in topsoil layers

Water stress is calculated in the top 30 cm of the soil profile for modelling stress effects on tiller population. In SASRI Canegro, only layers that are fully included in the top 30 cm are taken into account when calculating water stress, resulting in unintended variation in water stress calculation when compared to a soil with a different layer configuration. In DSSAT, this problem does not occur because the set soil layer configuration fits neatly into 30 cm, but the algorithm was corrected anyway, taking account of partial layers if necessary.

5.6. Tiller senescence

Originally, the model by Inman-Bamber used three empirical parameters to simulate the senescence of tillers after peak population has been reached. A new more robust method of simulating tiller senescence was introduced which uses only one parameter (POPDECAY). POPDECAY is the defined as the tiller senescence rate per unit thermal time, expressed as the fraction of tillers above the future mature tiller population (at a thermal time of 1600 ^{o}Cd).

6. Remaining coding issues that need attention

6.1. LAI calculation with the Canesim option for canopy development

The back calculation of leaf area index from fractional interception (FI) as simulated with the Canesim canopy option (see section 2.2.1) needs further refinement. The method estimates green LAI (GLAI). Total LAI (TLAI) was set equal to GLAI for the interim. This will have an impact on soil evaporation and transpiration, as TLAI is used to determine potential soil evaporation and the crop coefficient that related relates potential sugarcane transpiration to potential transpiration from a grass crop. Soil evaporation is expected to be slightly higher than that simulated with the Canegro canopy option, while transpiration is expected to be slightly lower. This will be addressed in a future release by estimating TLAI from GLAI and total and dead leaf number.

6.2. Root growth

We believe that in the model, roots penetrate the soil (in depth) far too quickly in plant crops. This has a limited effect because root length density (RLV)) increases at a realistic rate and water uptake is affected much more by RLV than by rooting depth. While the RLV error has been corrected (see 5.4), the root depth coding error has not yet been corrected in DSSAT4.5.

7. Model validation

7.1. Introduction

DSSAT-Canegro behaves differently to the stand-alone Canegro as a result of differences in non-plant routines, such as the water balance, so it is important to establish the usefulness of the new model by evaluating its performance against measured data.

Validation involves the comparison of simulated and measured data for certain variables. The differences are described statistically. This provides a measure of the accuracy of the model, which feeds into the decision making process where the model is applied. It should be noted that these validations were performed with no calibration of model parameters (except the reference evaporation crop coefficient, EORATIO) since the incorporation of the Canegro model into DSSAT (i.e. parameter values are the same as a those used in SASRI Canegro). It is possible that calibration of species/cultivar parameters may be required as a result of interactions between the plant growth and development routines and (different to Canegro) features of the DSSAT water balance.

The statistical parameters used for quantifying model performance are:

- the coefficient of determination (R^2),
- slope and intercept of the linear regression between simulated and observed values,
- root mean square error (RMSE). The error in RMSE refers to the difference between the simulated and observed value.
- the average difference between simulated and observed values (APE).

7.2. Validation data sets

Two South African experiments have been chosen, namely (1) a dryland experiment conducted at La Mercy, South Africa (described by Inman-Bamber, 1994a, b) and (2) an irrigated experiment conducted in Pongola, South Africa (described by Rostron, 1972 and Inman-Bamber, 1994b).

7.2.1 La Mercy experiment

The Agrowth7 trial (dryland) was conducted at La Mercy (29^o 34'S, 30^o 8'E) in coastal Kwazulu-Natal, South Africa. Cultivar NCo376 was ratooned and harvested at 8 different dates (Table 7.1) during the

period from 1989 to 1991. Row spacing was 1.2 m. (DSSAT4.5 experiment name - SATO8902). Soil profile information is listed in Table 7.2. It should be noted that the soil was somewhat difficult to describe, and can be considered a fairly 'extreme' soil in terms of clay content. Initial soil moisture conditions were assumed to be 50% of available water content.

Table 7.1. Trial treatments

Treatment	Crop type	Start	Harvest	Harvest Age (days)
1	Ratoon	01 Jun 1989	02 Oct 1990	488
2	Ratoon	01 Aug 1989	05 Dec 1990	491
3	Ratoon	01 Oct 1989	05 Feb 1991	492
4	Ratoon	01 Dec 1989	03 Apr 1991	488
5	Ratoon	01 Feb 1990	04 Jun 1991	488
6	Ratoon	01 Apr 1990	31 Jul 1991	486
7	Ratoon	01 Jun 1990	01 Oct 1991	487
8	Ratoon	01 Aug 1990	03 Dec 1991	489

Table 7.2. Soil profile information

Layer thickness	Lower limit (cm^3/cm^3)	Drained upper limit (cm^3/cm^3)	Saturated water content (cm^3/cm^3)	Rooting weight	Bulk density (g/cm^3)	Saturated water conductivity cm/h
5	0.102	0.255	0.387	1.0	1.3	0.8
10	0.102	0.255	0.329	0.8	1.61	0.8
15	0.102	0.237	0.316	0.7	1.63	0.8
15	0.131	0.228	0.319	0.6	1.59	0.8
15	0.132	0.238	0.345	0.5	1.48	0.7
15	0.142	0.258	0.359	0.45	1.46	0.6
15	0.221	0.329	0.390	0.40	1.48	0.5
15	0.307	0.349	0.385	0.37	1.56	0.5
15	0.346	0.375	0.391	0.35	1.6	0.5
45	0.357	0.405	0.413	0.32	1.56	0.05

7.2.2 Pongola experiment

The AgrowthHR trial was conducted in Pongola (27°24'S, 31°35'E) in Northern Kwazulu-Natal, South Africa from 1967 to 1970. Cultivar NCo376 was ratooned and harvested at 8 different times (Table 7.3). Row spacing was 1.5 m and crops were irrigated to avoid water stress. Soil profile information is listed in Table 7.4. Initial soil moisture conditions were assumed to be 50% of available water content. (DSSAT4.5 experiment name- SXPG6801).

Table 7.3. Trial treatments

Treatment	Crop type	Start	Harvest	Harvest age (days)
1	Ratoon	17 Dec 1968	05 May 1970	504
2	Ratoon	11 Feb 1969	30 Jun 1970	504
3	Ratoon	08 Apr 1969	25 Aug 1970	504
4	Ratoon	03 Jun 1969	20 Oct 1970	504
5	Ratoon	29 Jul 1969	15 Dec 1970	504
6	Ratoon	23 Sep 1969	09 Feb 1971	504
7	Ratoon	18 Nov 1969	06 Apr 1971	504
8	Ratoon	13 Jan 1970	29 May 1971	501

Table 7.4. Soil profile information

Layer thickness	Lower limit (cm^3/cm^3)	Drained upper limit (cm^3/cm^3)	Saturated water content (cm^3/cm^3)	Rooting weight	Bulk density (g/cm^3)
5	0.101	0.261	0.368	1.0	1.39
12	0.101	0.261	0.368	0.82	1.39
15	0.101	0.261	0.368	0.64	1.39
15	0.160	0.282	0.371	0.47	1.43
15	0.160	0.282	0.371	0.35	1.43
30	0.151	0.304	0.399	0.22	1.34
30	0.151	0.304	0.399	0.12	1.34
30	0.151	0.304	0.399	0.07	1.34
30	0.151	0.304	0.399	0.03	1.34
90	0.151	0.304	0.399	0.01	1.34

Note: Saturated hydraulic conductivity per layer is unknown for this soil; the soil is classified as draining 'moderately well' in the DSSAT Sbuild program, which corresponds with an overall profile drainage rate of 0.4 $cm.h^{-1}$.

7.3. Results

7.3.1 La Mercy experiment

Simulated and observed values of aerial dry mass, stalk dry mass, sucrose mass and leaf area index are compared in Figures 7.1 to 7.4. Corresponding statistical comparisons are given in Tables 7.5 to 7.8.

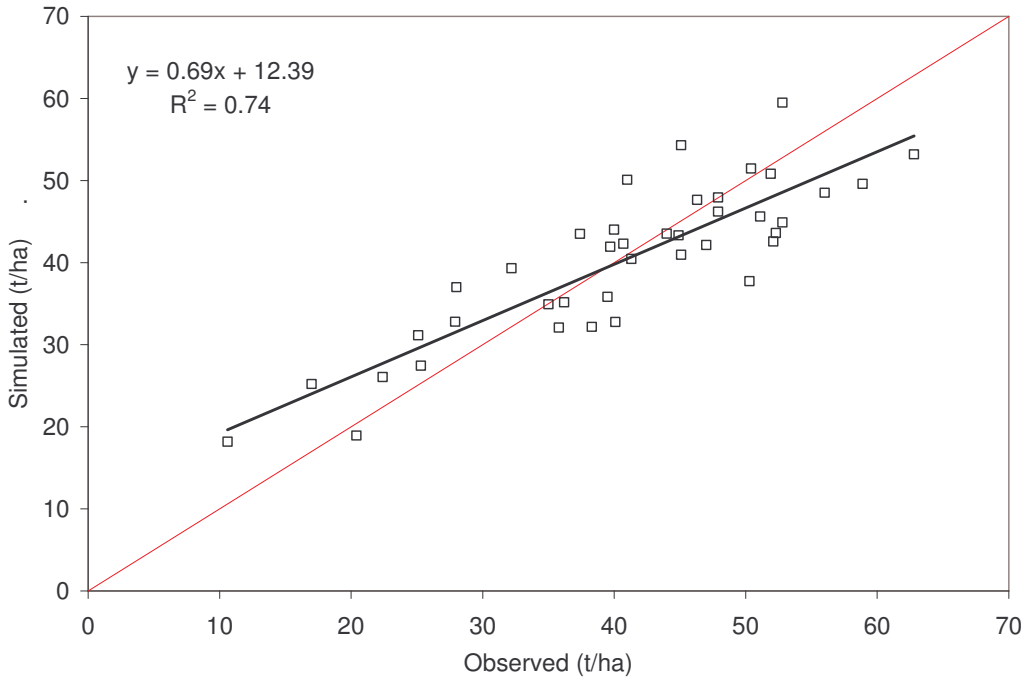


Figure 7.1. Simulated and observed aerial dry mass for the La Mercy experiment.

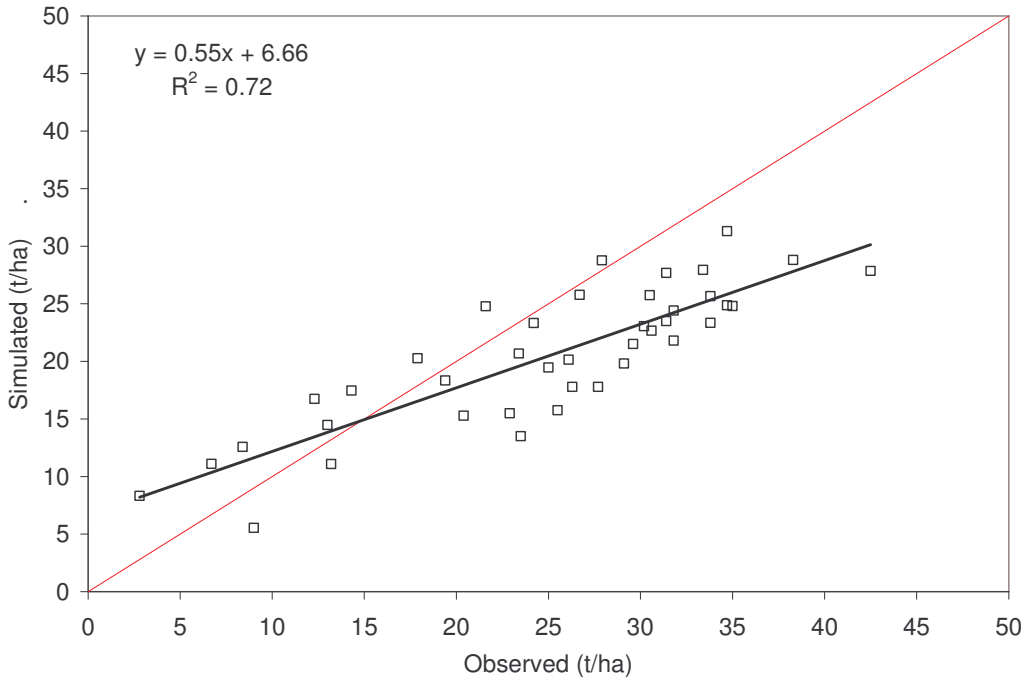


Figure 7.2. Simulated and observed stalk dry mass for the La Mercy experiment.

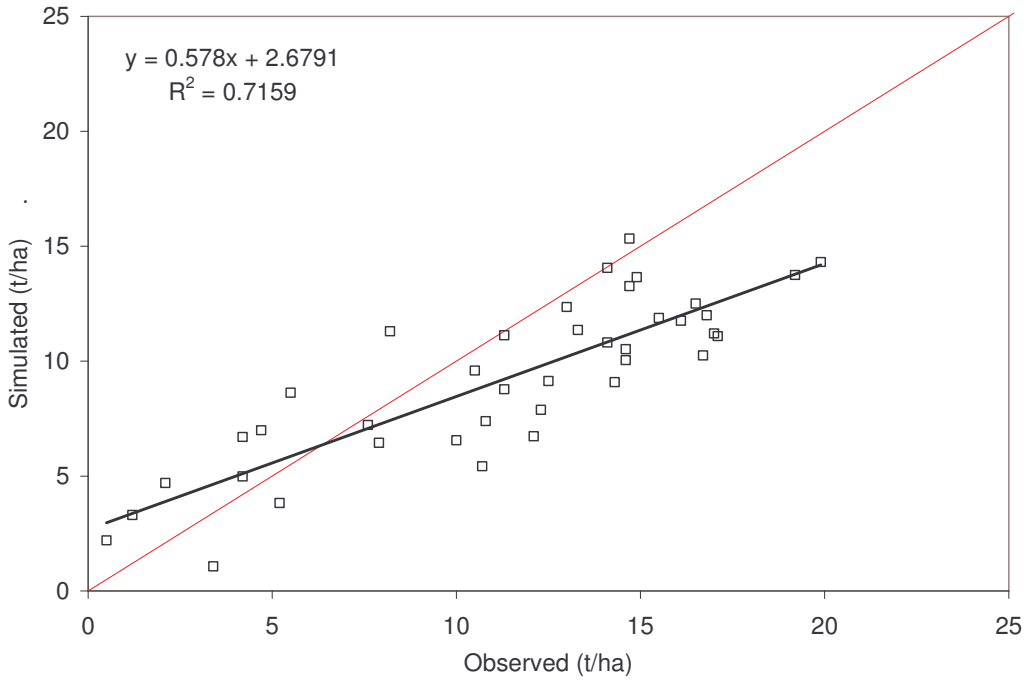


Figure 7.3. Simulated and observed sucrose mass for the La Mercy experiment.

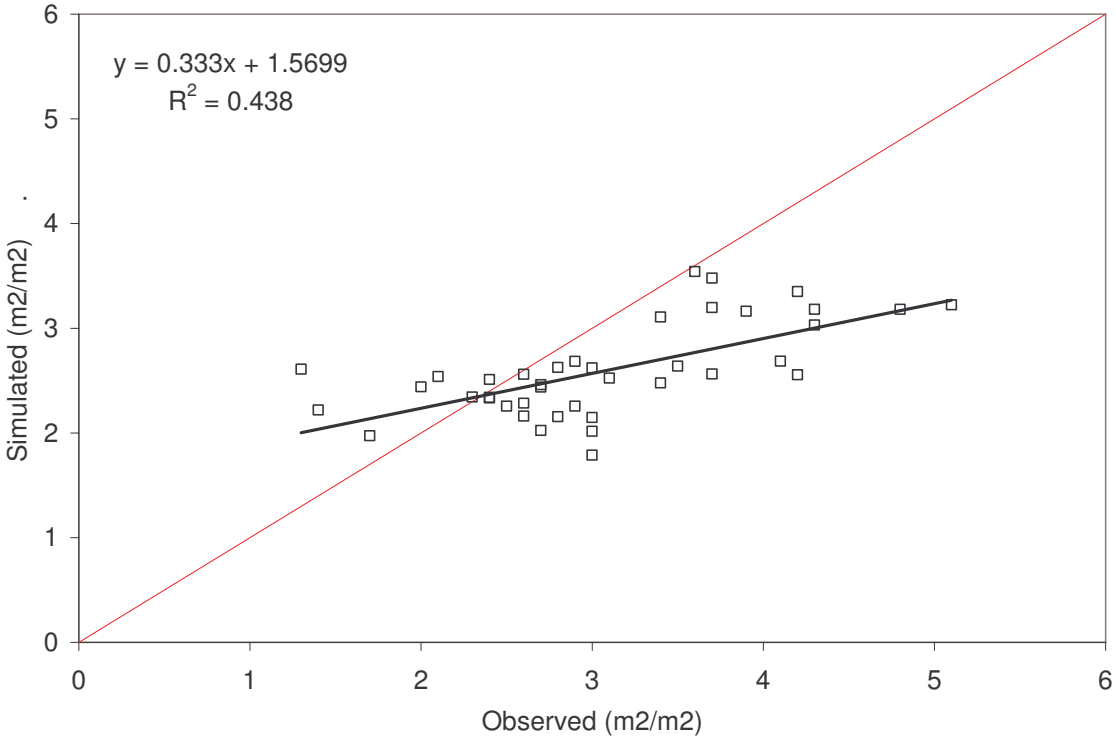


Figure 7.4. Simulated and observed green leaf area index (m^2/m^2) for the La Mercy experiment.

Table 7.5. Validation statistics for aerial biomass (t/ha) for La Mercy experiment

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	-8.94	-3.74	14.73	6.17	0.98	5	41.86	38.12	0.91
2	4.27	1.70	12.53	4.99	0.47	5	39.86	41.56	1.04
3	-7.22	-3.04	9.76	4.11	0.94	5	42.14	39.10	0.93
4	0.03	0.01	9.49	4.17	0.89	5	43.98	43.99	1.00
5	-13.48	-5.79	18.40	7.90	0.95	5	42.96	37.17	0.87
6	5.81	2.09	18.33	6.60	0.93	5	35.98	38.07	1.06
7	8.15	3.04	15.44	5.76	0.91	5	37.32	40.36	1.08
8	4.73	2.01	16.57	7.06	0.35	5	42.60	44.61	1.05
Pooled	*-1.14*	*-0.46*	*14.65*	*5.98*	*0.74*	*40*	*40.84*	*40.37*	*0.99*

Table 7.6. Validation statistics for stalk dry mass (t/ha) for La Mercy experiment

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	-19.19	-4.86	29.32	7.42	0.99	5	25.30	20.44	0.81
2	-18.11	-4.47	24.68	6.09	0.56	5	24.68	20.21	0.82
3	-26.99	-7.35	27.78	7.57	0.86	5	27.24	19.89	0.73
4	-24.73	-6.97	25.95	7.32	0.92	5	28.20	21.23	0.75
5	-31.65	-8.06	36.52	9.31	0.98	5	25.48	17.42	0.68
6	-0.06	-0.01	26.93	5.73	0.93	5	21.28	21.27	1.00
7	-5.61	-1.25	20.40	4.53	0.93	5	22.22	20.97	0.94
8	-12.85	-3.31	23.30	6.00	0.41	5	25.76	22.45	0.87
Pooled	*-18.13*	*-4.54*	*27.51*	*6.88*	*0.72*	*40*	*25.02*	*20.48*	*0.82*

Table 7.7. Validation statistics for sucrose mass (t/ha) for La Mercy experiment

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	-20.60	-2.38	33.97	3.92	0.98	5	11.54	9.16	0.79
2	-21.19	-2.43	28.99	3.33	0.69	5	11.48	9.05	0.79
3	-27.20	-3.35	30.19	3.72	0.64	5	12.32	8.97	0.73
4	-23.78	-3.02	28.33	3.59	0.79	5	12.68	9.66	0.76
5	-27.95	-2.90	31.20	3.24	0.96	5	10.38	7.48	0.72
6	0.96	0.09	33.68	3.21	0.87	5	9.54	9.63	1.01
7	-6.49	-0.66	30.93	3.13	0.88	5	10.12	9.46	0.94
8	-17.30	-2.18	32.00	4.03	0.49	5	12.60	10.42	0.83
Pooled	*-18.56*	*-2.10*	*31.21*	*3.54*	*0.72*	*40*	*11.33*	*9.23*	*0.81*

Table 7.8. Validation statistics for green leaf area index (m^2/m^2) for La Mercy experiment

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	-21.94	-0.70	24.03	0.77	0.74	5	3.20	2.50	0.78
2	-5.43	-0.16	14.54	0.42	0.80	5	2.90	2.74	0.95
3	-9.71	-0.28	14.20	0.41	0.66	5	2.86	2.58	0.90
4	-10.28	-0.30	13.63	0.40	0.64	5	2.90	2.60	0.90
5	-32.88	-1.25	35.55	1.35	0.63	5	3.80	2.55	0.67
6	-15.28	-0.47	24.94	0.77	0.54	5	3.10	2.63	0.85
7	-22.64	-0.73	31.42	1.02	0.92	5	3.24	2.51	0.77
8	2.74	0.07	34.32	0.88	0.60	5	2.56	2.63	1.03
Pooled	*-15.56*	*-0.48*	*26.56*	*0.82*	*0.44*	*40*	*3.07*	*2.59*	*0.84*

7.3.2 Pongola experiment

Simulated and observed values of stalk dry mass and sucrose massfor the Pongola experiment are compared in Figure 7.5 and 7.6. Statistical comparisons are given in Tables 7.9 and 7.10.

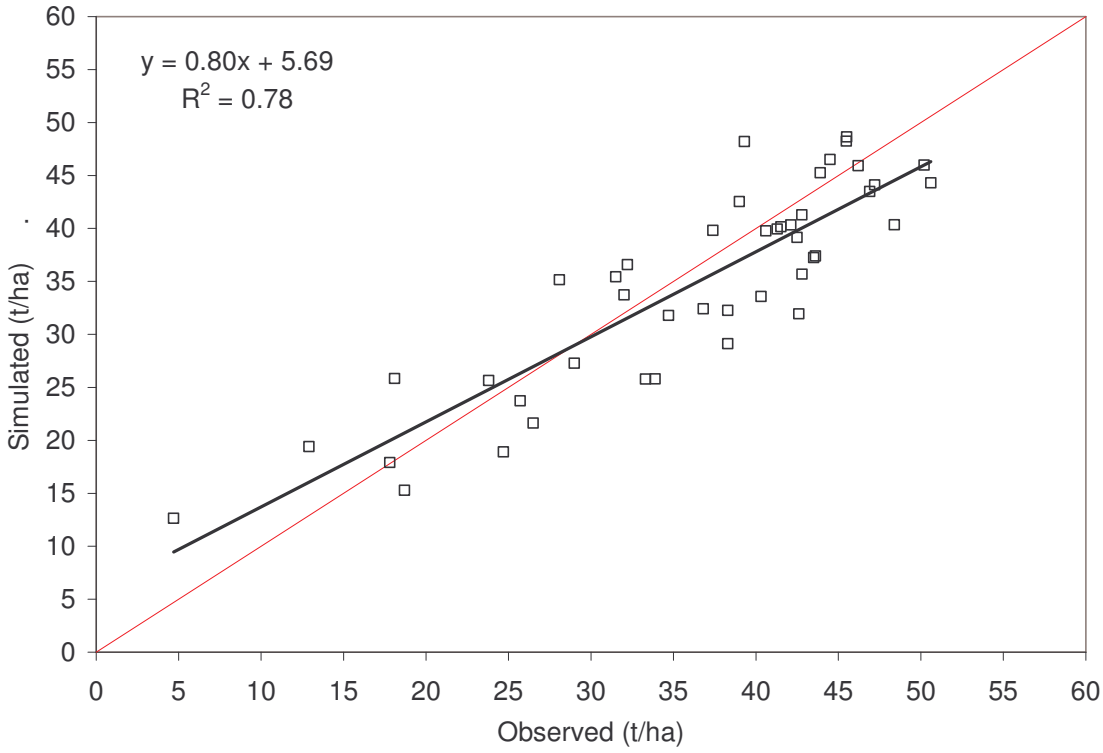


Figure 7.5. Simulated and observed stalk dry mass for Pongola experiment.

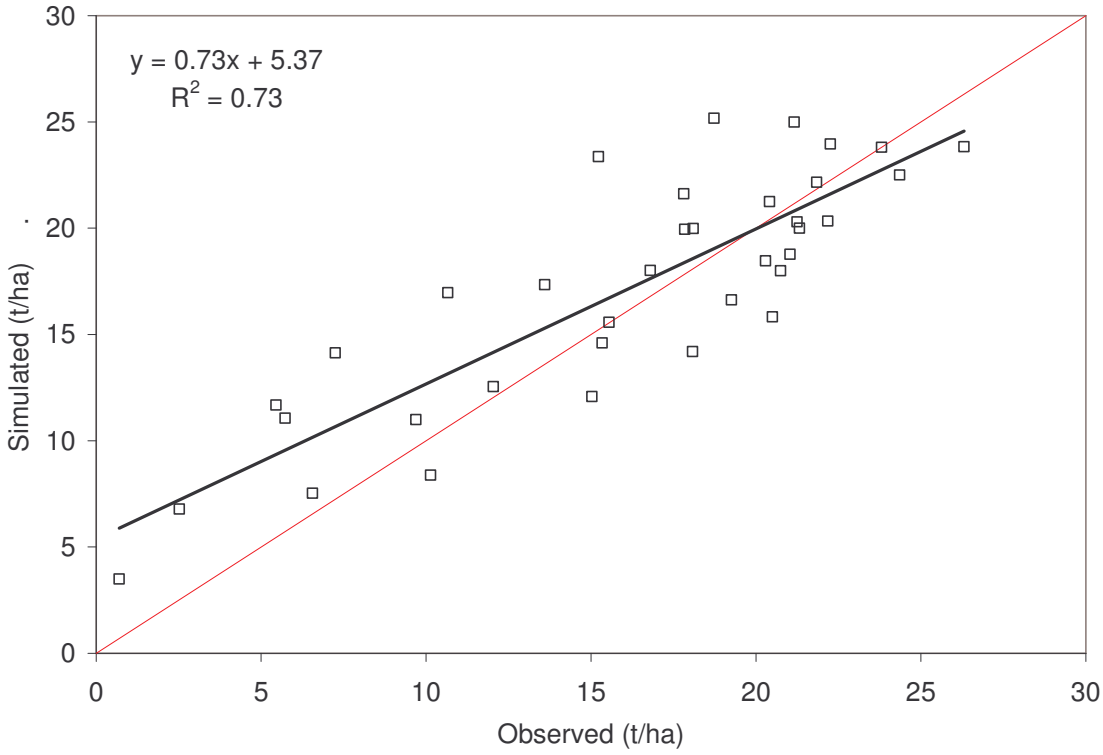


Figure 7.6. Simulated and observed sucrose mass for Pongola experiment.

Table 7.9. Validation statistics for stalk dry mass for Pongola experiment.

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	-5.05	-1.75	17.69	6.13	0.79	5	34.64	32.89	0.95
2	2.36	0.82	8.25	2.87	0.95	6	34.75	35.57	1.02
3	12.92	3.72	19.19	5.52	0.96	6	28.76	32.48	1.13
4	-1.18	-0.42	13.91	4.95	0.93	5	35.62	35.20	0.99
5	-11.41	-4.86	14.26	6.07	0.51	5	42.56	37.71	0.89
6	-11.44	-4.69	15.97	6.55	0.65	6	41.02	36.33	0.89
7	-6.84	-2.62	12.24	4.69	0.94	6	38.27	35.65	0.93
8	-6.72	-2.26	10.30	3.47	0.95	5	33.66	31.40	0.93
Pooled	*-3.97*	*-1.43*	*14.31*	*5.17*	*0.78*	*44*	*36.12*	*34.69*	*0.96*

Table 7.10. Validation statistics for sucrose mass for Pongola experiment.

Trt	APE%	APE	RMSE%	RMSE	R^2	No of obs	Mean Observed	Mean Simulated	Ratio
1	12.18	1.87	29.64	4.56	0.66	6	15.39	17.26	1.12
2	7.52	1.23	15.29	2.49	0.91	6	16.30	17.52	1.08
3	22.33	2.85	30.62	3.91	0.90	6	12.76	15.61	1.22
4	10.04	1.52	23.23	3.51	0.91	6	15.13	16.65	1.10
5	-0.17	-0.03	16.26	2.87	0.79	6	17.62	17.59	1.00
6	-6.79	-1.32	16.13	3.13	0.60	6	19.41	18.09	0.93
7	0.51	0.09	15.54	2.72	0.88	6	17.49	17.58	1.01
8	-3.81	-0.64	10.44	1.75	0.95	6	16.75	16.11	0.96
Pooled	*4.26*	*0.70*	*19.70*	*3.22*	*0.74*	*48*	*16.35*	*17.05*	*1.04*

7.4. Discussion

Although overall model performance was reasonable, the model consistently underestimated high values of green LAI, stalk mass and sucrose mass at La Mercy. Aerial dry mass at La Marcy was calculated accurately, as was stalk dry mass and sucrose mass at Pongola. As can be expected, simulation accuracy was best for aerial dry mass, followed by stalk dry mass and then sucrose mass. The accuracy achieved for this validation (RMSE values for stalk dry matter of 6.88 and 5.17 t/ha for the La Mercy and Pongola experiments respectively, and RMSE values for sucrose mass of 3.54 and 3.22 t/ha for the La Mercy and Pongola experiments respectively) is better than that quoted by O'Leary (2000) for the same data sets (RMSE value of 11.11 t.ha for stalk dry mass and 6.07 t/ha for sucrose mass respectively). Simulation accuracy also compares well with that found in a more comprehensive validation conducted by Singels & Bezuidenhout (2002). They found RMSE values for aerial dry mass, stalk dry mass and sucrose mass predictions of 6.94, 5.48 and 2.6 t/ha respectively.

Further investigation is required to identify reasons for the simulation bias observed for the La Mercy experiment. The soil input used indicates a poorly drained soil with very low conductivity in the bottom layer. This amounts to the presence of a water table most of time, the impacts of which are difficult to simulate.

7.5. Performance comparison of DSSAT v3.5 and v4.5 versions of Canegro

Simulations by the DSSAT v3.5 and DSSAT v4.5 versions of Canegro are compared with observed values in Figures 7.7 to 7.10. In the following graphs, mean values are displayed. In each season, measurements

were taken on a number of occasions. The observed and simulated values on these days have been averaged and plotted.

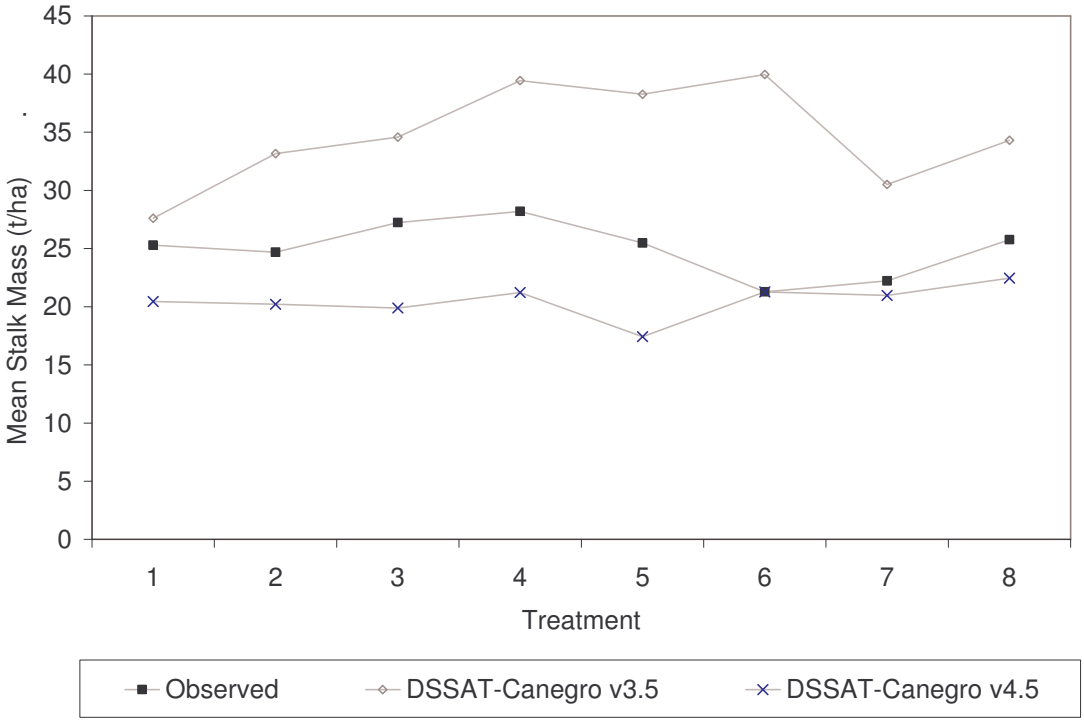


Figure 7.7. Mean stalk mass (t/ha) per treatment, simulated and observed, La Mercy 1989-90.

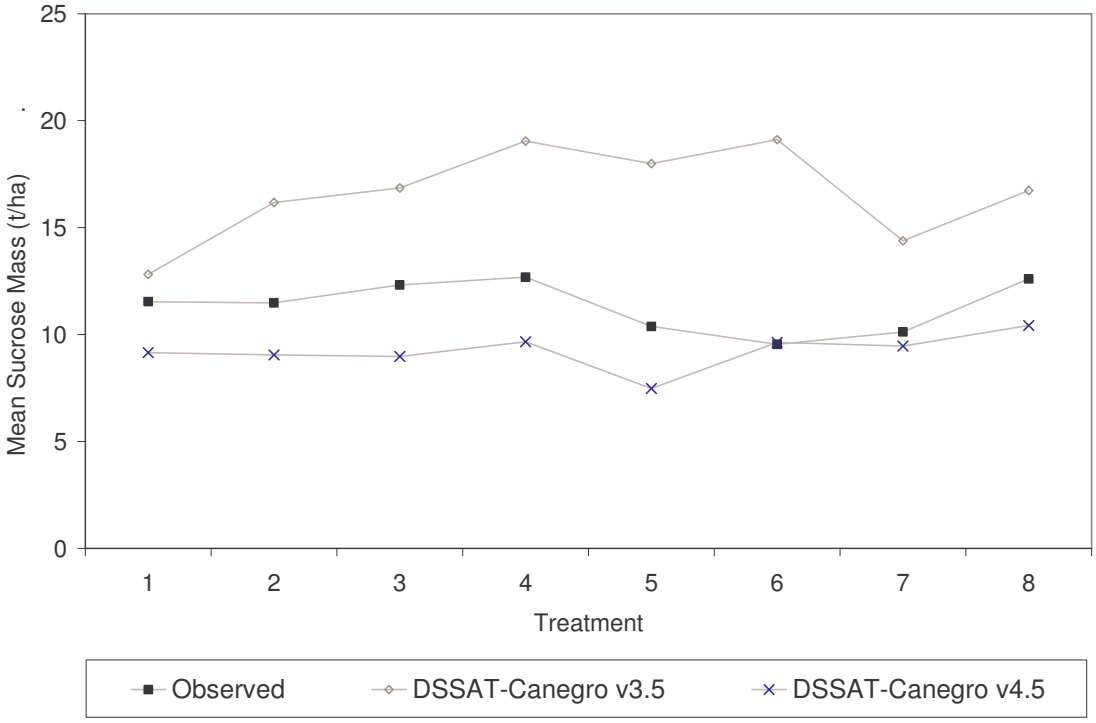


Figure 7.8. Mean sucrose mass (t/ha) per treatment, simulated and observed, La Mercy 1989-90.

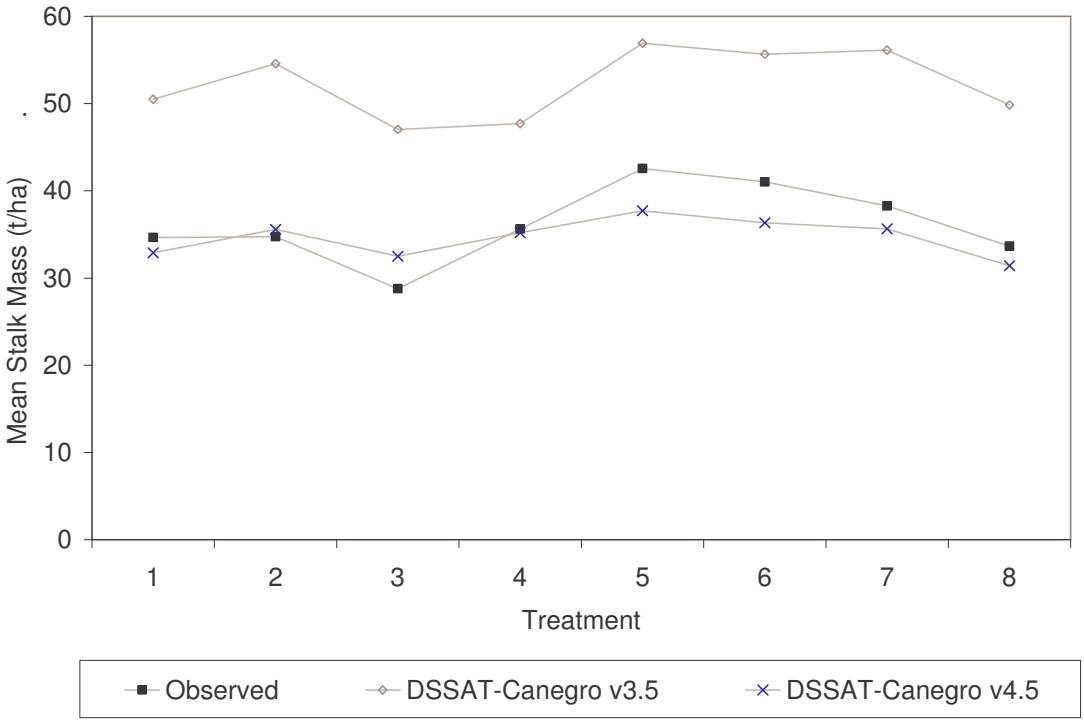


Figure 7.9. Mean stalk dry mass (t/ha), per treatment, simulated and observed, Pongola 1968-70.

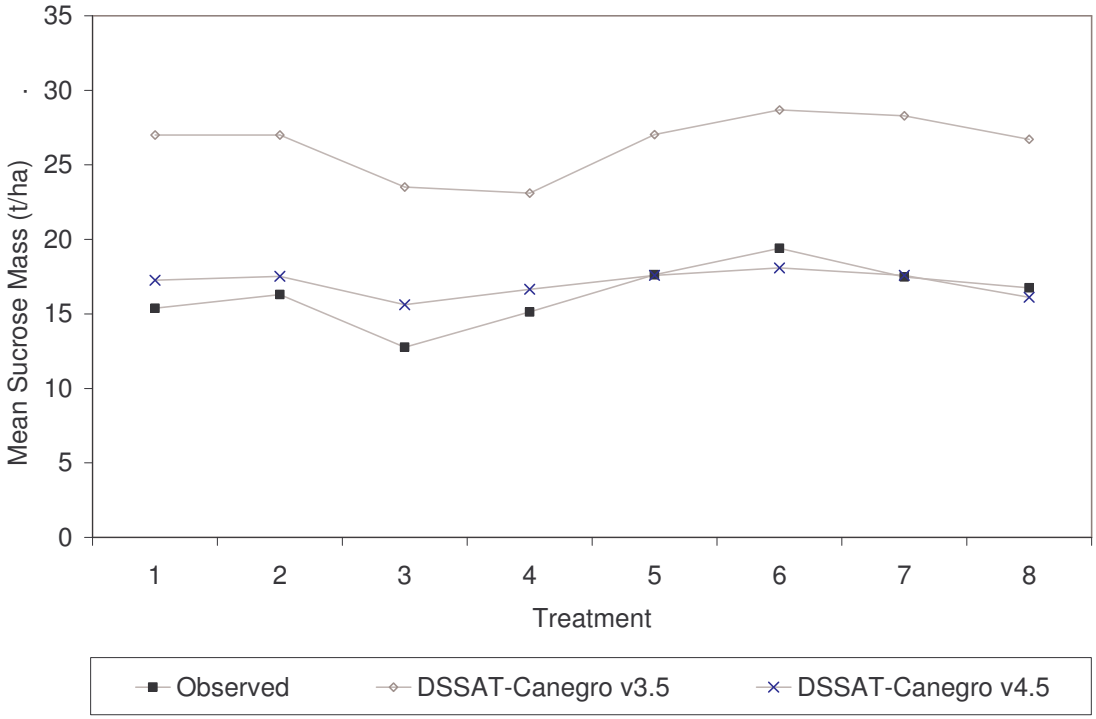


Figure 7.10. Sucrose mass (t/ha), per treatment, simulated and observed, Pongola 1968-70.

The DSSAT v4.5 version of Canegro appears to perform substantially better, overall, than the DSSAT v3.5 version. While absolute yields are closer to reality with DSSAT v4.5, DSSAT v3.5 appears to respond slightly better to seasonal variations.

8. Conclusions

An up-to-date version of Canegro with enhanced capabilities (temperature-dependent photosynthesis and radiation, source-sink approach to biomass partitioning, lodging and an option for thermal time driven canopy development) was incorporated successfully into DSSAT v4.5. A number of species, ecotype and cultivar parameters were defined and the latter are accessible to users for calibration of new genotypes. It is expected that the number of parameters will be reduced significantly as new insights are gained with respect to their impact on crop growth and the actual genotypic variability that exist in associated traits. Parameters for five ecotypes and seven real and three hypothetical cultivars are provided with the software.

The new code was verified by comparing the output of the DSSAT v4.5 version of the Canegro model with that of the SASRI stand-alone version. For well-watered scenarios the output was almost identical, while small discrepancies existed for water limited scenarios. The sources of these discrepancies were traced to differences in calculation methods in modules outside the plant module, such as reference evapotranspiration. The discrepancies were deemed acceptable.

The new model was validated by comparing simulated values with observation of LAI, aerial dry mass, stalk dry mass and sucrose mass for two South African experiment (irrigated and rainfed). The overall performance of the model was highly satisfactory and was better than that of the DSSAT3.5 version, although there was systematic under-estimation of high values of LAI, stalk dry mass and sucrose mass for the dryland experiment.

9. Acknowledgements

The following organisations and individuals are acknowledged for their contributions to developing the DSSAT v4.5 Canegro model:

- Jim Jones and Cheryl Porter of the University of Florida,
- Gerrit Hoogenboom of the University of Georgia,
- Rasack Nayamuth of MSIRI, Graham Kingston of BSES and Jim Shine of SCGC for the initiative and continued moral support,
- The ICSM, BSES, SCGC, ZSAES, SASRI and Chiang Mai University for financial and in-kind contributions.

10. References

Allen R.G, Pereira, L.S., Raes, D. and Smith, M., 1998. Crop evapotranspiration. Guidelines for computing crop water requirements. FAO irrigation and drainage paper no. 56. FAO of the UN, Rome.

Campbell, G. and Norman, J., 1998. An Introduction to Environmental Physics, Springer-Verlag New York, pp42-44.

Inman-Bamber N.G. and Kiker G.A., 1997. CANEGRO 3.10. DSSAT version 3.1 1998 distribution software. IBSNAT, University of Hawaii, Honolulu, Hawaii, USA.

Inman-Bamber, N.G., 1991. A growth model for sugarcane based on a simple carbon balance and the CERES-Maize water balance. S. Afr. J. Plant Soil, 8: 93-99.

Inman-Bamber, N.G., 1994a. Temperature and seasonal effects on canopy development and light interception of sugarcane. Field Crops Research 36, 41-51.

Inman-Bamber, N.G., 1994b. Effect of age and season on components of yield of sugarcane in South Africa. Proc S Afr Sug Technol Ass 68: 23-27.

Jones, C.A. and Kiniry, J.R., 1986. CERES-Maize model: A simulation model of maize growth and development. Texas A&M University Press, 194 pp.

Jones, J.W., Hoogenboom, G., Porter, C.H., Boote, K.J., Batchelor, W.D., Hunt, L.A., Wilkens, P.W., Singh, U., Gijsman, A.J. and Ritchie, J.T., 2003. The DSSAT cropping system model. Eur. J. Agron., 18: 235-265.

Jones, M.R. and Singels, A., 2008. DSSAT v4.5 - Canegro Sugarcane Plant Module. User documentation. ICSM report.

Liu, D.L. and Bull, T.A., 2001. Simulation of biomass and sugar accumulation in sugarcane using a process-based model. Ecological Modelling 144, 181-211.

McGlinchey, M.G. and Inman-Bamber, N.G., 1996. Predicting sugarcane water use with the Penman-Monteith equation. In: Camp, C.R., Sadler, E.J., Yoder, R.E. (Eds.), Evapotranspiration and irrigation scheduling: Proc. Int. Conf., Nov 3-6. San Antonio. ASAE, St Joseph, Michigan, pp. 592-598.

O'Leary, G.J., 2000. A review of three sugarcane simulation models in their prediction of sucrose yield. Field Crops Res. 68: 97-111.

Rostron, H., 1972. Some Effects of Environment, Age and Growth Regulating Compounds on the Growth, Yield and Quality of Sugarcane in South Africa. M.Sc. thesis, Leeds University.

Schmidt, E.J. and Schulze, R.E., 1987. SCS based design runoff. User manual. Technical report by the Agric. Catchments Research Unit to the Water Research Commission of South Africa (WRC). WRC project no. 155 (TT 33/87).

Singels, A. and Donaldson, R.A. 2000. A simple model of unstressed sugarcane canopy development. Proc S Afr Sug Technol Ass 74: 151-154.

Singels, A. & Bezuidenhout, C.N., 2002. A new method of simulating dry matter partitioning in the Canegro sugarcane model. Field Crops Research 78: 151-164.

Singels, A., Donaldson, R.A. and Smit, M.A., 2005. Improving biomass production and partitioning in sugarcane: theory and practice. Field Crops Res. 92: 291-303.

Singh G., Chapman, S.C., Jackson, P.A. and Lawn, R.J., 1999. Yield accumulation in sugrance under wet tropical conditions – Effect of lodging and crop age. Proc. Aust. Soc. Sugar Cane Technol., 21: 241-245.

Smit, M.A. and Singels, A., 2006. The response of sugarcane canopy development to water stress. Field Crops Research 98:91-97.

Tsuji, G.Y., Uehara, G. and Balas, S. [Eds], 1994. DSSAT version 3 Volumes 1, 2, and 3. International Benchmark Sites Network for Agrotechnology Transfer. University of Hawaii, Honolulu, Hawaii.

Van Antwerpen, R., 1998. Modelling Root Growth and Water Uptake of Sugarcane Cultivar NCo376. University of the Orange Free State, Bloemfontein (PhD thesis).

Volume 4
Chapter 4

Weather Module

C.H. Porter
N.B. Pickering
J.W. Jones
G. Hoogenboom

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
International Consortium for Agricultural Systems Applications

Weather Module

March 31, 2004
modified April 2010

C.H. Porter, N.B. Pickering, J.W. Jones, G. Hoogenboom

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
International Consortium for Agricultural Systems Applications

WEATHER MODULE
TABLE OF CONTENTS

1.0 Weather Module Overview 3

2.0 Weather Module Main Routine (WEATHR) 5
2.1 Run Initialization 5
2.2 Seasonal Initialization 5
2.3 Rate Calculations 6
2.4 Final 6

3.0 Weather Data Input (IPWTH) 7
3.1 Run Initialization 7
3.2 Seasonal Initialization 8
3.3 Rate Calculations 8
3.4 Final 9

4.0 Weather Generation Routines (WGEN) 9

5.0 Day Length Calculation (DAYLEN) 11

6.0 Atmospheric CO_2 12

7.0 Weather Modification Routines 13
7.1 WTHMOD 13
7.2 WTHSET 15
7.3 DECL 15

8.0 Solar Radiation Calculations (SOLAR) 16

9.0 Hourly Meteorological Parameters (HMET) 17
9.1 HANG 18
9.2 HTEMP 19
9.3 HRAD 21
9.4 FRACD 22
9.5 HPAR 24
9.6 VPSAT 24
9.7 VPSLOP 24

10.0 References 25

Cover photograph: Rainstorm, Nairobi, Kenya. Courtesy of H. David Thurston, Professor Emeritus - Department of Plant Pathology, Cornell University.

LIST OF TABLES

Table 1 – Subroutines called from WEATHR Module 3
Table 2 – Interface for WEATHR Module 5
Table 3 – Daily Weather Values Read or Generated 6
Table 4 – Interface for IPWTH Routine 7
Table 5 – Interface for WGEN Routines 10
Table 6 – Interface for DAYLEN subroutine 11
Table 7 – Methods of obtaining atmospheric CO_2 values 12
Table 8 – Interface for weather modification subroutines 13
Table 9 – Weather modification parameters read by WTHMOD 14
Table 10 – Adjustment factors computation 15
Table 11 – Interface for SOLAR subroutine 16
Table 12 – Interface for HMET subroutines 17

LIST OF FIGURES

Figure 1 – Components of Weather Module 4
Figure 2 – Hourly Solar Angle Calculations for 30°N latitude on May 20 (day 140) 19
Figure 3 – Daily Temperature Distribution 20
Figure 4 – Hourly solar radiation (subroutine HRAD) 21
Figure 5 – Hourly distribution of fraction diffuse total radiation (FRDIFR) and fraction diffuse PAR (FRDIFP) 22
Figure 6 – Fraction diffuse radiation computation 23

1.0 Weather Module Overview

The Weather module is called by CSM to input, generate or calculate the daily and hourly weather values used by other parts of the model. A user-supplied input variable, MEWTH, is used to determine if weather data is to be read from a file or generated. The FORTRAN code is organized in a modular format as described in "Module Structure in CROPGRO 3.5x Documentation and Source Code Listing" (Porter, et al., 1998).

The weather module is called a total of four times from the CROPGRO main program:
1) At the beginning of simulation for input of parameters from files (RUNINIT),
2) At the beginning of each seasonal simulation for initialization (SEASINIT),
3) Daily for input or generation of daily and hourly weather values (RATE) and
4) At the end of simulation to close any weather files which remain open (FINAL).

Figure 1 presents the components of the module. As shown, the WEATHR routine controls the flow of processing within the module and calls six other routines. Depending on the value of the variable, MEWTH, the program calls either IPWTH to read daily weather data from FILEW or WGEN to generate daily weather values. DAYLEN is called to generate day length for the given day and latitude.

The weather modification routine, WTHMOD is called as needed to apply user-specified modifications to historical or generated data.

Subroutine SOLAR computes declination, solar time of sunrise and sunset, a daily solar constant and a cloudiness factor. HMET is called to generate hourly values of weather data, which are required by some CROPGRO functions.

The components of the Weather module are summarized Table 1 including the order in which routines are called from the WEATHR main module for each of the four processing functions. For instance, in the 'Rate Calculations' section, six routines are called by WEATHR in the order shown. Each routine will be discussed separately in this document beginning with the main WEATHR module in Section 2.0. Subsequent sections will be discussed in the order in which they are called from the 'Rate calculations' section of the main Weather module (as listed in Table 1).

Table 1 – Subroutines called from WEATHR Module

Routine Called from WEATHR	Run Initialization	Seasonal Initialization	Rate Calculations	Final	Section number this document	Description
IPWTH	1	1	1	1	3.0	Read input data from FILEW
WGEN	2		2		4.0	Generate daily weather values (Note: these routines will be removed from future versions of the CROPGRO model)
DAYLEN			3		5.0	Calculate day length
WTHMOD	3		4		6.0	Modify daily weather values
SOLAR			5		7.0	Compute daily solar factors
HMET			6		8.0	Compute hourly weather factors

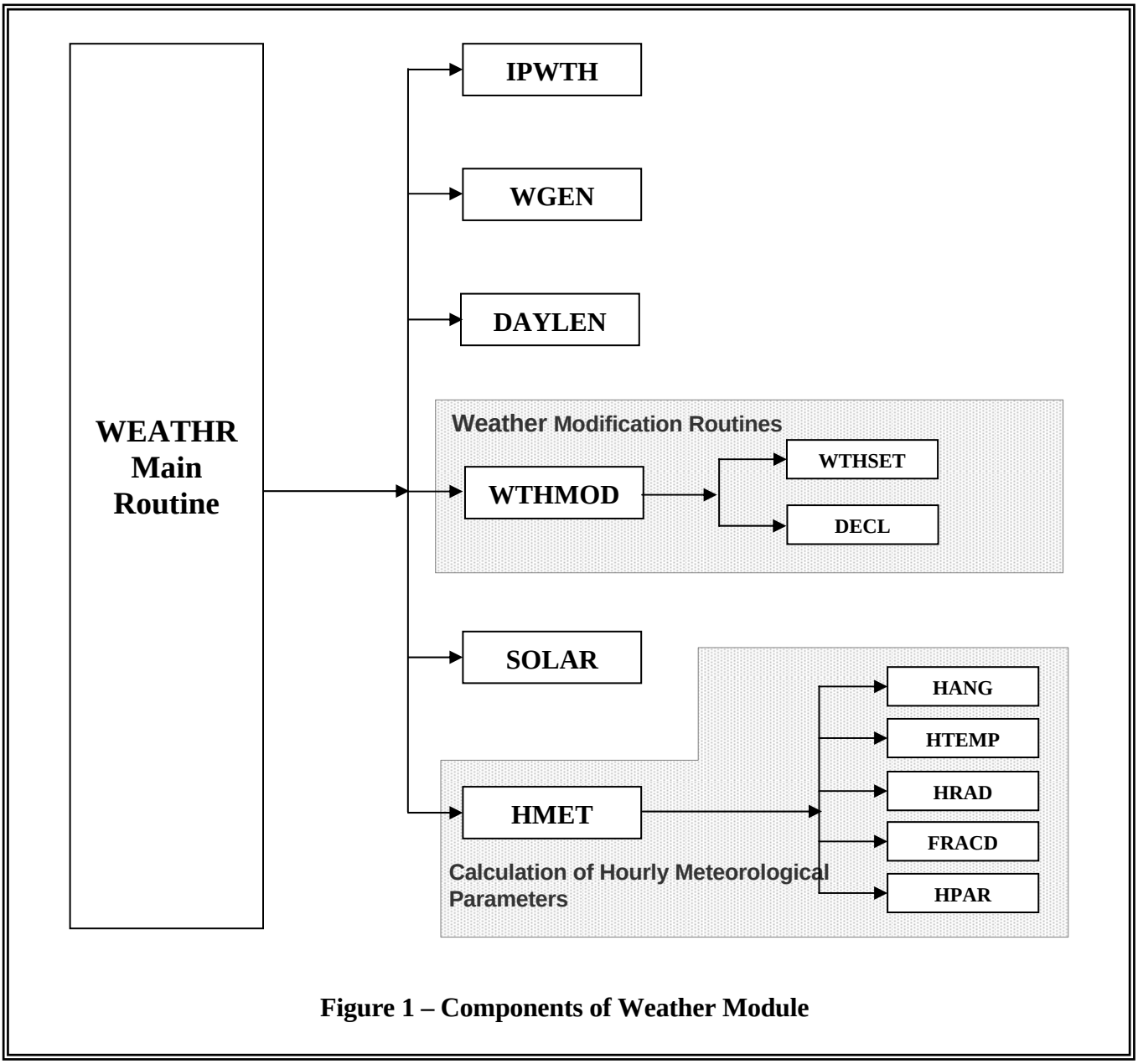


Figure 1 – Components of Weather Module

2.0 Weather Module Main Routine (WEATHR)

The Weather Module code is divided into the 4 phases of computation: 'Run Initialization', 'Seasonal Initialization', 'Rate Calculations', and 'Final'. Table 2 presents a list of variables that occur as calling arguments for the module. These are the only variables which interface with the main CROPGRO program and which are available for use by other portions of the model.

Table 2 – Interface for WEATHR Module

	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
Output	WEATHER	Composite variable containing daily and hourly weather data. See Appendix A.	--
	YREND	End of simulation indicator.	YYYYDOY

2.1 Run Initialization

Within the 'Run Initialization' section of code, two subroutines are called that read from file FILEIO. (FILEIO is a temporary input file that is generated by the input module for CSM. This preprocessor reads FILEX and other user data files and writes required information to FILEIO for use by the CSM. The current model assigns the name 'DSSAT45.INP' to FILEIO for single season, non-sequenced simulations.)

Subroutine IPWTH is called to read variable MEWTH from FILEIO. This variable determines if weather data will be read from the input file (MEWTH = 'M' or 'G') or generated by the WGEN routines (MEWTH = 'S' or 'W'). Variable RSEED1 is also read for use by the weather generator routines.

If weather data is to be generated (MEWTH = 'W' or 'S'), the weather generation routine (WGEN) is called to read input data for use by that routine.

Subroutine WTHMOD is called to determine if the user has specified that modification of the weather data be performed. This routine reads the "Environmental Modification" section of FILEIO. If no weather modification is to occur, the variable NEV (number of environmental modification records) is returned with a value of zero.

2.2 Seasonal Initialization

The 'Seasonal Initialization' portion of the Weather module is called once per season of simulation. If daily weather data is to be read from a file (MEWTH = 'M' or 'G'), subroutine IPWTH is called to determine the name of the weather data input file to be used for multi-year or sequenced simulations. Also within IPWTH, the weather data input file is opened and general information about the weather station is read.

If the weather generation routines are to be used (MEWTH = 'S' or 'W'), then daily weather parameters are initialized.

If a value for atmospheric carbon dioxide concentration (CCO2) has been input, then it will be used for calculations. Otherwise, the base value of CO2BAS (a parameter equal to 330 ppm) will be used:

```
IF (CCO2 .LE. 0.) THEN
   CO2 = CO2BAS
ELSE
   CO2 = CCO2
ENDIF
```

2.3 Rate Calculations

The 'Rate Calculations' section of the weather module is called once per day from the CROPGRO model. At the beginning of this section, the program branches depending on the value of MEWTH. If MEWTH is 'M' or 'G', a call is made to IPWTH to retrieve the daily weather data from FILEW. If MEWTH is 'S' or 'W', a call is made to WGEN to generate daily weather values. Table 3 lists the daily weather values that are returned by these routines.

Table 3 – Daily Weather Values Read or Generated

Variable Name	Definition	Units	Returned by Routine	
			IPWTH	WGEN
PAR	Daily photosynthetically active radiation or photon flux density, calculated as half the solar radiation	MJ / m2 - d	Yes	Yes
RAIN	Precipitation depth for current day	mm	Yes	Yes
RHUM	Relative humidity	%	Yes	No
SRAD	Solar Radiation	MJ/m2-d	Yes	Yes
TDEW	Dewpoint temperature	°C	Yes	Yes
TMAX	Maximum daily temperature	°C	Yes	Yes
TMIN	Minimum daily temperature	°C	Yes	Yes
WINDSP	Wind speed	km/d	Yes	Yes

Routine DAYLEN is called to determine the length of day, solar declination and time of sunrise and sunset.

If weather modification is to occur (NEV > 0), subroutine WTHMOD is called to adjust the applicable daily weather values.

Subroutine SOLAR is then called to compute the relative cloudiness factor (CLOUDS), and the extraterrestrial solar radiation (SON). A factor, ISINB is also returned which will be used to compute hourly solar radiation. Hourly meteorological factors are then computed by the HMET routines.

2.4 Final

A fourth call to WEATHR is made to close input files for the end of simulation. Subroutine IPWTH is called to accomplish this.

3.0 Weather Data Input (IPWTH)

Subroutine IPWTH is called four times from the Weather main routine: once each from the 'Run Initialization', 'Seasonal Initialization', 'Rate Calculations' and 'Final' sections. IPWTH is called for 'Run Initialization' regardless of whether daily weather data will be read or generated. Thereafter, IPWTH is called only if daily weather variables are to be read from the weather file rather than generated. Table 4 lists input and output variables for this routine.

Table 4 – Interface for IPWTH Routine

	Variable	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DYNAMIC	Module control variable; =RUNINIT, SEASINIT, RATE, EMERG, INTEGR, OUTPUT, or FINAL	--
Output Variables	CCO2	Atmospheric CO2 concentration read from input file	ppm
	FILEW	Weather data file	--
	MEWTH	Switch for method of obtaining weather data-- 'G' or 'M'- read weather data file 'S'- read SIMMETEO inputs and generate weather data 'W'- read WGEN inputs and generate weather data	--
	PAR	Daily photosynthetically active radiation or photon flux density, calculated as half the solar radiation	MJ / m2 - d
	PATHWT	Directory path for weather file	--
	RAIN	Precipitation depth for current day	mm
	REFHT	Reference height for wind speed	m
	RHUM	Relative humidity	%
	RSEED1	Random number generator seed- user input	--
	SRAD	Solar Radiation	MJ/m2-d
	TAMP	Amplitude of temperature function used to calculate soil temperatures	°C
	TAV	Average annual soil temperature, used with TAMP to calculate soil temperature.	°C
	TDEW	Dewpoint temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	WINDHT	Reference height for wind speed	m
	WINDSP	Wind speed	km/d
	XELEV	Field elevation (not used)	m
	XLAT	Latitude	deg.
	XLONG	Longitude	deg.

3.1 Run Initialization

At the beginning of simulation, IPWTH is called to read four values from FILEIO. These are:

- FILEW – File name for weather data. This file is of type 'UFGA7801.WTH' for daily weather data or of type 'UFGA.CLI' for monthly weather parameters for use in generation of weather data.
- PATHWT – Directory path for location of weather data file.
- MEWTH - Switch for method of obtaining weather data. A value of 'G' or 'M' directs the routine to read the weather data file for daily data; 'S' indicates read SIMMETEO inputs and generate weather data; 'W' indicates read WGEN inputs and generate weather data. At this time there is no difference is processing for the 'G' and 'M' values.
- RSEED1 – A random number generator seed. This value is used only if the weather

generator function is specified by a MEWTH value of ‘W’ or ‘S’.

3.2 Seasonal Initialization

At the beginning of each season of simulation, IPWTH is called for initialization. If this is the second or subsequent season of a multi-season simulation (MULTI > 1), the name of the weather file is updated to reflect the current year of simulation. For example, if the first year of weather data was read from file ‘UFGA7801.WTH’ representing weather data observed in 1978, and if MULTI = 3 for this season of simulation, then the current weather data file will be updated to ‘UFGA8001.WTH’, which contains data observed in 1980. If this weather file does not exist, an error routine is called.

If this is the second or subsequent simulation of a sequenced run (MESIC = ‘S’) a similar file name modification is performed.

The pathname and filename are then concatenated into FILEWW.

Weather file header data is read to obtain the following information:

- CCO2 - Atmospheric CO2 concentration read from input file (ppm)
- INSI - Location code for weather data
- REFHT - Reference height for wind speed (m)
- TAV - Average annual soil temperature, used with TAMP to calculate soil temperature (°C)
- TAMP - Amplitude of temperature function used to calculate soil temperatures (°C)
- WINDHT - Reference height for wind speed (m)
- XELEV - Field elevation (m)
- XLAT – Latitude (degrees)
- XLONG – Longitude (degrees)

Default values of 1.5m and 2.0 m are used for REFHT and WINDHT if user input values are less than or equal to zero.

The weather file records are then read up to the day before the start of simulation (YRSIMM-1).

3.3 Rate Calculations

The IPWTH routine is called once per day if MEWTH is equal to ‘M’ or ‘G’. The next record is read from the weather file using the IGNORE routine, which ignores comment lines in the input file. If the end-of-file is reached (FOUND=0), the weather year is incremented and a new weather file opened.

Table 3 in lists the daily weather values read from FILEW by this routine.

The following checks are made on the data to ensure accuracy:

- A day of weather has not been skipped,
- Solar radiation is not zero or negative;
- Rainfall is not negative, and
- Maximum temperature is greater than minimum temperature.

If any of these conditions are met, an error routine is called and the program execution is halted.

Default values for dewpoint temperature and wind speed are used if the input values are less than or equal to zero.

3.4 Final

The final call to IPWTH at the end of all seasonal simulations closes the weather input file.

4.0 Weather Generation Routines (WGEN)

In some cases, the period of record of historical daily weather data is insufficient, and it is desirable to have the capability to generate synthetic weather data with the same statistical characteristics as the actual weather data at the location. The WGEN routines have been developed to generate daily weather values for precipitation, maximum temperature, minimum temperature and solar radiation for use by the CROPGRO model.

The weather generation routines are called if MEWTH in FILEIO is set equal to 'W' or 'S'. WGEN is called from the main weather routine for seasonal initialization (input of seasonal weather parameters) and rate calculation (generation of daily weather data). Table 5 lists the calling arguments for this routine. The output variables PAR, RAIN, SRAD, TDEW, TMAX, TMIN, and WINDSP represent the daily values of weather data generated by the model.
A first-order Markov chain model is used to describe the occurrence of rainfall on a given day and a gamma distribution function is applied to fit the amount of rainfall. The occurrence of rain on a given day will influence the temperature and solar radiation for that day and the probability of rain on the following day.

Two methods are provided for supplying the monthly statistics to the weather generation model. These are WGENIN, an adaptation of the WGEN model developed by Richardson and Wright (1984) and SIMMIN, which is based on the SIMMETEO model devloped by Geng et al. (1986). The WGENIN subroutine reads monthly means and standard deviations for weather generation parameters, while SIMMIN computes standard deviations from monthly summaries of rainfall data. Sufficient historical daily weather data is required for the WGENIN data and therefore the SIMMIN method may be more applicable where daily rainfall data is limited or unavailable.

The weather generation routines are not described in detail in this document. For more information on the methodologies used, refer to Geng, et al. (1986), Pickering, et al. (1994) and Richardson, et al. (1984).

Table 5 – Interface for WGEN Routines			
	Variable Name	**Definition**	**Units**
Input Variables	FILEW	Pathname plus filename for weather file (e.g. UFGA.CLI)	--
	MEWTH	Switch for method of obtaining weather data-- 'G' or 'M'- read weather data file 'S'- read SIMMETEO inputs and generate weather data 'W'- read WGEN inputs and generate weather data	--
	MULTI	Current seasonal simulation (=1 for first or only simulation, =NYRS for last simulation	--
	PATHWT	Directory path for weather file	--
	REPNO	Replication number for current simulation	--
	RNMODE	Run mode: I=interactive, Q=sequenced, N=seasonal	--
	RSEED1	Random number generator seed- user input	--
	RUN	Change in date between two observations for linear interpolation	
	YRDOY	Current day of simulation (year + day of year)	YYDDD
	YRSIM	Beginning date of simulation	YYDDD
Output Variables	PAR	Daily photosynthetically active radiation or photon flux density, calculated as half the solar radiation	MJ / m2 - d
	RAIN	Precipitation depth for current day	mm
	RSEED(I)	Random number generator seeds	--
	SRAD	Solar Radiation	MJ/m2-d
	TDEW	Dewpoint temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	WINDSP	Wind speed	km/d
	XLAT	Latitude	deg.
	XLONG	Longitude	deg.

5.0 Day Length Calculation (DAYLEN)

Subroutine DAYLEN computes solar day length, solar declination, and times of sunrise and sunset. The calculation of solar day length (DAYL) is based on equations reported by Spitters, et al. (1986). Table 6 lists the calling arguments for the routine.

Table 6 – Interface for DAYLEN subroutine

	Variable Name	Definition	Units
Input Variables	DOY	Day of year	d
	XLAT	Latitude of weather station	deg.
Output Variable	DAYL	Day length	hr
	DEC	Solar declination	deg.
	SNDN	Time of sun set	hr
	SNUP	Time of sun rise	hr

Solar declination (DEC) varies with the day of year (DOY) and is calculated by:

```
DEC = -23.45 * COS(2.0*PI*(DOY+10.0)/365.0)
```

This calculation has an amplitude of +/-23.45° with a minimum at day 355 (December 21) and a maximum at day 172.5 (June 21/22).

Variable SOC represents an intermediate calculation that is equal to the product of the tangent of declination and tangent of latitude. This is allowed to vary between –1 and 1 and is accurate for latitudes below the polar circles.

```
SOC = TAN(RAD*DEC) * TAN(RAD*XLAT)
SOC = MIN(MAX(SOC,-1.0),1.0)
```

Day length (DAYL) is then calculated by Spitters, et al. (1986) Equation 17. Times of sunrise and sunset (SNUP and SNDN) are equally spaced around solar noon.

```
DAYL = 12.0 + 24.0*ASIN(SOC)/PI
SNUP = 12.0 - DAYL/2.0
SNDN = 12.0 + DAYL/2.0
```

6.0 Atmospheric CO_2

In DSSAT v4.5, the atmospheric CO2 concentration can be specified in four ways based on the CO2 method code in Simulation Controls shown in Table 7.

Table 7 – Methods of obtaining atmospheric CO_2 values	
ICO2 value	**CO_2 method**
M	Use measured values found in CO2045.WDA file
D	Use static value of 380 ppm
W	Read annual value from weather file.
--	Weather modification entries (see next section)

The measured values (ICO2 = 'M') in file CO2045.WDA were obtained from the observations at Mauna Loa which have been filtered to remove seasonal patterns (Keeling, et al., 2001). Values from 1958 through present day are provided.

The static value (ICO2 = 'D') is also read from file CO2045.WDA. For this method, a static value of atmospheric CO_2 is maintained throughout the simulation. This value is currently set to 380 ppm, corresponding to atmospheric CO_2 in 2005.

There is also an option to read atmospheric CO2 from the weather file header (ICO2 = 'W'). This value is treated as a static value until the next weather file is input.

The fourth option for atmospheric CO2 involves modifying values with the weather modification option as discussed in the next section.

7.0 Weather Modification Routines

The weather modification feature can be used to modify historical or generated daily weather values by user-specified multipliers or increments. Three subroutines are included in file WTHMOD.FOR to accomplish this:

- WTHMOD – Adjusts weather variables by user specified amounts
- WTHSET – Updates weather modification array
- DECL – computes solar declination from adjusted day length

Weather adjustment values can be set in FILEX by the user for eight variables: day length (DAYL), solar radiation (SRAD), maximum and minimum daily temperatures (TMAX and TMIN), daily precipitation (RAIN), atmospheric carbon dioxide concentration (CO2), dewpoint temperature (TDEW), and average wind speed (WINDSP). The calling arguments for the weather modification routines are listed in Table 8.

Table 8 – Interface for weather modification subroutines

	Variable Name	Definition	Units
Input Variables	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	FILEW	Weather data file (e.g.; UFGA.CLI)	--
	XLAT	Latitude of weather station	deg.
	YRDOY	Current day of simulation (year + day of year)	YYDDD
Input/Output Variables	DAYL	Day length	hr
	PAR	Daily photosynthetically active radiation or photon flux density	MJ / m2 - d
	RAIN	Precipitation depth for current day	mm
	SRAD	Solar Radiation	MJ/m2-d
	TDEW	Dewpoint temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	WINDSP	Wind speed	km/d
Output Variables	CO2	Atmospheric carbon dioxide concentration	ppm
	DEC	Solar declination	deg.
	NEV	Number of environmental modification records	--
	SNDN	Time of sun set	hr
	SNUP	Time of sun rise	hr

7.1 WTHMOD

Subroutine WTHMOD is called twice by the main weather routine; once for input of weather modification data and again to set daily modification values.

7.1.1 Run Initialization

Weather modification factors are read from the Environmental Modification section of FILEIO. Table 9 lists the variables read in each record of data. A counter (NEV) accumulates the number of records of weather modification data.

7.1.2 Rate Calculations

WTHMOD is called for 'rate calculations' if NEV, the number of environmental modification records is greater than zero. At the beginning of this section, the current weather modification record is selected (denoted by variable I). Subroutine WTHSET

is called eight times to determine the additive and multiplicative factors for each of the eight variables that can be modified. These adjustment factors are stored in array WTHADJ.

Table 9 – Weather modification parameters read by WTHMOD

Variable Name	Definition	Units
WMDATE	Date for weather modification to take effect	YYDDD
DAYFAC	Type of adjustment for day length (DAYL) values (see note 1 below)	--
DAYADJ	Adjustment amount for DAYL values	--
RADFAC	Type of adjustment for solar radiation (SRAD) values (see note 1 below)	--
RADADJ	Adjustment amount for SRAD values	--
TXFAC	Type of adjustment for maximum daily temperature (TMAX) values (see note 1 below)	--
TXADJ	Adjustment amount for TMAX values	--
TMFAC	Type of adjustment for minimum daily temperature (TMIN) values (see note 1 below)	--
TMADJ	Adjustment amount for TMIN values	--
PRCFAC	Type of adjustment for precipitation (RAIN) values (see note 1 below)	--
PRCADJ	Adjustment amount for RAIN values	--
CO2FAC	Type of adjustment for atmospheric carbon dioxide concentration (CO2) values (see note 1 below)	--
CO2ADJ	Adjustment amount for CO2 values	--
DPTFAC	Type of adjustment for dew point temperature (TDEW) values (see note 1 below)	--
DPTADJ	Adjustment amount for TDEW values	--
WNDFAC	Type of adjustment for wind speed (WINDSP) values (see note 1 below)	--
WNDADJ	Adjustment amount for WINDSP values	--

(1) Types of adjustments for data: A = add adjustment amount to weather value, S = subtract adjustment amount from weather value, M = multiply weather value by adjustment amount, R = replace weather value with adjustment amount.

Each of the eight modification variables are checked to see if modification is required today, i.e., the additive value is non-zero or the multiplier is not equal to one. For day length adjustment, the following code performs the check and makes the adjustment to day length:

```
IF (WTHADJ(1,1) .NE. 0. .OR. WTHADJ(2,1) .NE. 1.) THEN
  DAYL = WTHADJ(1,1) + DAYL*WTHADJ(2,1)
```

In the case of day length, associated adjustments to time of sunrise, time of sunset and solar declination are also made:

```
  SNUP = 12.0 - DAYL/2.0
  SNDN = 12.0 + DAYL/2.0
  DEC = DECL(DAYL,XLAT)
ENDIF
```

For solar radiation, an associated adjustment to photosynthetically active radiation is also made:

```
IF (WTHADJ(1,2) .NE. 0. .OR. WTHADJ(2,2) .NE. 1.) THEN
```

```
      HOLD = WTHADJ(1,2) + SRAD*WTHADJ(2,2)
      IF (PAR .GT. 0.) PAR = PAR * HOLD/SRAD
      SRAD = HOLD
    ENDIF
```

Each of the other six adjustments are made with a single statement as no associated variables are modified.

7.2 WTHSET

Subroutine WTHSET is called eight times from WTHMOD to set the array of adjustment factors (WTHADJ) for modification of weather variables. Table 10 describes the computation of adjustment factors stored in the WTHADJ array.

Table 10 – Adjustment factors computation

Adjustment Type [1] TYPEC	**Additive Value [2,3] WTHADJ(1,VNUM)**	**Multipier [2,3] WTHADJ(2,VNUM)**
A (addition)	AMOUNT	1.0
S (subtraction)	–AMOUNT	1.0
M (multiplication)	0.0	AMOUNT
R (replacement)	AMOUNT	0.0
none	0.0	1.0

[1] TYPEC = DAYFAC, RADFAC, etc.

[2] AMOUNT = DAYADJ, RADADJ, etc.

[3] VNUM = 1 for DAYL, 2 for SRAD, etc.

7.3 DECL

In function subroutine DECL, the solar declination is back-calculated based on the adjusted day length. The equations used to compute day length (DAYL) in the DAYLEN subroutine are rearranged to compute declination.

8.0 Solar Radiation Calculations (SOLAR)

Subroutine SOLAR calculates the cloudiness factor (CLOUDS) for use in photosynthesis and evapo-transpiration routines. Normal extraterrestrial radiation (SON) is calculated for use in computing the hourly fractional diffuse radiation in the HMET routines. ISINB is returned from SOLAR as an intermediate variable required for calculation of hourly solar radiation, also in the HMET routines. This variable represents the integral portion of Equation 6 in the Spitters, et al. (1986) paper and is also defined by Equation 19 in that paper. Table 11 summarizes the input and output variables for this routine.

Table 11 – Interface for SOLAR subroutine

	Variable Name	Definition	Units
Input Variables	DAYL	Day length	hr
	DEC	Solar declination	deg.
	SRAD	Solar Radiation	MJ/m2-d
	XLAT	Latitude of weather station	deg.
Output Variables	CLOUDS	Relative cloudiness factor (0-1)	--
	ISINB	Integral of sin of BETA, in Spitter's Equation 6	--
	S0N	Normal extraterrestrial radiation (set equal to average solar constant; elliptical orbit ignored)	$J/m^2/s$

First, daily solar radiation is converted from MJ/m2 to J/m2 and saved as SRADJ. The product of the sine of solar declination and sine of latitude is saved as SSIN. CCOS is similarly calculated for the product of cosines. The variable SOC (sine over cosine) is calculated and is the same as SOC in the DAYLEN subroutine. It is recalculated here in the event that the weather modification routines have changed the value of day length.

DSINB represents the integral of the sine of solar elevation over the day and is used to calculate SOD, daily extraterrestrial irradiance based on Spitters equation 18.

The normal extra-terrestrial radiation (SON) is set equal to the solar constant of 1368.0 J/m2/s and is used to calculate SOD.

The Spitters paper demonstrates that under a standard clear sky, 77% of global irradiance is direct and 23% is diffuse. Thus the parameter AMTRCS is set to 0.77 to calculate SCLEAR, the clear sky radiation. CLOUDS, the relative cloudiness factor is calculated from the ratio of the actual solar radiation to clear sky radiation:

```
SCLEAR = AMTRCS * S0D * 1.E-6
CLOUDS = MIN(MAX(1.0-SRAD/SCLEAR,0.0),1.0)
```

ISINB is then calculated as defined by Equation 19 in the Spitters paper, for use in the hourly meteorological computation routines (HMET).

9.0 Hourly Meteorological Parameters (HMET)

The HMET routines compute hourly values of temperature, PAR, solar radiation, fraction diffuse radiation, and relative humidity for use by other CROPGRO modules. The routine is called once per day at the end of the 'Rate Calculations' section of WEATHR. Table 12 summarizes the calling arguments for these routines.

Table 12 – Interface for HMET subroutines

	Variable Name	Definition	Units
Input Variables	CLOUDS	Relative cloudiness factor (0-1)	--
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	DEC	Solar declination	deg.
	ISINB	Integral of sin of BETA, in Spitter's Equation 6	--
	PAR	Daily photosynthetically active radiation or photon flux density, calculated as half the solar radiation	MJ / m2 - d
	REFHT	Reference height for windspeed	m
	SNDN	Time of sunset	hr
	SNUP	Time of sunrise	hr
	S0N	Normal extraterrestrial radiation (set equal to average solar constant; elliptical orbit ignored)	$J/m^2/s$
	SRAD	Solar Radiation	MJ/m2-d
	TDEW	Dewpoint temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	WINDHT	Reference height for wind speed	m
	WINDSP	Wind speed	km/d
	XLAT	Latitude of weather station	deg.
Output Variables	AZZON	Hourly solar azimuth (+/- from South)	deg.
	BETA	Hourly solar elevation (+/- from horizontal)	deg.
	FRDIFP	Hourly fraction diffuse photon flux density after correcting for circumsolar radiation (Spitters, 1986)	--
	FRDIFR	Hourly fraction diffuse solar radiation after correcting for circumsolar radiation (Spitters, 1986)	--
	PARHR	hourly PAR	J / m2 - s
	RADHR	Total hourly solar radiation	J/m2-s
	RHUMHR	Relative humidity hourly value	%
	TAIRHR	Hourly air temperature (in some routines called TGRO)	°C
	TAVG	Average daily temperature	°C
	TDAY	Average temperature during daylight hours	°C
	TGRO	Hourly air temperatures	°C
	TGROAV	Average daily temperature (modified by photosynthesis routine)	°C
	TGRODY	Average temperature during daylight hours (modified by photosynthesis routine)	°C
	WINDHR	Hourly wind speed	m/s

At the beginning of daily calculations, average temperatures are initialized to zero and average windspeed is normalized to the reference height and converted to m/s.

A loop cycles through 24 hours to compute the hourly weather data. Subroutine HANG (Section 8.1) is called to compute solar elevation and azimuth for a given time of day. HTEMP (Section 8.2) distributes maximum and minimum temperatures over the 24 hour period.

Relative humidity is computed once hourly air temperatures are known:

```
RH = VPSAT(TDEW) / VPSAT(TAIRHR(H)) * 100.0
```

Where VPSAT is a function which calculates saturated vapor pressure of air at both the dewpoint temperature and current air temperature.

During daylight hours, wind speed is estimated to be:

```
WINDHR(H) = 2.0 * WINDAV / (1.0+DAYL/24.0)
```

where WINDAV is the average daily windspeed and DAYL is the daylength in hours. Windspeeds for nighttime hours are estimated at half this value.

Subroutine HRAD computes total hourly solar radiation. FRACD is called to compute hourly fraction diffuse total radiation and PAR, and HPAR calculates hourly PAR.

Average daily and average daytime temperatures (TAVG and TDAY) are accumulated and the number of daylight hours is counted (NDAY).

If a negative or zero value of PAR is computed, a default value of SRAD*2.0 is used.

9.1 HANG

HANG computes the solar elevation (BETA) and azimuth (AZZON) for the time of day as described by Spitters, et al. (1986). Figure 2 illustrates the daily variation of these variables for a latitude of 30°N on May 20.

The product of the sine of solar declination (DEC) and sine of latitude (XLAT) is saved as SSIN. CCOS is similarly calculated for the product of cosines.

```
SSIN  = SIN(RAD*DEC) * SIN(RAD*XLAT)
CCOS  = COS(RAD*DEC) * COS(RAD*XLAT)
```

HANGL is computed to be the hour angle or hourly time as a proportion of day and expressed as radians.

```
HANGL = (HS-12.0)*PI/12.0
```

BETA, the angle of the sun above the horizon, is calculated using Spitters equation (15):

```
BETAS = MIN(MAX(SSIN + CCOS*COS(HANGL),-1.0),1.0)
BETA  = ASIN(BETAS)
```

AZZON, the solar azimuth is then calculated using the following code:

```
SINAS = COS(RAD*DEC) * SIN(HANGL) / COS(BETA)
SINAS = MIN(MAX(SINAS,-1.0),1.0)
AZZON = ASIN(SINAS)
```

BETA and AZZON are then converted from radians to degrees.

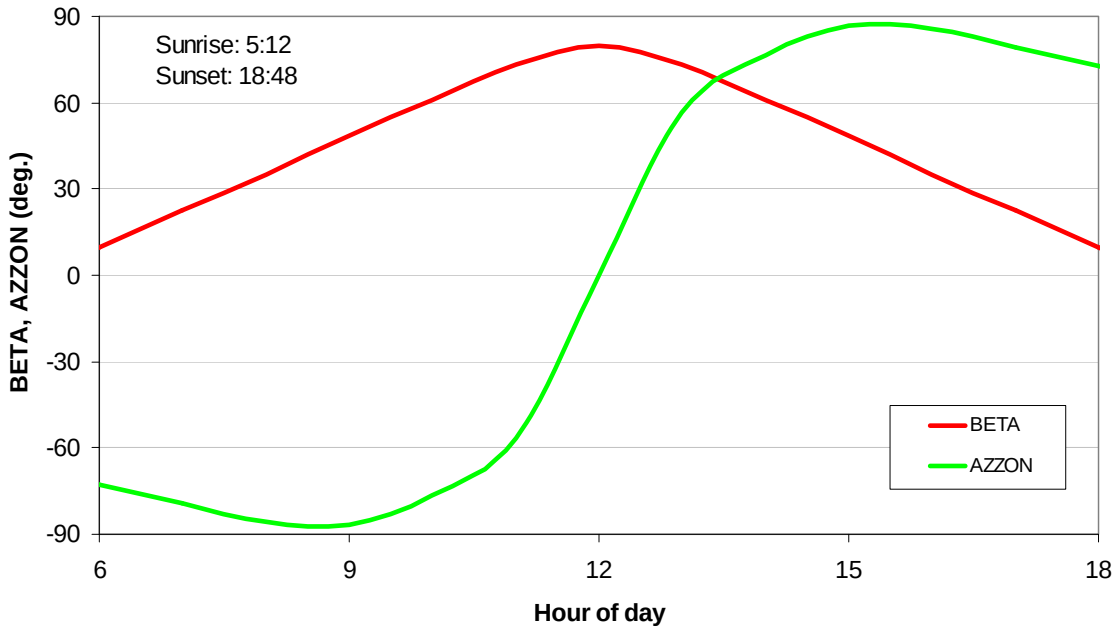


Figure 2 – Hourly Solar Angle Calculations for 30°N latitude on May 20 (day 140)

9.2 HTEMP

Hourly temperatures are calculated using a sine curve for temperatures during daylight hours and an exponential decay curve for decreasing temperatures from sunset to the minimum temperature. The time of maximum and minimum temperatures and the exponential decay characteristics are estimated as described by Parton and Logan (1981). Figure 3 illustrates the components of the hourly temperature computations for a day.

Coefficients A, B and C are defined as parameters at the beginning of the code based on values reported by Parton and Logan (1981):

```
PARAMETER (A=2.0, B=2.2, C=1.0, PI=3.14159)
```

The minimum daily temperature is assumed to occur one hour after sunrise. The maximum daily temperature is similarly assumed to occur at a fixed interval after solar noon.

```
MIN = SNUP + C
MAX = MIN + DAYL/2.0 + A
```

where MIN and MAX are the times of minimum and maximum temperatures, respectively.

The exponential curve is defined by a theoretical minimum temperature that would occur at an infinite time after sunset (TMINI) and an exponential decay coefficient (HDECAY):

```
T = 0.5 * PI * (SNDN-MIN) / (MAX-MIN)
```

```
TSNDN = TMIN + (TMAX-TMIN)*SIN(T)
TMINI = (TMIN-TSNDN*EXP(-B)) / (1.0-EXP(-B))
HDECAY = 24.0 + C - DAYL
```

Between the time of the minimum temperature and sundown, a sine curve describes the hourly air temperature distribution (TAIRHR).

```
T = 0.5 * PI * (HS-MIN) / (MAX-MIN)
TAIRHR = TMIN + (TMAX-TMIN)*SIN(T)
```

Before and after these times, an exponential decay curve defines the temperatures.

```
IF (HS .LT. SNUP+C) THEN
  T = 24.0 + HS - SNDN
ELSE IF (HS .GT. SNDN) THEN
  T = HS - SNDN
ENDIF
ARG = -B * T / HDECAY
TAIRHR = TMINI + (TSNDN-TMINI)*EXP(ARG)
```

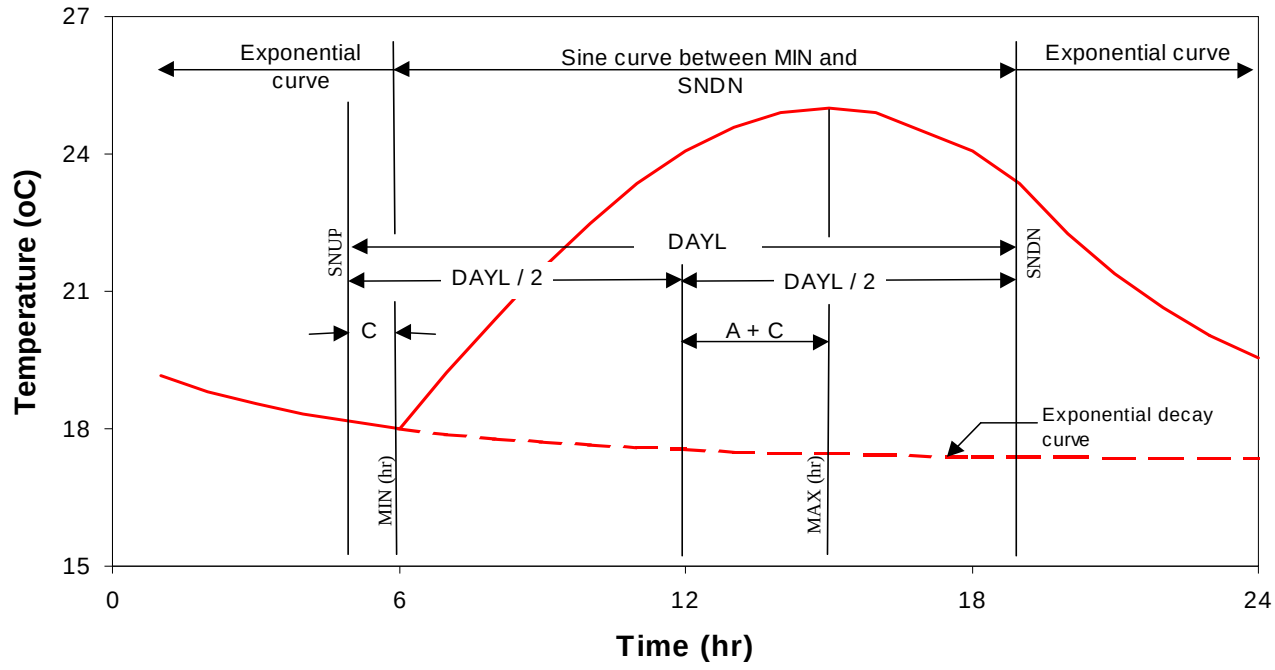


Figure 3 – Daily Temperature Distribution

9.3 HRAD

The HRAD subroutine computes a total hourly solar radiation based on Equation 6 in the Spitters (1986) paper. Figure 4 presents the hourly distribution of solar radiation for day 140 at 30° latitude.

The hourly solar radiation, RADHR, is computed from the hourly solar elevation (BETA), the daily solar radiation (SRAD) and the integral of the sine of BETA over day (ISINB) (Spitters, Eqn. 18), which was computed in the SOLAR subroutine:

```
SINB = SIN(RAD*BETA)
RADHR = SINB * (1.0+0.4*SINB) * SRAD*1.0E6 / ISINB
```

Before sun rise and after sun set, RADHR is set to zero.

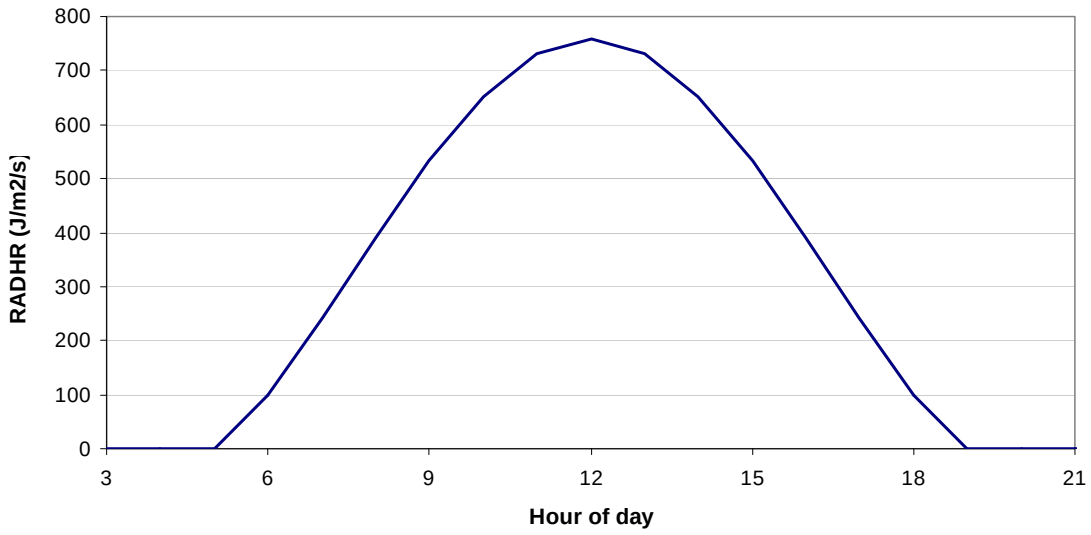


Figure 4 – Hourly solar radiation (subroutine HRAD)

9.4 FRACD

Subroutine FRACD computes hourly fraction diffuse total radiation (FRDIFR) and hourly fraction diffuse photon flux density (FRDIFP) based on the methods outlined by Spitters (1986). Figure 5 presents the hourly distribution of these variables for a specific latitude (XLAT = 30°N, day of year (DOY=140), and total daily solar radiation (SRAD=21.8 $MJ/m^2/d$). These variables are used by the canopy photosynthesis computation routines. The hourly solar elevation (BETA), which was calculated in subroutine HANG, is input to the subroutine and used to compute the following trigonometric relationships:

```
SINB = SIN(RAD*BETA)
COSB = COS(RAD*BETA)
COS90B = COS(RAD*(90.0 - BETA))
```

The ratio of solar to extraterrestrial radiation is computed as AMTRH:

```
S0 = S0N * SINB
AMTRH = RADHR / S0
```

where S0 = Extraterrestrial radiation on a plane parallel to earth surface (J/m2/s)
S0N = Normal extraterrestrial radiation (J/m2/s)
RADHR = Total hourly solar radiation (J/m2/s)

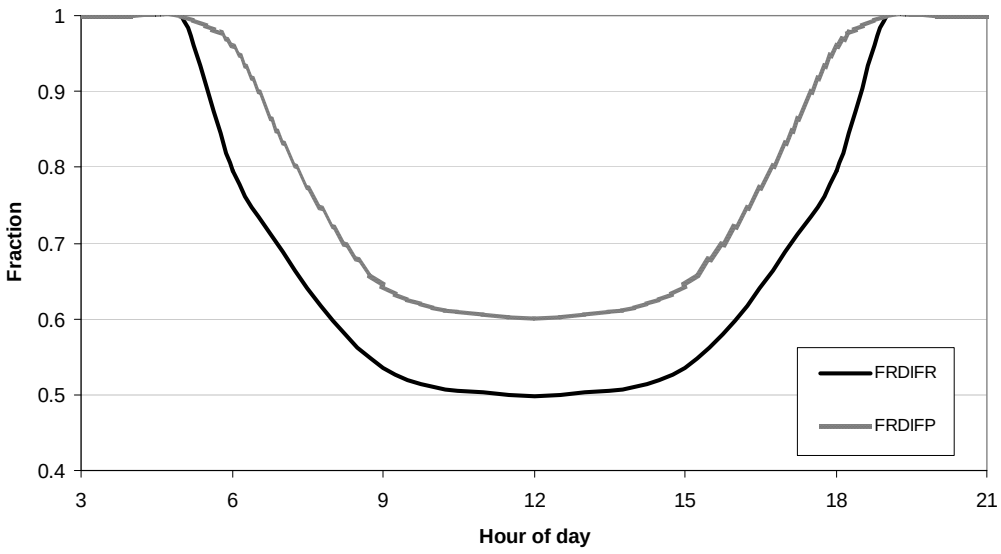


Figure 5 – Hourly distribution of fraction diffuse total radiation (FRDIFR) and fraction diffuse PAR (FRDIFP)

The hourly fraction diffuse solar radiation (FRDFH) is computed based on the value of AMTRH as illustrated in Figure 6. The following equations are used in the calculation:

```
IF (AMTRH .LE. 0.22) THEN
  FRDFH = 1.0
```

```
ELSE IF (AMTRH .GT. 0.22 .AND. AMTRH .LE. 0.5) THEN
  FRDFH = 1.0 - 4.3*(AMTRH-0.22)**2
ELSE
  FRDFH = 0.1 + 0.563*EXP(-5.5*(AMTRH-0.5))
```

Instantaneous diffuse global irradiance (SDF) is calculated as:

```
SDF = RADHR * FRDFH
CORR =  (1.0-CLOUDS) * COS90B**2 * COSB**3
SDF = SDF / (1.0 + CORR)
```

where CORR = a correction factor to decrease diffuse solar radiation by the circumsolar component which acts like direct radiation, and
CLOUDS = cloudiness factor computed in subroutine SOLAR

The hourly fraction of diffuse solar radiation (FRDIFR) is then computed as the ratio of SDF to total hourly solar radiation:

```
FRDIFR = SDF / RADHR
```

The fraction diffuse photon flux density (FRDIFP) is then calculated as (equation 10, Spitters):

```
FRDIFP = (1.0+0.3*(1.0-CLOUDS)) * FRDIFR

Both fractions are checked that values range between 0.0 and 1.0.
Before sunrise or after sunset, the fractional values are set to 1.0.
ENDIF
```

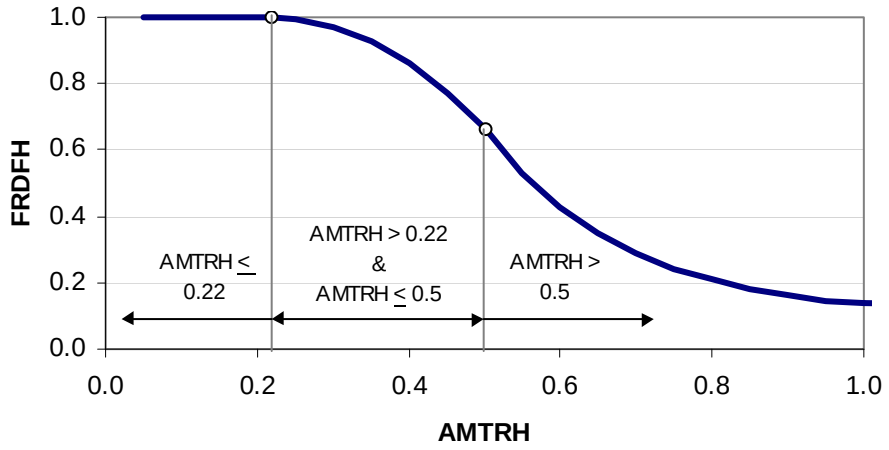


Figure 6 – Fraction diffuse radiation computation

9.5 HPAR

Subroutine HPAR computes hourly photosynthetically active radiation (PARHR) using methods outlined by Spitters (1986) and Pearcy (1991). If the value of PAR (photosynthetically active radiation) which was obtained in IPWTH or WGEN is non-zero, then a simple ratio is computed:

```
PARHR = RADHR * PAR/SRAD
```

where RADHR = Hourly solar radiation (J/m2/s), and
SRAD = Daily solar radiation (MJ/m2/d)

Otherwise, PARHR is computed to be two times RADHR.

9.6 VPSAT

Function subroutine VPSAT computes the saturated vapor pressure of air (VPSAT) based on current air temperature (T in °C) using the following equation:

```
VPSAT = 610.78 * EXP(17.269*T/(T+237.30))
```

9.7 VPSLOP

Function subroutine VPSLOP calculates the slope of the saturated vapor pressure versus temperature curve. This subroutine is not used in the weather routines, but is placed in file HMET because it is closely related to and calls function subroutine VPSAT.

10.0 References

Geng, S., F.W.T. Penning de Vries, and J. Supit. 1986. A simple method for generating daily rainfall data. J. Agric. Meteorol.

Keeling, C. D., S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, H. A. Meijer. Exchanges of atmospheric CO2 and 13CO2 with the terrestrial biosphere and oceans from 1978 to 2000. I. Global aspects, SIO Reference Series, No. 01-06, Scripps Institution of Oceanography, San Diego, 88 pages, 2001. http://scrippsco2.ucsd.edu/data/in_situ_co2/monthly_mlo.csv

Parton, W.J. and and J.A. Logan. 1981. A model for diurnal variation in soil and air temperature. Agric. For. Meteorol. 23:205.

Pearcy, R.W. 1991. Radiation and Light Measurements. In Plant Physiological Ecology. Field Measurements and instrumentation. 2nd ed. R.W. Pearcy, J. Ehleringer, H.A. Mooney and P.W. Rundel, eds. Chapman and Hall, New York, p. 97.

Pickering, N.B., J.W. Hansen, J.W. Jones, C.M. Wells, V.K. Chen, and D.C. Godwin. 1994. WeatherMan: A utility for managing and generating daily weather data. Agronomy Journal, 86:332-337.

Porter, C.H. and J.W.Jones. 1998. Modular Structure in CROPGRO Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Richardson, C.W. and D.A. Wright. 1984. WGEN: A model for generating daily weather variables. U.S. Department of Agriculture, Agricultural Research Service, ARS-8.

Spitters, C.J.T., H.A.J.M. Toussaint and J. Goudriaan. 1986. Separating the diffuse and direct component of global radiation and its implications for modeling canopy photosynthesis. Part I. Components of incoming radiation. Agric. For. Meteorol., 38: 217-229.

Volume 4
Chapter 5

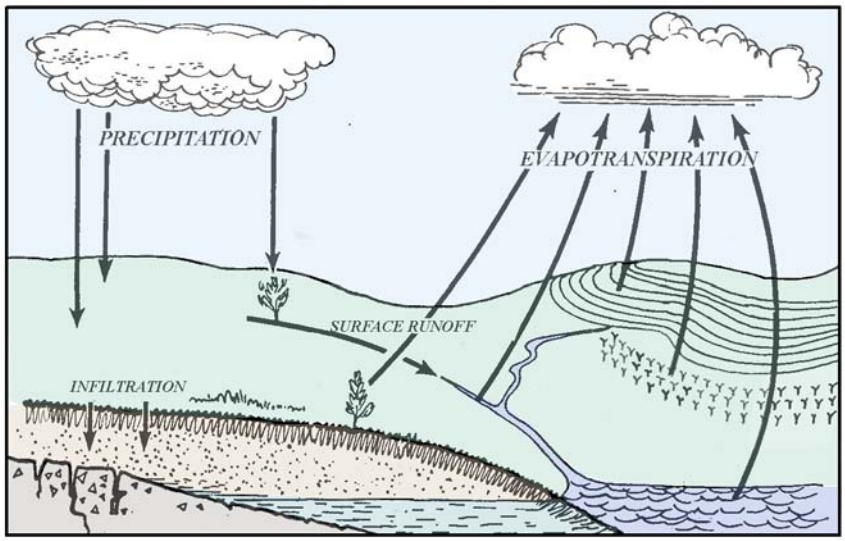
Soil Water Balance Module (WATBAL)

C.H. Porter
J.W. Jones
G. Hoogenboom
P. W. Wilkens
J.T. Ritchie
N.B. Pickering
K.J. Boote
B. Baer

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
International Center for Soil Fertility and Agricultural Development
International Consortium for Agricultural Systems Applications

SOIL WATER BALANCE MODULE

March 31, 2004



C.H. Porter, J.W. Jones, G. Hoogenboom, P. W. Wilkens,
J.T. Ritchie, N.B. Pickering, K. Boote, B. Baer

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
International Center for Soil Fertility and Agricultural Development
International Consortium for Agricultural Systems Applications

SOIL WATER BALANCE MODULE
TABLE OF CONTENTS

1.0 Soil Water Balance Model Overview 3

2.0 Soil water balance module structure 4

3.0 WATBAL Module 7
3.1 Run Initialization 8
3.2 Seasonal Initialization 8
3.3 Rate Calculations 8
3.4 Integration 10
3.5 Output 10
3.6 Final 11

4.0 Soil Water Drainage Routines (INFIL and SATFLO) 12

5.0 RNOFF – Surface Water Runoff Rate 18

6.0 SNOWFALL – Snow Melt and Accumulation 19
6.1 Seasonal Initialization 19
6.2 Rate Calculations 19

7.0 UPFLOW – Upward movement of Water 20

8.0 WBAL – Seasonal Water Balance 22
8.1 Seasonal Initialization 22
8.2 Final 22

9.0 References 23

SOIL WATER BALANCE MODULE

LIST OF FIGURES

Figure 1 - Soil Water Processes Modeled in WATBAL 3
Figure 2 - Water Balance Module Structure 5
Figure 3 - Calling Sequence for WATBAL 'Rate Calculations' Section 9
Figure 4 - Flowchart for INFIL Subroutine 14
Figure 5 - Flowchart for SATFLO Subroutine 17

LIST OF TABLES

Table 1 - Routines in WATBAL Module 6
Table 2 - Routines in WATBAL Module 6
Table 3 - WATBAL Interface Variables 7
Table 4 - Input and Output Variable Requirements for INFIL and SATFLO 12
Table 5 – Input and Output Data Requirements for RNOFF 18
Table 6 – Input and Output Data Requirements for SNOWFALL 19
Table 7 – Input and Output Data Requirements for UPFLOW 20
Table 8 – Input Variables for WBAL 22

1.0 Soil Water Balance Model Overview

The soil water balance module of the DSSAT models computes, on a daily basis, all processes that directly affect water content in the soil profile throughout the seasonal simulation. Ritchie (1985) describes many of these algorithms in detail. The change in soil water content for the soil profile is calculated on a daily time step using the equation:

$$\Delta S = P + I - EP - ES - R - D$$

Where ΔS = Change in soil water content
P = Precipitation
I = Irrigation
EP = Transpiration
ES = Soil Evaporation
R = Surface Runoff
D = Drainage from Soil Profile

Figure 1 illustrates the processes modeled. A maximum of 20 soil layers can be specified to represent the soil profile. Soil evaporation, root absorption, or flow to an adjacent layer can decrease the water content in any layer, while infiltration of rain, melted snow and irrigation water or flow from an adjacent layer can increase the water content in a layer.

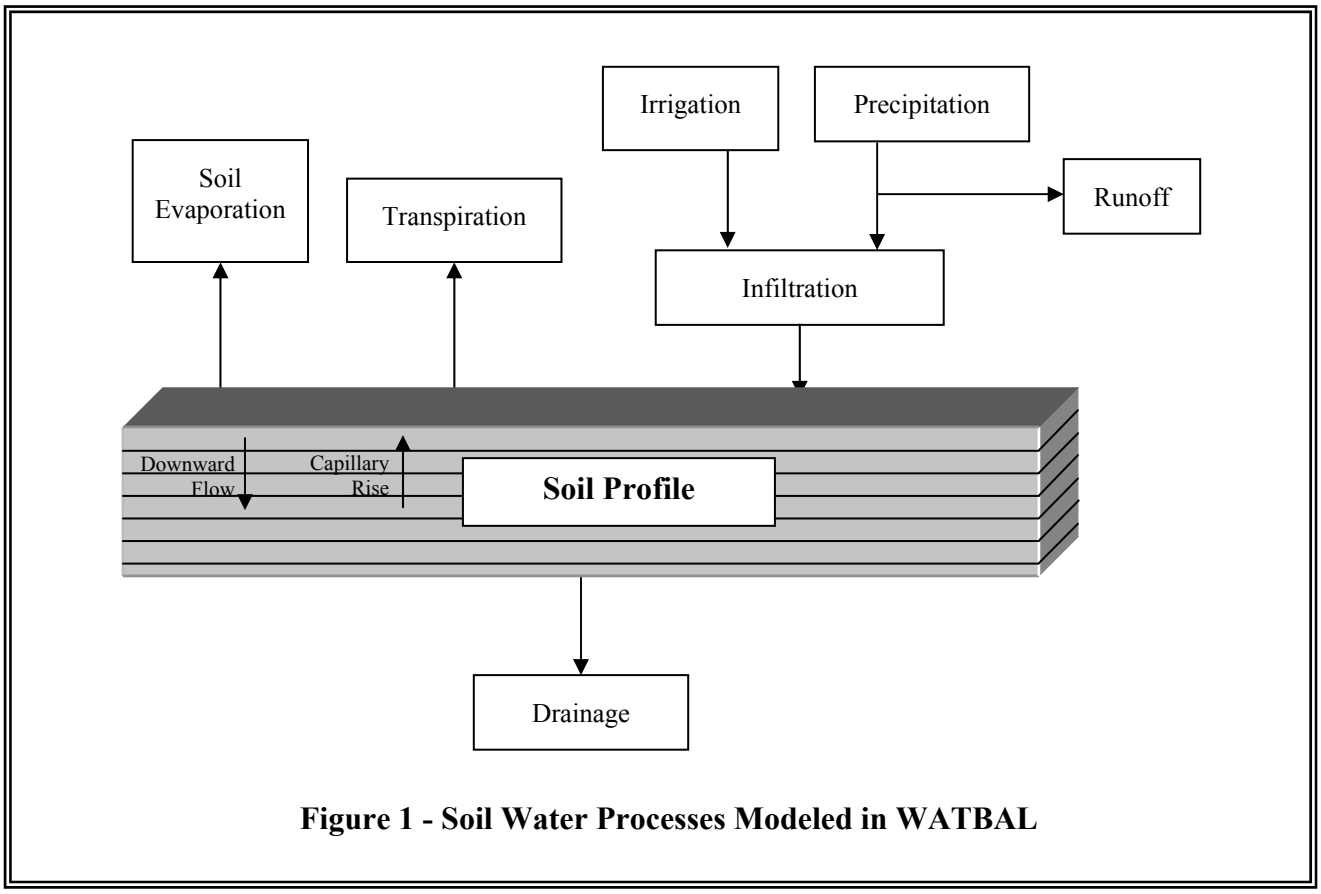


Figure 1 - Soil Water Processes Modeled in WATBAL

2.0 Soil water balance module structure

A complete description of the modular structure used in DSSAT-CSM is included in Chapter 1 of this Volume. The soil water balance module computes daily soil water content for each layer in the soil profile and is called from the Land unit module. The module is called a total of six times: once at the beginning of simulation for 'Run Initialization', once at the beginning of each season of simulation for 'Seasonal Initialization', three times daily for 'Rate calculations', 'Integration', and 'Output', and at the end of each season of simulation for 'Final' calculations.

Figure 2 presents the components of the module and the processing order within the main WATBAL module. As shown, the WATBAL module controls the flow of processing within the module and calls the other routines and sub-modules. Very few calculations are performed within the WATBAL main module; rather, the code is used to call other routines to perform computations.

Many of the routines called by the WATBAL module can be considered to be independent modules, as they meet the criteria presented by Jones, Hoogenboom, et. al. 2002. These routines perform two or more processing functions such as input and initialization of variables, rate calculation, integration of state variables, output of data and final processing.

The components of the WATBAL module are summarized in Table 2. Each of the main routines are discussed separately in this document beginning with the main module, WATBAL, in Section 3.0.

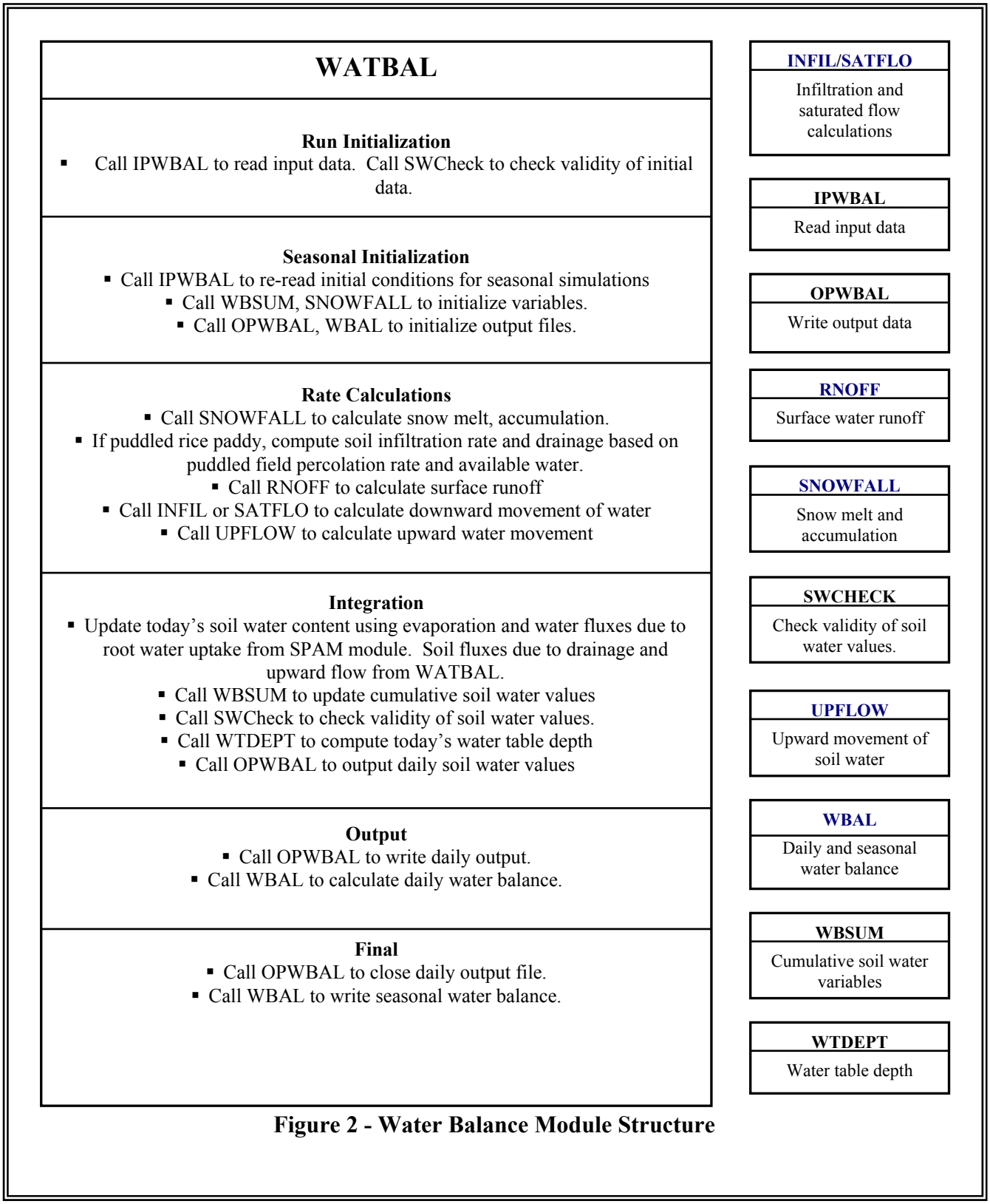


Figure 2 - Water Balance Module Structure

Table 1 - Routines in WATBAL Module

Table 2 - Routines in WATBAL Module

Name of Routine	Phase of Model in which Routine is Called						Description
	RUNINIT	SEASINIT	RATE	INTEGR	OUTPUT	FINAL	
IPWBAL	X	X					Reads input data from FILEIO (e.g., DSSAT40.INP)
SNOWFALL		X	X				Determines daily snowmelt and snow accumulation.
RNOFF			X				Calculates daily surface runoff.
INFIL			X				Calculates downward flow of water based on infiltration
SATFLO			X				Calculates downward flow of water based for saturated flow conditions
UPFLOW			X				Calculates upward movement of water in soil layers due to evapotranspiration.
SWCHECK	X			X			Checks validity of soil water content.
WBSUM		X		X			Performs daily summation of water balance variables.
WTDEPT				X			Determines water table depth.
OPWBAL		X		X	X	X	Writes daily output
WBAL		X			X	X	Writes seasonal water balance summary

3.0 WATBAL Module

The main code of the WATBAL module is used to control the flow of processing within the module. Few calculations are done within this routine, but rather, other routines are called to input data, initialize variables, perform rate calculations, and output data. However, integration of the main state variable for this module, volumetric soil water content (SW), is performed within the main module on a daily computational time step. Table 3 presents interface data required for the WATBAL module.

Table 3 - WATBAL Interface Variables

Variable Name	Definition	Units
Input Variables		
CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A..	--
ES	Actual soil evaporation rate	mm/d
IRRAMT	Irrigation amount for today	mm / d
ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
RAIN	Precipitation depth for current day	mm
SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
SWDELTX(L)	Change in soil water content due to root water uptake in layer L	cm3 [water] / cm3 [soil]
TMAX	Maximum daily temperature	°C
Input/Output Variables		
FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
Output Variables		
DRN(L)	Drainage rate through soil layer L	cm/d
FLOW(L)	Movement of water between unsaturated soil layers due to soil evaporation: + = upward, -- = downward	cm/d
SNOW	Snow accumulation	mm
SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
SWDELTS(L)	Change in soil water content due to drainage in layer L	cm3 [water] / cm3 [soil]
SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L	cm3 [water] / cm3 [soil]
WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d

Each of the six processing functions for the main module is discussed in the following sections. The subroutines and modules called by WATBAL are discussed separately in Sections 4.0 through 8.0.

3.1 Run Initialization

In the 'Run Initialization' section of the code, all input data needed within the soil water balance module are read from FILEIO. Subroutine IPWBAL (located in the WBSUBS.FOR file) is called to read this input data. The input file, FILEIO, is a temporary file generated by input module MINPT030.EXE and is called 'DSSAT40.INP' for single season simulations. It is generated from the user-specified FILEX and soils files.

Subroutine SWCheck is called to check that initial soil water content values are within allowable ranges. Warning messages are printed to file WARNING.OUT if values are unacceptable.

The next two statements initialize two components of the composite variable, FLOODWAT. A description of this, and other composite variables is given in Appendix A.

3.2 Seasonal Initialization

In the 'Seasonal Initialization' portion of the WATBAL code, IPWBAL is called to re-initialize soil water conditions for seasonal runs. Additional calls are made to subroutines, WBSUM and SNOWFALL for initialization of variables local to those routines. Subroutines OPWBAL and WBAL are called to open and write output file headings to the daily and seasonal output files, SoilWat.OUT and SoilWatBal.OUT, respectively.

3.3 Rate Calculations

Most of the calculations for the WATBAL module are performed within the 'Rate Calculations' section. All process rate variables are computed based on today's weather input and yesterday's state variable values.

Figure 3 illustrates the sequence of processing rate calculations for the water balance module and some of the main input and output variables, which are passed between the various routines.

If the maximum temperature is below 1°C or if there is an existing snow accumulation, the SNOWFALL routine is called to adjust snow accumulation (SNOW), or water available for infiltration and runoff (WATAVL). Otherwise, the WATAVL is simply set to the rainfall depth. This section is called even if water balance is not being simulated (ISWWAT = 'N') because snow accumulations can be an important factor for some winter crop simulations such as wheat.

The remainder of the 'Rate Calculations' section is done only if water balance is being simulated (ISWWAT = 'Y').

Drainage calculations are split into two conditions: puddled field or non-puddled field. If puddled conditions are being simulated for a lowland paddy field, such as rice, available water from the current flood pool and from irrigation and rainfall added today, is first used to fill remaining soil void space, if any. Drainage from the bottom of the soil profile is limited by the puddled percolation rate which is a user-specified input parameter in flooded

irrigation records of FILEX. This PUDPERC variable is a component of the FLOODWAT composite variable (see Appendix A).

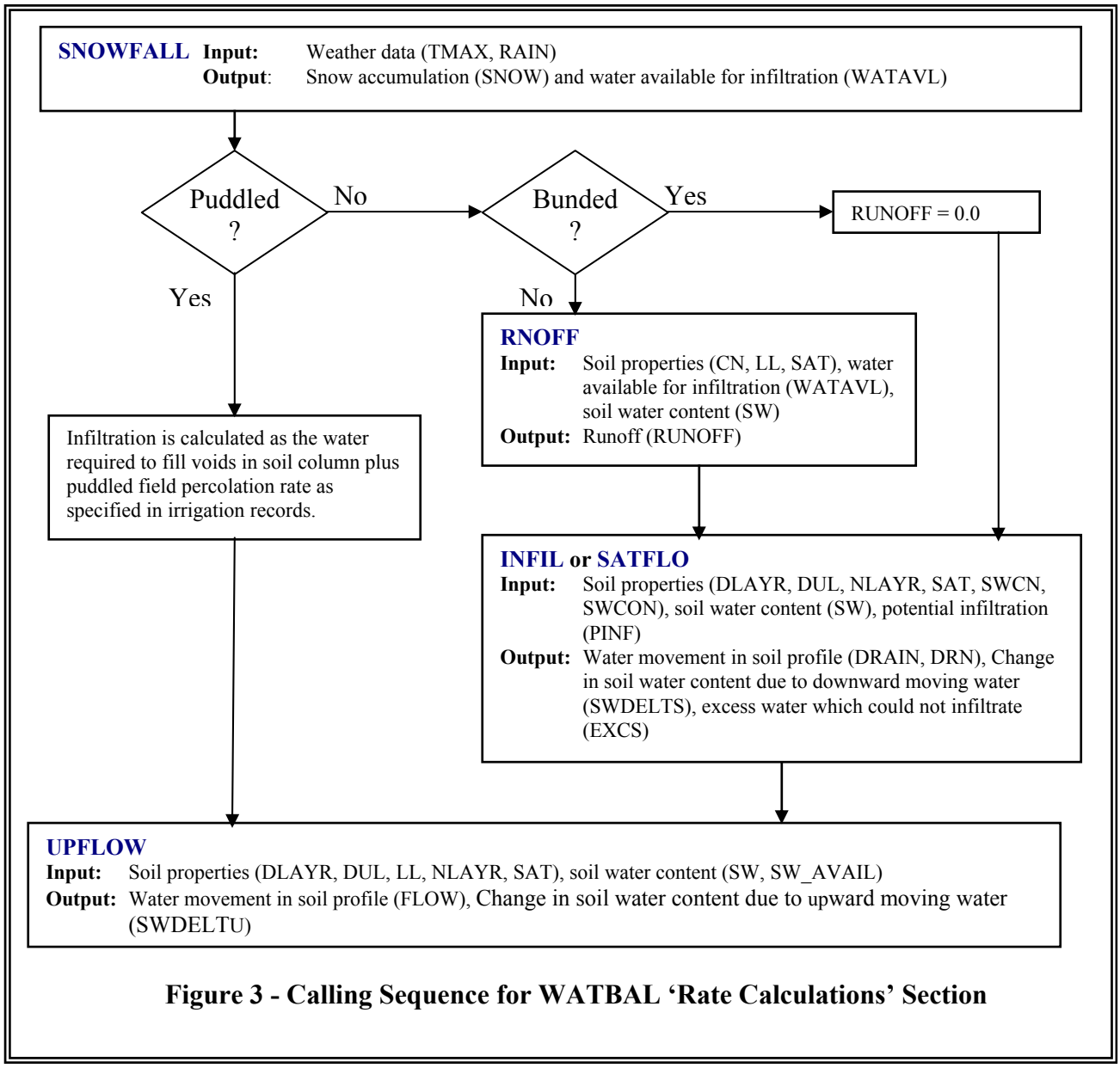


Figure 3 - Calling Sequence for WATBAL 'Rate Calculations' Section

For non-puddled conditions, the field can be either bunded (bounded by an earthen embankment to allow flooded conditions) or non-bunded; the major difference being that bunded fields will not allow runoff directly from the field. For non-bunded (upland) fields, RNOFF is called to compute surface runoff using the Soil Conservation Service (SCS, 1985) runoff curve number technique.

Values of PINF and WINF (potential infiltration in cm and mm, respectively) are computed:

```
WINF = WATAVL - RUNOFF
PINF = WINF * 0.1
```

The downward flow of water through the soil profile is computed by either INFIL, for days with rainfall or irrigation, or SATFLO. If the downward flow of water is impeded by saturated conditions or low soil water conductivity, an excess amount of water is returned in variable EXCS, which is then added to the previously calculated surface runoff rate.

A change in soil water content due to downward water movement (SWDELTS) is sent back to the main routine from INFIL or SATFLO. The actual values of volumetric soil water content (SW) are not modified in the 'Rate Calculations' section, but rather are updated at one time in the 'Integration' section of the WATBAL module.

Subroutine UPFLOW (found in WBSUBS.FOR) is called to compute the upward movement of water due to capillary flow. To prevent soil water fluxes that result in impossible values of soil water content, a "pseudo-integration" of soil water content is done which results in variable SW_AVAIL. The soil water flux due to upward movement of water is returned as SWDELTU. Again, values of SW are not changed at this point in the program.

3.4 Integration

The 'Integration' portion of the WATBAL module updates the volumetric soil water content (SW) based on changes calculated in the 'Rate' sections of WATBAL and the Soil-Plant-Atmosphere Module (SPAM). Soil evaporation (ES), which was calculated in the SPAM module is subtracted from the top soil layer after adjusting for units:

```
SW(1) = SW(1) - 0.1 * ES / DLAYR(1)
```

Next, soil water content is adjusted for drainage (INFIL or SATFLO), soil evaporation (UPFLOW), and root water extraction (SPAM).

```
DO L = 1, NLAYR
  SW(L) = SW(L) + SWDELTS(L) + SWDELTU(L) + SWDELTX(L)
ENDDO
```

Where `SWDELTS(L)` = Change in soil water content in layer L due to drainage (saturated flow)
`SWDELTU(L)` = Change in soil water content in layer L due to soil evaporation (upflow)
`SWDELTX(L)` = Change in soil water content in layer L due to root extraction (from SPAM module).

WBSUM is called to compute cumulative soil water values. Subroutine SWCheck is called to check that the volumetric soil water content values for each layer of soil are within the acceptable limits; i.e., between lower limit (LL) and saturated water content (SAT). No adjustments are made if the computed value is outside acceptable limits, but a warning message is sent to the WARNING.OUT file.

The current depth to the water table (WTDEP) is computed by WTDEPT. OPWBAL is called to accumulate some variables for the daily output.

3.5 Output

In the 'Output' section, OPWBAL is called to print daily output to the SoilWat.OUT file.

3.6 Final

The final section of the WATBAL code calls OPWBAL to close the soil-water output files and WBAL to produce a seasonal summary and water balance.

4.0 Soil Water Drainage Routines (INFIL and SATFLO)

The drainage routines calculate the downward flow of water based on infiltration and saturated flow and are called from WATBAL in the 'Rate Calculations' section. Table 4 lists variables that are input to and output from these routines.

Table 4 - Input and Output Variable Requirements for INFIL and SATFLO

Input Variables:	**Description**	**INFIL**	**SATFLO**
DLAYR(L)	Thickness of soil layer L (cm)	X	X
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L (cm3[water]/cm3[soil])	X	X
NLAYR	Actual number of soil layers	X	X
PINF	Potential water available for infiltration (cm/d)	X	
SAT(L)	Volumetric soil water content in layer L at saturation (cm3 [water] / cm3 [soil])	X	X
SW(L)	Volumetric soil water content in layer L (cm3 [water] / cm3 [soil])	X	X
SWCN(L)	Saturated hydraulic conductivity in layer L (cm/hr)	X	X
SWCON	Soil water conductivity constant; whole profile drainage rate coefficient (1/d)	X	X
Output Variables:	**Description**	**INFIL**	**SATFLO**
DRAIN	Drainage rate from soil profile (mm/d)	X	X
DRN(L)	Drainage rate through soil layer L (cm/d)	X	X
EXCS	Excess water to be added to runoff (cm/d)	X	
SWDELTS	Change in soil water content due to drainage in layer L (cm3 [water] / cm3 [soil])	X	X

The water content in each soil layer varies between a lower limit, LL(L), and a saturated upper limit, SAT(L). If the water content in a given layer is above the drained upper limit, DUL(L) specified for that layer, then water is drained to the next layer with the "tipping bucket" approach, using a drainage coefficient specified in the soil file.

Variables DRN(L) and DRAIN contain drainage volumes for individual soil layers and for the total soil profile, respectively.

At the beginning of the routines, a temporary soil water content variable, SWTEMP(L), is set equal to SW(L), the current soil water content, for all soil layers. This temporary variable is modified to compute rates of soil water movement between the layers due to infiltration and saturated flow. The original values of SW(L) are thus preserved to compute subsequent rate processes throughout the model.

If irrigation or rainfall occurs today, the INFIL subroutine is called by WATBAL to compute drainage. Otherwise, SATFLO computes drainage with no daily input of water. The use of the variable SWCN(L), the saturated hydraulic conductivity, in the subroutines INFIL and SATFLO allows for perched water tables in the profile and prevents the water flux from exceeding the most limiting layer below it.

At the end of both routines, SWDELTS(L) is calculated to be the difference in soil water content in each layer due to drainage through the soil profile:

```
DO L = 1, NLAYR
   SWDELTS(L) = SWTEMP(L) - SW(L)
ENDDO
```

4.1 INFIL - Infiltration

Figure 4 presents a flowchart of the algorithms used in subroutine INFIL. Infiltration water is distributed according to the water holding capacity of each soil horizon:

```
HOLD = (SAT(L) - SWTEMP(L)) * DLAYR(L)
```

If the potential inflow to a layer, PINF, exceeds the holding capacity of that soil layer, the water in excess of HOLD is passed directly to the layer below by saturated flow and the drainage from the soil profile in cm (DRCM) is calculated using SWCON, a conductivity parameter for the entire soil profile:

```
DRCM = SWCON * (SAT(L) - DUL(L)) * DLAYR(L)
DRN(L) = PINF - HOLD + DRCM
```

A

The coefficient SWCON varies between 0 and 1 and represents the fraction of water between the actual water content and the drained upper limit that drains from a soil layer in one day. Drainage is also limited by the saturated hydraulic conductivity for each soil layer, SWCN(L) (cm/hr):

```
IF (SWCN(L) .GT. 0.0 .AND. DRN(L) .GT. SWCN(L)*24.) THEN
   DRN(L) = SWCN(L)*24.
   DRCM = DRN(L) + HOLD - PINF
ENDIF
```

The change in soil water content is calculated as the difference between inflow from above, PINF, and drainage to a lower layer, DRN(L):

```
SWTEMP(L) = SWTEMP(L) + (PINF - DRN(L)) / DLAYR(L)
```

If the modified soil water content of a layer exceeds saturation, the excess water is distributed to the layers above. Any excess water after distribution will be added to surface runoff by means of variable EXCS. The new value of PINF (representing water moving into the layer below) is the sum of movement from the layer by both saturated and unsaturated flows:

```
PINF = DRN(L)
```

If the potential holding capacity of a soil layer is not exceeded by the potential infiltration from upper layers, the soil water content is updated with the entire infiltration amount:

```
SWTEMP(L) = SWTEMP(L) + PINF / DLAYR(L)
```

B

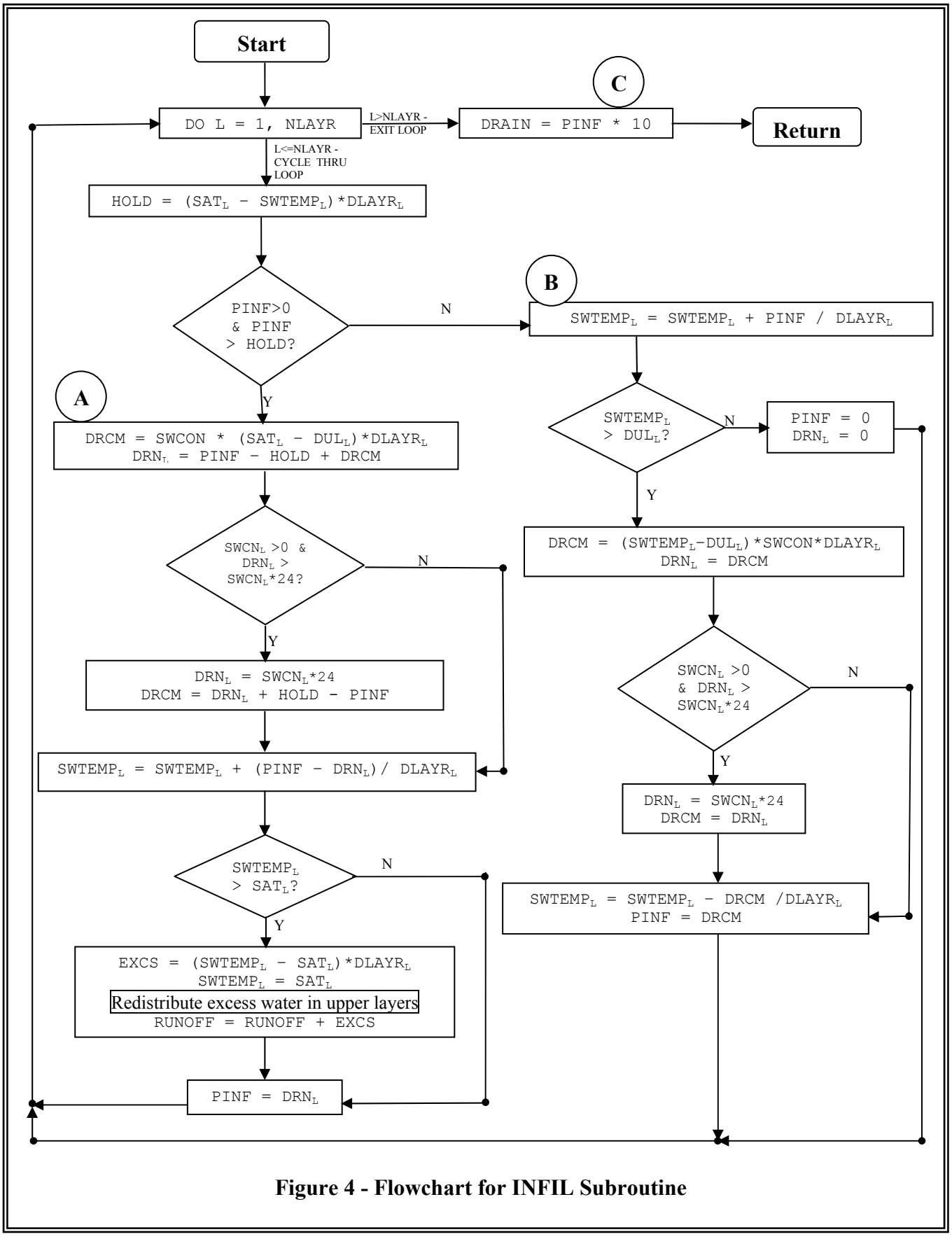


Figure 4 - Flowchart for INFIL Subroutine

If the updated soil water content is less than the drained upper limit, then no drainage to lower soil layers occurs and computation proceeds to the next layer:

```
PINF = 0.0
DRN(L) = 0.0
```

If the drained upper limit volumetric soil water in the layer (DUL) is exceeded, drainage by non-saturated flow (DRCM) is computed based on SWCON. Again, drainage to lower layers is limited by the saturated hydraulic conductivity for each soil layer. A new value of PINF is set equal to DRCM for computation of drainage within the next soil layer.

```
DRCM = (SWTEMP(L) - DUL(L)) * SWCON * DLAYR(L)
DRN(L) = DRCM
```

After movement through all soil layers has been calculated, any drainage from the bottom layer of the profile is set equal to PINF. For convenience, this value (times 10.0 to convert to mm) is stored in variable, DRAIN (mm/d), which then represents total saturated and unsaturated flow out of the lowest layer of the soil profile:

```
DRAIN = PINF * 10.0
```

4.2 SATFLO – Saturated Flow

Figure 5 presents a flowchart for the SATFLO subroutine. In subroutine SATFLO, saturated flow is calculated for days with no irrigation or rainfall. A maximum saturated flow, DRMX(L), from each layer is calculated for layers with soil water content above the drained upper limit:

```
DRMX(L) = (SWTEMP(L) - DUL(L)) * SWCON * DLAYR(L)
```

For the top soil layer, drainage from the layer is set equal to DRMX(1). The drainage from lower layers is calculated by the following equation, where HOLD is the amount of additional water the layer can hold:

```
DRN(L) = MAX(DRN(L-1) + DRMX(L) – HOLD , 0.0)
```

Drainage is limited by the saturated hydraulic conductivity for each soil layer, SWCN(L) in cm/hr:

```
IF (SWCN(L) .GT. 0.0 .AND. DRN(L) .GT. SWCN(L)*24.)
&       DRN(L) = SWCN(L)*24.
```

After drainage rates are calculated for each layer, the updated soil water content in each layer is computed.

```
SWOLD = SWTEMP(L)
SWTEMP(L) = SWTEMP(L) + (DRN(L-1) - DRN(L)) / DLAYR(L)
```

If soil water values exceed saturated water content, drainage from the upper layer is reduced for layers 2 through NLAYR:

```
IF (SWTEMP(L) .GT. SAT(L)) THEN
   DRN(L-1) = (SAT(L) - SWOLD) * DLAYR(L) + DRN(L)
   SWTEMP(L) = SAT(L)
ENDIF
```

The soil water content in the top layer is updated last:

```
SWTEMP(1) = SWTEMP(1) - DRN(1) / DLAYR(1)
```

The drainage from the soil profile is set equal to the drainage from the lowest layer and converted to mm:

```
DRAIN = DRN(NLAYR) * 10.0
```

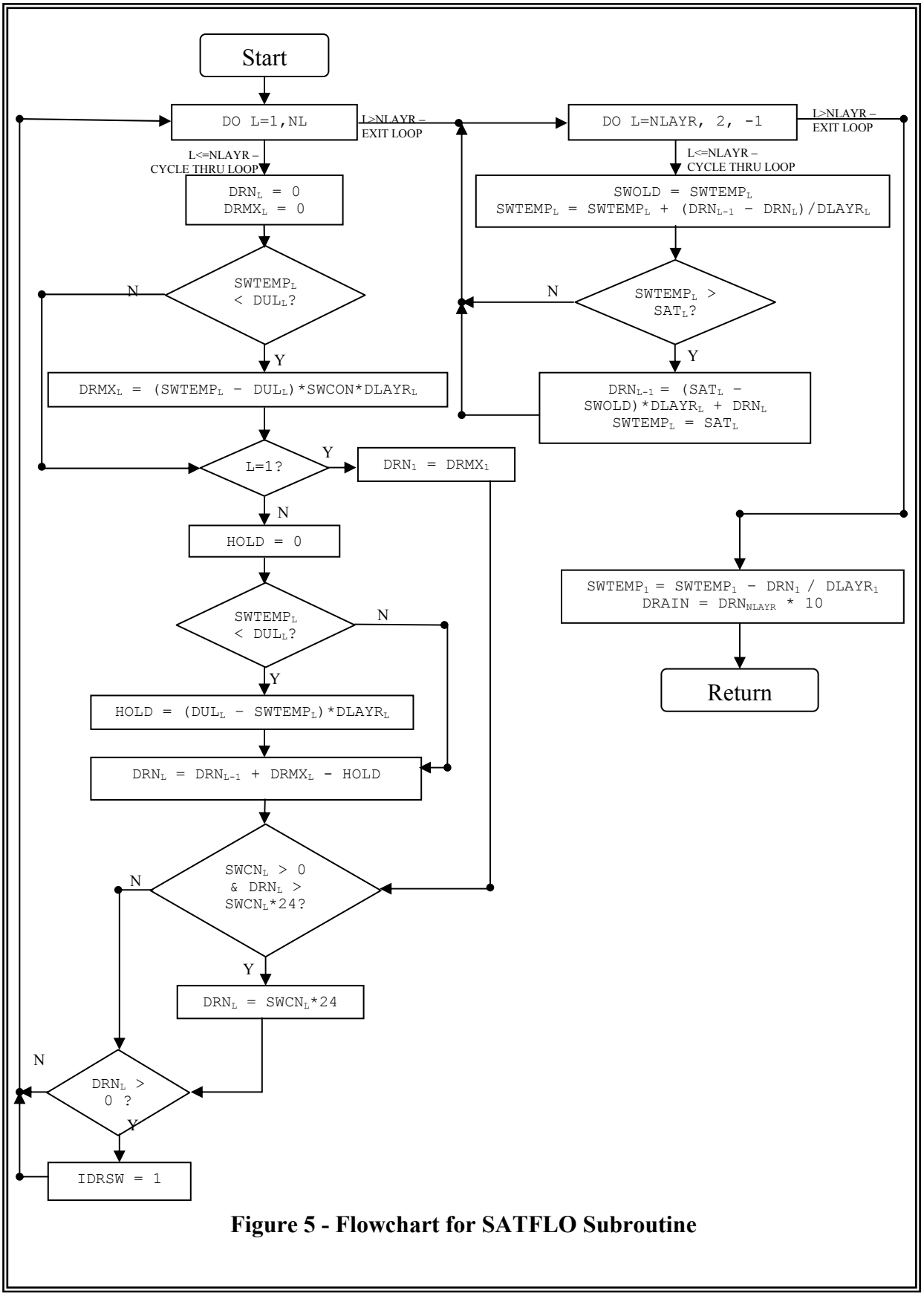


Figure 5 - Flowchart for SATFLO Subroutine

5.0 RNOFF – Surface Water Runoff Rate

The Runoff module calculates the daily surface water runoff rate using the USDA Soil Conservation Service (SCS) curve number technique (Soil Conservation Service, 1985), which estimates the total daily runoff rate from the total daily precipitation. The runoff curve number (CN) supplied as input, ranges from 0 (no runoff) to 100 (all runoff) based on land cover and soil type. This SCS technique was developed for a uniform soil (i.e. non-layered soil profile) and a single storm event simulation, but has been modified to account for daily variations in soil moisture content.

Table 5, below, lists input and output data requirements for RNOFF. This module is called once from the WATBAL main module for 'Rate Calculations'.

Table 5 – Input and Output Data Requirements for RNOFF

Input Variables:	
CN	Runoff Curve Number - measure of runoff potential based on soil type and current soil water content
LL(L)	Volumetric soil water content in soil layer L at lower limit (cm3 [water] / cm3 [soil])
SAT(L)	Volumetric soil water content in layer L at saturation (cm3 [water] / cm3 [soil])
SW(L)	Volumetric soil water content in layer L (cm3 [water] / cm3 [soil])
WATAVL	Water available for infiltration or runoff (rainfall plus irrigation) (mm/d)
Output Variables:	
RUNOFF	Calculated runoff (mm/d)

Variable SWABI is used to modify the soil storage conditions based on current soil moisture content:

```
 SWABI = 0.15 * ((SAT(1) - SW(1)) / (SAT(1) - LL(1) * 0.5) +
&      (SAT(2) - SW(2)) / (SAT(2) - LL(2) * 0.5))
```

The daily runoff rate, RUNOFF, is then computed using the SCS curve number equations:

```
   PB = WATAVL - SWABI * SMX
   IF (PB .GT. 0) THEN
      RUNOFF = PB**2/(WATAVL + (1.0-SWABI) * SMX)
   ELSE
      RUNOFF = 0.0
   END IF
```

6.0 SNOWFALL – Snow Melt and Accumulation

The SNOWFALL subroutine is called if maximum temperature is below freezing or if there is an accumulation of snow. SNOWFALL either calculates snowmelt, or accumulates additional snow based on maximum daily temperature and today's rainfall and irrigation. The value of WATAVL, water available for infiltration or runoff, is modified based on snowmelt computations. Table 6 lists calling arguments for the routine.

Table 6 – Input and Output Data Requirements for SNOWFALL

Input Variables:	
TMAX	Maximum daily temperature (°C)
RAIN	Daily precipitation depth (mm)
Output Variables:	
SNOW	Current snow accumulation (mm)
WATAVL	Water available for infiltration (mm)

6.1 Seasonal Initialization

In the 'Seasonal Initialization' portion of the code, snow accumulation (SNOW) is set to zero.

6.2 Rate Calculations

If the maximum daily temperature (TMAX) is greater than 1.0°C, then snowmelt occurs and the water available for infiltration (WATAVL) is increased:

```
SNOMLT = TMAX + RAIN*0.4
IF (SNOMLT .GT. SNOW) THEN
   SNOMLT = SNOW
ENDIF
SNOW   = SNOW - SNOMLT
WATAVL = RAIN + SNOMLT
```

Otherwise, snow is accumulated and water available for infiltration is reduced to zero:

```
SNOW   = SNOW   + RAIN
WATAVL = 0.0
```

7.0 UPFLOW – Upward movement of Water

Subroutine UPFLOW computes the changes in volumetric soil water content due to the upward movement of soil water in the profile (SWDELTU). A temporary soil water content variable, SWTEMP(L), is used in order to calculate changes in water content without modifying today's value of soil water content, SW(L). This preserves values of SW(L) for day (t-1) for computation of rates throughout the DSSAT models. The routine is called once from WATBAL for 'Rate Calculations'. **Table 7** lists the calling arguments.

Table 7 – Input and Output Data Requirements for UPFLOW

Input Variables:	
DLAYR(L)	Thickness of soil layer L (cm)
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L (cm3[water]/cm3[soil])
LL(L)	Volumetric soil water content in soil layer L at lower limit (cm3 [water] / cm3 [soil])
NLAYR	Actual number of soil layers
SAT(L)	Volumetric soil water content in layer L at saturation (cm3 [water] / cm3 [soil])
SW(L)	Volumetric soil water content in layer L (cm3 [water] / cm3 [soil])
SW_AVAIL(L)	Soil water content in layer L available for evaporation, plant extraction, or movement through soil (cm3 [water] / cm3 [soil])
Output Variables:	
FLOW(L)	Movement of water between unsaturated soil layers due to soil evaporation: + = upward, -- = downward (cm/d)
SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L (cm3 [water] / cm3 [soil])

The values of soil water content (SW) for each layer are stored in a temporary variable (SWTEMP) for purposes of computation of the upward flow of water in the soil profile, as was done in subroutines INFIL, and SATFLO. This preserves the value of the soil water state variable, SW, for use by other routines.

The computed upward flow cannot result in a soil water content above saturation (SAT(L)) or below the lower limit (LL(L)) in layer L or adjacent layers. The previously computed drainage volumes (from subroutines INFIL and SATFLO) are included in the soil water available for upward flow (SW_AVAIL(L)) which was passed from the main program in order to check these limiting soil water conditions. SW_AVAIL(L) and SW_INF(L) are used for computing these limitations.

The upward movement of water from the soil layers is calculated based on capillary flow theory (Jones, et al. 1986). The variables THET1 and THET2 represent the amount of volumetric soil water above the lower limits in layer L and (L+1), respectively. Water movement between layers, FLOW(L), is calculated with the following equations:

```
 DBAR = 0.88 * EXP(35.4*((THET1*DLAYR(L) + THET2*DLAYR(M))/
&    (DLAYR(L) + DLAYR(M)))*0.5)
 DBAR = MIN(DBAR, 100.0)

 GRAD = (THET2/ESW(M) - THET1/ESW(L)) * (ESW(M) * DLAYR(M) +
&    ESW(L)*DLAYR(L)) / (DLAYR(M) + DLAYR(L))

 FLOW(L) = DBAR * GRAD / ((DLAYR(L) + DLAYR(M)) * 0.5)
```

If THET1 is greater than THET2, FLOW is a negative number and water moves downward.

The upward flow rate, FLOW(L), exponentially increases with the difference in available soil water in the two adjacent soil layers (THET1 and THET2). At the drained upper limit of the lower soil layer, DUL(L+1), a maximum flow rate is assumed to occur.

Volumetric soil water in layers L and (L+1) are increased and decreased, respectively, by the amount of FLOW(L). Variable FLOWFIX is used to adjust the computed flow rates if soil water contents fall below the drained upper limit, or above saturation as a result of upward flow calculations.

At the end of the routine, the reported upward flow is the difference between the adjusted soil water content and the preserved value:

```
DO L = 1, NLAYR
  SWDELTU(L) = SWTEMP(L) - SW(L)
ENDDO
```

8.0 WBAL – Seasonal Water Balance

WBAL is called twice by the WATBAL routine for 'Seasonal Initialization', and 'Final' or seasonal output. Table 8 lists input data associated with this routine.

8.1 Seasonal Initialization

The SoilWatBal.OUT file is opened and headers are written.

8.2 Final

The 'Final' call to WBAL generates a seasonal water balance, which is computed as:

```
 WBALAN = (TOTIR * EFFIRR) + CRAIN - TDRAIN - TRUNOF -
&       CES - CEP - (TSW * 10) + (TSWINI * 10)
```

The summed variables represent cumulative values of effective irrigation (TOTIR * EFFIR), rainfall (CRAIN), drainage from the soil profile (TDRAIN), surface runoff (TRUNOF), soil evaporation (CES), plant transpiration (CEP), ending soil water content (TSW) and beginning soil water content (TSWINI). This is an end-of-season check on the conservation of mass equation presented earlier where the variable WBALAN should be equal or very close to zero.

Table 8 – Input Variables for WBAL

CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.
CRAIN	Cumulative precipitation (mm)
DLAYR(L)	Thickness of soil layer L (cm)
DRAIN	Drainage rate from soil profile (mm/d)
ES	Actual soil evaporation rate (mm/d)
FLOODWAT	Composite variable containing information related to bund management. See Appendix A.
IRRAMT	Irrigation amount for today (mm / d)
ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.
NLAYR	Actual number of soil layers
RAIN	Precipitation depth for current day (mm)
RUNOFF	Calculated runoff (mm/d)
SNOW	Snow accumulation (mm)
SWDELTS(L)	Change in soil water content due to drainage in layer L (cm3 [water] / cm3 [soil])
SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L (cm3 [water] / cm3 [soil])
SWDELTX(L)	Change in soil water content due to root water uptake in layer L (cm3 [water] / cm3 [soil])
TDRAIN	Cumulative daily drainage from profile (mm)
TRUNOF	Cumulative runoff (mm)
TSW	Total soil water in profile (cm)
TSWINI	Initial soil water content (cm)

9.0 References

Jones, C.A., J.T. Ritchie, J.R. Kiniry, and D.C. Godwin. 1986. *Subroutine Structure*. Ch. 4 in CERES-Maize: A Simulation Model of Maize Growth and Development. Eds. C.A. Jones and J.R. Kiniry. Texas A&M University Press.

Jones, J.W., Hoogenboom, G., Porter, C.H., Boote, K.J., Batchelor, W.D., Hunt, L.A., Wilkens, P.W., Singh, U. , Gijsman, A.J., Ritchie, J.T. 2002. *The DSSAT cropping system model*. European Journal of Agronomy. 18(3-4) 2002. : Pp. 235 -265.

Ritchie, J.T. 1985. A User Oriented Model of the Soil water Balance in Wheat. In Wheat Growth and Modeling. Eds. W. Day and R.K. Atykin. Plenum Press.

Soil Conservation Service. 1985. National Engineering Handbook Section 4: Hydrology. USDA.

Volume 4
Chapter 6

Inorganic Soil Nitrogen Module

UNDER CONSTRUCTION

Volume 4
Chapter 7

Inorganic Soil Phosphorus Module

J. W. Jones
U. Singh
C. H. Porter
K. A. Dzotsi
A. J. Gijsman
J.B. Naab
S.G.K. Adiku

University of Florida, Gainesville, FL
International Center for Soil Fertility and Agricultural Development
International Consortium for Agricultural Systems Applications

SOIL INORGANIC PHOSPHORUS MODULE (SoilPi)

May 1, 2008

J. W. Jones, U. Singh, C. H. Porter,
K. A. Dzotsi, A.J. Gijsman, J.B. Naab, S.G.K. Adiku

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Soil Inorganic Phosphorus Module
TABLE OF CONTENTS

1.0 Overview of the Soil Inorganic Phosphorus Module....3

2.0 Structure of the Soil Inorganic Phosphorus Module....3
2.1 Main Subroutine – SoilPi....4
Seasonal Initialization....4
Rate Calculations....5
Integration....7
Output and Final....7
2.2 Subroutine SoilPi_init....7

3.0 References....11

LIST OF TABLES

Table 1. Soil Inorganic Phosphorus Module Interface....4
Table 2. Fertilizer availability index....6
Table 3. P Extraction methods....8
Table 4. Criteria used in Subroutine SOILDYN for assigning soil types....8

LIST OF FIGURES

Figure 1. Inorganic Soil P Module Structure....5
Figure 2. Inorganic P transformations....7
Figure 3. Flowchart for initialization of P components based on measured extractable P....10

1.0 Overview of the Soil Inorganic Phosphorus Module

The soil inorganic phosphorus module computes labile, active and stable inorganic phosphorus quantities in the soil profile, as well as their rates of transformation on a daily time step (Dzotsi et al., 2010). Additions of inorganic P to the soil include fertilizer applications and mineralization of organic matter. Depletion of inorganic phosphorus occurs by plant uptake and immobilization to organic components.

This module interacts with the soil organic matter modules, the plant phosphorus module and the management operations module to determine the complete soil-plant phosphorus cycle for the simulation.

For the purposes of this model, it is assumed that plants take up only soluble phosphorus that is in direct contact (within 2 mm) of roots. Two soil volumes are computed daily which represent the volume of soil which is in direct contact with roots and the remaining soil volume which is not in contact with roots. The model tracks the three inorganic phosphorus species in these two soil pools. Modifications in quantities of P in each soil pool occur not only due to transformations between species, fertilizer applications and plant uptake, but also due to plant growth, which may shift soil volume, and therefore P, from the non-root soil pool to the root soil pool.

2.0 Structure of the Soil Inorganic Phosphorus Module

The module structure follows the guidelines described by Porter and Jones (1998) and Jones (2003) for inclusion in the DSSAT4 Cropping Systems Model. The routine is called from any plant growth routine for input and initialization of variables, computation of rates, integration of state variables and for daily and seasonal output.

Table 1 lists the interface variables for the module. The inputs include four composite variables:

- Control data, which provide information about the current simulation day and phase of simulation (e.g., initialization, rate calculation, etc.);
- Switch data which provide information on whether phosphorus is to be simulated as well as the level of output requested;
- Fertilizer data which include information on fertilizer dates and application rates; and
- Soil properties including soil layer thicknesses and chemical and physical properties of soil layers.

Additional input variables which affect the daily soil P balance include mineralization and immobilization rates, which are computed by the soil organic matter modules, plant P uptake, computed by the plant P module, and SVROOTS, which is the volume of soil which is in direct contact with roots. This last factor is important in computing the P which is available for uptake by plants.

Output from the module consists of P which is available for uptake by the plants, SPi_AVAIL, and total labile P, which is output for computation of daily rates of mineralization and immobilization of

P in the organic matter modules.

Figure **1** illustrates the components of the module. Three additional subroutines are called by the main module. These are:

- SoilPi_init – initializes all inorganic P species at the beginning of a simulation;
- OpSOILPi – daily output; and
- SoilPiBal – seasonal P balance.

Table 1. Soil Inorganic Phosphorus Module Interface

Variable Names	Definition	Units
Input:		
CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
FERTDATA	Composite variable which contains daily and cumulative information about fertilizer applications. See Appendix A.	
IMM(L,I)	Immobilization of element I (1=Nitrogen, 2=Phosphorus) in soil layer L on current day	kg[E]/ha/d
ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
MNR(L,I)	Mineralization of element I (1=Nitrogen, 2=Phosphorus) in soil layer L on current day	kg[E]/ha/d
PUptake(L)	Plant uptake of P from soil layer L	kg[P]/ha/d
SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
FracRts(L)	Fraction of soil volume in direct contact with roots in soil layer L	cm3/cm2
TillVals	Daily tillage operations data	
Output:		
SPi_AVAIL(L)	Inorganic soil P which is available for plant uptake in soil layer L	kg[P]/ha
SPi_Labile(L)	Labile soil P in soil layer L	kg[P]/ha

2.1 Main Subroutine – SoilPi

Seasonal Initialization

At the beginning of each season of simulation, subroutine SoilPi_init is called to initialize inorganic P variables. The initialization process is hierarchical and computation of initial values of P species depends on the soil chemistry information which the user is able to provide. Details of this procedure are covered in Section 2.2. Subroutine SoilPi_init returns the initial concentrations of stable, active, and labile P and the transformation rate coefficients for these soil P components.

Initial soil P quantities (kg/ha) are then converted to concentrations (ppm) for each layer and constituent.

Inorganic phosphorus is computed for both the soil which is in direct contact with roots (FracRts) and soil that is not in direct contact with roots. The initial soil volume in direct contact with roots is assumed to be zero at the beginning of each simulation. The entire mass

of inorganic P is thus initially assigned to the non-root soil zone.

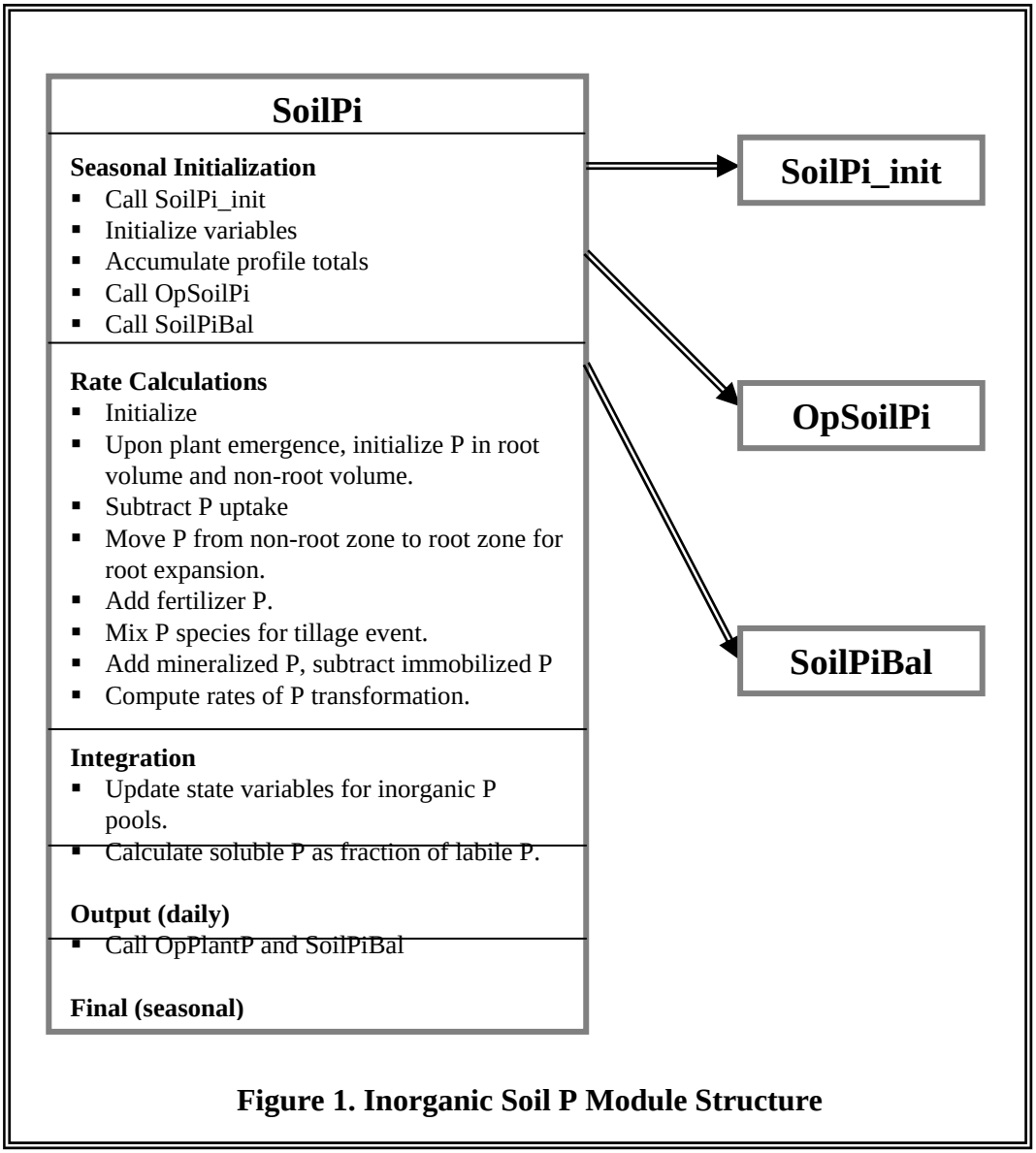


Figure 1. Inorganic Soil P Module Structure

Rate Calculations

Initialization upon emergence. Upon the first growth of roots, the mass of P in the root zone is established and P is proportioned to the two soil volumes.

Root uptake of P. P uptake by roots, calculated on the previous day in the Plant P module, is subtracted directly from the labile P in the root zone at the beginning of the rate section.

Root expansion. Increase in root soil volume due to expansion of roots brings a proportionate amount of P into the root zone pools. This increase to root pools and decrease to non-root

pools is stored in the rates of change variables, (DltSPiLabR, DltSPiLabNR, etc.).

Fertilizer application. On days for which P fertilizer applications are made, the fertilizer amount is added to the appropriate soil pools. If fertilizer is applied in bands or hills (BANDED = .TRUE.), then it is assumed that all of the P is applied directly into the root soil volume. For broadcast or other application types, the fertilizer is distributed proportionally to root and non-root zones.

The function subroutine, P_FERTINDEX, is used to determine the fertilizer P availability based on the work of Jones (1984) and Sharpley, et al. (1984, 1989). The fertilizer availability index is computed using soil composition as shown in Table 2 and is expressed as a fraction of fertilizer which enters the labile pool. The remaining fertilizer is added to the active pool.

Table 2. Fertilizer availability index

Soil type	**Fertility Index (P_FERTINDEX)**
Calcareous	`-0.0058 * CaCO3(L) + 0.60`
Slightly weathered	`0.0043 * TOTBAS + 0.0034 * Pi_LABILE + 0.11 * PH - 0.70`
Highly weathered	`-0.30 * ALOG10(CLAY) + 0.68`
Other (or insufficient data)	`0.75 + 0.00023**PiLabile(L)`

Mineralization and immobilization. Rates of mineralization and immobilization are computed in the soil organic matter modules (Century or SoilOrg) and input to the inorganic P and N modules. The net mineralization rate, PMINERAL, is computed for each layer of soil:

```
    IF (L == 1) THEN
!     First layer takes mineralization from surface also.
      PMINERAL = MNR(0,P) + MNR(1,P) - IMM(0,P) - IMM(1,P)              (1)
    ELSE
!     Add in residual from previous layer to preserve P balance
      PMINERAL = PMINERAL + MNR(L,P) - IMM(L,P)                         (2)
    ENDIF
```

Net mineralized P (PMINERAL > 0) is added proportionally to the roots and no-roots zones. Net immobilized P (PMINERAL < 0) is subtracted from labile P in the root and no-root zones. If insufficient P is available, the residual demand for immobilization is passed to the next lower layer in the variable PMINERAL in equation (2) in order to maintain the overall P balance.

P transformation rates. Rates of P transformation between labile and active pools and between active and stable pools follow a first order relationship as shown in Figure 2. These transformations are performed separately for the root zone and non-root zone soil pools. The rate coefficients, e.g., K_{LA} in Figure 2, are computed in the initialization section based on soil type and composition (see Section 2.2).

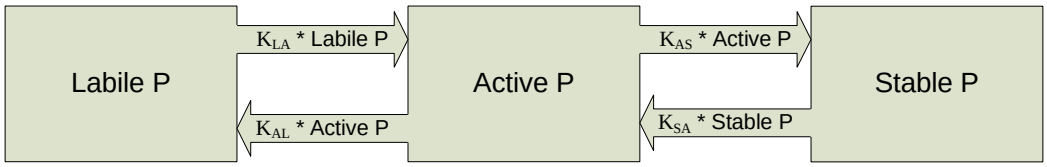


Figure 2. Inorganic P transformations

Integration

In the integration section of SoilPi, all of the inorganic soil P pools are updated based on rates of change calculated previously. Checks are performed to ensure that no negative values are computed. Soil profile totals are computed for each inorganic P pool. The P available for uptake by plants is computed as the labile P in the soil volume which is in direct contact with roots. Conversions from mass to concentration are computed for each constituent.

Output and Final

The daily output routine, OpSoilPi is called to produce the SoilPi.OUT file. Subroutine SoilPiBal is called to perform seasonal soil inorganic P balances as indicated by the user-specified IDETL from FILEX.

2.2 Subroutine SoilPi_init

This subroutine is called to estimate initial values for inorganic P constituents based on the available data that the user is able to supply. Two methods are available for estimating labile, active and stable P species and processing is done in a hierarchical manner in the following order:

1. Values of measured extractable P are used to estimate labile P based on soil type and P extraction method.
2. Labile P is estimated based on other general soil chemistry and soil type.

If the available data are insufficient to estimate P components for any of the three methods, the program stops with an error message.

Measured extractable P. If available, then estimation of labile P is based on direct measurements of extractable P. The logic flow for this estimation is shown in Figure 3. Also included in this figure are the default estimation methods, if no measured P data are supplied. Table 3 lists the available extraction methods and the P extraction method identification code (SMPX, from the soils file).

Soil type is an important determinant in the estimation of P constituents. This value is

assigned to each soil layer in the SoilDyn subroutine of CSM. The criteria used for this determination are listed in Table 4.

After computation of labile P, active and stable P quantities are estimated based on labile P and the P availability index.

P Availability Index. The phosphorus availability index, P_AVAILINDEX, is used as a measure of the activity level of the P in the soil. This factor is computed based on CaCO3 for calcareous soils; base saturation, labile P and pH for slightly weathered soils; clay content for highly weathered soils; and on labile P for all other cases.

Transformation rate constants. Transformation rates between inorganic P species are determined by first order equations. The rate coefficients are calculated for each soil layer based on the P availability index:

$$K_{LA} = 0.03 * \left[\frac{(1 - P)}{P} \right]^{0.5} \qquad K_{AS} = e^{-1.77 * P - 7.05}$$

$$K_{AL} = \frac{K_{LA}}{3} * P \qquad K_{SA} = 0.0001$$

Where P = P availability index

K_{LA} = rate constant for transformation from labile P to active P

K_{AL} = rate constant for transformation from active P to labile P

K_{AS} = rate constant for transformation from active P to stable P

K_{SA} = rate constant for transformation from stable P to active P

Table 3. P Extraction methods

PMethod	P Extraction method	PMethod	P Extraction method
SA001	Olsen	SA009	Water
SA002	Bray No. 1	SA010 *	IFDC Pi strip
SA003 *	Bray No. 2	SA011 *	pH in Water
SA004	Mehlich I (double acid, 1:5)	SA012 *	pH in KCl
SA005	Anion exchange resin	SA013 *	Mehlich 3
SA006	Truog	SA014	Morgan's solution
SA007	Mehlich I (double acid, 1:10)	SA015 *	NH4OAc
SA008	Colwell		
* Not currently supported in CSM			

Table 4. Criteria used in Subroutine SOILDYN for assigning soil types

SOILLAYERTYPE	Criteria
ANDISOL	Soil description or taxonomy includes the terms 'ANDOSOL' or 'ANDISOL' or 'VOLCAN' or 'ANDEPT'.
CALCAREOUS	CaCO3(L) > 0.15%
SLIGHTLYWEATHERED	CEC(L)/(CLAY(L)/100.) > 16.
HIGHLYWEATHERED	CEC(L)/(CLAY(L)/100.) <= 16.

UNSPECIFIED	Any other soils

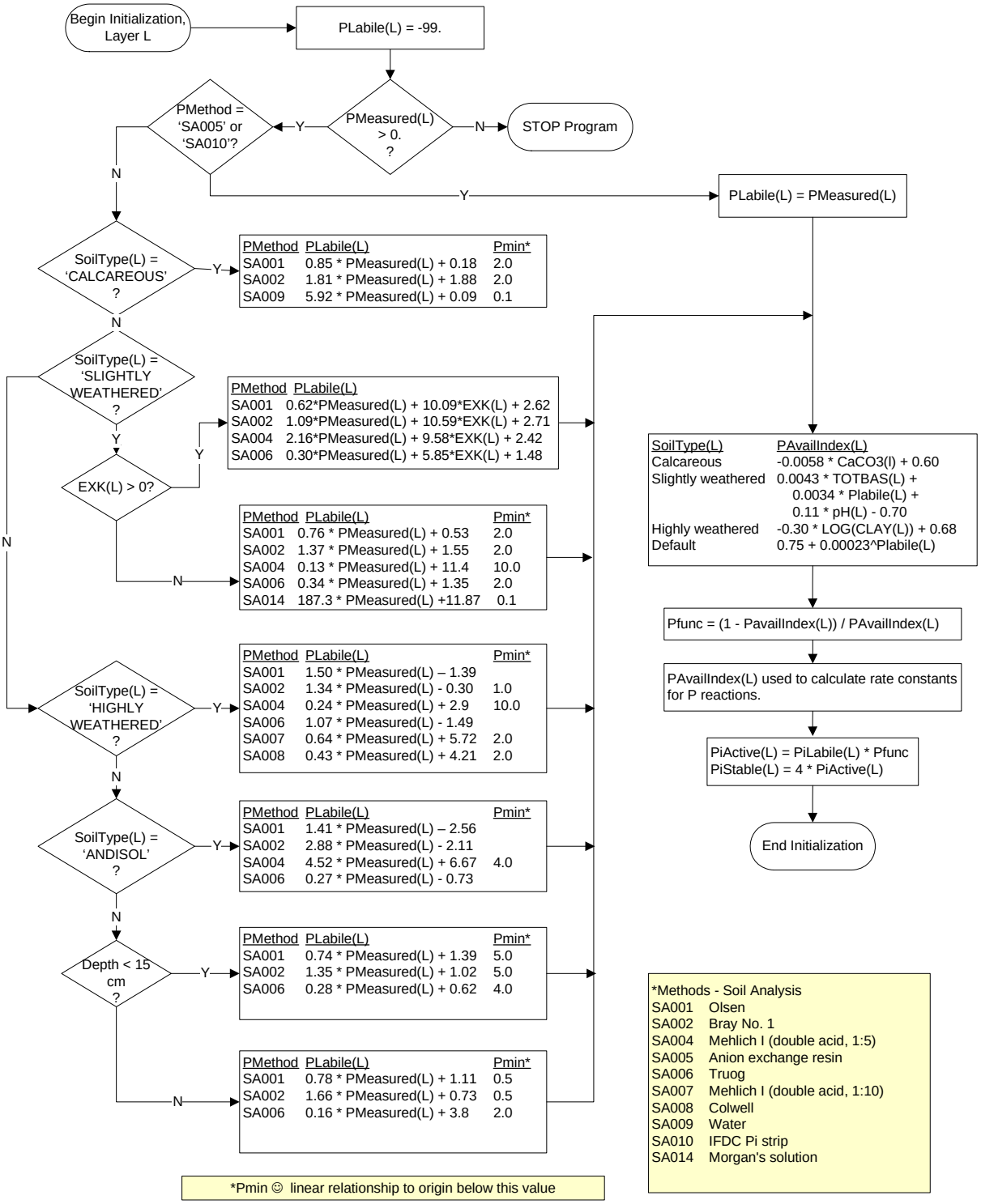


Figure 3. Flowchart for initialization of P components based on measured extractable P.

3.0 References

Dzotsi K.A., Jones J.W., Adiku S.G.K., Naab J.B., Singh U., Porter C.H., and Gijsman A.J., 2010. Modeling soil and plant phosphorus within DSSAT. Ecological Modelling 221: 2839-2849

Jones C.A., Cole C.V., Sharpley A.N. and Williams J.R. 1984a. A simplified soil and plant phosphorus model I. Documentation. Soil Sci. Soc. Am. J. 48:800-805.

Jones, J.W., G. Hoogenboom, C.H. Porter, K.J. Boote, W.D. Batchelor, L.A. Hunt, P.W. Wilkens, U. Singh, A.J. Gijsman, J.T. Ritchie. The DSSAT cropping system model. European Journal of Agronomy 18 (2003) 235- 265.

Porter, C.H. and J.W.Jones. 1998. Modular Structure in CROPGRO Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Sharpley A.N., Jones C.A., Gray C. and Cole C.V. 1984. A simplified soil and plant phosphorus model II. Prediction of labile, organic, and sorbed phosphorus. Soil Sci. Soc. Am. J. 48:805-809.

Sharpley A.N., Singh U., Uehara G. and Kimble J. 1989. Modeling soil and plant phosphorus dynamics in calcareous and highly weathered soils. Soil Sci. Soc. Am. J. 53:153-158.

Singh, U. 1985. A crop growth model for predicting corn (Zea mays L.) performance in the tropics. Ph.D. thesis. Univ. of Hawaii, Honolulu.

Volume 4
Chapter 8

CENTURY-based Soil Organic Matter Module

A.J. Gijsman
C.H. Porter
P.W. Wilkens
G. Hoogenboom

UNDER CONSTRUCTION

University of Florida, Gainesville, FL
The University of Georgia, Griffin, GA
International Center for Soil Fertility and Agricultural Development
International Consortium for Agricultural Systems Applications

MODELING

Modifying DSSAT Crop Models for Low-Input Agricultural Systems Using a Soil Organic Matter–Residue Module from CENTURY

Arjan J. Gijsman,* Gerrit Hoogenboom, William J. Parton, and Peter C. Kerridge

ABSTRACT

In low-input systems, where most nutrients become available from soil organic matter (SOM) and residue turnover, the applicability of DSSAT (Decision Support System for Agrotechnology Transfer) crop simulation models is limited because (i) it recognizes only one type of SOM (i.e., humus) and recently added, but not yet humified, residue; (ii) it does not recognize a residue layer on top of the soil; (iii) newly formed humus is given a fixed C/N ratio of 10; (iv) only one litter pool is recognized for N although three are recognized for C; (v) for residues with C/N ratios <25, the three litter pools for C decompose at a rate that is independent of the residue's N concentration; and (vi) SOM and residue flows are independent of soil texture. A SOM–residue module from the CENTURY model was incorporated in the DSSAT crop simulation models, and a residue layer was added on top of the soil. Modifications were also made in the senescence module of CROPGRO, a model within DSSAT, so that senesced material is now added daily to the soil. Evaluation of the model, using a data set of 40 yr of bare fallow, showed an excellent fit [product moment correlation coefficient (*r*) of 0.983] between simulated and measured values for SOM-C. Soil N from decomposing SOM and residues was evaluated with data from a Brazilian experiment with seven leguminous residue types. By incorporating the CENTURY SOM–residue module, DSSAT crop simulation models have become more suitable for simulating low-input systems and conducting long-term sustainability analyses.

In developing countries, detailed and long-term field experiments are often difficult to conduct due to financial or personnel limitations. The application of simulation models, which have been developed with data measured under more accommodating conditions and whose mathematical relationships apply to a wide range of conditions, is therefore an attractive option. However, just using an *off-the-shelf* model will not necessarily lead to results that are applicable to a local situation, especially when the model is being applied in a situation that is outside its tested range of conditions and where crucial differences in agricultural systems or environmental conditions exist.

The Decision Support System for Agrotechnology Transfer (DSSAT; Tsuji et al., 1994) is a comprehensive decision support system for assessing agricultural management options. It has been widely used in both developed and developing countries (Algozin et al., 1988; Bowen and Wilkens, 1998; Jagtap et al., 1993; Lal et al., 1993; Singh et al., 1993; Thornton and Wilkens, 1998). DSSAT version 3.5 incorporates 16 crops {maize (*Zea mays* L.), wheat (*Triticum aestivum* L.), rice (*Oryza sativa* L.), sorghum [*Sorghum bicolor* (L.) Moench], millet [*Pennisetum typhoides* (Burm.) Stapf & Hubb.], barley (*Hordeum vulgare* L.), bean (*Phaseolus vulgaris* L.), soybean [*Glycine max* (L.) Merr.], peanut (*Arachis hypogaea* L.), chickpea (*Cicer arietinum* L.), cassava (*Manihot esculenta* Crantz), potato (*Solanum tuberosum* L.), sugarcane (*Saccharum officinarum* L.), tomato [*Lycoersicon lycopersicum* (L.) Karsten], bahiagrass (*Paspalum notatum* Fluegge), and sunflower (*Helianthus annuus* L.)}, with several more under development. The model handles management strategies that involve crop rotations, irrigation, fertilization, and organic applications. Although crops (or cultivars) and crop management (e.g., mechanization) may differ from country to country or even from village to village, the effect of fertilizer or irrigation on crop production is likely to follow similar biophysical and biochemical pathways. However, there is an important difference between high-input and low-input agricultural systems. In the former, almost all nutrients required by crops are supplied by chemical fertilizers, whereas in the latter, nutrients become available mainly through decomposition of soil organic matter (SOM) and plant residues.

The DSSAT crop simulation models include a module for simulating SOM and residue dynamics, which is based on the PAPRAN model (Seligman and Van Keulen, 1981). Godwin and Jones (1991) adapted PAPRAN's SOM–residue section for the CERES model, one of the main crop simulation models of DSSAT (see also Godwin and Singh, 1998). This module has been called the "CERES-based SOM–residue module" and was also incorporated in the CROPGRO model of DSSAT (Boote et al., 1998).

The PAPRAN model was developed for annual pastures and small-grain crops in a semiarid environment. Its SOM–residue module, therefore, may not apply to very different systems, as Seligman and Van Keulen (1981, p. 195) state "the application of the model is

A.J. Gijsman and P.C. Kerridge, Centro Internacional de Agricultura Tropical (CIAT), Apartado Aéreo 6713, Cali, Colombia; G. Hoogenboom, Dep. of Biol. and Agric. Eng., Univ. of Georgia, Griffin, GA 30223; and W.J. Parton, Nat. Resour. Ecology Lab., Colorado State Univ., Fort Collins, CO 80523. Received 12 July 2000. *Corresponding author (a.gijsman@cgiar.org).

Published in Agron. J. 94:462–474 (2002).

Abbreviations: AEC, anion exchange (adsorption) capacity; FOM, fresh organic matter; FON, fresh organic matter nitrogen (or residue N); HUMN, humus nitrogen; RMSE, root mean square error; SOM, soil organic matter; SOM1, active (microbial) soil organic matter; SOM2, intermediate soil organic matter; SOM3, passive soil organic matter.

limited to situations which do not differ widely from those for which the model has been calibrated." Moreover, the rather low level of resolution of the model, with a black-box approach for many processes, raises particular concern for its application to systems where decomposing SOM and residues are the main source of nutrients for a crop. Major limitations of the CERES-based SOM–residue module are:

1. It recognizes only recently added, but not yet humified, residue [fresh organic matter (FOM)] and one type of SOM (humus). Most SOM models recognize more than just one SOM pool (Jenkinson et al., 1992; Li et al., 1994; Parton et al., 1994; Van Veen et al., 1984; Verberne et al., 1990). For example, old, almost inert, organic material with a very low turnover rate vs. young, often microbial, material with a very rapid turnover. Campbell et al. (1994) split the one-pool humus into a slowly decomposing fraction and a more rapid fraction, which resulted in a closer fit between simulated and measured data.
2. Newly formed SOM is given a fixed C/N ratio of 10. This may be a good average value, yet it is too restrictive for application to any soil and all conditions of N availability.
3. It does not account for a residue layer that is located on the soil surface. Residues either have to be incorporated, or residue that is placed on the surface is not accounted for. However, in legume-based rotational systems, which are very common in the tropics and low-input agricultural systems, a thick layer of senesced shoot parts may lie on top of the soil. Although this layer is never incorporated physically by the farmer through plowing or other land preparations, it becomes the main source of N for the following crop. Moreover, small farmers often prefer not to plow because it is labor intensive, particularly if they lack adequate equipment. In addition, incorporation of residues increases the risk for soil erosion.
4. Although the CERES-based module distinguishes three litter C pools (carbohydrates, cellulose, and lignin), there is only one litter N pool. This implicitly means that decomposition of a certain amount of lignin has the same effect on litter N as decomposition of a similar amount of cellulose or carbohydrates. In reality, these components have very different N concentrations.
5. For residues with C/N ratio <25, the three litter pools decompose at a rate that is independent of the residue's N concentration. This may be too limiting as some immobilization may still be needed for the formation of SOM from high-N residue.
6. In the CERES-based module, SOM and residue decomposition are independent of soil texture. Soil texture only affects the soil water conditions and, thus, the value of the multiplier for the SOM decomposition rate. But heavy clays are well known to protect SOM particles such as microbes, which hide in small pores (Beare et al., 1994; Hassink et al., 1993; Ladd et al., 1993; Van Veen and Kuikman, 1990).

Despite these limitations, Bowen et al (1993) reported a realistic simulation of N release from legume residues with the CERES-based module. However, Gabrielle and Kengni (1996, p. 142) reported that "the original CERES mineralization submodel did not correctly simulate N supply from potentially degradable SOM." Hasegawa et al. (1999, p. 255) stated that "the N transformation submodel substantially underestimated flushes in inorganic soil N immediately after rainfall or irrigation." This clearly demonstrates the need to improve the SOM–residue module of DSSAT crop simulation models, including CROPGRO and CERES.

In a special issue of *Geoderma* (Vol. 81, 1997), nine SOM models were evaluated with 12 long-term data sets, including inorganic fertilizer, organic manures, and different rotations. Measured and simulated data were compared using an array of statistical analysis tools. The CENTURY model (Parton et al., 1988, 1994) was among the models that performed best. It consistently produced low errors for all data sets but one, showed the lowest overall bias, and was able to simulate both low- and high-N treatments (Kelly et al., 1997; Smith et al., 1997).

Many other authors evaluated the CENTURY model under a range of conditions. Powlson et al. (1996) compared several models, including CENTURY, with long-term data sets. Paustian et al. (1992) compared different types of organic inputs with the CENTURY model. Parton and Rasmussen (1994) evaluated the performance of CENTURY with a 55-yr winter wheat experiment. Metherell et al. (1995) compared the impact of different types of tillage, and Probert et al. (1995) evaluated CENTURY for a long-term fallow experiment. Because of this extensive and detailed evaluation of the CENTURY model, as well as its wide adoption by many others, this model was a good candidate for being linked to the DSSAT crop simulation models. The main objective of this paper is to present the linkage between the DSSAT crop simulation models and the SOM–residue section from the CENTURY model and especially its implementation in the grain legume model CROPGRO.

The majority of the CENTURY evaluations dealt with SOM-C but less with N. Our prime focus for the use of the DSSAT crop simulation models with an improved SOM–module is for low-input systems that are driven by nutrients from SOM and residue decomposition. Nitrogen is thus of crucial importance. The second objective of this paper is to present an initial evaluation of the linked system models for SOM-C with a long-term (40 yr) experiment and for N with a 1-yr experiment with seven different leguminous residues as well as a control.

MATERIALS AND METHODS

Model Description

The pool and flow structure of the two modules is illustrated in Fig. 1. The CERES-based module (Fig. 1a) receives newly added residues in the pool of FOM where it is divided into 20% carbohydrates, 70% cellulose, and 10% lignin. These percentages are default values but can be changed by the user. The three pools decompose daily at a rate obtained by multiplying the potential decomposition rate by rate factors for the temperature, soil water, and C/N ratio. If the N concen-

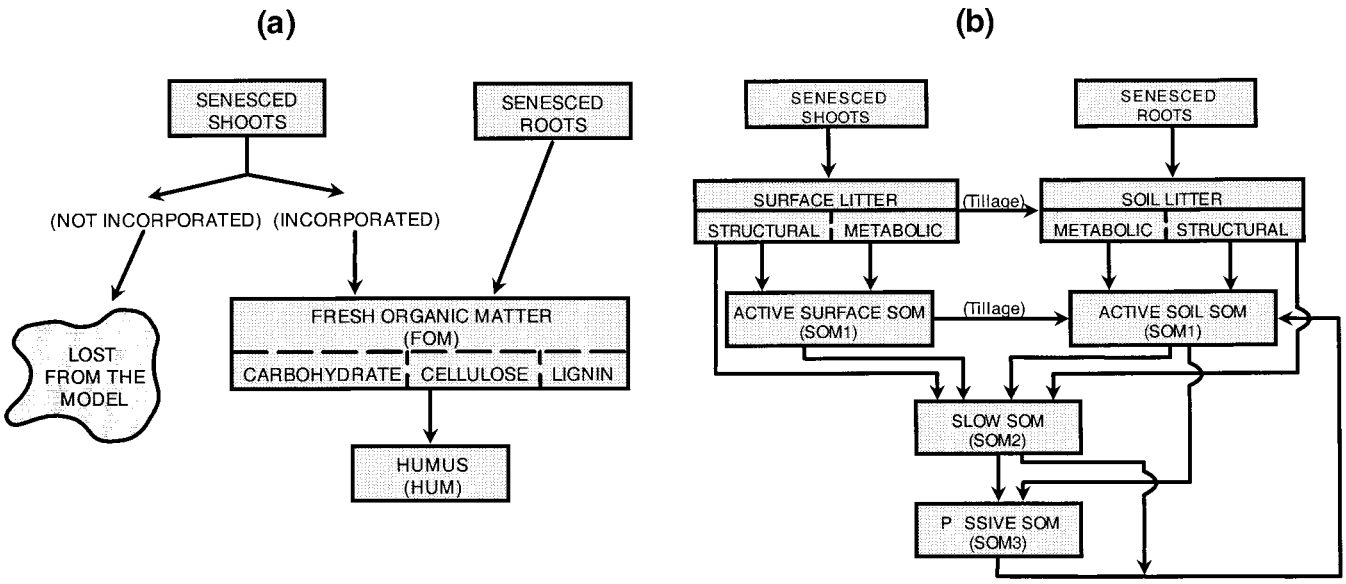


Fig. 1. The pool and flow structure of the (a) CERES-based and (b) CENTURY-based soil organic matter (SOM)–residue modules of DSSAT. Each of the decomposition flows (vertical arrows) between pools (boxes) is accompanied by either N mineralization or immobilization.

tration in the litter is <2%, N immobilization may occur. Similarly, the potential decomposition rate of the humus pool is multiplied by soil temperature and water factors. For atypical soils, an additional factor is used to either decrease or increase decomposition. During humus decomposition, no N immobilization occurs.

Immobilization of N by decomposing litter in the CERES-based module will occur if its C/N ratio is wider than 20. In the original PAPRAN model of Seligman and Van Keulen (1981), N immobilization by litter depended on the litter's *initial* C/N ratio, but in the CERES-based module of DSSAT, N immobilization depends on the litter's *actual* C/N ratio. This distinction is important because, in the latter, the critical ratio for immobilization is fixed at a value of 20 while in the original model, it depended on the type of litter entering the system. An error was found in the original DSSAT source code that stated that the litter would decompose even if there was not enough mineral N available to accommodate the immobilization. In the CERES-based module, this has been changed.

In the CENTURY-based module (Fig. 1b), organic residues are handled as either surface litter or soil litter. Both types of litter are divided into easily decomposable, metabolic materials (e.g., sugars and proteins) and recalcitrant, structural materials (e.g., lignin and other fibers). This division depends on the lignin/N ratio: metabolic fraction = 0.85 − 0.013(lignin/N). The structural material is given a fixed C/N ratio of 200, whereas the C/N ratio of the metabolic material varies with the N concentration of the litter.

After decomposition, metabolic material in the CENTURY-based module becomes part of the pool of active (microbial) SOM (SOM1), which exists in both the surface layer and in the soil (surface SOM1 and soil SOM1, respectively). Structural material is composed of fibrous material and of easily decomposable material that is encapsulated by structural fibers. The easily decomposable fraction becomes part of the surface or soil SOM1 while the fibrous fraction becomes part of the intermediate SOM pool (SOM2), which only exists in the soil. The surface or soil SOM1 decomposes into SOM2, and the soil SOM1 also decomposes into passive SOM (SOM3). The SOM2 decomposes into SOM3 or can be reactivated as soil SOM1; the SOM3 can also be reactivated as soil SOM1. All these processes are accompanied by a loss of CO_2 and by either mineralization or immobilization of N, depending on the C/N ratio of the decomposing material and the mineral N available for immobilization. Decomposition rates are, as in the CERES-based module, calculated as the potential decomposition rate multiplied by temperature and water factors and also involve factors for soil texture and cultivation, of which the latter is used to accommodate an increased decomposition rate after soil disturbance.

In the CENTURY-based module, soil texture directly affects the SOM and residue flows so that, with increasing clay content, there is:

1. A reduced decomposition rate of soil SOM1;
2. An increased partitioning of the flow out of soil SOM1 towards SOM3, and thus a reduced flow from soil SOM1 to SOM2;
3. A reduced fraction of SOM-C lost as CO_2 at decomposition; and
4. A different value for the multiplier on the optimal decomposition rate to accommodate for the effect of soil water conditions on decomposition.

Adapting the CENTURY Soil Organic Matter–Residue Module to DSSAT

For the SOM–residue section of DSSAT, two modules were developed[1]: the original CERES-based module (Gijsman et al., 2002) and a new CENTURY-based module (Gijsman and Porter, 2002). Both modules are part of the new modular version of CROPGRO (Jones et al., 2001). Though DSSAT consists of several crop models, in this paper, we only discuss the adaptation of the grain legume model CROPGRO.

We used the CENTURY version 4.0 (the so-called agroecosystem version), supplemented by updated code from versions that have not been released. Only the SOM-C and N subroutines from CENTURY were activated because the current version of CROPGRO does not handle P interactions and simulations (though P is soon to follow). CENTURY's SOM–

[1] In the new modular version of DSSAT, some variables have been renamed in a more systematic way. For example, the humus C and N content, which used to be HUM and NHUM, are now named HUMC and HUMN.

residue module was separated from the rest of the model, and its subroutines were restructured. The CENTURY model recognizes a surface litter layer and one soil layer (the top 20 cm of the soil). In DSSAT, as many as 20 soil layers can be defined; this characteristic was also applied to the CENTURY-based module.

In contrast to the monthly time intervals of the CENTURY model, the DSSAT crop simulation models use daily intervals. Decomposition rate parameters of the CENTURY-based module were thus recalculated accordingly. The CENTURY units are in g m^{-2}, whereas the DSSAT soil modules use kg ha^{-1}; these were also brought to the DSSAT standard.

The ability to simulate NH_4 was added to the CENTURY-based module so that immobilization is first *taken* from the NH_4 pool and then from the NO_3 pool while mineralized N is added to the NH_4 pool.

A new, recently developed set of equations was used for calculating the effect of soil water (Parton et al., 1996) and temperature (W.J. Parton, unpublished data, 1999) conditions on SOM and litter decomposition with the CENTURY-based module. The section on anaerobic conditions was removed.

Before conducting the decomposition of a SOM or residue pool, in the original CENTURY model, it is determined whether enough mineral N is available for the decomposition to proceed. After calculating the decomposition of this SOM–residue pool, N from mineralization is added to the soil mineral N pool, and N removed by immobilization is subtracted. Then the next SOM–residue pool is handled. This, however, means that there is a hierarchy concerning which pool has priority in accessing the mineral N. If the first pool has immobilized all N, then the pools that have lower priority cannot immobilize any more N, and thus cannot decompose. This has been changed by first calculating the net immobilization of all of the pools together, and if this is greater than the mineral N available, all immobilizing flows are reduced; the mineralizing flows can continue.

In the CENTURY-based module, soil disturbance results in a temporary enhancement of the SOM and residue decomposition rate. A new routine was added so that surface litter or surface SOM1 present on top of the soil—and already partly decomposed—will be (partly) incorporated by tillage, and thus add to the litter or soil SOM1 pool of one or more soil layers.

Parameterization

The CENTURY model includes many parameters that are stored in a *fixed-parameter* file, and users are not expected to change these parameters. All fixed parameters were recalculated to the DSSAT units, as discussed previously. One parameter value was changed to remove an inconsistency. The only new parameter needed in DSSAT for the CENTURY-based module is the long-term management history of the soil (cultivation vs. grassland, savanna, or steppe), which is used for SOM initialization. The default initial ratio SOM1/SOM2/SOM3 is 0.02:0.54:0.44 for a cultivated soil and 0.02:0.064:0.34 for a grassland soil, but it can also be set to specific values.

If no data on SOM-N are supplied, each SOM pool is initialized with its default C/N ratio: for SOM1, C/N = 10; for SOM2, C/N = 17, and for SOM3, C/N = 7. User-supplied total SOM-N data overrule these values for SOM1 and SOM2. The SOM3, which is an almost inert pool hardly affected by recent soil management or residue inputs, is left at its default C/N ratio of 7. In the CENTURY-based module, the C/N ratio of material flowing from one SOM or residue pool to another SOM pool is flexible (between a minimum and maximum value), depending on the C/N ratio of the incoming material, the C/N ratio of material allowed to enter the receiving SOM pool, and the availability of mineral N in the soil.

If the CENTURY-based module were applied with the immobilization parameters set to the values used in the original CENTURY model, a large part of the N released from the residues would be immobilized by microbes. These settings were recalibrated by modifying the minimum C/N ratio of the microbes and the critical value for the soil mineral N content at which this ratio is reached. Microbes were given a C/N ratio between 6 (conditions of ample mineral N availability) and 14 (where no mineral N is available). These values are more in agreement with the recently modified parameter settings in the CENTURY model. For this calibration, use was made of a legume experiment similar to the one used in the validation (see Experimental Data Used for Model Comparison section) but carried out in the rainy season (Bowen et al., 1993). It was checked by using one of the nine treatments [the *Cajanus cajan* (L.) Millsp. treatment] from the validation experiment. The simulated results in this paper are based on these modified parameter settings.

Senesced Material Added Daily to the Soil

The DSSAT model CROPGRO calculates daily senescence. The senesced material, however, is not added to the soil until the end of the cropping season when harvest residues are also added. When preparing the soil for a following crop, such incorporation may happen. For a simulation run that refers only to a single cropping season, senesced material is thus never added to the soil. Therefore, a surface litter layer was introduced in the CENTURY-based module. In addition, senesced material is now added daily to the soil as shoot residues decomposing in the surface litter layer or as root residues in their respective soil layers.

Experimental Data Used for Model Comparison

To validate the CENTURY-based module, two data sets were used: one for a long-term simulation of soil C and the other for a 1-yr simulation of N release from leguminous residues. The first data set was collected at the Highfield bare fallow plot located at the Institute of Arable Crops Research (IACR), Rothamsted, UK. This plot, which is located on a silty clay loam with 23.4% clay, 23% silt (2–20 μm), and 49% sand (rest is gravel), had been under grassland for several hundreds of years until it was plowed in 1959 (Jenkinson et al., 1987). Since then it has been left bare. It was plowed to 23-cm depth four times a year, with an estimated 500 kg ha^{-1} weeds per year that were plowed under. Atmospheric dry and wet N deposition plus N input by symbiosis of free-living, heterotrophic N-fixing bacteria was in the range of 40 to 50 kg ha^{-1}. Soil samples were taken in 1959, 1963, 1971, 1978, and 1987 to a depth of 23 cm and analyzed for C. The initial SOM-C content of the 0- to 23-cm layer was 76.0 t ha^{-1} (2.5% at a bulk density of 1.33 g cm^{-3}), with a C/N ratio of 12. In DSSAT, the SOM concentration for the top three soil layers (i.e., 0–5, 5–15, and 15–23 cm) was set to the same value but adapted for bulk density differences. Because the previous land use had been long-term grassland, the ratio SOM1/SOM2/SOM3 at initialization was set to 0.02:0.64:0.34 for the CENTURY-based module. Soil water retention characteristics were estimated from the texture and SOM concentration according to Rawls et al. (1982).

To evaluate the soil N component of the model, detailed data from a set of experiments conducted in the Brazilian *cerrados* (savannas) were used (Bowen et al., 1993). In this experiment, residues from leguminous green manures were

added to the soil, and the pattern of mineral N in the soil at various depths was measured several times during the year. The treatments involved fallow, with or without any of seven leguminous residue types, during the (irrigated) dry season and subsequent wet season. An earlier experiment conducted at the same site (Bowen et al., 1993) was used for calibration (see Parameterization section). The soil type was a Dark Red Latosol in the Brazilian classification system, which classifies as clayey, oxidic, isothermic, Anionic Acrustox in the U.S. classification system. Soil texture was 59% clay, 22% silt, and 19% sand, and the plot had previously been under cropping, with 1.81% organic C in the topsoil. The SOM C/N ratio was not known; thus, the models' default values were used. For the CENTURY-based module, the ratio SOM1/SOM2/SOM3 at initialization was set to 0.02:0.54:0.44.

In these variable-charge soils, NO_3 leaching depends on the anion exchange (adsorption) capacity (AEC) of the soil. This parameter was not measured, but the AEC value of each soil layer was adjusted for a fallow treatment that had not received leguminous residues to fit the model's results to the measured data, as was done by Bowen et al. (1993). The same AEC values were also applied to other treatments that received residues. This fitting of the simulated data of the fallow plot to the measured values means that the estimated AEC value may implicitly correct for other inaccuracies in the model, which may have been unknown (e.g., inaccuracy in the soil temperature module, water module, and leaching module). The two modules, therefore, resulted in different AEC estimates, with rather small values for the CENTURY-based module for each successive 15-cm soil layer and bigger values for the CERES-based module, as estimated by Bowen et al. (1993):

AEC estimates for CENTURY-based module =

0.0, 0.0, 0.0, 0.1, 0.1, 0.1, 0.1, and 0.15 $cm^3\ g^{-1}$

AEC estimates for CERES-based module =

0.0, 0.0, 0.0, 0.4, 0.8, 1.0, 1.2, and 1.60 $cm^3\ g^{-1}$

Van Raij and Peech (1972) stated that the AEC value of the topsoil is generally very small (sometimes zero) and increases with depth. Estimates of AEC were thus unlikely to lead to important errors of the simulated mineral N data in the top layers of the soil, which are mainly affected by residue.

The CERES-based module requires input data to define the fraction of carbohydrates, cellulose, and lignin in residues while the CENTURY-based module requires input data on lignin. For the Brazilian experiment, the residue amounts and their N concentrations were known, but they had to be estimated for the Rothamsted experiment (which had only input of weeds). For both experiments, the carbohydrates, cellulose, and lignin fractions were set to the default values of 0.20, 0.70, and 0.10, respectively, because they were not measured during the experiment. Similar values were also used by Bowen et al. (1993) for the Brazilian experiment.

Statistical Analysis

The measured data did not include replicates. Following Whitmore's (1991) recommendation, the product moment correlation coefficient and mean difference were used as statistical analysis tools. The correlation coefficient (r) is a measure of the degree of association between simulated and measured data. The mean difference [mean difference = Σ(measured − simulated)/N] reveals a possible trend of the model to overestimate or underestimate the data (N is the number of data pairs). We also used the root mean square error {RMSE = $[\Sigma(\text{simulated} - \text{measured})^2/N]^{0.5}$}, which quantifies the dispersion between simulated and measured data (Gabrielle and Kengni, 1996; Quemada and Cabrera, 1995). Note that the RMSE is sometimes also expressed in relative instead of absolute terms, i.e., multiplying it with 100/mean of measured (e.g., Loague and Green, 1991).

THE SOIL ORGANIC MATTER–RESIDUE MODULE STRUCTURES

Soil Organic Matter Nitrogen Initialization and Carbon/Nitrogen Ratio as a Decomposition Rate Determinant

If the SOM–residue modules are initialized with the modules' default SOM-N concentrations, it may result in a very different organic N content at initialization for the two modules. In general, this is lower for the CENTURY-based module than for the CERES-based module. If the user provides the initial data, the total SOM-N at the start of the simulation is identical.

Twenty percent of the residue N that is decomposed in the CERES-based module is added to the humus pool, taking with it an amount of C that gives the newly formed humus a C/N ratio of 10. The fraction of litter C added to humus or lost to microbial respiration is thus a function of the litter's N concentration. In contrast, in the CENTURY-based module, decomposing soil litter will only become part of soil SOM1 and SOM2 if its C/N ratio after decomposition is below 14 and 20, respectively, or if it can immobilize enough N to bring (part of) the litter to such a C/N ratio; for the surface litter, these values are variable, depending on the litter's N concentration.

In the CERES-based module, the SOM decomposition rate depends only on environmental conditions and not on the humus' C/N ratio. The amount of humus that decomposes is calculated as a fraction of the humus N pool (HUMN). Thus, 100 units of SOM with 2% N mineralizes the same amount of N as 200 units with 1% N, but it gives has half as much C decomposition. In contrast, in the CENTURY-based module, the *receiving* SOM pool determines what kind of material can enter, and thus which material may decompose. This means, for example, that for the flow from SOM2 toward SOM3, mineral N availability is the rate determinant because the material goes from a pool with a relatively wide C/N ratio to a pool with a narrower C/N ratio; N immobilization may thus be needed. The flow from SOM3 toward SOM1, in contrast, goes from a pool with a relatively narrow C/N toward a pool with a wider C/N, and thus is never N limited.

Temperature and Water as Decomposition Rate Determinants

The two modules use different mathematical relationships to describe the effects of soil water conditions and soil temperature on the decomposition. For the soil temperature factor (Fig. 2a), the CERES-based module uses a linear relationship, with its optimum at 35°C or higher, while the CENTURY-based module uses a curvilinear relationship, reaching its optimum at 30°C. The soil water factor with the CERES-based module (Fig.

2b) decreases from optimal at field capacity to zero at wilting point for all layers, except for the topsoil (response not shown). The CERES-based module allows the topsoil layer to dry out to a water content below wilting point because of water evaporation from the soil surface. The equations used for this layer, however, are such that the soil water factor differs from those of the other layers, even if the soil water content is the same. The CENTURY-based module distinguishes two soil classes for the soil water factor: fine-medium and coarse (Fig. 2c). The soil water factor is at its optimum at a water-filled porosity of 60% (fine- or medium-textured soil) or 55% (coarse soil), equaling a volumetric water content of about 0.32 and 0.21 $cm^3\ cm^{-3}$, respectively (depending on the bulk density). Above field capacity, both modules let the soil water factor decline because of the excess of available water.

Both the CERES- and CENTURY-based modules assume that the SOM decomposition rate decreases under very wet soil conditions. But, as Greenland et al. (1992) pointed out, SOM decomposition in the tropics may be sufficiently rapid so as not to be limited by poor soil aeration. The low SOM content of many wetland paddy rice soils is testimony to this suggestion (Kyuma, 1985). For such soils, both modules have to be used with caution.

Litter Decomposition

The CERES-based module calculates the daily decomposition of the three litter (FOM) pools—i.e., carbohydrates, cellulose, and lignin. It then reduces the litter C and N by the relative contribution of each pool to the total amount of litter × its fraction that decomposes. For C, this is correct, but for N, it ignores the fact that the N concentration of the three litter pools will not be the same. In the CERES-based module, fresh litter, which is relatively rich in carbohydrates, has the same N concentration as old litter of the same type, which consists mainly of lignin (unless immobilization has occurred, adding N to the litter).

From a 6-mo incubation study with various crop residues, Quemada and Cabrera (1995) concluded that allowing the user to vary the relative size of the residue (FOM) pools greatly improves the simulation compared with having fixed fractions of the three pools. The present version (3.5) of the DSSAT crop models has such flexible settings as a standard option.

Earlier calibrations and validations with the CERES-based SOM model resulted in different estimates of the decomposition rate constant of the three litter pools under nonlimiting conditions. For carbohydrates, it ranged from 0.05 to 0.8 d^{-1}; for cellulose, 0.0034 to 0.05 d^{-1}; and for lignin, 0.00095 to 0.0095 d^{-1} (Bowen et al., 1993; Quemada and Cabrera, 1995; Vigil et al., 1991). The currently used values are 0.2, 0.05, and 0.0095 d^{-1}, respectively, while for the humus pool, the rate constant is 0.000083 d^{-1}.

The pattern of residue decomposition, simulated by both modules, was similar in dry matter terms until about 80 d, after which the CENTURY-based module

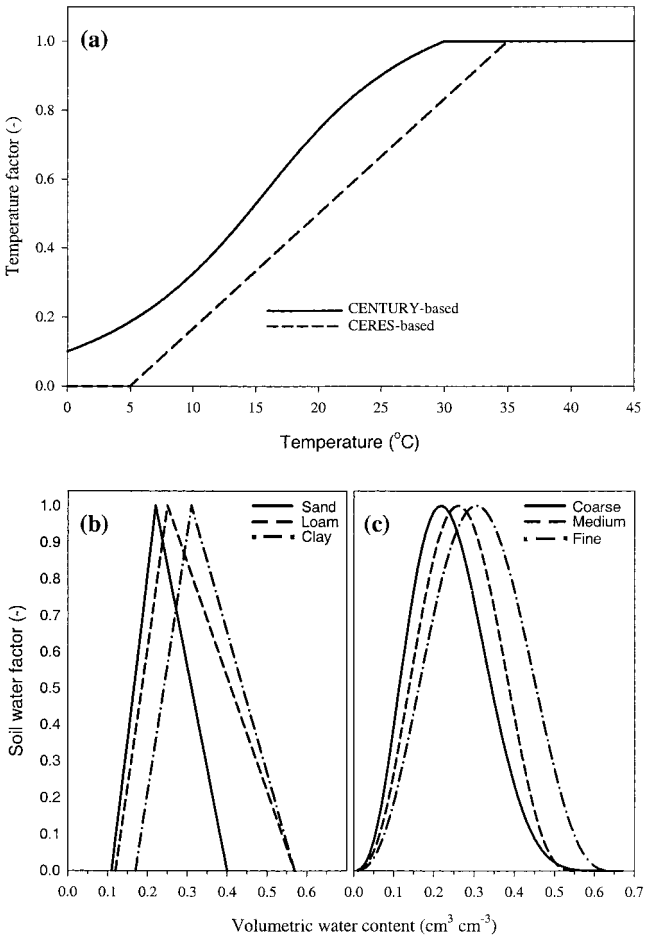


Fig. 2. (a) Soil temperature and (b and c) soil water factors in the (b) CERES-based and (c) CENTURY-based soil organic matter (SOM)–residue modules. For both modules, the curves of the soil water factor will be different for each soil as they vary with the soil water parameters field capacity and wilting point (CERES-based module) and soil texture plus bulk density (CENTURY-based module). The curves presented here are just examples. For the CENTURY-based module, "coarse," "medium," and "fine" have been given bulk densities of 1.6, 1.5, and 1.3 g cm^{-3}, respectively.

resulted in a slower decomposition of the remaining residues than did the CERES-based module. An example for *C. cajan*, one of the residue types from the Brazilian experiment, is shown in Fig. 3. Although the CENTURY-based module was slower in residue decomposition (Fig. 3a), it was faster in releasing N from the residues (Fig. 3b, top two lines). The CENTURY-based module calculated the metabolic fraction of the residues at 79%, based on the lignin/N ratio, but this fraction took up 98.3% of the residue's N, as the structural pool is given a fixed C/N ratio of 200. However, in the CERES-based module, no distinction is made between the N concentrations of the various residue fractions. The slowly decomposing lignin fraction, which made up 10% of the residue, thus also took up 10% of the residue N.

Although the rate of N release from the residues was faster with the CENTURY-based module than with the CERES-based module, this does not mean that this N is

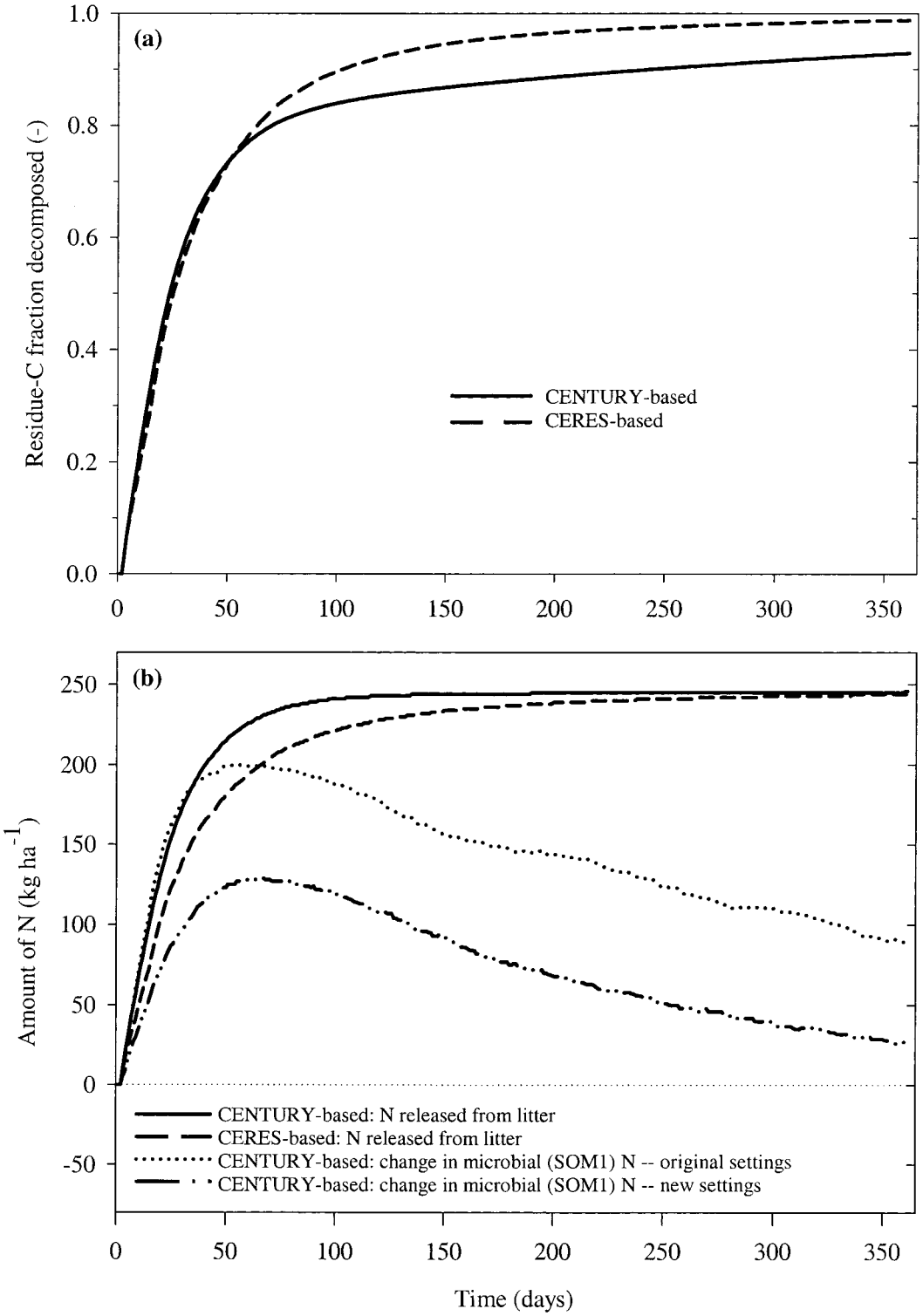


Fig. 3. (a) Decomposition pattern of *Cajanus cajan* residues as simulated with the CERES-based (dashed line) and CENTURY-based (solid line) soil organic matter (SOM)–residue modules. (b) Nitrogen mineralized from *C. cajan* residues as simulated with the CERES-based (dashed line) and CENTURY-based (solid line) SOM–residue modules, and N immobilized by microbes with the original parameter settings (dotted line) and with the modified parameter settings (dashed-dotted line), as simulated by the CENTURY-based SOM–residue module. Residues with a N concentration of 2.35% were applied at 10 495 kg ha^{-1}; they were incorporated at a 20-cm depth.

available in the soil as mineral N. With the CENTURY-based module, part of the N released from decomposing residues was immobilized by the microbial biomass, thus ending up in SOM1. This N was gradually released again after 50 to 100 d (Fig. 3b, dotted line). As described in the Parameterization section, we recalibrated the microbial N immobilization to limit its impact (dash-dot line).

The CERES-based module does not recognize a mi-

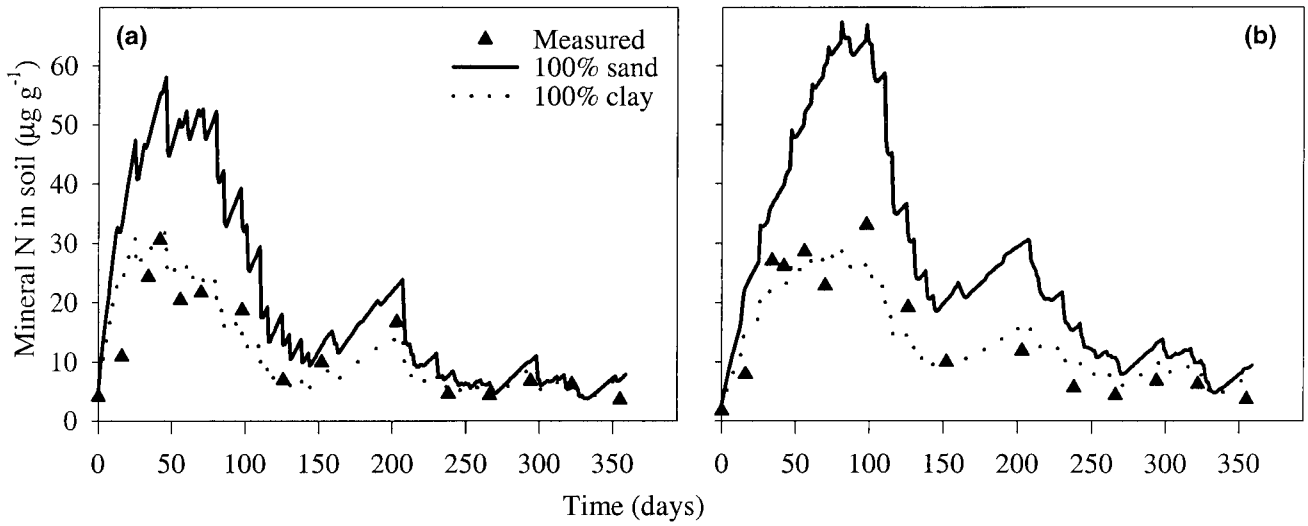


Fig. 4. Mineral N concentration of the (a) 0- to 15- and (b) 15- to 30-cm soil layers under fallow with *Cajanus cajan* residues, as measured (▲) and as simulated with the CENTURY-based soil organic matter (SOM)–residue module with the soil texture of all layers set to 100% sand (solid line) or 100% clay (dotted line). The residues applied were 10 495 kg ha^{-1} of *C. cajan* tops with a N concentration of 2.35%.

crobial biomass pool as microbes are assumed to be part of the FOM (Seligman and Van Keulen, 1981). Immobilization is then handled by adding N to the residue N pool (FON) during its decomposition, instead of adding it to the HUMN that is derived from FON. Although this is theoretically not correct, it is one way of overcoming the limitation that in the CERES-based module, the fractions of FOM and FON that decompose are the same. The residue's C/N ratio would thus always remain constant over time. Residue with a high C/N ratio would thus always remain recalcitrant to decomposition. Adding N from immobilization to FON, however, gives a gradually narrowing C/N ratio.

In the CERES-based module, a fixed 20% fraction of N from decomposing residues is transferred to the humus pool while the remaining 80% is released as mineral N. This amount may be reduced by the N needed for immobilization to bring the decomposing litter to a 2% N concentration. For SOM decomposition, there is no N immobilization in the CERES-based module. In the CENTURY-based module, net N mineralization or immobilization during SOM–residue decomposition depends on the C/N ratios of the decomposing material and of the material allowed to enter the receiving pool.

Soil Texture

In the CENTURY-based module, the flows between the various SOM–residue pools depend on the soil texture, whereas in the CERES-based module, texture does not play a role. Figure 4 shows the impact of the two extremes in soil texture—100% clay and 100% sand—on simulated pattern of soil mineral N with an example from the Brazilian experiment. For this, only the soil texture effect on the SOM–residue decomposition was taken into account. Its effect on the soil water conditions was not dealt with. This figure clearly shows the importance of accurate soil texture data for the CENTURY-based module.

EVALUATION

Long-Term Soil Organic Matter Carbon

For the long-term bare fallow experiment, the CENTURY-based module resulted in a close fit between simulated and measured SOM-C (Fig. 5), with a mean difference of 1.48 t ha^{-1} C, a RMSE of 3.36 t ha^{-1} C, and a product moment correlation coefficient (r) of 0.983. Despite the very good congruence, the shape of the curve compared with the measured values suggests that some of the rate parameters (particularly of SOM2) may have to be increased to obtain a steeper decline and the fraction of SOM3 be increased to obtain the correct platform level. There are, however, a number of other parameters that should be considered, especially for long-term experiments.

The CERES-based module did not simulate SOM-C very well and gave a poor fit to the measured data. The CERES-based module has only one SOM pool with a maximum decomposition rate of 0.000083 d^{-1}, whereas the three SOM pools in the CENTURY-based module have rates of 0.02, 0.00054, and 0.000012 d^{-1}, respectively, for SOM1 (soil pool, not the surface pool), SOM2, and SOM3. The SOM2 pool, which at initialization occupied 64% of the total SOM content, thus had a maximum decomposition rate that was almost seven times higher than that of the SOM in the CERES-based module. Even the lower maximum decomposition rate for SOM3 did not even this out as this pool occupied only 34% of the total SOM content at initialization.

One element of uncertainty is that the soil texture data that were used as inputs for the simulation were expressed in European (or ISSS) textural units, in which silt is the 2- to 20-μm class while DSSAT and CENTURY use the American unit system with silt equal to 2 to 50 μm. This not only affects the SOM decomposition process directly, as the fractions of sand and clay influence the SOM decomposition rate and the flows between pools in the CENTURY-based module, it also

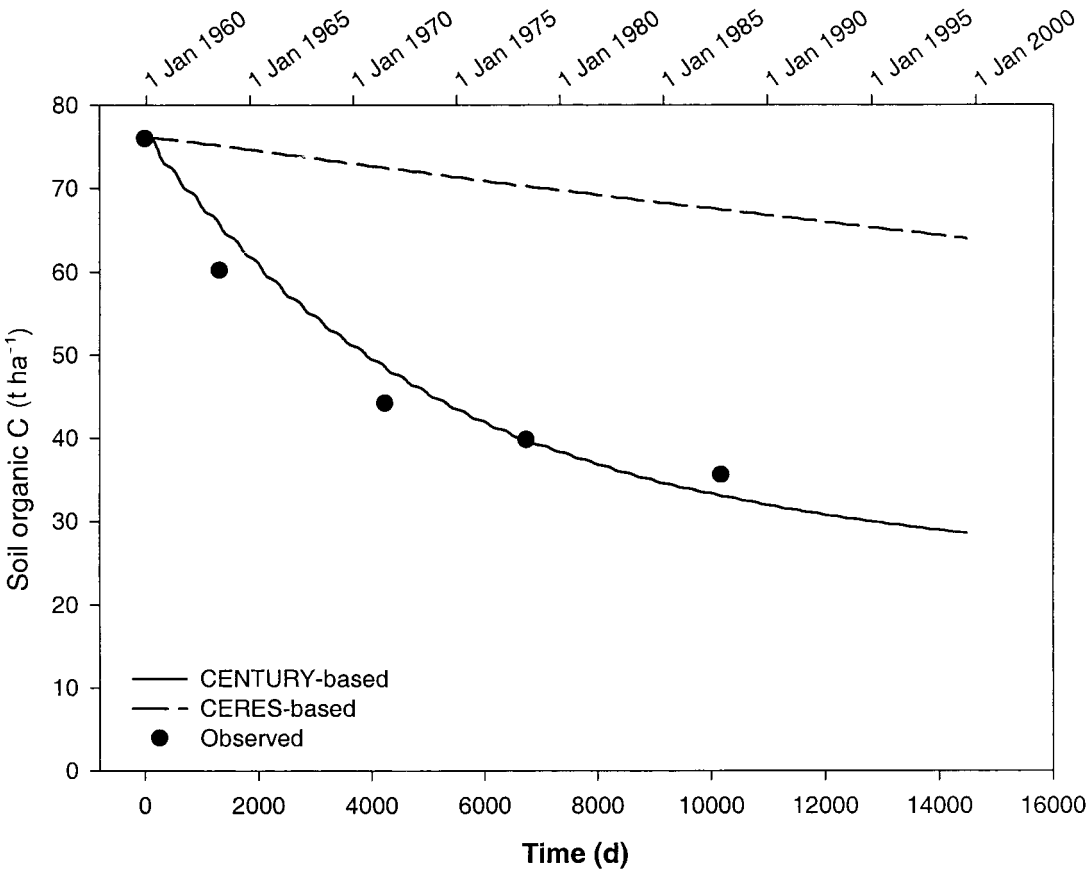


Fig. 5. Soil organic C content of the 0- to 23-cm layer of the soil under bare fallow in the Rothamsted Highfield bare fallow experiment, as simulated with the CERES-based (dashed line) and CENTURY-based (solid line) soil organic matter (SOM)–residue modules.

affects it indirectly as texture was used for estimating the soil water retention characteristics. A sensitivity analysis was conducted in which the model was run for both the measured silt (2–20 μm) fraction of 23% and an estimated silt (2–50 μm) fraction of 30% and, thus, a reduced sand fraction. This analysis showed that the percentage of silt hardly affected the results: After a simulation of 40 yr, the SOM-C level differed <95 kg ha^{-1}.

Nitrogen Profile in the Soil with Various Types of Residues

The leguminous residues in the Brazilian experiment were incorporated to a 20-cm depth while the mineral N content of the soil was measured in 15-cm increments. The top layers of the soil (0–15 and 15–30 cm) should thus be most affected by the residue application. The mineral N concentrations for these layers, as estimated by the application of the two modules to bare fallow plots and fallow plots with seven different leguminous residue types, are shown in Fig. 6. The high mineral N values between Day 30 and 70 reflect the flush of N from the decomposing residue. The second (smaller) peak around Day 200 is due to the start of the rainy season when decomposition of the remaining residue sped up again.

Both modules gave a reasonable-to-good fit between simulated data and measured data for some treatments and soil layers but a poor fit for others. The CENTURY-based module resulted in better estimations of soil mineral N for the topsoil layer of bare fallow, *C. cajan*, *Calopogonium mucunoides* Desv., *Canavalia brasiliensis* Mart. ex Benth., *Pueraria phaseoloides* (Roxb.) Benth., and *Mucuna aterrima* (Fig. 6a, 6c, 6e, 6g, 6m, and 6o) and for the 15- to 30-cm layer of bare fallow, *C. cajan*, and *M. aterrima* (Fig. 6b, 6d, and 6p). The CERES-based module performed better for both layers of *C. ensiformis* (L.) DC. and *Crotalaria striata* Aiton (Fig. 6i, 6j, 6k, and 6l) and for the deeper soil layer of *C. mucunoides* and *C. brasiliensis* (Fig. 6f and 6h).

In general, the CERES-based module provided a better simulation for mineral N for the 15- to 30-cm layer. In the topsoil layer, estimated mineral N obtained with the CENTURY-based module gave a better agreement between simulated and measured data than that from the CERES-based module as demonstrated by the statistical parameters shown in Table 1. It should be noted that the statistics point to a different conclusion than the graphical representation for some of the treatments where the CERES-based module seemed to perform better (Fig. 6i and 6k). This suggests that, though the CENTURY-based module missed the high peaks around Day 40, in general, it provided a more accurate simulation of the mineral N for the 0- to 15-cm layer. The CERES-based module simulated the main peak in mineral N but did not simulate the decrease in mineral N between Day 40 and 80 shown by the measured data,

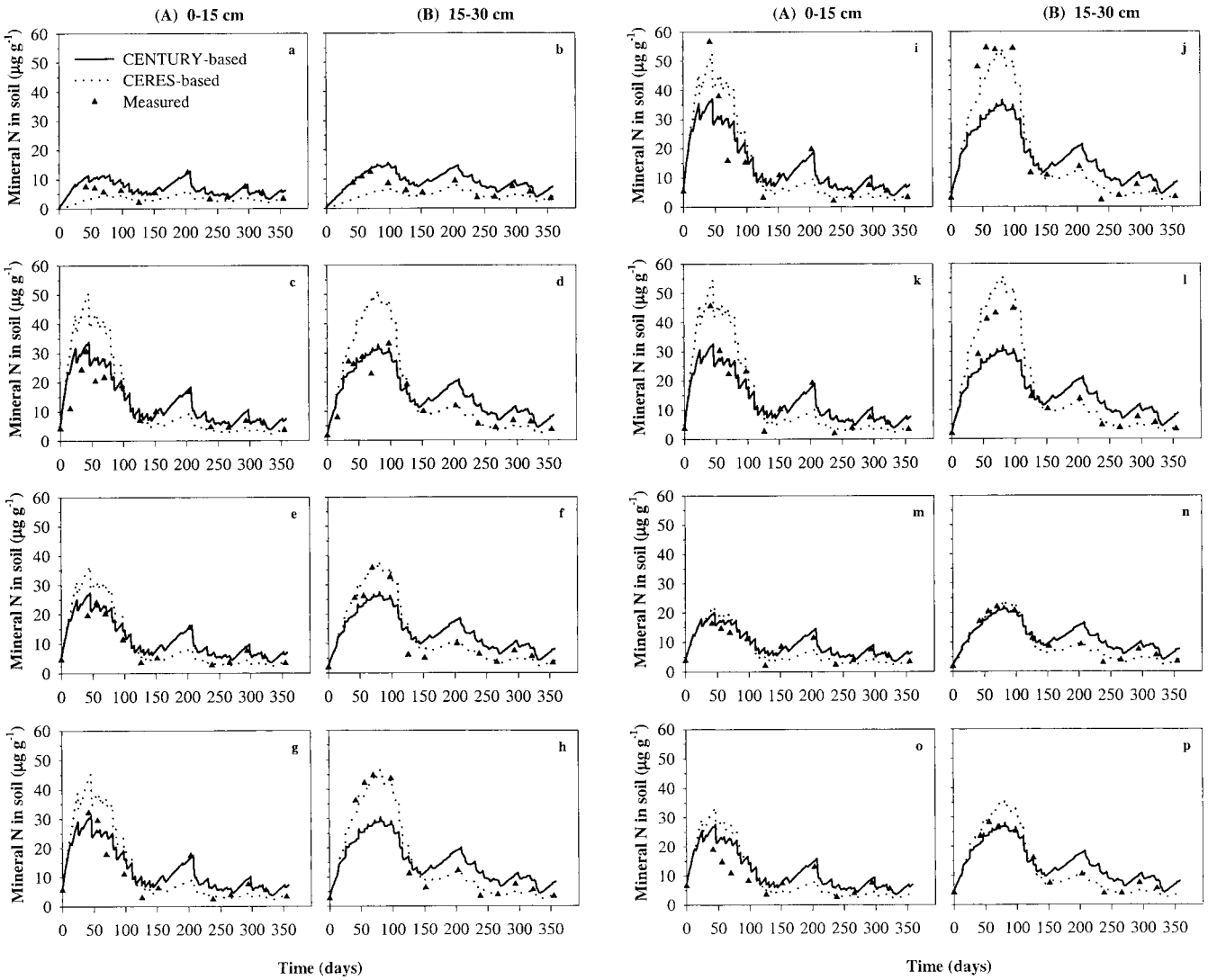


Fig. 6. Mineral N concentration of the (A) 0- to 15- and (B) 15- to 30-cm soil layers under fallow, without or with various residues, as measured (▲) and as simulated with the CERES-based (dotted line) and CENTURY-based (solid line) soil organic matter (SOM)–residue modules. The treatments applied were (a and b) bare fallow, (c and d) *Cajanus cajan* (10 495 kg ha^{-1} with a N conc. of 2.35%), (e and f) *Calopogonium mucunoides* (7586 kg ha^{-1} with a N conc. of 2.17%), (g and h) *Canavalia brasiliensis* (8214 kg ha^{-1} with a N conc. of 2.41%), (i and j) *C. ensiformis* (9514 kg ha^{-1} with a N conc. of 2.68%), (k and l) *Crotalaria striata* (12 444 kg ha^{-1} with a N conc. of 2.16%), (m and n) *Pueraria phaseoloides* (4758 kg ha^{-1} with a N conc. of 1.91%), and (o and p) *Mucuna aterrima* (5907 kg ha^{-1} with a N conc. of 2.37%).

especially for *C. striata* (Fig. 6k). The mean difference across all treatments for Day 56 and 70 was, respectively, 8.39 and 14.85 μg g^{-1} for the CERES-based module and 0.53 and 7.04 for the CENTURY-based module. The CERES-based module did not simulate the second mineral N peak around Day 200: The mean difference was −7.44 μg g^{-1} for the CERES-based module vs. 0.18 μg g^{-1} for the CENTURY-based module.

Table 1. Mean difference (Md), product moment correlation coefficient (*r*), and root mean square error (RMSE) of the mineral N concentration in the 0- to 15- and 15- to 30-cm soil layers, as simulated with the CENTURY-based and CERES-based soil organic matter (SOM)–residue modules. For an explanation of the treatments, see Fig. 6 caption.

	Letter in Fig. 6	CENTURY based 0–15 cm			CERES based 0–15 cm			CENTURY based 15–30 cm			CERES based 15–30 cm		
		Md	*r*	RMSE	Md	*r*	RMSE	Md	*r*	RMSE	Md	*r*	RMSE
		μg g^{-1}		μg g^{-1}	μg g^{-1}		μg g^{-1}	μg g^{-1}		μg g^{-1}	μg g^{-1}		μg g^{-1}
Bare fallow	a, b	2.04	0.78	2.69	−2.23	0.82	2.96	2.95	0.77	3.52	−2.37	0.38	3.53
Cajanus cajan	c, d	3.02	0.94	4.67	5.18	0.91	11.28	2.47	0.93	4.65	4.82	0.94	9.02
Calopogonium mucunoides	e, f	2.26	0.96	3.10	2.00	0.92	6.07	1.25	0.93	5.67	1.29	0.97	3.20
Canavalia brasiliensis	g, h	2.04	0.95	3.70	2.71	0.91	7.43	−1.49	0.95	9.37	−0.07	0.99	2.71
Canavalia ensiformis	i, j	0.49	0.90	7.71	1.21	0.87	8.61	−3.25	0.95	12.54	−1.65	0.98	5.04
Crotalaria striata	k, l	0.12	0.95	5.15	2.27	0.92	8.17	−0.89	0.96	7.84	2.56	0.99	4.48
Pueraria phaseoloides	m, n	1.90	0.95	2.53	0.27	0.89	3.27	1.84	0.93	3.31	−0.16	0.98	1.55
Mucuna aterrima	o, p	3.89	0.92	5.26	3.05	0.84	7.66	1.70	0.95	3.73	1.23	0.98	3.40

For the 15- to 30-cm layer, the statistical analysis showed similar results as the graphical representation and the CERES-based module generally performed better. The higher mean difference of the mineral N calculated by the CENTURY-based module (Table 1) was largely due to the measurements at Day 42 and 56 when the mean difference across all treatments was −6.68 and −7.54 μg g^{-1}, respectively. The CERES-based module calculated a mean difference of −0.91 and 1.04, respectively, for those dates.

For the layers below a depth of 30 cm, which were not directly affected by the added residues, the pattern of soil mineral N was mainly a function of N leaching from the top layers. The mineral N concentration at the various depths was different between the two modules (data not shown) because leaching was determined by the settings of the anion adsorption coefficients, which for the deeper layers, were different for the two modules (see Materials and Methods section). No measured data were available for anion adsorption strength, so a proper evaluation cannot be made. For the top 60 cm of the soil, the anion adsorption coefficient was set to zero for both modules; thus, the mineral N for the top two layers was not affected (Fig. 6).

Simulation of soil mineral N is inevitably prone to a larger error than simulation of soil C because mineral N is influenced by many processes besides plant uptake. Leaching, denitrification, volatilization, microbial immobilization, and adsorption onto the cation or anion adsorption complex of the soil all affect the pattern of soil mineral N. A SOM–residue decomposition model depends on the soil-water-balance and soil-temperature routines through the influence of water and temperature on the decomposition rate factor. In addition, the mineral N pattern is also affected by the impact of water and temperature on previously mentioned processes. These processes are not handled equally well by the model; therefore, soil mineral N simulation can be prone to errors. Soil C simulation does not face these problems and is thus a better measure of the quality of a SOM–residue model.

Changes in Simulated Soil Organic Matter Pools

The initial and final values of the simulated SOM pools in the 0- to 15-cm soil layer of the Brazilian experiment and the 0- to 23-cm layer of the Rothamsted experiment are shown in Table 2. In the CENTURY-based module, the most dynamic SOM pool is composed of the microbes (SOM1). These microbes feed on fresh residues or easily decomposable SOM, thereby forming new microbial biomass (SOM1) and intermediate SOM2. In the Rothamsted experiment, most of the decline of SOM simulated with the CENTURY-based module was due to losses from the SOM2 pool. The SOM2 pool decreased from from 48 645 kg ha^{-1} C at the start of the experiment to 3913 kg ha^{-1} C after 40 yr. The more resistant SOM3 pool only lost 1323 kg ha^{-1} C during this period. The C/N ratio of the total SOM decreased from 11.99 to 7.59 as the SOM3 pool that remained had a much narrower ratio than the C/N ratio of the SOM2 pool that was lost. Because the CERES-based module had only one SOM pool, the C/N ratio did not change.

In the Brazilian experiment, the CENTURY-based module simulated the largest SOM changes for the SOM2 pool, which lost between 464 and 1603 kg ha^{-1} C. The highest loss, as expected, was found in the bare fallow plot because it did not receive any residues. The SOM2 C/N ratio decreased slightly for the treatments that received legume residues. Despite the ample avail-

Table 2. Carbon and N contents and C/N ratios of the sum of all of the soil organic matter (SOM) pools (total SOM), active SOM (SOM1), intermediate SOM (SOM2), and passive SOM (SOM3) pools of the 0- to 15-cm soil layer, as simulated with the CENTURY-based SOM–residue module. Also indicated is the humus pool (HUM), as simulated with the CERES-based module. For an explanation of the treatments, see Fig. 6 caption.

	CENTURY-based module												CERES-based module		
	C				N				C/N ratio				C	N	C/N ratio
Treatment	Total SOM	SOM1	SOM2	SOM3	Total SOM	SOM1	SOM2	SOM3	Total SOM	SOM1	SOM2	SOM3	HUM	HUM	HUM
	kg ha^{-1}												kg ha^{-1}		
Rothamsted long-term fallow															
Initialization	76 008	1 520	48 645	25 843	6 338	109	2 537	3 692	11.99	14.00	19.17	7.00	76 008	6 337	11.99
After 40 yr	28 564	131	3 913	24 520	3 765	11	244	3 510	7.59	11.80	16.07	6.99	63 960	5 347	11.96
Brazilian legume experiment															
Initialization	27 950.9	559.0	15 093.5	12 298.4	2 700.7	55.9	887.9	1 756.9	10.35	10.00	17.00	7.00	27 951.0	2 795.1	10.00
Bare fallow	26 261.5	468.8	13 490.5	12 302.2	2 588.4	37.3	793.9	1 757.2	10.15	12.58	16.99	7.00	27 513.9	2 751.4	10.00
Cajanus cajan	27 306.0	634.2	14 340.6	12 331.2	2 675.0	55.0	857.9	1 762.1	10.21	11.53	16.71	7.00	27 839.5	2 784.0	10.00
Calopogonium mucunoides	27 122.7	610.5	14 188.4	12 323.8	2 656.0	51.0	844.2	1 760.8	10.21	11.96	16.81	7.00	27 777.1	2 777.7	10.00
Canavalia brasiliensis	27 337.4	647.0	14 361.3	12 329.1	2 673.4	55.2	856.5	1 761.7	10.23	11.71	16.77	7.00	27 851.5	2 785.1	10.00
Canavalia ensiformis	27 307.5	639.3	14 339.5	12 328.7	2 675.4	56.6	857.1	1 761.7	10.21	11.30	16.73	7.00	27 905.1	2 790.5	10.00
Crotalaria striata	27 666.78	700.0	14 629.4	12 337.4	2 699.5	60.9	875.6	1 763.0	10.25	11.50	16.71	7.00	27 943.8	2 794.4	10.00
Pueraria phaseoloides	26 810.8	559.7	13 935.1	12 316.0	2 630.6	45.4	825.7	1 759.5	10.19	12.32	16.88	7.00	27 671.3	2 767.1	10.00
Mucuna aterrima	26 933.1	579.0	14 034.9	12 319.2	2 642.6	48.3	834.2	1 760.1	10.19	11.99	16.83	7.00	27 733.2	2 773.3	10.00

ability of N-rich residues, the SOM1 C/N ratio widened somewhat, indicating that the initialization value of C/N = 10 may have been too small, as newly formed microbes had C/N ratios around 12, within the allowable range from 6 to 14. The C and N contents of the SOM3 pool were hardly affected because the SOM3 pool received little new material. Its C/N ratio therefore remained unchanged.

The CERES-based module responded far less to changes in land use for the Brazilian experiment. The HUMN content remained almost the same for the treatments that included *C. striata* residues and declined only 438 kg ha^{-1} C for the bare fallow plot. The CERES-based module's humus remained at a fixed C/N ratio of 10. The lack of a microbial pool in this module and the fixed C/N ratio of newly formed humus mean that its SOM content and composition cannot respond well to changes in residue input.

CONCLUSIONS

The CENTURY-based module did an excellent job in simulating the development of the SOM content for the long-term bare fallow experiment. The CERES-based module, which is the original SOM–residue module of the present DSSAT version, performed very poorly and clearly demonstrated that it is not suitable for long-term simulations.

For short-term simulations with fresh residues added to the field, both modules gave a fair, though inconsistent, congruence between measured and simulated data. The relatively adequate job of the CERES-based module in simulating soil mineral N is noteworthy, considering its reported limitations. However, it raises questions of how widely applicable this module is; how sensitive it is to changes in system conditions, such as SOM levels, litter type, and soil disturbance; and how it behaves with soils of very different textures.

With the incorporation of the CENTURY-based module, DSSAT has become more flexible in handling different agricultural systems and more suitable for long-term simulations. This holds particularly for low-input systems where almost all nutrients are derived from SOM–residue decomposition and systems with a litter layer on top of the soil, such as green-manure–based smallholder systems. The DSSAT models also have become more responsive to systems that have different soil texture, and they accommodate an increased decomposition rate after soil disturbance.

A thick layer of residues on top of the soil will affect the soil temperature profile and water evaporation; the same holds for topsoil loosening when the material is incorporated. There is, however, no module yet that accommodates a potential *damping* effect of surface litter or a *mulching* effect of a loosened topsoil.

A word of caution: Gijsman et al. (1996) demonstrated the limited applicability of the CENTURY model to highly weathered, strongly P-sorbing soils. This applies also for the CENTURY-based module in DSSAT (though work is underway to correct this). The CERES-based module does not handle P.

REFERENCES

Algozin, K.A., V.F. Bralts, and J.T. Ritchie. 1988. Irrigation strategy selection based on crop yield, water, and energy use relationships: A Michigan example. J. Soil Water Conserv. 43:428–431.

Beare, M.H., M.L. Cabrera, P.F. Hendrix, and D.C. Coleman. 1994. Aggregate-protected and unprotected organic matter pools in conventional- and no-tillage soils. Soil Sci. Soc. Am. J. 58:787–795.

Boote, K.J., J.W. Jones, G. Hoogenboom, and N.B. Pickering 1998. The CROPGRO model for grain legumes. p. 99–108. *In* G.Y. Tsuji et al. (ed.) Understanding options for agricultural production. System approaches for sustainable agricultural development. Kluwer Academic Publ., Dordrecht, the Netherlands.

Bowen, W.T., J.W. Jones, R.J. Carsky, and J.O. Quintana. 1993. Evaluation of the nitrogen submodel of CERES-maize following legume green manure incorporation. Agron. J. 85:153–159.

Bowen, W.T., and P.W. Wilkens. 1998. Application of a decision support system (DSSAT) at the field level: Nitrogen management in variable-charged soils. p. 23–31. *In* Quantitative approaches in systems analysis, no. 16. Proc. Int. Workshop Inf. Technol. as a Tool to Assess Land Use Options in Space and Time, Lima, Peru. 28 Sept.–4 Oct. 1997. Centro Internacional de la Papa (CIP), Lima, Peru.

Campbell, C.A., Y.W. Jame, O.O. Akinremi, and H.J. Beckie. 1994. Evaluating potential nitrogen mineralization for predicting fertilizer nitrogen requirements of long-term field experiments. p. 81–100. *In* J.L. Havlin and J.S. Jacobsen (ed.) Soil testing: Prospects for improving nutrient recommendations. SSSA Spec. Publ. 40. SSSA, Madison, WI.

Gabrielle, B., and L. Kengni. 1996. Analysis and field-evaluation of the CERES models' soil components: Nitrogen transfer and transformations. Soil Sci. Soc. Am. J. 60:142–149.

Gijsman, A.J., A. Oberson, H. Tiessen, and D.K. Friesen. 1996. Limited applicability of the CENTURY model to highly weathered tropical soils. Agron. J. 88:894–903.

Gijsman, A.J., and C.H. Porter. 2002. The CENTURY-based soil-organic-matter/residue and soil-N module in DSSAT v.4.0—documentation and source code listing. Rep. 2002-602. Agric. and Biol. Eng. Dep., Univ. of Florida, Gainesville, FL (in press).

Gijsman, A.J., C.H. Porter, P.W. Wilkens, and G. Hoogenboom. 2002. The soil-organic-matter/residue and soil-N module in DSSAT v.4.0—documentation and source code listing. Rep. 2002-601. Agric. and Biol. Eng. Dep., Univ. of Florida, Gainesville, FL (in press).

Godwin, D.C., and C.A. Jones. 1991. Nitrogen dynamics in soil-plant systems. p. 287–321. *In* J. Hanks and J.T. Ritchie (ed.) Modeling plant and soil systems. Agron. Monogr. 31. ASA, CSSA, and SSSA, Madison, WI.

Godwin, D.C., and U. Singh. 1998. Nitrogen balance and crop response to nitrogen in upland and lowland cropping systems. p. 55–77. *In* G.Y. Tsuji et al. (ed.) Understanding options for agricultural production. System approaches for sustainable agricultural development. Kluwer Academic Publ., Dordrecht, the Netherlands.

Greenland, D.J., A. Wild, and D. Adams. 1992. Organic matter dynamics in soils of the tropics—from myth to complex reality. p. 17–33. *In* R. Lal and P.A. Sánchez (ed.) Myths and science of soils of the tropics. SSSA Spec. Publ. 29. SSSA and ASA, Madison, WI.

Hasegawa, H., J.M. Labavitch, A.M. McGuire, D.C. Bryant, and R.F. Denison. 1999. Testing CERES model predictions of N release from legume cover crop residue. Field Crops Res. 63:255–267.

Hassink, J., L.A. Bouwman, K.B. Zwart, J. Bloem, and L. Brussaard. 1993. Relationships between soil texture, physical protection of organic matter, soil biota, and C and N mineralization in grassland soils. Geoderma 57:105–128.

Jagtap, S.S., M. Mornu, and B.T. Kang. 1993. Simulation of growth, development and yield of maize in the transition zone of Nigeria. Agric. Syst. 41:215–229.

Jenkinson, D.S., D.D. Harkness, E.D. Vance, D.E. Adams, and A.F. Harrison. 1992. Calculating net primary production and annual input of organic matter to soil from the amount and radiocarbon content of soil organic matter. Soil Biol. Biochem. 24:295–308.

Jenkinson, D.S., P.B.S. Hart, J.H. Rayner, and L.C. Parry. 1987. Modelling the turnover of organic matter in long-term experiments at Rothamsted. INTECOL Bull. 15:1–8.

Jones, J.W., B.A. Keating, and C.H. Porter 2001. Approaches to modular model development. Agric. Syst. 70:421–443.

Kelly, R.H., W.J. Parton, G.J. Crocker, P.R. Grace, J. Klír, M. Körschens, P.R. Poulton, and D.D. Richter. 1997. Simulating trends in soil organic carbon in long-term experiments using the CENTURY model. Geoderma 81:75–90.

Kyuma, K. 1985. Fundamental characteristics of wetland soils. p. 191–206. *In* Wetland soils: Characterization, classification, and utilization. IRRI, Los Baños, Philippines.

Ladd, J.N., R.C. Foster, and J.O. Skjemstad. 1993. Soil structure: Carbon and nitrogen metabolism. Geoderma 56:401–434.

Lal, H., G. Hoogenboom, J.P. Calixte, J.W. Jones, and F.H. Beinroth. 1993. Using crop simulation models and GIS for regional productivity analysis. Trans. ASAE 36:175–184.

Li, C., S. Frolking, and R. Harris. 1994. Modeling carbon biogeochemistry in agricultural soils. Global Biogeochem. Cycles 8:237–254.

Loague, K., and R.E. Green. 1991. Statistical and graphical methods for evaluating solute transport models: Overview and application. J. Contam. Hydrol. 7:51–73.

Metherell, A.K., C.A. Cambardella, W.J. Parton, G.A. Peterson, L.A. Harding, and C.V. Cole. 1995. Simulation of soil organic matter dynamics in dryland wheat–fallow cropping systems. p. 259–270. *In* R. Lal, J. Kimball, E. Levine, and B.A. Steward (ed.) Soil management and greenhouse effect. CRC Press, Boca Raton, FL.

Parton, W.J., A.R. Mosier, D.S. Ojima, D.W. Valentine, D.S. Schimel, K. Weier, and A.E. Kulmala. 1996. Generalized model for N_2 and N_2O production from nitrification and dentrification. Global Biogeochem. Cycles 10:401–412.

Parton, W.J., D.S. Ojima, C.V. Cole, and D.S. Schimel. 1994. A general model for soil organic matter dynamics: Sensitivity to litter chemistry, texture and management. p. 147–167. *In* R.B. Bryant and R.W. Arnold (ed.) Quantitative modeling of soil forming processes. SSSA Spec. Publ. 39. SSSA, Madison, WI.

Parton, W.J., and P.E. Rasmussen. 1994. Long-term effects of crop management in wheat–fallow: II. CENTURY model simulations. Soil Sci. Soc. Am. J. 58:530–536.

Parton, W.J., J.W.B. Stewart, and C.V. Cole. 1988. Dynamics of C, N, P and S in grassland soils: A model. Biogeochemistry 5:109–131.

Paustian, K., W.J. Parton, and J. Perssons. 1992. Modeling soil organic matter in organic-amended and nitrogen-fertilized long-term plots. Soil Sci. Soc. Am. J. 56:476–488.

Powlson, D.S., P.S. Smith, and J.U. Smith (ed.) 1996. Evaluation of soil organic matter models using existing long-term datasets NATO ASI 138. Springer-Verlag, Berlin.

Probert, M.E., B.A. Keating, J.P. Thompson, and W.J. Parton. 1995. Modelling water, nitrogen, and crop yield for a long-term fallow management experiment. Aust. J. Exp. Agric. 35:941–950.

Quemada, M., and M.L. Cabrera. 1995. CERES-N model predictions of nitrogen mineralized from cover crop residues. Soil Sci. Soc. Am. J. 59:1059–1065.

Rawls, W.J., D.L. Brakensiek, and K.E. Saxton. 1982. Estimation of soil water properties. Trans. ASAE 25:1316–1328.

Seligman, N.C., and H. Van Keulen. 1981. PAPRAN: A simulation model of annual pasture production limited by rainfall and nitrogen. p. 192–221. *In* M.J. Frissel and J.A. Van Veen (ed.) Simulation of nitrogen behaviour of soil–plant systems. Centrum voor Landbouwpublikaties en Landbouwdocumentatie (PUDOC), Wageningen, the Netherlands.

Singh, U., P.K. Thornton, A.R. Saka, and J.B. Dent. 1993. Maize modelling in Malawi: A tool for soil fertility research and development. p. 253–273. *In* F.W.T. Penning de Vries et al. (ed.) Systems approaches for agricultural development. Kluwer Academic Publ., Dordrecht, the Netherlands.

Smith, P., J.U. Smith, D.S. Powlson, W.B. McGill, J.R.M. Arah, O.G. Chertov, K. Coleman, U. Franco, S. Frolking, D.S. Jenkinson, L.S. Jensen, R.H. Kelly, H. Klein-Gunnewiek, A.S. Komarov, C. Li, J.A.E. Molina, T. Mueller, W.J. Parton, J.H.M. Thornley, and A.P. Whitmore. 1997. A comparison of the performance of nine soil organic matter models using datasets from seven long-term experiments. Geoderma 81:153–225.

Thornton, P.K., and P.W. Wilkens. 1998. Risk assessment and food security. p. 329–345. *In* G.Y. Tsuji et al. (ed.) Understanding options for agricultural production system approaches for sustainable agricultural development. Kluwer Academic Publ., Dordrecht, the Netherlands.

Tsuji, G.Y., G. Uehara, and S. Balas (ed.). 1994. DSSAT v.3. Univ. of Hawaii, Honolulu.

Van Raij, B., and M. Peech. 1972. Electrochemical properties of some Oxisols and Alfisols of the tropics. Soil Sci. Soc. Am. Proc. 36:587–593.

Van Veen, J.A., and P.J. Kuikman. 1990. Soil structural aspects of decomposition of organic matter by micro-organisms. Biogeochemistry 11:213–233.

Van Veen, J.A., J.H. Ladd, and M.J. Frissel. 1984. Modelling C and N turnover through the microbial biomass in soil. Plant Soil 76: 257–274.

Verberne, E.L.J., J. Hassink, P. De Willigen, J.J.R. Groot, and J.A. Van Veen. 1990. Modelling organic matter dynamics in different soils. Neth. J. Agric. Sci. 38:221–238.

Vigil, M.F., D.E. Kissel, and S.J. Smith. 1991. Field crop recovery and modeling of nitrogen mineralized from labeled sorghum residues. Soil Sci. Soc. Am. J. 55:1031–1037.

Whitmore, A.P. 1991. A method for assessing the goodness of computer simulation of soil processes. J. Soil Sci. 42:289–299.

Volume 4
Chapter 9

Ceres-based Soil Organic Matter Module

UNDER CONSTRUCTION

Volume 4
Chapter 10

Soil Dynamic Processes Module

Oper Res Int J
DOI 10.1007/s12351-009-0067-1

ORIGINAL PAPER

Issues of spatial and temporal scale in modeling the effects of field operations on soil properties

Jeffrey W. White · Jim W. Jones · Cheryl Porter · Gregory S. McMaster · Rolf Sommer

Received: 5 January 2009 / Revised: 30 July 2009 / Accepted: 14 September 2009
© Springer-Verlag 2009

Abstract Tillage is an important procedure for modifying the soil environment to enhance crop growth and conserve soil and water resources. Process-based models of crop production are widely used in decision support, but few explicitly simulate tillage. The Cropping Systems Model (CSM) was modified to simulate tillage and related field operations for single seasons or multiple years. This paper provides an overview of how the new routines were implemented and discusses issues of spatial and temporal scaling that influenced the underlying strategy. The processes considered included effects of crop residues on the soil surface and on chemical and physical properties that vary with soil depth. Each event is described by date and implement used. The implement is characterized by its effects on soil properties, including mixing of soil layers and crop residues and changes in soil bulk density. The modeled responses are illustrated with a hypothetical case comparing effects of four implements (mold board plow, tandem disk, tine harrow, and planking) and a field experiment where winter wheat (*Triticum aestivum* L.) was grown with different tillage and residue management practices. From a modeling viewpoint, a key issue was how to manage different spatial and time scales. The soil is simulated as varying only with depth but in reality, the thickness of the soil is affected by tillage. This poses challenges for ensuring that the masses for water, nutrients, residues and

J. W. White (✉)
USALARC, USDA ARS, 21881 North Cardon Lane, Maricopa, AZ 85138, USA
e-mail: Jeffrey.white@ars.usda.gov

J. W. Jones · C. Porter
Department Agricultural and Biological Engineering, University Florida, Gainesville, FL, USA

G. S. McMaster
Agricultural Systems Research Unit, USDA ARS, Ft. Collins, CO, USA

R. Sommer
International Center for Agricultural Research in the Dry Areas (ICARDA),
P.O. Box 5466, Aleppo, Syria

Published online: 15 October 2009

the soil per se are conserved as soil layers are mixed and the density of each layer is altered. The model runs on a daily time step, but events such as tillage, application of residue, and irrigation can all happen within a single day and the sequence/timing can influence simulations. The new routines for field operations improve representation of tillage and residue management in the CSM model, but they are best viewed as providing a framework for future work that explicitly considers effects of residue type, soil type and distribution, and soil moisture on tillage effects and that deal with effects of rainfall kinetic energy in more detail.

Keywords Agriculture · Scale · Simulation · Soil · Tillage

Mathematics Subject Classification (2000) 90 · 92 · 93

1 Introduction

Agricultural fields are tilled for diverse reasons, including providing favorable conditions for planting and seedling establishment, controlling weeds, and incorporating crop residues and fertilizers. Leveling a field or forming furrows substantially alters field topography. Tillage can involve a diverse range of implements that often are designed and manufactured locally to match needs dictated by local farm practices and soil conditions. In addition to tillage operations, field operations such as planting and fertilizer applications also affect soil cover and physical conditions.

A field is tilled to benefit the current or future crops, but tillage can have numerous negative impacts and is often perceived as a major cause of environmental degradation (Lobb et al. 2007). Loosening of the soil surface and burial of surface residues increase the potential for soil erosion. Soil crusting and degradation of the natural structure of the soil can inhibit plant growth (Lipiec and Hatano 2003). Tillage also directly impacts scheduling of farm operations and of course, production costs. The need to conduct tillage when a field is neither too dry nor too wet can constrain planting and other operations, and long delays in tilling can prevent crops from being planted. Tillage and other operations are costly due to the large energy inputs and the costs of equipment and labor. Recognizing the multiple constraints and trade-offs, producers increasingly seek to minimize tillage (Knowler and Bradshaw 2007). Reduced and zero-tillage are usually combined with the practice of retaining crop residues. Such "conservation tillage" reduces production costs, conserves soil moisture, reduces erosion, and can improve soil health (Cannell and Hawes 1994; Hobbs et al. 2008). Producers thus face numerous, complex decisions relating to tillage, including which practices to apply (or withhold), when to till, and with what intensity. Beyond these are decisions to purchase, modify and maintain implements and tractors.

Process-based crop simulators are widely used to support decision making in agriculture. Applications include optimizing irrigation and nutrient management, precision farming, and assessing the potential impacts of climatic variation (as risk) and of climatic change. However, applications are often constrained because few models explicitly consider tillage or other field operations (see comparisons in Table

A. 1 of Sommer et al. 2007). The present paper describes new routines that allow field operations to be simulated in the Cropping Systems Model (CSM; Jones et al. 2003), which is distributed in the Decision Support System for Agrotechnology Transfer version 4.5 (DSSAT; Hoogenboom et al. 2009). Much of the description involves general principles that are relevant to other crop models. Two example applications illustrate features of the tillage routine. From an information systems perspective, programming the tillage routines is challenging due to the need to simulate processes occurring in a three dimensional space using a one dimensional representation of the soil and to consider events that involve different time scales. Variables referenced in the text are defined in Table 1.

2 Overview of tillage

Tillage alters the distribution of soil components and soil structure (the arrangement of aggregates of finer particles). These changes affect surface properties, the concentrations of particles, nutrients, organic matter and moisture within the profile, and the distribution and amount of pore spaces and channels.

Deep tillage such as with a moldboard plow is often performed to reduce compaction. Surface modifications include smoothing or loosening the soil surface in order to improve soil conditions for seed germination and seedling establishment. Larger changes may include leveling or formation of furrows or beds. When crops are well established, secondary tillage may be used to control weeds, hill soil around plants to stimulate root growth or reduce lodging, or seal soil cracks to reduce evaporation from deeper soil layers.

The potential for surface water runoff can be characterized using the runoff curve number (CN), which varies from 0 for no runoff to 100 for maximum runoff (Ritchie 1998). Surface residues affect infiltration and runoff, soil temperatures, and evaporation, so the portion of residue incorporated with tillage also needs to be tracked.

The mixing effect of tillage involves burial of residue and redistribution of soil particles. In actual field situations, this process largely involves bulk mixing where crop residues, soil particles and moisture are mixed in a similar manner somewhat independently of shape or density. It is mainly the type of farm implement and tillage intensity that determine the degree of mixing, the affected depth and the level of homogeneity of mixing (over depth). Soil type and wetness as well as the speed of tillage are further determining factors.

The soil loosening and compacting effects of tillage manifest themselves as changes in bulk density (the mass to volume ratio), which typically varies from 1.0 to 1.6 g cm^{-3} for dry soils differing in texture and structure. The bulk density of soil particles is approximately 2.66 g cm^{-3}. The lower values of soils mainly reflect the presence of air-filled pore spaces. Variation in pore space relates to packing of soil particles, differences in size of soil particles (particle size heterogeneity), as well as the structure provided by soil aggregates. Moderate tillage is often used to loosen the soil, i.e., decrease the bulk density, favoring root growth and water infiltration. Intensive tillage, however, can degrade the structure, thus in the long run increasing bulk density as compared to soils with intact aggregates.

Table 1 Definitions of variables used in the equations describing effects of tillage

Variable	Definition	Units	Name(s) in CSM
AS	Aggregate stability scaled from 0 to 1	Unitless	AS
$B_C(L)$	Current soil bulk density of a soil layer, L	g cm^{-3}	XBD
$B_T(L)$	Post-tillage event bulk density of a soil layer, L	g cm^{-3}	BDTEMP(L)
$B\%(M)$	Change in soil bulk density for a given implement and soil layer, M	%	SBDT, BDP(I, M)
CN	SCS soil runoff curve number	Unitless	CN
CN%	Change in SCS soil runoff curve number after a given tillage event	%	CN2T
F	Factor for change in soil properties due to effects of kinetic energy from rainfall and irrigations	cm^2 J^{-1}	RSTL
$H\%$	Reduction in hard pan (if tillage depth is sufficient to reach pan)	%	HPAN
$K\%(M)$	Change in K_{SAT} for a given tillage implement and soil depth, M	%	SKST, SWCNP(I, M)
$K_0(L)$	Saturated hydraulic conductance of a soil layer, L, prior to tillage	cm h^{-1}	XSWCN(NS)
$K_T(L)$	Saturated hydraulic conductance of a soil layer, L, after tillage	cm h^{-1}	SCTEMP(L)
L	Index for a soil layer	Unitless	L
M	Index of soil layer used to define implement effects	Unitless	M
$M\%$	Mixing efficiency of tillage event	%	MIXT, MIXPCT
N	Index for a soil layer	Unitless	NS
OC(L)	Soil organic carbon concentration	%	OC(L)
P	Total precipitation on a given day	mm	RAIN
R_0	Mass of residue which is on top of soil	kg ha^{-1}	MULCHMASS
R_T	Residue that is incorporated	kg ha^{-1}	RESINC
$R\%$	Percent of surface residue that is incorporated by a tillage event	%	RINP, TILRESINC
S	Fraction of soil surface covered by residues	m^2 m^{-2}	SOILCOV
SS	Soil settling rate	cm^2 J^{-1}	Srate
$S\%$	Percentage of soil surface is disturbed by a tillage event	%	SSDT
$U_S(L)$	Value of a soil parameter for layer L after modification by rainfall kinetic energy	(v)[a]	X
$U_T(L)$	Value of a soil parameter for layer L after tillage	(v)	X_t
$U_0(L)$	Value of a soil parameter for layer L prior to tillage	(v)	X_0
$X_M(L)$	Amount of a soil component available for mixing	(v)	AMT_2B_MIXED(L)
$X_0(L)$	Amount of a soil component to be partitioned between mixed and unmixed portions	(v)	VALUE_IN(L)
$X_T(L)$	Amount of soil component following tillage	(v)	VALUE_MIXED(L)
$X_U(L)$	Amount of soil component that is unmixed	(v)	AMT_UNMIXED(L)
$Y_L(L)$	Lower boundary of a soil layer, L	cm	SLB(L)
Y_T	Depth of tillage for a given tillage event	cm	TDEP
$\overline{Y}(L)$	Mean cumulative depth of a soil layer, L	cm	MCUMDEP
$Z(L)$	Thickness of a soil layer, L	cm	ZLAYER(L)

Table 1 continued

Variable	Definition	Units	Name(s) in CSM
ΣE_K	Cumulative kinetic energy since the last tillage event	J cm^{-2}	SUMKE
$\Sigma E_{K(t)}$	Cumulative kinetic energy through day t	J cm^{-2}	SUMKE
$\Sigma E_K(L)$	Cumulative kinetic energy on a given day that reaches a soil layer, L, with mean depth $\overline{Y}(L)$	J cm^{-2}	SUMKEL
ΣP_t	Cumulative (from previous days) precipitation or irrigation on day t	mm	CRAIN$_t$
ΣX_M	Total amount of a soil component to be mixed across all tilled soil layers	(v)	SUMMIXED
β	Proportional change in a soil property due kinetic energy effects of rainfall or irrigation	Unitless	KECHGE
$\theta(L)$	Volumetric soil water content	mm^3 mm^{-3}	SW(L)
$\theta_S(L)$	Saturated volumetric soil water content	mm^3 mm^{-3}	SAT(L), SATTEMP(L)
$\theta_U(L)$	Drained upper limit for soil water content	mm^3 mm^{-3}	DUL(L)
(na)[b]	Index for a tillage event	Unitless	I
(na)	Identifier used to link implement/operation from parameter file to field management file	Unitless	TIMPL
(na)	Day of a given tillage event	Date	TDATE
(na)	Whole profile soil drainage rate	mm day^{-1}	SWCON

The equivalent names used in the CSM parameter files and FORTRAN code are also given

[a] Units vary according to soil property

[b] Not applicable. Variable not used in equations in text

Severe compaction of soil below the tillage implement can result in formation of a plow pan, a layer of especially dense soil that roots cannot easily penetrate. To characterize effects of tillage, one thus needs to consider effects of a given operation on the bulk density, which varies by depth. Soils with similar bulk densities may differ in how easily water passes through a given layer, which is characterized by the hydraulic conductivity. The hydraulic conductivity of a soil is not a fixed value but changes with the water content of the soil. Still, for simplicity, water movement in soils is often simulated by considering only a whole profile drainage rate and/or a saturated hydraulic conductivity (K_{sat}). Thus, additional parameters are needed to characterize effects of tillage on drainage.

Effects of tillage on soil structure and other properties vary with implements used, the intensity of tillage, and soil conditions at the time of tillage. Four implements are compared qualitatively in Fig. 1, starting from a hypothetical soil that has three distinct sublayers. The mold board plow is assumed to penetrate to 30 cm soil depth. It mixes and loosens the soil, but leaves a rough surface. The tandem disk mixes and loosens to a lesser depth (approximately 15 cm), and it breaks large soil clumps, somewhat smoothing the soil surface. The tine harrow mixes and loosens only to about 8 cm but provides even more smoothing of the soil surface. Planking smoothes the surface with minimal mixing, but the smoothing operation increases the bulk density of the top layer (manifested as a decrease in thickness—here exaggerated).

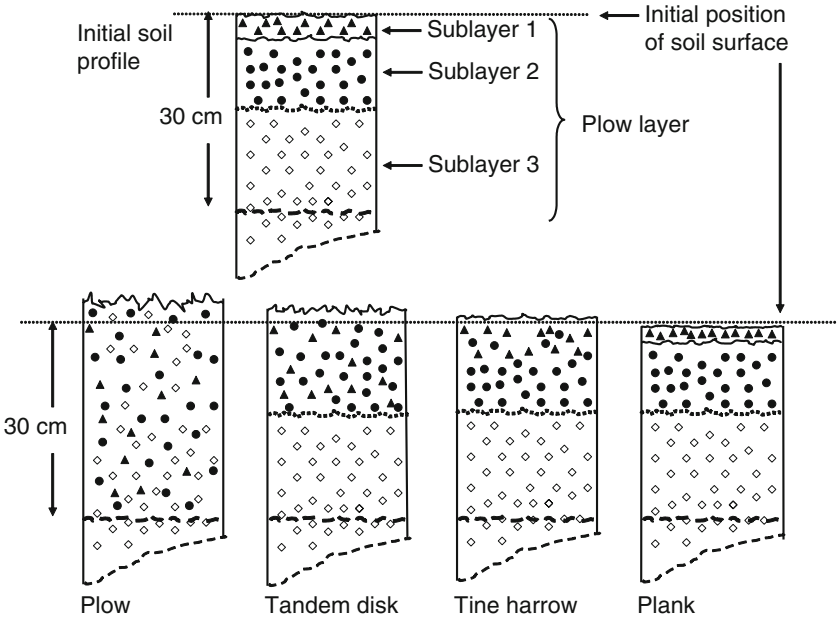


Fig. 1 Examples of a hypothetical soil profile showing effects of tillage with a plow, tandem disk, tine harrow, and planking. *Symbols* within profile layers indicate qualitative effects on soil mixing

Under actual field conditions, the impact of tillage varies with initial conditions of the soil, especially the moisture content, bulk density, and texture. Furthermore, the kinetic energy associated with precipitation or irrigation and the effect of water in reducing soil strength also affect the overall physical condition of the soil (tilth). The most notable effect of water is to increase the bulk density of soils loosened by tillage.

3 Implementation of tillage in CSM

The CSM model describes plant, atmospheric and soil processes and is implemented for over 20 crop species (Jones et al. 2003; Hoogenboom et al. 2009). The growth and development of a population of plants is simulated through routines that consider physiological processes such as phenological development, photosynthesis, and respiration. Partitioning rules allocate newly formed mass to different plant organs such as leaves, roots and stems. Leaves intercept solar radiation and are the site of uptake of CO_2 and loss (transpiration) of water vapor. Roots are the site of water and nutrient uptake from the soil. They grow downward in the soil in response to crop demand for water and nutrients, but growth is constrained by soil temperature and a root growth hospitality factor, which varies with soil depth.

The soil is described by surface properties and characteristics of soil layers and thus essentially has a one dimensional representation. Surface or whole profile

properties are the albedo (the reflectance for incoming shortwave radiation), runoff potential, and a drainage factor. Soil layers are identified by depth increments (e.g., 0–5 cm, 5–15 cm). Key parameters for a layer are the water retention characteristics, bulk density, soil texture (portions of sand, silt, clay and coarse material), and water, soil carbon and nutrient levels.

Water from precipitation or irrigation either infiltrates into the soil or is lost through surface runoff. Once in the soil, moisture may evaporate from the soil surface, drain below the bottom of the profile, or be taken up by roots. Runoff varies with daily rainfall (including intensity), CN and the amount of soil moisture at the time of the rainfall event. Infiltration is the difference between precipitation and runoff. The infiltrated water is allocated from the upper layer down through other soil layers following the so-called 'cascade' or 'tipping bucket' approach (Ritchie 1998). For each given soil layer (L), water contents at saturation ($\theta_S(L)$) and drained upper limit ($\theta_U(L)$) are assigned. The difference between $\theta_S(L)$ and the current soil water content ($\theta(L)$) determines the capacity of the layer to hold additional water. If the volume of water to be infiltrated exceeds this amount, then the excess water is allocated to the next layer until all infiltrated water is distributed among soil layers. After infiltration events, a fraction of water in excess of $\theta_U(L)$ is drained based on a whole-profile drainage rate constant (SWCON). If the value of $\theta(L)$ for the lowest soil layer exceeds $\theta_U(L)$, that excess water is assumed to drain out of the profile. If the potential drainage for a layer is very large, the net drainage may be limited by K_{sat}. A more complete explanation is given by Ritchie (1998).

Additional soil processes involve nutrient availability and decomposition of organic matter (mainly from crop residues). These processes are affected by soil moisture and temperature, but the key effects of tillage are indirect, involving incorporation and mechanical degradation of surface crop and weed residues, mixing of soil layers, and changes in bulk density and K_{sat}.

The tillage routines used in CSM are based on procedures first developed by Dadoun (1993) as the CERES-Till model for maize and refined by Andales et al. (2000) for the CROPGRO-Soybean model. The timing and type of tillage are specified in a crop management input file, which also provides data on cultivars planted, the field environment, planting dates and arrangements, and irrigation and fertilizer regimes. This file can describe multiple treatments such as found in field experiments and has a relational architecture (Hunt et al. 2001). The tillage section (Table 2) of the crop management file specifies an identifier for the tillage event, the date of the event, an implement code to identify the implement used, the depth of tillage, and a name or description. The implement code links to a parameter file that specifies the effects of different implements. Multiple instances of a single operation are allowed, such as when a field is disked with multiple passes to achieve a desired tilth.

An input file describing implement effects provides two sets of parameters (Table 3). The first set mainly concerns tillage effects on the soil surface. The four most important parameters are the percent change in CN immediately after an operation (CN%), the percent of surface residue incorporated (R%), the percent of the soil surface area that is disturbed by the field operation (S%), and the mixing efficiency of tillage event (M%). A parameter to specify a reduction in the hard pan

Table 2 Example of the portion of a control file for the CSM model that specifies tillage or other field operations affecting soil properties

*TILLAGE AND ROTATIONS				
@T	TDATE	TIMPL	TDEP	TNAME
1978 Hypothetical tillage in bare fields at Gainesville, FL				
1	78182	TI042	25	Moldboard plow, 30 cm depth
1	78212	TI042	25	Moldboard plow, 30 cm depth
1	78242	TI042	25	Moldboard plow, 30 cm depth
1	78272	TI042	25	Moldboard plow, 30 cm depth
2	78182	TI009	15	Tandem disk
2	78212	TI009	15	Tandem disk
2	78242	TI009	15	Tandem disk
2	78272	TI009	15	Tandem disk
3	78182	TI015	10	Tine harrow
3	78212	TI015	10	Tine harrow
3	78242	TI015	10	Tine harrow
3	78272	TI015	10	Tine harrow
4	78182	TI023	4	Plank
4	78212	TI023	4	Plank
4	78242	TI023	4	Plank
4	78272	TI023	4	Plank
1996–1997 Tillage experiment at Ft. Collins, CO				
1	96289	TI031	7	No-till drill
2	96247	TI003	21	Moldboard plow
2	96252	TI039	10	Roller harrow
2	96289	TI031	7	No-till drill

The field *T* provides an identifier that links to a master table defining all combinations of treatments (e.g., cultivars, planting dates and fertilizer levels) to be simulated, TDATE is the date of the tillage event, TIMPL is a code to identify the field operations that allows linking treatments to a file containing parameters that describe effects of specific implements or operations (examples of values are given in Table 3), TDEP is the depth of tillage in centimeters, and TNAME is the name assigned to the specific tillage event

is also defined, but effects of a hard pan have not yet been implemented in the CSM model.

Two effects of tillage vary with depth, indicated by the lower boundary of the affected depth (Table 3). The first effect is the percent change in bulk density ($B\%(M)$) for soil layer M, and the second is the percent change in K_{sat} ($K\%(M)$). The effects for a given implement and depth may be positive or negative.

The implement file currently contains values for 36 implements or operations that range from moldboard plows to hand hoeing (Table 3). To simulate field operations such as planting that may also affect soil properties (e.g., through soil compaction), entries for seed drills and planters are included. Default values are provided for

Table 3 Examples of parameters provided as default settings to describe effects of different tillage practices or other field operations in the cropping systems model

Implement	Implement code[a]	Change in curve no.	Residue incorporated (%)	Soil surface disturbed (%)	Mixing efficiency (%)	Soil depth (cm)	Change in bulk density (%)	Change in K_{SAT} (%)
Moldboard plow, 30 cm depth	TI0042	−10	95	100	90	25.0	−10	10
						30.0	5	−5
Moldboard plow, 20 cm depth	TI003	−10	95	100	90	15.0	−10	10
						20.0	5	−5
Hand hoeing	TI040	−5	5	100	50	5.0	−5	5
No-till seed drill	TI034	0	20	20	5	5.0	5	0
Plank	TI023	5	20	100	20	4.0	10	−5
Planting stick	TI035	0	10	10	0	5.0	0	0
Roller harrow	TI039	−10	20	100	55	15.0	5	−5
Tandem disk	TI009	−10	50	100	20	7.5	−5	5
Tine harrow	TI015	−5	15	100	20	5.0	−5	5

The actual parameter file is TILOP045.SDA

[a] Corresponding variable names in the parameter file are: implement code, TIMPL; change in curve number, CN2T; residue incorporated, RINP; soil surface disturbed, SSDT; mixing efficiency, MIXT; soil depth, SLB; change in bulk density, SBDT; and change in K_{SAT}, SKST

testing, but users are expected to modify the values based on their tillage implements and soil conditions.

3.1 Effects of tillage via soil mixing

Individual soil components are mixed across soil layers based on the mixing value that corresponds to the tillage implement and the range of soil depths. The routine starts by comparing the depth of tillage specified for the tillage event (Y_T) with the cumulative soil depth to determine which layers (or portions of layers) require mixing. For each soil layer L and a given constituent $X_0(L)$ (e.g., soil moisture or nitrate), the constituent is apportioned into fractions to be mixed, $X_M(L)$, or left unmixed, $X_U(L)$, based on the mixing percentage, $M\%$ for the tillage practice. Thus,

$$X_M(L) = X_0(L) \times M\%/100$$

$X_U(L)$ is calculated as the remaining portion of $X_0(L)$, and $X_M(L)$ is added to the total amount to be mixed (over all layers affected by tillage), ΣX_M. The mixed portion is then added back into each layer to give $X_T(L)$, the amount of the soil component following tillage,

$$X_T(L) = \Sigma X_M \times Z(L)/Y_T + X_U(L)$$

where $Z(L)$ is the depth of the layer L. $Z(L)$ is reduced if only a portion of a layer is tilled.

Surface residues are incorporated separately in routines for soil organic matter, recognizing separate components of carbon, nitrogen and phosphorus. Thus for carbon, the incorporated residue carbon, R_T, is allocated from the total residue mass, R_0, as:

$$R_T = R_0 \times R\%/100$$

where $R\%$ is the portion of R_0 incorporated, as specified for the implement. R_T is then allocated uniformly through the soil layers up to Y_T. Currently, the model does not account for spatial variation in residue incorporation such as from strip tillage, although the implement parameter file allows for specifying the percent of the soil surface that is disturbed by a given field operation. For further information on simulation of soil organic matter dynamics in CSM see the paper by Porter et al. in this special issue (Porter et al. 2009).

3.2 Effects of tillage on soil properties exclusive of mixing effects

The primary effect of tillage on soil physical properties is on bulk density, which varies through the profile and affects the saturated water content, $\theta_S(L)$. Tillage can increase or decrease bulk density, which requires slightly different processing, but we discuss only the case of a decrease in soil bulk density.

To simulate the effects of tillage on bulk density, three soil depths indices are required. The soil depths for the effect of an implement are indexed by M. The depths in the field soil are indexed by N. The third index, L, is a combined index for depths of tillage and soil layers, which is required since the depths indexed by M and N seldom match.

For each soil layer, the current bulk density, $B_C(N)$, is reduced by the implement-specific tillage effect, $B\%(M)$,

$$B_T(L) = B_C(N) \times (1.0 + B\%(M)/100)$$

For tillage that reduces bulk density, values of $B\%(M)$ are negative.

The saturated water content for a soil layer L, $\theta_S(L)$, is calculated from the bulk density by assuming that soil particles have a density of 2.66 g cm^{-3} and that 95% of the air space can be occupied by water,

$$\theta_S(L) = 0.95 \times (1.0 - (B_T(L)/2.66)).$$

Effects of tillage on K_{SAT} are also specified as a percent change that varies with soil depth, $K\%(M)$. Thus, the value prior to tillage, $K_0(N)$, is adjusted accordingly to give the tilled value, $K_T(L)$.

$$K_T(L) = K_0(N) \times (1.0 + K\%(M)/100).$$

After calculating tillage effects on a given day, further processing is required to redistribute the soil properties, so that the post-tillage soil layers correspond to pre-tillage soil layers. Soil layer thicknesses are recomputed based on post-tillage bulk densities to preserve mass for all soil constituents.

3.3 Soil settling or loosening with rainfall (from SOILDYN.FOR)

When effects of tillage on bulk density are simulated, the processes of soil settling or loosening due to rainfall or irrigation also need to be considered. Settling and loosening of soil are assumed to be driven by the kinetic energy of water from rainfall and irrigation. Since the model uses a daily time step, these effects are simulated on a daily basis. A relation was estimated by regression analysis of rainfall data of Treynor, Iowa, by Andales et al. (2000) as:

$$\Sigma E_{\mathrm{K}(t)} = \Sigma E_{\mathrm{K}(t-1)} + 0.00217 \times \Sigma P_t$$

where $\Sigma E_{\mathrm{K}(t)}$ is the cumulative kinetic energy on day t, with units of J cm^{-2}, and ΣP_t is the cumulative daily rainfall in millimeters for the same period. Currently, irrigations are assumed to impart the same kinetic energy as rainfall, with the exception that drip irrigation imparts no energy.

In the routine for effects of kinetic energy, a flag indicating that tillage has occurred is set the first day that the current bulk density of any soil layer differs from the initial bulk density of that layer by more than 0.01 g cm^{-3}. The cumulative kinetic energy since the last tillage event, ΣE_{K}, is then reset to zero. Each day that there is rainfall (or an irrigation), the daily total, P in mm, is used to calculate the increase in kinetic energy. Soil cover (S, as a fraction from 0 to 1) is assumed to protect the soil, so the actual increment is

$$\Sigma E_{\mathrm{K}(t)} = \Sigma E_{\mathrm{K}(t-1)} + 0.00217 \times P \times (1 - S)$$

where t and $t-1$ denote the current and previous days, respectively.

The effective kinetic energy affecting a layer L, $\Sigma E_{\mathrm{K}}(L)$, is assumed to decrease exponentially with mean soil depth of the layer, $\overline{Y}(L)$,

$$\Sigma E_{\mathrm{K}}(L) = \Sigma E_{\mathrm{K}} \times \mathrm{e}^{-0.05 \times \overline{Y}(L)}.$$

The proportional change in a given soil property, β, is determined by the product of $\Sigma E_{\mathrm{K}}(L)$ and a rate of change in the soil property, F, with units of cm^2 J^{-1},

$$\beta = \mathrm{e}^{-F \times \Sigma E} K^{(L)}$$

The value of F varies with the relative strength of the soil aggregates, AS (scaled from 0 to 1), which is assumed to increase with the organic content of the soil, OC(L) expressed as a percentage. Thus, AS is estimated as

$$\mathrm{AS} = 0.205 \times \mathrm{OC}(L)$$

and F as

$$F = \mathrm{SS} \times (1.0 - \mathrm{AS})$$

where SS is a soil settling rate, currently assigned a value of 30 cm^2 J^{-1} for all soils and soil parameters.

Paralleling the effects of tillage on soil properties, rainfall also affects bulk density, saturated hydraulic conductivity, saturated water content, and a root growth impedance factor. The same form of effect is assumed for all parameters,

$$U_S(L) = U_0(L) + ((U_T(L) - U_0(L)) \times \beta)$$

where $U_S(L)$ is the value of the parameter after settling or recovery from compaction. Note that β proportionally reduces the difference of the tilled value from the prior value, regardless of whether tillage resulted in a decrease (settling) or increase in a given soil parameter.

4 Evaluation of the model

To illustrate the response of the model to tillage, two cases were simulated. The first is a hypothetical case of a bare field tilled with four implements. The second is for a tillage and residue management study involving winter wheat (McMaster et al. 2002).

4.1 Sensitivity analysis with four implements

The simulations of a bare field used the four implements whose qualitative effects are presented in Fig. 1. The soil was a Millhopper fine sand. Weather data were used from Gainesville, Florida, USA. The field was tilled once every 30 days, starting on 1 July 1978 and continuing for four tillage dates. Three soil layers were considered: 0–5 cm, 5–15 cm and 15–30 cm.

The first series of simulations excluded rainfall and irrigations, so the effects on bulk density are due only to tillage. The deepest tillage effect was from the mold board plow, which was assumed to reach 25 cm deep, as specified in Tables 2 and 3. Between the soil surface and 5 cm depth, each plowing loosened the soil, reducing the bulk density (Fig. 2a). In the 5–15 cm layer (Fig. 3a), plowing still decreased bulk density, but for the 15–30 cm layer (Fig. 3b), there was little change in bulk density. The latter response reflects counterbalancing effects of tillage within the layer. For the upper portion of 15–30 cm, the parameters for plowing reduced bulk density, for the middle portions of the layer, bulk density increased, and for the lower portion, no change was expected because plowing only reached to 25 cm.

When an effect of rainfall was considered, the loosening effect of plowing was lost if there was sufficient rainfall. Thus, after the first tillage, bulk density returned to the initial value within 10 days, whereas after the third tillage, when there was much less rainfall, the change in bulk density was largely sustained (Fig. 2b).

Disking penetrated only to 7.5 cm, so most of the effects were in the top 5 cm of soil (Figs. 2a, 3a, b). Tine harrowing and planking only affected this top soil layer. Rather than loosening the soil, planking smoothed and compacted the soil (e.g., for improved seed-soil contact). The planking effect was largely offset by rainfall events (Fig. 2b).

Figure 4 shows additional effects of the plow, tandem disk and tine harrow on soil water content, excluding rainfall. With each tillage event, the uppermost soil layer briefly increased in moisture content due to soil mixing (Fig. 4a), mirrored as a decrease in soil moisture in the second layer (Fig. 4b). As for effects on bulk density, moisture effects in the 15–30 cm layer were minimal except with plowing (Fig. 4c).

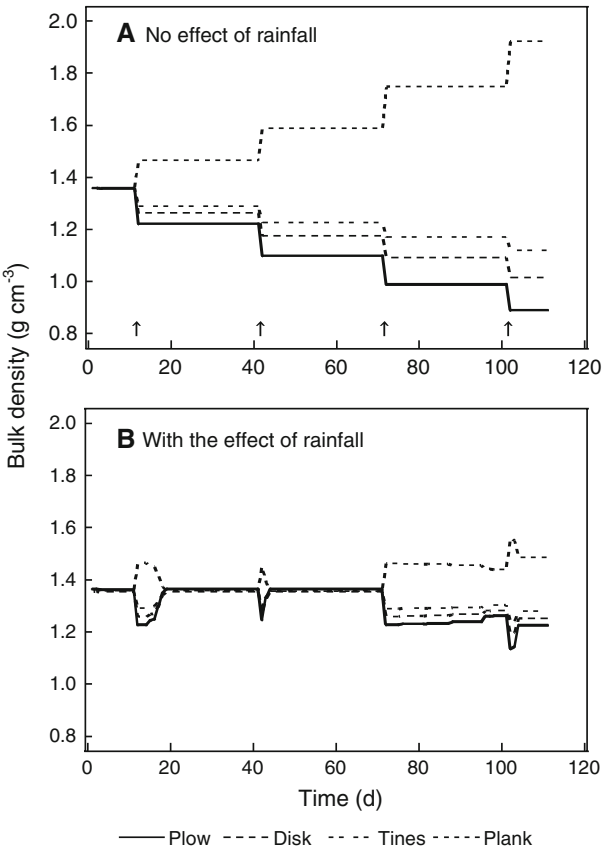


Fig. 2 Simulated changes in soil bulk density for the 0–5 cm soil layer for tillage every 30 days by a plow, tandem disk, tine harrow, and planking. **a** With no effect of rainfall to cause settling or to relieve compaction. **b** With the effect of rainfall. *Upward arrows* in **a** indicate dates of the four tillage events

4.2 Application to a winter wheat tillage experiment

Winter wheat in northern Colorado, USA, is normally grown under rainfed conditions, and water deficits usually limit grain yield. The crop is sown from mid-September to mid-October. The crop over-winters in a semi-dormant phase, resumes growth in the spring, and matures in July. McMaster et al. (2002) conducted an experiment over 7 years to compare effects of pre-planting tillage vs. no-till with three levels of crop residues (no residue, normal mass of residue from the previous wheat crop, and double the mass from a previous crop). No-till was expected to conserve soil moisture prior to planting, resulting in better seed germination and seedling growth. Improved crop establishment should also allow the crop to compete better with weeds. Early crop growth and residue are important in capturing more snow, so the enhanced growth from no-till plus benefits of the residue should further enhance spring growth. The conventional tillage consisted of moldboard plowing to 21 cm soil depth, followed about 1 week later by roller harrowing.

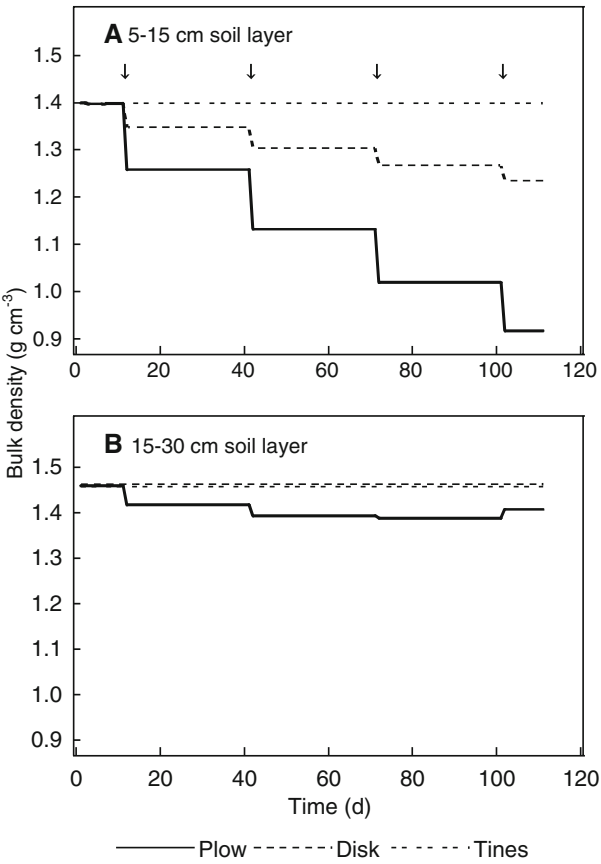


Fig. 3 Simulated changes in soil bulk density for the 5–15 cm and 15–30 cm soil layers for tillage every 30 days by a plow, tandem disk, and tine harrow. No effect of rainfall to cause settling or to relieve compaction was simulated. **a** 5–15 cm soil layer. **b** 15–30 cm soil layer. *Downward arrows* in **a** indicate dates of the four tillage events

Results are presented for the 1994–1995 and 1996–1997 seasons. The 1994–1995 season received over 100 mm more precipitation than the 1996–1997 season (Fig. 5a), but the two seasons had nearly identical patterns of potential demand for water, as indicated by potential evapotranspiration (Fig. 5b). Observed and simulated end-season above-ground dry matter was higher in 1994–1995 (Fig. 6). In part, the growth difference was likely due to the 1996–1997 crop being planted about a month later (14 Sept. 1994 vs. 15 Oct. 1996). In both seasons, the no-till treatments performed better than the plowed treatments, and this difference was reflected in the simulations. Subsequent discussion focuses on the no-till and plowed treatments, considering only the treatments with the normal residue level.

The soil layer that extends from 5 to 15 cm deep—this is a nominal range since the depth varies with tillage—best indicated the effects of tillage because responses in this layer were less dynamic than the surface layer and because the second layer

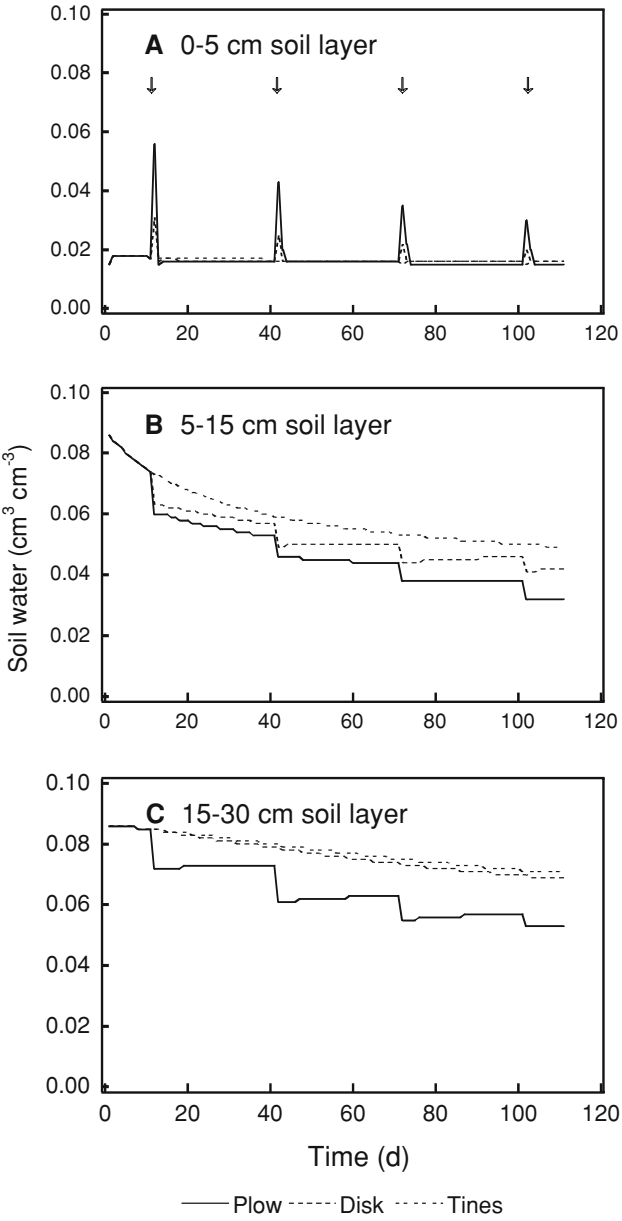


Fig. 4 Simulated changes in soil water content for three soil layers tilled every 30 days by a plow, tandem disk, and tine harrow. No effect of rainfall to cause settling or to relieve compaction was simulated. **a** 0–5 cm soil layer. **b** 5–15 cm. **c** 15–30 cm. *Downward arrows* in **a** indicate dates of the four tillage events

supplied an important portion of moisture to the crop, especially during early growth (data not shown). In both seasons, the bulk density decreased with plowing (Fig. 7a, b), but the return to a value around 1.3 g cm^{-3} was slower in 1994–1995

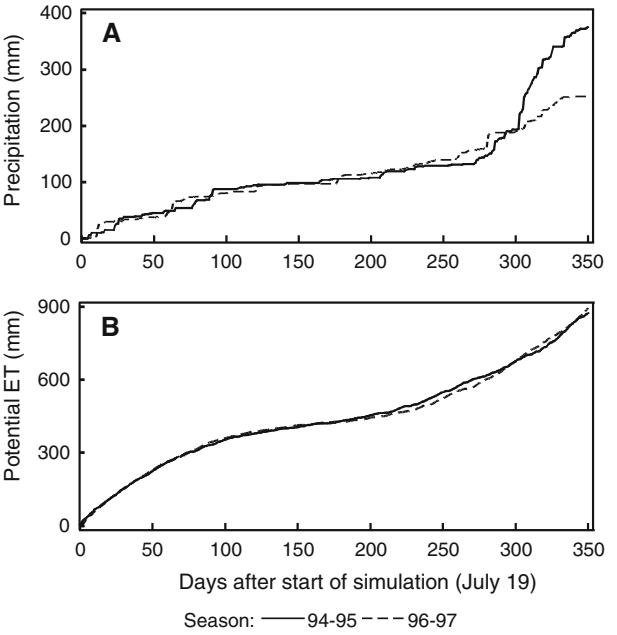


Fig. 5 Cumulative precipitation and potential evapotranspiration for the tillage experiments at Ft. Collins, CO for the 1994–1995 and 1996–1997 cropping seasons. **a** Precipitation. **b** Potential evapotranspiration

because there was less rainfall after the tillage event. The no-till treatment showed a negligible change in bulk density due to an effect of the seed drill.

The effect of soil mixing is seen in the large differences in soil moisture between the tillage treatments after plowing. Mixing of the 5–15 cm layer with drier soil from the top layer substantially decreased soil moisture. In 1994–1995, however, 20 mm of precipitation from day 88–92 partially replenished the soil. Since the no-till crop had access to more moisture during early growth, plants were larger and extracted a greater volume of water, causing the decline in moisture from day 90 to 150 in the no-till treatment. In 1996–1997, soil moisture started higher in the no-till crop, but the two treatments converged as the crop progressed.

Space limitation does not allow a more detailed discussion of how the simulated differences relate to field performance, including effects of crop residue on yield. However, the described results provide a qualitative view of the effects of tillage on soil bulk density and moisture over time.

5 Discussion

The implementation of tillage in CSM represents an important advance in allowing users to examine effects of tillage and residue management on crop production and soil resources. The qualitative responses appear realistic, and basic tests for

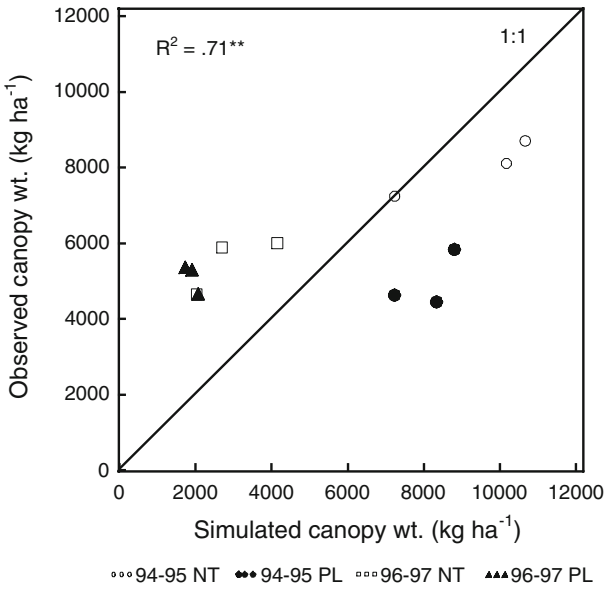


Fig. 6 Comparison of simulated versus observed effects of no-till (NT) and conventional tillage with a plow (PL) on winter wheat above ground (canopy) dry matter at Ft. Collins, CO for the 1994–1995 and 1996–1997 cropping seasons. Three values within each tillage and season combination are no residue, normal residue and 2× residue treatments

conservation of mass indicate that the complexities of soil mixing and changes in bulk density are properly implemented. Further testing is needed to refine the parameterization of different tillage practices, but the results from the wheat study suggest that the model already is suitable for exploring effects of various tillage practices. Among possible applications are assessing the potential for zero or reduced tillage to conserve soil moisture in different regions and estimating effects of residue incorporation on soil carbon.

The current implementation, however, is viewed largely as providing a framework for further evaluation and future improvement. Issues of spatial and temporal scale are discussed below. The effects of specific types of tillage also merit further study. Variation in crop residue types is ignored, but Buckingham et al. (2007) indicated that degree of incorporation varies with the texture of the residue. Effects of tillage also should vary with soil type and moisture content. The possibility of describing soil mixing by soil fauna, which is especially important in conservation tillage (Kladivko 2001), also merits study.

5.1 Problems of scale in modeling tillage

Among the challenges encountered in implementing the tillage effects were issues of spatial and temporal scales. Soils are represented in a single dimension using soil layer increments that usually are 5 cm or thicker. Unidimensionality constrains simulation of any management that involves furrows, hilling or horizontal

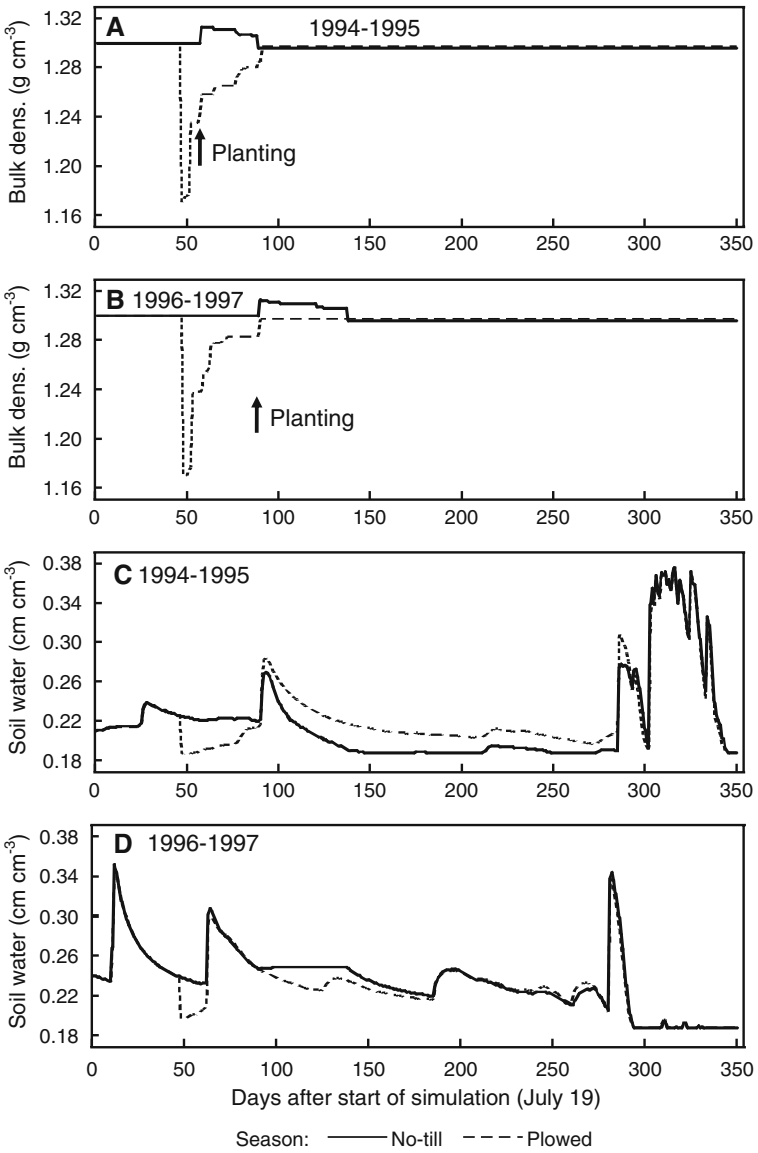


Fig. 7 Comparison of effects of no-till (NT) and conventional tillage with a plow (PL) on simulated bulk density and soil moisture for the 5–15 cm soil layer (nominal depth) at Ft. Collins, CO for the 1994–1995 and 1996–1997 cropping seasons and normal residue. **a** Soil bulk density for 1994–1995. **b** Soil bulk density for 1996–1997. **c** Soil water for 1994–1995. **d** Soil water for 1996–1997

distribution of crop residues on the soil surface. Temporal scale is an issue when dealing with events whose sequencing within a day would impact model outputs and in estimating the kinetic energy of rainfall. We examine these issues in the context of the CSM model.

Various tillage practices modify the topography of the soil surface. The passage of tractor wheels can be a major contributor to compaction as well as create ruts that become channels for runoff (Défossez et al. 2003). Furrows or beds are created to facilitate irrigation management or other operations. Furrows concentrate water in a limited portion of the soil, which can reduce the portion of the soil surface that becomes moist and thus has a large impact on soil evaporation. An extreme instance is with alternate row irrigation, where the wetted surface is limited to every other furrow. Depending on the orientation of the furrow to the slope of the field, furrows also may substantially increase or decrease potential for runoff. Strip-tillage involves tillage conducted parallel to crop rows and that disturbs less than the full width of a row. It is often performed to prepare a good seed-bed while otherwise limiting disturbance to soil and residues (Bolton and Booster 1981). Besides providing good tilth for seed germination, strip-tillage can result in a drier and warmer seed bed (Licht and Al-Kaisi 2005). Hilling involves heaping soil higher than the original base of the plant. Functions include protecting harvested materials (e.g., to prevent green skins in potato tubers), improving harvest quality (blanching of vegetables), stimulating growth, controlling weeds, and reforming furrows.

Some models consider a two dimensional soil, but they do not consider topography at sub-meter scales, which would be required to handle furrows, strip-tillage or effects of wheel tracks. A solution intermediate to a two or three dimensional representation of the soil surface with residues would be to define two zones, one as tilled and the other as undisturbed. Processes in these zones would only be simulated separately until partial canopy cover was achieved. For example, water balances and soil temperatures could be calculated for each zone. The approach seems best suited for models that estimate canopy width and track sunlit and shaded soil surfaces. Similarly, effects of furrows on runoff, infiltration and evaporation might be modeled either by partitioning the soil surface into zones or using empirical factors that adjust relevant processes based on the presence of furrows.

One difficulty with temporal scales concerns timing of management practices. For example, whether the soil is tilled prior to or after the fertilizer has been applied to the soil surface affects how the fertilizer is distributed in the soil. When events occur within a single day, CSM assumes a fixed order for management events based on expectations from the most common practices. Fertilizers and residues are applied first, followed by tillage, planting, and irrigation. Effects of rainfall on soil compaction are estimated at the same time as irrigations. Thus, for a given day, an ordinal time scale is assumed. While continuous scheduling with actual start and completion times could be implemented, this approach would substantially increase model complexity because of the need to consider simultaneous or overlapping processes. Irrigations often are conducted overnight and are sometimes used to deliver fertilizer. Tilling a large field may require multiple days, while "one pass" implements may perform strip-tillage, apply a fertilizer, and plant the crop in one operation. Relatively few users would require precise accounting of timing, and the additional net benefit to CSM users likely would be offset by the need to request detailed inputs, increasing the potential for errors. A compromise would be to allow users to input the order of a series of events and field operations within a single day.

Estimation of the kinetic energy from rainfall or irrigation also involves temporal scale. Rainfall intensity varies at scales of minutes, but CSM infers characteristics of intensity from daily rainfall. The review by van Dijk et al. (2002) noted that there is great uncertainty in estimating kinetic energy, even when the droplet size distribution is characterized. Where rainfall intensity data are available, daily totals of rainfall kinetic energy might be estimated and included as an additional input. Alternately, one might use rainfall disaggregation algorithms to infer the rainfall intensity (e.g., Kandel et al. 2004; Debele et al. 2007), and thence the intensity and kinetic energy. Most forms of irrigation are currently assumed to provide the same kinetic energy as rainfall, which is unrealistic. Analogous problems involving rainfall and irrigation intensity exist with estimation of runoff using daily totals, so better estimation of intensity or kinetic energy might improve estimation of runoff as well.

A common thread to the concerns over spatial and temporal scales is that for over 20 years, simple approaches for representing space and time have worked well in CSM and its predecessors. This simplicity undoubtedly contributed to the widespread adoption of the CSM model. As needs of users evolve, simplifications impose constraints. However, further modifications, such as subdividing the upper soil layer to accommodate strip-tillage, may be less expedient than they first seem. Introducing multiple approximations makes the code more complex and difficult to maintain and may require users to specify values for parameters that are not readily measured from basic physical or biological responses. The general expectation is that increasing the number of parameters can decrease model reliability due to propagation of parameter errors (Reynolds and Acock 1985). Thus, the question of appropriate temporal and spatial scales is also linked to long-discussed tradeoffs between models that provide accurate predictions for immediate decision support use vs. ones that attempt to fully integrate the known biological, chemical and physical processes (Nihoul 1994; Passioura 1996).

6 Conclusions

The ability of CSM to simulate effects of tillage increases the potential applications of the model, especially in relation to water, nutrient and soil organic matter management. The simple approach based mainly on consideration of soil mixing and changes in bulk density, runoff, and soil water holding capacity provides a framework for further improvements. However, as capabilities of the model are expanded, the number of parameters required increases, increasing the potential for errors in parameterization and as well as making the model more difficult to use correctly. For the CSM model, which uses a one dimensional soil profile and a daily time step, further enhancements to representation of tillage might require a more complex representation of the soil and use of finer time steps. Alternatively, judicious subdivision of the soil surface and chronological indexing of different management events might improve the robustness and applicability of the model.

References

Andales AA, Batchelor WD, Anderson CE (2000) Incorporating tillage effects into a soybean model. Agric Syst 66:69–98

Bolton FE, Booster DE (1981) Strip-till planting in dryland cereal production. Trans ASAE 24:59–62

Buckingham F et al (2007) Tillage. Deere & Company, Moline

Cannell RQ, Hawes JD (1994) Trends in tillage practices in relation to sustainable crop production with special reference to temperate climates. Soil Till Res 30:245–282

Dadoun FA (1993) Modeling tillage effects on soil physical properties and maize (*Zea mays* L.) development and growth. Unpublished Ph.D. thesis, Michigan State University, MI

Debele B, Srinivasan R, Yves Parlange J (2007) Accuracy evaluation of weather data generation and disaggregation methods at finer timescales. Adv Water Resour 30:1286–1300

Défossez P, Richard G, Boizard H, O'Sullivan MF (2003) Modeling change in soil compaction due to agricultural traffic as function of soil water content. Geoderma 116:89–105

Hobbs PR, Sayre K, Gupta R (2008) The role of conservation agriculture in sustainable agriculture. Philos Trans R Soc B Biol Sci 363:543–555

Hoogenboom G, Jones JW, Wilkens PW, Porter CH, Batchelor WD, Hunt LA, Boote KJ, Singh U, Uryasev O, Bowen WT, Gijsman AT, White JW, Tsuji GY (2009) Decision support system for agrotechnology transfer, DSSAT4.5. University of Hawaii, CD-ROM, Honolulu

Hunt LA, White JW, Hoogenboom G (2001) Agronomic data: advances in documentation and protocols for exchange and use. Agric Syst 70:477–492

Jones JW, Hoogenboom G, Porter CH, Boote KJ, Batchelor WD, Hunt LA, Wilkens PW, Singh U, Gijsman AJ, Ritchie JT (2003) DSSAT cropping system model. Eur J Agron 18:235–265

Kandel DD, Western AW, Grayson RB, Turral HN (2004) Process parameterization and temporal scaling in surface runoff and erosion modelling. Hydrol Processes 18:1423–1446

Kladivko EJ (2001) Tillage systems and soil ecology. Soil Till Res 61:61–76

Knowler D, Bradshaw B (2007) Farmers' adoption of conservation agriculture: a review and synthesis of recent research. Food Policy 32:25–48

Licht MA, Al-Kaisi M (2005) Strip-tillage effect on seedbed soil temperature and other soil physical properties. Soil Till Res 80:233–249

Lipiec J, Hatano R (2003) Quantification of compaction effects on soil physical properties and crop growth. Geoderma 116:107–136

Lobb DAR, Huffman E, Reicosky DC (2007) Importance of information on tillage practices in the modelling of environmental processes and in the use of environmental indicators. J Environ Manage 82:377–387

McMaster GS, Palic DB, Dunn GH (2002) Soil management alters seedling emergence and subsequent autumn growth and yield in dryland winter wheat-fallow systems in the Central Great Plains on a clay loam soil. Soil Till Res 65:193–206

Nihoul JCJ (1994) Do not use a simple model when a complex one will do. J Mar Syst 5:401–406

Passioura JB (1996) Simulation models: science, snake oil, education, or engineering? Agron J 88:690–694

Reynolds JF, Acock B (1985) Predicting the response of plants to increasing carbon dioxide: a critique of plant growth models. Ecol Modell 29:107–129

Ritchie JT (1998) Soil water balance and plant water stress. In: Tsuji GY, Hoogenboom G, Thornton PK (eds) Understanding options for agricultural production. Kluwer, Dordrecht, pp 41–54

Sommer R, Wall PC, Govaerts B (2007) Model-based assessment of maize cropping under conventional and conservation agriculture in highland Mexico. Soil Till Res 94:83–100

van Dijk AIJM, Bruijnzeel LA, Rosewell CJ (2002) Rainfall intensity-kinetic energy relationships: a critical literature appraisal. J Hydrol 261:1–23

Porter CH, Jones JW, Gijsman AJ, Adiku S, Dzotsi K, Gargiulo O, Naab JB (2009) Modeling organic carbon in DSSAT v4.5. ORIJ. doi:10.1007/s12351-009-0059-1

Volume4
Chapter 11

Soil-Plant-Atmosphere Module (SPAM)

SOIL PLANT ATMOSPHERE MODULE

January 29, 2004

C.H. Porter, J.W. Jones, G. Hoogenboom, P. W. Wilkens,
J.T. Ritchie, N.B. Pickering, K. Boote, B. Baer

Agricultural and Biological Engineering Department
Research Report No 2000-1202
University of Florida, Gainesville, Florida

SOIL-PLANT-ATMOSPHERE MODULE
TABLE OF CONTENTS

1.0 Soil-Plant-Atmosphere Module Overview 3

2.0 Module Structure 6

3.0 SPAM Module 7
3.1 Run Initialization 7
3.2 Seasonal Initialization 7
3.3 Rate Calculations 9
3.4 Integration 11
3.5 Output 11
3.6 Final 11

4.0 Soil Temperature (STEMP) 13
4.1 Run Initialization ..... **Error! Bookmark not defined.**

5.0 ROOTWU - Root Water Uptake Rate 15
5.1 Run Initialization 15
5.2 Rate Calculations 16

6.0 Potential Evapotranspiration Rate Calculations 17
6.1 PETPEN – FAO-56 Penman Evapotranspiration Method 17
6.2 PETPT – Priestly-Taylor Evapotranspiration Method 17
6.2 PETPT – Priestly-Taylor Evapotranspiration Method **Error! Bookmark not defined.**
6.3 PETDYN – Dynamic 19

7.0 Potential Soil Evaporation Rate (PSE) 20

8.0 SOILEV - Soil Evaporation Rate 21
8.1 Run Initialization 21
8.2 Seasonal Initialization 21
8.3 Rate Calculations 22

9.0 Flood Pool Evaporation Rate ..... Error! Bookmark not defined.

10.0 TRANS - Potential Transpiration Rate 25
10.1 Run Initialization 25
10.2 Rate Calculations 25

11.0 XTRACT – Root Water Extraction 26

12.0 OPSPAM – Output routine 27

20.0 References 28

LIST OF FIGURES

Figure 1 – Components of Soil-Plant-Atmosphere Interface Module .**Error! Bookmark not defined.**
Figure 2 - Calling Sequence for SPAM 'Rate Calculations' Section 12
Figure 3 – Flowchart for SOILEV Rate Calculations..... 23

LIST OF TABLES

Table 1 - Routines in SPAM Module 6
Table 2 – Interface Variables for the Soil-Plant-Atmosphere Module 8
Table 3 – Input and Output Data Requirements for ROOTWU 15
Table 4 - Input and Output Data Requirements for PETPEN..... 18
Table 5 - Input and Output Data Requirements for PETPT..... 17
Table 6 - Input and Output Data Requirements for PSE Module 20
Table 7 – Input and Output Data Requirements for SOILEV 21
Table 8 – Input and Output Data Requirements for TRANS..... 25
Table 9 – Input and Output Data Requirements for XTRACT..... 26

1.0 Soil-Plant-Atmosphere Module Overview

The Soil-Plant-Atmosphere Module (SPAM) computes processes rates for the energy balance components at the interfaces of the plant, the soil and the atmosphere. This module brings together soil, plant and atmosphere inputs and computes light interception by the canopy, potential evapotranspiration as well as actual soil evaporation and plant transpiration. It also computes the amount of water extracted by plant roots from each soil layer. The daily weather values as well as all soil properties and current soil water content, by layer, are required as input. In addition, leaf area index and root length density for each layer are needed.

An option exists to perform an hourly integrated photosynthesis / evapotranspiration computation using a detailed energy balance method. This hourly energy balance routine, ETPHOT, is documented separately, although it is referred to extensively herein, as it is called from the Soil-Plant-Atmosphere Module.

Daily soil temperatures, root water uptake and evapotranspiration variables are exported from the SPAM routine to the daily water balance module (WATBAL), the soil nitrogen balance modules (NTRANS or Century), and the plant growth routines.

4.6 Soil plant atmosphere interface module

This module computes daily soil evaporation and plant transpiration. The current version was originally developed by Ritchie (1972) and was used in all of the DSSAT v3.5 crop models as part of the soil water balance. This module brings together soil, plant and atmosphere inputs and computes light interception by the canopy, potential evapotranspiration, as well as actual soil evaporation and plant transpiration (**Error! Reference source not found.**). It also computes the root water uptake of each soil layer. The daily weather values as well as all soil properties and current soil water content, by layer, are required as input. In addition, leaf area index (LAI) and root length density for each layer are needed.

Potential evapotranspiration. The module first computes daily net solar radiation, taking into account the combined soil and plant canopy albedo. It calculates potential evapotranspiration (ET) using one of four current options. The default Priestley-Taylor (1972) method requires only daily solar radiation and temperature, and was described in detail by Ritchie (1972; 1985) and Jones and Ritchie (1991). The Penman-FAO (Doorenbos and Pruit, 1977) method for computing potential evapotranspiration can optionally be used to better account for arid or windy conditions, but weather data files must include wind and humidity data. We have also created options for using the Penman-Monteith (Monteith, 1986) method for daily potential ET calculations and for using hourly energy balance (unpublished).

The potential ET is partitioned into potential soil evaporation and potential plant transpiration.

Soil evaporation. Two methods are available to compute soil evaporation. The original method, as described by Ritchie (1972) is also briefly described in this document. A new soil evaporation model (Ritchie, 2009) is described in a paper which follows this document. This Extended Sulieman-Ritchie model (ESR) is the default model in DSSAT v4.5.

Plant transpiration. To determine whether the soil or atmosphere limits plant transpiration, potential root water uptake is computed by calculating a maximum water flow to roots in each layer and summing these values (Ritchie, 1985, 1998; Jones and Ritchie, 1991). These calculations account for root length density in each layer and the soil water content in the layer. The equation that computes potential root water uptake in each layer is an approximation to the radial flow equation, where assumptions are made about soil texture effect on hydraulic conductivity, root diameter, and a maximum water potential difference between roots and the soil. The actual plant transpiration is then computed as the minimum of potential plant transpiration and the potential root water uptake. Thus, the atmosphere can limit transpiration by low solar radiation and cool temperatures, the canopy can limit it by low LAI, and the soil can limit it by low soil water content, low root length density, and their distributions relative to each other.

This method for computing evapotranspiration has provided an excellent functional approach for determining water stress in the plant without explicitly modeling water status in the plant component. The ratio of actual ET to potential ET, if less than 1.0, indicates that stomatal conductance would have had to be decreased sometimes during the day to prevent plant desiccation. This ratio is typically used in the plant modules to reduce photosynthesis in proportion to relative decreases in transpiration. Similarly, a ratio of potential root water uptake and potential transpiration is used to reduce plant turgor and expansive growth of crops. The rationale for this is that as soil water becomes more limiting, turgor pressure in leaves would decrease and affect leaf expansion before photosynthesis is reduced. In the current plant modules this ratio is set to 1.5.

Hourly hedgerow photosynthesis sub module. An option exists for CROPGRO crops to have photosynthesis computed by the hourly hedgerow photosynthesis light interception approach as described by Boote and Pickering (1994). On an hourly time step during each day, interception and absorption of direct and diffuse light components are computed based upon canopy height and width, leaf area index (LAI), leaf angle, row direction, latitude, day of year, and time of day (Boote and Pickering, 1994). Photosynthesis of sunlit and shaded leaves is computed hourly using the asymptotic exponential response equation, where quantum efficiency and light-saturated photosynthesis rate variables are dependent on CO2 and temperature (Boote and Pickering, 1994). Hourly canopy photosynthesis on a land area basis is computed from the sum of sunlit and shaded leaf contributions by multiplying sunlit and shaded leaf photosynthetic rates by their respective LAIs. Gross photosynthesis is integrated hourly to provide a daily total value for use by other modules.

Currently the CROPGRO model is the only crop growth model of DSSAT-CSM that is linked to the hourly hedgerow photosynthesis model. This routine reads several crop-specific parameters from the species file, which are not currently available for other crops.

Soil Temperature sub module

The soil temperature model is a sub-module of the Soil-Plant-Atmosphere module. It was originally derived from the EPIC model (Williams et al., 1984; Jones et al., 1991) and is the same as the one in the CERES and CROPGRO models in DSSAT v3.5. Soil temperature is computed from air temperature and a deep soil temperature boundary condition that is calculated from the average annual air temperature and the amplitude of monthly mean temperatures. It also includes a simple approach to calculate the impact of solar radiation and albedo on the soil surface temperature. However, it does not consider differences in soil wetness or surface conditions. Soil temperature is used to modify plant processes (emergence) and soil organic matter decomposition. Additional details on this component are in Jones and Kiniry (1986) in the description of the CERES-Maize model.

2.0 Module Structure

The soil-plant-atmosphere interface module is called from the Land Unit module of the Cropping Systems Model. The FORTRAN code is organized in a modular format as described in the chapter on Modular Structure (Porter, Jones, et al., 2000). The module is called a total of six times: once at the beginning of simulation for 'Run Initialization', once at the beginning of each season of simulation for 'Seasonal Initialization', once daily for 'Rate calculations', 'Integration', and 'Output', and at the end of each season for 'Final' calculations.

Table 1 presents the components of the SPAM module. The SPAM routine controls the flow of processing within the module and calls 13 other routines. Many of the routines called by the SPAM module can be considered to be independent modules, as they meet the criteria presented by Porter, Jones, et al. (2000). These routines perform two or more processing functions such as input and initialization of variables, rate calculation, integration of state variables, output of data and final processing.

Table 1 - Routines in SPAM Module

Routine	Phase of Model in which Routine is Called						Description
	RUNINIT	SEASINIT	RATE	INTEGR	OUTPUT	FINAL	
STEMP		X	X		X	X	Calculates soil temperature
ROOTWU		X	X				Calculates root water uptake rate for each soil layer and total rate for profile.
PET			X				Potential evapotranspiration rate
PSE			X				Calculates potential soil evaporation rate.
FLOOD_EVAP			X				
MULCH_EVAP		X	X				Evaporation rate from surface mulch layer
SOILEV		X	X				Calculates actual soil evaporation (Ritchie 1972 method).
ESR_SoilEvap			X				Actual soil evaporation (Extended Sulieman-Ritchie method)
TRANS		X	X				Calculates actual transpiration rate. (Includes subroutines TRANS, TRATIO and BLRRES).
ETPHOT	X	X	X		X	X	
XTRACT			X				Calculates root extraction of water for each soil layer.
OPSPAM		X		X	X	X	Writes daily output to file OUTW (e.g., WATER.OUT)

3.0 SPAM Module

The main code of the SPAM module is used to control the flow of processing to the various routines within the module. Few calculations are done within this routine, but rather, other routines are called to initialize variables, perform calculations, and output data.

Table 2 presents input and output data required by the SPAM module. The input data include:

- Daily weather data (CLOUDS, CO2, DAYL, RAIN, SRAD, TA, TAMP, TAV, TAVG, TDEW, TMAX, TMIN, WINDSP, and XLAT),
- Plant species parameters (EORATIO, KSEVAP, KTRANS, PORMIN, and RWUMX),
- Plant growth variables (CANHT, RLV, and XHLAI),
- Soil properties (SOILPROP),
- Soil water content and water fluxes (SW, SWDELTS, SWDELTU, and WINF), and
- Simulation information (CONTROL and ISWITCH).

Three interface variables are both input and output to the SPAM routine. FLOODWAT is a composite variable, meaning that it is comprised of various component variables related to flood water quantity. The structure and content of the FLOODWAT is defined in Appendix A. One of the components of the composite variable, EF, is the evaporation rate from the floodwater surface. If a flooded field is being simulated, the SPAM module modifies this component.

Additionally, two variables, which are normally computed in the weather module, TGRO and TGROAV, the daily and hourly canopy temperatures, are re-computed in ETPHOT if the hourly energy balance photosynthesis and/or evapotranspiration methods are selected. Therefore, these are also listed as input/output variables.

Output data include:

- Daily soil temperatures (ST and SRFTEMP) and
- Daily evapotranspiration rates (EO, EOP, ES, SWDELTX, TRWUP).

3.1 Run Initialization

In the 'Run Initialization' section of the code, ETPHOT is called if the hourly energy balance method is chosen to read input data for that routine.

3.2 Seasonal Initialization

In the 'Seasonal Initialization' portion of the SPAM code, calls are made to subroutines STEMP, ROOTWU, SOILEV, TRANS, and ETPHOT for initialization of variables that are local to those routines. Subroutine OPSPAM is called to write output file headings to file ET.OUT. Daily and cumulative evapotranspiration variables are initialized to zero.

Table 2 – Interface Variables for the Soil-Plant-Atmosphere Module

Variable Name	Definition	Units
Input Variables		
CANHT	Canopy height	m
CLOUDS	Relative cloudiness factor (0-1)	--
CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
EORATIO	Ratio of increase in potential evapotranspiration with increase in LAI (up to LAI=6.0) for use with FAO-56 Penman reference potential evapotranspiration.	--
ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
KSEVAP	Light extinction coefficient used for computation of soil evaporation	--
KTRANS	Light extinction coefficient used for computation of plant transpiration	--
PORMIN	Minimum pore space required for supplying oxygen to roots for optimal growth and function	cm3/cm3
RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
RWUMX	Maximum water uptake per unit root length, constrained by soil water	cm3[water] / cm [root]
SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
SRAD	Solar radiation	MJ/m2-d
SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
SWDELTS(L)	Change in soil water content due to drainage in layer L	cm3 [water] / cm3 [soil]
SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L	cm3 [water] / cm3 [soil]
TA	Daily normal temperature	°C
TAMP	Amplitude of temperature function used to calculate soil temperatures	°C
TAV	Average annual soil temperature, used with TAMP to calculate soil temperature.	°C
TAVG	Average daily temperature	°C
TDEW	Dew point temperature	°C
TMAX	Maximum daily temperature	°C
TMIN	Minimum daily temperature	°C
WINDSP	Wind speed at 2m	km/d
WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d
XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
XLAT	Latitude	deg.
Input/Output Variables		
FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
TGRO(I)	Hourly canopy temperature	°C
TGROAV	Average daily canopy temperature	°C
Output Variables		
EO	Potential evapotranspiration rate	mm/d
EOP	Potential plant transpiration rate	mm/d
ES	Actual soil evaporation rate	mm/d
SRFTEMP	Temperature of soil surface litter	°C
ST(L)	Soil temperature in soil layer L	°C
SWDELTX(L)	Change in soil water content due to root water uptake in layer L	cm3 [water] / cm3 [soil]
TRWUP	Potential daily root water uptake over soil profile	cm/d

3.3 Rate Calculations

Most of the calculations for the SPAM module are performed within the 'Rate Calculations' section. All process rate variables are computed based on today's weather input and yesterday's values of state variables. Figure 1 illustrates the sequence of processing rate calculations for the soil-plant-atmosphere module and some of the main input and output variables which are passed between the various routines. If the hourly energy balance method is used, many of the other routines in the SPAM module are bypassed.

The model first computes the soil temperature within each layer of the soil profile. This methodology is discussed in Section 4.0 Soil Temperature Module (STEMP).

The potential root water uptake is computed by routine ROOTWU, a

It calculates potential evapotranspiration (ET) using one of four current options. The default Priestley-Taylor (1972) method requires only daily solar radiation and temperature, and was described in detail by Ritchie (1972; 1985) and Jones and Ritchie (1991). The Penman-FAO (Doorenbos and Pruit, 1977) method for computing potential evapotranspiration can optionally be used to better account for arid or windy conditions, but weather data files must include wind and humidity data. We have also created options for using the Penman-Monteith (Monteith, 1986) method for daily potential ET calculations and for using hourly energy balance (unpublished).

The potential ET is partitioned into potential soil evaporation based on the fraction of solar energy reaching the soil surface, based on a negative exponential function of LAI, and potential plant transpiration. Actual soil evaporation is based on a two-stage process (Ritchie, 1972). After the soil surface is first wetted due to either rainfall or irrigation, evaporation occurs at the potential rate until a cumulative soil evaporation amount since wetting is reached. Then, a soil-limiting daily soil evaporation amount is computed as a square root function of time since stage one ended. Actual soil evaporation is the minimum of the potential and soil-limiting calculations on a daily basis. If evaporation is less than potential soil evaporation, this difference is added back to potential plant transpiration to account for the increased heat load on the canopy when the soil surface is dry (Ritchie, 1972).

To determine whether the soil or atmosphere limits plant transpiration, potential root water uptake is computed by calculating a maximum water flow to roots in each layer and summing these values (Ritchie, 1985, 1998; Jones and Ritchie, 1991). These calculations account for root length density in each layer and the soil water content in the layer. The equation that computes potential root water uptake in each layer is an approximation to the radial flow equation, where assumptions are made about soil texture effect on hydraulic conductivity, root diameter, and a maximum water potential difference between roots and the soil. The actual plant transpiration is then computed as the minimum of potential plant transpiration and the potential root water uptake. Thus, the atmosphere can limit transpiration by low solar radiation and cool temperatures, the canopy can limit it by low LAI, and the soil can limit it by low soil water content, low root length density, and their distributions relative to each other.

This method for computing evapotranspiration has provided an excellent functional approach for determining water stress in the plant without explicitly modeling water status in the plant component. The ratio of actual ET to potential ET, if less than 1.0, indicates that stomatal conductance would have had to be decreased sometimes during the day to prevent plant desiccation. This ratio is typically used in the plant modules to reduce photosynthesis in proportion to relative decreases in transpiration. Similarly, a ratio of potential root water uptake and potential transpiration is used to reduce plant

turgor and expansive growth of crops. The rationale for this is that as soil water becomes more limiting, turgor pressure in leaves would decrease and affect leaf expansion before photosynthesis is reduced. In the current plant modules this ratio is set to 1.5.

The soil temperature routine is called first to update soil temperatures based on today's weather parameters. These soil temperature values are then available throughout the model for use in today's rate calculations.

A pseudo-integration is done to compute the amount of soil water that is currently available including any drainage that was calculated in the WATBAL routine, but not yet incorporated into the soil water content variable. This prevents the rates of evapotranspiration computed in the soil-plant-atmosphere routines to cause soil water content to fall above or below acceptable values.

Potential root water uptake is computed in ROOTWU if leaf area exists.

Five methods of evapotranspiration are available for use in the Soil-Plant-Atmosphere Module. If a field is currently flooded, the Priestly-Taylor method is automatically used with an assumed surface albedo of 0.08. Otherwise, the method is determined by variable MEEVP, which is a user-specified input in FILEX. Table 3 summarizes the evapotranspiration methods available in the model.

Table 3 – Available methods of computing potential evapotranspiration

MEEVP	Subroutine	Method
R	PETPT	Priestly-Taylor (default)
F	PETPEN	Penman-Monteith (FAO-56)
D	PETDYN	Penman-Monteith with dynamic input of LAI and crop height
P	PETPNO	Penman (FAO-24)
Z	ETPHOT	Hourly energy balance method

The first four potential evapotranspiration methods are discussed in this document. ETPHOT, the hourly energy balance solution is documented in a separate chapter. The ETPHOT option can be used to compute photosynthesis only (MEPHO = 'L') or to compute both photosynthesis and potential evapotranspiration (MEPHO = 'L' and MEEVP = 'Z'). These options are not available for all crops as the method requires some specific parameters to be read from the species file.

After computing potential evapotranspiration, subroutine PSE is called to compute potential soil evaporation (EOS). For flooded fields, the flood pool evaporation is calculated in FLOOD_EVAP. Otherwise, actual soil evaporation is computed by SOILEV. After plant emergence, TRANS is called to compute actual plant transpiration (EOP).

When the hourly energy balance method is selected, calls to ROOTWU, PSE, FLOOD_EVAP, SOILEV, and TRANS are bypassed as all of these functions are computed in ETPHOT.

The XTRACT routine is called to calculate actual soil water uptake rates (RWU and TRWU) and the actual plant transpiration rate (EP) and changes to the soil water content in each layer (SWDELTX).

3.4 Integration

The 'Integration' portion of the SPAM module updates the cumulative evapotranspiration variables and calls OPSPAM to compute composite values.

3.5 Output

In the 'Output' section, both OPSPAM and STEMP are called to print to daily output files (ET.OUT and SoilTemp.OUT). If the user selected the leaf level photosynthesis option (MEPHO = 'L'), ETPHOT is called to generate output. The GetPutSPAM subroutine is used to send water balance variables to the daily soil water balance routine so that a mass balance can be done both daily and seasonally.

3.6 Final

The final section of the SPAM code calls OPSPAM, STEMP and ETPHOT to close output files at the end of each season of simulation.

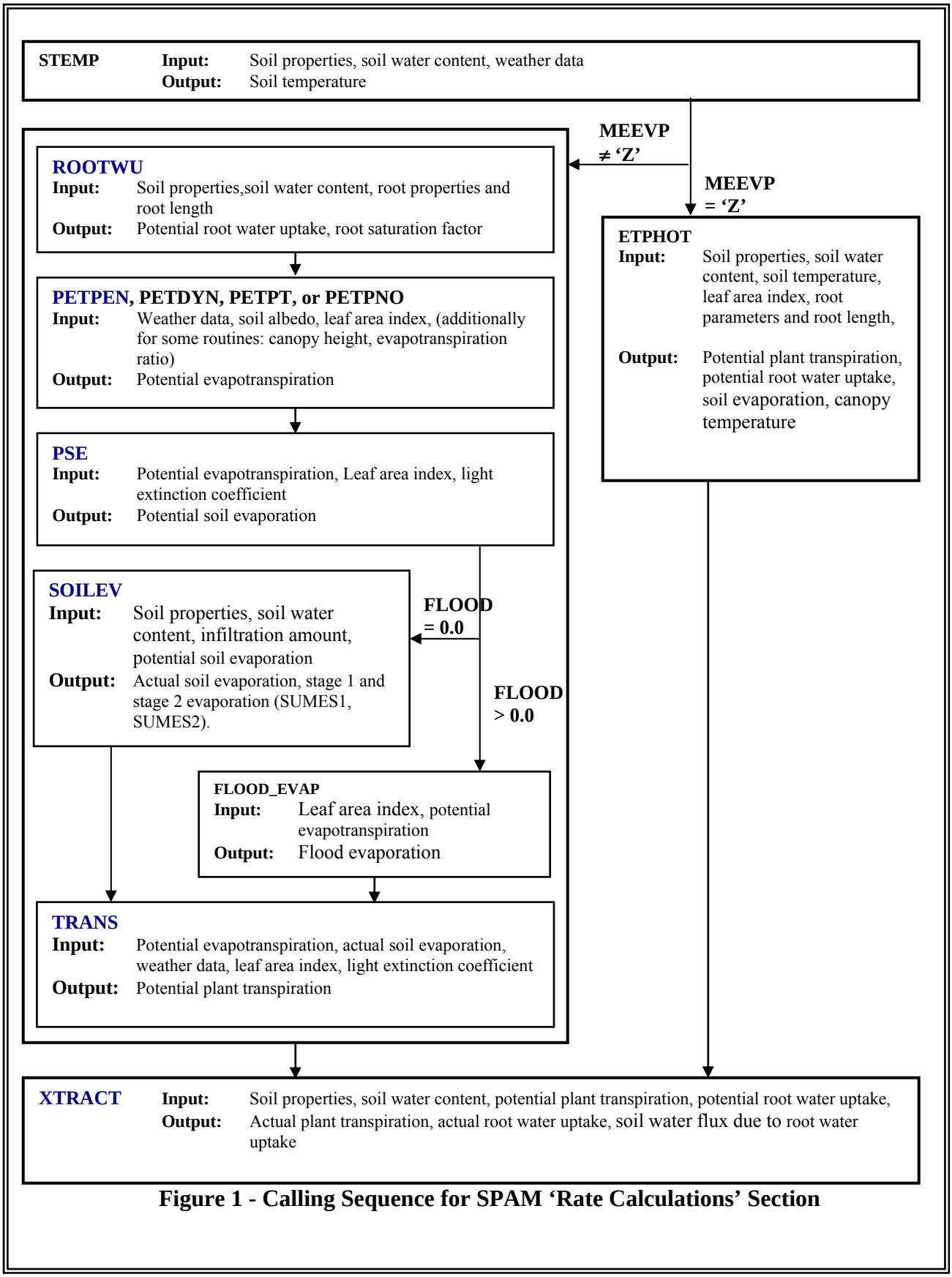


Figure 1 - Calling Sequence for SPAM 'Rate Calculations' Section

4.0 Soil Temperature Module (STEMP)

The soil temperature model currently in the DSSAT cropping system model was originally derived from the EPIC model (Williams et al., 1984; Jones et al., 1991) and is the same as the one in the CERES and CROPGRO models in DSSAT v3.5. Soil temperature is computed from air temperature and a deep soil temperature boundary condition that is calculated from the average annual air temperature and the amplitude of monthly mean temperatures. It also includes a simple approach to calculate the impact of solar radiation and albedo on the soil surface temperature. However, it does not consider differences in soil wetness or surface conditions. Soil temperature is used to modify plant processes (emergence) and soil organic matter decomposition. Additional details on this component are in Jones and Kiniry (1986) in the description of the CERES-Maize model.

The soil temperature routine is contained in file STEMP.FOR. This is technically, a separate module, but a very simple one and so separate documentation was not warranted. Subroutine STEMP is called from the main program for 'Run Initialization', 'Seasonal Initialization' and 'Rate Calculations'. Table 4 lists the calling arguments which are passed to and from the STEMP routine.

Table 4 – Calling arguments for STEMP

Variable Name	Definition	Units
Input Variables		
BD(L)	Bulk density, soil layer L	g [soil] / cm3 [soil]
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
MESIC	Code for sequenced runs: 'S'=second or subsequent run in sequenced simulation, 'M'=single run or first run of sequenced simulation.	--
PESW	Potential extractable soil water (= SW - LL) summed over root depth	cm
SRAD	Solar radiation	MJ/m2-d
TAMP	Amplitude of temperature function used to calculate soil temperatures	°C
TAV	Average annual soil temperature, used with TAMP to calculate soil temperature.	°C
TAVG	Average daily temperature	°C
TMAX	Maximum daily temperature	°C
XLAT	Latitude	deg.
YRDOY	Current day of simulation	YYDDD
Output Variables		
ATOT	Sum of TMA array (last 5 days soil temperature)	°C
SRFTEMP	Temperature of soil surface litter	°C
ST(L)	Soil temperature in soil layer L	°C
TMA(I)	Array of previous 5 days of average soil temperatures.	°C

Run Initialization

Soil albedo (SALB) and soil depth information DS(L) are read from FILEIO.

Seasonal Initialization

Values of soil temperature equation coefficients are initialized based on soil bulk density (BD). Unless this is a sequenced run, subroutine SOILT is called seven times to obtain a stable, average soil temperature profile.

Rate Calculations
Subroutine SOILT is called to obtain soil temperature profile.

SOILT
The previous five days of average soil temperature are stored in array TMA. At the beginning of the procedure, the last value is dropped, and a new value calculated for today based on soil albedo, current temperatures and solar radiation. The average soil temperature is then distributed to soil layers.

5.0 ROOTWU - Root Water Uptake Rate

This module calculates the daily root water uptake rate for each soil layer and the total root water uptake rate for the soil profile. It is called twice from the main SPAM module for 'Run Initialization' and 'Rate Calculations'. Appendix I lists the program code for the root water uptake module. Table 5 lists input and output variables.

Table 5 – Input and Output Data Requirements for ROOTWU

Input Variables:		
CROP	Crop identification code	--
DLAYR(L)	Soil Depth in layer L	cm
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
LL(L)	Volumetric soil water content in soil layer L at lower limit	cm3/cm3
NLAYR	Actual number of soil layers	--
RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
SAT(L)	Volumetric soil water content in layer L at saturation	cm3[water]/cm3[soil]
SW(L)	Volumetric soil water content in layer L	cm3[water]/cm3[soil]
Output Variables:		
RWU(L)	Root water uptake from soil layer L	cm/d
SATFAC	Root length weighted soil water excess stress factor (0 = no stress; 1 = saturated stress)	--
TRWU	Total potential daily root water uptake	cm/d

The rate of root water uptake from a soil layer is calculated using a "law of the limiting" approach in which the larger of the soil or the root resistance determines the flow rate of water into roots. Also, TWRU (total root water uptake) is a potential rate, defined as the maximum root water extraction under high evaporative demand days. At the end of the day actual soil water extraction is limited by either TRWU or by potential transpiration of the crops. To compute potential root water uptake, the model assumes radial flow to single roots (Ritchie, 1985).

5.1 Run Initialization

The 'Run Initialization' section of the ROOTWU code is used to read the following data from FILEC (e.g., SBGRO980.SPE):

- RWUMX - maximum rate that water can be extracted by roots, and
- PORMIN, the minimum pore space required for supplying oxygen to roots for optimal growth and function.

FILEIO is opened solely to obtain the name of FILEC for reading these parameters.

Next, values of SWCON2(L) are computed for each soil layer:

```
DO L = 1,NLAYR
  SWCON2(L) = 120. - 250. * LL(L)
  IF (LL(L) .GT. 0.30) SWCON2(L) = 45.0
ENDDO
```

Where SWCON2(L) is a texture dependent coefficient in the radial root water uptake equation. Two additional coefficients, SWCON1 and SWCON3 are also initialized here.

5.2 Rate Calculations

If a soil layer, L, contains roots (RLV(L) > 0.00001) and if sufficient soil water is present (SW(L) > LL(L)), then root water uptake in that layer is calculated using the radial root water uptake equation.

```
  RWU(L) = SWCON1*EXP(MIN((SWCON2(L)*(SW(L)-LL(L))),40.))/
&      (SWCON3-ALOG(RLV(L)))
```

where RWU(L) = rate of root water uptake from layer L (cm^3 cm^{-1} day^{-1}),
SW(L) = soil water content in soil layer L (cm3 cm-3),
LL(L) = soil water content at the lower limit in soil layer L (cm3 cm-3),
40. = upper limit of the rate of flow that can occur in a root (cm3 cm-1),
RLV(L) = root length density volume (cm cm-3).

The maximum daily root water uptake per unit root length (RWUMX) is adjusted by the SWEXF if the soil is saturated. Then, root water uptake for each layer (RWU(L)) is calculated, and the total root water uptake (TRWU) is calculated by adding RWU(L) for each soil layer.

```
RWU(L) = RWU(L) * DLAYR(L) * RLV(L)
TRWU = TRWU + RWU(L)
```

A root length weighted soil water excess stress factor (SATFAC) is computed for the whole soil profile by dividing the sum over all layers of water excess factor times depth times root length density (SUMEX) by the sum of root length density (SUMRL). Whereas excess water affects different processes differently (e.g., root depth growth, root proliferation, and root water uptake), this factor provides a summary indicator of excess soil water impact.

6.0 Potential Evapotranspiration Rate Calculations

The DSSAT models currently contain two stand-alone evapotranspiration models. The Priestley-Taylor method (Priestley and Taylor, 1972), also described by Ritchie (1985) requires temperature and solar radiation to compute the equilibrium evapotranspiration equation. The FAO version of the Penman evapotranspiration equation (Doorenbos, et al. 1977 and Jones, et al. 1984) additionally requires wind speed and humidity data.

Either one or the other of the potential evapotranspiration methods is called (unless MEEVP = 'Z' as discussed in Section 3.3). These routines are called from the SPAM main module: once for 'Run Initialization' and a second time for 'Rate Calculations'. Appendix F presents a code listing of both routines.

6.1 PETPT – Priestly-Taylor Evapotranspiration Method

The input and output arguments for the Priestley-Taylor evapotranspiration routine are listed in Table 6.

Table 6 - Input and Output Data Requirements for PETPT

Input Variables:		
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
SRAD	Solar radiation	MJ/m2-d
TMAX	Maximum daily temperature	°C
TMIN	Minimum daily temperature	°C
XHLAI	Leaf area index	m2[leaf] / m2[ground]
Output Variables:		
EO	Potential evapotranspiration rate	mm/d

6.1.1 Run Initialization

In the 'Run Initialization' section of the code, SALB, the bare soil albedo is read from FILEIO. Weather variables and the leaf area index, XHLAI, are passed into the routine as arguments.

6.1.2 Rate Calculations

The average daily temperature, TD, is calculated from the maximum and minimum daily temperatures. As in the Penman method, the integrated crop/soil albedo (ALBEDO) is computed with a formula which weights bare soil albedo (SALB) with leaf albedo based on the leaf area index (XHLAI).

```
ALBEDO = 0.23 - (0.23 - SALB) * EXP(-0.75 * XHLAI)
```

The equilibrium evaporation rate (EEQ) is then calculated from solar radiation (SLANG), albedo (ALBEDO) and average daily temperature (TD):

```
EEQ = SLANG * (2.04E-4-1.83E-4 * ALBEDO) * (TD + 29.0)
```

Potential evapotranspiration (EO) is estimated as a function of maximum temperature. When maximum daily temperature, TMAX, is between 5°C and 35°C:

```
EO = EEQ * 1.1
```

If TMAX is greater than 35°C:

```
EO = EEQ * ((TMAX - 35.0) * 0.05 + 1.1)
```

And if TMAX is less than 5°C:

```
EO = EEQ * 0.01 * EXP(0.18 * (TMAX + 20.0))
```

These equations increase the coefficient relating EEQ to EO to 1.35 when TMAX is 40°C and they reduce the coefficient to 0.5 when TMAX is 2.0°C.

6.2 PETPEN – FAO-56 Penman Evapotranspiration Method

The input and output arguments for the FAO-Penman evapotranspiration routine are listed in Table 7.

Table 7 - Input and Output Data Requirements for PETPEN

Input Variables:		
CLOUDS	Relative cloudiness factor (0-1)	--
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
SRAD	Solar radiation	MJ/m2-d
TAVG	Average daily temperature	°C
TDEW	Dewpoint temperature	°C
TMAX	Maximum daily temperature	°C
TMIN	Minimum daily temperature	°C
WINDHT	Reference height for wind speed	m
WINDSP	Wind speed	km/d
XHLAI	Leaf area index	m2[leaf] / m2[ground]
Output Variables:		
EO	Potential evapotranspiration rate	mm/d

6.2.1 Run Initialization

In the 'Run Initialization' section of the code, SALB, the bare soil albedo is read from FILEIO.

6.2.2 Rate Calculations

The Penman equation (Penman, 1956) expresses potential evapotranspiration as a function of net radiation (RNET), atmospheric vapor pressure deficit (VPD) temperature (TMAX, TMIN, TAVG) and wind speed (WINDSP). First, the latent heat of vaporization (LHVAP), psychrometric constant (PSYCON), and vapor pressure deficit (VPD) are computed as functions of temperature:

```
LHVAP = (2501.0 - 2.373 * TAVG) * 1000.0
PSYCON = SHAIR * PATM / (0.622 * LHVAP)
ESAT = (VPSAT(TMAX) + VPSAT(TMIN)) / 2.0
EAIR = VPSAT(TDEW)
VPD = ESAT - EAIR
```

ESAT and EAIR are the average vapor pressure of air and the vapor pressure of air at dewpoint temperature, respectively. The rate of change of saturated vapor pressure of air with temperature, S, is computed using the function subroutine VPSLOP found in file HMET.FOR. Wind speed, WIND2, is calculated based on a 2 meter reference height.
The soil heat flux density term, G, is assumed to be zero for the 24 hour average calculations, because it is assumed that energy absorbed by the soil during the day is, for the most part, lost at night.

The integrated crop/soil albedo (ALBEDO) is computed with a formula which weights bare soil albedo (SALB) with leaf albedo based on the leaf area index (XHLAI):

```
IF (XHLAI .LE. 0.0) THEN
   ALBEDO = SALB
ELSE
   ALBEDO = 0.23  - (0.23 - SALB) * EXP(-0.75*XHLAI)
ENDIF
```

The net outgoing thermal radiation, RADB, is calculated based on the Stefan-Bolzmann constant, SBZCON, temperature (°K), the dewpoint vapor pressure (EAIR), and a cloudiness factor (CLOUDS). Net radiant energy exchanged at the earth's surface (RNET) is computed as a function of albedo, solar radiation (SRAD) and net thermal radiation, RADB:

```
RADB = SBZCON * TK4 * (0.4 - 0.005 * SQRT(EAIR)) *
  &     (1.1 * (1. - CLOUDS) - 0.1)
RNET = (1.0 - ALBEDO) * SRAD - RADB
```

The FAO wind function represents an empirical relationship that accounts for the effects of wind on potential evapotranspiration.

```
WFNFAO = 0.0027 * (1.0 + 0.01 * WIND2)
```

The energy (radiation) and aerodynamic (wind and humidity) terms are combined in the Penman equation to compute the potential evapotranspiration, EO:

```
RNETMG = (RNET - G) / LHVAP * 1.0E6
EO = (S * RNETMG + PSYCON * WFNFAO * VPD)/(S + PSYCON)
```

6.3 PETPNO FAO-23

6.4 PETDYN – Dynamic

7.0 Potential Soil Evaporation Rate (PSE)

Subroutine PSE, which computes potential soil evaporation (EOS), is called once from the main SPAM module for 'Rate Calculations'. Table 8 lists the calling arguments for the routine. Appendix G lists the FORTRAN code.

Table 8 - Input and Output Data Requirements for PSE Module

Input Variables:		
EO	Potential evapotranspiration rate	mm/d
XHLAI	Leaf area index	m2[leaf] / m2[ground]
Output Variables:		
EOS	Potential rate of soil evaporation	mm/d

When the leaf area index, (XHLAI) is less than 1.0, the following equation is used:

```
EOS = EO * (1.0 - 0.43 * XHLAI)
```

If XHLAI is greater than 1.0,

```
EOS = EO / 1.1 * EXP(-0.4 * XHLAI)
```

8.0 SOILEV - Soil Evaporation Rate

This module computes soil evaporation using a method described by Ritchie (1972). Evaporation takes place in two stages: the constant and the falling rate stages. In the constant rate stage (stage 1), the soil is sufficiently wet for the water to be transported to the surface at a rate at least equal to the evaporation potential. In the falling rate stage (stage 2), the surface soil water content has decreased below a threshold value, so that soil evaporation (ES) depends on the flux of water through the upper layer of soil to the evaporating site near the surface. For a thoroughly wetted soil, stage 1 evaporation proceeds until a user-specified amount of water has evaporated (variable U). Soil evaporation then enters stage 2, in which ES is a declining function of time since the beginning of stage 2 (variable T). The variables SUMES1 and SUMES2 are the sums of evaporation in stages 1 and 2, respectively, and they are used to determine which stage of soil evaporation is occurring on a day. (Jones, C.A., et al. 1986.)

The SOILEV module is called three times from the SPAM main module for 'Run Initialization', 'Seasonal Initialization' and 'Rate Calculations'. A listing of the SOILEV program code is contained in Appendix K. Table 9 lists the calling arguments for the SOILEV module.

Table 9 – Input and Output Data Requirements for SOILEV

Input Variables:		
DLAYR(L)	Soil Depth in layer L	cm
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L	cm3[water]/cm3[soil]
EOS	Potential rate of soil evaporation	mm/d
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
LL(L)	Volumetric soil water content in soil layer L at lower limit	cm3/cm3
SW(L)	Volumetric soil water content in layer L	cm3[water]/cm3[soil]
SW_AVAIL(L)	Soil water content in layer L available for evaporation, plant extraction, or movement through soil	cm3[water]/cm3[soil]
WINF	Potential Precipitation for infiltration	mm
Output Variables:		
ES	Actual soil evaporation rate	mm/d
SUMES1	Cumulative soil evaporation in stage 1	mm
SUMES2	Cumulative soil evaporation in stage 2	mm
SWEF	Soil water evaporation fraction; fraction of lower limit content to which evaporation can reduce soil water content in top layer	fraction
T	Number of days into Stage 2 evaporation (SPAM); or time factor for hourly temperature calculations	--

8.1 Run Initialization

The SOILEV 'Run Initialization' section reads the value of U, Stage 1 evaporation limit, from FILEIO. The value is limited to a maximum of 6.0 cm.

8.2 Seasonal Initialization

Based on initial soil water content in the upper layer of soil, the program determines whether initial evaporation rates should be calculated using stage 1 or stage 2 equations. The soil water ratio, SWR is defined as the fraction of water in the first soil layer between the lower

limit and the drained upper limit. Variable USOIL represents a deficit of water in the first layer of soil below the drained upper limit.

```
SWR = MAX(0.0,(SW(1) - LL(1)) / (DUL(1) - LL(1)))
USOIL = (DUL(1) - SW(1)) * DLAYR(1) * 10.
```

If the initial soil water content is greater than the drained upper limit (SWR≥1.0), then accumulations of stage 1 and stage 2 evaporation are set to zero and initial soil evaporation will be computed as stage 1 (T = 0.).

If the value of USOIL is less than U, the limit for stage 1 evaporation, then stage 1 evaporation occurs, with a stage 1 accumulation of USOIL:

```
SUMES2 = 0.
T = 0.
SUMES1 = USOIL
```

If the initial soil water content is low enough (less than U below the drained upper limit), the initial evaporation rates are calculated as stage 2:

```
SUMES2 = USOIL - U
SUMES1 = U
T = (SUMES2/3.5)**2
```

Finally, in the 'Seasonal Initialization' section, SWEF is calculated to be the soil water evaporation fraction, or the fraction of the lower limit of water content to which evaporation can reduce soil water. The initial value of the soil water content in the upper layer of soil is limited by this value.

8.3 Rate Calculations

Figure 2 presents a flowchart of the rate calculations in the SOILEV module. The circled numbers in the text below refer to locations on the flowchart.

Stage 2 evaporation is computed if SUMES1, the stage 1 evaporation accumulation, is greater than or equal to U, the stage 1 evaporation upper limit. If rainfall or irrigation has occurred, and is greater than the stage 2 evaporation accumulation, then stage 1 rates are calculated.

If rainfall and/or irrigation occur on a day and the water available for infiltration (WINF) is greater than or equal to the stage 1 evaporation accumulation (SUMES1), then SUMES1 is set equal to zero. If WINF is less than SUMES1, SUMES1 is updated as follows:

```
SUMES1 = SUMES1 – WINF
```

(1)

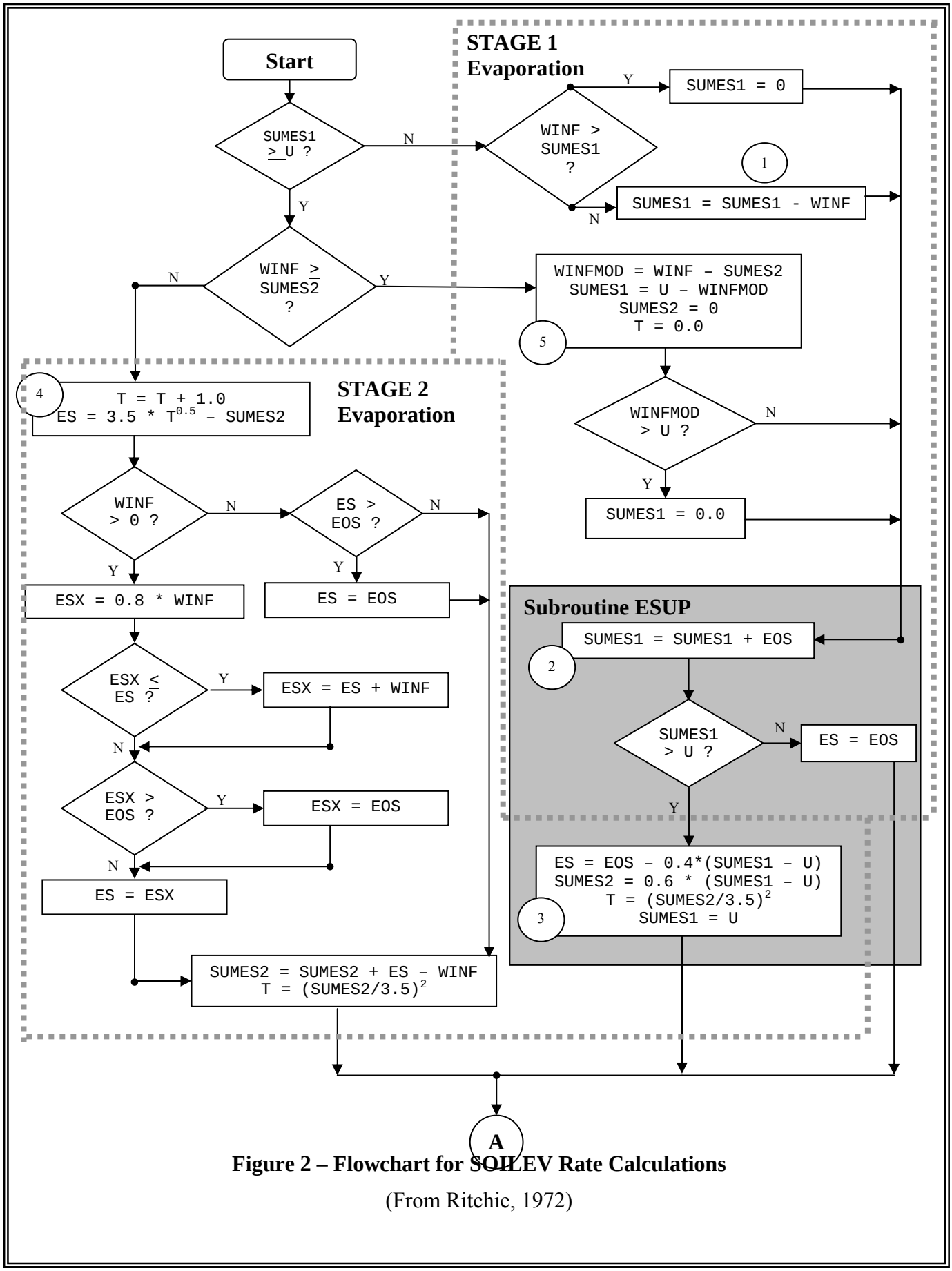


Figure 2 – Flowchart for SOILEV Rate Calculations

(From Ritchie, 1972)

Whenever SUMES1 is less than U, the stage 1 evaporation limit, SUMES1 is updated daily according to the following equation in subroutine ESUP:

```
SUMES1 = SUMES1 + EOS
```
(2)

If the new value of SUMES1 is less than or equal to U, actual soil evaporation (ES) is set equal to potential soil evaporation (EOS).

If the new value of SUMES1 exceeds U, then stage 2 evaporation is computed. Actual soil evaporation (ES), stage 2 evaporation accumulation (SUMES2) and the time that has elapsed since the start of stage 2 evaporation (T) are calculated as:

```
ES = EOS - 0.4 * (SUMES1 - U)
SUMES2 = 0.6 * (SUMES1 - U)
T = (SUMES2/3.5)**2
```
(3)

During stage 2 evaporation, as the soil continues to dry, T increases by 1.0 each day, and ES is calculated as follows:

```
ES = 3.5 * T**0.5 - SUMES2
```
(4)

with ES limited to no greater than the potential soil evaporation rate (EOS).

If rainfall and/or irrigation wet the soil surface slightly, but WINF is less than SUMES2, ES is the minimum of EOS, (0.8 * WINF) or (ES + WINF). If, during stage 2 evaporation, rainfall and/or irrigation wet the soil surface slightly more and WINF exceeds SUMES2, the model resets T to zero and recalculates WINF and SUMES1:

```
WINF = WINF - SUMES2
SUMES1 = U - WINF
```
(5)

SUMES2 and T are reset to zero. The model then calculates stage 1 evaporation as described earlier.

The Ritchie model was modified for use in the SPAM model by the addition of an algorithm that compares the current extractable soil water in the top layer, AWEV1, to the computed soil evaporation. Stage 1 and stage 2 evaporation rates are adjusted if the water available for evaporation is less than the calculated daily soil evaporation rate. This adjustment is done after point 'A' in the flowchart in Figure 7.

10.0 TRANS - Potential Transpiration Rate

This module calculates the daily potential transpiration rate, EOP, in mm/d. Appendix M lists the FORTRAN program code for TRANS. Table 10 lists input and output variables. This routine is called twice from SPAM for 'Run Initialization' and 'Rate Calculations'.

Table 10 – Input and Output Data Requirements for TRANS

Input Variables:		
CO2	Atmospheric carbon dioxide concentration	ppm
CROP	Crop identification code	--
EO	Potential evapotranspiration rate	mm/d
ES	Actual soil evaporation rate	mm/d
FILEIO	Filename for input file (e.g., IBSNAT35.INP)	--
TAVG	Average daily temperature	°C
WINDSP	Wind speed	km/d
XHLAI	Leaf area index	m2[leaf] / m2[ground]
Output Variables:		
EOP	Potential plant transpiration	mm/d

10.1 Run Initialization

At the beginning of simulation, the value of KCAN, the canopy light extinction coefficient for daily PAR calculations, is read from FILEC (e.g., SBGRO980.SPE). FILEIO is opened to obtain the name of FILEC.

10.2 Rate Calculations

The light interception by leaf canopy, FDINT, is estimated with:

```
FDINT = 1.0 - EXP(-(KCAN + 0.15) * XHLAI)
```

Potential plant transpiration is defined by the following equation:

```
EOP = EO * FDINT * TRAT
```

Where EOP = Potential plant transpiration (mm)
EO = Potential evapotranspiration (mm)
FDINT = Leaf light interception factor
TRAT = Relative transpiration rate for CO2 values other than 330 ppm

The following equation reduces potential transpiration if the sum of potential transpiration plus potential evaporation exceeds the calculated potential evapotranspiration:

```
IF ((EOP + ES) .GT. (EO * TRAT)) EOP = EO * TRAT - ES
```

Function subroutine TRATIO computes the value of TRAT, used in the above equations. This factor reduces transpiration to account for the effects of increased atmospheric CO2 concentrations on stomatal resistance. The ratio is typically close to 1.0 (Peart, et al. 1989). Subroutine BLRRES is called by TRATIO to compute the boundary layer resistance.

11.0 XTRACT – Root Water Extraction

Subroutine XTRACT computes the change in volumetric soil water content due to plant transpiration, SWDELTX(L). A temporary soil water content variable, SWTEMP(L), is used in order to calculate changes in water content without modifying today's value of soil water content, SW(L). This preserves values of SW(L) for rate computations in other modules.

Table 11 lists input and output variables used by XTRACT. The FORTRAN code is listed in Appendix R.

Table 11 – Input and Output Data Requirements for XTRACT

Input Variables:		
DLAYR(L)	Soil Depth in layer L	cm
LL(L)	Volumetric soil water content in soil layer L at lower limit	cm^3[water]/cm^3[soil]
NLAYR	Actual number of soil layers	--
SW(L)	Volumetric soil water content in layer L	cm^3[water]/cm^3[soil]
SW_AVAIL(L)	Soil water content in layer L available for evaporation, plant extraction, or movement through soil	cm^3[water]/cm^3[soil]
Input / Output Variables:		
EP	Actual plant transpiration rate	mm/d
RWU(L)	Root water uptake from soil layer L	cm/d
TRWU	Total potential daily root water uptake	cm/d
Output Variables:		
SWDELTX(L)	Change in soil water content due to root uptake in layer L	cm^3[water]/cm^3[soil]

Subroutine XTRACT calculates the change in soil water content due to root water uptake, RWU, as calculated in ROOTWU. If the transpiration rate, EP, limits root water uptake, values of RWU for each soil layer are proportionately reduced with variable WUF. The lower limit of each soil layer (LL) further limits root water uptake. Array, SW_AVAIL(L) contains the current soil water content which has been modified to include drainage and upward flows calculated in routines INFIL, SATFLO and UPFLOW.

At the end of the routine, the root extraction in each layer, SWDELTX(L) is computed as the difference in soil water content and the modified SWTEMP variable:

```
DO L = 1, NLAYR
  SWDELTX(L) = SWTEMP(L) - SW(L)
ENDDO
```

12.0 OPSPAM – Output routine

The supplementary subroutines, which are used to input and output data for the SPAM main module, are IPWBAL and OPSPAM.
Subroutine OPSPAM is called four times from the SPAM module for 'Seasonal Initialization', 'Integration', 'Output' and 'Final' calculations.

12.1 Seasonal Initialization

Upon initialization, the OPSPAM routine opens and writes headings to the OUTW file. The switch IDETL is used to determine if detailed output is required for this simulation.

Variables that are used to calculate average values of evapotranspiration and weather data are initialized to zero.

12.2 Integration

Average values of evapotranspiration and weather data are calculated as a function of the output interval.

12.3 Output

Computed data are written to the OUTW file if the variable IPRINT, computed in the CROPGRO main program, indicates that it is required.

12.4 Final

The OUTW file is closed in the 'Final' section of code.

13.0 References

Allen, R.G., L.S. Pereira, D. Raes, M. Smith. 1998. Crop Evapotranspiration, Guidelines for computing crop water requirements. Irrigation and Drainage Paper No. 56. Food and Agriculture Organization fo the United Nations, Rome, 300pp.

Boote, K.J., J.W. Jones and G. Hoogenboom. 1998. Simulation of Crop Growth: CROPGRO Model. In Agricultural Systems Modeling and Simulation. Ch. 18. Eds. R.M. Peart and R. B. Curry. Marcel Dekker, Inc.

Doorenbos, J. and W.D. Pruitt. 1977. Guidelines for Predicting Crop Water Requirements. Food and Agriculture Organization of the United Nations, Irrigation and Drainage Paper No. 24.

IBSNAT. 1994. Decision Support System for Agrotechnology Transfer, Version 3 (DSSAT V3). Eds. G.Y. Tsuji, G. Uehara, and Sharon Balas. International Benchmark Sites Network for Agrotechnology Transfer Project. Honolulu, HI: University of Hawaii.

Jones, James W., Hoogenboom, G. , Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., Wilkens, P. W., Singh, U. , Gijsman, A. J., Ritchie, J. T.. *The DSSAT cropping system model*. European Journal of Agronomy. 18(3-4) 2003. : Pp. 235 -265.

Jones, C.A., J.T. Ritchie, J.R. Kiniry, and D.C. Godwin. 1986. *Subroutine Structure*. Ch. 4 in CERES-Maize: A Simulation Model of Maize Growth and Development. Eds. C.A. Jones and J.R. Kiniry. Texas A&M University Press.

Jones, J.W., L.H. Allen, S.F. Shih, J.S. Rogers, L.C. Hammond, A.G. Smajstrala, and J.D. Martsolf. 1984. "Estimated and Measured Evapotranspiration for Florida Climate, Crops, and Soils". Bulletin 840, Agricultural Experiment Stations, institute of Food and Agricultural Sciences, University of Florida.

Jones, J.W. and J. T. Ritchie. 1990. Crop Growth Models. In *Management of Farm Irrigation Systems*, Ch. 4, eds. G. J. Hoffman, T. A. Howell and K. H. Solomon. American Socient of Agricultural Engineers.

Kenig A., and Jones J.W., 1997. Model structure for dynamic crop-greenhouse simulations. In: Seginer I., Jones J.W., Gutman P., and Vallejos C.E. (eds), Optimal environmental control for indeterminate greenhouse crops. Final Report, BARD Research Project No. IS-1195-91RC, Agricultural Engineering Department, Technion, Haifa, Israel.

Kraalingen D.W.G. van, 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis Report no. 1. AB/DLO, PE, Wageningen, 88pp.

Monteith, J.L. 1986. How do crops manipulate water supply and demand? Phil.Trans R. Soc. Lond. A 316:245-259.

Monteith, J.L., A.K.S. Huda, and D. Midya. 1989. RESCAP: A resource capture model for Sorghum and Pearl Millet. p. 30-34. In: Modeling the growth and development of sorghum and pearl millet. Eds. S.M. Virmani, H.L.S. Tandon and G. Alagarswamy. Research Bulletin No. 12. ICRISAT. Andhra Pradesh, India.

Passioura, J.B. 1991. Soil Structure and Plant Growth. Aust. J. Soil Sci. Res. 29:717-728.

Peart, R.M., J.W. Jones, R.B. Curry, K.J. Boote and L.H. Allen, Jr. 1989. Impact of climate change on crop yield in the southeastern U.S.A.: A simulation study. In The potential effects of global climate change on the United States, eds. J.B. Smith and D.A. Tirpak, Vol. 21:2-1 to 2-54. EPA-2330-05-89-050, U.S.E.P.A., Appendix C, Agriculture, EPA-230-05-053.

Penman, H.L. 1956. Evaporation: An introduction survey. Neth. J. Agr. Sci. 4:8-29.

Porter, C. H., J. W. Jones, R. Braga. 2000. An approach for modular crop model development. International Consortium for Agricultural Sytems Application, 2440 Campus Rd., 527 Honolulu, HI 96822, pp. 13. Available from http://www.icasanet.org/modular/index.html.

Porter, C.H. and J.W. Jones. 1998. Modular Structure in DSSAT Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Priestley, C.H.B. and R.J. Taylor. 1972. On the assessment of surface heat flux and evaporation using large scale parameters. Mon. Weath. Rev. 100:81-92.

Ritchie, J.T. 1972. Model for Predicting Evaporation from a Row Crop with Incomplete Cover. *Water Resources Research* 8(5):1204-1213.

Ritchie, J.T. 1985. A User Oriented Model of the Soil water Balance in Wheat. In Wheat Growth and Modeling. Eds. W. Day and R.K. Atykin. Plenum Press.

Ritchie, J.T., D.C. Godwin and U. Singh. 1989. Soil and Weather Inputs for the IBSNAT Crop Models. In *Proceedings of the IBSNAT Symposium: Decision Support System for Agrotechnology Transfer*. American Society of Agronomy, Las Vegas. Nevada.

Soil Conservation Service. 1985. National Engineering Handbook Section 4: Hydrology. USDA.

Extension of an Existing Model for Soil Water Evaporation and Redistribution under High Water Content Conditions

Joe T. Ritchie
Cheryl H. Porter*
Jasmeet Judge
James W. Jones
Agricultural and Biological Engineering Dep.
Univ. of Florida
P.O. Box 110570
Gainesville, FL 32611

Ayman A. Suleiman
Dep. of Land, Water and Environment
Univ. of Jordan
Amman, Jordan

Most crop, hydrology, and water quality models require the simulation of evaporation from the soil surface. A model developed by J.T. Ritchie in 1972 provides useful algorithms for estimating soil evaporation, but it does not calculate the soil water redistribution resulting from evaporation. A physically-based model using diffusion theory, described previously by Suleiman and Ritchie in 2003, provides efficient algorithms for soil water redistribution and soil evaporation. However, the model is appropriate only for second stage drying when the soil in the entire profile being simulated is below the drained upper limit (θ_{DUL}) and no more drainage occurs due to gravity. This paper extends the Suleiman–Ritchie model for soil water contents higher than θ_{DUL} where soil evaporation rates are usually higher than second stage drying. New algorithms were developed for these wetter conditions that are functions of soil depth and the wetness of the near-surface soil. New model parameters were calibrated with data measured in laboratory soil column studies. The resulting model was integrated into DSSAT-CSM (Decision Support System for Agrotechnology Transfer Cropping Systems Model). Simulated soil evaporation rates and soil water contents obtained using the Suleiman–Ritchie model with the developed extensions and the previous DSSAT soil evaporation model were compared and evaluated with field measurements of soil water content during several drying cycles for parts of 3 yr in North Central Florida. Computed soil water contents from the model agreed well with the measured soil water contents near the surface, and provided more accurate estimations than the original DSSAT soil evaporation model, especially for the 5-cm surface layer.

Abbreviations: θ_{DUL}, Soil water content at drained upper limit ($cm^3\ cm^{-3}$); CSM, Cropping Systems Model; DSSAT, Decision Support System for Agrotechnology Transfer; ES, evaporation of water from soil surfaces (cm); ESR, Extended Suleiman and Ritchie soil evaporation model; LAI, Leaf area index ($m^2\ m^{-2}$); MAE, mean average error; MicroWEX, Microwave, Water, and Energy Balance Experiment; ORD, Original Ritchie DSSAT soil evaporation model; SR, Suleiman and Ritchie method of soil evaporation using diffusion theory.

Evaporation of water from soil surfaces (ES) is a major component in the soil water balance for field crops with incomplete cover and for bare soil conditions. Quantification of ES is necessary in evaluating the water balance of soils for use in environmental and hydrologic studies and for crop management. Two basic approaches have been used to simulate ES: (i) mechanistic models of soil water and heat transfer following basic theory reported by Philip and DeVries (1957), and (ii) functional models similar to that of Ritchie (1972). The mechanistic models have proven to work well for uniform laboratory soil conditions and have been demonstrated to work reasonably well for field conditions (Jackson et al., 1974; Rose, 1968a, 1968b; Lascano and van Bavel, 1983, 1986). Mechanistic models require more detailed spatial and temporal inputs and calculations (Ritchie and Johnson, 1990).

Functional models have also proven to work well for field conditions (Ritchie, 1972; Black et al., 1969). Unlike mechanistic models, functional models for ES have used crude but logical approximations for soil water redistribution (Ritchie, 1998). Recently, Suleiman and Ritchie (2003), hereafter referred to as SR, introduced a physically-based model for soil water redistribution during evaporation using a diffusion-based concept. The procedure required an initial condition of soil water content being equal to or less than the drained upper limit (θ_{DUL}) throughout the profile being simulated. Under these conditions, water would no longer drain from the profile. When the soil water content is above θ_{DUL} in any part of the profile, the empiricisms in the SR procedure underestimate ES and thus do not simulate the soil water distribution accurately. Without modification, the SR model was not suitable for implementation in a general soil water simulation model because it could not adequately model wet soil conditions. This paper develops and tests an extension to the SR approach when all or part of the soil profile is above θ_{DUL} such as after rainfall or irrigation or when a shallow water table is present. The model has been implemented in the comprehensive DSSAT-CSM (Jones et al., 2003), which uses a daily time increment.

Soil Sci. Soc. Am. J. 73:792-801
doi:10.2136/sssaj2007.0325
Received 6 Sept. 2007.
*Corresponding author (cporter@ufl.edu).
© Soil Science Society of America
677 S. Segoe Rd. Madison WI 53711 USA
All rights reserved. No part of this periodical may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Permission for printing and for reprinting the material contained herein has been obtained by the publisher.

The objectives of this study were: (i) to extend the SR soil evaporation model for use with wetter soil water conditions; (ii) to implement this extended model into DSSAT-CSM; (iii) to evaluate the model performance within DSSAT using field and lab experimental data; and (iv) to compare soil evaporation rates and soil water contents predicted by the extended SR model with those of the previous soil evaporation model within DSSAT-CSM.

BACKGROUND

The SR paper presented two sets of laboratory soil column studies using different initial conditions. Although not mentioned in the procedures section of that paper, 60 cm deep columns were brought to the initial conditions of θ_{DUL} throughout the profile by using suction at the bottom of the columns. The columns were placed in a container of dry soil to provide the suction until the θ_{DUL} water contents were obtained. Under those initial conditions, measured soil water in depth and time was in good agreement with the model based on the SR diffusion theory. A second set of 150 cm deep columns were used with stable initial conditions established by gravity drainage with no suction at the bottom. Measured soil water contents during drying of these columns did not result in values that agreed with the diffusion theory concept. The slope of cumulative ES values plotted against square root of time was about four times larger than those obtained when the boundary condition of θ_{DUL} were met throughout the soil profile. Water contents plotted against the Boltzmann transformation (depth · time$^{-0.5}$) did not result in a single-valued function as did the soils with the appropriate water content boundary conditions.

The model of Ritchie (1972) simulates ES in two stages: (i) first stage or energy-limited, whereby soil water can move to the surface faster than the evaporativity of the air could remove it and (ii) second stage, whereby soil properties control the rate of water movement to the surface. There was assumed to be a constant amount of water available for evaporation in the energy limited stage, ranging from 2 to 15 mm. This value was a required input parameter to the model and was thought to be a function of the soil hydraulic properties below the surface. However, the laboratory columns in the SR study that were not drained with a suction at the bottom had a much larger ES in the energy limited stage than the Ritchie (1972) paper implied for both the sandy loam and loam soil. This evidence suggests that the energy limited evaporation stage is primarily controlled by hydraulic properties of the soil below the surface that prevent water from draining out of the soil as rapidly as it does in well-drained soils.

Data from the SR paper also demonstrate that ES was smaller than the potential evaporation after about 3 d of drying but much higher than the typical second stage evaporation. Thus, true second stage drying, when ES is proportional to the square root of time after wetting, only occurs when the water in both the upper and the lower levels of the soil have drained to near or below θ_{DUL}. This implies that there are no clear boundaries to describe first and second stage ES.

MATERIALS AND METHODS

Soil Evaporation Model Based on Diffusion Theory

The primary equation derived from diffusion theory by Suleiman and Ritchie (2003) is shown below. This equation was derived for soils after they had been allowed to drain to θ_{DUL} throughout the soil profile.

$$\Delta\theta_{ESz,t} = -\left(\theta'_{z,t} - \theta_{ADz}\right)F_z \qquad \text{for } \theta'_{z,t} < \theta_{DULz} \qquad [1]$$

where $\Delta\theta_{ESz,t}$ = daily change in volumetric water content at depth z, due to evaporation on day t (cm^{3}cm^{-3}d^{-1}). $\theta'_{z,t}$ = volumetric water content at depth z, on day t calculated from the previous day's $\theta_{z,t-1}$ and simultaneous drainage and infiltration processes on day t (cm^{3}cm^{-3}). Computation of this term is detailed later in this paper. θ_{ADz} = air dry volumetric water content at depth z (cm^{3}cm^{-3}). θ_{DULz} = volumetric water content at drained upper limit at depth z (cm^{3}cm^{-3}). F_z = transfer coefficient for soil at depth z (d^{-1}). z = mean depth of soil layer (cm).

The upward flow coefficient (F_z) in Eq. [1], as reported by SR is computed as:

$$F_z = a_z z^{(b_z)} \qquad \text{for } z > 1 \text{ cm} \qquad [2]$$

Coefficients a_z and b_z are empirical coefficients at depth z, which were developed according to diffusion theory. The parameters used to compute these coefficients were based on laboratory experiments. These regression equations are independent of soil type and are given by the following equations:

$$a_z = 0.5 + 0.24\theta_{DULz} \qquad [3]$$

$$b_z = -2.04 + 0.20\theta_{DULz} \qquad [4]$$

The total daily soil evaporation, ES_t, is computed by summing the evaporation from each soil layer, accounting for layer thickness, d_z, and by limiting the total to no more than the potential soil evaporation, E_{OSt}, as shown in Eq. [5].

$$ES_t = MIN\left\{E_{OSt}, \sum_z \Delta\theta_{ESz,t} d_z\right\} \qquad [5]$$

The potential evapotranspiration is calculated separately in DSSAT-CSM by the user's choice of several methods. When the total soil evaporation is limited by potential soil evaporation, the computed evaporation from each layer is reduced proportionally.

Pseudo-Integration Step for Modular Daily Model

The modular structure of DSSAT-CSM (Jones et al., 2003) requires that all rate processes be calculated each day based on state variables computed on the previous day, using a daily time step. Integration of the model processes such as drainage and evaporation, which occur simultaneously and are computed in separate modules, becomes problematic. The $\theta'_{z,t}$ in Eq. [1] would normally represent the soil water content at the end of the previous day. However, soil evaporation calculations need to take into account the effects of both drainage and infiltration, which are occurring simultaneously during the current day of simulation. Therefore, a pseudo-integration step computes $\theta'_{z,t}$, which represents an average value of soil water content on day t as computed in Eq. [6a] and [6b].

$$\theta'_{z,t} = \theta_{z,t-1} + 0.5\Delta\theta_{Dz,t} \qquad \text{for } \Delta\theta_{Dz,t} > 0 \qquad [6a]$$

$$\theta'_{z,t} = \theta_{z,t-1} + \Delta\theta_{Dz,t} \qquad \text{for } \Delta\theta_{Dz,t} \le 0 \qquad [6b]$$

where $\theta_{z,t-1}$ = volumetric water content at depth z, on day t-1 (cm^{3}cm^{-3}) and $\Delta\theta_{Dz,t}$ = change in volumetric water content at depth z, due to drainage and infiltration on day t (cm^{3}cm^{-3}d^{-1}).

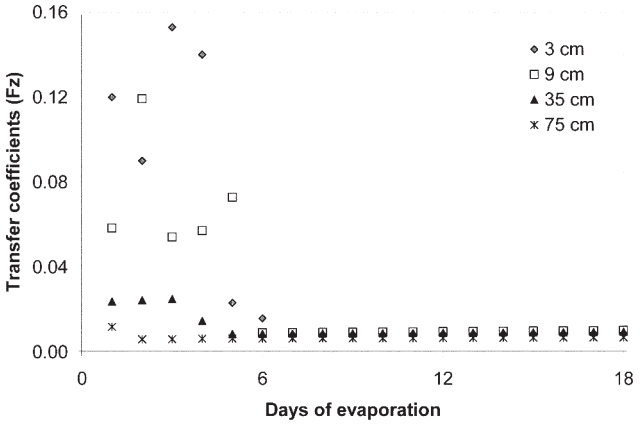


Fig. 1. Transfer coefficients computed from measured data in the Suleiman and Ritchie 150-cm wet soil column study for four depths of the Loam soil.

Any increase in soil water content due to rainfall on the current day would normally not be "seen" by the soil evaporation routine until the following day. The 0.5 parameter in Eq. [6a] allows half of the simultaneously occurring increase in soil water content to become available for evaporation on the current day of simulation rather than on the subsequent day, implying that rainfall or irrigation occur closer to midday than midnight. Equation [6b] applies to days where drainage reduces the soil water available for evaporation. This equation prevents calculation of excess evaporation that could lead to negative computed soil water content.

Drainage is computed in a separate module of DSSAT-CSM. This procedure uses a tipping-bucket approach for computing soil water drainage when water content in a soil layer is greater than θ_{DUL} (Ritchie et al., 1986). Drainage through the profile is first calculated based on an overall soil drainage parameter assumed to be constant with depth. The rate of drainage in each soil layer is limited by the saturated hydraulic conductivity of the layer.

Extension of Suleiman and Ritchie Model for Wetter Soils

In the SR paper, soil water data for wet conditions were presented, but not used in deriving the transfer coefficients for Eq. [1]. When the water contents were plotted against the Boltzmann transformation

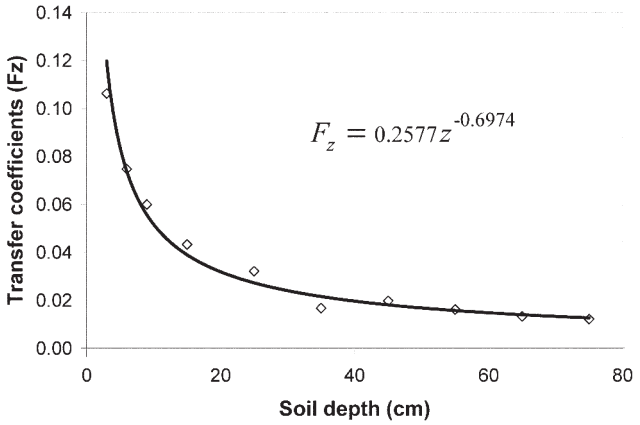


Fig. 2. Transfer coefficients averaged from both soils in the Suleiman and Ritchie 150-cm wet soil column study for the first 4 d of the drying cycle as a function of depth.

(depth · time$^{-0.5}$), no single-valued function could be obtained, indicating that the diffusivity theory was not valid for conditions where there was a water table within 150 cm of the soil surface.

Equations [1] and [2] can be applied to the wetter soil profiles, but the coefficients used in the transfer equation must be re-evaluated. Equation [1] can be rearranged to solve for F_z when all other factors are known:

$$F_z = \frac{-\Delta\theta_{ESz,t}}{\left(\theta'_{z,t} - \theta_{ADz}\right)} \quad [7]$$

Data from the SR soil column studies during the 18-d drying cycle were used to compute the transfer coefficients for wetter soils using Eq. [7]. Typical values obtained are depicted in Fig. 1 for four depths for the loam soil. Results from the sandy loam were similar to those for the loam soil. The transfer coefficients were approximately constant after about 5 d of drying with a value of about 0.011 d^{-1}. Both the sandy loam and the loam soil lost water at approximately this rate at all depths. However for the first 4 to 5 d of drying the transfer coefficients were higher and decrease rapidly with soil depth. These transfer coefficients were similar for both soils with depth and time, thus they were averaged for the first 4 d of the drying cycle. The averages are depicted as a function of depth in Fig. 2. The data were fit to the transfer coefficient function in Eq. [2] resulting in a parameters of best fit, a_z = 0.26 and b_z = -0.70. These results provide the necessary transfer coefficients to describe evaporation from the wetter soil profiles. We refer to three conditions of soil water, each with a different set of transfer coefficients as: (i) wet, when the upper layers have the highest water contents, as in Fig. 2; (ii) equilibrium, when the lower layers have water content above θ_{DUL} with constant transfer coefficients, as in Fig. 1; and (iii) dry, when the soil water contents in all layers are below θ_{DUL} and the SR transfer coefficients (Eq. [3] and [4]) are appropriate to use.

We developed an appropriate empiricism to determine the conditions for transitioning between the three soil moisture conditions based on the data available from the SR paper for the wetter initial soil conditions. The soil water content near the surface appears to be the factor that controls the timing of the change from the wet condition with higher transfer coefficients to the equilibrium condition with smaller transfer coefficients. The thickness of the surface layer is a factor in the time of transition to the equilibrium condition since there is a steep water content gradient near the surface at the time of the transition. We extrapolated the water content with depth on day four for the sandy loam and day five for the loam and found resulting water contents of 0.13 cm^{3}cm^{-3} for the sandy loam (θ_{DUL} = 0.24 cm^{3}cm^{-3}) and 0.20 cm^{3}cm^{-}3 for the loam (θ_{DUL} = 0.32 cm^{3}cm^{-3}). Since θ_{DUL} varies with soil type, we fit an empiricism to θ_{DUL} to approximate the point of transition between wet and equilibrium conditions. The resulting equation defines θ_{eq}, the threshold water content for transitioning between wet and equilibrium conditions:

$$\theta_{eq} = 0.275\theta_{DULz} + 1.165\left(\theta_{DULz}\right)^2 + 1.2z\left(\theta_{DULz}\right)^{3.75} \quad [8]$$

This equation extrapolates to a logical zero when θ_{DUL} is zero. For practical reasons, a depth of 100 cm was used to determine when to transition to the dry transfer coefficients of the SR paper. When all soil water contents within the top 100 cm are at or below θ_{DUL}, the SR equations are applied. The theory assumes that there is no water table at infinite depth, however our data did not include greater

depths and many simulation models simulate only the upper 100 cm. Further testing with deeper soil water measurements in the presence of a water table may allow for improvements in these transition details. However, it appears that these equations can provide a reasonable approximation of soil evaporation from systems with dynamic soil water conditions for various soil types until more data are available from a broader range of soils.

Implementation of the Extended Suleiman and Ritchie Model in Decision Support System for Agrotechnology Transfer

The modeling details presented herein were developed for functional models such as the DSSAT system using a 1 d time step and a relatively small number of depth increments. The depth increments used in the development and testing of the model within the DSSAT system were 5, 10, 15, 15, 15, 30, and 30 cm for the 120-cm soil profile for the field experiments, which are described below. An additional 30-cm layer was used in the simulations of the 150-cm laboratory soil column study. The procedure should not be used, at least without modification, for the simulation of a surface layer thinner than 1.5 cm. Such a shallow layer could result in a transfer coefficient extrapolated to near or above one, which could result in negative water content calculations.

Using the extension of the SR model for wetter soils and the SR model for the dry conditions, the equations were included in the DSSAT-CSM system using the three-threshold system described above. Equations [3] and [4] are used to compute the transfer coefficients for the dry soil profile condition, when the soil water contents of all layers within the top 100 cm are less than θ_{DUL}. The wet soil profile transfer coefficients are computed using a_z = 0.26 and b_z = -0.70, from Fig. 2, when any soil layer in the top 100 cm has a water content above θ_{DUL} and the top soil layer is wetter than a threshold value, θ_{eq} (Eq. [8]). The equilibrium transfer coefficient, F_z = 0.011, is used when the soil profile is wet, but the top layer is dryer than the threshold value.

Field Experiments—Microwave, Water, and Energy Balance Experiments

Microwave, Water, and Energy Balance Experiments (MicroWEXs) are intensive, season-long field experiments conducted to monitor micrometeorological, soil, and vegetation conditions as well as the remotely sensed microwave signatures during growing seasons for corn and cotton. In this study, we use data from the second (Judge et al., 2005), third (Lin et al., 2005), fourth (Casanova et al., 2005), and fifth (Casanova et al., 2007) MicroWEXs. Table 1 lists the starting and ending dates of simulations associated with each of the four MicroWEXs analyzed.

The experimental site was a 3.6 ha field located at the University of Florida Plant Science Education and Research Unit (29° 24' 0" N, 82° 10' 12" W) in Citra, Florida. The soils of the site were Millhopper Fine sand (loamy, siliceous, semiactive, hyperthermic Grossarenic Paleudults), with 92% sand and a bulk density of 1.4 g/cm^3. Water and fertilizer were applied through a linear-move irrigation/fertigation system. Datasets used for the study included observations of soil water at depths of 2, 4, 8, 16, 32, 64, and 100 cm, downwelling solar radiation, air temperature, water table depth, and precipitation/irrigation. In addition, weekly leaf area index (LAI) measurements were used to determine when to end the simulations of soil evaporation.

Table 1. List of datasets from experiments used for analyses.

Experiment†	Crop‡	Planting Date	Simulation Start Date	Simulation End Date
MicroWEX-2	Corn	18 Mar. 2004	26 Mar. 2004	16 Apr. 2004
MicroWEX-3	Cotton	9 July 2004	13 July 2004	6 Aug. 2004
MicroWEX-4	Corn	11 Mar. 2005	17 Mar. 2005	8 Apr. 2005
MicroWEX-5	Corn	9 Mar. 2006	14 Mar. 2006	5 Apr. 2006

† MicroWEX, Microwave, Water, and Energy Balance Experiment.
‡ Corn (*Zea mays* L.) and Cotton (*Gossypium hirsutum* L.).

Although extensive soil and environmental measurements were recorded throughout the growing season, the soil evaporation model simulations were run only for the early portion of the season, before significant shading of the soil surface by plant leaves and soil water uptake by roots. The simulation start dates corresponded to dates of availability of observed data during each experiment after planting. The simulation end dates for each experiment corresponded to the date when plant transpiration just exceeded 10% of the total potential evapotranspiration. Based on simulations of the maize growth in MicroWEX-2, this criterion was met at a measured LAI of approximately 0.3 m^{2}m^{-2}. Under these conditions the losses of soil water resulted only from evaporation and drainage since significant soil shading or root uptake had not yet occurred. When the water content dropped below θ_{DUL}, the daily change in soil water content represented soil evaporation only.

The values of θ_{DUL} used in the simulations for these experiments were determined based on graphical inspection of the detailed soil water data collected in the MicroWEXs. Values of θ_{DULz}, empirical coefficients, a_z and b_z, and the transfer coefficients, F_z, for the upper two layers as used in the MicroWex simulations are presented in Table 2. The soil layers have similar properties. The soil profile drainage parameter used for simulations of the MicroWEX studies was computed based on a method outlined by Suleiman and Ritchie (2004).

Model Comparisons and Evaluation

The extended SR soil evaporation model as described herein was implemented in DSSAT v4.5 (Hoogenboom et al., 2009) as an optional method for computing soil evaporation. Table 3 presents the differences between the computer algorithms of the original DSSAT soil evaporation model (Ritchie, 1972, 1998) (ORD) and the extended (ESR) model. The ESR routine is simpler with fewer input parameters.

Table 2. Extended Suleiman and Ritchie (ESR) soil evaporation transfer coefficients for the soils used in the simulations of Microwave, Water, and Energy Balance Experiments (MicroWEXs) for the three soil moisture conditions. Values of the volumetric water content at drained upper limit at depth z (θ_{DULz}) are 0.096 and 0.105 cm^3 cm^{-3}, respectively for the 0- to 5-cm and 5- to 15-cm layers.

	a_z†	b_z†	F_z†
Dry profile condition:			
0–5 cm	0.523	–2.02	0.0821
5–15 cm	0.525	–2.02	0.00503
Equilibrium profile condition:			
0–5 cm	0.011	0.0	0.011
5–15 cm	0.011	0.0	0.011
Wet profile condition:			
0–5 cm	0.26	–0.70	0.137
5–15 cm	0.26	–0.70	0.0519

† a_z and b_z, empirical coefficients at depth z; F_z, upward flow coefficient.

Table 3. Comparison of the two algorithms used for modeling soil evaporation in Decision Support System for Agrotechnology Transfer (DSSAT).

	ORD†	ESR†
Input data required		
Soil properties	•	•
Soil water content, $mm^3\ mm^{-3}$	•	•
Potential soil evaporation, cm	•	•
Change in soil water content due to drainage, $mm^3\ mm^{-3}$	•	•
Stage-1 evaporation limit, cm	•	
Infiltration from rainfall or irrigation, cm	•	
Procedures		
Multi-stage evaporation	•	•
Evaporation and redistribution calculated for each soil layer		•
Lines of computer code	252	91

† Original Ritchie DSSAT (ORD) and Extended Suleiman and Ritchie (ESR) soil evaporation models.

Weather and soil inputs to the DSSAT model during the four MicroWEX field experiments were obtained from the observations. Each experiment was simulated using both the ORD and ESR soil evaporation models within DSSAT-CSM to predict soil water content at various depths. The DSSAT system divides the soil into several layers, the upper two layers having depth increments of 5 and 10 cm, respectively. Water content changes at depths below 15 cm were small and thus not provided in this paper although they were both simulated and measured. The simulated soil water contents from both models were compared with the measured data from the MicroWEXs, both qualitatively (graphically) and quantitatively (statistically) for their capability to simulate soil water contents and soil evaporation.

For each of the models, the bias, mean average error (MAE), root mean square error (RMSE), relative root mean square error (RRMSE), model efficiency (ME), and Wilmott D-index were computed. Methods described in Wallach et al. (2006), Nash and Sutcliffe (1970), and Willmott (1982) were used for these calculations.

Kobayashi and Salam (2000) presented an approach for model evaluation based on the mean squared error and showed that it is better suited to model evaluation than regression. The method uses mean squared error (MSE) and breaks it down into its relevant components, which can be calculated from common statistical parameters such as bias, standard deviation and correlation coefficient:

$$\text{MSE} = \text{SB} + \text{SDSD} + \text{residual error} \quad [9]$$

where SB is the squared bias of error and SDSD is the squared difference of standard deviations. The SDSD component indicates how well the model simulates the magnitude of fluctuation between measured values. A smaller SDSD indicates that the model provides a better simulation of the magnitude of fluctuation.

$$\text{SB}=\left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i-x_i\right)\right]^2 \quad [10]$$

where N is the number of observations and x_i and y_i are the simulated and observed values, respectively.

$$\text{SDSD}=\left(\sigma_x-\sigma_y\right)^2 \quad [11]$$

where σ_x and σ_y are the standard deviations of the simulated and measured values.

Modeling efficiency (Nash and Sutcliffe, 1970), ranges from minus infinity (poor model) to 1.0 (perfect model) and can be computed as:

$$\text{ME}=1-\frac{\text{MSE}}{\sigma_y^{\,2}} \quad [12]$$

Willmott index of agreement, or D-index, is calculated as:

$$\text{D-index}=1-\frac{\sum_{i=1}^{N}\left(y_i-x_i\right)^2}{\sum_{i=1}^{N}\left(\left|x_i-\overline{y}\right|+\left|y_i-\overline{y}\right|\right)^2} \quad [13]$$

where $\overline{y}$ = mean of observed values. D-index varies from 0.0, for a poor model, to 1.0, for a perfect model.

Each of these measures are used in this paper to quantify and compare errors between simulated and observed soil water values for the top two soil layers for both the ORD and the ESR soil evaporation models. Additionally, for the ESR model, statistical parameters were computed for data grouped by the individual MicroWEXs, for the 0- to 5-cm soil layer.

Soils and environmental conditions for the wetter soil column study from the SR paper were also input to the ESR model within DSSAT-CSM to simulate soil water content and soil evaporation rates for this laboratory study. The model outputs were compared graphically with the measured laboratory data to evaluate the ESR model performance. Because model transfer coefficients were developed using these SR experimental data, this analysis does not represent an independent evaluation of the ESR as it does for the MicroWex field studies. Therefore, only graphical rather than statistical analyses are presented for the simulations of the data from the SR column study.

RESULTS AND DISCUSSION

Qualitative Comparison of Model Errors with Graphical Analyses for MicroWEXs

Table 4 presents a summary of environmental conditions and simulated outputs for each of the four MicroWEXs including the rainfall and irrigation inputs to the fields and simulated evaporation. Figures 3a and 3b present graphical comparisons of the simulated model outputs with the data collected during MicroWEX-2 for the 0- to 5-cm and 5- to 15-cm soil layers, respectively. Similar graphs are presented in Fig. 4a and 4b for MicroWEX-3, in Fig. 5a and 5b for MicroWEX-4, and in Fig. 6a and 6b for MicroWEX-5.

Changes in water content were greatest on days with rainfall or irrigation events when water was added to the system. Drainage and evaporation caused the subsequent loss of water. Most of the simulated water content in the 0- to 5-cm top soil layer agreed well with the trend in the measured water content. The ORD model simulated soil water values that responded well after rainfall or irrigation, but were consistently lower than measured water contents thereafter. The ESR model behaved much more realistically in the upper soil layer. Both models agreed well with the measured data in the 5- to 15-cm depth and for lower soil layers, which are not shown.

Figure 7 presents daily simulated soil evaporation rates for both models for the four MicroWEXs. For all four experiments, the cumulative soil evaporation simulated by the ESR model is higher than that computed by the ORD model

Table 4. Comparisons of environmental conditions and simulated evaporation during the four MicroWEXs.

Measured or computed data	MicroWEX-2 2004 Corn†	MicroWEX-3 2004 Cotton†	MicroWEX-4 2005 Corn	MicroWEX-5 2006 Corn
Simulation duration, d	22	25	23	23
Measured total rainfall, mm	26	196	162	4
Measured total irrigation, mm	55	15	76	48
Average rainfall + irrigation, mm d^{-1}	3.7	8.4	10.3	2.3
Measured maximum temperature, °C	29.6	34.3	31.1	30.2
Measured minimum temperature, °C	4.2	20.1	1.0	0.3
Measured average solar radiation, MJ m^{-2} d^{-1}	21.1	18.7	14.8	22.1
Computed total potential evapotranspiration, mm	121	132	83	117
Computed total soil evaporation (ORD model), mm ‡	37	48	40	31
Computed total soil evaporation (ESR model), mm ‡	53	68	55	45

† Corn (Zea mays L.) and Cotton (Gossypium hirsutum L.).

‡ Original Ritchie DSSAT (ORD) and Extended Suleiman and Ritchie (ESR).

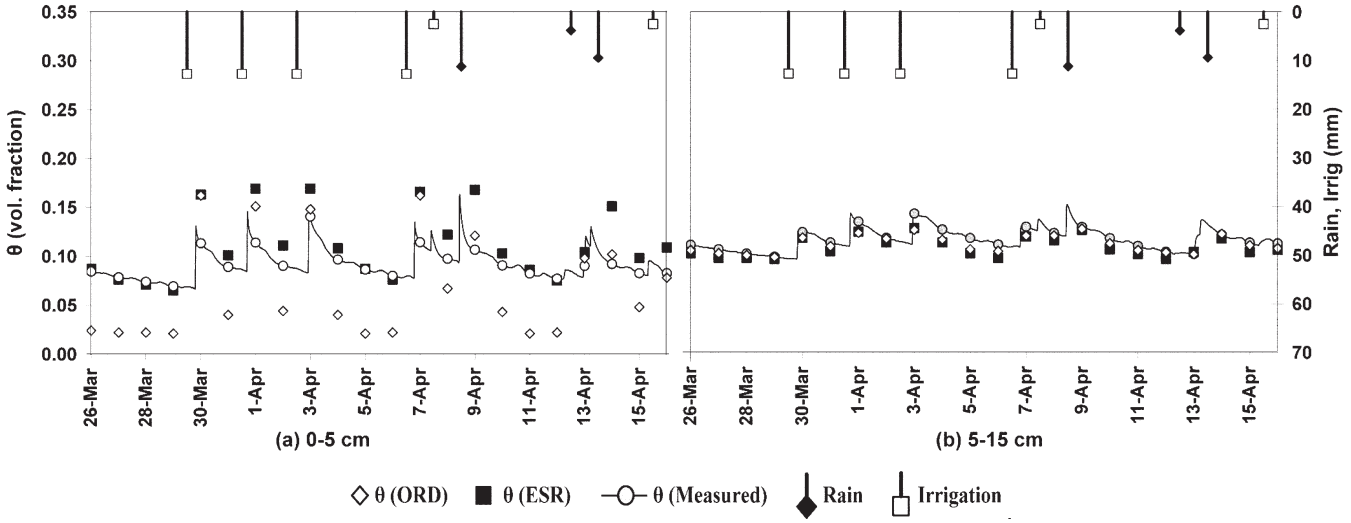


Fig. 3. Simulated and measured volumetric soil water content by date for the Extended Suleiman and Ritchie (ESR) and the Original Ritchie DSSAT (ORD) soil evaporation models during Microwave, Water, and Energy Balance Exp. 2 (MicroWEX-2) in 2004 for (a) 0- to 5-cm and (b) 5- to 15-cm soil layers. Measured points used in statistical analyses are shown.

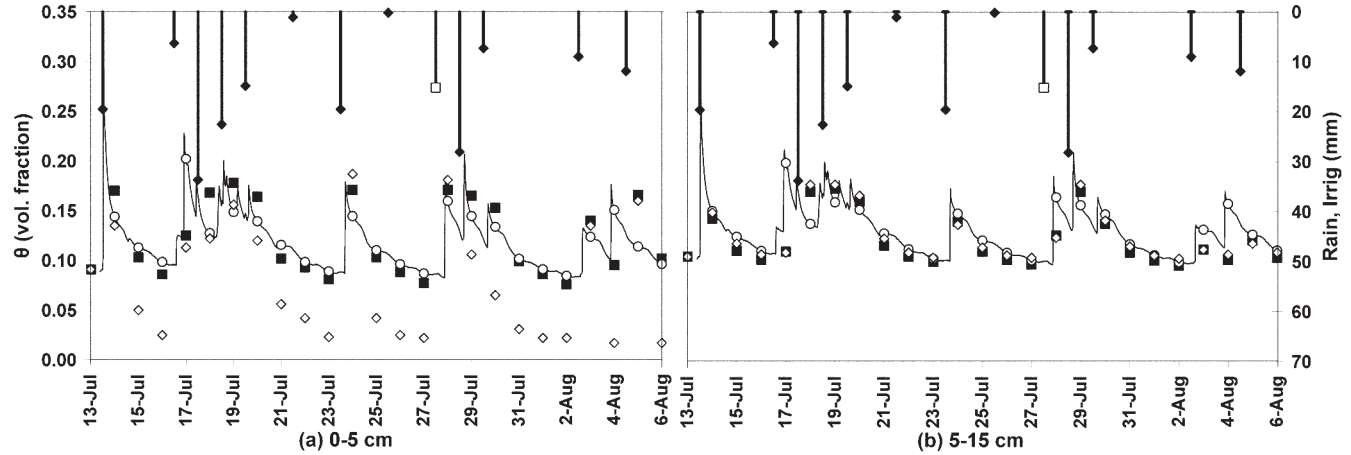


Fig. 4. Simulated and measured volumetric soil water content by date for the Extended Suleiman and Ritchie (ESR) and the Original Ritchie DSSAT (ORD) soil evaporation models during Microwave, Water, and Energy Balance Exp. 3 (MicroWEX-3) in 2004 for (a) 0- to 5-cm and (b) 5- to 15-cm soil layers. Legend as in Fig. 3.

(Table 4), and the pattern of evaporation is quite different (Fig. 7). In these sandy soils, drainage is fast, the soils do not reach saturation, and simulated soil evaporation rates rarely exceed potential evaporation for either model. The ORD model limits soil evaporation to available water in the top 5-cm soil layer and quickly depletes the soil water in this layer, resulting in the chaotic behavior shown in Fig. 7. The ESR model simulated more stable evaporation rates and soil water contents, consistent with the measured data.

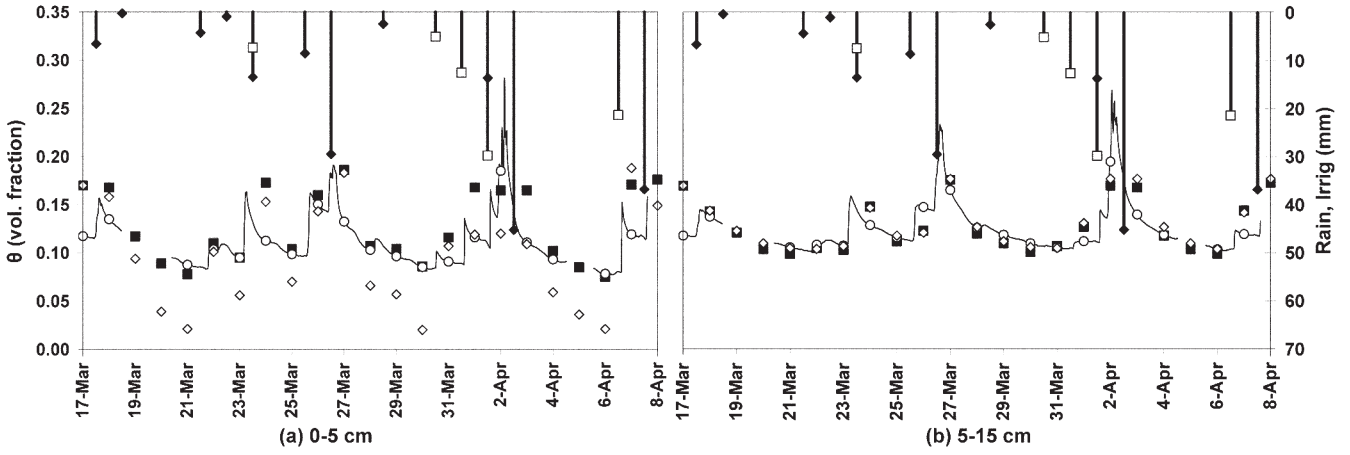


Fig. 5. Simulated and measured volumetric soil water content by date for the Extended Suleiman and Ritchie (ESR) and the Original Ritchie DSSAT (ORD) soil evaporation models during Microwave, Water, and Energy Balance Exp. 4 (MicroWEX-4) in 2005 for (a) 0- to 5-cm and (b) 5- to 15-cm soil layers. Legend as in Fig. 3.

The daily time step of the DSSAT-CSM model produces a potential source of errors as rainfall is assumed to occur at the beginning of the day with 24 h of drainage occurring subsequently. Timing of rainfall and irrigation events can have a major impact on the actual drainage and soil evaporation amounts which occur in a day because the processes can occur so rapidly. The use of pseudo-integration in Eq. [6a] and [6b] was an attempt to minimize this source of error. This error is particularly evident on the days with rainfall or irrigation, where the timing of these additions of water to the system is critical to the simulated end-of-day soil water content. For this reason, the simulated soil water contents are not adequately simulated by either model on wet days.

Graphical Analyses for Column Studies Data

Figure 8 presents simulated versus measured soil evaporation data for the wetter soil column studies in the SR paper using the ESR model. The measured and simulated soil water contents are presented in Fig. 9a for the sandy loam soil and in Fig. 9b for the loam soil. The graphs of cumulative evaporation indicate that the ESR model works well under these conditions. The soil water distribution near the surface was in good agreement with measurements for the sandy loam soil. The loam was biased with too much water removed from the 0- to 5-cm layer, although there was apparent compensation with errors in the opposite direction in the lower layers, resulting in the total evaporation being adequately simulated. The bias in the loam soil indicates that a more accurate assessment of other soils using the ESR approach is needed.

Quantitative Comparisons of Model Errors

Table 5 summarizes the statistical comparisons of the simulated versus measured data for the four MicroWex's for both the ORD and ESR models. The bias, or sum of the residuals, is a measure of how much the simulated results differ from the observed. The positive values for bias shown in Table 5 indicate that the simulations for the ORD model tended to underpredict volumetric soil water content on average. The biases for the ESR model were improved, or decreased in magnitude, although the model tended to overpredict soil water content for the surface layer and underpredict for the 5- to 15-cm layer. The overall bias for the 0- to 5-cm layer for the ESR model was near zero. The magnitudes of biases for the individual experiments were small although the model tended to overpredict soil water content for the top layer.

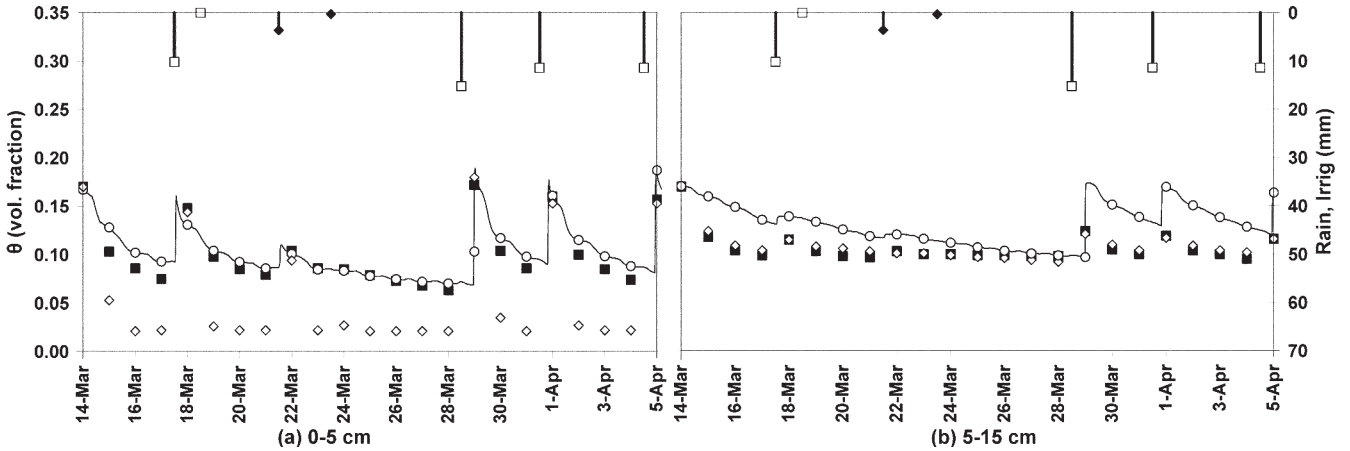


Fig. 6. Simulated and measured volumetric soil water content by date for the Extended Suleiman and Ritchie soil evaporation model (ESR) and the Original Ritchie DSSAT (ORD) soil evaporation models during Microwave, Water, and Energy Balance Exp. 5 (MicroWEX-5) in 2006 for (a) 0- to 5-cm and (b) 5- to 15-cm soil layers. Legend as in Fig. 3.

The mean average error in Table 5 is a measure of the overall magnitude of the errors. The MAE was decreased overall for the surface layer with the ESR model, although the 5- to 15-cm layer showed a slight increase in MAE. All indicators of error demonstrated the improvement in the ESR model's ability to simulate soil water content in the top soil layer. The measures for the grouped data were consistent with that of the entire data set.

The model efficiency for the ESR model is improved over that of the ORD model for the top 5-cm layer, from a −3.003 to +0.051. This efficiency is better than the previous model, but still not a good indication of model adequacy. However, as shown in Fig. 2 through 6, the measured values used in the analysis were the last value of the day (i.e., at midnight), corresponding to the simulated end-of-day values. The inability of a daily model to capture the temporal dynamics of the soil water balance system is expected on days with rainfall or irrigation, when large variations in soil water content occur.

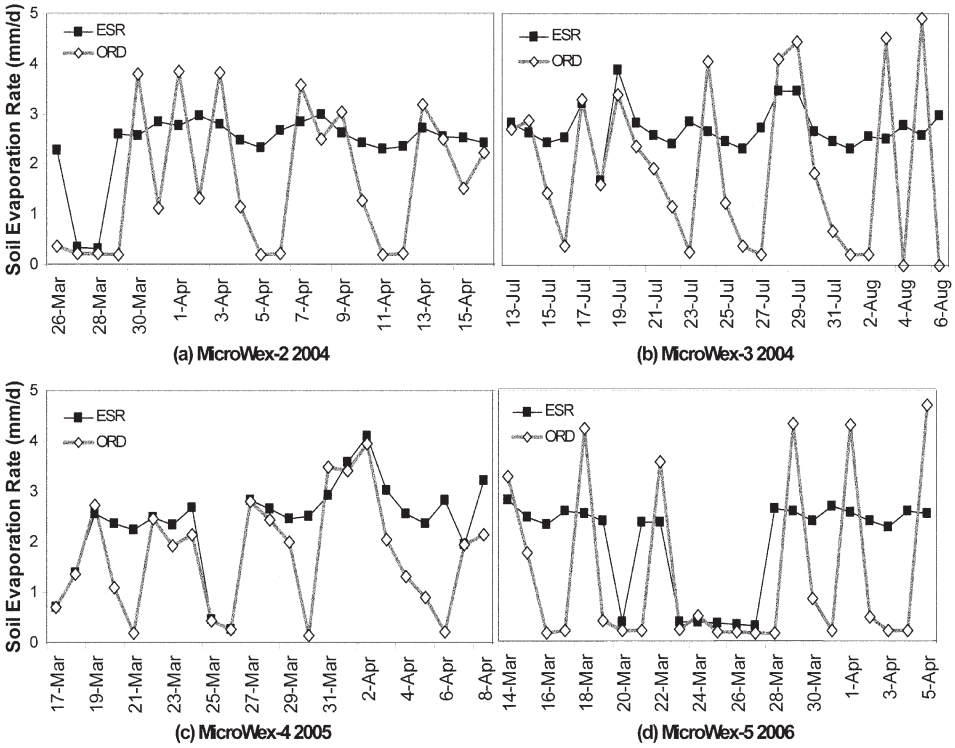


Fig. 7. Simulated daily soil evaporation rates for four Microwave, Water, and Energy Balance Experiments (MicroWEXs), for Original Ritchie DSSAT (ORD), and for Extended Suleiman and Ritchie (ESR) soil evaporation models.

For the 5- to 15-cm layer, the model efficiency was not improved. Model efficiencies for the MicroWex-3 and MicroWex-5 experiments were better than for the overall dataset for the top soil layer; however, MicroWex-2 and MicroWex-4 had negative model efficiencies, indicating poor performance of the model for those experiments. Again, the pattern of simulated soil water content for MicroWex-2 and MicroWex-4 seems to follow measured values closely. The bad model performance is due to timing of intense short duration rainfall events which occurred early in the day.

The Willmott-D index showed improved model performance for the 0- to 5-cm layer, and approximately equal performance for the 5- to 15-cm layer. As with the model efficiency, the MicroWex-3 and MicroWex-5 simulations compared more favorably.

The MSE component analysis, shown in Fig. 10, indicated much better performance for the ESR model compared to the ORD model for the 0- to 5-cm layer. In this analysis, the mean

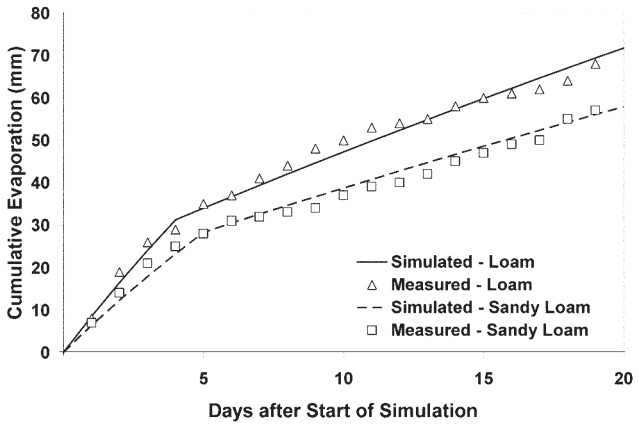


Fig. 8. Simulated and measured cumulative evaporation from top 75 cm of soil for Suleiman and Ritchie (2003; SR) laboratory wet soil column with two soil types.

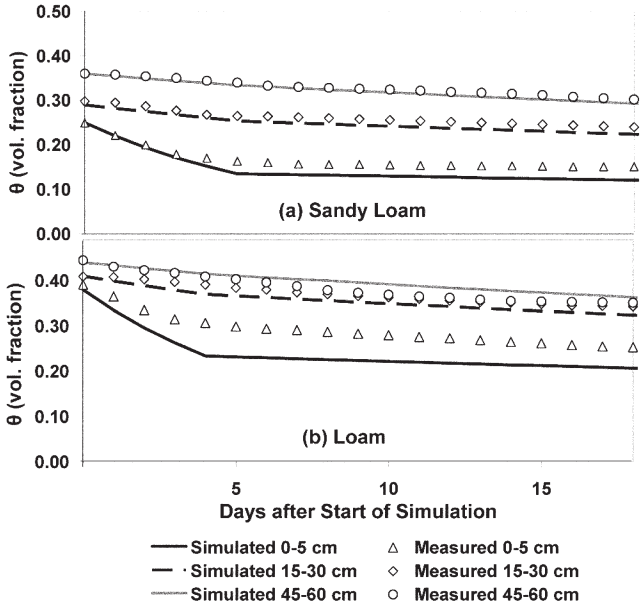


Fig. 9. Simulated and measured soil water content at several depths from (a) sandy loam soil and for (b) loam soil for the Suleiman and Ritchie laboratory wet soil column studies.

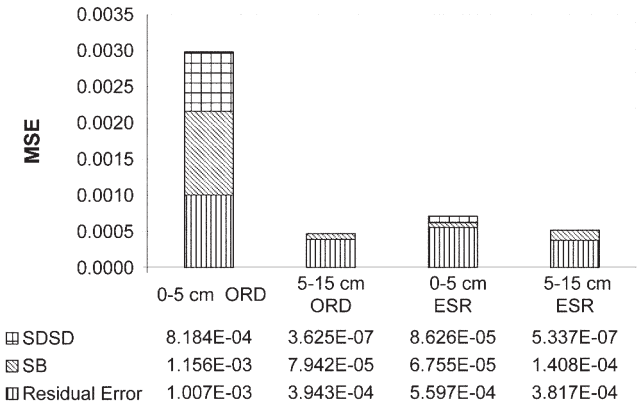


Fig. 10. Components of mean squared error estimate for Original Ritchie DSSAT (ORD) and Extended Suleiman and Ritchie (ESR) soil evaporation models and top two soil layers.

squared error was separated into components of error. The ESR model resulted in decreased squared bias over the ORD model by a factor of 3.6 for the 0- to 5-cm soil layer. The second component of MSE is squared difference of standard deviations. A reduced squared difference of standard deviation indicated that the ESR model provided a better simulation of the magnitude of fluctuations in simulated soil water content as compared to the fluctuations of the measured data. As shown in Fig. 10, the ESR soil evaporation model improved simulation of squared difference of standard deviations by an order of magnitude for the 0- to 5-cm layer. No improvement in model prediction is gained by the ESR model for the 5- to 15-cm soil layer.

CONCLUSIONS

A procedure was developed that extends the SR soil evaporation model to wetter conditions for general use in a dynamic soil–crop model. The procedures added to the SR model retain a daily time step, as used in the DSSAT model system although there are rapid changes in the soil water content following a rain or irrigation. The cumulative soil water changes, after taking drainage into account, estimate the evaporation from bare soil with reasonable accuracy. This ESR procedure has fewer equations and simpler logic than the ORD model previously used in DSSAT-CSM. Despite the simpler logic and input, the ESR soil evaporation algorithm predicted near surface soil evaporation as well as, or better than, the ORD model. The prediction of soil water redistribution near the surface, where large fluxes occur, is a major improvement over the previous method. Thus, these modeling procedures should be highly useful for improving the near surface soil water content calculations for daily incrementing models. Such near surface water dynamics should also be useful in models with tillage and other near surface soil management.

This analysis also revealed that there is no clear distinction between the so-called first and second stages of evaporation. With the presence of water tables or slowly draining upper surface layers, the evaporation rate is higher than estimated with a two-stage system used in the Ritchie (1972) model. Water loss from the soil by evaporation can be higher than a usual second stage estimate but lower than the potential evaporation because of the restriction of water flowing through unsaturated soil near the surface a few days after rainfall or irrigation. The procedures developed herein do not require a first-second stage analysis and are simpler in equations and logic than the previous methods used in DSSAT.

ACKNOWLEDGMENTS

This paper was partially supported by the Office of Natural Resources Management and Office of Agriculture in the Economic Growth, Agriculture, and Trade Bureau of the U.S. Agency for International Development, under terms of Grant No. LAG-G-00-97-00002-00. The observations during the MicroWEXs used for this study were funded by the NSF-Hydrology Program (EAR-0337277) and the NASA New Investigator Program (NASA-NIP-00050655). The authors would like to thank the anonymous reviewers and the editors for providing constructive suggestions to improve the manuscript.

REFERENCES

Black, T.A., W.R. Gardner, and G.W. Thurtell. 1969. The prediction of evaporation, drainage, and soil water storage for a bare soil. Soil Sci. Soc. Am. J. 33:655–660.

Casanova, J.J., T.Y. Lin, M.Y. Jang, K.J. Tien, J. Judge, O. Lanni, and L.W. Miller. 2005. Field observations during the fifth microwave, water, and energy balance experiment (MicroWEX-4). 10 Mar.–14 June 2005. Circ. no. 1482. Center for Remote Sensing, Univ. of Florida, Gainesville. Available online at http://edis.ifas.ufl.edu/AE362, verified 12 Jan. 2009.

Casanova, J.J., F. Yan, J. Judge, K.C. Tien, L.W. Miller, and O. Lanni. 2007. Field observations during the fifth microwave, water, and energy experiment (MicroWEX-5). 9 Mar.–26 May 2006. Circ. no. 1514. Center for Remote Sensing, Univ. of Florida, Gainesville. Available online at http://edis.ifas.ufl.edu/AE407, verified 12 Jan. 2009.

Hoogenboom, G., J.W. Jones, P.W. Wilkens, C.H. Porter, L.A. Hunt, K.J. Boote, U. Singh, O. Uryasev, J.I. Lizaso, A.J. Gijsman, J.W. White, W.D. Batchelor and G.Y. Tsuji. 2009. Decision Support System for Agrotechnology Transfer Version 4.5 [CD-ROM]. Univ. of Hawaii, Honolulu, HI.

Table 5. Summary of statistical model evaluation analyses.

Statistic	ORD Model† All experiments			ESR Model† All experiments			ESR Model (0–5 cm) Individual experiments			
	Complete Data Set (*n* = 178)	0–5 cm (*n* = 89)	5–15 cm (*n* = 89)	Complete Data Set (*n* = 178)	0–5 cm (*n* = 89)	5–15 cm (*n* = 89)	MicroWEX-2 (*n* = 22)	MicroWEX-3 (*n* = 25)	MicroWEX-4 (*n* = 19)	MicroWEX-5 (*n* = 23)
Bias	0.0215	0.0340	0.0089	0.0018	–0.0082	0.0119	–0.0200	–0.0025	–0.0181	0.0051
MAE‡	0.0314	0.0482	0.0145	0.0177	0.0190	0.0164	0.0214	0.0205	0.0216	0.0131
RMSE	0.0417	0.0549	0.0216	0.0248	0.0267	0.0227	0.0294	0.0273	0.0303	0.0193
RRMSE	0.355	0.510	0.170	0.211	0.248	0.179	0.319	0.222	0.274	0.186
ME		–3.003	0.022		0.051	–0.081	–0.673	0.281	–0.442	0.514
D-index		0.604	0.738		0.808	0.725	0.697	0.845	0.768	0.884

† Original Ritchie DSSAT (ORD) and Extended Suleiman and Ritchie (ESR) soil evaporation models.

‡ Mean average error (MAE), root mean square error (RMSE), relative root mean square error (RRMSE), model efficiency (ME), and Willmott index of agreement (D-index).

Jackson, R.D., R.J. Reginato, B.A. Kimball, and F.S. Nakayama. 1974. Diurnal soil-water evaporation: Comparison of measured and calculated soil-water fluxes. Soil Sci. Soc. Am. Proc. 38:861–866.

Judge, J., J.J. Casanova, T.Y. Lin, K.C. Tien, M. Jang, O. Lanni, and L.W. Miller. 2005. Field observations during the second microwave, water, and energy experiment (MicroWEX-2). 17 Mar.–3 June 2004. Circ. no. 1480. Center for Remote Sensing, Univ. of Florida, Gainesville. Available online at http://edis.ifas.ufl.edu/AE360, verified 12 Jan. 2009.

Jones, J.W., G. Hoogenboom, C.H. Porter, K.J. Boote, W.D. Batchelor, L.A. Hunt, P.W. Wilkens, U. Singh, A.J. Gijsman, and J.T. Ritchie. 2003. The DSSAT cropping system model. Eur. J. Agron. 18:235–265.

Kobayashi, K., and M.U. Salam. 2000. Comparing simulated and measured values using mean squared deviation and its components. Agron. J. 92:345–352.

Lascano, R.J., and C.H.M. van Bavel. 1983. Experimental verification of a model to predict soil moisture and temperature profiles. Soil Sci. Soc. Am. J. 47:441–448.

Lascano, R.J., and C.H.M. van Bavel. 1986. Simulation and measurement of evaporation from a bare soil. Soil Sci. Soc. Am. J. 50:1127–1133.

Lin, T., J. Judge, K.C. Tien, J.J. Casanova, M. Jang, O. Lanni, L.W. Miller, and F. Yan. 2005. Field observations during the third microwave water and energy balance experiment (MicroWEX-3). 16 June–21 Dec. 2004. Circ. Number 1481. Center for Remote Sensing, University of Florida, Gainesville, FL. Available online at http://edis.ifas.ufl.edu/AE361, verified 12 Jan. 2009.

Nash, J.E., and J.V. Sutcliffe. 1970. River flow forecasting through conceptual models. Part I. A discussion of principles. J. Hydrol. 10:282–290.

Philip, J.R., and D.A. de Vries. 1957. Moisture movement in porous materials under temperature gradients. Trans. Am. Geophys. Union 38:222–232.

Ritchie, J.T. 1972. Model for predicting evaporation from a row crop with incomplete cover. Water Resour. Res. 8:1204–1213.

Ritchie, J.T. 1998. Soil water balance in plant water stress. p. 41–54. *In* G.Y. Tsuji et al. (ed.) Understanding options for agricultural production. Kluwer Academic Publ., Dordrecht, The Netherlands.

Ritchie, J.T., and B.S. Johnson. 1990. Soil and plant factors affecting evaporation. p. 363–390. *In* Irrigation of agricultural crops. Agron. Monogr. 30. ASA, CSSA, SSSA, Madison, WI.

Ritchie, J.T., J.R. Kiniry, C.A. Jones, and P.T. Dyke. 1986. Model inputs. p. 37–48. *In* C. A. Jones and J.R. Kiniry (ed.) CERES-maize: A simulation model of maize growth and development. Texas A&M Univ. Press, Collecge Station, TX.

Rose, C.W. 1968a. Evaporation from bare soil under high radiation conditions. Trans. Int. Congr. Soil Sci. 1:57–66.

Rose, D.A. 1968b. Water movement in porous materials III. Evaporation of water from soil. Br. J. Appl. Phys. 1:1779–1791.

Suleiman, A.A., and J.T. Ritchie. 2003. Modeling soil water redistribution during second-stage evaporation. Soil Sci. Soc. Am. J. 67:377–386.

Suleiman, A.A., and J.T. Ritchie. 2004. Modification to the DSSAT vertical drainage model for more accurate soil water dynamics estimation. Soil Sci. 169:745–757.

Wallach, D., D. Makowski, and J.W. Jones (ed.). 2006. Working with dynamic crop models: Evaluation, analysis, parameterization, and applications. Elsevier. New York.

Willmott, C.J. 1982. Some comments on the evaluation of model performance. Bull. Am. Meteorol. Soc. 63:1309–1313.

Volume 4
Chapter 12

Management Operations Module

UNDER CONSTRUCTION

Volume 4
Chapter 13

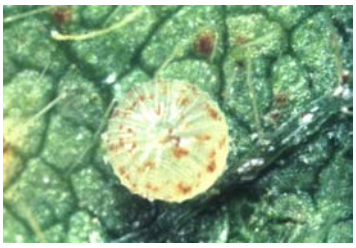
Pest and Disease Damage Module

W.D. Batchelor
J.W. Jones
K.J. Boote
C.H. Porter

Iowa State University, Ames, IA
University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

PEST AND DISEASE DAMAGE MODULE

March 31, 2004



W.D. Batchelor, J.W. Jones, K.J. Boote, C.H. Porter

Iowa State University, Ames, IA
University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Table of Contents

1.0 Introduction **1**

2.0 General Approach **1**

3.0 Theory and Equations **6**

4.0 PEST Module Structure **9**

5.0 Pest Module Main Routine (PEST) **12**
5.1 Run Initialization 12
5.2 Seasonal Initialization 12
5.3 Rate Calculations 12
5.4 Integration 12
5.5 Output 13
5.6 Final 13

6.0 Input and Output Routines **13**
6.1 IPPEST 13
6.2 IPPARM 13
6.3 IPPROG 13
6.4 OPPEST 15

7.0 Linear Interpolation of Daily Damage Rates (LINDM) **17**

8.0 Pest Damage Levels Applied to Coupling Points (PESTCP) **18**
8.1 Seasonal Initialization 18
8.2 Rate Calculations 18

9.0 Assimilative Damage (ASMDM) **23**
9.1 Seasonal Initialization 23
9.2 Integration 23

10.0 Seed and Shell Damage (SEEDDM) **24**
10.1 Seasonal Initialization 24
10.2 Rate Calculations 24
10.3 Integration 26

11.0 Vegetative Damage (VEGDM) **27**
11.1 Seasonal Initialization 27
11.2 Rate Calculations 27
11.3 Integration 30

12.0 Root Damage (ROOTDM) **31**
12.1 Seasonal Initialization 31
12.2 Rate Calculations 31
12.3 Integration 32

13.0 Data file preparation for Pest Damage Routines **33**

14.0 References **47**

LIST OF FIGURES

Figure 1 – PEST Module Structure 10
Figure 2 - Plot of selected pest damage data (refer to Table 8) 34
Figure 3 - Results of Example 1 - Leaf Mass 38
Figure 4 – Results of Example 1 - Seed Mass 38
Figure 5 - Results of Example 2 40
Figure 6 - Results of Example 3 - Leaf and Pod Mass 42

LIST OF TABLES

Table 1 - Coupling points and damage types used to apply damage in the CROPGRO crop models. 2
Table 2 - Typical time series file (FILET) containing pest progress data for six major pests for soybean. 3
Table 3 - A typical pest coefficient file for soybean (FILEP or *.PST file). 4
Table 4 - Order of Processing in PEST Module 9
Table 5 – Interface Variables for PEST Module 11
Table 6 - Summary of Input Routines 12
Table 7 – Output variables for PEST.OUT file 16
Table 8 - Typical time series file (FILET) containing observed pest data for five major pests for soybean. The pest names corresponding to each pest header are: CEW6 - 6th instar corn earworm; SL6 - 6th instar soybean looper; VBC5- 5th instar velvetbean caterpillar; VBC6 - 6th instar velvetbean caterpillar; and PRP – percent reduction in photosynthesis. Each pest and damage is defined in the pest coefficient file (FILEP). 33
Table 9 - A typical pest coefficient file for soybean (FILEP or SBGRO030.PST file) 35
Table 10 – Example 1 - Observed Corn Earworm Population 36
Table 11 - Observed Defoliation – Experiment 2 39
Table 12 – Observed Defoliation and Percent Diseased Leaf Area 41
Table 13 – Example 1 - Time series file for corn earworm damage (UFGA7802.SBT) 43
Table 14 – Example 2 - Time series file for defoliation (UFQU7902.SBT) 44
Table 15 – Example 3 - Time series file for defoliation and percent diseased leaf area (UGA8602.PNT) 45
Table 16 -- Coupling points and associated damage variables 46

Cover photographs: Velvetbean caterpillar life cycle. Courtesy of Dr. Carl S. Barfield, University of Florida.

1.0 Introduction

Failure to account for pest effects in crop models is one of the major limitations to their adoption as farm level management tools. In the IBSNAT (International Benchmark Sites Network for Agricultural Technology Transfer) crop growth models (Uehara and Tsuji, 1993), this problem has limited validation and testing to fields with good pest control. Researchers such as Boote et al. (1989) and Gilbert (1992) have found that data collected on crops experiencing pest damage were not very useful for model validation.

2.0 General Approach

In the CROPGRO model, pest linkages, called coupling points, have been defined for the 23 crop components shown in Table 1. These coupling points are associated with state variables for which damage from various sources can be applied. Damage can be expressed in four ways: (1) daily absolute damage rate, (2) percent observed damage, (3) daily percent damage rate, (4) daily absolute damage rate with pest competition and food preference effects.

Absolute daily damage is useful when damage results from pest populations. If pest populations and feeding rates are known, daily damage to coupling points can be computed in units of mass per unit area.

Percent observed damage typically occurs when observations of plant components are compared between some scientific treatment and control. Defoliation can occur in one treatment, resulting in a percent difference in leaf mass between a treatment and control. In this case, the source of damage may not be known, however, time series measurements describing the percent reduction between treatment and control is known.

Daily percent damage rate is useful when damage can be measured in percent on a daily basis. For instance, in a manual defoliation study, 33% and 66% defoliation may be applied on a particular day during the season. In this case, damage is applied as a percent daily damage. Another example would be the application of a shade cloth that blocks 80% of incoming light, resulting in an 80% reduction in daily assimilate production during the period of shading.

The last damage type is used for insects that compete for preferential feeding sites. If insect demand for the primary food source is too high, some damage is partitioned to secondary food sources.

Table 16 expands on Table 1 by including the associated coupling point damage variables, damage rate variables and the state variables which are affected by each coupling point damage type. As shown in Table 16, the plant damage variables depend on damage type. For example, damage to root length expressed as a daily absolute damage rate, would be represented by coupling point identifier 'RLF' with damage magnitude recorded by variable TRTLF (in subroutine PESTCP). Rates of damage are computed to be WRIDOT and RLVDOT (Daily pest damage to root mass and daily root length damage, respectively) in subroutine ROOTDM. These damage rates are applied to several variables in the model: RLV (root length density), WRDOT (net root growth rate), NRDOT (net

nitrogen addition to roots), NROFF (nitrogen loss from roots), and WTRO (cumulative root losses).

Table 1 - Coupling points and damage types used to apply damage in the CROPGRO crop models.

COUPLING POINT	UNITS[1]	AVAILABLE DAMAGE TYPES[2]	COUPLING POINT ID (PCPID)
leaf area index	m^2 m^{-2}	1, 2, 3, 4	'LAD'
leaf mass	g m^{-2}	1, 2, 3, 4	'LMD'
stem mass	g m^{-2}	1, 2, 3, 4	'SMD'
root mass	g m^{-2}	1, 3, 4	'RMD'
root length	cm root/cm^{-2} soil	1, 3, 4	'RLF'
root length density	cm root/cm^{-3} soil	1, 3, 4	'RLV'
small seed number	no. m^{-2}	1, 3, 4	'SDNS'
large seed number	no. m^{-2}	1, 3, 4	'SDNL'
mature seed number	no. m^{-2}	1, 3, 4	'SDNM'
small seed mass	g m^{-2}	1, 3, 4	'SDMS'
large seed mass	g m^{-2}	1, 3, 4	'SDML'
mature seed mass	g m^{-2}	1, 3, 4	'SDMM'
small shell number	no. m^{-2}	1, 3, 4	'SHNS'
large shell number	no. m^{-2}	1, 3, 4	'SHNL'
mature shell number	no. m^{-2}	1, 3, 4	'SHNM'
small shell mass	g m^{-2}	1, 3, 4	'SHMS'
large shell mass	g m^{-2}	1, 3, 4	'SHML'
mature shell mass	g m^{-2}	1, 3, 4	'SHMM'
whole plant	no m^{-2}	1, 3	'WPD'
assimilate	g m^{-2}	1, 3	'ASM'
necrotic leaf area index	cm^2 cm^{-2}	1, 3	'PDLA'
pod numbers	%/day	3	'PPDN'
vegetative nodes	no. nodes	1,3,4	'VSTG'

[1] per unit ground area
[2] where damage type are: (1) daily absolute damage rate, (2) percent observed damage, (3) daily percent damage rate, (4) daily absolute damage rate with pest competition and food preference effects

A simple approach has been used to allow damage from multiple sources to be applied to any of these defined coupling points. The user must supply the level of pest or damage at different times during the season, and the coupling point and feeding rate that the pest or damage acts upon. This is done by means of two external data files: FILET (e.g., UFGA7801.SBT) and FILEP (e.g., SBGRO030.PST). These files are described briefly below, and in more detail in Appendix B of this document.

An example of a typical FILET for soybean is shown in Table 2. Up to six pest or damage levels can be recorded in the time series file, FILET, as defined by the DSSAT standard input/output data file structure (Jones, J.W. et al. 1994). A header, unique to that pest, defines each pest level entered in FILET. This allows the model to recognize the data in these columns as pest data. The data required for this file are day of year and the measured level of the pest or damage. Linear interpolation is used to compute levels or rates of damage for days between entries in the FILET time series. For pest populations, this would typically be the measured population reported by field scouting. However, pest populations could also be generated from pest models and input into FILET. For damage, such

as defoliation or depodding, dates and levels would be recorded.

Each pest or damage must be defined so that the model can apply damage to the appropriate coupling point. Up to 40 pests can be defined in a crop-specific file called the pest coefficient file (FILEP or *.PST). A typical pest coefficient file for soybean is shown in Table 3. This file contains the pest identification (PID), which matches the header defining the pest level in FILET. Up to six coupling points (PCPID) can be defined for each pest. This provides flexibility for describing pests that damage more than one coupling point throughout the season. For instance, corn earworm consumes soybean leaves early in the season, however, they prefer to consume primordial seed during pod formation. They feed on large seed late in the season. Their feeding rate is different depending on the feeding site and the instar (age) of the larva. Each pest must also be defined by a damage characterization method (PCTID), which defines the units and method by which damage is to be applied. For example, damage for some pests may be computed as an absolute damage per day and damage for other pests may be computed as a percent damage per day.

Table 2 - Typical time series file (FILET) containing pest progress data for six major pests for soybean.

```
*PERFORMANCE DATA - TIME COURSE OF AVERAGE VALUES
EXP: UFBI9101 SB Drew field drought study
*GROWTH

@TRNO  DATE   CEW6   SL6  VBC5  VBC6
     1 91147   0.0   1.5  0.25   0.4
     1 91154   0.5   2.8  0.25   0.6
     1 91161   0.9   3.3   0.1   1.2
     1 91168   1.5   6.4   0.0   1.6
     1 91175   0.2   0.5   0.0   1.8
     1 91181   0.0   1.2   0.0   1.0
     1 91188   -99   1.5   -99   0.2
     1 91195   -99   3.3   -99   0.0
```

The pest name corresponding to each pest header is:

CEW6 - 6th instar corn earworm;
SL6 - 6th instar soybean looper;
VBC5 - 5th instar velvetbean caterpillar;
VBC6 - 6th instar velvetbean caterpillar;

Each pest and damage is defined in the pest coefficient file (*.PST file).

Table 3 - A typical pest coefficient file for soybean (FILEP or *.PST file).

The file consists of the following columns: 1) pest number (LN); 2) pest abbreviation (PID); 3) pest name (PNAME); 4) damage characterization method (PCTID); 5) coupling point identifier (PCPID) defined in Table 1; 6) feeding rate coefficient (PDCF1); 7) units of the feeding rate coefficient (Units), and 8) source for damage rates (Source).

LN	PID[1]	PNAME	PCTID	PCPID	PDCF1	UNITS	SOURCE
01	CEW6	Corn Earworm[2]	4	SDNS	10.0	no. larva^{-1} day^{-1}	Batchelor et al., 1989
				SDNL	2.5	no. larva^{-1} day^{-1}	Szmedra et al., 1988
				LAD	.00505	m^2 larva^{-1} day^{-1}	Szmedra et al., 1988
02	VBC5	5th Instar Velvetbean[3]	1	LAD	0.00081	m^2 larva^{-1} day^{-1}	Reid, 1975
03	VBC6	6th Instar Velvetbean[3]	1	LAD	0.00144	m^2 larva^{-1} day^{-1}	Reid, 1975
04	SL4	4th Inst. Soybean Looper[4]	1	LAD	0.00044	m^2 larva^{-1} day^{-1}	Reid and Greene, 1975
05	SL5	5th Inst. Soybean Looper[4]	1	LAD	0.00071	m^2 larva^{-1} day^{-1}	Reid and Greene, 1975
06	SL6	6th Inst. Soybean Looper[4]	1	LAD	.00124	m^2 larva^{-1} day^{-1}	Reid and Greene, 1975
07	SGSB	Stinkbug[5]	4	SDNS	15.00	no. m^{-2} day^{-1}	Batchelor et al., 1989
				SDNM	5.000	no. m^{-2} day^{-1}	Batchelor et al., 1989
08	FAW	Fall Armyworm	1	LMD	2.00	g larva^{-1} day^{-1}	estimated
09	RTWM	Rootworm	1	RLV	1.000	cm cm^{-2} larva^{-1} day^{-1}	estimated
10	PCLA	Observed % Defoliation	2	LAD	1.000	%	estimated
11	PSTM	Obs.% Stem damage	2	SMD	1.000	%	estimated
12	PDLA	% Diseased Leaf Area	3	PDLA	1.000	%/day	estimated
13	PRP	% Reduction in Photo	3	ASM	1.000	%/day	estimated
14	PLAI	% daily LAI dest.	3	LAD	1.000	%/day	estimated
15	PLM	% daily Leaf Mass	3	LMD	1.000	%/day	estimated
16	PWP	% Whole Plants	3	WPD	1.000	%/day	estimated
17	PSDN	% All Seed Dest.	3	SDNL	1.000	%/day	estimated
				SDNS	1.000	%/day	estimated
				SDNM	1.000	%/day	estimated
18	PSHN	%All Shell Dest.	3	SHNL	1.000	%/day	estimated
				SHNS	1.000	%/day	estimated
				SHNM	1.000	%/day	estimated
19	PPDN	% All Pod Dest.	3	PPDN	1.000	%/day	estimated
20	PRTM	% Root mass dest.	3	RMD	1.000	%/day	estimated
21	HARV	Harvest Date	3	SMD	1.000		
				LMD	1.000		
22	HRVS	Harvest Date	3	VSTG	1.000		

[1] Pest identifier or abbreviation for the pest or damage type.
[2] Corn Earworm (*Heliothis Zea.*)
[3] Velvetbean Caterpillar (*Anticarsia gemmatalis*)
[4] Soybean Looper (*Pseudoplusia includens*)
[5] Southern Green Stinkbug (*Nezara virdula L.*)

A pest coefficient file (*.PST file or FILEP - See Table 3) defines each pest or damage type for a particular crop. Pests are defined by a five-character identification (PID), 20-character name (PNAME), a damage characterization method (PCTID), up to six coupling points (PCPID), and a damage coefficient (PDCF1) associated with each coupling point. The damage characterization method corresponds to the type of pest or damage level in the time series file. The PCTID code corresponds to:

1 - Absolute daily damage rate
2 - Percent observed damage
3 - Daily percent damage rate
4 - Absolute daily damage rate with preference and competition

Refer to Table 1 for a listing of coupling points that are valid for each of these four pest damage characterization methods. The coupling point identification (PCPID) is a character abbreviation (from Table 1) for the feeding or damage site for the pest. Up to six coupling points can be defined for each pest and damage will be applied to each coupling point defined. If the damage method is 4 (absolute damage rate with preference and competition), the order of coupling points for a pest represents the order of food preference. Thus, when this method is used, the model uses the descending order of the coupling points to compute damage based on availability and competition for each subsequent food source. The damage coefficient (PDCF1) corresponds to the rate of damage applied to each coupling point. For instance, if the pest were an insect, PDCF1 would correspond to the feeding rate of the insect on each coupling point. However, if the pest is a damage type, such as percent defoliation, the coefficient would typically be 1.0 and the time series observed defoliation levels would be recorded in the pest progress file. An example pest progress file is shown in Table 3. Also, refer to Appendix B for details on generating input data for pest simulation.

Generic damage, such as defoliation, depodding, or shading, can also be defined using this flexible approach. This is useful when studying plant response to artificially applied damage. In this case, observed damage levels are recorded in FILET. The damage is then defined in the pest coefficient file by selecting the coupling point and damage characterization method. The feeding rate for damage would be 1.0 so that the model would simply apply the level of damage recorded in FILET.

This approach allows many combinations of pest and damage to be applied to the crop model. Damage from up to six pests can be simulated in each experiment. There was no attempt to incorporate the effects of competition among different species of pests, however, intraspecies pest competition has been incorporated. This approach has been used to test the response of the SOYGRO (Jones et al., 1988) and PNUTGRO (Boote et al., 1986) models to different pest damages (Batchelor et al., 1993). This approach has also been incorporated into the CERES-RICE model (Pinnschmidt et al., 1990) in order to test the sensitivity of the model to single and multiple species pest damage.

3.0 Theory and Equations

Mathematically, the application of damage to crop model variables can be expressed by:

$$X_{it} = X^{*}_{it} - \sum_{p=1}^{m} D_{ipt} \quad (1)$$

where:

X_{it} = state (or other model) variable i on day t, after damage has been applied,
X_{it}^{*} = state (or other model) variable i on day t before applying pest damage,
D_{ipt} = amount of damage applied to state (or other model) variable i on day t by pest type p, and
m = total number of pest types.

Note that the D_{ipt} are summed over all pest types to show that more than one pest can damage a given crop component. In this work, a specific set of crop variables has been selected for application of pest damage (Table 1). These variables are referred to as coupling points. Implementing this approach in crop models requires that all coupling points be defined (i=1,2,...,n) and that damage to each coupling point be defined for all pests or damage types (p=1,2,...,m).

The computation of damage by pest p to coupling point i on day t, D_{ipt}, is not a straightforward problem. One complicating factor is that units for reporting pest scouting observations are not uniform. In some cases, pest population dates are reported, while in other cases, damage to the crop is reported in absolute or relative terms. In the simple case when pest population level, N_{pt} pests per unit ground area on day t, is reported for pest p, and this pest damages only one coupling point (i), then D_{ipt} can be computed by:

$$D_{ipt} = (N_{pt})(C_{ip}) \quad (2)$$

In this equation N_{pt} would be the input (observed) pest population on day t for pest p and C_{ip} is a "pest coefficient" that represents the amount of coupling point i damage per individual pest per day. We refer to this method of computing D_{ipt} as the "absolute damage rate" method.

If the availability of the crop component is small relative to the amount being consumed in a day, the damage to variable i may be lower than that computed by Equation (2) because of search/encounter frequencies or because of intraspecies competition for food sites. We have defined a method of applying damage that will allow for competition for sites, based on work by Frazer and Gilbert (1976), and for a pest switching to alternate coupling points for feeding. This method is referred to as "daily absolute damage rate with competition and preference". If this method is used to define damage by a pest, damage rate is computed by:

$$D_{ipt} = X^{*}_{it} * (1 - e^{-N_{pt} C_{ip} / X^{*}_{it}}) \quad (3)$$

When X_{it}^{*} is large compared to the demand for the site (N_{pt} C_{ip}), then D_{ipt} approaches (N_{pt} C_{ip}), as in equation (2). When X_{it}^{*} is small, then equation (3) insures that D_{ipt} remains below both X_{it}^{*} and (N_{pt} C_{ip}). This results in lower consumption than specified by the value of C_{ip}, so we allow damage to a secondary food source, j, when it has been defined for a pest. Demand for the second food

preference j at time t (FE_{jpt}) remaining after feeding on the first food preference is computed by:

$$FE_{jpt} = (N_{pt} C_{ip} - D_{ipt}) * \frac{C_{jp}}{C_{ip}} \quad (4)$$

and the damage rate to the secondary food preference is computed by:

$$D_{jpt} = X^*_{jt} * (1 - e^{-FE_{jpt} / X^*_{jt}}) \quad (5)$$

A third preference for food can be defined similarly.

Consider the following example of corn earworm feeding on soybean. From Table 2, the population (N_{pt}) of corn earworm (CEW6) on day 168 is 1.5 larva m^{-2}. Corn earworm prefer to feed on small seed (C_{ip}) at a rate of 10 seed $larva^{-1}$ d^{-1} (Table 3). They feed on the secondary food source, large seed (C_{jp}), at a rate of 2.5 seed $larva^{-1}$ d^{-1}. At time t, the plant contains 30.0 small seed m^{-2} (X_{it}^*) and 400.0 large seed m^{-2} (X_{jt}^*). Ignoring pest density and competition effects, damage to small seed would be 15 seed $m^{-2}d^{-1}$ by equation (2). However, using equation (3), the effects of density and competition for the primary feeding site results in a computed D_{ipt} of 11.8 small seed $m^{-2}d^{-1}$. The demand not satisfied by the primary food source (15.0 minus 11.8 small seed $m^{-2}d^{-1}$) can be converted into the equivalent of 0.80 large seed $m^{-2}d^{-1}$ of the secondary food source (FE_{jpt}) according to equation (4). The damage to be applied to large seed can then be computed using equation (5). This results in damage of 0.799 large seed $m^{-2}d^{-1}$. The unmet demand can again be computed as 0.001 large seed $m^{-2}d^{-1}$ (0.80 minus 0.799 $m^{-2}d^{-1}$) and the same procedure can be used to convert this damage to equivalent units of the third food source preference for corn earworm, which is leaf mass.

In some cases, scouting reports describe damage as daily percent damage to a coupling point (eg. defoliation due to hail). Thus, a third method of defining damage is when field observations report a percent loss to coupling point i at time t. We refer to this as the "percent damage" method. If this method is selected, D_{ipt} is computed by:

$$D_{ipt} = \frac{R_{ipt}}{100} * X^*_{it} \quad (6)$$

where R_{ipt} is the percent damage to coupling point i caused by pest p (or in this case, reported damage) on day t.

In some experimental plots, it is more convenient to report relative damage throughout the season. In this case, damage to coupling point i is measured by comparison to the same coupling point of a different treatment. We refer to this as the "percent observed damage" method. In this case, the total amount of damage is known for observation dates throughout the season, however, neither the daily percent or absolute damage rate is known. The total amount of damage on a given day would have been accumulated over previous days, and this reporting method does not contain information on when the damage occurred. Therefore, we defined equations to estimate the daily absolute damage that would be required to simulate the percent observed damage for leaf mass, leaf area, and stem mass coupling points. If damage is reported by this method, D_{ipt} is computed by first writing an expression for the percent observed damage (P_{it}) in terms of absolute damage (D_{ipt})

$$P_{it} = 100 * \left(1 - \frac{X_{it}^{*} - D_{ipt}}{Xt_{it} - Xs_{it}} \right) \tag{7}$$

Or, by rearranging:

$$D_{ipt} = X_{it}^{*} - \left(1 - \frac{P_{it}}{100} \right) \left(Xt_{it} - Xs_{it} \right) \tag{8}$$

In these equations, P_{it} is the percent difference in the amount of coupling point i in two treatments at time t (ie. one treatment that has not been damaged and the other that has accumulated damage). The variable Xt_{it} is the cumulative amount of coupling point i produced by the crop model up to time t without any damage or senescence and Xs_{it} is the cumulative senescence of coupling point i up to time t. The variable X_{it}^{*} is the value of the coupling point before damage is applied at time t. In the field, Xt_{it} and Xs_{it} are not known, but in crop models, they can easily be computed so that the observed P_{it} values can be used to compute absolute damage rates as defined by equation (8).

4.0 PEST Module Structure

The PEST Module of the CROPGRO model consists of 11 subroutines, which are organized in a modular format as described by Porter and Jones (1998). The main subroutine, PEST, controls processing and calls each of the other 10 subroutines. PEST is called six times from the plant growth and partitioning module (PLANT) of CROPGRO: once each for 'Run Initialization', 'Seasonal Initialization', 'Rate Calculations', 'Integration', 'Output' and a 'Final' call to close files.

Figure 1 illustrates the components of the PEST module. Table 4 lists the 10 components called by the main pest routine in the order called.

Table 4 - Order of Processing in PEST Module

	RUNINIT	SEASINIT	RATE	INTEGR	OUTPUT	FINAL	Description
IPPEST	X						Reads FILEIO (general information)
IPPARM	X						Reads FILEP (pest data)
IPPROG		X					Reads FILET (time series data)
LINDM			X				Linearly interpolates pest observations
PESTCP		X	X				Computes daily pest damage rates at each coupling point
ASMDM		X		X			Computes assimilative damage
SEEDDM		X	X	X			Computes seed and shell damage
VEGDM		X	X	X			Computes leaf and stem damage
ROOTDM		X	X	X			Computes root damage
OPPEST		X			X	X	Writes daily pest damage to PEST.OUT file

Calling order is top to bottom, left to right; e.g., RUNINIT routines are called first in the order shown, then SEASINIT routines, etc.

Table 5 lists input and output variables for the PEST module. A full listing of variables used in the module and their definitions is included in Appendix C. A listing of the FORTRAN code for the subroutines that comprise the module is included in Appendix D. The following sections detail the function of the individual subroutines in the PEST Module.

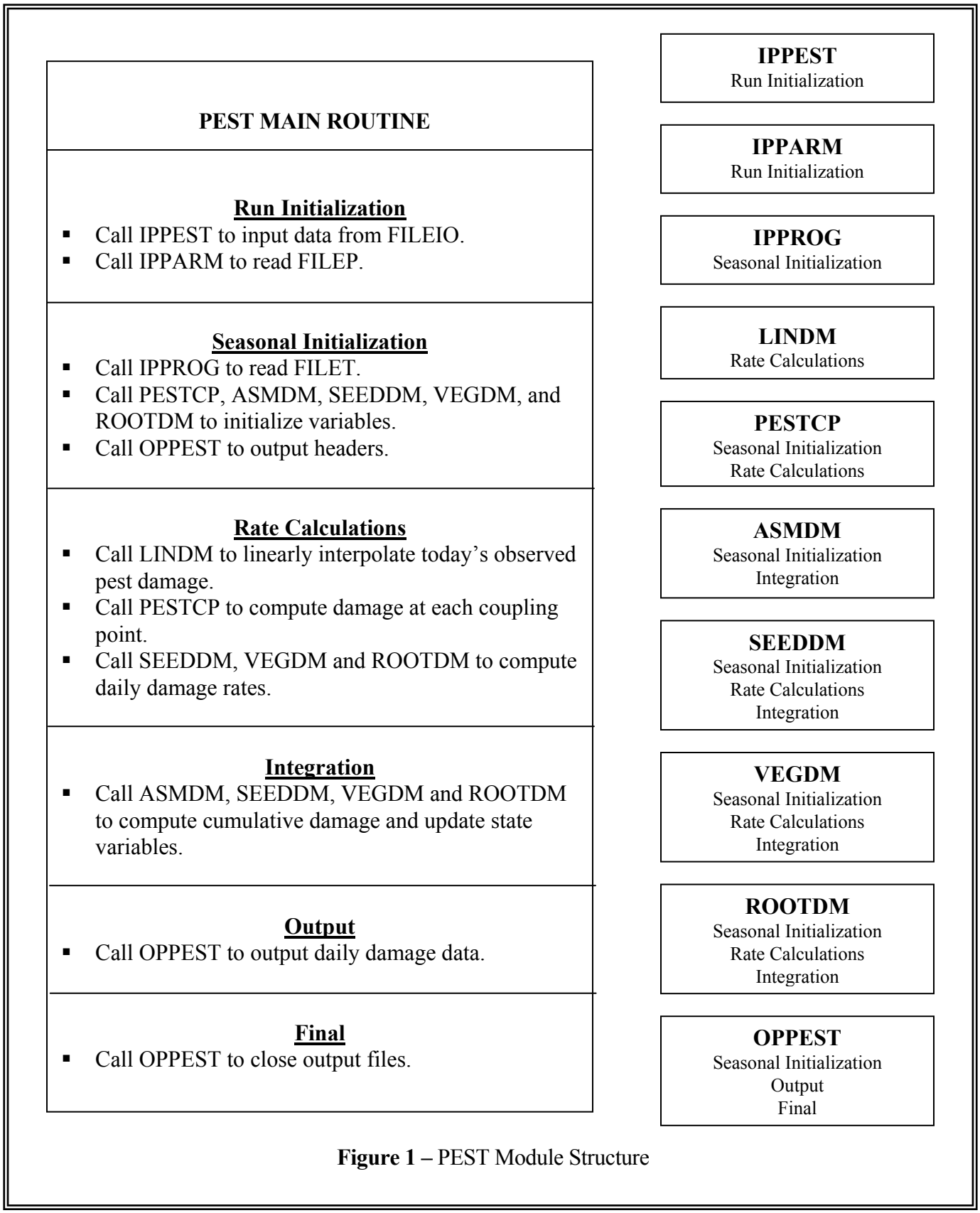


Figure 1 – PEST Module Structure

Table 5 – Interface Variables for PEST Module			
	Variable Name	**Definition**	**Units**
Input Variables	AREALF	Area of leaves (one side) per unit ground area	cm2[leaf] / m2[ground]
	CLW	Cumulative leaf growth	g[leaf]/m2
	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	CSW	Cumulative stem growth	g[stem]/m2
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	LAGSD	Time required between shell growth and seed growth, per cohort	Photo-thermal days
	LNGPEG	Time between start of peg (full flower) and shell formation (for peanuts only). Defines slow growth period.	Photo-thermal days
	NR2	Day when 50% of plants have one fruit (pod or peg)	d
	PGAVL	Total available CH2O available for growth & respiration	g[CH2O] / m2
	PHTIM	Cumulative photothermal time ages of seeds and shells	--
	PLTPOP	Plant population	# plants / m2
	RTWT	Dry mass of root tissue, including C and N	g[root] / m2[ground]
	SLA	Specific leaf area	cm2[leaf] / m2[ground]
	SLDOT	Defoliation due to daily leaf senescence	g/m2/day
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SSDOT	Daily senescence of petioles	g / m2 / d
	STMWT	Dry mass of stem tissue, including C and N	g[stem] / m2[ground
	TOPWT	Total weight of above-ground portion of crop, including pods	g[tissue] / m2
	WLFDOT	Leaf weight losses due to freezing	g[leaf]/m2-d
	WTLF	Dry mass of leaf tissue including C and N	g[leaf] / m2[ground]
	YRPLT	Planting date	YYYYDDD
Input/Output Variables	RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
	SDNO(J)	Number of seeds for cohort J	#/m2
	SHELN(J)	Number of shells for cohort J	#/m2
	SWIDOT	Daily seed mass damage	g/m2/day
	VSTAGE	Number of nodes on main stem of plant	Nodes
	WSHIDT	Weight of shell tissue consumed by pests today	g[shell]/m2-d
	WTSD(J)	Seed mass for cohort J	g/m2
	WTSHE(J)	Shell mass for cohort J	g/m2
Output Variables	ASMDOT	Daily assimilative damage	g[CH2O] /m2 / d
	DISLA	Diseased leaf area	cm2[leaf]/m2[ground]/d
	NPLTD	Number of plants destroyed	#/m2/d
	PPLTD	Percent plants destroyed	%/m2/d
	SDDES(I)	Mass of seeds destroyed in each cohort by pests	g/m2/day
	SDWT	Dry mass of seed tissue, including C and N	g[seed] / m2[ground]
	WLIDOT	Daily pest or freeze damage to leaf mass	g/m2/day
	WRIDOT	Daily pest damage to root mass	g/m2/day
	WSIDOT	Daily pest damage to stem mass	g/m2/day

5.0 Pest Module Main Routine (PEST)

The main component of the PEST module is the PEST.FOR subroutine. No calculations are performed within this subroutine. It is used to control the flow of processing within the module. Detailed descriptions of the functions performed by each subroutine called are presented in subsequent sections of this document.

5.1 Run Initialization

In the 'Run Initialization' section of code, two subroutines are called to input data for use in the PEST module: IPPEST and IPPARM. Table 6 summarizes the data read from the input files in the 'Run Initialization' and 'Seasonal Initialization' sections. These routines are discussed further in Section 6.0.

The input file, FILEIO, is a temporary file generated by input module, MINPT030.EXE, and is called 'DSSAT40.INP' for single season simulations. It is generated from the user-specified FILEX and soils files.

Table 6 - Summary of Input Routines

Input Routine	When Called	Input Filename	Variables Read
IPPEST	RUNINIT	FILEIO (e.g., DSSAT40.INP)	MODEL, FILET, FILEP, PATHPE, IDETD, IDETL, EXPER, ENAME, TRTNO, TITLET, CROP, PHTHRS8, DS, DSOIL
IPPARM	RUNINIT	FILEP (e.g., SBGRO030.PST)	PID, PNAME, PCTID, PCPID, PCPID, PDCF1,
IPPROG	SEASINIT	FILET (e.g., UFGA7801.SBT)	YPL, IDAP

5.2 Seasonal Initialization

Subroutine IPPROG is called to read time series pest damage data from FILET. Five routines (PESTCP, ASMDM, SEEDDM, VEGDM, and ROOTDM) are called to initialize data prior to each season of simulation. Subroutine OPPEST is called to output header information to the output file, PEST.OUT.

5.3 Rate Calculations

First, LINDM is called to linearly interpolate observed pest levels. Then subroutine PESTCP is called to translate pest levels, damage coefficients and damage methods into damage variables to be applied to specific coupling points for today.

Based on which coupling points are active today, i.e., have damage occurring, today, routines SEEDDM, VEGDM and ROOTDM are called to calculate absolute damage rates to be applied to seeds and shells, vegetative tissue, and roots, respectively.

5.4 Integration

Subroutines ASMDM, SEEDDM, VEGDM and ROOTDM are called to update cumulative variables and state variables, where appropriate.

5.5 Output

Subroutine OPPEST is called daily to output damage data to the PEST.OUT file.

5.6 Final

A final call to OPPEST is made to close the output file after all seasonal simulations are complete.

6.0 Input and Output Routines

6.1 IPPEST

Subroutine IPPEST is called once from subroutine PEST during the 'Run Initialization' section to read the following variables from FILEIO:

- FILET – Name of time series file.
- FILEP and PATHPE – Name and path for pest file
- IDETD and IDETL – Switches to generate PEST.OUT file (IDETD) and to generate detailed output (IDETL).
- DS(L) - Cumulative depth in soil layer L, used to calculate DSOIL(L), depth of soil layer L.
- PHTHRS8 - Threshold time that must accumulate in phase 8 for the next stage to occur. Equivalent to PHTHRS(8) in the phenology module.
- MODEL, EXPER, ENAME, TRTNO, TITLET, and CROP – These variables are used in the output file header.

6.2 IPPARM

The subroutine, IPPARM, called once from the PEST subroutine during 'Run Initialization', reads the pest coefficient file for the crop being simulated (FILEP). The file is read using the IGNORE subroutine (found in file READS.FOR), which ignores blank lines and comment lines. When the first line of data for the first pest is found, the pest number (LN), pest identification (PID), pest name (PNAME), pest curve type identification (PCTID), pest coupling point identification (PCPID), and the damage coefficient (PDCF1) are read into subscripted arrays. If coupling points are defined for a pest, PCPID and PDCF1 for the remaining coupling points are read for the first pest. This procedure is repeated until the end of the file is encountered. The variable NPEST represents the number of pests read from FILEP, maximum of 40.

6.3 IPPROG

Each experiment can have a time series file (FILET - see Table 2) which records observed data for the experiment. Up to six different pest or damages can be recorded in FILET. A header is placed at the top of the column that associates the column with a pest identification defined in the pest coefficient file (FILEP - see Table 3). Observed pest or damage levels are then recorded over time in the same manner as other data in FILET.

The subroutine IPPROG, called from the PEST subroutine during 'Seasonal Initialization',

reads the pest progress data for each of the pests defined in the time series file. The subroutine IGNORE2, found in the IPPROG.FOR file, is used to read each line of the file. This subroutine, which is a modified version of the IGNORE utility, allows data to be stored in tiers associated with each treatment of data in FILET. This feature was necessary because of width limitations in ascii data files.

The subroutine searches for the header line and reads the first header into the variable HEAD by:

```
READ(C200,220) HEAD
IF (HEAD.EQ. '      ') GOTO 400
   DO J = 1, NPEST
      DO K = 1,3
         IF(HEAD(K:K+3) .EQ. PID(J)) THEN
               PCOL(COUNT) = COLNUM
               PSTHD(COUNT) = HEAD(K:K+3)
               PNO(COUNT) = J
               COUNT = COUNT + 1
               GO TO 300
         ENDIF
      ENDDO
   ENDDO
```

If the header is blank, the end of the header line has been reached and the subroutine searches for the next header line (if the data were entered in tiers). The outer loop cycles through each of the pests read from FILEP (1 through NPEST). Each six character header is compared to each PID from the pest coefficient file in order to determine which header and associated column of data contains pest progress or damage information. The inner do loop (K=1,3) checks for the pest identification code (PID) to be anywhere within the 6 character header.

The variable COUNT is the number of pest columns found so far in FILET. If the column of data contains pest or damage information, the variable PCOL(COUNT) stores the column number containing the pest progress or damage data. The header is stored in the variable PSTHD(COUNT). Variable PNO(COUNT) is set equal to the current column number and is used to associate the pest number in FILEP with the correct column in FILET for subsequent processing. The variable COUNT is incremented.

At the end of this loop, variable PCN is set equal to the number of pests or damage types which are to be simulated.

Next, the treatment number (TR) and Julian date (YRPST) for observations of pest data are read from FILET.

```
READ(C200,510,END=600) TR,YRPST
```

The observed date is broken down into year and day portions, and adjusted if this is a seasonal simulation (MULTI > 1) and the date is converted into a ‘days after planting’,

DAPP.

```
CALL YR_DOY(YRPST,YR,IPST)
YRPST = (YR + MULTI - 1) * 1000 + IPST
DAPP = MAX(0,TIMDIF(YRPLT,YRPST))
```

The treatment number (TR) is compared to TRTNO, the treatment number that is being simulated by the crop model.

```
IF(TR. EQ. TRTNO) THEN
```

A loop through the pests to be simulated is used to read observed values for this date and assign to the appropriate pest or damage type. Data in the pest columns (PCOL) are stored in the YPL(i,j) array, which can contain up to 100 time series observations for pest or damage i on date j. All negative values are ignored, thus, negative numbers can be entered for missing values. The POBS(i) variable counts the number of observations for pest i. If any two rows of data have the same date, which represents replication in the data, the data in YPL(i,j) are averaged over the number of replications. The observation dates are stored in the IDAP(i,j) array, where i is the subscript for the pest number and j is the subscript for the observation number.

6.4 OPPEST

Subroutine OPPEST is called three times by the PEST module for 'Seasonal Initialization', 'Output' and 'Final'.

6.4.1 Seasonal Initialization

Upon 'Seasonal Initialization', output headings are written to the PEST.OUT file. Table 7 lists the variables that are output by this routine.

6.4.2 Output

The main program determines if today is a day to print daily output. If so, the PLANT module is called, which in turn calls the PEST module, which calls OPPEST to perform this function for daily pest damage computations. These data are written to the PEST.OUT file with the level of detail determined by variable IDETL.

6.4.3 Final

A final call is made to OPPEST after all seasonal simulations are complete in order to close the output file.

Table 7 – Output variables for PEST.OUT file			
Variable Name	**PEST.OUT Heading**	**Definition**	**Units**
DISLA	DLA	Diseased leaf area	cm2/m2/d
DISLAP	DLA%	% Diseased leaf area	%/d
LAIDT	DLAI	Daily change in leaf area index due to pest damage	m2/m2/d
WLIDOT	DLFM	Daily pest or freeze damage to leaf mass	g/m2/day
WSIDOT	DSTM	Daily pest damage to stem mass	g/m2/day
SWIDOT	DSDM	Daily seed mass damage	g/m2/day
SDIDOT	DSD#	Number of seeds destroyed on the current day	#/m2/d
WSHIDT	DSHM	Daily shell mass damage	g/m2/day
SHIDOT	DSH#	Number of shells destroyed on current day	#/m2/d
WRIDOT	DRTM	Daily pest damage to root mass	g/m2/day
RLVDOT	DRLV	Daily root length damage	cm root / cm3 soil
RLFDOT	DRLF	Daily root length flux damage	cm root / cm2 ground
ASMDOT	DASM	Daily assimilative damage	g CH2O/m2/d
PPLTD	DPO%	Percent plants destroyed	%/m2/d
CLAI	CLAI	Cumulative leaf area index destroyed	m2/m2
CLFM	CLFM	Cumulative leaf mass destroyed	g/m2
CSTEM	CSTM	Cumulative stem mass destroyed	g/m2
CSDM	CSDM	Cumulative seed mass destroyed	g/m2
CSDN	CSD#	Cumulative number of seeds destroyed	#/m2
CSHM	CSHM	Cumulative shell mass destroyed	g/m2
CSHN	CSH#	Cumulative number of shells destroyed	#/m2
CRTM	CRTM	Cumulative root mass	g/m2
CRLV	CRLV	Cumulative root length density	cm root / cm3 soil
CRLF	CRLF	Cumulative root length flux	cm root / cm2 ground
CASM	CASM	Cumulative assimilate damage	g CH2O/m2
CPPLTD	CPO%	Cumulative percent of plants destroyed	%

7.0 Linear Interpolation of Daily Damage Rates (LINDM)

The subroutine LINDM, which is called daily from the rate section of the PEST subroutine, linearly interpolates the daily pest or damage level using the YPL(i,j) pest levels occurring on day IDAP(i,j) as read from FILET by Subroutine IPPROG. The subroutine searches the IDAP(i,j) variable for each pest (i) by incrementing down through the data (j) in order to find the first entry where IDAP is greater than the planting date.

```
IF (IDAP(I,ROW(I)) .LT. 0) THEN
   ROW(I) = ROW(I) + 1
   GOTO 450
ENDIF
```

The variable ROW(I) is a counter that increments through the data points and points to the current row of data for each pest progress curve (I). Next, the pest or damage level on the planting date is computed. If the first pest progress data occurs after the current day of simulation, the pest level is set to zero.

```
IF (DAP .LT. IDAP(I,1)) THEN
   PL(I) = 0
```

If the first pest progress data corresponds to the current day of simulation, then the first pest or damage level is set equal to the pest level:

```
ELSEIF (DAP .EQ. IDAP(I,1)) THEN
   PL(I) = YPL(I,ROW(I))
   ROW(I) = ROW(I) + 1
```

Pest levels can begin before planting date and are linearly interpolated as follows:

```
ELSEIF (ROW(I) .GT. 1 .AND. IDAP(I,ROW(I)-1) .LT. 0
  &        .AND. DAP .LE. 1) THEN
   RISE = YPL(I,ROW(I)) - YPL(I,ROW(I)-1)
   RUN = IDAP(I,ROW(I)) - IDAP(I,ROW(I)-1)
   SLOPE = RISE/RUN
   PL(I) = YPL(I,ROW(I)-1) + SLOPE * (DAP - IDAP(I,ROW(I)-1))
```

Pest levels after the planting date are linearly interpolated between observation dates.

```
ELSEIF(DAP .LE. IDAP(I,ROW(I))) THEN
   RISE = YPL(I,ROW(I)) - YPL(I,ROW(I)-1)
   RUN = IDAP(I,ROW(I)) - IDAP(I,ROW(I)-1)
   SLOPE = RISE/RUN
   PL(I) = PL(I) + SLOPE * 1
```

The pest level is set to zero by default until the first observation is encountered. In order to override this default and have a linearly increasing pest or damage level up to the first observed value, a dummy date with damage of 0 can be entered in FILET on the date that pest or damage levels begin.

This provides a starting point for linear interpolation. The pest or damage level after the last observation recorded in FILET is also set to zero by default. In order to have a linearly decreasing pest or damage level, a last dummy observation should be entered on the estimated last date of pest or damage with a level of zero.

8.0 Pest Damage Levels Applied to Coupling Points (PESTCP)

The subroutine PESTCP is called from the 'Seasonal Initialization' and 'Rate Calculations' sections of the PEST subroutine and is responsible for translating pest levels, damage coefficients and damage methods into damage variables to be applied to specific coupling points. The damage is stored in daily damage variables, which are applied to the appropriate coupling points later in the model.

8.1 Seasonal Initialization

In the 'Seasonal Initialization' section of PESTCP, the variables CPPLTD, cumulative percent of whole plants destroyed, and PL, pest level are set to zero for the beginning of simulation.

8.2 Rate Calculations

At the beginning of the 'Rate Calculations' portion of the subroutine, all daily damage variables including leaf, stem, root, seed, shell, whole plant, and photosynthesis variables are initialized to zero. The remainder of the subroutine contains a nested loop. The outer loop (K) cycles through each pest defined in the pest time series file (i.e., pests 1 through PCN). The inner loop (J) cycles through up to six possible coupling points as defined in the pest coefficient file. Variable PNO(K) represents the number of coupling points associated with pest K.

The damage characterization method (PCTID) defined in the pest coefficient file for the first pest is used to control program branching. The variable PCTID defines the damage characterization method used to apply damage computed from the pest or damage level in FILET and the associated damage coefficients in the pest coefficient file, FILEP, and can have valid values of 1 through 4. The subroutine is routed to different sections based on the method used to apply damage. If the damage characterization method is less than one or greater than four, the pest is skipped.

Damage Type 1. If the damage characterization method is 1, the daily absolute damage is computed by multiplying the pest level (PL) by the damage coefficient (PDCF1).

```
DAM = PL(K) * PDCF1(I,J)
```

If PL is a pest level, PDCF1 should be a feeding rate that converts population into damage in units defined for the coupling point in Table 1, such as no./m^2/day or g/m^2/day. If PL contains an actual amount of damage, PDCF1 should convert the units of PL to units

corresponding to the units defined in Table 1 for the coupling point.

This damage rate will be applied to the coupling point later in the code, following the computations for Damage Type 4.

Damage Type 2. If the damage characterization method is 2, damage is applied based on a percent observed damage. As shown in Table 1, this damage type is valid for leaf and stem coupling points: LMD, LAD, and SMD. Damage (DAM) is computed by multiplying pest level (PL) by the damage coefficient (PDCF1). Typically, PL contains observed percent damage and PDCF1 will be 1.0. Thus, the percent damage recorded in PL, which is directly measured in the field, will be applied.

```
      ELSEIF (PCTID(I) .EQ. 2) THEN
         DAM = PL(K) * PDCF1(I,J)
```

The coupling point abbreviation (PCPID) is then compared to each of the available coupling points:

```
       IF (INDEX (PCPID(I,J),'LMD') .GT. 0 .OR.
     &             INDEX(PCPID(I,J),'LAD') .GT. 0) THEN
          PCLMT = PCLMT + DAM
       ENDIF
       IF (INDEX (PCPID(I,J),'SMD') .GT. 0) THEN
          PCSTMD = PCSTMD + DAM
       ENDIF
```

Damage Type 3. If the damage characterization method is 3, damage is applied as a percent daily damage rate. First, damage is computed by multiplying PL by PDCF1. If PL represents a pest population level, PDCF1 should be the feeding rate of each pest in terms of %/pest/day. If PL is a damage level measured in terms of %/day, the value of PDCF1 should be 1.0.

```
      ELSEIF (PCTID(I) .EQ. 3) THEN
         DAM = PL(K) * PDCF1(I,J)
```

Next, the pest coupling point identification is compared to all possible coupling points for this damage type. When a match is found, damage is stored in the appropriate damage variable in the same manner as for absolute daily damage discussed above. For example, if PCPID is LMD, or leaf mass damage, the damage is added to the PLFMD variable (percent leaf mass damage).

```
      IF (INDEX (PCPID(I,J),'LMD') .GT. 0) PLFMD = PLFMD + DAM
```

If the coupling point abbreviation is whole plant damage (WPD), the percent plants destroyed (PPLTD) is computed. Then, damage is applied to the percent damage variables for leaf mass, stem mass, root mass, seed mass, and shell mass as shown below:

```
      IF (INDEX (PCPID(I,J),'WPD') .GT. 0) THEN
```

```
   PPLTD = PPLTD + DAM
   PPLTD = MIN(100.,PPLTD)
   CPPLTD = CPPLTD + PPLTD
   PLFMD = PLFMD + DAM
   PSTMD = PSTMD + DAM
   PRTMD = PRTMD + DAM
   PSDDS = PSDDS + DAM
   PSDDL = PSDDL + DAM
   PSDDM = PSDDM + DAM
   PSHDS = PSHDS + DAM
   PSHDL = PSHDL + DAM
   PSHDM = PSHDM + DAM
ENDIF
```

Damage Type 4. If the damage characterization method is 4, damage is applied using daily absolute damage with preference and competition effects. Damage is applied to each coupling point defined for the pest in the pest coefficient file. Damage is applied sequentially to each coupling point, thus creating a food preference hierarchy. Thus, when defining coupling points for pests that will be used with this preference and competition option, the order of the coupling points in the pest coefficient file will be the order of food preference. For more detail on computation of competition for a food source, refer to equations 3 through 5 in Section 3.0.

The food source availability (FSM) represents the mass or number of available food source for pest I at coupling point J. This is computed for each coupling point by:

```
FSM = 0
IF (INDEX(PCPID(I,J), 'LAD') .GT. 0) FSM = WTLF*SLA / 10000
IF (INDEX(PCPID(I,J), 'LMD') .GT. 0) FSM = WTLF
IF (INDEX(PCPID(I,J), 'SMD') .GT. 0) FSM = STMWT
IF (INDEX(PCPID(I,J), 'RMD') .GT. 0) FSM = RTWT
IF (INDEX(PCPID(I,J), 'SDNS') .GT. 0) FSM = TSDNOS
IF (INDEX(PCPID(I,J), 'SDNL') .GT. 0) FSM = TSDNOL
IF (INDEX(PCPID(I,J), 'SDNM') .GT. 0) FSM = TSDNOM
IF (INDEX(PCPID(I,J), 'SDMS') .GT. 0) FSM = TSDWTS
IF (INDEX(PCPID(I,J), 'SDML') .GT. 0) FSM = TSDWTL
IF (INDEX(PCPID(I,J), 'SDMM') .GT. 0) FSM = TSDWTM
IF (INDEX(PCPID(I,J), 'SHNS') .GT. 0) FSM = TSHNOS
IF (INDEX(PCPID(I,J), 'SHNL') .GT. 0) FSM = TSHNOL
IF (INDEX(PCPID(I,J), 'SHNM') .GT. 0) FSM = TSHNOM
IF (INDEX(PCPID(I,J), 'SHMS') .GT. 0) FSM = TSHWTS
IF (INDEX(PCPID(I,J), 'SHML') .GT. 0) FSM = TSHWTL
IF (INDEX(PCPID(I,J), 'SHMM') .GT. 0) FSM = TSHWTM
```

Next, the potential amount of damage to the coupling point (DDAM) is computed. For the first coupling point (J = 1), this potential damage represents the maximum damage if food source preference and competition did not occur and is equal the pest level times the feeding rate. Damage for the remaining coupling points is assigned the value of the remaining damage (REMDAM) not applied to the previous coupling point.

```
IF (J .EQ. 1) THEN
  DDAM = PL(K) * PDCF1(I,J)
ELSE
  DDAM = REMDAM
ENDIF
```

Next, a temporary variable (PDAM) is computed and used to compute the actual damage (DAM) to be applied to the coupling point (see equation (5) in the theory section). The units of DAM are absolute daily damage (ie. g/m^2/day or no./m^2/day).

```
IF (FSM .GT. 0.0001) THEN
  PDAM = EXP(-DDAM/FSM)
  DAM = FSM * (1 - PDAM)
ENDIF
```

The remaining damage (REMDAM) is the amount of potential damage that is not applied to the coupling point. Thus, it represents unmet demand required by the pest. The amount of unmet demand is converted to equivalent units of the next coupling point mass (see equation 4 of the theory section). The next coupling point will then use REMDAM as the potential amount of damage. If there are no additional coupling points (PCPID(I,J+1) = 'xxxxx'), then REMDAM is set to zero.

```
IF (PCPID(I,J+1) .NE. 'xxxxx') THEN
   REMDAM = (DDAM-DAM)*PDCF1(I,J+1)/PDCF1(I,J)
ELSE
   REMDAM = 0
ENDIF
```

In this way, each subsequent coupling point for damage type 4 will be assigned diminishing damage levels.

Damage Types 1 and 4. Next, for damage types 1 and 4, the coupling point abbreviation (PCPID) is compared to the possible coupling points from Table 1. When a match is found, damage is added to a daily damage variable that is specific to each coupling point and damage characterization method. For instance, if the PCPID is 'LAD ', then the coupling point is leaf area and the damage is added to the variable TLFAD, which will be subtracted from the leaf area in the VEGDM subroutine. Or, if the PCPID is 'SDMS ', the damage is added to the weight of small seed destroyed (WSDDS) damage variable, which is applied in the SEEDDM subroutine.

```
IF (INDEX(PCPID(I,J), 'LAD') .GT. 0) TLFAD = TLFAD + DAM
IF (INDEX(PCPID(I,J), 'SDMS') .GT. 0) WSDDS = WSDDS + DAM
```

Note that a damage variable may contain damage from different pests. Damage is added linearly without regard for interspecies competition for a coupling point.

If the coupling point abbreviation is whole plant damage (WPD), the number of plants destroyed (NPLTD) and the percent plants destroyed (PPLTD) are computed. Then, damage

is applied to the percent damage variables for leaf mass, stem mass, root mass, seed mass, and shell mass as shown below:

```
IF (INDEX(PCPID(I,J), 'WPD') .GT. 0) THEN
   NPLTD = DAM
   IF(PLTPOP.GT.0) THEN
      PPLTD = PPLTD + 100 * NPLTD/PLTPOP
   ENDIF
   PPLTD = MIN(PPLTD,100.0)
   CPPLTD = CPPLTD + PPLTD
   PLFMD = PLFMD + PPLTD
   PSTMD = PSTMD + PPLTD
   PRTMD = PRTMD + PPLTD
   PSDDS = PSDDS + PPLTD
   PSDDL = PSDDL + PPLTD
   PSDDM = PSDDM + PPLTD
   PSHDS = PSHDS + PPLTD
   PSHDL = PSHDL + PPLTD
   PSHDM = PSHDM + PPLTD
ENDIF
```

9.0 Assimilative Damage (ASMDM)

ASMDM computes reduction in daily photosynthesis due to pest damage. The damage variable, ASMDOT, is applied to variable PGAVL (Total CH2O available for growth & respiration) in the PLANT subroutine.

Because values of PGAVL are updated throughout the PLANT routine, the value is different when the PEST module is called for 'Rate Calculations' than when 'Integration' calculations are done. If ASMDM were called for both rate calculations and integration, (with two different values of PGAVL), unpredictable values of PGAVL could result. Therefore, currently, this routine forces both rate and integration calculations to be performed in the 'Integration' section of code.

9.1 Seasonal Initialization

In the 'Seasonal Initialization' portion of the code, variable CASM, cumulative assimilative damage, is set to zero.

9.2 Integration

Assimilative damage, ASMDOT, is set to zero.

Absolute daily assimilative damage, TPSR, is limited to between zero and PGAVL and percent assimilative damage (PPSR) is limited to between zero and 100 percent. TPSR is then calculated to include any percentage reported damage:

```
TPSR = TPSR + PPSR*PGAVL/100.
```

The total daily assimilative damage, ASMDOT, and cumulative damage, CASM, are updated:

```
ASMDOT = ASMDOT + TPSR
CASM = CASM + TPSR
```

10.0 Seed and Shell Damage (SEEDDM)

Seed and shell are maintained on a cohort and total basis. Cohort values for seed and shell mass (WTSD, WTSHE) and number (SDNO, SHELN) are used in PODS and DEMAND, while GROW maintains state variables for total seed and shell mass. The SEEDDM subroutine applies damage to seed and shell mass and number cohorts, and computes the daily seed (SWIDOT) and shell (WSHIDT) mass damage which are subtracted from the seed and shell mass state variables in GROW.

The SEEDDM subroutine is called from the PEST subroutine for 'Seasonal Initialization', 'Rate calculations' and 'Integration'.

10.1 Seasonal Initialization

In the 'Seasonal Initialization' call to the subroutine, cumulative values of seed and shell damage are set to zero for the current season of simulation.

10.2 Rate Calculations

First, daily damage rates are set to zero. The total seed and shell mass and number for each of three size classes are then computed. The photothermal age of each cohort (PAGE) is computed and compared to threshold ages, which define three age classes of pods. Small seed and shell are less than LAGSD thermal days old, and have a seed weight of zero. Large seed and shell are between LAGSD and PHTHRS(8) photothermal days old and mature seed are greater than PHTHRS(8) photothermal days old. The PEST module uses the variable PHTHRS8, which is the scalar equivalent to the PHTHRS(8) array element used in the phenology module.

Next, damage is applied to seed mass. A loop cycling through each cohort is used to check cohort age and apply the appropriate seed damage to each cohort. The code that applies percent small seed mass damage (PSDDS) to cohort NPP is shown below.

```
      IF (PSDDS .GT. 0.0 .AND. WTSD(NPP) .LT. 0.0001 .AND.
     &          SDNO(NPP) .GT. 0.0) THEN
        SDWDES(NPP) = SDWDES(NPP) + PSDDS * WTSD(NPP) / 100.0
        SDNDES(NPP) = SDNDES(NPP) + PSDDS * SDNO(NPP) / 100.0
        IF (PSHDS .LT. 0.001 .AND. NSHDS .LT. 0.001) THEN
          SDDES(NPP) = SDDES(NPP) + PSDDS * SDNO(NPP) / 100.0
        ENDIF
      ENDIF
```

The array elements SDWDES(NPP) and SDNDES(NPP) are the seed mass and number, respectively, destroyed in cohort NPP on the current day. When seeds are destroyed but shells (PSHDS) are not, the number of seeds destroyed (SDDES(NPP)) is accumulated. This is used in the subroutines DEMAND and PODS to limit growth of the remaining seeds in a pod when some seed are destroyed. This procedure is repeated for percent daily loss of large (PSDDL) and mature (PSDDM) seed.

A similar approach is used when seed damage is applied as daily number of small (NSDDS),

large (NSDDL), or mature (NSDDM) seeds. The age of each cohort is computed in order to determine if it contains small, large, or mature seed. Then, the appropriate damage is applied by

```
      IF (NSDDS .GT. 0.0 .AND. WTSD(NPP) .LT. 0.0001 .AND.
     &      SDNO(NPP) .GT. 0.0 .AND. TSDNOS .GT. 0.0) THEN
        NSDDS = MIN(NSDDS,TSDNOS)
        SDWDES(NPP) = SDWDES(NPP) + WTSD(NPP) * NSDDS/TSDNOS
        SDNDES(NPP) = SDNDES(NPP) + SDNO(NPP) * NSDDS/TSDNOS
        IF (PSHDS .LT. 0.001 .AND. NSHDS .LT. 0.001) THEN
          SDDES(NPP) = SDDES(NPP) +  NSDDS * SDNO(NPP)/TSDNOS
        ENDIF
      ENDIF
```

In this case, the variable NSDDS is divided by the total number of small seed (TSDNOS) to convert to a percent of seed (WTSD) damage for cohort NPP. Thus, the number of seed removed from each cohort is weighted by the fraction of total seeds in that cohort. The remainder of the procedure is identical to the case for percent seed mass damage discussed above. A similar procedure is used when damage is specified as number of large or mature seeds.

Damage can also be applied as daily percent of small (WSDDS), large (WSDDL) and mature (WSDDM) seed.

```
      IF (WSDDS .GT. 0.0 .AND. PAGE .LT. LAGSD .AND.
     &      PAGE .GT. LNGPEG .AND. TSDNOS .GT. 0.0) THEN
        WSDDS = MIN(WSDDS,TSDWTS)
        SDWDES(NPP) = SDWDES(NPP) + WSDDS*SDNO(NPP)/TSDNOS
        SDNDES(NPP) = SDNDES(NPP) + WSDDS*SDNO(NPP)*WPS/TSDNOS
        IF (PSHDS .GT. 0.0 .AND. NSHDS .GT. 0.0 .AND.
     &            TSDWTS .GT. 0.0001) THEN
          SDDES(NPP) = SDDES(NPP) + WPS*WSDDS*SDNO(NPP)/TSDNOS
        ENDIF
      ENDIF
```

The mass of seed taken out of each cohort is proportional to the total number of seed (TSDNOS) and the number of seed in the cohort. The procedure and variables computed are similar to the other damages above.

Shell damage can be applied as percent number, or mass of shells destroyed per day. The code used to apply shell damage is similar to that used to apply seed damage, however damage is applied to shell variables.

Total mass and numbers of seed and shell destroyed are limited to the available mass and number.

The variables SWIDOT and SDIDOT are the total seed mass and numbers, respectively, destroyed on the current day of simulation. WSHIDT and SHIDOT are the daily shell mass

and number damage variables. These values represent the sum of losses from all cohorts:

```
SWIDOT = SWIDOT + SDWDES(NPP)
SDIDOT = SDIDOT + SDNDES(NPP)
WSHIDT = WSHIDT + SHWDES(NPP)
SHIDOT = SHIDOT + SHNDES(NPP)
```

10.3 Integration

Cumulative seed and shell losses are updated in the 'Integration' section. The variables CSDM and CSDN are the cumulative seed mass and number destroyed, respectively. Cumulative shell loss variables are CSHM and CSHN.

```
CSDM = CSDM + SWIDOT
CSDN = CSDN + SDIDOT
CSHM = CSHM + WSHIDT
CSHN = CSHN + SHIDOT
```

State variable arrays for seed mass (WTSD), seed number (SDNO), shell mass (WTSHE) and shell number (SHELN) are updated in a loop through cohorts.

```
DO NPP = 1, DAS - NR2 - 1
  PAGE = PHTIM(DAS - NR2) - PHTIM(NPP)
  WTSD(NPP)  = MAX(0., WTSD(NPP)  - SDWDES(NPP))
  SDNO(NPP)  = MAX(0., SDNO(NPP)  - SDNDES(NPP))
  WTSHE(NPP) = MAX(0., WTSHE(NPP) - SHWDES(NPP))
  SHELN(NPP) = MAX(0., SHELN(NPP) - SHNDES(NPP))
ENDDO
```

11.0 Vegetative Damage (VEGDM)

The daily rate of change in leaf mass and stem mass are maintained in the plant growth module (Boote et al., 2000). Two variables, WLIDOT and WSIDOT, which are the daily pest damage to leaf and stem mass ($g/m^2/day$), are subtracted from the leaf and stem state variables in the GROW subroutine of the plant growth module. Leaf area index is computed in GROW from leaf mass and specific leaf area, so damage applied to leaf area is converted and applied as leaf mass damage. The purpose of the VEGDM subroutine is to compute WLIDOT and WSIDOT for all stem and leaf damage identified in the PESTCP subroutine. The subroutine VEGDM is called three times from the PEST subroutine: once each for 'Seasonal Initialization', 'Rate Calculations' and 'Integration'.

11.1 Seasonal Initialization

The 'Seasonal Initialization' portion of the VEGDM code sets cumulative variables to zero for the beginning of the seasonal simulation.

11.2 Rate Calculations

Daily leaf and stem damage variables are set to zero at the beginning of the 'Rate Calculations' section of code.

Stem Damage Calculations. Stem damage can be input as daily percent damage (PSTMD-damage type 3), daily absolute damage (WSTMD – damage types 1 or 4), or observed percent damage (PCSTMD – damage type 2).

Observed percent damage (type 2) is used when the daily damage rate for stem mass reduction is not known. However, stem mass can be compared between control and treatment experiments and the percent reduction can be computed at different times throughout the season. The model maintains the cumulative stem mass (CSW) produced up to the current day of simulation. The stem mass required (DSTEM) to maintain the desired percent reduction (PCSTMD) is computed. This observed damage includes both senescence and pest damage, therefore, the variable SSDOT, daily stem senescence, is subtracted from the current stem weight (STMWT) before calculating today's stem damage (SDAM).

```
DSTEM = CSW * (1.0 - PCSTMD / 100.0)
IF ((STMWT - SSDOT) .GT. DSTEM) THEN
  SDAM = STMWT - SSDOT - DSTEM
ELSE
  SDAM = 0.0
ENDIF
```

If the desired stem mass (DSTEM) is greater than actual stem mass after senescence, (STMWT – SSDOT), then damage is set to zero.

If damage is applied as daily percent reduction (type 3), stem damage, SDAM, is computed based on the percent reduction (PSTMD) and the total stem mass (STMWT).

```
SDAM = PSTMD*STMWT/100.0
```

If stem damage is applied as daily absolute damage (types 1 or 4), SDAM is computed as the minimum of the daily damage rate (WSTMD) and the current stem weight (STMWT).

```
SDAM = MIN(WSTMD, STMWT)
```

The daily stem damage variable (WSIDOT) is updated after each type of damage is computed and is limited to being no less than zero or greater than the total stem weight.

```
WSIDOT = WSIDOT + SDAM
WSIDOT = MAX(0.,WSIDOT)
WSIDOT = MIN(WSIDOT, STMWT)
```

Leaf Damage Calculations. Leaf mass damage can be applied as daily percent damage (PLFMD – damage type 3), daily absolute damage (TLFMD – damage types 1 or 4), or percent observed damage (PCLMA – damage type 2). Leaf area damage can also be specified as daily percent (PLFAD – damage type 3) or daily absolute (TLFAD – damage types 1 or 4). The daily percent reduction is often used when artificial damage is applied to plants in order to measure the plant response to damage. The absolute damage rate is often used for pest damage when feeding rates on leaf mass are known. The percent observed damage is used when leaf mass is compared between treatments, and leaf damage is observed, but the daily amount or source of damage is not known.

For leaf area damage expressed as a daily absolute rate (TLFAD), leaf area index damage (LAIDAM) is set equal to TLFAD, the observed leaf area damage, limited to no greater than actual current leaf area. Leaf area damage is then converted to leaf mass damage (LDAM).

```
LAIDAM = MIN(TLFAD, WTLF * SLA / 10000.0)
LDAM = LAIDAM * 10000.0 / SLA
```

Daily leaf damage and leaf area index damage (cumulative from all sources) is updated by:

```
WLIDOT = WLIDOT + LDAM
LAIDOT = LAIDOT + LAIDAM
```

If leaf mass damage is applied as absolute daily damage (TLFMD), the leaf damage is the minimum of the pest damage level and the total leaf weight. Leaf mass damage is then converted to leaf area damage.

```
LDAM = MIN(TLFMD,WTLF)
LAIDAM = LDAM * SLA / 10000.
```

If daily percent leaf area damage (PLFAD) is applied, leaf damage is the percent observed damage applied to the current leaf weight. Leaf mass damage is then converted from the corresponding leaf area damage.

```
LDAM = WTLF*(PLFAD/100.0)
LAIDAM = LDAM * SLA / 10000.0
```

A similar calculation is done for percent daily leaf mass damage.

If percent observed defoliation (PCLMT) is reported, the potential available leaf mass after the observed leaf damage is applied (PWTLF) is calculated as:

```
PWTLF = CLW * (1.0 - PCLMT / 100.0)
```

Leaf damage (LDAM) is computed to be the difference between this potential leaf mass and actual leaf mass after subtracting today's senesced and frozen leaf tissue.

```
IF ((WTLF - SLDOT - WLFDOT) .GT. PWTLF) THEN
  LDAM = (WTLF - SLDOT - WLFDOT) - PWTLF
ELSE
  LDAM = 0.0
ENDIF
```

Computed leaf damage is then combined with leaf damage from other sources in variables WLIDOT and LAIDOT, as in previous leaf damage computations.

```
IF (LDAM .GT. 0.0) THEN
  WLIDOT = WLIDOT + LDAM
  LAIDAM = LDAM * SLA / 10000.0
  LAIDOT= LAIDOT + LAIDAM
ENDIF
```

Necrotic spots on leaves intercept light but are not capable of photosynthesis. Furthermore, a toxic effect reduces electron transport required for photosynthesis in the surrounding tissue. Disease leaf area can be specified as percent of leaf area (PDLA) or absolute leaf area (TDLA). If diseased leaf area is applied as PDLA percent, the total diseased leaf area (DISLA, m^2/m^2) is computed by

```
IF (PDLA .GT. 0.0) THEN
   DISLA = WTLF * SLA * PDLA/100.
ENDIF
```

If total disease leaf area is applied as TDLA cm^2/cm^2, DISLA is computed by

```
IF (TDLA .GT. 0.0) THEN
   DISLA = DISLA + TDLA
ENDIF
```

Percentage of diseased leaf area (DISLAP) is updated:

```
IF (AREALF .GT. 0.) THEN
   DISLAP = DISLA / AREALF * 100.
ELSE
   DISLAP = 0.0
ENDIF
```

Diseased leaf area is used in the GROW subroutine of the plant growth module to reduce healthy leaf area (AREAH) and leaf area index (XHLAI).

11.3 Integration

At the beginning of the 'Integration' section, the cumulative variables that are used by this routine are updated. Cumulative frozen and senesced leaf tissue (CLSEN and CLFRZ) are maintained here, as they are not used elsewhere in the model. CLFM, CLAI and CSTEM represent cumulative values of leaf mass, leaf area and stem damage.

For V-Stage damage, the state variable VSTAGE, representing number of nodes on the mainstem of the plant, is updated:

```
IF (PVSTGD .GT. 0.0)  THEN
   VSTAGE = VSTAGE * (1.0 - PVSTGD/100.0)
ENDIF

IF (VSTGD .GT. 0.0) THEN
   VSTAGE = VSTAGE - VSTGD
ENDIF
VSTAGE = MAX(0.,VSTAGE)
```

12.0 Root Damage (ROOTDM)

Two primary variables are used to describe roots in the crop model. The variable RTWT, which is a state variable in the GROW subroutine of the plant growth module (Boote et al., 2000), is the total root mass in the soil profile (g/m^2). It can be reduced by the variable WRIDOT, which is the daily pest damage to root mass (g/m^2/day). Root length density (RLV) (cm root/cm^3 soil), maintained in the ROOTS subroutine of the plant growth module, is a primary characteristic used for computing root water uptake in the soil water balance module (Porter et al., 2000). Root length density is maintained for each soil zone defined in the soil profile. The purpose of the subroutine ROOTDM is to compute and apply damage to root mass and root length density. Damage to root length density is applied directly in this subroutine and damage to root mass is applied in GROW through the variable WRIDOT, which is computed in this subroutine.

12.1 Seasonal Initialization

In the 'Seasonal Initialization' section of ROOTDM, cumulative variables are set to zero for the start of simulation. Total depth of the soil profile (CUMDEP) is calculated.

12.2 Rate Calculations

Daily damage variables (RLVDOT, RLFDOT and WRIDOT) are initialized to zero and total root length density (TRLV) is calculated. The distribution of roots by layer is computed and stored in array, RTFRAC.

Root damage can be defined as daily percent or absolute damage to root mass (g/m^2/day), root length density (cm root/cm^3 soil), or root flux (cm root/cm^2 soil). If damage is applied as daily percent root mass (PRTMD), then WRIDOT is summed over all soil layers L:

```
IF (PRTMD .GT. 0) THEN
   DO L = 1, NLAYR
      WRIDOT = WRIDOT + RTWT*(PRTMD/100) * (RLV(L)/TRLV)
      RLVDOT = RLVDOT + RLV(L)*(PRTMD/100)
   ENDDO
ENDIF
```

TRLV is the total root length density over all layers L in the soil profile. The variable RLVDOT is the daily root damage, which is written to the output file PEST.OUT. Damages expressed as percent root length density (PRTLV) and percent root length flux (PRTLF) are calculated similarly.

If damage is applied as daily absolute root mass reduction (WRTMD), then WRIDOT is computed by

```
IF (WRTMD .GT. 0) THEN
   WRTMD = MIN(WRTMD,RTWT)
   DO L = 1, NLAYR
      WRIDOT = WRIDOT + WRTMD * (RLV(L)/TRLV)
      RLVDOT = RLVDOT + RLV(L) * (WRTMD/RTWT)
   ENDDO
```

```
ENDIF
```

When root damage is expressed as total root length density damaged (TRTLV), damage is calculated by:

```
IF(TRTLV .GT. 0) THEN
   TRTLV = MIN(TRTLV,TRLV)
   DO L=1,NLAYR
      WRIDOT = WRIDOT + RTWT * (TRTLV/TRLV) * (RLV(L)/TRLV)
      RLVDOT = RLVDOT + TRTLV * (RLV(L)/TRLV)
   ENDDO
ENDIF
```

The term TRTLV/TRLV is the fraction of root damage, and WRIDOT is summed over all soil layers. Root length density in each soil layer is also reduced by the fraction of root damage. Damage for root length flux is handled in a similar manner.

At the end of the 'Rate Calculations' section, daily root length flux damage is computed to be:

```
RLFDOT = RLVDOT * TDEPTH
```

where TDEPTH is the total depth of the soil profile.

12.3 Integration

In the 'Integration' portion of the model, the root length density (RLV) for each zone in the soil profile is reduced. For damage expressed as percent root mass destroyed (PRTMD), RLV is reduced by:

```
IF (PRTMD .GT. 0) THEN
   DO L = 1, NLAYR
      RLV(L) = RLV(L) - RLV(L)*(PRTMD/100)
      RLV(L) = MAX(0.,RLV(L))
   ENDDO
ENDIF
```

Each of the damage types reduce RLV in each soil layer in a similar manner.

Cumulative damage variables, CRTM (root mass damage), CRLV (root length density), and CRLF (root length flux) are updated at the end of the 'Integration' section:

```
CRTM = CRTM + WRIDOT
CRLV = CRLV + RLVDOT
CRLF = CRLF + RLFDOT
```

13.0 Data file preparation for Pest Damage Routines

Two external data files are used to define the pest linkage. A field-specific record of observed pest or damage levels is contained in the time series file (FILET, e.g., UFGA7801.SBT) associated with each experiment, while FILEP (e.g., SBGRO030.PST) contains information specific to each pest or damage type.

FILET – Time Series File

A typical FILET with recorded pest populations is shown in Table 8. Each column has a header that is unique for each type of pest or damage. The first two columns of data contain the treatment number and observation date. The measured level of pest and/or damage is input in a column under the appropriate pest identification header. This approach for recording observed pest and damage levels follows a typical field scouting record format.

Multiple treatments can be recorded in a single time series file, with subsequent treatment data following the same format. Note that FILET can also contain other types of observed data; for example, leaf area index, seed weight, or canopy height, among others.

Table 8 - Typical time series file (FILET) containing observed pest data for five major pests for soybean. The pest names corresponding to each pest header are: CEW6 - 6th instar corn earworm; SL6 - 6th instar soybean looper; VBC5- 5th instar velvetbean caterpillar; VBC6 - 6th instar velvetbean caterpillar; and PRP – percent reduction in photosynthesis. Each pest and damage is defined in the pest coefficient file (FILEP).

Reference lines for fixed format coding:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
*PERFORMANCE DATA - TIME COURSE OF AVERAGE VALUES

EXP: UFBI9101 SB Drew field drought study

*GROWTH

@TRNO   DATE  CEW6   SL6  VBC5  VBC6   PRP
     1 91147   0.0   1.5  0.25   0.4  30.0
     1 91154   0.5   2.8  0.25   0.6  30.0
     1 91161   0.9   3.3   0.1   1.2   0.0
     1 91168   1.5   6.4   0.0   1.6   -99
     1 91175   0.2   0.5   0.0   1.8   -99
     1 91181   0.0   1.2   0.0   1.0   -99
     1 91188   -99   1.5   -99   0.2   -99
     1 91195   -99   3.3   -99   0.0   -99
```

Figure 2 provides a graphical presentation of the data listed in Table 8 for pest types CEW6, VBC5 and VBC6. Missing data is denoted by a '–99' entry and is not plotted on the chart. The program ignores observations with values less than zero. This allows entries for pest damage to be made selectively on a given date.

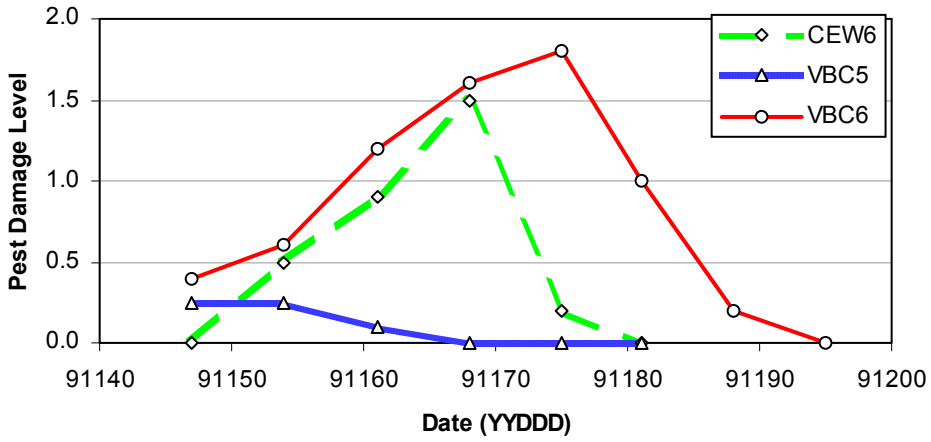


Figure 2 - Plot of selected pest damage data (refer to Table 8)

FILEP – Pest Coefficient File

Observed pest and damages, as defined by the headers in the pest progress file (FILET), must have a corresponding definition in the pest coefficient file (FILEP). This crop-specific pest coefficient file defines individual pests or damage in terms of coupling points and feeding rates. Feeding rate coefficients, which have designated units, provide a means of converting from pest population to damage at coupling points.

A typical pest coefficient file for soybean is shown in Table 9. The first column is the pest number (LN) that is used internally for record keeping in the crop model. The second column contains a pest identifier (PID) that corresponds to the column headers in the observed data time series file (FILET). The next column contains a 20-character description of the pest or damage (PNAME). The fourth column contains the damage characterization method (PCTID). This is a code corresponding to the method used to apply damage to the coupling point variable in the model. These damage methods are:

1) daily absolute damage rate,
2) percent observed damage,
3) daily percent damage rate,
4) daily absolute damage rate with pest competition and food preference effects.

Next, up to 6 coupling point identifiers (PCPID) can be listed for each pest or damage type. Damage is applied to coupling points based on these coupling point identifiers and damage characterization

method (PCPID and PCTID). The variables in the model that are affected by each coupling point and damage method are listed in Table 16.

Table 9 - A typical pest coefficient file for soybean (FILEP or SBGRO030.PST file)
The file consists of the following columns of data: (1) pest number (LN), col. 1-2; (2) pest abbreviation (PID), col. 4-8; (3) pest name (PNAME), col. 10-29; (4) damage characterization method (PCTID), col. 31; (5) coupling point identifier (PCPID) defined in Appendix A, col. 33-37; (6) feeding rate coefficient (PDCF1), col. 39-48; (7) units of the feeding rate coefficient, and (8) source for damage rates.

Reference lines for fixed format coding:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
LN PID   PNAME             PCTID PCPID      PDCF1    UNITS        SOURCE
```

```
*SOYBEAN PEST COEFFICIENTS - CRGRO030 MODEL

01 CEW6  Corn Earworm         4 SDNS  10.0000000    no./larva/d  Batchelor et al., 1989
                                SDNL  2.50000000    no./larva/d  Szmedra et al., 1988
                                LAD   0.00505000    m2/larva/d   Szmedra et al., 1988
02 VBC5  5 Instar Velvetbean  1 LAD   0.00081000    m2/larva/d   Reid, 1975
03 VBC6  6 Instar Velvetbean  1 LAD   0.00144000    m2/larva/d   Reid, 1975
04 SL4   Soybean Looper       1 LAD   0.00044000    m2/larva/d   Reid and Green, 1975
05 SL5   Soybean Looper       1 LAD   0.00071000    m2/larva/d   Reid and Green, 1975
06 SL6   Soybean Looper       1 LAD   0.00124000    m2/larva/d   Reid and Green, 1975
07 SGSB  Stinkbug             4 SDNS  15.0000000    no./m2/d     Batchelor et al., 1989
                                SDNM  5.00000000    no./m2/d     Batchelor et al., 1989
08 FAW   Fall Armyworm        1 LMD   2.00000000    g/larva/day  estimated
09 RTWM  rootworm             1 RLV   1.00000000    cm/cm2/lar/d estimated
10 PCLA  Obs.% defoliation    2 LAD   1.00000000    %            estimated
11 PSTM  Obs.% Stem damage    2 SMD   1.00000000    %            estimated
12 PDLA  % Diseased Leaf Area 3 PDLA  1.00000000    %/day        estimated
13 PRP   % Reduction in Photo 3 ASM   1.00000000    %/day        estimated
14 PLAI  % daily LAI dest.    3 LAD   1.00000000    %/day        estimated
15 PLM   % daily Leaf Mass    3 LMD   1.00000000    %/day        estimated
16 PWP   % Whole Plants       3 WPD   1.00000000    %/day        estimated
17 PSDN  % All Seed Dest.     3 SDNL  1.00000000    %/day        estimated
                                SDNS  1.00000000    %/day        estimated
                                SDNM  1.00000000    %/day        estimated
18 PSHN  % All Shell Dest.    3 SHNL  1.00000000    %/day        estimated
                                SHNS  1.00000000    %/day        estimated
                                SHNM  1.00000000    %/day        estimated
19 PPDN  % All Pod Dest.      3 PPDN  1.00000000    %/day        estimated
20 PRTM  % Root mass dest.    3 RMD   1.00000000    %/day        estimated
21 HARV  Harvest Date           3 SMD   1.00000000    %/day        estimated
                                  LMD   1.00000000    %/day        estimated
22 HRVS Harvest Date            3 VSTG  1.00000000    %/day        estimated
```

The last variable read is a damage coefficient (PDCF1). In CROPGRO, the amount of damage applied each day is the product of the damage coefficient (PDCF1) and the damage level from the

pest observation time series file (FILET). Thus, for pests, this coefficient should be the feeding rate that converts population to daily feeding rate in terms of mass consumption per day. Then, the amount of damage caused by the population can be computed by multiplying pest population (from FILET) by PDCF1. For damage types such as manual defoliation, the damage coefficient should be 1.0, indicating that percent defoliation is recorded in the observation time series file.

The last two columns in Table 9 are included in FILEP only for the user's information and are not read by the model.

Example 1: Observed Corn Earworm in Soybean

In the first example, corn earworm (CEW) damage was applied to both irrigated and non-irrigated treatments in the soybean experiment UFGA7802.SBX. In this experiment, treatments 1 and 2 contained irrigated and non-irrigated Bragg soybean without pest damage. In treatments 3 and 4, pest damage due to observed levels of corn earworm was applied to the irrigated and non-irrigated treatments. In this example, the CEW population has been measured through periodic field scouting, resulting in Table 10.

Table 10 – Example 1 - Observed Corn Earworm Population

Day of year	Population (#/m^2)
222	0.0
229	0.5
236	1.0
243	2.5
250	5.5
257	11.1
264	14.1
271	13.1
278	0.5
285	0.8
292	0.9

In order to apply the pest damage in the model, the following steps were followed:

1) The pest was defined in the pest coefficient file (SBGRO030.PST, see Table 9),
2) The observed population levels were entered in the time series file (UFGA7802.SBT, see Table 13), and
3) The pest damage switch was set to 'Y' in the experiment control file (UFGA7802.SBX).

The corn earworm pest was previously defined in Table 9 and was assigned the identifier of CEW6. Because the damage type is set to PCTID=4, damage is applied preferentially to small seed first at a rate of 10.0 small seed $larva^{-1}$ day^{-1}. If the food demand of the population cannot be satisfied by small seed, some of the damage will be partitioned to large seed at a damage rate of 2.5 seed $larva^{-1}$ day^{-1}. Damage can also be partitioned to leaf mass at a rate of 50.5 cm^2 $larva^{-1}$ day^{-1} if the population is large enough.

A column was established in the time series file UFGA7802.SBT (see Table 13) containing the CEW6 header. The CEW6 header in the time series file must match the pest identifier (PID) for corn earworm in the pest coefficient file (Table 9) in order for the model to link these two tables.

Next, the observed population levels were placed in the time series file. Note that the populations were entered only in treatments 3 and 4. During the simulation, daily populations will be linearly interpolated between observations entered in this file.

Finally, the disease simulation and output options were set to 'Y' in the simulation controls section of the file UFGA7802.SBX as shown below:

```
*SIMULATION CONTROLS
@N GENERAL     NYERS NREPS START SDATE RSEED SNAME.....................
 1 GE              1     1     S 78166  2150 BRAGG, IRRIGATED & NON-IR
@N OPTIONS     WATER NITRO SYMBI PHOSP POTAS DISES  CHEM  TILL
 1 OP              Y     Y     Y     N     N     Y     N     N
@N METHODS     WTHER INCON LIGHT EVAPO INFIL PHOTO HYDRO
 1 ME              M     M     E     R     S     L     R
@N MANAGEMENT  PLANT IRRIG FERTI RESID HARVS
 1 MA              R     R     R     R     M
@N OUTPUTS     FNAME OVVEW SUMRY FROPT GROUT CAOUT WAOUT NIOUT MIOUT DIOUT LONG CHOUT OPOUT
 1 OU              N     Y     Y     3     Y     Y     Y     Y     N     Y     Y     N     N
```

Figure 3 and Figure 4 show the resulting leaf and seed mass for each of the four treatments. The leaf and seed mass was higher in the irrigated than the non-irrigated treatment, demonstrating the model's response to water stress. In the irrigated + CEW treatment, corn earworm began consuming leaf mass before seed growth was initiated. Once seed growth began, CEW damage shifted from leaf mass to seed mass because CEW has a hierarchy of food preference for small seed, large seed, then leaf mass. The CEW population reduced the seed mass by approximately 40% by the end of the season. This was primarily due to seed mass consumption rather than limitations of photosynthesis due to leaf consumption.

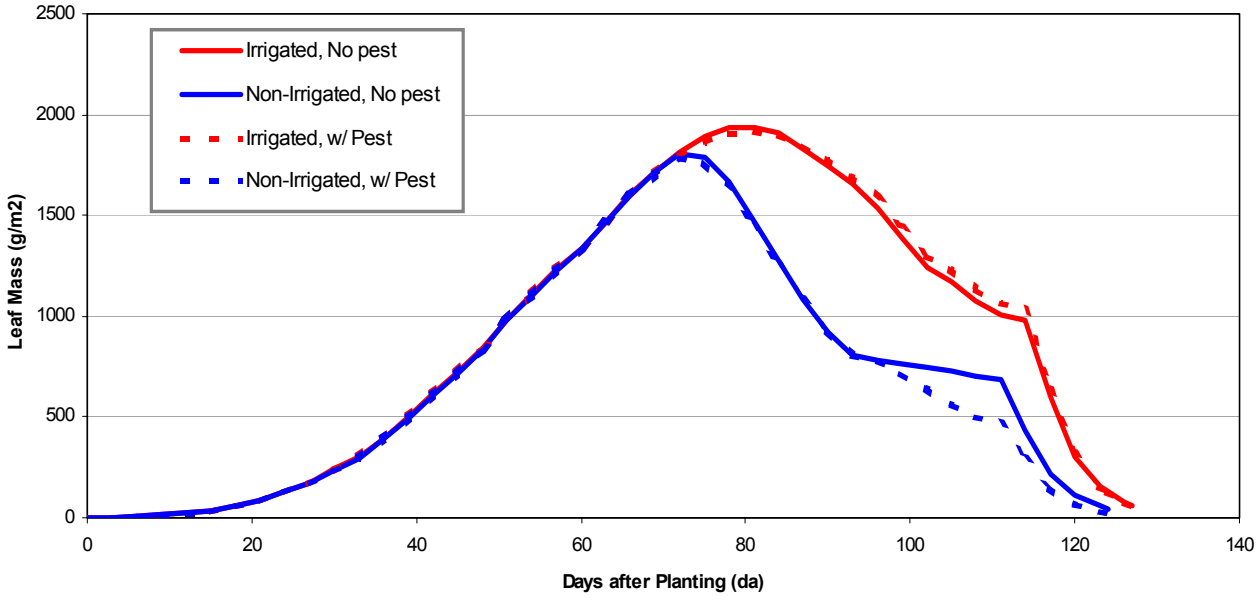


Figure 3 - Results of Example 1 - Leaf Mass

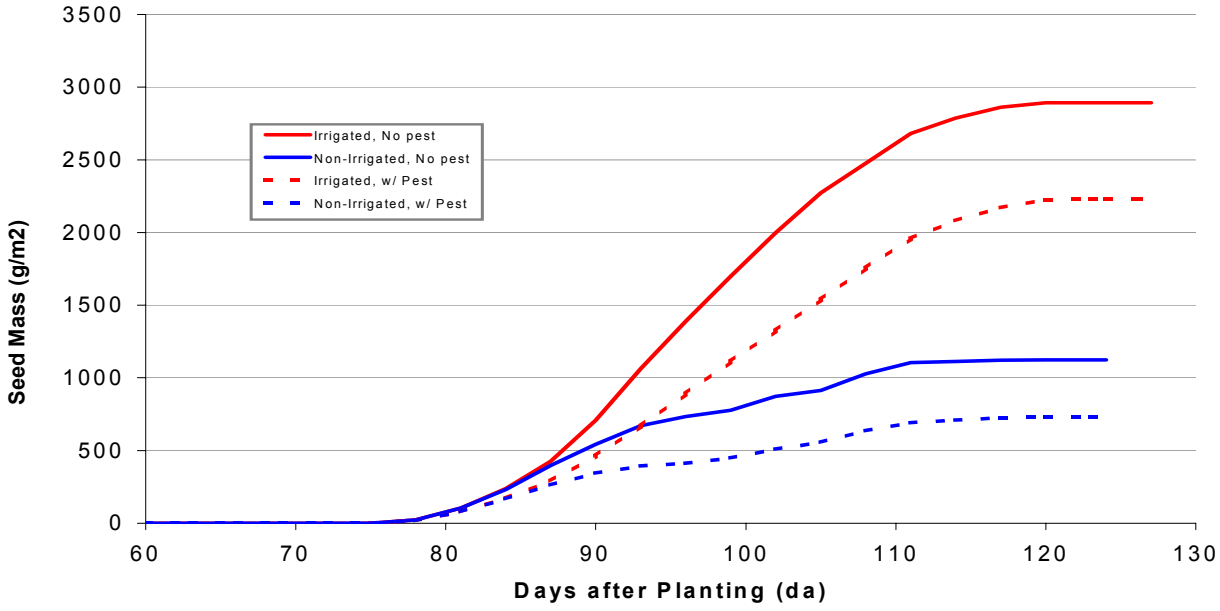


Figure 4 – Results of Example 1 - Seed Mass

Example 2: Observed Defoliation in Soybean (Source of Damage is Unknown)

The experiment UFQU7902 contained two soybean treatments. An insecticide was used in the control treatment and a second treatment was not sprayed. Periodic measurements of leaf area index showed severe defoliation due to an unknown source in the untreated case. In this experiment, data were collected and percent defoliation between the treatments was computed as shown in Table 11.

Table 11 - Observed Defoliation – Experiment 2

DOY	% Defoliation
241	0.0
248	22.6
255	56.1
262	50.0
271	52.5
276	57.1
283	57.9
288	61.7
292	53.3
295	71.4
303	100.0

In this experiment, defoliation can be simulated by specifying the percent observed defoliation over time. To simulate damage, the following steps were taken:

1) Observed defoliation was defined in the pest coefficient file (SBGRO030.PST, see Table 9),
2) Observed defoliation was entered in the time series file UFQU7902.SBT (see Table 14), and
3) Pest damage option was set to 'Y' in the control file UFQU7902.SBX.

The coefficients defining an observed defoliation damage type were added in the pest coefficient file (Table 9). The pest identifier (PID) for percent cumulative leaf area damage is 'PCLA'. The coupling point is leaf area index, indicated by a value of 'LAD' for the coupling point identifier variable, PCPID. The damage characterization method (PCTID) selected is 2, which indicates that damage will be entered in the time series file as observed defoliation. The crop model will compute the daily damage required to obtain this level of defoliation on the observation dates. The damage coefficient is 1.0, so that damage recorded in the time series file can be directly applied to the crop model. This damage coefficient is typically used to convert insect populations to feeding rates. When a damage type, such as this one, is defined, direct measures of damage are typically recorded in the time series file and the damage coefficient is set to 1.0.

A column with the header 'PCLA' was entered in the time series file UFQU7902.SBT (see Table 14). Percent damage on each observation date was entered for treatment 2 as shown. The crop model will linearly interpolate between observations to obtain daily levels of observed damage. Note that an entry of '-99' indicates that an observation was not available and is ignored by the crop model.

Finally, the DISES variable in the *Simulation Controls section of UFQU7902.SBX was set to 'Y' as in the previous example.

In this example, damage was applied to leaf area index. The resulting leaf area index for control and defoliation treatments in shown in Figure 5. The crop model computed the daily leaf mass damage

required to obtain the defoliation observed on the observation dates. The effect of defoliation on seed yield is also shown on Figure 5. Seed yield was diminished due to a reduction of daily photosynthesis.

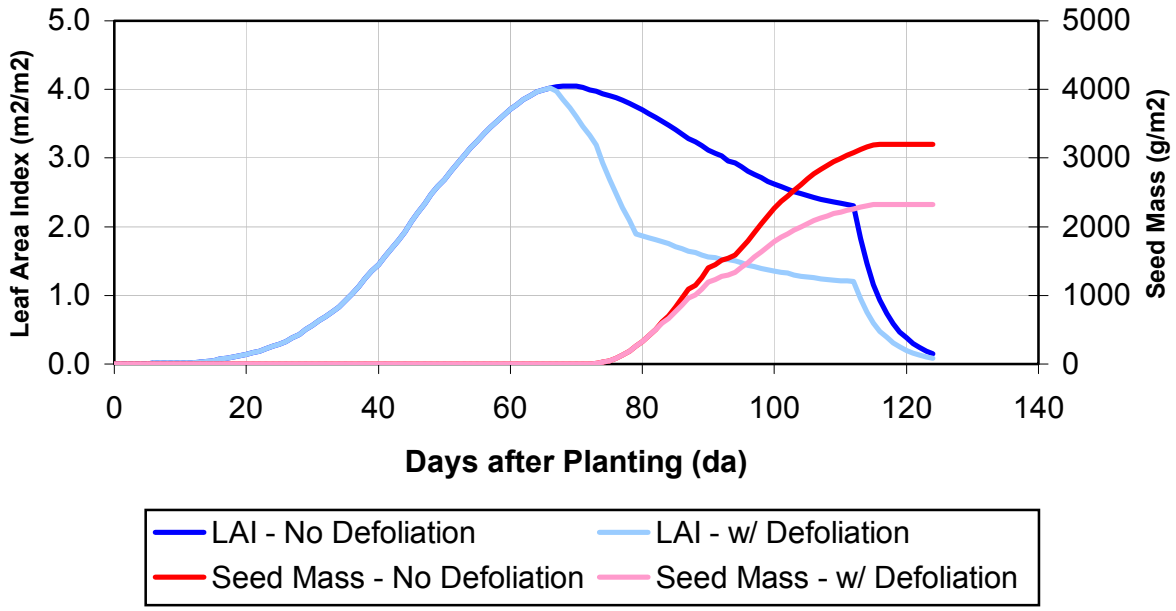


Figure 5 - Results of Example 2

Example 3: Peanut With Leafspot Disease

In another example, an experiment was conducted to determine the effect of late leafspot on peanut pod yield. In the control treatment, a fungicide was sprayed following extension guidelines to obtain good control of leafspot disease throughout the growing season. In the second treatment, no leafspot control measures were taken. Observed defoliation between the control and disease treatments were taken periodically throughout the growing season. Percent diseased leaf area was also measured on each treatment. In this experiment, defoliation and percent diseased leaf area for the disease treatment were recorded as shown in Table 12.

Table 12 – Observed Defoliation and Percent Diseased Leaf Area

DAP	PCLA	PDLA
218	0.0	0.0
230	0.0	0.25
232	4.8	0.28
238	17.3	1.49
246	25.5	41.3
254	26.0	58.8
260	28.0	56.5
266	29.5	75.7
274	31.2	90.1
280	32.0	88.5
288	30.5	99.1
294	29.0	98.6
302	-99	100.0

As in the previous examples, the following steps were taken to link this damage to the crop model:

1) Observed defoliation and percent diseased leaf area were defined in the pest coefficient file (PNGRO030.PST, similar to SBGRO030.PST in Table 9),
2) Observed defoliation and percent diseased leaf area were entered in the time series file UFGA8602.PNT (see Table 15), and
3) Pest damage option was set to 'Y' in the control file UFGA8602.PNX.

An observed defoliation variable, 'PCLA', was defined in the peanut pest coefficient file, PNGRO030.PST, in the same manner as Example 2, above. A diseased leaf area damage type was also defined in the pest coefficient file. The pest identification (PID) for this damage is 'PDLA' and the associated coupling point damage identification, (PCPID) is also 'PDLA', which reduces healthy leaf area in the crop model. This subsequently reduces photosynthetic capacity. The damage characterization method is 3 indicating that diseased leaf area is described as a percent per day.

After the two damage types were defined in the pest coefficient file, the observed levels of damage were input in the time series file UFGA8602.PNT in columns corresponding to the pest identification code for the respective damage types. Daily values of damage will be generated by the model through linearly interpolating between observations. Finally, the DISES variable in the 'Simulation Controls' section of the experiment control file UFGA8602.PNX was set to 'Y' to simulate pest damage.

The leaf mass and pod mass for the two treatments are shown in Figure 6. Disease in the disease treatment began to cause defoliation approximately 70 days after planting. This resulted in slightly lower overall growth rate for pod. Diseased leaf area reduced healthy leaf area available for photosynthesis, resulting in reduced plant growth. Severe pod detachment began approximately 120 days after planting in the disease treatment. This occurred because of peg deterioration resulting from carbon limitations under severe defoliation.

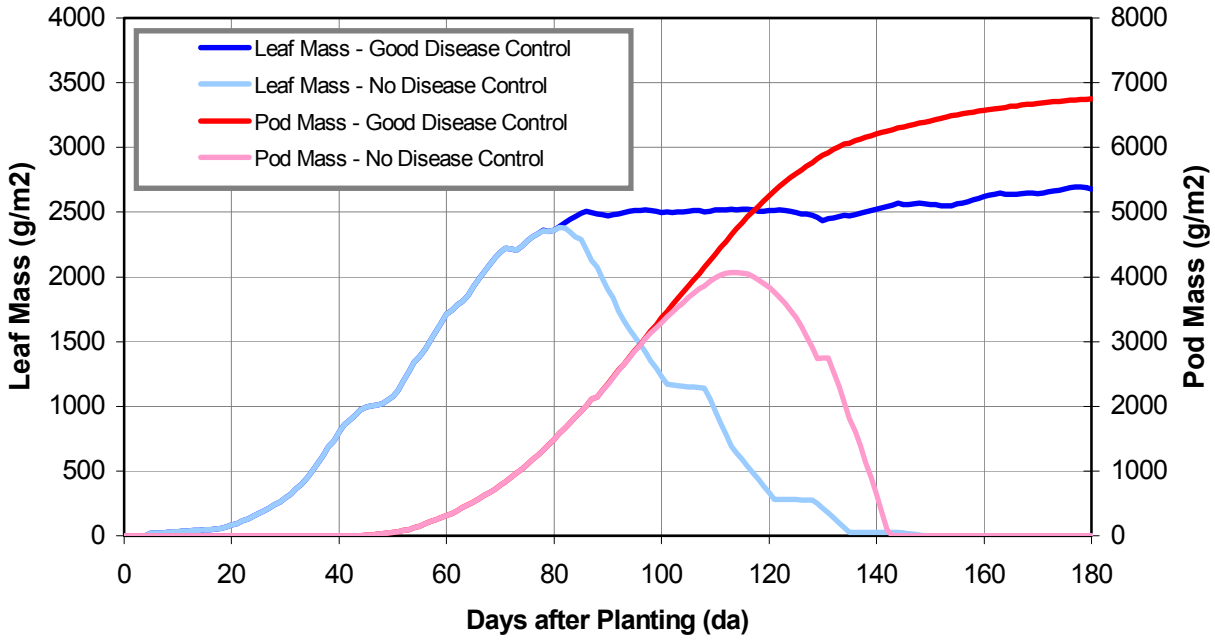


Figure 6 - Results of Example 3 - Leaf and Pod Mass

Table 13 – Example 1 - Time series file for corn earworm damage (UFGA7802.SBT)

```
*EXP.DATA(T): UFGA7802 SB BRAGG, IRR*INSECT DAMAGE
@TRNO  DATE  L#SD  LAID  P#AD  SWAD  GWAD  LWAD  CWAD  PWAD  SHAD  SH%D  SLAD  HIAD  CEW6
    1 78194    .0   .89    .0 178.0    .0 266.0 444.0   0.0   0.0  0.00 334.6   0.0   0.0
    1 78201    .0  1.28    .0 300.0    .0 366.0 667.0   0.0   0.0  0.00 349.7   0.0   0.0
    1 78208    .0  1.91    .0 551.0    .0 656.0  1207   0.0   0.0  0.00 291.2   0.0   0.0
    1 78215    .0  2.86    .0 943.0    .0 843.0  1786   0.0   0.0  0.00 339.3   0.0   0.0
    1 78222    .0  4.17    .0  1561    .0  1187  2748   0.0   0.0  0.00 351.3   0.0   0.0
    1 78229    .0  3.90    .0  1956    .0  1204  3160   0.0   0.0  0.00 323.9   0.0   0.0
    1 78236    .0  4.66    .0  2947    .0  1723  4792 123.0 123.0  0.00 270.5   0.0   0.0
    1 78243    .0  4.47    .0  3144    .0  1772  5224 308.0 308.0  0.00 252.3   0.0   0.0
    1 78250    .0  4.44    .0  3303 182.0  1631  5740 805.0 623.0 22.60 272.2 0.030   0.0
    1 78257    .0  3.99    .0  3326 754.0  1568  6507  1613 859.0 46.75 254.5 0.116   0.0
    1 78264    .0  4.67    .0  3657  1912  1769  8586  3161  1249 60.49 264.0 0.222   0.0
    1 78271    .0  2.83    .0  2732  2223  1180  7144  3232  1009 68.78 239.8 0.311   0.0
    1 78278    .0  2.09    .0  2515  2730 858.0  7136  3763  1033 72.55 243.6 0.383   0.0
    1 78285    .0   .47    .0  1851  2913 170.0  5866  3845 932.0 75.76 276.5 0.497   0.0
    1 78292    .0   .09    .0  2064  3169  34.0  6270  4172  1003 75.96 264.7 0.505   0.0

    2 78194    .0   .75    .0 160.0    .0 244.0 405.0   0.0   0.0  0.00 307.4   0.0   0.0
    2 78201    .0  1.08    .0 251.0    .0 311.0 563.0   0.0   0.0  0.00 347.3   0.0   0.0
    2 78208    .0  1.81    .0 535.0    .0 626.0  1161   0.0   0.0  0.00 289.1   0.0   0.0
    2 78215    .0  3.29    .0  1080    .0 974.0  2054   0.0   0.0  0.00 337.8   0.0   0.0
    2 78222    .0  4.38    .0  1663    .0  1261  2923   0.0   0.0  0.00 347.3   0.0   0.0
    2 78229    .0  4.30    .0  2083    .0  1392  3475   0.0   0.0  0.00 308.9   0.0   0.0
    2 78236    .0  4.21    .0  2556    .0  1517  4184 112.0 112.0  0.00 277.5   0.0   0.0
    2 78243    .0  4.50    .0  2673    .0  1439  4431 319.0 319.0  0.00 312.7   0.0   0.0
    2 78250    .0  2.09    .0  1998  42.0 838.0  3104 268.0 226.0 15.67 249.4 0.014   0.0
    2 78257    .0  3.24    .0  2650 253.0  1220  4530 661.0 408.0 38.28 265.6 0.056   0.0
    2 78264    .0  2.30    .0  2609 471.0 931.0  4440 901.0 430.0 52.28 247.0 0.106   0.0
    2 78271    .0  1.14    .0  2113 775.0 498.0  3812  1200 425.0 64.58 228.9 0.203   0.0
    2 78278    .0   .85    .0  1905 782.0 371.0  3398  1122 340.0 69.70 229.1 0.230   0.0
    2 78285    .0   .53    .0  1922  1149 236.0  3732  1574 425.0 73.00 224.5 0.308   0.0
    2 78292    .0   .07    .0  1590  1206  30.0  3250  1630 424.0 73.99 233.3 0.371   0.0

    3 78194    .0   .89    .0 178.0    .0 266.0 444.0   0.0   0.0  0.00 334.6   0.0   0.0
    3 78201    .0  1.28    .0 300.0    .0 366.0 667.0   0.0   0.0  0.00 349.7   0.0   0.0
    3 78208    .0  1.91    .0 551.0    .0 656.0  1207   0.0   0.0  0.00 291.2   0.0   0.0
    3 78215    .0  2.86    .0 943.0    .0 843.0  1786   0.0   0.0  0.00 339.3   0.0   0.0
    3 78222    .0  4.17    .0  1561    .0  1187  2748   0.0   0.0  0.00 351.3   0.0   0.0
    3 78229    .0  3.90    .0  1956    .0  1204  3160   0.0   0.0  0.00 323.9   0.0   0.5
    3 78236    .0  4.66    .0  2947    .0  1723  4792 123.0 123.0  0.00 270.5   0.0   1.0
    3 78243    .0  4.47    .0  3144    .0  1772  5224 308.0 308.0  0.00 252.3   0.0   2.5
    3 78250    .0  4.44    .0  3303 182.0  1631  5740 805.0 623.0 22.60 272.2 0.030   5.5
    3 78257    .0  3.99    .0  3326 754.0  1568  6507  1613 859.0 46.75 254.5 0.116  11.1
    3 78264    .0  4.67    .0  3657  1912  1769  8586  3161  1249 60.49 264.0 0.222  14.1
    3 78271    .0  2.83    .0  2732  2223  1180  7144  3232  1009 68.78 239.8 0.311  13.1
    3 78278    .0  2.09    .0  2515  2730 858.0  7136  3763  1033 72.55 243.6 0.383   0.5
    3 78285    .0   .47    .0  1851  2913 170.0  5866  3845 932.0 75.76 276.5 0.497   0.8
    3 78292    .0   .09    .0  2064  3169  34.0  6270  4172  1003 75.96 264.7 0.505   0.9
```

Table 13 – Example 1 - Time series file for corn earworm damage (UFGA7802.SBT) **(cont.)**

```
    4 78194    .0   .75    .0 160.0    .0 244.0 405.0   0.0   0.0  0.00 307.4   0.0   0.0
    4 78201    .0  1.08    .0 251.0    .0 311.0 563.0   0.0   0.0  0.00 347.3   0.0   0.0
    4 78208    .0  1.81    .0 535.0    .0 626.0  1161   0.0   0.0  0.00 289.1   0.0   0.0
    4 78215    .0  3.29    .0  1080    .0 974.0  2054   0.0   0.0  0.00 337.8   0.0   0.0
    4 78222    .0  4.38    .0  1663    .0  1261  2923   0.0   0.0  0.00 347.3   0.0   0.0
    4 78229    .0  4.30    .0  2083    .0  1392  3475   0.0   0.0  0.00 308.9   0.0   0.5
    4 78236    .0  4.21    .0  2556    .0  1517  4184 112.0 112.0  0.00 277.5   0.0   1.0
    4 78243    .0  4.50    .0  2673    .0  1439  4431 319.0 319.0  0.00 312.7   0.0   2.5
    4 78250    .0  2.09    .0  1998  42.0 838.0  3104 268.0 226.0 15.67 249.4 0.014   5.5
    4 78257    .0  3.24    .0  2650 253.0  1220  4530 661.0 408.0 38.28 265.6 0.056  11.1
    4 78264    .0  2.30    .0  2609 471.0 931.0  4440 901.0 430.0 52.28 247.0 0.106  14.1
    4 78271    .0  1.14    .0  2113 775.0 498.0  3812  1200 425.0 64.58 228.9 0.203  13.1
    4 78278    .0   .85    .0  1905 782.0 371.0  3398  1122 340.0 69.70 229.1 0.230   0.5
    4 78285    .0   .53    .0  1922  1149 236.0  3732  1574 425.0 73.00 224.5 0.308   0.8
    4 78292    .0   .07    .0  1590  1206  30.0  3250  1630 424.0 73.99 233.3 0.371   0.9
```

Table 14 – Example 2 - Time series file for defoliation (UFQU7902.SBT)

```
*EXP: UFQU7902 SB BRAGG, WELL IRRIGATED  (defoliation)
@TRNO  DATE  L#SD  LAID  P#AD  SWAD  GWAD  LWAD  CWAD  PWAD  SHAD  SH%D  SLAD  PCLA
    1 79234   -99  4.15   -99  1945   -99  1214  3228   -99   -99   -99 323.5   -99
    1 79241   -99  5.48   -99  2650   -99  1659  4455 132.0 132.0   -99 327.6   -99
    1 79248   -99  5.53   -99  3075  60.0  1925  5491 566.0 506.0 10.60 298.8   -99
    1 79255   -99  5.54   -99  3509 303.0  1718  6755  1357  1054 15.24 293.4   -99
    1 79262   -99  5.32   -99  3435 777.0  1866  7395  1988  1211 39.08 269.8   -99
    1 79271   -99  4.04   -99  3362  1319  1481. 7351  2509  1190 48.59 272.4   -99
    1 79276   -99  4.25   -99  3103  1943  1540  7635  3013  1070 64.49 279.8   -99
    1 79283   -99  3.92   -99  3754  2635  1659  9372  4039  1404 65.24 248.3   -99
    1 79288   -99  3.85   -99  3601  3313  1600  9957  4883  1570 67.85 261.2   -99
    1 79292   -99  1.36   -99  2633  3039 592.0  7578  4267  1228 71.22 200.6   -99
    1 79295   -99  0.54   -99  2084  3896 148.0  7666  5406  1510 72.07 243.2   -99
    1 79303   -99  0.47   -99  2388  3541 140.0  7345  4763  1222 74.34 241.0   -99

    2 79234   -99   5.0   -99   -99   -99 1540.   -99   -99   -99   -99   -99   0.0
    2 79241   -99   5.2   -99   -99   -99 1540.   -99 100.0   -99   -99   -99   0.0
    2 79248   -99   4.1   -99   -99   -99 1600.   -99 600.0   -99   -99   -99  22.6
    2 79255   -99   2.5   -99   -99 300.0 1096.   -99  1200   -99   -99   -99  56.1
    2 79262   -99   2.7   -99   -99 700.0 1185.   -99  1800   -99   -99   -99  50.0
    2 79271   -99   1.9   -99   -99  1000  859.   -99  1950   -99   -99   -99  52.5
    2 79276   -99   1.8   -99   -99  1600  948.   -99  2500   -99   -99   -99  57.1
    2 79283   -99   1.5   -99   -99  2100  740.   -99  3200   -99   -99   -99  57.9
    2 79288   -99   1.3   -99   -99  2300  592.   -99  3600   -99   -99   -99  61.7
    2 79292   -99   0.7   -99   -99  2400  266.   -99  3400   -99   -99   -99  53.3
    2 79295   -99   0.2   -99   -99  3000   88.   -99  4200   -99   -99   -99  71.4
    2 79303   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99 100.0
    2 79304   -99   -99   -99   -99  2640   -99   -99  3740   -99   -99   -99   -99
```

Table 15 – Example 3 - Time series file for defoliation and percent diseased leaf area (UGA8602.PNT)

*EXP: UFGA8602 PN RAINFED, FLORUNNER,GOOD DISEASE CONTROL

@TRNO	DATE	L#SD	PCLA	PDLA	LAID	P#AD	SWAD	GWAD	LWAD	CWAD	PWAD	ATPW	DTPW	SHAD	G#AD	HIAD	SH%D
1	86176	7.4	0.0	0.0	0.29	-99	85.	-99	137.	222.	-99	-99	0.0	-99	-99	-99	-99
1	86190	10.6	0.0	0.0	1.14	-99	397.	-99	492.	889.	-99	-99	0.0	-99	-99	-99	-99
1	86204	14.8	0.0	0.0	2.48	19.	1332.	-99	1164.	2509.	13.	13.	0.0	13.	-99	0.005	-99
1	86211	15.6	0.0	0.0	3.16	32.	1714.	-99	1526.	3306.	67.	67.	0.0	67.	-99	0.020	-99
1	86218	17.1	0.0	0.0	5.52	226.	3049.	56.	2248.	5545.	248.	248.	0.0	192.	67.5	0.045	20.7
1	86230	21.0	0.0	0.0	5.64	273.	3411.	264.	2440.	6528.	678.	678.	0.0	414.	197.1	0.104	36.1
1	86232	21.8	0.0	0.0	6.50	461.	4052.	486.	2640.	7683.	990.	990.	0.0	504.	331.6	0.129	48.8
1	86238	23.3	0.0	0.0	7.34	468.	4696.	900.	3153.	9322.	1474.	1474.	0.0	574.	400.8	0.158	60.6
1	86246	25.0	0.0	0.02	7.55	630.	4572.	1449.	3078.	9812.	2162.	2162.	0.0	713.	519.3	0.220	67.0
1	86254	26.5	0.0	0.0	7.75	797.	5073.	2375.	3457.	11958	3429.	3429.	0.0	1054.	684.2	0.287	69.3
1	86260	28.8	0.0	0.08	6.85	583.	4548.	2477.	2966.	10896	3381.	3381.	0.0	904.	724.2	0.310	73.0
1	86266	27.0	0.0	0.01	8.06	860.	5568.	3492.	3640.	13850	4642.	4642.	0.0	1150.	868.8	0.335	75.6
1	86274	29.0	0.0	0.06	6.75	694.	4764.	3343.	2903.	12108	4450.	4440.	10.0	1097.	760.0	0.367	75.2
1	86280	28.0	0.0	0.07	4.53	602.	5076.	4168.	2030.	12512	5416.	5405.	11.0	1237.	833.8	0.432	77.1
1	86288	29.8	0.0	0.13	4.67	794.	5438.	4413.	2223.	13189	5687.	5528.	159.0	1115.	869.9	0.419	79.6
1	86294	31.5	0.0	0.57	4.85	640.	5330.	4289.	2644.	13477	5683.	5503.	180.0	1214.	808.4	0.408	77.9
1	86302	30.5	0.0	0.15	4.57	604.	5667.	4638.	2342.	13748	6190.	5739.	451.0	1101.	881.0	0.417	80.8
2	86176	7.6	0.0	0.0	0.29	-99	85.	-99	137.	222.	-99	-99	-99	-99	-99	-99	-99
2	86190	10.8	0.0	0.0	1.14	-99	397.	-99	492.	889.	-99	-99	-99	-99	-99	-99	-99
2	86204	14.3	0.0	0.0	2.45	12.	1366.	-99	1165.	2538.	7.	7.	0.0	7.	-99	0.003	-99
2	86211	15.6	0.0	0.0	4.03	247.	1940.	-99	1662.	3692.	90.	90.	0.0	90.	-99	0.024	-99
2	86218	16.8	0.0	0.0	5.28	224.	2933.	33.	2253.	5421.	235.	235.	0.0	232.	37.0	0.043	12.8
2	86230	21.7	0.0	0.25	6.18	378.	3767.	436.	2513.	7217.	937.	937.	0.0	501.	281.0	0.130	45.0
2	86232	21.8	4.8	0.28	6.02	365.	3832.	419.	2503.	7229.	895.	895.	0.0	475.	293.6	0.124	46.4
2	86238	23.7	17.3	1.49	5.94	549.	4106.	915.	2608.	8269.	1554.	1554.	0.0	639.	431.8	0.188	59.0
2	86246	25.5	41.3	4.22	4.24	552.	4616.	1438.	1807.	8438.	2015.	2015.	0.0	577.	487.3	0.239	71.0
2	86254	26.0	58.8	5.05	2.34	627.	4872.	1933.	1076.	8486.	2538.	2538.	0.0	605.	523.2	0.299	76.2
2	86260	28.0	56.5	4.77	2.10	564.	4161.	2588.	995.	8568.	3412.	3412.	0.0	824.	646.7	0.398	75.8
2	86266	29.5	75.7	4.15	1.75	786.	5448.	2973.	886.	10224	3890.	3890.	0.0	917.	705.4	0.380	76.5
2	86274	31.2	90.1	4.88	0.52	454.	3732.	2921.	270.	7664.	3718.	3662.	56.0	741.	588.1	0.478	79.6
2	86280	32.0	88.5	9.17	0.40	706.	3804.	3477.	234.	8355.	4409.	4316.	93.0	839.	741.2	0.516	80.7
2	86288	30.5	99.1	7.89	0.04	394.	3473.	2300.	19.	6258.	3876.	2766.	1110.	466.	441.0	0.442	83.2
2	86294	29.0	98.6	0.0	0.04	342.	3241.	2195.	36.	5926.	4439.	2649.	1790.	454.	404.5	0.447	82.8
2	86302	-99	100.0	0.0	-99	17.	3087.	111.	-99	3223.	2538.	136.	2402.	25.	22.9	0.042	62.0

Table 16 -- Coupling points and associated damage variables

Coupling Point	Coupling Point ID (PCPID) from FILEP	Coupling Point Damage Variables by Damage Type (calculated in PESTCP)				Daily Damage Rate Variables	State Variable(s) Affected (Subroutine which applies damage to state variable)
		(1) Daily absolute damage rate	(2) Percent observed damage	(3) Daily percent damage rate	(4) Daily absolute damage rate with competition		
leaf area index	LAD	TLFAD	PCLMT	PLFAD	TLFAD	WLIDOT, LAIDOT	WTLF, WDOT (GROW)
leaf mass	LMD	TLFMD	PCLMT	PLFMD	TLFMD		
stem mass	SMD	WSTMD	PCSTMD	PSTMD	WSTMD	WSIDOT	STMWT, WDOT (GROW)
root mass	RMD	WRTMD	--	PRTMD	WRTMD	WRIDOT, RLVDOT	RTWT, WDOT (GROW) RLV(L) (ROOTDM)
root length	RLF	TRTLF	--	PRTLV	TRTLF		
root length volume	RLV	TRTLV	--	PRTLF	TRTLV		
small seed number	SDNS	NSDDS	--	PSDDS	NSDDS	SWIDOT, SDIDOT, SDWDES(J), SDNDES(J)	SDWT, WDOT (GROW) SEEDNO (GROW, PODS) WTSD(J) (SEEDDM) SDNO(J) (SEEDDM)
small seed mass	SDMS	WSDDS	--	PSDDS	WSDDS		
large seed number	SDNL	NSDDL	--	PSDDL	NSDDL		
large seed mass	SDML	WSDDL	--	PSDDL	WSDDL		
mature seed number	SDNM	NSDDM	--	PSDDM	NSDDM		
mature seed mass	SDMM	WSDDM	--	PSDDM	WSDDM		
small shell number	SHNS	NSHDS	--	PSHDS	NSHDS	WSHIDT, SHIDOT, SHWDES(J), SHNDES(J)	SHELWT, WDOT (GROW) PODNO (PODS) WTSHE(J) (SEEDDM) SHELN(J) (SEEDDM)
small shell mass	SHMS	WSHDS	--	PSHDS	WSHDS		
large shell number	SHNL	NSHDL	--	PSHDL	NSHDL		
large shell mass	SHML	WSHDL	--	PSHDS	WSHDL		
mature shell number	SHNM	NSHDM	--	PSHDM	NSHDM		
mature shell mass	SHMM	WSHDM	--	PSHDM	WSHDM		
pod numbers	PPDN	--	--	PSDDS, PSDDL, PSDDM, PSHDL, PSHDS, PSHDM	--	All seed and shell variables as listed above	All seed and shell variables as listed above
whole plant	WPD	NPLTD, PPLTD, CPPLTD, PLFMD, PSTMD, PRTMD, PSDDS, PSDDL, PSDDM, PSHDS, PSHDL, PSHDM	--	PPLTD, CPPLTD, PLFMD, PSTMD, PRTMD, PSDDS, PSDDL, PSDDM, PSHDS, PSHDL, PSHDM	NPLTD, PPLTD, CPPLTD, PLFMD, PSTMD, PRTMD, PSDDS, PSDDL, PSDDM, PSHDS, PSHDL, PSHDM	All leaf, stem, root, seed and shell variables as listed above	PLTPOP (GROW) and all leaf, stem, root, seed and shell variables as listed above
assimilate	ASM	TPSR	--	PPSR	TPSR	ASMDOT	PGAVL (PLANT)
necrotic leaf area index	PDLA	TDLA	--	PDLA	TDLA	DISLA, DISLAP	XHLAI (GROW)
vegetative nodes	VSTG	VSTGD	--	PVSTGD	VSTGD	--	VSTAGE (VEGDM)

14.0 References

Batchelor, W.D., J.W. Jones, K.J. Boote, and H. Pinnschmidt. 1993. Extending the use of crop models to study pest damage. Transactions of the ASAE. Vol 36(2), pp 551-558.

Boote, K.J., J.W. Jones, G. Hoogenboom, W.D. Batchelor, C.H. Porter. 2000. Plant Growth and Partitioning Module in DSSAT v4.0 Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 2000-1204. University of Florida, Gainesville, Florida.

Boote, K.J., J.W. Jones, J.W. Mishoe, and G.G. Wilkerson. 1986. Modeling growth and yield of groundnut. Agrometeorology of groundnut. In Proceedings of an International Symposium, Aug 21-25, 1985, ICRISAT Sahelian Center, Niamey, Niger.

Frazer, B. D., and N. Gilbert. 1976. Coccinellids and aphids: a quantitative study of the impact of adult ladybirds (Coleoptera: Coccinellidae) preying on field populations of pea aphids (Homoptera: Aphididae). J. Entomol. Soc. B.C. 73: 34-56.

Gilbert, R. 1992. On-farm testing of the PNUTGRO crop model in Florida. MS Thesis, Agronomy Dept., University of Florida, Gainesville, Fl. 32611.

Jones, J.W., K.J. Boote, S.S. Jagtap, G. Hoogenboom, and G.G. Wilkerson. 1987. SOYGRO 5.4. Soybean crop growth simulation model user's guide. Florida Agricultural Experiment Station Journal no. 8304, University of Florida, Gainesville, Fl. 32611. 75 pp.

Jones, J.W., L.A. Hunt, G. Hoogenboom, D.C. Godwin, U. Singh, G.Y. Tsuji, N.B. Pickering, P.K. Thornton, W.T. Bowen, K.J. Boote and J.T. Ritchie. 1994. Input and output files, Volume 2-1 in DSSAT v.3, ed. G.Y. Tsuji, G. Uehara and S. Balas. IBSNAT. University of Hawaii.

Pinnschmidt, H., Teng, P.S. and Yuen, J.E. 1990. Pest effects on crop growth and yield. In Proceedings of the workshop on modeling pest-crop interactions, edited by Teng, P. and Yuen, H. University of Hawaii, Monoa, Honolulu, Hawaii. Jan 7-10. pp 26-29.

Porter, C.H., J.W. Jones. 1998. Modular Structure in DSSAT v4.0 Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Porter, C.H., J.W. Jones, G. Hoogenboom, P. W. Wilkens, J.T. Ritchie, N.B. Pickering, K. Boote, B. Baer. 2000. Soil Water Balance Module in DSSAT v4.0 Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 2000-1202. University of Florida, Gainesville, Florida.

Uehara, G., and G.Y. Tsuji. 1993. The IBSNAT project. In Systems Approaches for Agricultural Development, F.W.T. Penning de Vries et al. (eds.). Klewer Academic Publishers. pp 505-513.

Volume 4
Chapter 14

Generic Plant Phosphorus Module

J.W. Jones
U. Singh
C.H. Porter
W.M. Bostick
K. A. Dzotsi

University of Florida, Gainesville, FL
International Center for Soil Fertility and Agricultural Development
International Consortium for Agricultural Systems Applications

GENERIC PLANT PHOSPHORUS MODULE (P_PLANT)

June 29, 2005

J. W. Jones, U. Singh, C. H. Porter, W. M. Bostick, K. A. Dzotsi

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Generic Plant Phosphorus Module
TABLE OF CONTENTS

1.0 Overview of the Generic Plant Phosphorus Module 1

2.0 P_Plant Module Structure 3
2.1 Subroutine P_Plant 6
2.2 Subroutine P_Demand 10
2.3 Subroutine P_Uptake 11
2.4 Subroutine P_Partition 12
2.5 Subroutine P_IpPlant 12
2.6 Subroutine OpPlantP 13

3.0 Root Soil Volume computation 13

4.0 Species File Parameters 14

5.0 References 15

LIST OF TABLES

Table 1 - P_Plant Module Interface 4
Table 2 – Root Soil Volume Interface 13

LIST OF FIGURES

Figure 1. Processes in Plant P module 2
Figure 2 – Generic Plant Phosphorus Module Structure 5
Figure 3. Optimum and minimum P concentrations in vegetative tissue as a function of physiological development for soybean 7
Figure 4. P concentration in soybean shoots 7
Figure 5 - P Stress factors for modifying vegetative partitioning and for reducing photosynthesis due to phosphorus deficiency 10
Figure 6. Phosphorus content section of Soybean Species file 15

1.0 Overview of the Generic Plant Phosphorus Module

The generic plant phosphorus module computes uptake and partitioning of phosphorus on a daily time step. This is an important component of the phosphorus cycle, especially in areas where soil P is deficient or limiting. The generic plant phosphorus module is based on the work done by Daroub et al. (2003) and by Bostick (2002). Adaptation for DSSAT-CSM are described by Dzotsi et al. (2010).

Figure 1 illustrates the processes modeled by the plant P module. Plant phosphorus content is computed for four plant components: shoots (leaf plus stem), roots, shell and seed. Optimum and minimum plant phosphorus concentrations for each of these plant components are defined at three critical stages in the growth of the plant. These P concentrations are stored in the species file (*.SPE) for each crop, thus allowing different crops to use stages that define the seasonal changes in optimum and minimum plant P content. The definition of the critical stage may vary from species to species, but for most crops, these will represent the vegetative and reproductive phases of development. Crop development is expressed as a fraction of developmental time between these critical stages.

Demand for P by the plant is computed as the amount of P needed to bring the four plant components up to optimum concentrations, including the demand for new growth. Total plant P demand is computed as the sum of demand from each of the four plant components. Phosphorus uptake by roots may be limited by either P supply in the soil or by plant demand. The distribution of uptake is related to the soil volume explored by roots in each soil layer. Partitioning of phosphorus taken up by plant roots is preferential to seeds, then shells, then vegetative tissue. If insufficient P is available for seeds, additional P can be translocated from vegetative and shell tissue.

Two phosphorus stress factors are computed, which can affect photosynthesis and dry matter growth partitioning. Threshold P deficit ratios for these stress factors are defined for each crop in its species file.

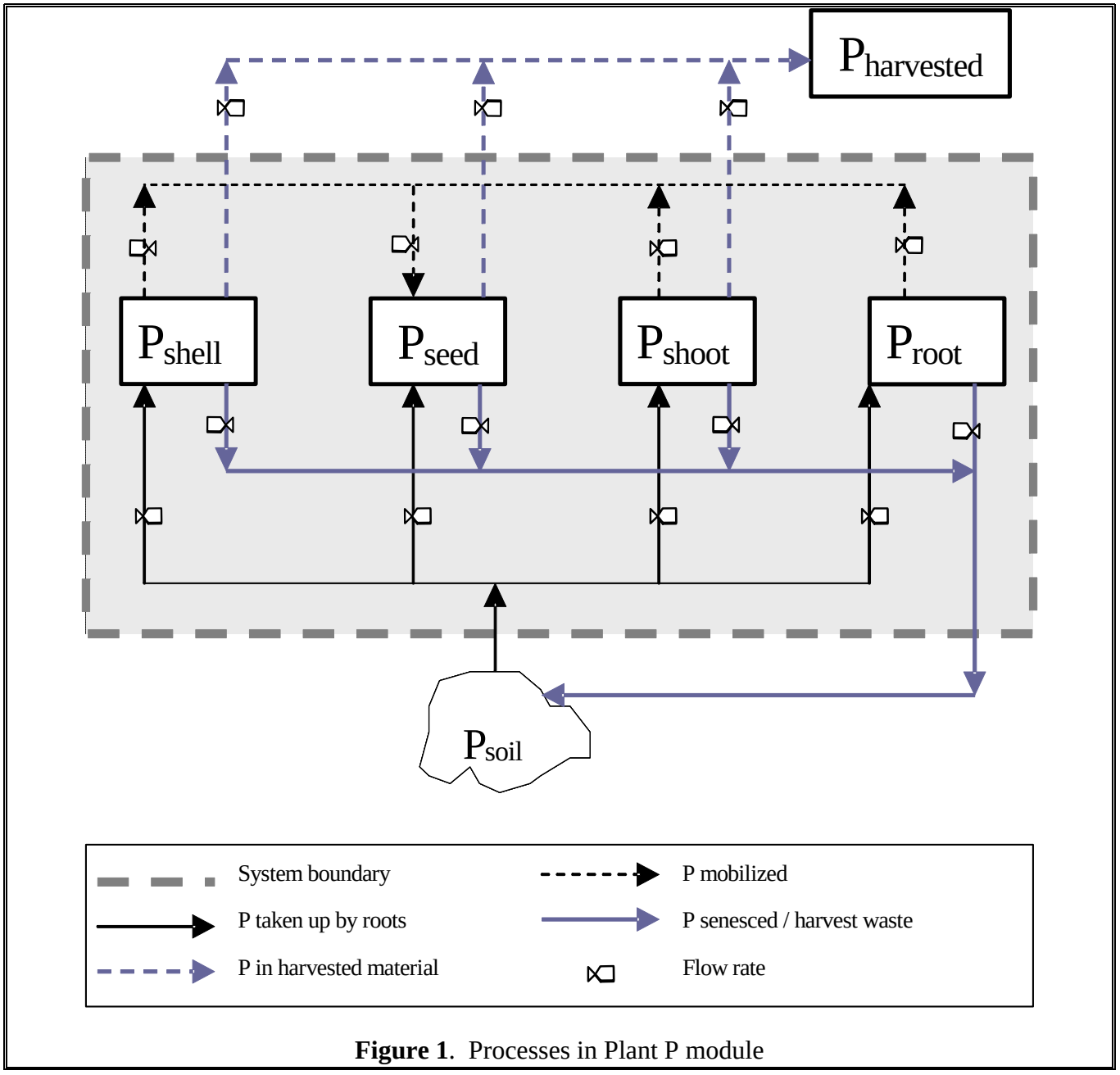


Figure 1. Processes in Plant P module

2.0 P_Plant Module Structure

The module structure follows the guidelines described by Porter and Jones (1998) and Jones (2003) for inclusion in the DSSAT4 Cropping Systems Model. The routine is called from any plant growth routine for input and initialization of variables, computation of rates, integration of state variables and for daily and seasonal output.

Table 1 lists the interface variables for the module. In order to link a crop growth module to this phosphorus module, the plant growth routine must compute and supply the input variables as defined in Table 1. The plant P module computes and returns the output variables as listed for use by each crop growth module. The interface is designed to be generic, such that any crop can generate the appropriate input data, which, along with data parameters supplied in the species file, will define plant phosphorus demand and partitioning during the crop growth cycle.

Figure **2** illustrates the components of the module. The main routine, P_Plant is called from each of the DSSAT4 crop growth modules to perform initialization at the beginning of each season and at plant emergence, state variable integration and daily and seasonal output. This main routine performs most of the computations involved in the plant phosphorus cycle, including daily computation of minimum and optimum P levels for each plant component, computation of actual P concentrations after uptake and partitioning, and computation of stresses due to P deficits. Five additional subroutines are called by the main module:

- P_IPPLANT, which reads data from the species file;
- P_DEMAND, which computes the daily demand for P;
- P_UPTAKE, which computes the daily uptake of P;
- P_PARTITION, which computes the daily partitioning of P to the plant components; and
- OPPLANTP, which prints daily output, and if directed, computes daily and seasonal balances for the Plant P routine.

The plant uptake routine depends on computation of available phosphorus, which is the phosphorus located in the soil volume in direct contact with roots (i.e., within 2 mm). This “root soil volume” is updated daily by the RootSoilVol subroutine, which is called by each plant growth routine. Actual P available within this root zone is computed by the inorganic soil P module.

Table 1 - P_Plant Module Interface		
Input Variables:	**Definition**	**Units**
CROP	Two-character crop identification code	--
DLAYR(L)	Thickness of soil layer L	cm
DS(L)	Depth to bottom of soil layer L	cm
DYNAMIC	Module control variable; =RUNINIT, SEASINIT, RATE, EMERG, INTEGR, OUTPUT, or FINAL	--
FILECC	Path plus filename for species file (*.spe)	--
ISWPHO	Phosphorus simulation switch (Y or N)	--
Leaf_kg	Mass of leaf tissue	kg[leaf]/ha
MDATE	Harvest maturity date	YYYYDDD
NLAYR	Actual number of soil layers	--
PCNVeg	Percent nitrogen in vegetative tissue (leaf and stem)	kg [N]/ha
PestShut	Reduction in shoot mass due to pest or disease damage	kg/ha
PestRoot	Reduction in root mass due to pest or disease damage	kg/ha
PestShel	Reduction in shell mass due to pest or disease damage	kg/ha
PestSeed	Reduction in seed mass due to pest or disease damage	kg/ha
PHFRAC1	Fraction of physiological time which has occurred between first and second critical phases for computation of optimum and minimum plant P concentrations	
PHFRAC2	Fraction of physiological time which has occurred between second and third critical phases for computation of optimum and minimum plant P concentrations	
RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
Root_kg	Root mass	kg/ha
RootMob	Reduction in root mass due to N mobilization	kg/ha
RWU(L)	Root water uptake from soil layer L	cm/d
Seed_kg	Seed mass	kg/ha
SenSoilP	P in senesced root and nodule tissue	kg[P]/ha/d
SenSurfP	P in senesced canopy tissue	kg[P]/ha/d
Shel_kg	Shell mass	kg/ha
ShelMob	Reduction in shell mass due to N mobilization	kg/ha
ShutMob	Reduction in shoot mass due to N mobilization	kg/ha
SPi_AVAIL(L)	Soil P which is available for uptake by plants from soil layer L.	kg/ha
Stem_kg	Stem mass	kg/ha
YRPLT	Planting date	YYYYDDD
Output Variables:	**Definition**	**Units**
PConc_Root	P concentration in root	g[P]/g[root]
PConc_Seed	P concentration in seed	g[P]/g[seed]
PConc_Shel	P concentration in shell	g[P]/g[shell]
PConc_Shut	P concentration in shoots (leaf and stem)	g[P]/g[shoot]
PRoot_kg	P content in roots	kg[P]/ha
PSeed_kg	P content in seed	kg[P]/ha
PShel_kg	P content in shell	kg[P]/ha
PShut_kg	P content in shoots (leaf and stem)	kg[P]/ha
PSTRES1	P stress which affects vegetative partitioning (1=no stress, 0=max stress)	--
PSTRES2	P stress factor for reducing photosynthate (1=no stress, 0=max stress)	--
PUptake(L)	Plant uptake of P from soil layer L	kg[P]/ha/d

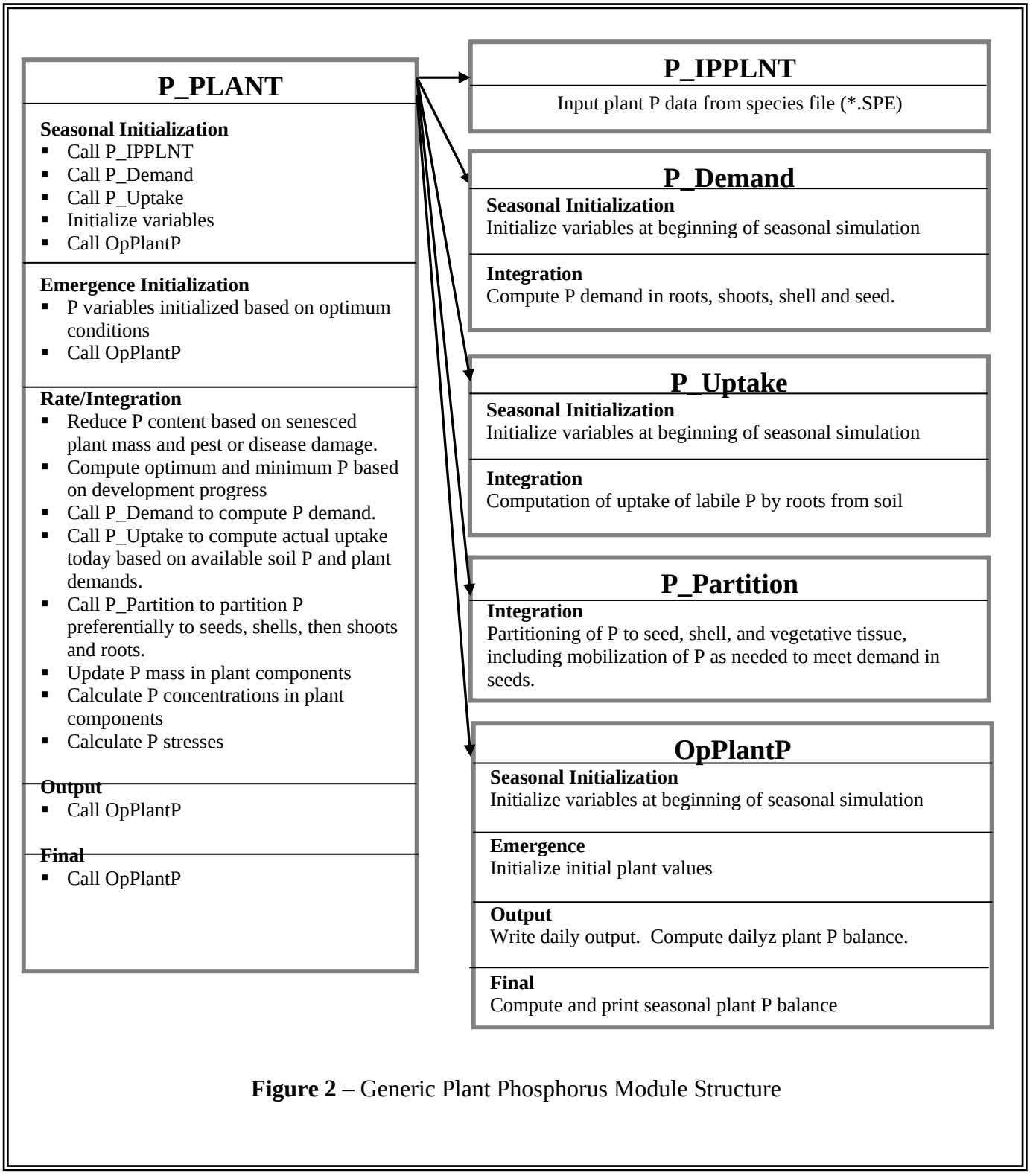


Figure 2 – Generic Plant Phosphorus Module Structure

2.1 Subroutine P_Plant

Seasonal Initialization

At the beginning of each season, P variables are set to zero. This initialization is done regardless of whether P is being simulated, so that values from the previous season will not be "carried-over". If P is to be simulated this season, the species file parameters are read by the P_IPPLNT routine and headers are written to the daily output file by OpPlantP.

Two options are available for input of data from the species file. Because some species will report leaf and stem variables separately and some crop species will report only the combined value, the generic plant phosphorus model can accept either format. The variable "UseShoots" is sent back from the P_IPPLNT routine to distinguish between the two methods of input:

- UseShoots = "FALSE" – indicates that data for optimum and minimum P concentrations were read for leaves and stems individually, rather than for shoots.
- UseShoots = "TRUE" – indicates that data for optimum and minimum P concentrations were read for shoots rather than separately for leaves and stems.

Emergence Initialization

On the day of plant emergence or planting day for transplants, the initial P concentrations in shoots and roots are set to the optimum concentration as read from the species file.

Integration

Each day, leaf and stem masses are combined into the shoot mass. If the leaf and stem optimum and minimum P concentrations were read from the species file (i.e., UseShoots logical variable is equal to "FALSE"), then the optimum and minimum P concentrations for the shoots are computed based on a weighted average of today's leaf and stem masses. Thereafter, in the plant P module, state variables are kept only for the combined shoot variable rather than for leaf and stem separately.

Amounts of P lost due to senescence and pest or disease damage are subtracted from the total P content of each of the four plant components.

The phase fraction variables which are sent from the plant growth routines (PhFrac1 and PhFrac2) represent the fraction of physiological development between critical plant growth stages. These critical stages correspond to the optimum and minimum P concentrations read from the species file for each plant component. The actual physical meaning of the critical stage is unimportant in the P module and may vary from crop to crop, thus allowing some flexibility in the representation of P partitioning. Optimum and minimum P concentrations are linearly interpolated between stages as depicted in Figure 3. An example of the resulting P concentration for a soybean run is shown in Figure 4.

In a similar way, minimum and maximum N:P ratios are computed daily based on values read from the species file which correspond to critical growth stages.

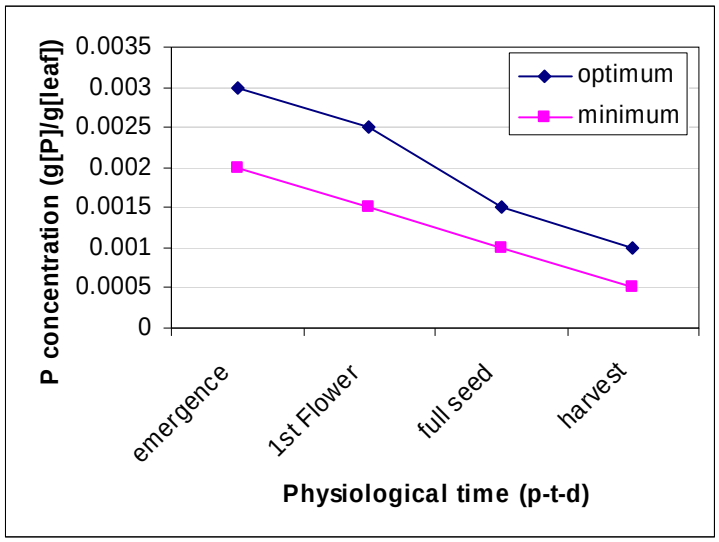


Figure 3. Optimum and minimum P concentrations in vegetative tissue as a function of physiological development for soybean.

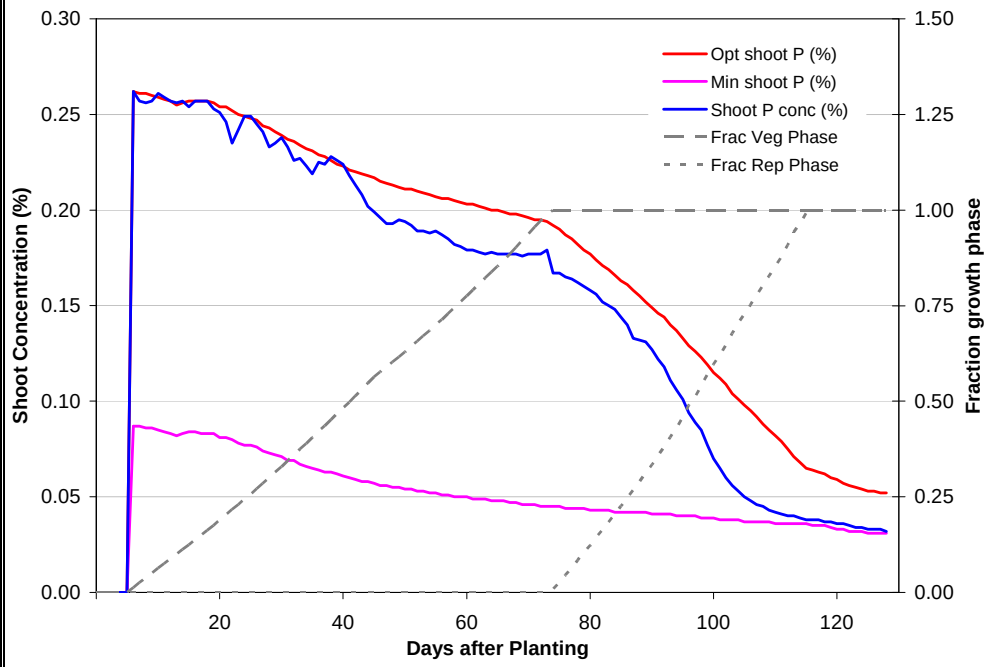


Figure 4. P concentration in soybean shoots

After computation of minimum and optimum P concentrations, the plant P demands are computed in Subroutine P_Demand. This subroutine includes P, which is available to meet plant demand from remobilization of N and C from leaves and stem tissue.

Total uptake of P from the soil is computed in the subroutine, P_Uptake based on the demand and the available phosphorus in the soil. This uptake is computed for each soil layer that contains roots and is stored in array, PUptake. Partitioning of P to the various plant parts with preference to seeds, then shells, then vegetative tissue is performed by the P_Partition subroutine. After partitioning, the plant state P variables are updated and concentrations of P in plant matter are computed.

Two stress factors are computed which can be used to reduce plant function when phosphorus levels are insufficient to meet the plant demand. These stress values are computed based on the ratio of actual P concentration to optimum P concentration, or PSTRESS_RATIO:

```
 PSTRESS_RATIO = MIN(1.0, (PConc_Shut - PConc_Shut_Min) /
&                     (PConc_Shut_opt - PConc_Shut_Min))
```

Equation 1

Two threshold values of this stress ratio are input from the species file to represent the sensitivity of each crop to P deficiency. These species parameters are labeled SRATPART and SRATPHOTO to reflect that the sensitivities are related to partitioning and photosynthesis in the CROPGRO routines. Other crop growth routines could assign different meanings to the two sensitivity parameters and use the stress factors in different ways. The stress factors are computed as follows:

P stress =
- 0 → for `PSTRESS_RATIO` <= 0 (maximum stress)
- `PSTRESS_RATIO` / sensitivity ratio
  → for 0 < `PSTRESS_RATIO` < sensitivity ratio
- 1 → for `PSTRESS_RATIO` >= sensitivity ratio (minimum stress)

Equation 2

PStress1, the stress factor that is used to modify partitioning, is computed using SRATPART, or the partitioning sensitivity ratio. PStress2, the stress factor that is used to reduce photosynthesis, is computed using SRATPHOTO, or the photosynthesis sensitivity ratio. Figure 5 shows the computation of the two stress parameters (labeled “P stress veg” and “P stress photo”). Pstress_Ratio is computed as shown in Equation 1, above. Based on the threshold values of SRAT_Part and SRAT_Photo (values of 1.0 and 0.8, respectively, as read from the species file), the two stress parameters are computed as shown in Equation 2.

Output and Final

The output routine is called to produce daily and seasonal output.

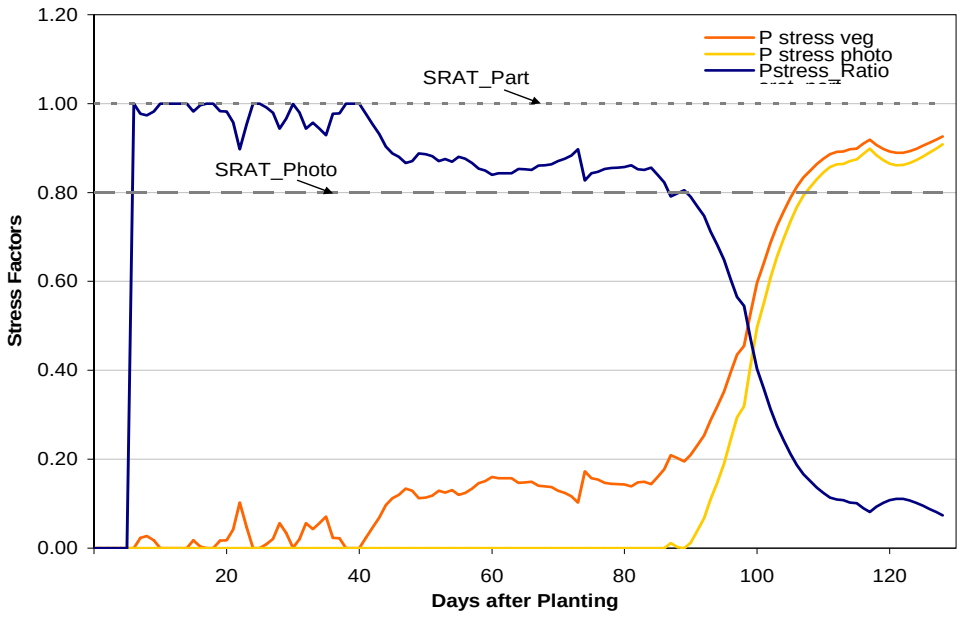


Figure 5 - P Stress factors for modifying vegetative partitioning and for reducing photosynthesis due to phosphorus deficiency

2.2 Subroutine P_Demand

Seasonal Initialization

Mobilization pools for the plant parts are initialized to zero.

Rate/Integration

Demands for P are calculated based on the difference between actual P concentration and optimum concentration in the four plant components, including the demand for today's new growth of tissue.

Demand for P is first met by mobilization of P from other plant components. Mobilization of P from plant tissues can occur in three ways:

- Supply of P from shoots, roots and shells when actual P concentrations exceed optimum.
- Release of P from shoots, roots and shells due to the loss of tissue mass from N and C mobilization. Currently, the CROPGRO model is the only crop growth model within the DSSAT suite that computes this source of mobilization. These plant tissue mass losses are input to the P_Plant routine as variables ShutMob, RootMob, and ShelMob.
- Mining of P from shoot, root or shell tissue to support the growth of seeds, even when this reduces the P content of these plant components below optimum concentrations.

The first two sources of mobilized P are handled in this subroutine, while the third source is computed during the partitioning of P uptake in subroutine P_Partition.

Each day minimum and optimum P content is computed for each plant component based on the daily minimum and optimum concentrations previously computed. Demand for P for each plant component is then the difference between the optimum P content and the actual P content. If this demand is less than zero, for example when P concentrations are above the optimum, P becomes available for mobilization. This amount is stored in a mobilization pool from which demand can be met.

```
PShutDem = ShutPOpt - PShut_kg
IF (PShutDem < 0.0) THEN
  PShutMobPool = PShutMobPool - PShutDem    !(neg. val.)
  PShutDem = 0.0
ENDIF
```

The P released due to tissue loss with N and C mobilization is added to the mobilization pool:

```
PShutMobPool = AMAX1(0.0, ShutMob * PConc_Shut) + PShutMobPool
```

The amount of P mobilized in a day due to N and C mobilization is limited by the minimum P concentration for each plant component:

```
PShutMobToday = AMIN1(PShutMobPool, PShut_kg - ShutPMin)
```

If any P mobilization is computed, it is first used to reduce today's demand for that plant component:

```
IF (PShutDem >= PShutMobToday) THEN
  PShutMobPool = PShutMobPool - PShutMobToday
  PShutDem = PShutDem - PShutMobToday
  PShutMobToday = 0.0
ELSE
  PShutMobPool = PShutMobPool - PShutDem
  PShutMobToday = PShutMobToday - PShutDem
  PShutDem = 0.0
ENDIF
```

The mobilized P pools are then used to meet demand for other plant components, thereby reducing the demand for uptake of P from the soil.

2.3 Subroutine P_Uptake

Seasonal Initialization
Uptake variables are initialized to zero.

Rate/Integration
Daily P uptake by the plant is equal to the minimum of plant demand or a fraction of the soil supply. The soil P supply is the available labile P in the root zone of the soil profile, as

computed by the inorganic soil phosphorus routine. The fraction of the available labile P which can be taken up in a day, FracPUptake, is read from the species file.

The soil layer distribution of P uptake is computed as the proportion of supply that exists in that layer to the total supply. The supply in each layer is a function of the available roots in the layer and the P concentration in the layer.

P uptake can also be limited if the N:P ratio in the vegetative tissue falls below a minimum value. This critical N:P ratio, N2P_min, is computed daily based on values which are supplied at the critical stages of plant development, in the same manner that minimum and optimum P concentrations are computed.

2.4 Subroutine P_Partition

During reproductive growth of many crops, phosphorus can be "mined" from the vegetative tissue to support the demand for growth of seeds and shells. The portion P that can be mined in a day is defined by the species file parameter, FracPMobil. P which can be mobilized from vegetative tissue is computed as:

```
      PMine_Avail = PShut_kg  + PRoot_kg  + PShel_kg   !Actual P
     &            - ShutPMin - PRootMin - PShelMin     !Minimum P

!     Maximum that can be mobilized in one day
      P_Mobil_max = MAX(0.0, FracPMobil * PMine_Avail)
```

This P available for mobilization is then divided into that which would be mobilized from roots shoots and shells by the variables RootMineFrac and ShutMineFrac and ShelMinFrac. Note that this computation of P which can be mobilized differs from that computed in the P_Demand routine in that it is driven by demand for seed growth rather than by an excess of P in shoot, root and shell tissue, or by a loss of leaf mass due to mobilization of N or C from the plant tissues.

Partitioning of P to the various plant components proceeds as follows. Phosphorus taken up from the soil is first used to meet seed demand. If the uptake is not sufficient to meet this demand, the seeds will take P from vegetative tissue, if P_Mobil_max is positive. If any P supply is left after seed demand is met, then it will be sent to shells. Vegetative tissue is supplied after reproductive demand is met if any P remains.

2.5 Subroutine P_IpPlant

Subroutine P_IpPlant simply reads the species file for phosphorus parameters and checks the data for validity. Error messages are printed and the program is halted if errors are found in the species file data.

When the optimum shoot P concentrations are read from the species file, the UseShoots logical variable is set. If any of the optimum shoot P concentration data are missing (or have

negative values), then the UseShoots variable is set to FALSE. Otherwise, it is TRUE.

2.6 Subroutine OpPlantP

Seasonal Initialization
Headers are written for the daily plant phosphorus output file (PlantP.OUT) and the seasonal P balance (PlantPBal.OUT).

Emergence Initialization
Initial plant phosphorus content is stored.

Output
Phosphorus stresses are averaged over the print period (variables PS1_AV and PS2_AV), so that for a print interval greater than 1 day (FROP > 1), any stresses which occur during the interval since last printout will be seen.
An option exists for a daily plant P balance, which is triggered with the detail output switch, IDETL = 'D'.

Final
At the end of each season, a seasonal plant phosphorus balance report is written.

3.0 Root Soil Volume computation

This subroutine is called by each plant growth module, rather than by the plant phosphorus module, but is included in this chapter because it is a necessary and parallel component of the P uptake routine. Table 2 lists the interface variables for the RootSolVol subroutine.

Table 2 – Root Soil Volume Interface

Variable Names	Definition	Units
Input:		
DLAYR(L)	Thickness of soil layer L	cm
DS(L)	Depth to bottom of soil layer L	cm
DYNAMIC	Module control variable; =RUNINIT, SEASINIT, RATE, EMERG, INTEGR, OUTPUT, or FINAL	--
ISWPHO	Phosphorus simulation switch (Y or N)	--
NLAYR	Actual number of soil layers	--
PLTPOP	Plant Population	#/m^2
RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
RTDEP	Root depth	cm
Output:		
SVRoots	Soil volume in direct contact with roots	cm^3 / cm^2

Seasonal Initialization
At the beginning of each season of simulation, the soil volume adjacent to roots is set to zero and the soil volume not adjacent to roots is set equal to the total soil volume. The value of

ROWSP is read from FILEIO.

Rate or Integration section
This section of code can be called from either the rate or integration section. This allows flexibility for the various plant growth routines. A logical variable, FIRST, indicates whether roots have previously grown. If no roots are present, no computations are done.

Upon the first appearance of roots, the root soil volume is initialized as a 5 cm square around each plant to the depth of RTDEP. The initial root soil volume for the field is found by multiplying by the plant population. If plant spacing limits the initial soil volume, the value is adjusted.

After the initialization upon emergence, new root soil volume is added based on new root growth every day and an assumed radius of 2mm around each root.

4.0 Species File Parameters

The additional input data required by the species file for each crop is the same. It is intended to be flexible enough to allow different crops to supply the appropriate data, even though the various crop components may be quite different. The important information is the minimum and optimum P concentrations, which change based on the development progress of the plant.

Figure 6 shows the phosphorus content section for the soybean species file. The file contains optimum and minimum P concentrations for each of five plant components: shoots, leaves, stems, shell and seed. Two options exist for entering this data:

(1) data for shoots can be entered which contains the combined P concentrations for leaf and stem, or
(2) data can be entered separately for leaf and stem.

If data is entered for both options, the values for shoots are used in computations and those for leaf and stem are discarded. In the case of the soybean file, values for leaf and stem are entered and those for shoots are shown as -99 values. In the case of soybean, the values of optimum and minimum P concentrations are entered for the various plant components at three critical stages: emergence, first flower and full seed. Between these stages of development, optimum and minimum P concentrations are linearly interpolated.

Maximum and minimum N:P ratios are input as a means to limit the P uptake if insufficient N is present.

The two species file stress parameters SRATPHOTO and SRATPART are defined as the minimum value of the ratio of P in vegetative tissue to the optimum P, below which photosynthesis is reduced (SRATPHOTO) or vegetative partitioning is affected (SRATPART). FracPMobil is the fraction per day of vegetative P that can be mined per day to supply the demands for seed and shell growth. FracPUptake is the maximum fraction of available P which can be taken up in a day.

```
!*PHOSPHORUS CONTENT
! Two options for Optimum and minimum P concentrations for shoots:
! (1) supply values for shoots (leaf plus stem)
! (2) supply values individually for leaf and for stem
! If positive values for shoots are supplied, any values supplied for leaf and
! stem will be ignored.
   -99.0   -99.0   -99.0  Optimum Shoot Conc (emergence, 1st flower, full seed)
  0.0028  0.0025  0.0010  Optimum Leaf Conc  (emergence, 1st flower, full seed)
  0.0018  0.0016  0.0005  Optimum Stem Conc  (emergence, 1st flower, full seed)
  0.0014  0.0012  0.0005  Optimum Root Conc  (emergence, 1st flower, full seed)
   -99.0  0.0040  0.0005  Optimum Shell Conc (emergence, 1st flower, full seed)
   -99.0  0.0045  0.0040  Optimum Seed Conc  (emergence, 1st flower, full seed)
   -99.0   -99.0   -99.0  Minimum Shoot Conc (emergence, 1st flower, full seed)
  0.0010  0.0007  0.0005  Minimum Leaf Conc  (emergence, 1st flower, full seed)
  0.0003  0.0003  0.0003  Minimum Stem Conc  (emergence, 1st flower, full seed)
  0.0003  0.0003  0.0003  Minimum Root Conc  (emergence, 1st flower, full seed)
   -99.0  0.0003  0.0003  Minimum Shell Conc (emergence, 1st flower, full seed)
   -99.0  0.0030  0.0030  Minimum Seed Conc  (emergence, 1st flower, full seed)

    50.0    20.0    10.0  Maximum Veg N:P ratio (emergence, 1st flower, full seed)
     5.0     5.0     1.0  Minimum Veg N:P ratio (emergence, 1st flower, full seed)

    0.80    1.00  SRATPHOTO, SRATPART
    0.20          FracPMobil
    0.50          FracPUptake - Maximum fraction of available P which can be taken up /
day
```

Figure 6. Phosphorus content section of Soybean Species file

5.0 References

Bostick, W.M.. 2002. Integration of a soil-plant phosphorous model into DSSAT-CropGro. Project Report for ABE 6933 University of Florida, July, 2002.

Daroub, S.H, A. Gerakis, J.T. Ritchie, D.K. Friesen, J. Ryan. 2003. Development of a soil-plant phosphorus simulation model for calcareous and weathered tropical soils. Agricultural Systems 76(2003): 1157-1181.

Dzotsi K.A., Jones J.W., Adiku S.G.K., Naab J.B., Singh U., Porter C.H., and Gijsman A.J., 2010. Modeling soil and plant phosphorus within DSSAT. Ecological Modelling 221: 2839-2849

Jones, J.W., G. Hoogenboom, C.H. Porter, K.J. Boote, W.D. Batchelor, L.A. Hunt, P.W. Wilkens, U. Singh, A.J. Gijsman, J.T. Ritchie. The DSSAT cropping system model. European Journal of Agronomy 18 (2003) 235- 265.

Porter, C.H. and J.W.Jones. 1998. Modular Structure in CROPGRO Documentation and Source Code Listing. Agricultural and Biological Engineering Department. Research Report No 98-601. University of Florida, Gainesville, Florida.

Volume4
Appendix A

Structure and Definition of Composite Variables

C.H. Porter

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Appendix A
Structure of composite variables
as defined in ModuleDefs.FOR

Table A- 1 CONTROL Variable 1
Table A- 2 Flood Nitrogen (FLOODN) 2
Table A- 3 Flood water (FLOODWAT) 2
Table A- 4 Harvest Residues (HARVRES) 2
Table A- 5 Switches and Methods (ISWITCH) 3
Table A- 6 Oxidation layer (OXLAYR) 4
Table A- 7 Plant Stresses (PlantStres) 4
Table A- 8 Senesced Plant Matter (SENESCE) 4
Table A- 9 Soil Properties (SOILPROP) 5
Table A- 10 Daily weather variables (WEATHER) 6

Table A- 1 CONTROL Variable

Variable Name	Definition
CROP	Two-character crop identification code
DAS	Days after start of simulation (d)
DYNAMIC	Module control variable; =RUNINIT, SEASINIT, RATE, EMERG, INTEGR, OUTPUT, or FINAL
FILEIO	Filename for input file (e.g., IBSNAT35.INP) (--)
FROP	Frequency of output (d)
LUNIO	Logical unit number for FILEIO
MESIC	Code for sequenced runs: 'S'=second or subsequent run in sequenced simulation, 'M'=single run or first run of sequenced simulation. (--)
MODEL	Name of CSM executable file
MULTI	Current simulation year (=1 for first or single simulation, =NYRS for last seasonal simulation)
NYRS	Number of years of simulations to perform for multi-season run (each with identical initial conditions, but different weather years) (--)
REPNO	Replication number for current simulation
RNMODE	Simulation run mode (I=Interactive, A=All treatments, B=Batch mode, E=Sensitivity, D=Debug, N=Seasonal, Q=Sequence)
RUN	Simulation run number
YRDIF	Increment in years which must be added to operations dates for seasonal or sequenced simulations (yr)
YRDOY	Current day of simulation (YYYYDDD)
YRSIM	Start of simulation date (YYYYDDD)

Table A- 2 Flood Nitrogen (FLOODN)	
Variable Name	**Definition**
DLTFNH4	Rate of change of ammonium in floodwater (kg [N] / ha / d)
DLTFNO3	Rate of change of nitrate in floodwater (kg [N] / ha / d)
DLTFUREA	Rate of change of urea in floodwater (kg [N] / ha / d)
FLDH4	Mass of ammonium in flood water (kg[N]/ha)
FLDH4C	Ammonium concentration in flood water (mg[N]/l[water])
FLDN3	Mass of nitrate in flood water (kg[N]/ha)
FLDN3C	Nitrate concentration in flood water (mg[N]/l[water])
FLDU	Mass of urea in flood water (kg[N]/ha)
FRNH4U	Plant uptake of ammonium from flood water (kg[N]/ha)
FRNO3U	Plant uptake of nitrate from flood water (kg[N]/ha)
NDAT	Number of days after transplanting (d)

Table A- 3 Flood water (FLOODWAT)	
Variable Name	**Definition**
ABUND	Bund height (mm)
BUNDED	Logical variable indicating if field is currently bunded
CEF	Cumulative seasonal evaporation from floodwater surface (mm)
EF	Evaporation rate from flood surface (mm / d)
FLOOD	Current depth of flooding (mm)
FRUNOFF	Loss of floodwater over the bund (mm / d)
INFILT	Infiltration rate (mm / d)
NBUND	Number of bund height records
PERC	Percolation rate, user input for puddle field (cm/d)
PUDDLED	Logical variable indicating whether lowland field has been puddle
PUDPERC	Potential percolation rate for puddle field (mm/d)
RUNOFF	Calculated runoff (mm/d)
TOTBUNDRO	Cumulative seasonal flood runoff over bund (mm)
YRDRY	First day of dry field conditions (after flood) (YYYYDDD)
YRWET	First day of new flood event (YYYYDDD)

Table A- 4 Harvest Residues (HARVRES)	
Component Name	**Description**
ResWt(L)	Mass of harvest residue dry weight in layer L (L=0 for surface layer) (kg/ha)
ResLig(L)	Mass of lignin in harvest residue in layer L (L=0 for surface layer) (kg[lignin]/ha)
ResE(L,I)	Mass of element component in harvest residue in layer L (L=0 for surface layer), I = 1-N, I=2-P, etc. (kg[E]/ha)

Table A- 5 Switches and Methods (ISWITCH)	
Variable Name	**Definition**
IDETC	Code to generate OUTC file (e.g., SoilC.OUT), Y or N
IDETD	Switch to generate PEST.OUT file (Y or N)
IDETG	Flag to print growth output file
IDETL	Switch for detailed printout (Y or N)
IDETN	Code to generate OUTN file (e.g., SoilN.OUT), Y or N
IDETO	Switch for printing OVERVIEW.OUT file
IDETR	Code to generate OUTR file (e.g., OPERAT.OUT), Y or N
IDETS	Code to generate summary output files
IDETW	Y=detailed water balance output, N=no detailed output
IFERI	Fertilizer switch (A= automatic, R= on specified dates (YYYYDDD format), D = on specified days after planting (DDD) format. (--)
IHARI	Harvest type code: M=at harvest maturity, R=on specified day of year (HDATE), D= on specified day after planting (HDATE), G= at specified growth stage (HSTG), A= within specified window when SW conditions are met (--)
IIRRI	Irrigation switch R=on reported dates, D=as reported, days after planting, A=automatic, when required., F=automatic w/ fixed amt, P=as reported thru last reported day then automatic, W=as reported thru last reported day then fixed amount, N=not irrigated
IPLTI	Switch for automatic planting (‘A’=automatic)
IRESI	Residue application method. A=Automatic residue application at a certain days after planting for multiple years; N=No residue; R=On reported dates; D=As reported, in DAP; F=Auto, with fixed amounts (--)
ISWCHE	Switch for observed chemical applications
ISWDIS	Pest damage simulation switch (Y or N)
ISWNIT	Nitrogen simulation switch (Y or N)
ISWPHO	Phosphorus simulation switch (Y or N)
ISWSYM	Nitrogen fixation simulation switch (Y = simulate nodule growth, N = no nodule growth, U = N-fixation occurs at a rate that carbon will allow, and nodules are not grown explicitly)
ISWTIL	Swith for observed tillage application
ISWWAT	Water simulation control switch (Y or N)
MEEVP	Method of evapotranspiration (‘P’=Penman, ‘R’=Priestly-Taylor, ‘Z’=Zonal)
MEPHO	Method for photosynthesis computation (‘C’=Canopy or daily, ‘L’=hedgerow or hourly)
MESOM	Method for soil N computations (‘G’=Godwin or Ceres-based, ‘P’=Parton or Century-based (future))

Table A- 6 Oxidation layer (OXLAYR)	
Variable Name	**Definition**
ALGACT	Flood water algae index (0-1) (--)
DAILY	Logical variable to determine whether daily or hourly flood chemistry or oxidation layer calculations are done.
DLTOXH4	Rate of change of ammonium in oxidation layer (kg [N] / ha / d)
DLTOXN3	Rate of change of nitrate in oxidation layer (kg [N] / ha / d)
DLTOXU	Rate of change of urea in oxidation layer (kg [N] / ha / d)
IBP	
OXH4	Ammonium in oxidation layer (kg[N]/ha)
OXLT	Thickness of oxidation layer (cm)
OXMIN3	Minimum amount of nitrate in oxidation layer (kg[N]/ha)
OXMIN4	Minimum amount of ammonium in oxidation layer (kg[N]/ha)
OXN3	Nitrate in oxidation layer (kg[N]/ha)
OXU	Urea in oxidation layer (kg[N]/ha)
UNINCO	Logical variable indicating whether fertilizer is fully incorporated; if true, ignore oxidation layer

Table A- 7 Plant Stresses (PlantStres)	
Variable Name	**Definition**
ACTIVE(I)	Logical variable indicating whether Stage I is currently active for computing environmental and stress summary for Overview.OUT
AGEFAC	Relative effect of current leaf N on canopy photosynthesis (0-1) (fraction)
NSTAGES	Number of stages reported in Overview.OUT file for environmental and stress summary
NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)
StageName(I)	Stage name for environmental and stress summary in Overview.OUT
SWFAC	Effect of soil-water stress on photosynthesis, 1.0=no stress, 0.0=max stress
TURFAC	Water stress factor for expansion (0 – 1)

Table A- 8 Senesced Plant Matter (SENESCE)	
Variable Name	**Definition**
ResWt(L)	Mass of senesced matter dry weight in layer L (L=0 for surface layer) (kg/ha)
ResLig(L)	Mass of lignin in senesced matter in layer L (L=0 for surface layer) (kg[lignin]/ha)
ResE(L,I)	Mass of element component in senesced matter in layer L (L=0 for surface layer), I = 1-N, I=2-P, etc. (kg[E]/ha)

Table A- 9 Soil Properties (SOILPROP)	
Variable Name	**Definition**
ADCOEF(L)	Anion adsorption coefficient for soil layer L; for reduced anion (nitrate) flow in variable-charge soils (ADCOEF = 0 implies no anion retention) (cm3 (H2O] / g [soil])
BD(L)	Bulk density, soil layer L (g [soil] / cm3 [soil])
CACO3(L)	Calcium carbonate in soil layer L (%)
CEC(L)	Cation exchange capacity, soil layer L (cmol kg-1)
CLAY(L)	Percentage of clay in soil layer L
CN	Runoff Curve Number – measure of runoff potential based on soil type and current soil water content.
DLAYR(L)	Thickness of soil layer L (cm)
DMOD	Factor to adjust the mineralization rate for certain atypical soils (range 0-1) (--)
DS(L)	Cumulative depth in soil layer L (cm)
DUL(L)	Volumetric soil water content at Drained Upper Limit in soil layer L (cm3[water]/cm3[soil])
LL(L)	Volumetric soil water content in soil layer L at lower limit (cm3 [water] / cm3 [soil])
NLAYR	Actual number of soil layers
OC(L)	Organic carbon content of layer (%)
PH(L)	Ph in soil layer L (--)
PHKCL(L)	Ph in buffer, soil layer L
SALB	Bare soil albedo (fraction)
SAND(L)	Percentage of sand in soil layer L
SAT(L)	Volumetric soil water content in layer L at saturation (cm3 [water] / cm3 [soil])
SILT(L)	Percentage of silt in soil layer L
SLPF	Soil photosynthesis factor, 0 to 1 scale
STONES(L)	Coarse fraction (>2 mm) (%)
SWCN(L)	Saturated hydraulic conductivity in layer L (cm/hr)
SWCON	Soil water conductivity constant; whole profile drainage rate coefficient (1/d)
TOTBAS(L)	Base saturation, soil layer L (cmol/kg)
TOTN(L)	Total N in soil (%)
U	Evaporation limit (cm)
WR(L)	Root hospitality factor, used to compute root distribution

Table A- 10 Daily weather variables (WEATHER)	
Variable Name	**Definition**
CLOUDS	Relative cloudiness factor (0-1)
CO2	Atmospheric carbon dioxide concentration (µmol[CO2] / mol[air])
DAYL	Day length on day of simulation (from sunrise to sunset) (hr)
PAR	Daily photosynthetically active radiation or photon flux density (moles[quanta]/m2-d)
RAIN	Precipitation depth for current day (mm)
REFHT	Reference height for wind speed (m)
SNDN	Time of sunset (hr)
SNUP	Time of sunrise (hr)
SRAD	Solar radiation (MJ/m2-d)
TA	Daily normal temperature (°C)
TAMP	Amplitude of temperature function used to calculate soil temperatures (°C)
TAV	Average annual soil temperature, used with TAMP to calculate soil temperature. (°C)
TAVG	Average daily temperature (°C)
TDAY	Average temperature during daylight hours (°C)
TDEW	Dew point temperature (°C)
TGROAV	Average daily canopy temperature (°C)
TGRODY	Average temperature during daylight hours (°C)
TMAX	Maximum daily temperature (°C)
TMIN	Minimum daily temperature (°C)
TWILEN	Daylength from twilight to twilight (h)
WINDHT	Reference height for wind speed (m)
WINDSP	Wind speed at 2m (km/d)
XLAT	Latitude (deg.)

Volume 4
Appendix B

CSM Module
Interface Tables

C.H. Porter

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Appendix B
List of CSM Module Interface Tables

Table B-1 – WEATHR Module Interface 1
Table B-2 – SOIL Module Interface 2
Table B-3 – Soil-Plant-Atmosphere Module Interface 3
Table B-4 – CROPGRO Interface 4
Table B-5 – Interface for Alt_Plant Plant Growth Modules 5
Table B-6 – Management Operations Module Interface 6
Table B-7 – SOILDYN Module Interface (Sub-module of SOIL) 7
Table B-8 – Soil Water Balance (WATBAL) Module Interface (Sub-module of SOIL) 7
Table B-9 – NTRANS Interface (Sub-module of SOIL) 8
Table B-10 – CENTURY Module Interface (Sub-module of SOIL) 9
Table B-11 – Soil Temperature (STEMP) Module Interface (Sub-module of SPAM) 10

Table B-1 – WEATHR Module Interface

	Variable Name	Definition	Units
Input Variables	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
Output Variables	CLOUDS	Relative cloudiness factor (0-1)	--
	CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	PAR	Daily photosynthetically active radiation or photon flux density	moles[quanta]/m2-d
	RAIN	Precipitation depth for current day	mm
	SRAD	Solar radiation	MJ/m2-d
	TA	Daily normal temperature	°C
	TAMP	Amplitude of temperature function used to calculate soil temperatures	°C
	TAV	Average annual soil temperature, used with TAMP to calculate soil temperature.	°C
	TAVG	Average daily temperature	°C
	TDAY	Average temperature during daylight hours	°C
	TDEW	Dew point temperature	°C
	TGRO(I)	Hourly canopy temperature	°C
	TGROAV	Average daily canopy temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	TWILEN	Daylength from twilight to twilight	h
	WINDSP	Wind speed at 2m	km/d
	XLAT	Latitude	deg.

Table B-2 – SOIL Module Interface			
	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	EO	Potential evapotranspiration rate	mm/d
	ES	Actual soil evaporation rate	mm/d
	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	IRRAMT	Irrigation amount for today	mm / d
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	RAIN	Precipitation depth for current day	mm
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	SRAD	Solar radiation	MJ/m2-d
	SRFTEMP	Temperature of soil surface litter	°C
	ST(L)	Soil temperature in soil layer L	°C
	SWDELTX(L)	Change in soil water content due to root water uptake in layer L	cm3 [water] / cm3 [soil]
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD
I/O	FLOODN	Composite variable which contains flood nitrogen mass and concentrations. See Appendix A..	var.
Output	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]
	SNOW	Snow accumulation	mm
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	SWDELTS(L)	Change in soil water content due to drainage in layer L	cm3 [water] / cm3 [soil]
	SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L	cm3 [water] / cm3 [soil]
	WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d

Table B-3 – Soil-Plant-Atmosphere Module Interface			
	Variable Name	Definition	Units
Input	CANHT	Canopy height	m
	CLOUDS	Relative cloudiness factor (0-1)	--
	CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	EORATIO	Ratio of increase in potential evapotranspiration with increase in LAI (up to LAI=6.0) for use with FAO-56 Penman reference potential evapotranspiration.	--
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	KSEVAP	Light extinction coefficient used for computation of soil evaporation	--
	KTRANS	Light extinction coefficient used for computation of plant transpiration	--
	PORMIN	Minimum pore space required for supplying oxygen to roots for optimal growth and function	cm3/cm3
	RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
	RWUMX	Maximum water uptake per unit root length, constrained by soil water	cm3[water] / cm [root]
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SRAD	Solar radiation	MJ/m2-d
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	SWDELTS(L)	Change in soil water content due to drainage in layer L	cm3 [water] / cm3 [soil]
	SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L	cm3 [water] / cm3 [soil]
	TA	Daily normal temperature	°C
	TAMP	Amplitude of temperature function used to calculate soil temperatures	°C
	TAV	Average annual soil temperature, used with TAMP to calculate soil temperature.	°C
	TAVG	Average daily temperature	°C
	TDEW	Dew point temperature	°C
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	WINDSP	Wind speed at 2m	km/d
	WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	XLAT	Latitude	deg.
Input/ Output	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
	TGRO(I)	Hourly canopy temperature	°C
	TGROAV	Average daily canopy temperature	°C
Output	EO	Potential evapotranspiration rate	mm/d
	EOP	Potential plant transpiration rate	mm/d
	ES	Actual soil evaporation rate	mm/d
	SRFTEMP	Temperature of soil surface litter	°C
	ST(L)	Soil temperature in soil layer L	°C
	SWDELTX(L)	Change in soil water content due to root water uptake in layer L	cm3 [water] / cm3 [soil]
	TRWUP	Potential daily root water uptake over soil profile	cm/d

Table B-4 – CROPGRO Interface			
	Variable Name	Definition	Units
Input	CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	EOP	Potential plant transpiration rate	mm/d
	HARVFRAC	Two-element array containing fractions of (1) yield harvested and (2) by-product harvested	fraction
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]
	PAR	Daily photosynthetically active radiation or photon flux density	moles[quanta]/m2-d
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	TAVG	Average daily temperature	°C
	TDAY	Average temperature during daylight hours	°C
	TGRO(I)	Hourly canopy temperature	°C
	TGROAV	Average daily canopy temperature	°C
	TMIN	Minimum daily temperature	°C
	TRWUP	Potential daily root water uptake over soil profile	cm/d
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD
Output	CANHT	Canopy height	m
	EORATIO	Ratio of increase in potential evapotranspiration with increase in LAI (up to LAI=6.0) for use with FAO-56 Penman reference potential evapotranspiration.	--
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	KSEVAP	Light extinction coefficient used for computation of soil evaporation	--
	KTRANS	Light extinction coefficient used for computation of plant transpiration	--
	MDATE	Harvest maturity date	YYYYDDD
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	PORMIN	Minimum pore space required for supplying oxygen to roots for optimal growth and function	cm3/cm3
	RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
	RWUMX	Maximum water uptake per unit root length, constrained by soil water	cm3[water] / cm [root]
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	STGDOY(I)	Day when plant stage I occurred	YYYYDDD
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	XLAI	Leaf area (one side) per unit of ground area	m2[leaf] / m2[ground]

Table B-5– Interface for Alt_Plant Plant Growth Modules			
	Variable Name	Definition	Units
Input	CO2	Atmospheric carbon dioxide concentration	µmol[CO2] / mol[air]
	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DAYL	Day length on day of simulation (from sunrise to sunset)	hr
	EOP	Potential plant transpiration rate	mm/d
	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
	HARVFRAC	Two-element array containing fractions of (1) yield harvested and (2) by-product harvested	fraction
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]
	SNOW	Snow accumulation	mm
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SRAD	Solar radiation	MJ/m2-d
	SRFTEMP	Temperature of soil surface litter	°C
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	TRWUP	Potential daily root water uptake over soil profile	cm/d
	TWILEN	Daylength from twilight to twilight	h
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD
I/O	FLOODN	Composite variable which contains flood nitrogen mass and concentrations. See Appendix A.	var.
Output	CANHT	Canopy height	m
	EORATIO	Ratio of increase in potential evapotranspiration with increase in LAI (up to LAI=6.0) for use with FAO-56 Penman reference potential evapotranspiration.	--
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	KSEVAP	Light extinction coefficient used for computation of soil evaporation	--
	KTRANS	Light extinction coefficient used for computation of plant transpiration	--
	MDATE	Harvest maturity date	YYYYDDD
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	PORMIN	Minimum pore space required for supplying oxygen to roots for optimal growth and function	cm3/cm3
	RLV(L)	Root length density for soil layer L	cm[root] / cm3[soil]
	RWUMX	Maximum water uptake per unit root length, constrained by soil water	cm3[water] / cm [root]
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	STGDOY(I)	Day when plant stage I occurred	YYYYDDD
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	XLAI	Leaf area (one side) per unit of ground area	m2[leaf] / m2[ground]

Table B-6– Management Operations Module Interface			
	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	RAIN	Precipitation depth for current day	mm
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	ST(L)	Soil temperature in soil layer L	°C
	STGDOY(I)	Day when plant stage I occurred	YYYYDDD
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
Output	HARVFRAC	Two-element array containing fractions of (1) yield harvested and (2) by-product harvested	fraction
	IRRAMT	Irrigation amount for today	mm / d
	MDATE	Harvest maturity date	YYYYDDD
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD

Table B-7– SOILDYN Module Interface (Sub-module of SOIL)		
I/O	Variable Name	Definition
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.
Input	MESOM	Method for soil N computations ('G'=Godwin or Ceres-based, 'P'=Parton or Century-based (future))
Output	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.

Table B-8– Soil Water Balance (WATBAL) Module Interface (Sub-module of SOIL)			
	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	ES	Actual soil evaporation rate	mm/d
	IRRAMT	Irrigation amount for today	mm / d
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	RAIN	Precipitation depth for current day	mm
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SWDELTX(L)	Change in soil water content due to root water uptake in layer L	cm3 [water] / cm3 [soil]
	TMAX	Maximum daily temperature	°C
I/O	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
Output	DRN(L)	Drainage rate through soil layer L	cm/d
	FLOW(L)	Movement of water between unsaturated soil layers due to soil evaporation: + = upward, -- = downward	cm/d
	SNOW	Snow accumulation	mm
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	SWDELTS(L)	Change in soil water content due to drainage in layer L	cm3 [water] / cm3 [soil]
	SWDELTU(L)	Change in soil water content due to evaporation and/or upward flow in layer L	cm3 [water] / cm3 [soil]
	WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d

Table B-9– NTRANS Interface (Sub-module of SOIL)			
	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DRN(L)	Drainage rate through soil layer L	cm/d
	ES	Actual soil evaporation rate	mm/d
	FLOODWAT	Composite variable containing information related to bund management. See Appendix A.	
	FLOW(L)	Movement of water between unsaturated soil layers due to soil evaporation: + = upward, -- = downward	cm/d
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SRAD	Solar radiation	MJ/m2-d
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	TMAX	Maximum daily temperature	°C
	TMIN	Minimum daily temperature	°C
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	XHLAI	Healthy leaf area index	m2[leaf] / m2[ground]
	YRPLT	Planting date	YYYYDDD
I/O	FLOODN	Composite variable which contains flood nitrogen mass and concentrations. See Appendix A.	var.
Out-put	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]

Table B-10– CENTURY Module Interface (Sub-module of SOIL)			
	Variable Name	Definition	Units
Input	CONTROL	Composite variable containing variables related to control and/or timing of simulation. See Appendix A.	--
	DRN(L)	Drainage rate through soil layer L	cm/d
	EO	Potential evapotranspiration rate	mm/d
	FLOW(L)	Movement of water between unsaturated soil layers due to soil evaporation: + = upward, -- = downward	cm/d
	HARVRES	Composite variable containing harvest residue amounts for total dry matter, lignin, and N amounts. See Appendix A.	
	ISWITCH	Composite variable containing switches which control flow of execution for model. See Appendix A.	
	NSTRES	Nitrogen stress factor (1=no stress, 0=max stress)	--
	SENESCE	Composite variable containing data about daily senesced plant matter. See Appendix A.	
	SOILPROP	Composite variable containing soil properties including bulk density, drained upper limit, lower limit, pH, saturation water content. See Appendix A.	
	SRFTEMP	Temperature of soil surface litter	°C
	ST(L)	Soil temperature in soil layer L	°C
	SW(L)	Volumetric soil water content in layer L	cm3 [water] / cm3 [soil]
	UNH4(L)	Rate of root uptake of NH4 in soil layer L	kg [N] / ha - d
	UNO3(L)	Rate of root uptake of NO3 in soil layer L	kg [N] / ha -d
	WINF	Water available for infiltration - rainfall minus runoff plus net irrigation	mm / d
	YREND	Date for end of season (usually harvest date)	YYYYDDD
	YRPLT	Planting date	YYYYDDD
Out-put	NH4(L)	Ammonium N in soil layer L	µg[N] / g[soil]
	NO3(L)	Nitrate in soil layer L	µg[N] / g[soil]

Volume 4
Appendix C

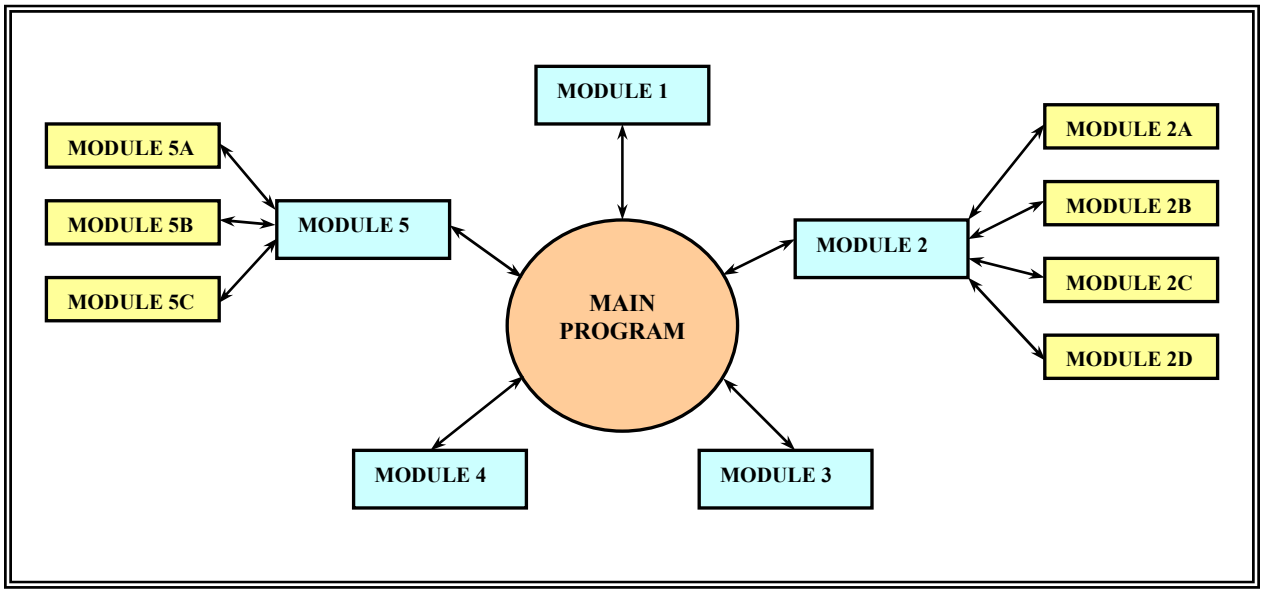
Module Structure

C.H. Porter
J.W. Jones

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

MODULE STRUCTURE

March 31, 2004



C.H. Porter and J.W. Jones

University of Florida, Gainesville, FL
International Consortium for Agricultural Systems Applications

Table of Contents

1.0 Introduction....1
2.0 Module Definition and Structure....1
3.0 Programming Guidelines....3
4.0 Modifications to main program....4
5.0 References....4

List of Figures

Figure 1 - Modular Structure....5
Figure 2 - Module Processing within Main Program....6
Figure 3 - FORTRAN Code for Module Structure....7

1.0 Introduction

As new components are added to crop growth models to expand capabilities, the models have become increasingly complex. This has generated a need for a modular structure for the crop models so that new components can be added, modified and maintained with minimal effort. This modular approach will:

1) Facilitate the ability to integrate knowledge from different disciplines, thus improving the prediction capability of the models.
2) Allow contributions from many authors,
3) Allow greater flexibility in future updates to models, with modules being added, modified or replaced with little impact to the main program or other modules.
4) Extend the life and utility of simulation models.

A modular approach has been developed for the CSM model based on the methods used in the Fortran Simulation Environment (FSE) software (Kraalingen, 1995). The CSM model has undergone restructuring to the modular format described herein. Modules have been developed for phenology, plant growth and partitioning, photosynthesis, soil water balance, soil nitrogen balance, weather and pest damage functions.

The following sections describe this modular concept and it's implementation in the CSM model.

2.0 Module Definition and Structure

Reynolds and Acock (1997) and Acock and Reynolds (1989) proposed design criteria and rules for a generic modular structure for crop models. These include:

1) The modules should relate directly to real world components or processes that are common to all plants to be modeled.
2) Modules should represent a separate disciplinary function;
3) Input and output variables should represent measurable values;
4) Communication between modules should be solely via the input and output variables; and
5) Modules should be validated independently of other modules.

These criteria have been used in the decision of what components of the CSM model will comprise each module. Based on the first criteria, above, each module represents a physical function or process related to crop growth and consists of, as a minimum, a single FORTRAN subroutine, or a group of linked subroutines. These module processes are interdependent, but can be linked solely through a defined set of input and output variables, which are passed into and out of the modules as subroutine arguments.

The following guidelines, based on the approach of van Kraalingen (1995) and adapted by Kenig and Jones (1997) are proposed for the construction of modules for the CSM model. Each module should:

1) Read its own parameters;
2) Initialize its own variables;

3) Accept variables passed to it from other modules and the environment;
4) Pass variables that are computed within the module;
5) "Own" its set of state variables;
6) Compute rates of change for its state variables;
7) Integrate its state variables;
8) Write its own variables as output; and
9) Operate when linked to a dummy test program.

Thus, all data input, initialization of variables, rate calculations, integration calculations and output of data related to a specific function are handled within a single module. Modules can be run as a stand-alone model, when linked to a main driver program.

The criteria for generic modular structure developed by Reynolds and Acock and the specific guidelines for construction of a module were used in the development of the CSM modules. It was difficult to decide on how many modules to create. Our first thoughts were to create a few relatively large modules, using the criteria and principles above; for instance, four main modules: crop growth, soil water, soil nitrogen and pest damage. This division would allow separation of these major components so that each could easily be replaced by other modules similarly structured, with exact matching of input and output variables. However, it quickly became apparent that some of the modules may be components of larger system models or they may be made up of smaller modules, each of which could have the same criteria applied. Thus, we realized that the determination of modules is, to some extent, arbitrary.

We have thus attempted to base our initial decisions on (1) whether each module makes sense physically and physiologically, (2) whether it may be beneficial in the future to replace this module with another as a unit, and (3) whether our current model approach would lend itself to such modularization.

Figure 1 illustrates the modular format used in the CSM model, in which each module has two or more of the following six components:
1) Run Initialization (RUNINIT)
2) Seasonal Initialization (SEASINIT)
3) Rate Calculations (RATE)
4) Integration (INTEGR)
5) Output (OUTPUT)
6) Final Calculations (FINAL)

The main program (CSM.FOR) controls the timing for simulation, season and daily loops and calls the Land Unit Module (LAND) six times for each of the timing components listed above. LAND, in turn, makes up to six calls to each module to accomplish each component of processing. Control of processing within the program is regulated with the DYNAMIC variable. Each module is called once at the beginning of simulation with DYNAMIC set equal to 'RUNINIT', resulting in execution of the 'Run Initialization' portion of the module. At the beginning of each season of simulation, a call is made to each module to perform 'Seasonal Initialization' (DYNAMIC = 'SEASINIT'). During

the daily time loop, each module is called three times: once each for rate calculations (DYNAMIC = 'RATE'), integration calculations (DYNAMIC = 'INTEGR') and writing computed data to output files (DYNAMIC = 'OUTPUT'). A final call to each module is made to write summary output files and to close input and output files (DYNAMIC = 'FINAL') at the end of each seasonal simulation.

The FORTRAN code used for directing calls to modules from the main program is presented in Figure 2. Figure 3 lists typical FORTRAN code used to control processing within a module. The following sections describe the six divisions of modules:

1. **Run Initialization**. Modules are called during the run initialization phase to read data from the CSM input data files (IBSNAT35.INP, species file, ecotype file, pest file, etc.). Some of this input data could have been passed to the module by the calling routine, but instead, are read directly by the module or modules that use the data and are treated as local variables. This minimizes the need for COMMON blocks, while reducing the number of arguments passed between the main program and modules. Initialization or computation of variables that need be set only once per simulation is also done in this section.

2. **Seasonal Initialization**. At the beginning of each season of simulation, it is necessary to initialize many variables. The 'Seasonal Initialization' section is used to perform this function. The initial values of some state variables are read from input files in this section. For multi-season runs, this section is repeated at the beginning of each season of simulation, whereas the 'Run Initialization' section is performed only once.

3. **Rate Calculations**. Rate calculations are performed at the beginning of the daily time step loop. This ensures that rates of change of state variables for a given day of simulation are all based on values of state variables for a common point in time, i.e. the end of the previous day.

4. **Integration Calculations**. The integration portion of the model updates state variables throughout the model for each day of simulation using the values calculated in the 'Rate Calculations' portion of the module.

5. **Output**. In the 'Output' section, daily summary data is written to output files.

6. **Final**. The 'Final' section of processing is used to write end-of-season summary output files and to close output files.

3.0 Programming Guidelines

A general list of guidelines used in programming the modules and subroutines has been developed. These include:

- Eliminate GO TO statements, which make it difficult to follow code sequence. In most cases, GO TO statements can be replaced with IF-THEN-ELSE or DO-loop constructs.

- Eliminate COMMON blocks, which invisibly pass variables to and from routines. These can be replaced with argument lists, which explicitly specify the flow of data to and from modules and subroutines. Exceptions to this rule occur in some of the original output routines for CSM, which were maintained in the modular version of the model so that familiar output formats are preserved.
- Label input and output lists for each module and subroutines. This labeling has been done in the CSM modules by grouping variables in the argument lists by input or output function for the module or subroutine.
- Include a description of each module or subroutine as it is called within comment statements.
- Include a list of variable definitions at the end of each module or subroutine.
- Read input variables from files rather than pass to modules in an argument list. This reduces the number of arguments passed to and from modules and preserves the "independence" of individual modules.

4.0 Modifications to main program

As modules were added to the CSM model, the main program was modified to perform the appropriate calls to each module. With the addition of modules, the main program was used less for reading input data and initializing variables, thus many of the subroutines that were previously called from the main program were eliminated or called from modules.

For example, rather than initializing and reading phenology variables with calls to subroutines such as IPECO, INPHEN, IPIBS and IPCROP, the main program calls the PLANT module, with variable DYNAMIC = 'RUNINIT'. The PLANT module, in turn calls the phenology module, PHENOL, which performs the appropriate data input and initialization functions.

5.0 References

Acock, B., and Reynolds J.F., 1989. The rationale for adopting a modular generic structure for crop simulators. Acta Horticulturae 248:391-396.

Kenig, A., and J.W.Jones, 1997. Model structure for dynamic crop-greenhouse simulations. In: Seginer I., J.W. Jones, P. Gutman, and C.E. Vallejos (eds), Optimal environmental control for indeterminate greenhouse crops. Final Report, BARD Research Project IS-1995-91RC, Agricultural Engineering Department, Technion, Haifa, Israel. Chap. II-4.

Kraalingen, D.W.G. van, 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis Report no. 1. AB/DLO, PE, Wageningen.

Reynolds, J.F. and B. Acock. 1997. Modularity and genericness in plant and ecosystem models. In Ecological Modeling. 94 (1997) 7-16.

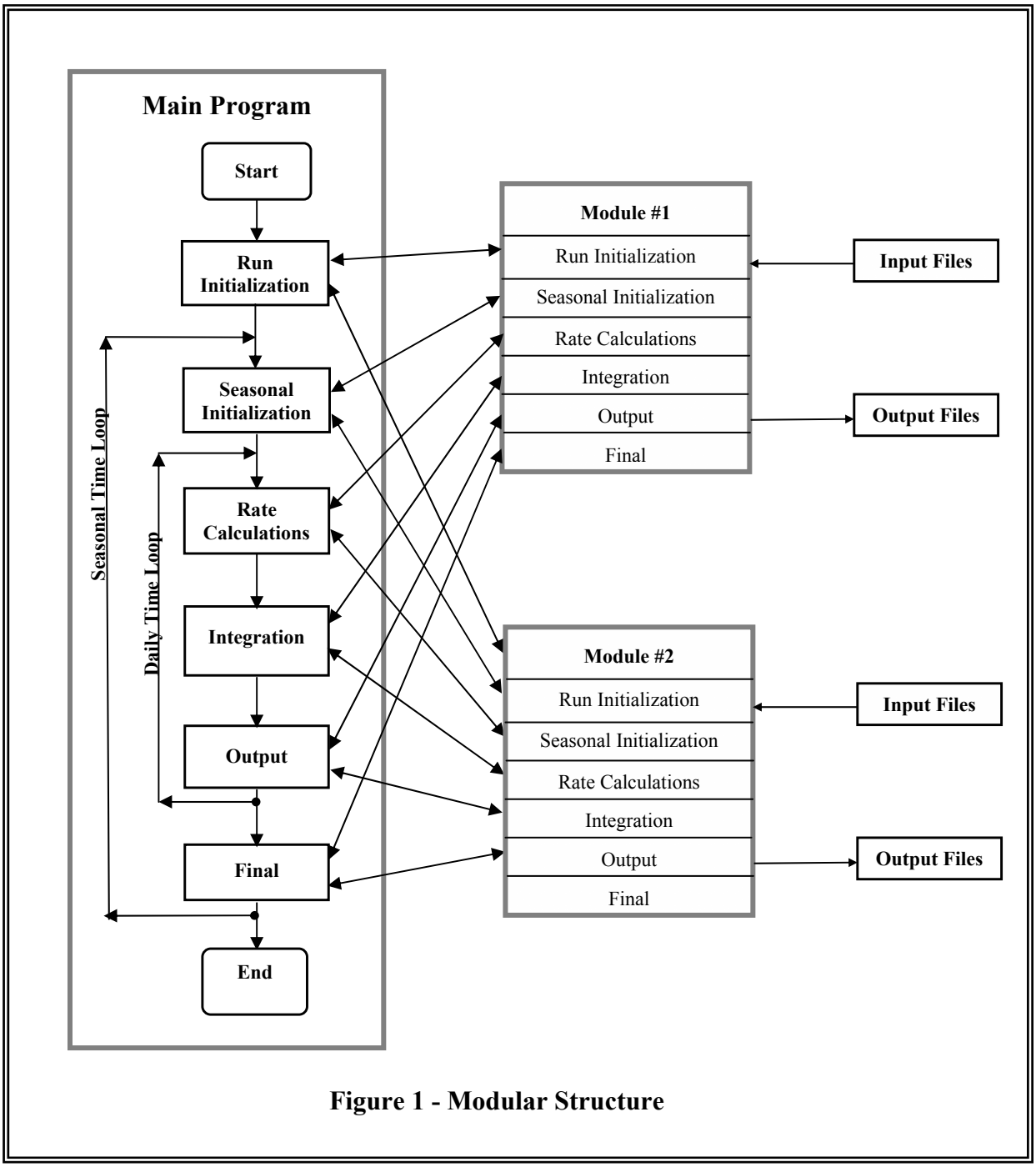


Figure 1 - Modular Structure

```
!=====================================================================
       PROGRAM DRIVER
!=====================================================================
       INTEGER, PARAMETER :: RUNINIT = 1, SEASINIT = 2, RATE = 3,
     &                       INTEGR = 4, OUTPUT =5, FINAL = 6
!=====================================================================
!      Input Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , RUNINIT)
       CALL MODULE2(arg1, arg2, . . . . , RUNINIT)
       .
       .
       .
!=====================================================================
!      Begin Seasonal Loop
!=====================================================================
!      Initialization Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , SEASINIT)
       CALL MODULE2(arg1, arg2, . . . . , SEASINIT)
       .
       .
       .
!=====================================================================
!      Begin Daily Loop
!=====================================================================
!      Rate Calculation Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , RATE)
       CALL MODULE2(arg1, arg2, . . . . , RATE)
       .
       .
       .
!=====================================================================
!      Integration Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , INTEGR)
       CALL MODULE2(arg1, arg2, . . . . , INTEGR)
       .
       .
       .
!=====================================================================
!      Output Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , OUTPUT)
       CALL MODULE2(arg1, arg2, . . . . , OUTPUT)
       .
       .
       .
!=====================================================================
!      End Daily Loop
!=====================================================================
!=====================================================================
!      Final Section
!=====================================================================
       CALL MODULE1(arg1, arg2, . . . . , FINAL)
       CALL MODULE2(arg1, arg2, . . . . , FINAL)
       .
       .
       .
!=====================================================================
!      End Seasonal Simulation Loop
!=====================================================================
!      End of Program
!=====================================================================
       END DRIVER
```

Figure 2 - Module Processing within Main Program