

DOCUMENTATION ON PHOSPHORUS MODULE V. 2.2

Argyrios Gerakis, [Samira Daroub](#), & [Joe T. Ritchie](#)

Michigan State University

Contact: gerakis at msu.edu

I. Disclaimer

The phosphorus module is not an official component of DSSAT 3.5. The authors developed this module on their own initiative to solve a specific problem. Until further testing, minor programming errors may exist.

II. Availability

The P versions of *CERES* and *CROPGRO* are available on the web at <http://nowlin.css.msu.edu/> or from Argyrios Gerakis, Dept. of Crop and Soil Sciences, Plant and Soil Sciences Bldg., Michigan State University, East Lansing, MI 48824-1325, USA (e-mail: gerakis@pilot.msu.edu).

III. Programming Environment

The P versions of *CERES* and *CROPGRO* were compiled with Digital Fortran 5.0 for Windows 95. It is possible to compile the models with Microsoft Fortran 5.1 but there may be memory limitations. When compiling with Microsoft Fortran 5.1, the following are recommended: In the makefile, make sure that option `/Gt64` is added to `FORFLAGS` and option `/SEG:512` is added to `LFLAGS`. If a subroutine is too big to fit in memory, compile it separately with the `/c` option, then link all subroutines with `nmake`.

The user can run the crop models outside the DSSAT shell. First, the models are run from within the shell to create file *ibsnat35.inp*. Then the name of the executable program is typed at the command line with five arguments:

```
>gecer980 ibsnat35.inp I I 1 1
```

```
>crgro980 ibsnat35.inp I I 1 1
```

This is the only way to debug the models with the Digital Fortran debugger or with *codeview*.

IV. Input - Output

DSSAT experiment details files: Make sure that the P switches for simulation and output are on. Make sure that the water balance switch is on, because the P module depends on information from the water balance.

Chemistry files: These are new files with extension *XXC*, where *XX* is the abbreviation of the crop (*MZ*, *SB*, etc.). They contain initial conditions for P and fertilizer applications. The model ignores P fertilizer information from the experiment details file.

Chemical.par: Contains rate constants of chemical transformations.

Phosphor.out: This is similar to the standard DSSAT output file for P, with more detail.

V. Common Blocks

We try to avoid using common blocks. Yet, the number of arguments that must be passed around exceeds the limit set in Microsoft Fortran 5.1. A new file called *chemical.blk* contains the variable declarations and the common blocks specific to the P module.

VI. Links to Crop Models

The P module receives the following information from the crop models *CERES* and *CROPGRO*: treatment number, current date, first day of simulation, soil bulk density (to compute mass of soil layers), number and thickness of soil layers, biomass (above and below ground, to compute potential tissue concentration), plant development stage (to compute uptake), planting density, and root length volume (to compute uptake and root senescence). The P module affects the crop models through the stress factor *ch_stress*, a 0 to 1 multiplier that affects carbon fixing (variable *carbo* in subroutine *grosb* in *CERES*, variable *pg* in subroutine *photo* in *CROPGRO*).

VII. Program Flow and Subroutines

The P module is as generic as possible so that it can be adapted easily for other chemicals. The major steps are in Figure 1, a decision making flowchart of subroutine *CHEM_GEN*, called by either *CERES* or *GRO*. The names of the main subroutines called by *CHEM_GEN* are in uppercase. The selection of initialization routine depends on the mode that DSSAT is run. Seasonal mode is when DSSAT simulates several seasons of crop growth independently, without any carryover of information from one season to the next. Sequence mode is when DSSAT simulates consecutive seasons of crop growth, with carryover of information from one season to the next. In each case, the initialization routines are slightly different.

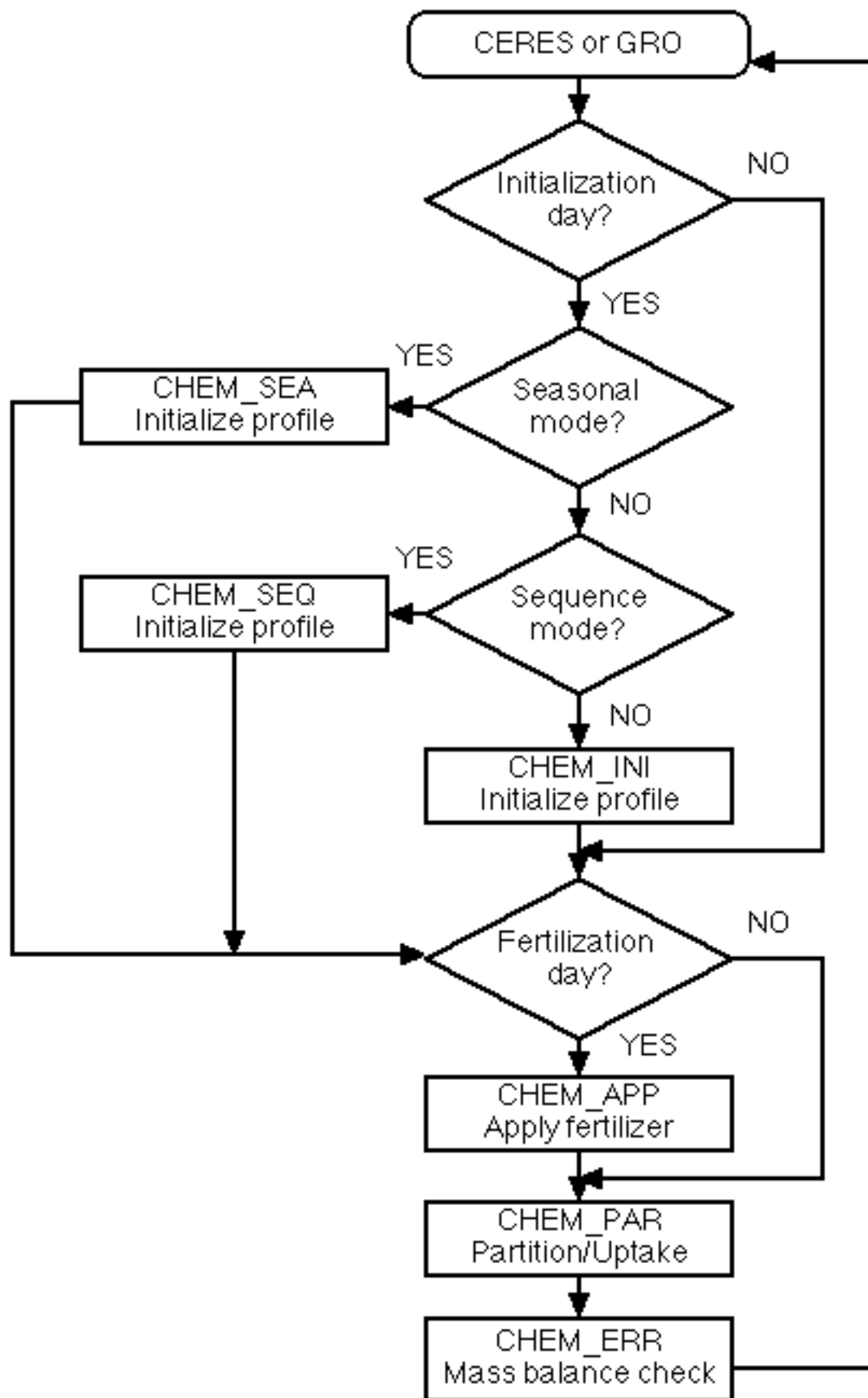


Figure 1. Decision making flowchart of subroutine CHEM_GEN.

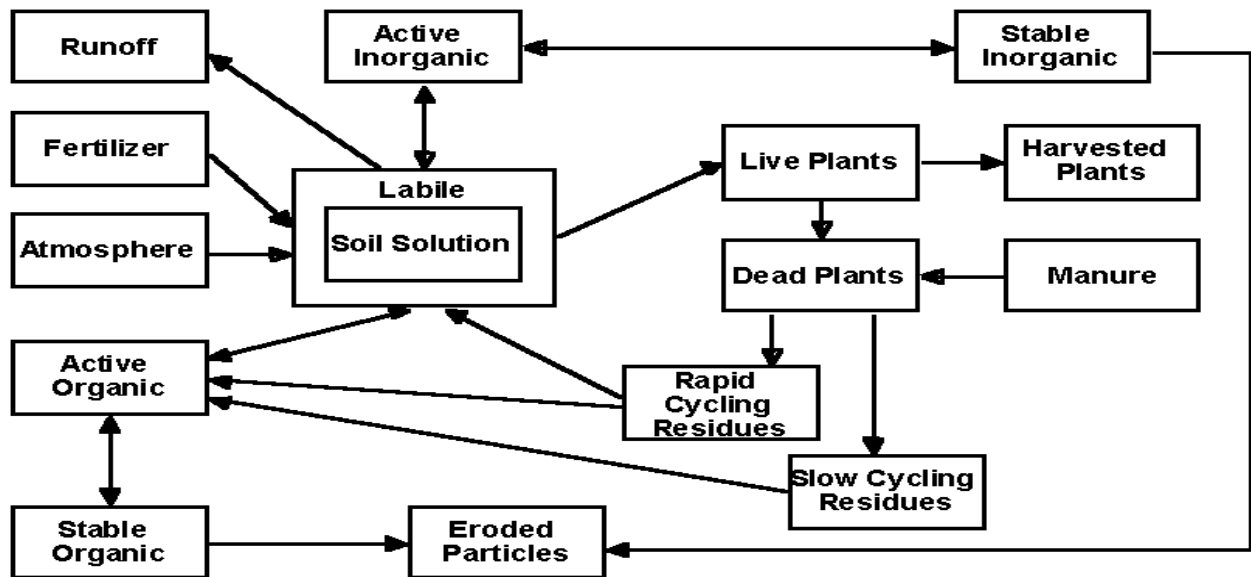


Figure 2. Mass transfer among P pools.

Chem_gen: The central P module. Calls *chem_ini*, *chem_app*, *chem_par*, *chem_err*, *clear*. On the first day of simulation it opens the input file, initializes variables by calling subroutine *chem_ini*. On the date of fertilization it calls subroutine *chem_app*. Then, P is partitioned into compartments in subroutine *chem_par*. Mass flow among the compartments or pools is according to Figure 2. Finally, at the end of the daily run, subroutine *chem_err* makes sure no mass is accidentally lost or created during the day. The total mass in the soil will be saved for next day's error check.

Chem_ini: Initializes P variables, calculates total P for mass balance. Reads P concentrations from the chemistry file in mg kg⁻¹ soil. At least the "Total" column in the file must have data. The columns correspond to fractions obtained during sequential P fractionation according to Hedley (1982). Then, the user-defined layer thicknesses in the soils file (*.sol) are converted to standard DSSAT thicknesses. Values of soil properties must be adjusted for the new layer thicknesses. This is done by subroutine *lmatch*. Then, P fertilizer information is read from the chemistry file (P fertilizer information from the experiment details file is ignored). If there are missing concentrations in the chemistry file, the program estimates them from literature on tropical soils (Ball and Coelho, 1993; Beck and Sanchez, 1994; Tiessen et al., 1992). Obviously, these estimates are approximate and good only for tropical soils. If the data are complete, then rows must add up horizontally to "Total." Diagnostics are printed to the screen. Next, the user may opt to estimate the labile P fraction from the soil P test. Several soil test options are available. After an option is selected, the program prints out the labile P estimated. If the value is unreasonable, the user may stop the program and start all over. Finally, the rest of the P pools are initialized. The user should not be too concerned at this point with P in runoff, eroded particles, or the atmosphere.

Chem_usr: User-defined initial conditions. Initializes the model using the matrix at the end of the chemistry file. Each pool becomes a weighted sum of the Hedley fractions. Make sure columns add up vertically to 1, except column "Total." Do not add up the columns that represent the minimum concentrations.

Chem_app: Applies P. Based on CERES subroutine *fplace*. Variable names have been changed to non-element-specific.

Chem_lyr: Integer type function to determine what layer the fertilizer is placed in. Based on CERES function *idlayr*.

Chem_par: Partitions P into pools. Calls *chem_dif*. If plants have emerged, plant uptake is calculated. Uptake is equal to either soil supply or plant demand, whichever is less. The soil supply is the weighted sum of labile P in every layer. The weight for each layer is a combination of the effect of root water uptake (*rwu*) and root density (*rlv*). Generally, the weight decreases with depth. This is meant to approximate the P supply from the point of view of the plant. Then the subroutine calculates the demand. We searched the literature for optimal P concentrations in maize (Jones, 1983), wheat and barley (Bauer et al., 1987a & b; Schulthess, 1992), soybeans (Hanway and Weber, 1971), chickpeas (Rashid and Bughio, 1993; Weber et al., 1992), millet, peanuts, and rice. We estimate the potential P concentration in plant tissue after the addition of new biomass on a given day:

$$ch_pot = ch_live / (dm + rtwt)$$

where *ch_pot* is potential concentration, *ch_live* is mass in live plants, and *dm* and *rtwt* are above and below ground biomass. The potential concentration is compared with the optimum concentration and demand is thus defined. A stress factor is calculated: 0 is too much stress, 1 is no stress. This factor is multiplied with the amount of potential carbon fixation for the next day. The effect is that in case of severe P deficit, growth rate will be reduced or stopped. After uptake is calculated, labile P in soil is reduced accordingly. Then P in senesced tissue is calculated. The mass of P in dying roots is proportional to the P in live tissue, and inversely proportional to the root density in a soil layer. Next, mass transfer from one P pool to the another is calculated as the concentration in the pool times a rate constant. This is done in subroutine *chem_dif* that calculates the change in mass of every pool and updates every pool.

Chem_dif: Calculates the delta (small difference) P for each pool, and updates pools.

Chem_err: Daily check for mass balance error. The mass of P in each pool is added in every layer. The total P is the sum of inorganic active, inorganic stable, organic active, organic stable, chemical in residues that decompose rapidly, chemical in residues that decompose slowly, chemical in fertilizer pool, labile, and chemical in live plants. The total P in the profile should be the total P of the previous day plus the gains (fertilizer), minus the losses in erosion, runoff, and in harvested tissue. If is there a mass balance error, the program prints a diagnostic message and stops.

VIII. Rotations

Users may simulate rotations where each crop uses the P left over from the previous crop. The executable program is *seqdrv.exe*. The data files are created follows:

1. Save a copy of an experiment details file with extension *sqx*.
2. Edit *sqx* file: At the top the of file, in **TREATMENTS*, under the 'N' column, enter the same number for all crops in the same rotation. Therefore, from treatment number this column becomes rotation number. Under the 'R' column, enter 1, 2, 3, etc. for each crop in the rotation. Edit factor levels in the matrix. In **SIMULATION CONTROLS*, in *MANAGEMENT*, under *HARVS* enter 'R' (harvest on reported date) for all crops. Then in **HARVEST DETAILS*, under *HDATE*, enter the harvest date. In

*SIMULATION CONTROLS, under *NYERS*, change the number of years to the total duration of sequence in years. If the specified number of years is less than the total duration of a rotation, then the model will repeat the sequence of crops, will not find the harvest date of the first crop, and will continue for as long as there are weather data. To avoid this, the harvest date of the last crop must be one day after the specified number of years under *NYERS* has passed (counting from the date of initial conditions).

3. Edit the chemistry information file for each crop: After the rotation starts, this file is used only to read fertilizer applications. In **P FERTILIZERS*, enter the application dates and amounts for the entire duration of the rotation.

IMPORTANT: The 'F' column is no longer the treatment number. This column now is the rotation number. So edit the 'F' column to be the rotation number. Lines with any other number are ignored.

4. Run *seqdrv.exe* from data directory. This driver will call *gecer980.exe* or *crgro980.exe* depending on the crop.

The way DSSAT works, debugging any but the first crop in the sequence requires several steps:

1. Delete file *tmp.dat*.

2. Run model for first crop (creates intermediate file *ibsnat35.inp*).

3. At the command line, type: *<executable of model> ibsnat35.inp I Q 1 1* (creates intermediate file *tmp.dat*).

4. Run model for second crop (creates intermediate file *ibsnat35.inp*). Copy this file in the directory with the rotation data, if different.

5. Run the debugger. In Digital Fortran, insert this argument in *Project -> Settings -> Debug -> Program arguments: ibsnat35.inp I Q 1 1*.

IX. Version History

1.0: Original version based on DSSAT 3.1.

2.0: Version based on DSSAT 3.5.

2.1: Changed the stress function to a non-linear one, modified the weight of the root length density on P supply, and added two new outputs in PHOSPHOR.OUT: Labile P by layer, and P uptake by layer.

2.2: Broke big partitioning routine into a generic routine and crop-specific routines. Fixed bug in sequencing so that a fallow treatment does not crash the program.

X. References

Ball-Coelho, B., I. H. Salcedo, H. Tiessen, and J. W. B. Stewart. 1993. Short- and long-term phosphorus dynamics in a fertilized Ultisol under sugarcane. *Soil Sci. Soc. Am. J.* 57:1027-1034.

Bauer, A., A. B. Frank, and A. L. Black. 1987a. Aerial parts of hard red spring wheat. III. Nitrogen and phosphorus concentration and content in kernels, anthesis to ripe stage. *Agron. J.* 79:859-864.

- Bauer, A., A. B. Frank, and A. L. Black. 1987b. Aerial parts of hard red spring wheat. II. Nitrogen and phosphorus concentration and content by plant development stage. *Agron. J.* 79:852-858.
- Beck, M. A., and P. A. Sanchez. 1994. Soil phosphorus fraction dynamics during 18 years of cultivation on a Typic Paleudult. *Soil Sci.* 34:1424-1431.
- Hanway, J. J., and C. R. Weber. 1971. N, P, and K percentages in soybean (*Glycine max* (L.) Merrill). *Agron. J.* 63:286-290.
- Hedley, M. J., J. W. B. Stewart, and B. S. Chauhan. 1982. Changes in inorganic and organic soil phosphorus fractions induced by cultivation practices and by laboratory incubations. *Soil Sci. Soc. A. Proc.* 46:970-976.
- Jones, C. A. 1983. A survey of the variability in tissue nitrogen and phosphorus concentrations in maize and grain sorghum. *Field Crops Research* 6:133-147.
- Rashid, A., and N. Bughio. 1993. Evaluating internal phosphorus requirement of rapeseed, chickpea, lentil, and wheat by seed analysis. *Commun. Soil Sci. Plant Anal.* 24:1359-1369.
- Schulthess, U. 1992. The impact of vertisol management systems on wheat production in the Ethiopian highlands. Doctoral diss. Swiss Federal Institute of Technology, Zürich.
- Tiessen, H., I. H. Salcedo, and E. V. S. B. Sampaio. 1992. Nutrient and soil organic matter dynamics under shifting cultivation in semi-arid northeastern Brazil. *Agric., Ecos. & Env.* 38:139-151.
- Weber, E., E. George, D. P. Beck, M. C. Saxena, and H. Marschner. 1992. Vesicular-arbuscular mycorrhiza and phosphorus uptake of chickpea grown in Northern Syria. *Expl. Agric.* 28:433-442.