MOTOR 2.0: Module for transformation of organic matter and nutrients in soil

MOTOR 2.0: Module for transformation of organic matter and nutrients in soil

User guide and technical documentation

F.B.T. Assinck
C. Rappoldt

Alterra-Rapport 933

Alterra, Wageningen, 2004

ABSTRACT

Assinck, F.B.T. & C. Rappoldt, 2004. *MOTOR 2.0: Module for transformation of organic matter and nutrients in soil, User guide and technical documentation*. Wageningen, Alterra, Alterra-Rapport 933. 47 pp. 6 figs.; 1 tables; 7 refs.

MOTOR is a MOdule describing the Transformation of Organic matteR and nutrients in soil. It calculates the transformations between pools of organic matter and mineral nitrogen in soil. Pools are characterized by a carbon and nitrogen content and can be labelled. MOTOR is a flexible tool because the pools are not fixed. They are specified, together with a list of transformations and additions, in a datafile. This user guide and technical documentation distinguishes two versions of the program. The stand-alone version of MOTOR is most suitable for evaluating laboratory experiments. The second version of MOTOR is coupled to the model FUSSIM2, this simulation model calculates water movement, solute transport and root uptake.

Keywords: FUSSIM2, immobilization, mineralization, model, MOTOR, nitrogen, nutrients, organic matter, pool, simulation

ISSN 1566-7197

© 2004 Alterra
P.O. Box 47; 6700 AA Wageningen; The Netherlands
Phone: + 31 317 474700; fax: +31 317 419000; e-mail: info@alterra.wur.nl

Alterra assumes no liability for any losses resulting from the use of the research results or recommendations in this report.

[Alterra-Rapport 933/maart/2004]

Contents

Preface

About a decade ago Andy Whitmore (now at Silsoe Research Institute, UK) developed a computer model for calculating the transformations of organic matter and nutrients in soil (MOTOR). Based on the concepts of Andy Whitmore's computer model, Kees Rappoldt and Klaas Harmanny developed a version of MOTOR, which came up to the requirements of a FSE (Fortran Simulation Environment, van Kraalingen, 1995) model. With this version of MOTOR communication with other agro-ecological models (also developed in FSE) is relatively easy. Marius Heinen and Peter de Willigen incorporated the FSE version of MOTOR in their model FUSSIM2, which describes water movement, solute transport and root uptake of water and solutes in porous media (Heinen & de Willigen, 2001). This report describes both the FSE stand-alone version and the coupled to FUSSIM2 version of MOTOR.

The authors are grateful to everyone, who helped in the development of MOTOR and to Peter de Willigen for helpful comments on the manuscript.

Summary

MOTOR is a **MO**dule describing the **T**ransformation of **O**rganic matte**R** and nutrients in soil. It calculates the transformations between pools of organic matter and mineral nitrogen in soil. Pools are characterized by a carbon and nitrogen content and can be labelled with C14 and N15 (in the stand-alone version only). MOTOR is a flexible tool because the number and names of the pools are not fixed. They are specified, together with a list of transformations and additions, in a datafile.

This user guide and technical documentation distinguishes two versions of the program. The stand-alone version of MOTOR is most suitable for evaluating laboratory experiments with small amounts of soil. The second version of MOTOR is coupled to the model FUSSIM2. MOTOR calculates mineralization and immobilization. FUSSIM2 calculates water movement, solute transport and root uptake of water and solutes.

1 Introduction

MOTOR is a **MO**dule describing the **T**ransformation of **O**rganic matte**R** and nutrients in soil. It calculates the transformations between pools of organic matter and mineral nitrogen in soil. Each pool is characterized by a carbon content and a nitrogen content. The pools can be labelled, with C14 and N15 respectively. The number of pools and the names of the pools are not fixed. They are specified in a data file together with a list of transformations from one pool into another. Hence, to a large extend, the datafile specifies the actual organic matter model. The program just executes it.

MOTOR is a flexible tool for studying various models of organic matter transformations, including their implications for C14 and N15 fractions of the various pools (stand-alone version only). It has been developed originally to evaluate the results of laboratory experiments with small amounts of soil to which labelled organic matter is added. As a consequence, sinks for mineral nitrogen like root uptake or denitrification are lacking. Multi-year runs show an ever increasing amount of mineral nitrogen. In reality, this nitrogen is taken up by plants or disappears otherwise. Hence, in case of long-term organic matter studies with MOTOR, sink terms will have to be added. However, MOTOR can also be used in combination with the model FUSSIM2 (Heinen & de Willigen, 2001). In this combination MOTOR calculates mineralization and immobilization, and FUSSIM2 calculates water movement, solute transport and root uptake of water and solutes.

MOTOR is a Fortran subroutine that calculates rates of change. It is called by the Runge-Kutta simulation driver RKDRIV (Rappoldt & van Kraalingen, 1990). This driver organizes the model run and integrates the state variables. It calls the MOTOR subroutine in order to initialize the model, to get rates of change belonging to a certain system status or to terminate the model.

The program has been written in Fortran 77 with some generally accepted extensions. All these extensions are part of the Fortran 90 standard. For input from file and output to file, the program heavily relies on the utility library TTUTIL (van Kraalingen & Rappoldt, 2000).

In chapter 2 the structure of the model is sketched. Chapter 3 and 4 deal with the input and output files respectively. Chapter 5 describes the different versions of MOTOR, that is the stand-alone FSE version and the version coupled with FUSSIM2. In chapter 6 some test results and examples are presented. In Appendix A essential parts of the Fortran program are explained in more detail.

2 Outline of the model

2.1 Carbon pools and Transformations

The number of organic matter pools is specified by the user in a datafile. Each pool is characterized by a carbon content and a nitrogen content. Part of the carbon content can be present in the form of C14 and part of the nitrogen content can be N15. Transformation processes move material from one pool to another. These transformation processes are also specified by the user in the datafile. Hence, the model is very much a bookkeeping program, just calculating the behaviour of pools as a result of the transformations. Figure 2.1 gives an example of an organic matter model.

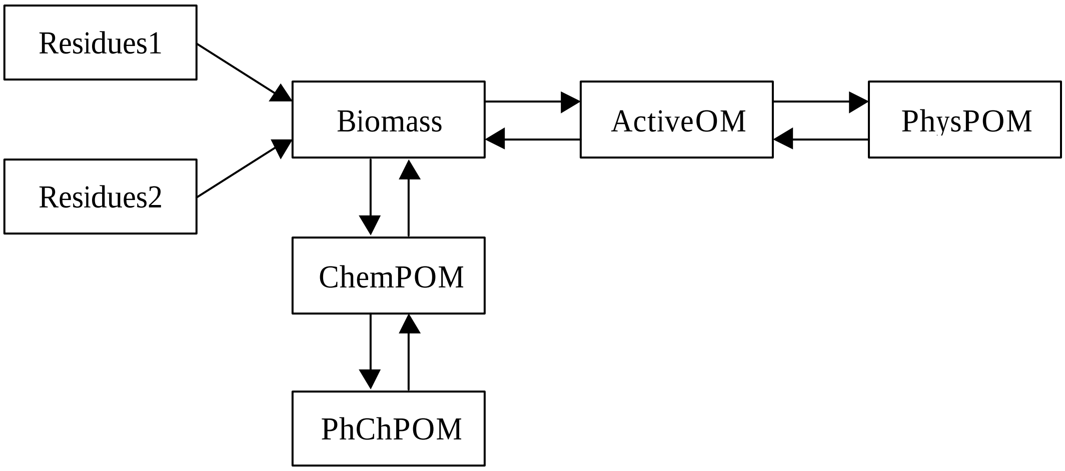

Figure 2.1. Example of an organic matter model.

Transformation processes are driven by the use and the formation of carbon. In principle, the nitrogen just moves with the carbon. The nitrogen content of a product pool may be smaller than the content of the source material. In that case, the excess of nitrogen is moved to the mineral nitrogen pool. A transformation product, however, may also contain more nitrogen than the source pool. In that case the additional nitrogen is taken up from the pool of available mineral nitrogen.

Organic matter calculations often involve addition of fresh material at certain moments in time. A list of additions can be specified in the input file. The times of addition can be given as an ordinary value of TIME, as a combination of year and day number or as a daynumber with automatic repetition after 365 days. This last option allows “climate runs”, the calculation of long-term behaviour of the organic matter content of the soil.

2.2 The mineral nitrogen pool

The pool of mineral nitrogen requires some attention. As explained above, some transformations require additional nitrogen, others produce mineral nitrogen, depending on the C/N ratio and efficiency of their respective source and product pools. In case of a transformation process requiring additional nitrogen the explanation is easy. Some of the available mineral nitrogen is taken up, as long as there is any (see Appendix A). In case of a transformation process leading to mineralization things are more complicated. Is the mineral nitrogen produced immediately available to the nitrogen demanding processes? Or should the nitrogen produced be simply added to (and mixed with) the mineral nitrogen pool?

For the amounts this makes no difference, however for the N15 fraction of the pools it does. If the source pool of a transformation is labelled and the produced (labelled) mineral nitrogen is immediately used in other transformations, the N15 will not come in the mineral pool. It appears only in that pool if there is a nett production of mineral N by the entire system. On the other hand, if produced nitrogen is added to the mineral pool, the mixing time has to be specified. The original model of Whitmore used a fixed time step of 1 day and the amount produced during a day was mixed with the mineral pool. This implicitly defines a mixing time of about a day. In a variable time step model this mixing time has to be made an explicit parameter.

In the datafile of MOTOR the variable **AvailPartProdNmin** specifies which part of the mineral nitrogen produced by a transformation is immediately available to nitrogen demanding convertions (NOT via the mineral N pool in the soil). The non-available part is added to the so-called hidden mineral N pool. The hidden mineral N pool has been defined only as a way to specify the mixing rate of "fresh" nitrogen with the existing mineral pool. The hidden pool simply empties into the ordinary, available mineral pool according to the time constant **HATimeConstant**. By specifying a value of, say, 1 day the content of the hidden pool will always be small and the model result will be insensitive to the precise value of the mixing time. A very small value like 0.01 day will not make much difference, but will hamper the numerical integration.

In case of a net demand of mineral nitrogen, the available pool may eventually reach zero. The moment in time, at which this happens will usually not correspond with a time reached in an ordinary time step. This is a well known type of problem in simulation and it is solved by generating a so called state event. A state event is a special status of the system, in this case a zero value of the available mineral nitrogen pool. By means of an iterative procedure, the simulation driver RKDRIV determines the time of the event (see Appendix A for details). From that time onwards, the demand of available mineral nitrogen must be kept equal to the supply. Mineral nitrogen demand can be reduced by reducing the transformation rates of nitrogen consuming processes, by increasing CO_2 production or by increasing the CN ratios of the products formed. This is discussed in more detail in section 3.3.6 and Appendix A.

2.3 Soil respiration

Except the mineral nitrogen pools, there is also a mineral carbon pool. This pool just contains the cumulative amount of CO_2 formed since the beginning of the simulation. With help of this pool, the overall carbon balance can be verified. Like the carbon and nitrogen organic matter pools, also the mineral pools can be enriched with C14 and N15.

3 The input files

The number of input files varies with the used version of MOTOR. The stand-alone version requires two input files, that is TIMER.DAT and MOTOR.DAT. The names of these files are obligatory. The version, coupled with FUSSIM2, requires two obligatory input files, that is MOTORRK.DAT and MOTORTYPE.DAT. MOTORTYPE.DAT specifies at least one additional input file, for example MOT10.DAT.

Basically TIMER.DAT and MOTORRK.DAT are equal. This also applies to MOTOR.DAT and MOT10.DAT, both specify the organic matter pools and the transformations that should be simulated. All input files are meant for reading with the TTUTIL input routines (van Kraalingen & Rappoldt, 2000). The sections below correspond to the various input files.

3.1 Timer.dat

The remarks on TIMER.DAT also apply to MOTORRK.DAT. These input files are primarily meant for the simulation driver and contain timing information and accuracy settings.

It is beyond the scope of this report to explain in detail the contents of these files. Changing the values of variables, other than the timer and weather control variables, is not advised, unless you are an experienced user. The timer and weather control variables are standard FSE variables and are controlled by FUSSIM2 in the coupled version of MOTOR.

3.2 Motortype.dat

This file is only used in the version of MOTOR coupled with FUSSIM2. FUSSIM2 is 2-dimensional computer model. The considered flow domain in FUSSIM2 is devided in control volumes (CV). MOTORTYPE.DAT specifies for each CV, which additional input file is to be used. Each additional input file specifies the organic matter model belonging to the CV(‘s). Listing 3.1 is an example of MOTORTYPE.DAT.

Listing 3.1. Example of MOTORTYPE.DAT.

```
* give names of different input files
FTYPES = 'MOT10.dat','MOT20.dat'

* give code (1 thru number of given FTYPES) per CV
* first NC values for layer 1, then NC values for layer 2, etc.
ITYPES = 2*1, 2*2, 6*0

9
```

The file contains the following variables:

Ftypes gives the names of the input files
Itypes indicates which input file belongs to which CV.

In listing 3.1 inputfile MOT10.DAT belongs to the first two CV's and MOT20.DAT to the next two CV's. For the remaining 6 CV's no input file (value = 0) is specified. This means that no organic matter calculations are performed for these CV's, because for example there is no organic matter present in these CV's.

3.3 Motor.dat

The remarks on MOTOR.DAT also apply to files, given by the variable Ftypes whenever the coupled version of MOTOR is used. The file MOTOR.DAT specifies the organic matter pools and the transformations that should be simulated. The sections below correspond to the various parts of the data file MOTOR.DAT.

3.3.1 The pool table

The pool table specifies the names of the organic matter pools used together with their initial amounts. Listing 3.2 gives an example with seven organic matter pools. The various columns of the table are read by the program as arrays with a common length, the number of pools found in the table. The maximum number of pools which can be handled by the program is 20. The minimum number of pools is 1. In that case the only transformation possible is a transformation of the pool into itself.

Listing 3.2. Example of a pool table in MOTOR.DAT

```
Substrate  CarbonInit FibreFr NitrogenInit CNratioInit C14InitEF N15InitEF
*---------* *--------* *-----* *----------* *---------* *-------* *-------*
'Residues1'       0.   0.0000       0.0         0.0       0.00      0.00
'Residues2'       0.   0.0000       0.0         0.0       0.00      0.00
'Biomass  '     100.   0.0000      16.6         6.0       0.00      0.00
'ActiveOM '     100.   0.0000       8.0        10.0       0.00      0.00
'PhysPOM  '    1000.   0.0000     100.0        12.0       0.00      0.00
'ChemPOM  '    4500.   0.0000     180.0        23.0       0.00      0.00
'PhChPOM  '   12000.   0.0000     550.0        24.0       0.00      0.00
```

The pool table contains the following columns:

Substrate The name of the pool. In listing 3.2 the abbreviation (P)OM means (Protected) Organic Matter. Pool names may have up to 27 characters.
CarbonInit The initial amount of carbon in the pool. The unit in the coupled version of MOTOR must be mg kg^{-1} soil.
FibreFr The fibre fraction of the pool. This fraction is optionally used in transformation equations.

NitrogenInit The initial amount of nitrogen. This column may be left out. In that case the amount of nitrogen is calculated from CNratioPool, the next column.

CNratioPool The C/N ratio belonging to the pool. The transformations can be set up in such a way that material formed into a pool has this CN ratio.

C14InitEF The initial C14 enriched fraction of the carbon pool (only useful in the stand-alone version).

N15InitEF The initial N15 enriched fraction of the nitrogen pool (only useful in the stand-alone version).

Note that the input procedure used by MOTOR is not case sensitive. The uppercase characters used in the column names are meant to get more readable variable names only.

3.3.2 The transformation table

Listing 3.3 gives an example of a substrate transformation table. This table specifies how substrates are transformed into other substrates and into CO_2 and mineral nitrogen. Each line of the table specifies a transformation by means of parameter values and settings. A transformation means that the contents of some pool is transformed into the contents of another pool. In this sense, the transformations form connections between the pools. The pools may be connected by a entire network of transformations, or just by a linear series of transformations, changing for instance Pool1 into Pool2 and Pool2 into Pool3 etc.

Listing 3.3. Example of pool transformations specified in MOTOR.DAT

```
  SubUsed     SubFormed   RConstant Eff Order Adjust KeepCN
*---------*  *---------*  *-------* *-* *---* *----* *----*
'ResiDues1'  'Biomass  '    0.12    0.6   2   'yyy'  .False.
'Residues2'  'Biomass  '    0.12    0.6   2   'yyy'  .False.
'Biomass  '  'ActiveOM '    0.006   1.0   2   'yy-'   .True.
'Biomass  '  'ChemPOM  '    0.002   0.6   2   'yy-'  .False.
'ActiveOM '  'Biomass  '    0.02    0.6   1   'yy-'  .False.
'PhysPOM  '  'ActiveOM '    0.0015  1.0   1   'y--'   .True.
'ChemPOM  '  'Biomass  '    2.0E-6  0.6   1   'yy-'  .False.
'PhChPOM  '  'ChemPOM  '    0.0015  1.0   1   'y--'   .True.
'ActiveOM '  'PhysPOM  '    0.015   1.0   1   'y--'   .True.
'ChemPOM  '  'PhChPOM  '    0.015   1.0   1   'y--'   .True.
```

A row of the table describes which substrate is transformed, what kind of material is made of it and how the nitrogen is handled. In order to explain the columns of the table we write the carbon content of some pool, for instance Residues1, as an amount as function of time $X(t)$. A transformation rate $R(t)$ is the amount of the pool transformed per unit of time, which is written as

$$R_t = R_{constant} X_t F_1 F_2 F_3 F_4, \qquad (1)$$

in which F_1, F_2, F_3 and F_4 are multiplicative factors reducing or enlarging the transformation rate.

The columns of the transformation table are

SubUsed The name of the substrate used. This must be one of the pool names given in the pool table.

SubFormed The name of the substrate produced. This must be one of the pool names given in the pool table.

RConstant The reaction constant in the transformation equation. This constant is modified by multiplicative factors. The unit is reciprocally as the time unit (t^{-1}). The time unit is usually day.

Eff The efficiency of product formation. Hence, in order to calculate the amount of product formed, the transformation rate is multiplied by the efficiency of the transformation. The carbon not appearing in the product pool is moved to the carbon mineral pool, the CO_2 pool, which just contains the cumulative amount of CO_2 formed since the beginning of the simulation.

Order The order of the transformation. The default equation above represents a first order transformation rate. If the order is 2, the transformation rate is multiplied by

$$F_1 = \frac{X_t}{X_{ref}}, \tag{2}$$

where X_{ref} is a reference amount. During initialization, the program sets the reference amounts of all pools equal to the initial amounts. Hence, initially F_1 is 1.0 and the reaction constant RConstant still is the fraction of the pool transformed per unit of time. With changing pool size, however, the second order transformation rates will be strongly enlarged or reduced by F_1.

Adjust This column enables or disables three adjustment factors by just giving a “y” (“Yes”) or any other character. The first flag causes the inclusion of a temperature adjustment factor calculated as (Jenkinson *et al.*, 1987)

$$F_2 = \frac{47.9}{1.0 + e^{\left(\frac{106.0}{Temperature + 18.3}\right)}}. \tag{3}$$

The second flag is a water content adjustment factor calculated according to Bradbury *et al.* (1993) as

$$F_3 = 1.0 - \frac{(1.0 - W_{\min})(Deficit - Onebar)}{D_{\max} - Onebar}, \tag{4}$$

with a maximum value of 1.0. The variables Wmin, Deficit, OneBar and Dmax have to be present on the input file in the stand-alone version or are determined by FUSSIM2 in the coupled version.

The third flag is an adjustment factor for the fibre content of the substrate calculated as

$$F_4 = e^{\left(-Parameter\frac{X_{ref}FibreFr}{X_t}\right)}, \tag{5}$$

in which the parameter is given in the input file as the variable '`FibreParameter'`, the variable FibreFr is the fibre fraction of the substrate pool specified in the pool table (Section 3.3.1), X(t) is the current carbon content of the substrate pool and X_{ref} its reference size.

For example, the string '`-y-`' causes the inclusion of the water adjustment factor in the expression for the transformation rate. The other factors are left out. With '`yyy`' all three adjustment factors are applied.

KeepCN If this flag is set .TRUE., the nitrogen of the substrate pool is just moved into the product pool. This option is used for physical transformations like absorption. The program accepts the .TRUE. KeepCN flag only if the efficiency of the transformation is 1.0. Hence, no carbon dioxide can be produced by such a transformation. If the KeepCN flag is .FALSE., the CN ratio of the product pool specified in the pool table (in column CNratioPool) is used to determine the nitrogen content of the material formed.

3.3.3 The substrate addition table

The substrate addition table specifies moments in time at which the program has to add material to a pool (see Appendix A). This facility will usually be used to add plant residues to one or more residue pools, but it can also be used to add material belonging to any of the other pools or available mineral nitrogen.

The entire substrate addition table is optional. If the variable AddToPool does not exist on MOTOR.DAT, the addition table is not read.

Listing 3.4 gives an example of a table with two rows. The first row specifies an addition to the pool 'Residues1' at time 180.5. The second row is a mineral nitrogen application at time 80.0. For a mineral nitrogen application the carbon amounts in the table have to be coded as missing values (with "-"). Note that the pool 'MineralN' is not defined in the pool table. Several additions can be specified for the same time value. The time values in the table need not to be ordered. The various addition times are sorted by the program and values outside the simulated time interval [STTIME, FINTIM] lead to a warning.

Listing 3.4. A substrate addition table in MOTOR.DAT. This table specifies times and amounts of the desired additions.

```
AddTime  AddToPool  AddCarbon AddNitrogen AddC14 AddN15 AddFibreFr
*-----* *---------* *-------* *---------* *----* *----* *--------*
 180.5  'Residues1'  2500.0      50.0      0.0     0.0   0.1761
  80.0  'MineralN '     -        40.0       -     20.0     -
```

The additions are realized by initiating so called time events. At the specified times special calls to the model are done by the driver during which the requested pools are updated. The meaning of the columns in listing 3.4 is self evident. The amount in column AddC15 should be less or equal to the amount in column AddCarbon, the amount in AddN15 should be less or equal to the one in AddNitrogen.

Except with a column AddTime, addition times can also be specified by means of columns AddInYear and AddDOY. These columns specify a year and a Day Of Year (daynumber) at which the addition takes place. Listing 3.5 specifies an addition in the Residues1 pool in 1985 and a mineral nitrogen addition in 1986.

Listing 3.5. A substrate addition table in MOTOR.DAT. This table specifies the addition times as combinations of a year and a day number.

```
AddInYear AddDOY   AddToPool  AddCarbon AddNitrogen AddC14 AddN15 AddFibreFr
*-------* *-----* *---------* *-------* *---------* *----* *----* *--------*
  1985    180.0   'Residues1'   2500.0      50.0      0.0     0.0   0.1761
  1986     80.0   'MineralN '      -        40.0       -     20.0     -
```

The third way of specifying addition times is the use of the year 1000. In the FSE simulation environment this year number is used for average weather data or climate data. If additions are specified for year 1000, they are repeated every 365.0 days. This facility allows so called climate runs in which the long-term behaviour of organic matter is simulated.

Listing 3.6. A substrate addition table in MOTOR.DAT. This table specifies the addition times as for the year 1000 which is used for climate data.

```
AddInYear AddDOY   AddToPool  AddCarbon AddNitrogen AddC14 AddN15 AddFibreFr
*-------* *-----* *---------* *-------* *---------* *----* *----* *--------*
  1000    180.0   'Residues1'   2500.0      50.0      0.0     0.0   0.1761
  1000     80.0   'MineralN '      -        40.0       -     20.0     -
```

The three different methods of specifying addition times cannot be mixed. The units of the added amounts must be mg kg^{-1} soil in the coupled version of MOTOR.

3.3.4 Mineral pools

The pool of available mineral nitrogen is controlled by the following variables on MOTOR.DAT.

AvailPartProdNmin The available part of produced mineral nitrogen. The remaining part is moved to the pool of hidden mineral nitrogen.

NminAvInitial The initial amount in the available pool of mineral nitrogen. In the coupled version of MOTOR this amount is calculated by FUSSIM2.

NminAvInitialEF The initial enriched fraction of the available mineral nitrogen pool.

HATimeConstant Indicator for mixing time. The hidden pool is emptied into the available mineral pool according to this time constant.

Also the pool of hidden mineral nitrogen can be initialized. However this is optional, which means that these variables need not to be present.

NminHdInitial The initial amount in the hidden pool of mineral nitrogen.

NminHdInitialEF The initial enriched fraction of the hidden pool.

Also the following variable is optional.

NminEventScale This amount sets the scale used by MOTOR in the state event generation. The default value of this variable is 10.0. An amount X of mineral N is regarded as zero if the ratio X/NminEventScale is less than the state event tolerance level SEVTOL on file TIMER.DAT.

3.3.5 PPOM saturation

Hassink & Whitmore (1997) describe the saturation of physically protected organic matter pools (PPOM pools). In order to simulate this effect, the user can optionally specify a list of pools to be regarded as PPOM pools together with a saturation level. These are the following variables on the input file

PPOMSatList This variable gives a list of PPOM pools, for instance

PPOMSaturationLevel This is the maximum value of the sum of the carbon contents of the listed pools. The transformations rates producing PPOM material are reduced by a factor F_5 given by

$$F_5 = 1.0 - \frac{\sum CarbonInPPOMpools}{PPOMSaturationLevel}, \qquad (6)$$

If, as a result of substrate addition, or by program failure, the sum of PPOM carbon pools exceeds the saturation level, simulation is halted.

3.3.6 Mineral N demand reduction

The reduction of mineral N demand takes place by reducing the transformation rate of nitrogen requiring processes, by reducing the efficiency of these processes or by reducing the nitrogen content of the formed products. If reduction of the mineral N occurs, demand takes place in steps.

The following input variables give the stepsize for the three reduction strategies and the lower bound of the final reduction. The reduction stepsizes are numbers somewhat below 1.0, say 0.97 or 0.90. The reduction limits are numbers between 0.0 and 1.0. A limit of 1.0 implies that the reduction method is not used.

RateRedStep	Reduction step used for the transformation rate
EffRedStep	Reduction step used for the efficiency
NCRatRedStep	Reduction step used for the NC ratio of the produced material
RateReductionLimit	lower bound of transformation rate reduction
EffReductionLimit	lower bound of efficiency reduction
NCRatReductionLimit	lower bound of NC ratio reduction

If the nitrogen demand has to be reduced below the level specified by the three limiting values, the simulation is halted.

4 The output files

There are basically three output files involved with MOTOR. The file RES.DAT gives ordinary formatted output. The file RES.BIN contains the numbers in RES.DAT in binary form and can be used to make graphs with the utility TTSELECT. Finally, the file MOTOR.LOG is a logfile. The contents of the logfile depend on the setting of the TRACE variable in the input file TIMER.DAT.

4.1 Res.dat

Listing 4.1 gives an example of the formatted output file RES.DAT. For each output variable there is a column with the value as function of time. In the example a time event took place right at the beginning, at time 1.0, at which, material was added to the pool RESIDUES1. At time 2.06021, the mineral nitrogen pool was empty and from time 5.99531 that pool grows again. At the time and state events there are always two lines of output, one line at event time but before the event has taken place and one line after the event has taken place.

In the file RES.DAT extensions of pool names are used to denote the various pool properties,

poolname.C	the carbon amount
poolname.N	the nitrogen amount
poolname.CN	the C:N ratio
poolname.C14	the C14 amount
poolname.CEF	the C14 enriched fraction
poolname.N15	the N15 amount
poolname.NEF	the N15 enriched fraction

The file also contains the mineral pools, amounts and enriched fractions, and the nitrogen demand reduction factors which are below 1.0 in periods of nitrogen limitation. Finally the total amounts of C, N, C14 and N15 are given. These should be constant except at time events when additions take place.

The file RES.DAT is the result of the call's to subroutine OUTDAT in the output section of the program. It is not difficult to add other variables in the same way. The construction of the output table is done by means of a terminal call to OUTDAT by the driver. Hence, a runtime crash of the program prevents the construction of formatted output.

Listing 4.1. Example of output file RES.DAT

```
*------------------------------------------------------------------------
* Output table number  :  0 (=first output table)
* Output table format  : Table output
* Simulation results
*          = = = = = = = = = = = = = = =
*           = = = =   M O T O R   = = = =
*           = = = = = ========= = = = = =
*  ----------------------------------------------
* |    MOdule describing the Transformation of    |
* |      Organic matteR and nutrients in soil     |
* |                Version 1.90                   |
* |          AB-DLO Haren, August 1997            |
* |  Concepts: Andy Whitmore                      |
* |   Program: Kees Rappoldt and Klaas Harmanny   |
* -------------------------------------------------

      TIME     RESIDUES1.C  RESIDUES1.N  RESIDUES1.CN RESIDUES1.C14

   1.00000      0.00000      0.00000      0.00000       0.00000
   1.00000       2502.0       227.00       11.022       0.00000
   2.06021       1933.2       175.39       11.022       0.00000
   2.06021       1933.2       175.39       11.022       0.00000
   3.00000       1633.1       148.17       11.022       0.00000
   5.00000       1242.2       112.70       11.022       0.00000
   5.99531       1117.0       101.34       11.022       0.00000
   5.99531       1117.0       101.34       11.022       0.00000
   7.00000       1014.8       92.074       11.022       0.00000
   9.00000       865.95       78.566       11.022       0.00000
   10.0000       809.72       73.464       11.022       0.00000

      TIME     RESIDUES1.CEF RESIDUES1.N15 RESIDUES1.NEF RESIDUES2.C

   1.00000       0.00000       0.00000       0.00000      0.00000
   1.00000       0.00000        227.00        1.0000      0.00000
   2.06021       0.00000        175.39        1.0000      0.00000
   2.06021       0.00000        175.39        1.0000      0.00000
   3.00000       0.00000        148.17        1.0000      0.00000
   5.00000       0.00000        112.70        1.0000      0.00000
   5.99531       0.00000        101.34        1.0000      0.00000
   5.99531       0.00000        101.34        1.0000      0.00000
   7.00000       0.00000        92.074        1.0000      0.00000
   9.00000       0.00000        78.566        1.0000      0.00000
   10.0000       0.00000        73.464        1.0000      0.00000

      TIME     RESIDUES2.N  RESIDUES2.CN RESIDUES2.C14 RESIDUES2.CEF

   1.00000      0.00000      0.00000       0.00000       0.00000
   1.00000      0.00000      0.00000       0.00000       0.00000
   2.06021      0.00000      0.00000       0.00000       0.00000
   2.06021      0.00000      0.00000       0.00000       0.00000
   3.00000      0.00000      0.00000       0.00000       0.00000
   5.00000      0.00000      0.00000       0.00000       0.00000
   5.99531      0.00000      0.00000       0.00000       0.00000
   5.99531      0.00000      0.00000       0.00000       0.00000
   7.00000      0.00000      0.00000       0.00000       0.00000
   9.00000      0.00000      0.00000       0.00000       0.00000
   10.0000      0.00000      0.00000       0.00000       0.00000

      TIME     RESIDUES2.N15 RESIDUES2.NEF BIOMASS.C    BIOMASS.N

   1.00000       0.00000       0.00000       100.00       16.667
   1.00000       0.00000       0.00000       100.00       16.667
   2.06021       0.00000       0.00000       423.92       70.654
   2.06021       0.00000       0.00000       423.92       70.654
   3.00000       0.00000       0.00000       548.68       90.115
   5.00000       0.00000       0.00000       623.65       102.06
   5.99531       0.00000       0.00000       619.26       101.58
   5.99531       0.00000       0.00000       619.26       101.58
   7.00000       0.00000       0.00000       605.97       99.573
   9.00000       0.00000       0.00000       565.42       93.146
   10.0000       0.00000       0.00000       544.00       89.706

      TIME     BIOMASS.CN   BIOMASS.C14  BIOMASS.CEF  BIOMASS.N15  BIOMASS.NEF

   1.00000       6.0000       5.0000      5.00000E-02  0.00000      0.00000
   1.00000       6.0000       5.0000      5.00000E-02  0.00000      0.00000
```

```
 2.06021       6.0000       4.6766      1.10317E-02   49.466      0.70011
 2.06021       6.0000       4.6766      1.10317E-02   49.466      0.70011
 3.00000       6.0887       4.2228      7.69616E-03   70.504      0.78238
 5.00000       6.1105       3.2708      5.24453E-03   85.911      0.84175
 5.99531       6.0965       2.8840      4.65720E-03   86.797      0.85450
 5.99531       6.0965       2.8840      4.65720E-03   86.797      0.85450
 7.00000       6.0856       2.5566      4.21897E-03   85.961      0.86330
 9.00000       6.0702       2.0628      3.64830E-03   81.367      0.87354
 10.0000       6.0643       1.8779      3.45193E-03   78.626      0.87649

    TIME     ACTIVEOM.C   ACTIVEOM.N   ACTIVEOM.CN  ACTIVEOM.C14 ACTIVEOM.CEF

 1.00000       100.00       10.000       10.000      0.00000      0.00000
 1.00000       100.00       10.000       10.000      0.00000      0.00000
 2.06021       112.34       12.191       9.2148      0.23761      2.11508E-03
 2.06021       112.34       12.191       9.2148      0.23761      2.11508E-03
 3.00000       146.59       17.952       8.1653      0.56042      3.82313E-03
 5.00000       249.13       34.918       7.1347        1.1904     4.77831E-03
 5.99531       298.99       43.176       6.9250        1.4192     4.74670E-03
 5.99531       298.99       43.176       6.9250        1.4192     4.74670E-03
 7.00000       343.96       50.647       6.7913        1.5936     4.63322E-03
 9.00000       413.99       62.345       6.6403        1.8038     4.35706E-03
 10.0000       439.54       66.643       6.5955        1.8570     4.22496E-03

    TIME     ACTIVEOM.N15 ACTIVEOM.NEF PHYSPOM.C    PHYSPOM.N    PHYSPOM.CN

 1.00000      0.00000      0.00000       1000.0       83.333       12.000
 1.00000      0.00000      0.00000       1000.0       83.333       12.000
 2.06021       1.6008      0.13131       997.28       83.141       11.995
 2.06021       1.6008      0.13131       997.28       83.141       11.995
 3.00000       6.1879      0.34468       995.24       83.035       11.986
 5.00000       20.715      0.59325       993.12       83.172       11.940
 5.99531       28.060      0.64989       993.29       83.443       11.904
 5.99531       28.060      0.64989       993.29       83.443       11.904
 7.00000       34.823      0.68757       994.21       83.840       11.858
 9.00000       45.666      0.73247       997.88       84.931       11.749
 10.0000       49.755      0.74660       1000.4       85.592       11.688

    TIME     PHYSPOM.C14  PHYSPOM.CEF  PHYSPOM.N15  PHYSPOM.NEF  CHEMPOM.C

 1.00000      0.00000      0.00000      0.00000      0.00000       4500.0
 1.00000      0.00000      0.00000      0.00000      0.00000       4500.0
 2.06021      1.68048E-03  1.68507E-06  8.22735E-03  9.89562E-05   4480.3
 2.06021      1.68048E-03  1.68507E-06  8.22735E-03  9.89562E-05   4480.3
 3.00000      7.66812E-03  7.70478E-06  6.38942E-02  7.69488E-04   4468.3
 5.00000      3.61289E-02  3.63792E-05  0.48630      5.84695E-03   4452.0
 5.99531      5.67930E-02  5.71765E-05  0.87252      1.04565E-02   4445.4
 5.99531      5.67930E-02  5.71765E-05  0.87252      1.04565E-02   4445.4
 7.00000      8.07274E-02  8.11972E-05   1.3730      1.63761E-02   4438.6
 9.00000      0.13425      1.34537E-04   2.6453      3.11461E-02   4424.2
 10.0000      0.16264      1.62578E-04   3.3891      3.95959E-02   4416.3

    TIME     CHEMPOM.N    CHEMPOM.CN   CHEMPOM.C14  CHEMPOM.CEF  CHEMPOM.N15

 1.00000       195.65       23.000      0.00000      0.00000      0.00000
 1.00000       195.65       23.000      0.00000      0.00000      0.00000
 2.06021       194.70       23.011      4.86960E-02  1.08689E-05  8.50219E-02
 2.06021       194.70       23.011      4.86960E-02  1.08689E-05  8.50219E-02
 3.00000       194.10       23.021      0.11743      2.62807E-05  0.33684
 5.00000       193.21       23.042      0.26317      5.91126E-05   1.1871
 5.99531       192.84       23.052      0.32319      7.27021E-05   1.6480
 5.99531       192.84       23.052      0.32319      7.27021E-05   1.6480
 7.00000       192.46       23.062      0.37450      8.43723E-05   2.0979
 9.00000       191.67       23.082      0.45298      1.02387E-04   2.9019
 10.0000       191.25       23.091      0.48279      1.09320E-04   3.2516

    TIME     CHEMPOM.NEF  PHCHPOM.C    PHCHPOM.N    PHCHPOM.CN   PHCHPOM.C14

 1.00000      0.00000       12000.       500.00       24.000      0.00000
 1.00000      0.00000       12000.       500.00       24.000      0.00000
 2.06021      4.36683E-04   12023.       501.09       23.993      3.42017E-04
```

```
2.06021      4.36683E-04   12023.       501.09       23.993      3.42017E-04
3.00000      1.73544E-03   12043.       502.03       23.988      1.58456E-03
5.00000      6.14413E-03   12083.       503.98       23.976      7.72669E-03
5.99531      8.54620E-03   12103.       504.92       23.970      1.23663E-02
5.99531      8.54620E-03   12103.       504.92       23.970      1.23663E-02
7.00000      1.09002E-02   12123.       505.85       23.965      1.79076E-02
9.00000      1.51401E-02   12160.       507.66       23.954      3.09311E-02
10.0000      1.70017E-02   12179.       508.53       23.949      3.82003E-02

   TIME     PHCHPOM.CEF  PHCHPOM.N15  PHCHPOM.NEF  CMINPOOL     C14MINPOOL

1.00000      0.00000      0.00000      0.00000      0.00000      0.00000
1.00000      0.00000      0.00000      0.00000      0.00000      0.00000
2.06021      2.84473E-08  4.35165E-04  8.68438E-07   232.14      3.50521E-02
2.06021      2.84473E-08  4.35165E-04  8.68438E-07   232.14      3.50521E-02
3.00000      1.31580E-07  3.43467E-03  6.84153E-06   367.47      9.01429E-02
5.00000      6.39448E-07  2.71552E-02  5.38815E-05   558.50      0.23178
5.99531      1.02174E-06  4.95804E-02  9.81941E-05   624.95      0.30441
5.99531      1.02174E-06  4.95804E-02  9.81941E-05   624.95      0.30441
7.00000      1.47720E-06  7.93710E-02  1.56905E-04   681.71      0.37669
9.00000      2.54357E-06  0.15822      3.11665E-04   774.14      0.51524
10.0000      3.13663E-06  0.20621      4.05506E-04   813.28      0.58147

   TIME     C14MINEF     NMINAVPOOL   N15MINAVPOOL NMINAVEF     NMINHDPOOL

1.00000      0.00000       5.2500      0.00000      0.00000      0.00000
1.00000      0.00000       5.2500      0.00000      0.00000      0.00000
2.06021      1.50995E-04  3.13538E-07 -2.87807E-06  0.00000      0.73474
2.06021      1.50995E-04  0.00000      0.00000      0.60935      0.73474
3.00000      2.45307E-04  0.00000      0.00000      0.69317       2.5062
5.00000      4.15016E-04  0.00000      0.00000      0.76024       7.8571
5.99531      4.87103E-04  0.00000      0.00000      0.77615       10.603
5.99531      4.87103E-04  0.00000      0.00000      0.77615       10.603
7.00000      5.52567E-04  0.24876      0.19552      0.78601       13.202
9.00000      6.65571E-04   1.9653       1.5668      0.79726       17.617
10.0000      7.14969E-04   3.2964       2.6406      0.80107       19.417

   TIME     N15MINHDPOOL NMINHDEF     NMINTOTAL    N15MINTOTAL    NMINEF

1.00000      0.00000      0.00000       5.2500      0.00000      0.00000
1.00000      0.00000      0.00000       5.2500      0.00000      0.00000
2.06021      0.44772      0.60935      0.73474      0.44771      0.60935
2.06021      0.44772      0.60935      0.73474      0.44772      0.60935
3.00000       1.7372      0.69317       2.5062       1.7372      0.69317
5.00000       5.9732      0.76024       7.8571       5.9732      0.76024
5.99531       8.2292      0.77615       10.603       8.2292      0.77615
5.99531       8.2292      0.77615       10.603       8.2292      0.77615
7.00000       10.396      0.78744       13.451       10.591      0.78741
9.00000       14.129      0.80200       19.582       15.695      0.80152
10.0000       15.667      0.80690       22.713       18.308      0.80605

   TIME     RATEREDUCTION EFFREDUCTION NCRATREDUCTION   CTOTAL

1.00000        1.0000       1.0000         1.0000       17700.
1.00000        1.0000       1.0000         1.0000       20202.
2.06021        1.0000       1.0000         1.0000       20202.
2.06021       0.94090      0.95703        0.95000       20202.
3.00000       0.97000      0.96000        0.95477       20202.
5.00000       0.97000      0.96231         1.0000       20202.
5.99531        1.0000       1.0000         1.0000       20202.
5.99531        1.0000       1.0000         1.0000       20202.
7.00000        1.0000       1.0000         1.0000       20202.
9.00000        1.0000       1.0000         1.0000       20202.
10.0000        1.0000       1.0000         1.0000       20202.

   TIME     C14TOTAL       NTOTAL     N15TOTAL

1.00000       5.0000       810.90      0.00000
1.00000       5.0000       1037.9       227.00
2.06021       5.0000       1037.9       227.00
2.06021       5.0000       1037.9       227.00
```

```
   3.00000       5.0000       1037.9       227.00
   5.00000       5.0000       1037.9       227.00
   5.99531       5.0000       1037.9       227.00
   5.99531       5.0000       1037.9       227.00
   7.00000       5.0000       1037.9       227.00
   9.00000       5.0000       1037.9       227.00
   10.0000       5.0000       1037.9       227.00
```

4.2 Res.bin

This file is made during simulation and contains in machine readable form all the values sent to output by the OUTDAT calls in the output section. The tables in RES.DAT are based on the contents of RES.BIN. The graphical utility TTSELECT uses the file RES.BIN to produce graphs.

4.3 Motor.log

MOTOR.LOG (in the coupled version this file is called KOPPEL.LOG) contains at least a summary of the model execution. The input routines from TTUTIL write messages on it, the driver writes messages on starting the initial phase, the dynamic phase and the terminal phase of the model run. The logfile is most useful in case of errors. By setting the variable TRACE in the input file TIMER.DAT, the simulation driver is instructed to report into greater detail. The possible values of TRACE are

0	message on Initial, Dynamic, Terminal
1	also short initialization report
2	also reports time and state events; writes times and number of steps to screen
3	also reports integration steps + output times
4	also reports on state event iteration

5 Running the program

As mentioned in chapter 1, this report describes two different version of MOTOR, that is the FSE stand-alone version and the version coupled with FUSSIM2. This chapter deals with the implementation of the two versions and with the similarities and differences.

5.1 Stand-alone version

To build a stand-alone version of MOTOR, the source code file MOTOR.FOR is needed. After the compilation with a Fortran 77 compiler with its non-standard switch on, or with a Fortran 90 compiler, the resulting object code is linked with the object libraries TTUTIL and DRIVERS.

To run the program the input files TIMER.DAT (section 3.1) and MOTOR.DAT (section 3.3) are needed. The program produces the output files, described in chapter 4.

5.2 Version coupled with FUSSIM2

5.2.1 Incorporating MOTOR

Heinen & de Willigen (2001) incorporated MOTOR in their water movement, solute transport and root uptake simulation model FUSSIM2.

To build this coupled simulation program FUSSIM2-MOTOR, the following files are needed besides the standard FUSSIM2 source code and library files:

MOTOR.FOR file containing the actual MOTOR formulations. This MOTOR.FOR is not entirely the same as the stand-alone version to make it suitable for coupling with FUSSIM2.

MOTORRK.FOR file designed to call the driver (and consequently the model MOTOR) by FUSSIM2.

DRIVERS.LIB library file containing the Runge-Kutta driver

GETSET.FOR file designed to get and store the status of (all) variables per CV.

The coupled model is compiled and linked within a Fortran 90 programming environment.

It was decided to retain as much as possible of the stand-alone version of MOTOR. However several important changes were unavoidable. The Runge-Kutta driver had to be changed in such a way, that it could be called from outside (i.e. from

FUSSIM2) with four different tasks: initialization, dynamic, output and terminal. Another change to the driver is that output, other than output during state and time events, is governed by FUSSIM2.

Furthermore it was essential that to every layer or CV its own input file with pool table, transformation table and addition table could be assigned. To make this possible an extra input file, MOTORTYPE.DAT (section 3.2), had to be created.

Also the states of the different variables have to be stored per CV in FUSSIM2. Two ways of storage are created: a simple storage of the states only, and a complete storage of all model variables. The second method is the most flexible, but costs a large amount of memory storage. The first method can only be used in a special rare case (i.e. when all the layers use the same transformations and the only difference is the initial distribution of the pools).

MOTOR needs information about moisture status and temperature to calculate the adjustment factors F_3 (equation 4) and F_2 (equation 3). In the coupled version the water content and temperature distribution are calculated by FUSSIM2 and passed on to MOTOR.

Due to the differences in their properties and behaviour FUSSIM2 distinguishes two kinds of mineral nitrogen, nitrate-N and ammonium-N. MOTOR does not make this distinction. There are several consequences of handling mineral nitrogen different in FUSSIM2 and MOTOR. Before MOTOR is called by FUSSIM2, the total amount of mineral nitrogen is calculated by summarizing the amount of nitrate and ammonium. The ratio nitrate/ammonium, before calling MOTOR, is stored in FUSSIM2. When the organic matter calculations by MOTOR result in a nett consumption of mineral nitrogen, both the amount of nitrate and ammonium are reduced. The reduction of the amount of nitrate and ammonium is not equal but depends on the ratio nitrate/ammonium before calling MOTOR. Nett production of mineral nitrogen in MOTOR results in an increasing amount of ammonium. The amount of nitrate stays the same.

It was decided that in the coupled simulation model FUSSIM2-MOTOR, the calculations of the organic matter transformations (by MOTOR) were performed after calculating the processes nutrient uptake by roots, transport of solutes and denitrification.

To ensure that an addition to a pool is executed at the right moment and to the right CV, several initialization and checking statements are added to the program explicitly.

5.2.2 Input files

To run the coupled version of MOTOR the input files MOTORRK.DAT (section 3.1) and MOTORTYPE.DAT (section 3.2) are needed. In MOTORTYPE.DAT additional input files are specified, containing information about the organic matter model and additions. The timer and weather control variables in MOTORRK.DAT are not used instead these variables are controlled by FUSSIM2.

In order to perform a simulation with MOTOR, the flag MOTOR in the FUSSIM2 file FUS2.CTR has to be set to ‘y’.

The variables NminAvInitial, Onebar, Deficit, Dmax and Temperature are not read from the MOTOR input file, but are calculated by FUSSIM2 and passed on to MOTOR. There are no other differences between MOTOR.DAT (stand-alone) and the input files, specified in MOTORTYPE.DAT.

5.2.3 Output files

In section 4.1 a description and example of the formatted output file RES.DAT was given. This example is the result of a simulation with the stand-alone version of MOTOR. In principle the RES.DAT of a simulation with the coupled version of MOTOR could be the same, however it isn’t very practicle. Imagine, a calculation with 7 different pools (see for example figure 2.1) and 10 CV’s would result in at least 490 (7 (pools) times 10 (CV’s) times 7 (pool properties)) columns with values as a function of time for the MOTOR output alone. FUSSIM2 also creates quite a lot of output columns.

Because it is easy to add variables to the output section of the coupled model, it was decided not to write all the pool properties to the file RES.DAT. Usually one is interested in only one or two pool properties for a few CV’s. These properties and CV’s can to be programmed in the output section of the coupled model before running a simulation. Unfortunately there is not yet an input file which defines the pool properties and CV’s someone is interested in.

5.2.4 Recommendations and remarks

Section 5.2.1 described that FUSSIM2 distinguishes two kinds of mineral nitrogen and MOTOR not. This distinction involves consequences for adding mineral nitrogen to a simulation. In the coupled model FUSSIM2-MOTOR mineral nitrogen should NOT be added to the simulation by using of the pool ‘MineralN’ in the addition table of MOTOR.DAT (or the input files specified in MOTORTYPE.DAT). Instead mineral nitrogen should be added by using the FUSSIM2 input file FERTIL.DAT. FERTIL.DAT has the possibility to distinguish an addition of nitrate-N, ammonium-N or both. Adding mineral nitrogen by using the addition table of MOTOR.DAT will result in an increasing amount of ammonium, even when a fertiliser containing (only) nitrate-N was added.

The stand-alone version of MOTOR can simulate the fate of enriched carbon (C14) and enriched nitrogen (N15). The coupled model FUSSIM2-MOTOR CANNOT handle enriched nitrogen pools. To handle enriched nitrogen properly, distinction has to be made between nitrogen and enriched nitrogen throughout the entire model FUSSIM2. Unfortunately such a version of FUSSIM2 does not yet exist.

When for instance a “nitrogen balance error of … at time …” is encountered during a simulation with the coupled version of MOTOR, there is a good possibility that the used time step is too big. Reducing the maximum allowable time step (variable DTMAX in the FUSSIM2 input file .._tm.dat) may solve this problem but the calculation time will increase considerable.

Using different organic matter models for every layer or CV is not recommended. Use the same pools and transformation table for every layer or CV, otherwise correct calculation of transformations cannot be ensured.

5.3 Accuracy

The accuracy tolerance of the Runge-Kutta integration procedure used by RKDRIV is the variable EPS in the input file TIMER.DAT. This tolerance level is used as a relative error for all pools except the available mineral nitrogen pools. If a carbon pool size, however, decreases below 1.0E-20, the contents of that pool are set to zero (carbon, nitrogen and enriched fractions). This is done to prevent the use of small time steps for the simulation of amounts which are effectively zero.

In principle, also the available nitrogen pools (N and N15) are simulated with a relative accuracy of EPS, which means that the absolute tolerance level is EPS * Poolsize. This absolute tolerance level, however cannot decrease below EPS * 0.01 * NminEventScale (see Section 3.3.4). The reason for this limitation is that the mineral nitrogen balance is the result of a combination of potentially large source and sink terms. That means that very low mineral nitrogen amounts may occur in combination with large fluxes. The simulation of such low amounts with a high relative accuracy is not possible due to the limitations of the floating point accuracy of the other pools.

5.4 Error messages

The are a few categories of error messages. Syntax errors in the input files lead to messages from the input parsing routines of TTUTIL. The input files should simply meet the demands of those routines.

A second category are driver errors during the initial phase. This will usually mean that something is wrong with the values on the input file TIMER.DAT.

A third category of error messages is initialization errors from MOTOR. Negative amounts, fractions exceeding 1.0 etc. The message is a short description given by MOTOR, usually in the style
“Error in MOTOR: *message*”.

After the start of the DYNAMIC phase (the driver always reports that on screen), error messages and warnings are more difficult to handle. Setting the TRACE variable and making an extended logfile is the first thing to do. The message may be the result of a programming error. In other cases, a change in the accuracy settings of the program will solve the problem. Two complicated cases are discussed below.

Every time step MOTOR makes an overall balance check of the amounts of carbon, nitrogen and enriched material. A small balance error will usually be the result of an integration error. Errors exceeding 1.0E-4 are reported as a warning. Large balance errors will usually be the result of programming errors. Sometimes they are caused, however, by a too low accuracy in state event iteration. If the mineral nitrogen pool reaches zero, a type 1 state event is generated. At event time, the available mineral amounts are explicitly set at zero, throwing away the tiny positive or negative amount left in those pools at event time. Hence, if the time of the state event is not found with sufficient accuracy, a balance error results.

Sometimes, the driver gives a warning saying that a state event iteration failed. This may happen if new material is added to the system during a period without available mineral nitrogen. If the addition of the new material suddenly leads to a positive Nmin balance, the available mineral pool starts to grow from the time of addition onwards. The driver, however, is confronted with a suddenly positive nitrogen balance. There is no time step, during which, somewhere, the nitrogen balance passed zero. The driver tries to find the precise time of the event and iterates closer and closer to the addition time without reaching it precisely (it uses bisection of a time interval). This causes a failure of the iterative procedure, which is not a failure of the program, however. By setting TRACE=4 in TIMER.DAT the entire sequence can be followed on the logfile report.

6 Test and examples

This chapter presents several test results and examples of simulations with the different versions of MOTOR.

6.1 Standard situation

In this section a comparison is made between the simulation results, calculated by the original version of MOTOR (Whitmore) and the stand-alone (FSE) version of MOTOR, described in this report. The used organic matter model is presented in figure 2.1.

In table 6.1 the used initial amounts of carbon and the C/N-ratios are presented. The initial amount of mineral nitrogen is 5.25 mg N/kg soil. The PPOM saturation level is 21000. At the first day of the simulation 2502 mg C/kg soil and 227 mg labelled N/kg soil were added to the pool Residue1.

Table 6.1. Pool table with pool names, initial amounts of carbon (mg C/kg soil) and C/N-ratios.

Pool name	Amount of carbon	C/N-ratio
Residue1	0	0
Residue2	0	0
Biomass	100	6
ActiveOM	100	10
PhysPOM	1000	12
ChemPOM	4500	23
PhChPOM	12000	24

Figure 6.1 presents the results of a simulation with the Whitmore (A) and the stand-alone (B) version of MOTOR. The simulation results basically show the same trend in time. The most obvious difference is shown by the nitrogen content in the Residue1-pool. In figure 6.1A and B the maximum amount of nitrogen in the Residue1-pool is 177 and 227 mg N/kg soil respectively. In other words all the added nitrogen is visible in figure B and not in figure A.

It seems that, based on the results in figure 6.1, the stand-alone version of MOTOR is as good as (or maybe even better than) the original version of Whitmore, because the added amount is present and the trend in time is more or less the same. Besides these reasons creating input files for the stand-alone version of MOTOR is much easier due to TTUTIL read and write routines.

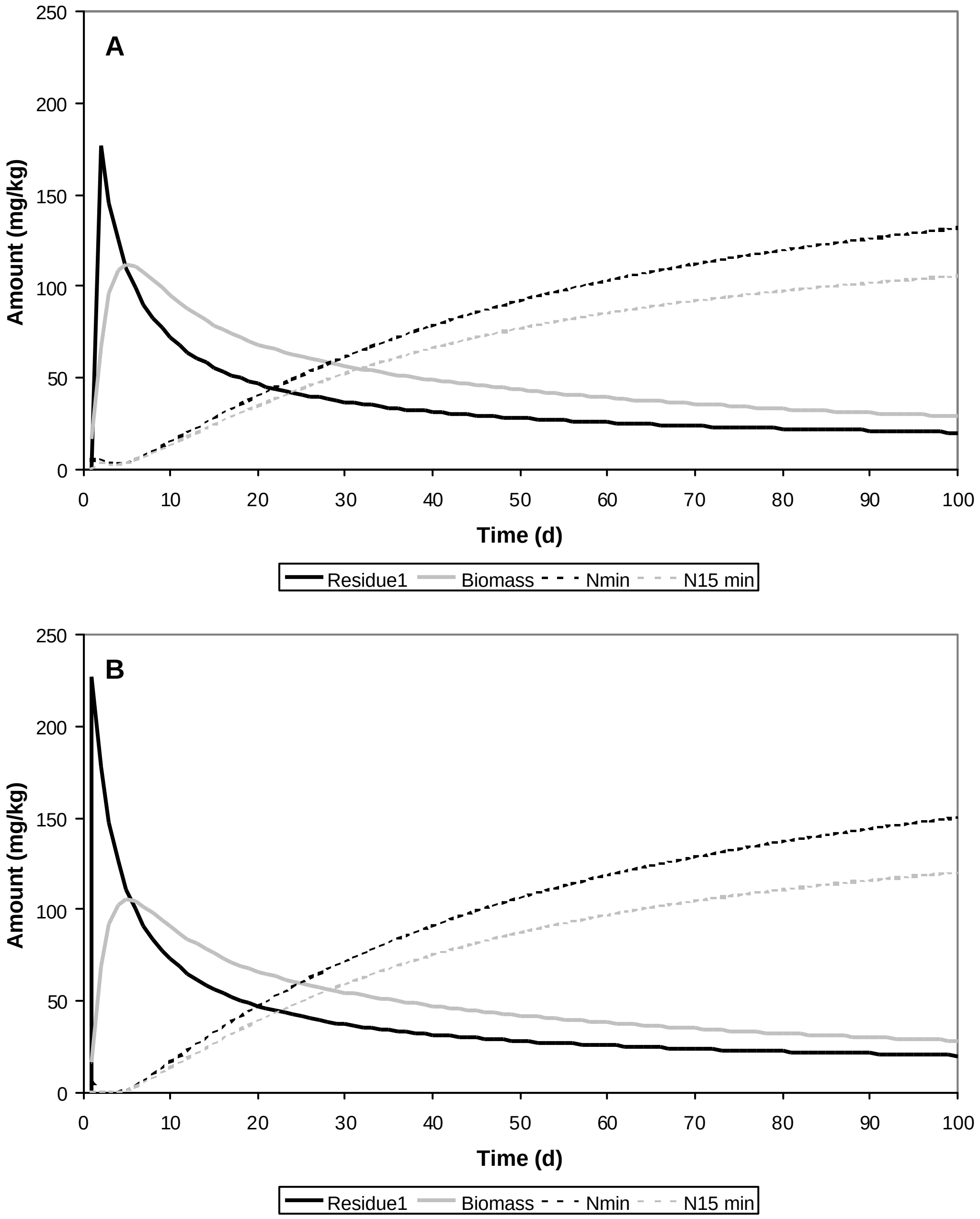

Figure 6.1. Simulated nitrogen amounts of the pools Residue1, biomass, mineral nitrogen and labelled (N15) mineral nitrogen. Figure A are results of a calculation with the Whitmore version of MOTOR, figure B are results of the stand-alone (FSE) version of MOTOR, presented in this report.

Also a comparative calculation was made for this situation with the stand-alone (FSE) version of MOTOR and the version coupled to FUSSIM2. The simulation-results, calculated by these different versions, were identical.

6.2 Reduced nitrogen situation

In this section almost the same calculations as in section 6.1 are performed. Only the initial amount of mineral nitrogen (55.25 mg N/kg soil) and the added amount of nitrogen to pool Residue1 (27 mg labelled N/kg soil) are different. Figure 6.2 presents the results of a simulation with the Whitmore (A) and the stand-alone (B) version of MOTOR.

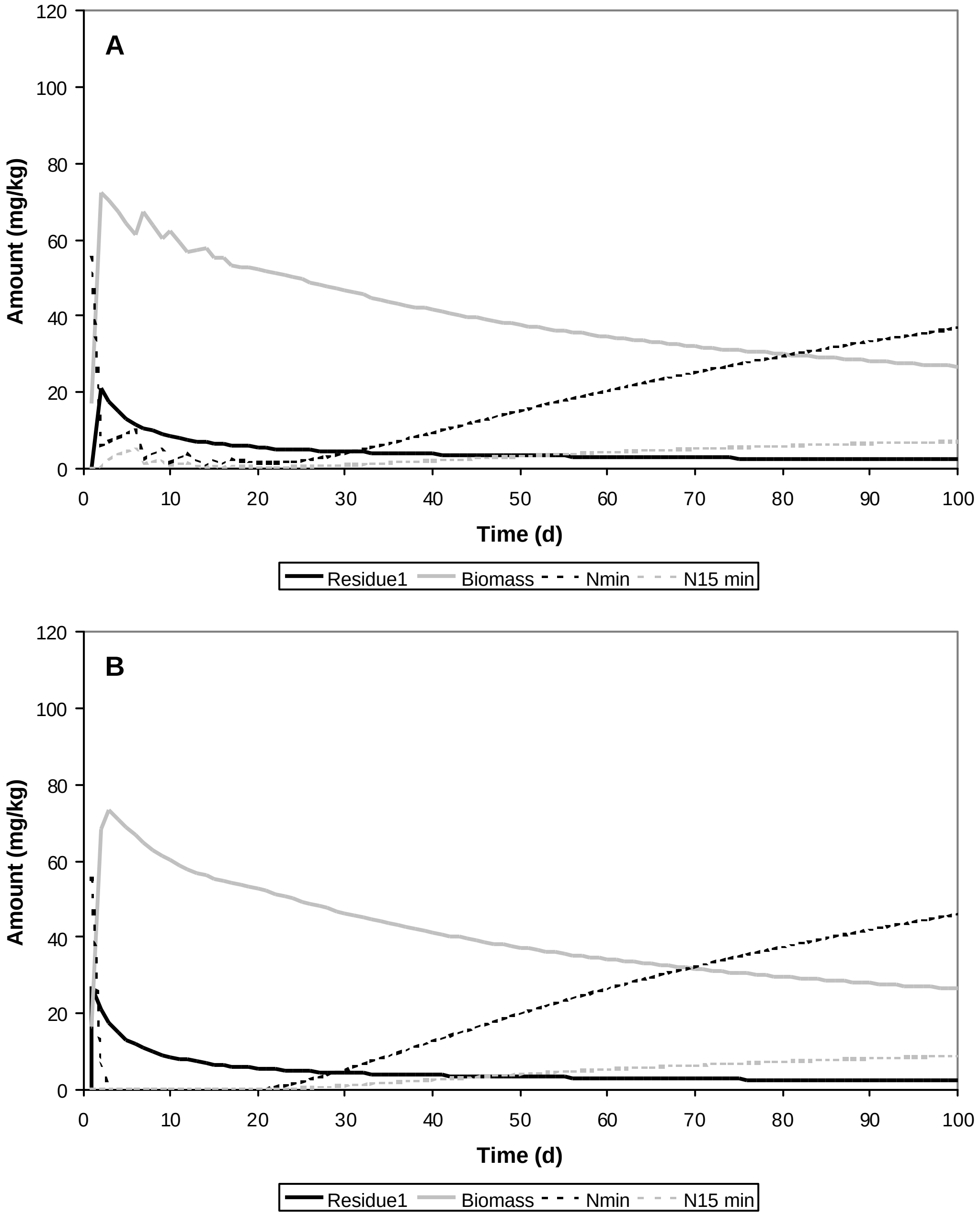

Figure 6.2. Simulated nitrogen amounts of the pools Residue1, biomass, mineral nitrogen and labelled (N15) mineral nitrogen of a situation with less nitrogen added to the Residue1-pool. Figure A are results of a calculation with the Whitmore version of MOTOR, figure B are results of the stand-alone (FSE) version of MOTOR.

It is obvious that in figure 6.2 less nitrogen is added to the Residue1-pool compared to figure 6.1. The effect is that in figure 6.2(B) no mineral nitrogen is present until day 22. Due to the settings of the state events, the nitrogen demand in the stand-alone calculation will be reduced until day 22.

The trend in time is much more smoother in figure 6.2B, compared to figure 6.2A, due to the Runge-Kutta driver and the use of state events (see Appendix A).

Simulation results, calculated with the coupled version of MOTOR, were identical to the ones, calculated with the stand-alone version of MOTOR.

The results, presented in figure 6.1 and 6.2, give us a lot of confidence in the performance of the stand-alone (FSE) version and the coupled version of MOTOR.

6.3 Many years

In this section the standard situation of section 6.1 was simulated for 3 consecutive years, using the climate run option of the stand-alone version of MOTOR. In figure 6.3 the simulation results are presented. Figure 6.3 shows that the amount of mineral nitrogen is increasing due to the lack of sinks. This result is the reason for coupling MOTOR to a water movement and nutrient transport model like FUSSIM2. Sinks, like root uptake and denitrification, can be accounted for in the coupled version of MOTOR. Simulating this situation with the coupled version of MOTOR, without any sinks defined, produces the same results as presented in figure 6.3.

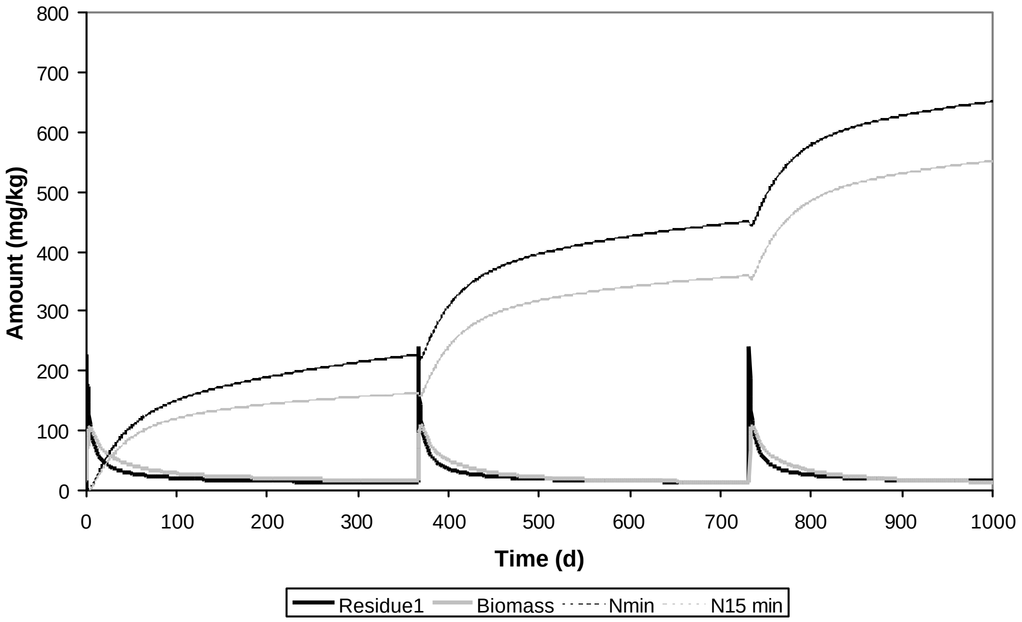

Figure 6.3. Simulated nitrogen amounts of the pools Residue1, biomass, mineral nitrogen and labelled (N15) mineral nitrogen, running the standard situation 3 consecutive years with the stand-alone (FSE) version of MOTOR.

6.4 Sugar beets grown on a sandy soil

In this section we will present an example of a simulation with the coupled version of MOTOR. It is a more realistic example of the behaviour of organic matter and nutrients in soil and it also shows the capabilities of the coupled model FUSSIM2-MOTOR. The used organic matter model is not the same as in the previous simulation examples. This organic matter model has three plant material pools (DPM, SPM, RPM), a labile organic matter (LOM) and a stabile organic matter (SOM) pool (figure 6.4).

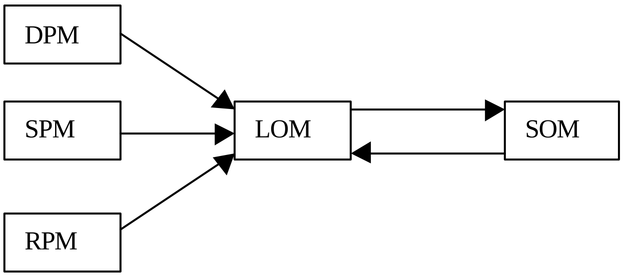

Figure 6.4. The organic matter model used in this section.

In this example sugar beets where grown on a sandy soil. At day 32 of the simulation pig slurry was mixed through the soil to a depth of 15 cm. The slurry consisted of mineral and organic nitrogen. At day 95 of the simulation artificial manure (consisting nitrogen) was applied to this plot. At day 323 the harvest residues were added to the soil. These residues only consisted organic nitrogen. Figure 6.5 shows the simulated mineral nitrogen content in the layer 0-30 cm, the nitrate uptake by the sugar beets and nitrate leaching. The additions of pig slurry and artificial manure are clearly visible in the amount of mineral nitrogen (Nmin). This example also shows that most of the nitrate leaching happens when the uptake has stopped.

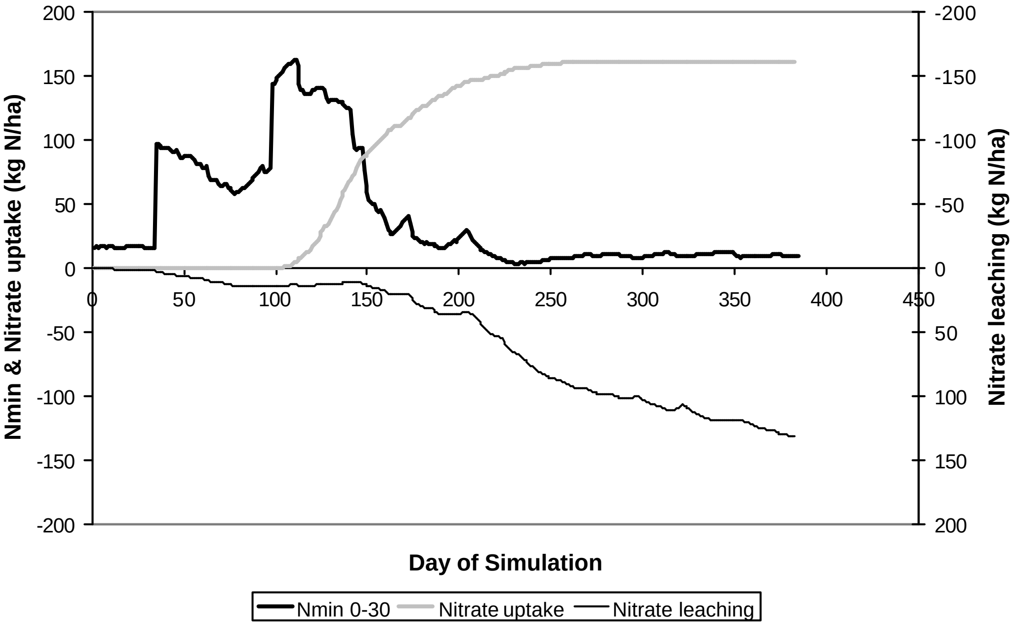

Figure 6.5. The mineral nitrogen content (Nmin) in layer 0-30 cm, the nitrate uptake and nitrate leaching, calculated with the coupled model FUSSIM2-MOTOR for a situation with sugar beets grown on a sandy soil.

References

Bradbury, N.J., A.P. Whitmore, P.B.S. Hart & D.S. Jenkinson, 1993. Modelling the fate of nitrogen in crop and soil in the years following application of ^{15}N-labelled fertilizer to winter wheat. Journal of Agricultural Science 121:363-379.

Hassink, J. & A.P. Whitmore, 1997. A model of the physical protection of organic matter in soils. Soil Science Society American Journal 61:131-139.

Heinen, M. & P. de Willigen, 2001. FUSSIM2 version 5. New features and updated user's guide. Alterra-rapport 363, Alterra, Wageningen.

Jenkinson, D.S., P.B.S. Hart, J.H. Rayner & L.C. Parry, 1987. Modelling the turnover of organic matter in long-term experiments at Rothamsted. INTECOL Bulletin 15:1-8.

Rappoldt, C. & D.W.G. van Kraalingen, 1990. Reference manual of the Fortran utility library TTUTIL with applications. Simulation report CABO-TT nr. 20, CABO-DLO & TPE, Wageningen.

Van Kraalingen, D.W.G., 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis No. 1. DLO Research Institute for Agrobiology and Soil Fertility and The C.T. de Wit graduate school for Production Ecology, Wageningen.

Van Kraalingen, D.W.G. & C. Rappoldt, 2000. Reference manual of the Fortran utility library TTUTIL v. 4. Report 5. Plant Research International, Wageningen.

Appendix A: Time and state events

Time events are prescribed moments in time at which something special happens. State events are moments in time on which the system reaches a specified state. In MOTOR, time events correspond to the addition of new material to the soil during the model run. State events signal the status of the mineral N pool, which, at an unknown moment in time, may become empty. By setting up a state event for an empty mineral N pool, the driver automatically finds this moment. After that moment MOTOR has to reduce the mineral N demand.

A.1 Substrate addition or time events

In subroutine Read_Addition_List the list of addition times is read. Years and days are transformed into values of TIME (which always increases). The addition times are sorted. The number of the smallest one is AddTimeIndex(1), the next one has number AddTimeIndex(2), etc. The first addition that has to take place is then found.

By setting the TEVREQ flag, the driver is instructed to call the model at TEVTIM, the first addition time. The model itself then handles the time event, i.e. adds new material to one or more pools. This is done at the beginning of the Dynamic section. After handling the substrate addition, the next time event is requested, or, if no further additions were specified, the request flag TEVREQ is reset.

In the special case of a climate run (year 1000, see Section 3.3.3), the handled events are repeated after 365 days by adding 365.0 to the event time in variable AddTime.

A.2 State events

There are two possible state events:

1 The mineral N reaches zero. From this moment on the mineral nitrogen demand is reduced.

2 The mineral N balance changes sign. This event marks the minima and maxima in the amount of available mineral nitrogen, in other words Nmin consumption changes in Nmin production or vice versa. It also marks the end of a period without nitrogen.

The difficulty of state event 1 lies in its consequences: the reduction of the mineral N demand. The implementation of the event itself is very easy. The NminZERO flag is used in the program to flag the zero status of the available Nmin pool. Together with the state events, this flag is initialized towards the end of the Initial section.

If the mineral pool is empty, the NminZERO flag is set, but the mineral N balance in the dynamic section has to make a final decision. The two possible state events are simply requested by setting two flags.

As a result of the request flags SEVREQ, the driver will monitor the state event functions calculated in a special call to the model, in the ITASK=3 section:

```
ELSE IF (ITASK.EQ.3) THEN

*        State Event Functions
*        ====================
*        function for Event 1:  The Nmin pool reaches zero
         SEVFUN(1) = NminAvPool / NminEventScale

*        function for Event 2: Nmin Production  <--> Nmin Consumption
         SEVFUN(2) = (NminSumToAvPot + FromHdToAv - NminSumDemandPot) /
         $NOTNUL (Max (NminSumToAvPot+FromHdToAv, NminSumDemandPot))
```

The first function just calculates a dimensionless size of the available mineral N pool. The second event function is a dimensionless difference between supply and demand of available mineral nitrogen. These event functions are always called after ordinary rate calls. Hence, the variable values used in the ITASK=3 section are always up to date.

If one of the event functions crosses zero, the time of that event is found iteratively and then the model is called with the state event flag STEVNT set. This causes the execution of the following model section, on top of the dynamic part.

```
*       State event handling
*       --------------------
        IF (STEVNT) THEN
*       which one ??
        IF (SECROS(1)) THEN
*       NminAvPool   = 0.0
        N15minAvPool = 0.0
        NminZERO     = .True.

*       the end of the period without mineral nitrate
*       will be the next type 2 state event
*       the end of the rates of change section
        AssignStateArray = .True.

        ELSE IF (SECROS(2)) THEN
*       State Event 2: Switches between net Nmin production
*       and net Nmin consumption
        NminZERO = .False.
        END IF
        END IF
```

Note that at the type 1 event, the mineral pools are forced to zero. A tiny positive or negative amount in these pools is removed. A balance error resulting from that requires a more accurate event detection (decrease SEVTOL in TIMER.DAT).

When during a period of no nitrogen (state event 1) the (potential) Nmin production becomes bigger than the (potential) Nmin consumption, a state event 2 will be set. The amount of Nmin will now be bigger than zero.

Index