



**Department of Soil and Environment  
Division of Biogeophysics and Water Quality**

# **Users guide to MACRO 5.2, a model of water flow and solute transport in macroporous soil**

**Fredrik Stenemo and Nicholas Jarvis**

**2010**



**Fredrik Stenemo  
ÅF konsult  
Stockholm, Sweden**

**Nicholas Jarvis  
Department of Soil and Environment  
Swedish University of Agricultural Sciences**

## Contents

1	STRUCTURE OF THE PROGRAM	3
1.1	File formats	4
1.1.1	Weather data	5
1.1.2	Measured data	5
2	CREATING A PROJECT	6
3	DEFINING A SIMULATION	7
3.1	Options	7
3.2	Simulation setup	8
3.3	Soil properties	11
3.4	Parameters	12
3.5	Outputs	14
4	SAVING A SIMULATION	15
4.1	Save to parameter file	16
5	OPENING AN EXISTING PROJECT AND SIMULATION	16
6	EXECUTING SIMULATIONS	17
6.1	Batch mode	18
7	TOOLS	19
7.1	Converting files	19
7.1.1	File conversion from binary format to ASCII format	19
7.1.2	File conversion from ASCII to binary format	19
7.1.3	Daily to hourly rainfall	21
7.2	Change parameters	22
7.3	Parameter estimation with SUFI	23
7.3.1	Creating a new project	24
7.3.2	Creating a new iteration	26
7.3.3	Executing an iteration	27
7.3.4	Analysing an iteration	28
7.3.5	Measured data	34
7.4	Monte Carlo simulation	34
7.4.1	Sampling of parameter values	35
7.4.2	Execution of parameter variation file	36
7.4.3	Analysing the results of the Monte Carlo simulations	39
7.5	Compute goal function	40
7.6	The PG graphics program	41
8	REFERENCES	41

## Appendices

- Appendix 1 Pedotransfer functions
- Appendix 2 Useful commands in PG

# 1 Structure of the program

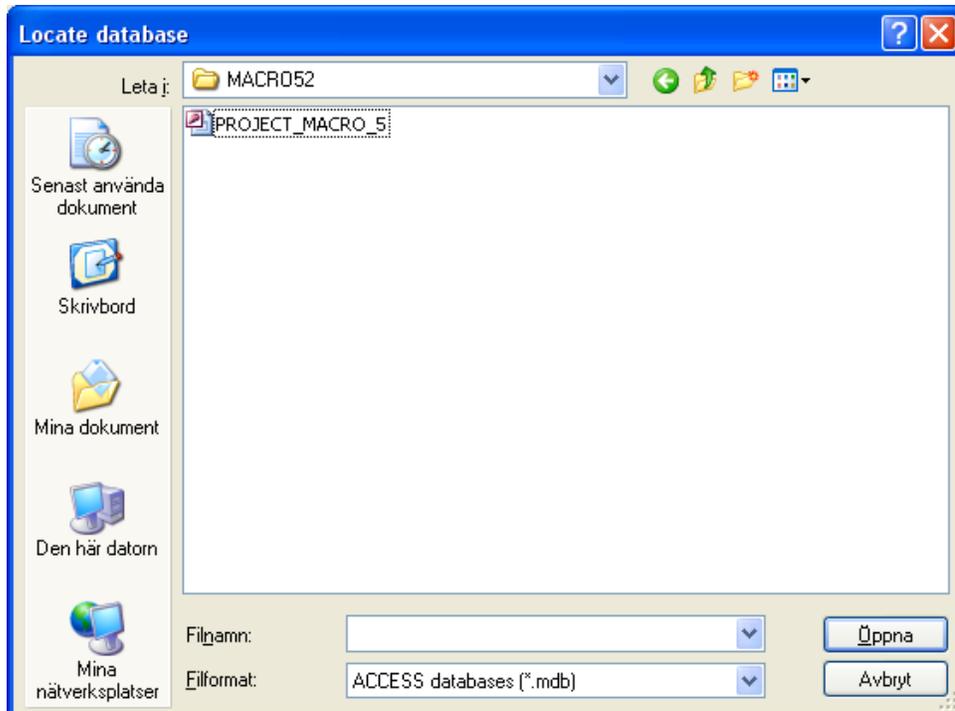
MACRO 5.22 consists of a number of different linked modules and components:

1. The MACRO graphical user interface
2. The MACRO model code used to run a simulation
3. An MS Access database used for storage of projects and simulations
4. Formatted files containing driving data, measured data etcetera
5. Additional program, (e.g. programs for Monte-Carlo simulations)

The model code itself is written in FORTRAN, and this is linked to a graphical user-interface (GUI) written in Microsoft Visual Basic 2008. The simulation model and the GUI communicate through files (e.g. driving data files) that are either in ASCII (extension .txt) or binary format (extension .bin). All information needed to run the system is stored in a Microsoft Access database. Although this is not necessary, several different MACRO5.2 databases can be maintained, containing information on different projects. You can switch between these different databases by clicking on “*File->Change database*”.



The default database supplied with the model is called PROJECT\_MACRO\_5.mdb.



The GUI is also ‘loose-linked’ to some auxiliary software programs, for example the UNCSAM programs for model sensitivity and uncertainty analysis.

## 1.1 File formats

This section describes the formats used for input weather data files and files containing measured data, how to create these files in text format, and how to convert them into MACRO binary formatted files using the MACRO 5.22 Wizard.

Input files should be prepared as ASCII files with .txt extension, using an ordinary text editor (e.g. Microsoft Wordpad or Notepad) or Excel. Columns should be tab-separated, and without headers. It is important to make sure that the decimal separator (comma or point) in the text files matches the regional settings on your PC. The ordering of file columns is free, but the date/time should always be in the first column, specified as follows:

YYYYMMDDHHMM (e.g. 199903171245 which is 17th March 1999 at 12:45). The following separators between the year, month, day and time values are allowed: “ “, “-“, “/”, “:”, for example “1999:03:17 12:45”, or “19990317 12:00”.

If all your data has a daily resolution, then the hours and minutes can be skipped (e.g. 19990317).

### 1.1.1 Weather data

Two weather data files are needed to run a simulation in MACRO 5.2, one containing only precipitation data, the second containing air temperatures and either pre-calculated daily potential evapotranspiration, or the meteorological data required to calculate potential evapotranspiration:

1.) Either hourly or daily rainfall data may be used. Figures are given in millimetres. A daily rainfall data file might look like this:

```
19990104    2.2
19990105    0.0
19990106    0.2
19990107    0.6    and so on.....
```

An hourly rainfall file might look like this:

```
199901041030    2.2
199901041130    0.0
199901041230    0.2
199901041330    0.6
199901041430    0.0    and so on.....
```

2.) The remaining weather data required by the model can be specified in two different ways. In both cases, a daily resolution is required:

a.) A file containing the meteorological data required to calculate potential evapotranspiration internally in the model using the Penman-Monteith equation. The data needed are: solar radiation ( $Wm^{-2}$ ), maximum and minimum air temperatures ( $^{\circ}C$ ), vapour pressure (kPa) and wind speed ( $m\ s^{-1}$ ). The columns of data can be in any order. If maximum and minimum temperatures are not available, the daily mean temperature can be substituted for both variables (i.e. two identical columns of temperature data must be input).

b.) A file containing pre-calculated potential evapotranspiration and maximum and minimum temperatures. Again, the columns of data can be in any order. If maximum and minimum temperatures are not available, the daily mean temperature can, as before, be used twice.

See section 7.1.3 for information on how to convert daily rainfall data to hourly rainfall data.

### 1.1.2 Measured data

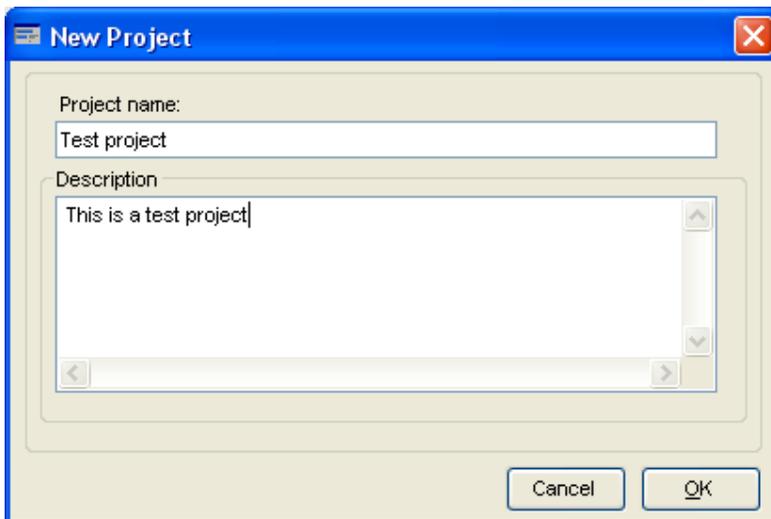
MACRO 5.2 allows statistical comparisons of model simulations and measured data, and also includes an inbuilt parameter estimation methodology ('inverse modelling') based on quantitative estimates of model goodness-of-fit to data. The file containing the measured data should be created as a .txt file formatted in the same way as the weather data files. Missing data should be denoted with an unusual integer number such as -999.

## 2 Creating a project

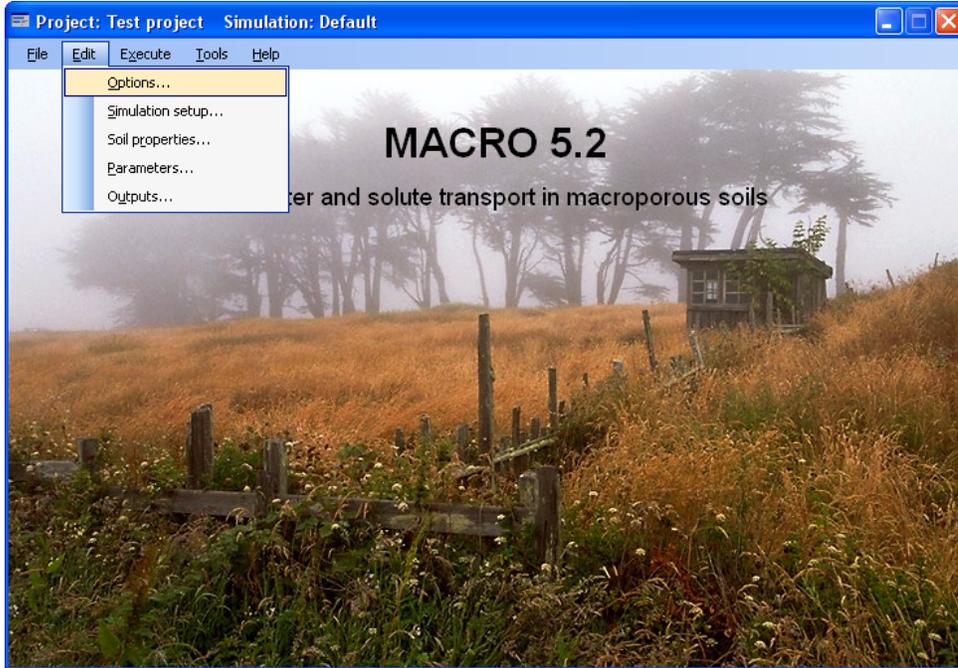
You can create a new project in MACRO 5.2 by clicking on “*File->New*”.



Give an appropriate project name and description, and then click on ‘*O.K.*’



All inputs in MACRO 5.2 are then automatically given default values. You now define your simulation by modifying these default inputs under ‘*Edit*’.



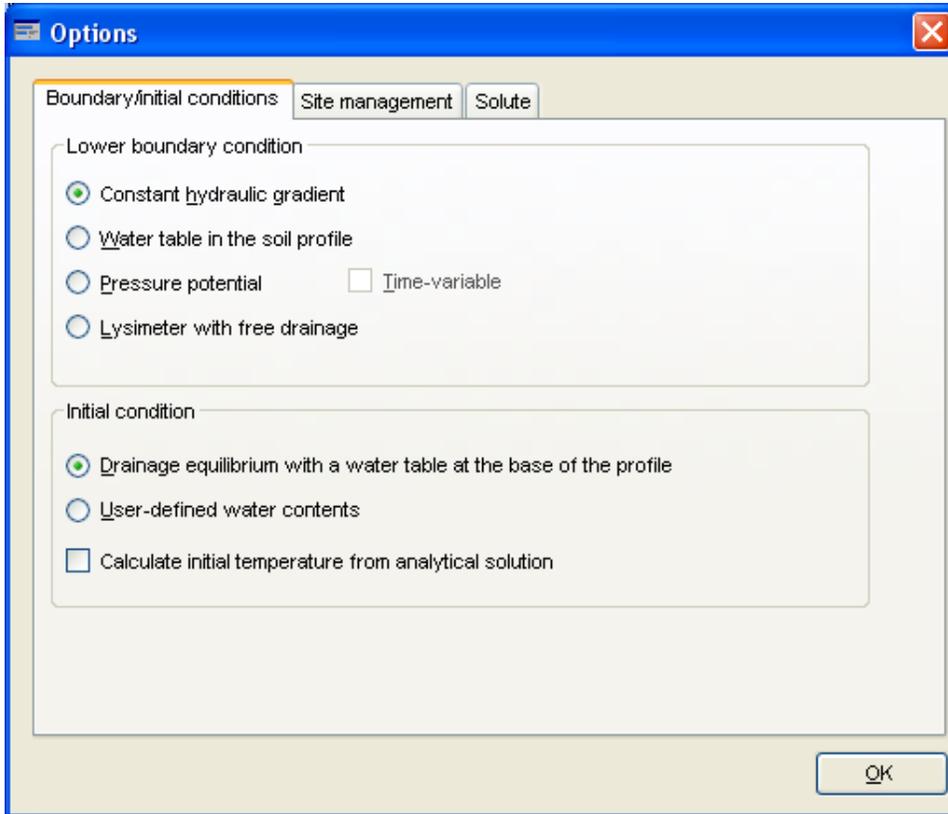
You specify the options, the simulation setup (driving data, start/stop times for the simulation, output interval etc.), the soil properties, the parameter values and the output variables, that together define a unique simulation under the ‘umbrella’ of the project that is currently open.

### **3 Defining a simulation**

The different components of a simulation can be defined in any order, but it usually makes sense to follow the order of the menu items.

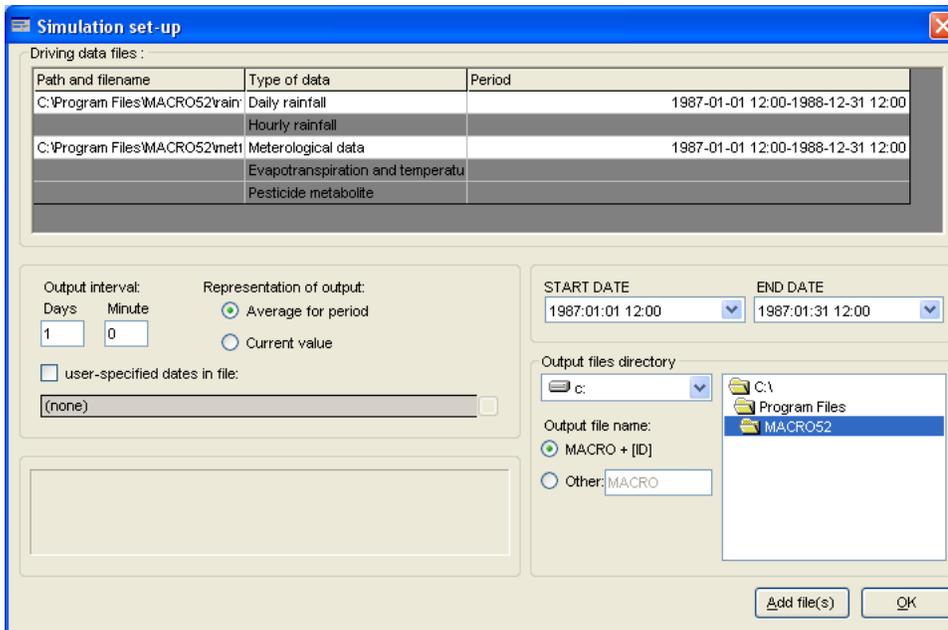
#### **3.1 Options**

Under “*Options*”, you make basic choices that define the kind of simulation you want to run. These options are classified under three headings: ‘*Boundary/initial conditions*’, ‘*Site Management*’ and ‘*Solute*’. The parameters needed to run the simulation and the output variables that are available, depend on these choices.

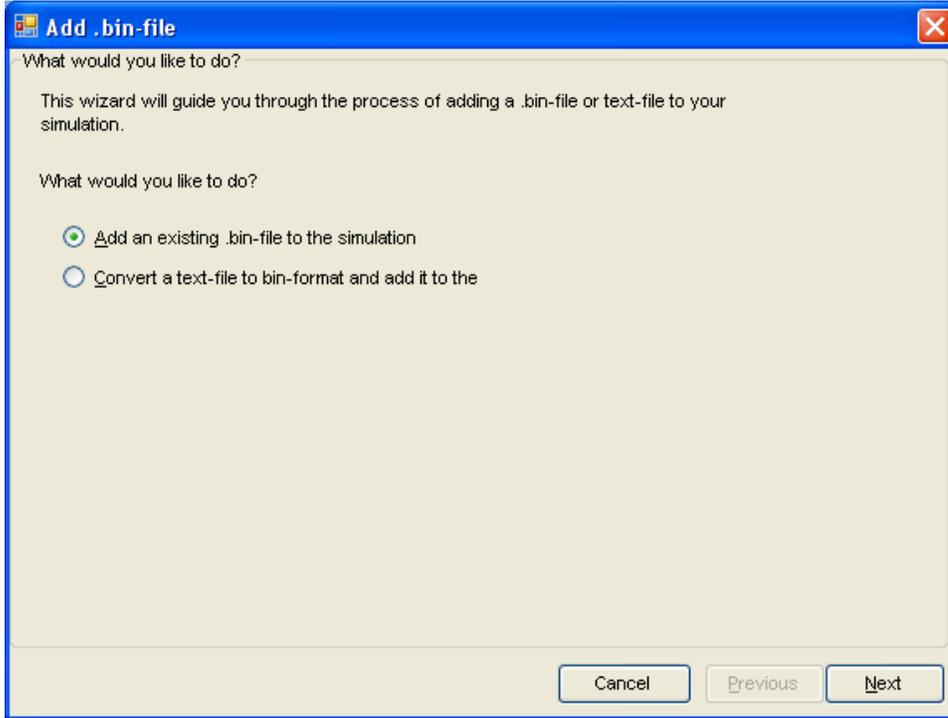


### 3.2 Simulation setup

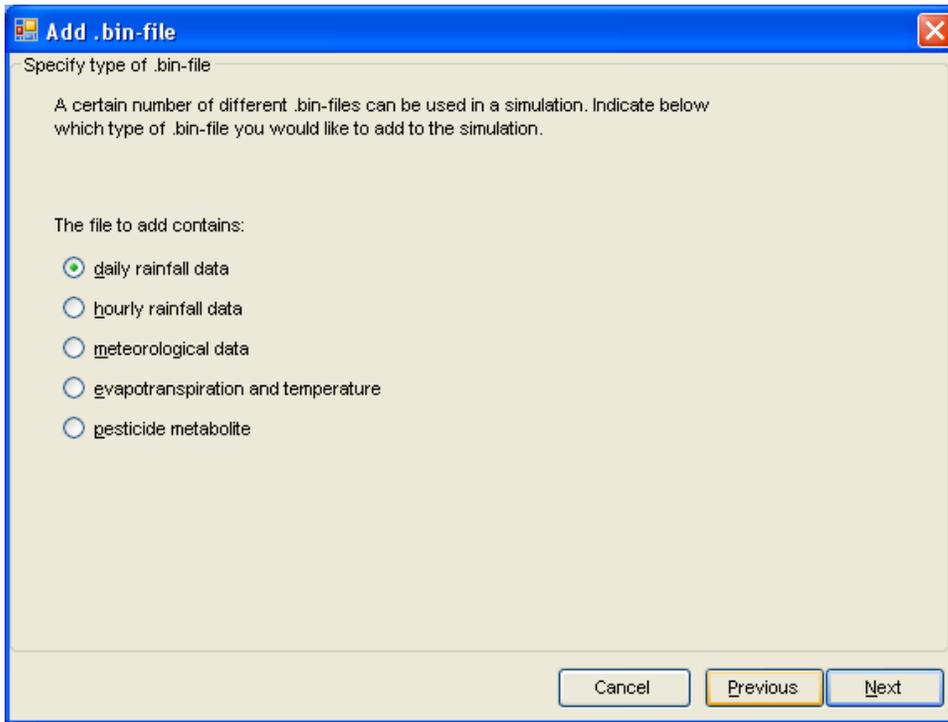
In this screen, you define all the driving data files (e.g. weather data files) that are needed for the currently defined simulation. Click on “Add file(s)”.



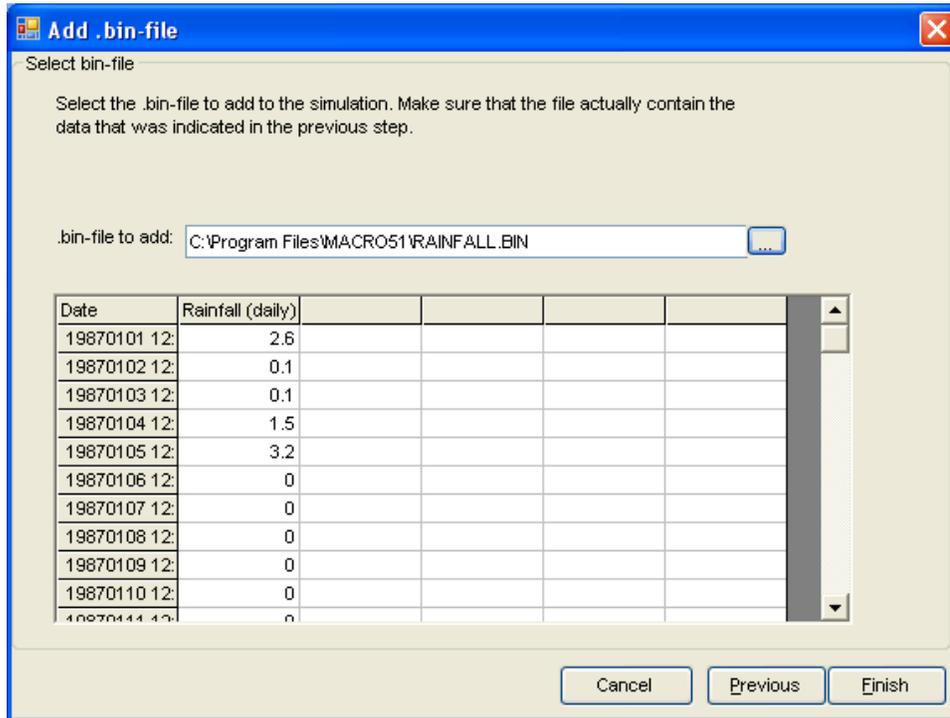
A Wizard will then guide you through the process of defining a driving data file.



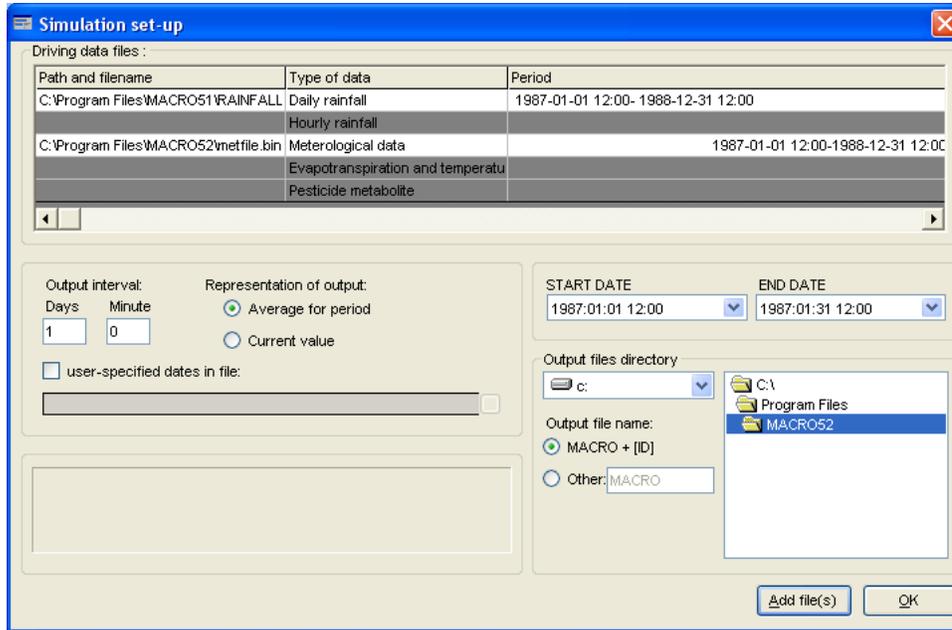
If you have already created the required \*.bin file, click “Next”, and then define the kind of driving data that the file contains.



Then select the .bin file using the browse button, check that the data appears correct, and click on “*Finish*” to add the file to the simulation.



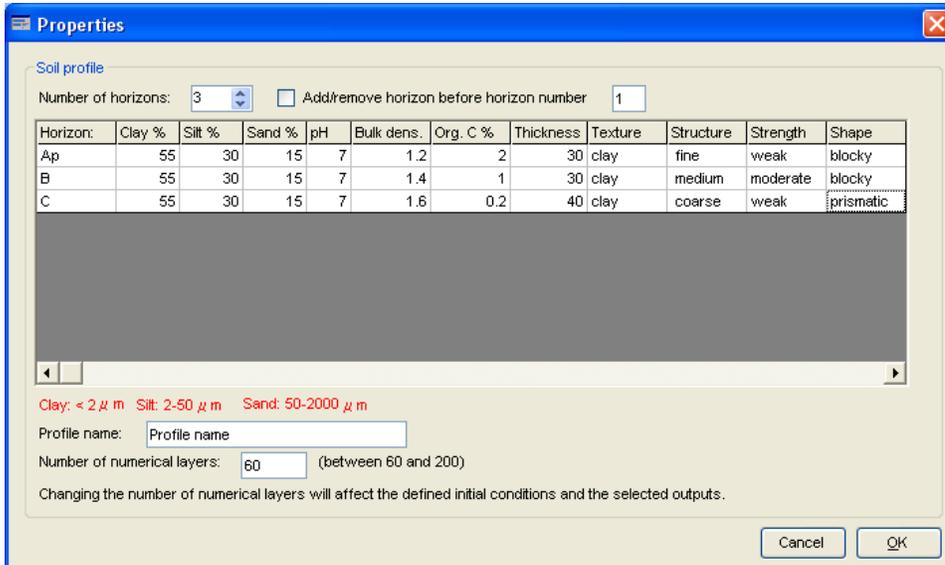
You will then return to the main menu under “*Simulation set-up*”. On this screen, you also set the start and stop times for the simulation and define how you would like the outputs to be calculated. You can choose between values calculated as averages for each output interval or the actual values at the end of each output interval. Output intervals can either be defined as a constant value (e.g. daily) or irregularly spaced times can be read in from a \*.bin file. An output directory path and file name can also be specified here. The default name is MACROXXX.bin where XXX is a unique run ID number given to the simulation.



### 3.3 Soil properties

Here, you can give a name to the soil profile that you are simulating. You also define basic properties of each horizon in the soil profile. Each horizon is characterised by a designation, a thickness, the texture (% sand, silt, clay), FAO soil structure, pH, organic carbon content (%) and bulk density. These properties of the soil are not model parameters as such, but can be used to estimate hydraulic parameters in the model (hydraulic conductivity, water retention) using inbuilt pedotransfer functions (see Appendix 1). The organic carbon content, clay content and bulk density are also used to automatically estimate thermal properties of the soil (heat conductivity, thermal capacity, see Larsbo and Jarvis, 2003). You should ensure that the horizon designation for the topsoil includes the capital letter 'A'. This is because different pedotransfer functions for soil hydraulic properties are used for the topsoil and subsoil horizons, and the program will only recognize a topsoil horizon by the designation 'A'.

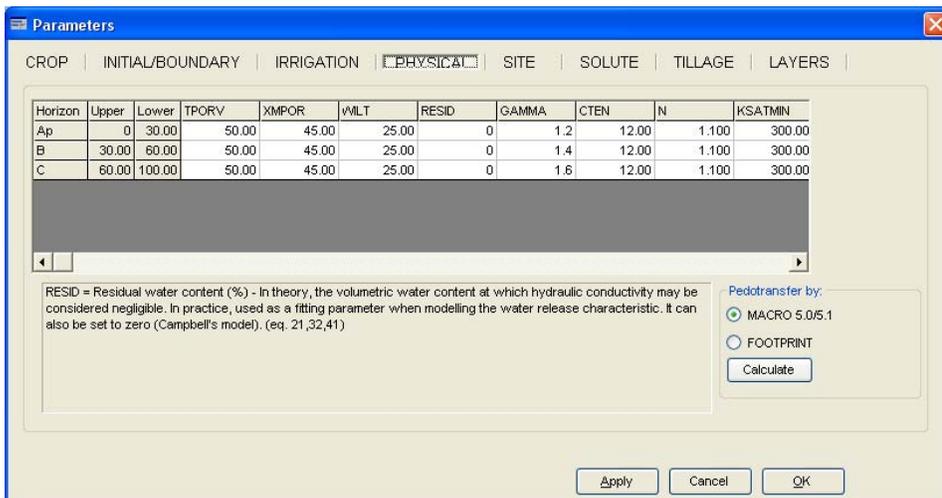
You should also specify the number of numerical layers in the profile. This may vary between 60 and 200 depending on the depth of profile you are simulating. Internally, the program then allocates these numerical layers to each specified horizon. Model parameters are constant for all numerical layers in each horizon.



### 3.4 Parameters

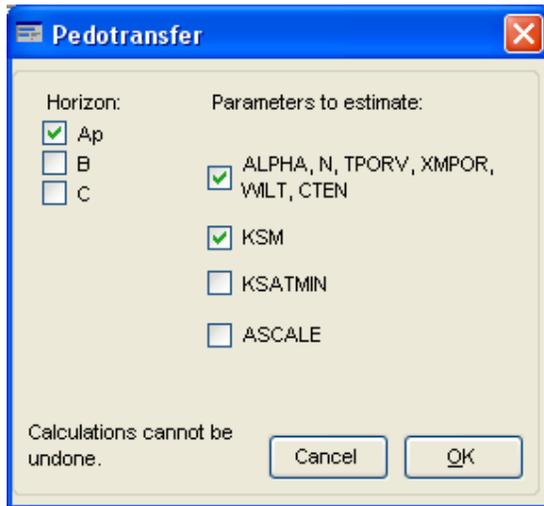
Once you have set all the options and provided basic soil properties, you should click on “Parameters” to define all the parameter values required by the model for this simulation.

These parameters are grouped under eight headings: ‘Crop’, ‘Initial/Boundary’, ‘Irrigation’, ‘Physical’, ‘Site’, ‘Solute’, ‘Tillage’ and ‘Numerical Layers’. For a new project, all parameters will have been given default values, and this is denoted by the black colour.



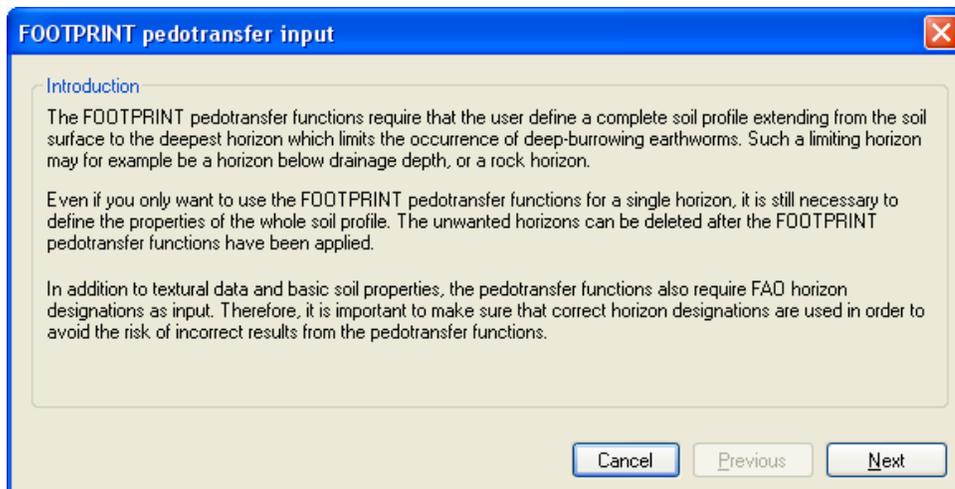
When you overwrite the default value, the font colour will change to green to denote a user-defined parameter value. ‘Tool-tip’ help text is available for all parameters, together with cross-references to the relevant equations in the technical description (Larsbo and Jarvis, 2003).

Soil physical properties can also be estimated from the basic soil properties you have already defined by using the inbuilt pedotransfer functions. There are two sets of pedotransfer functions: (i) pedotransfer functions from the MACRO 5.0/5.1 models, and (ii) pedotransfer functions from the FOOTPRINT project. Select the appropriate set of pedotransfer functions you wish to use and click on the “*Calculate*” button. For the MACRO5.0/5.1 set of pedotransfer functions the following dialogue is then shown:



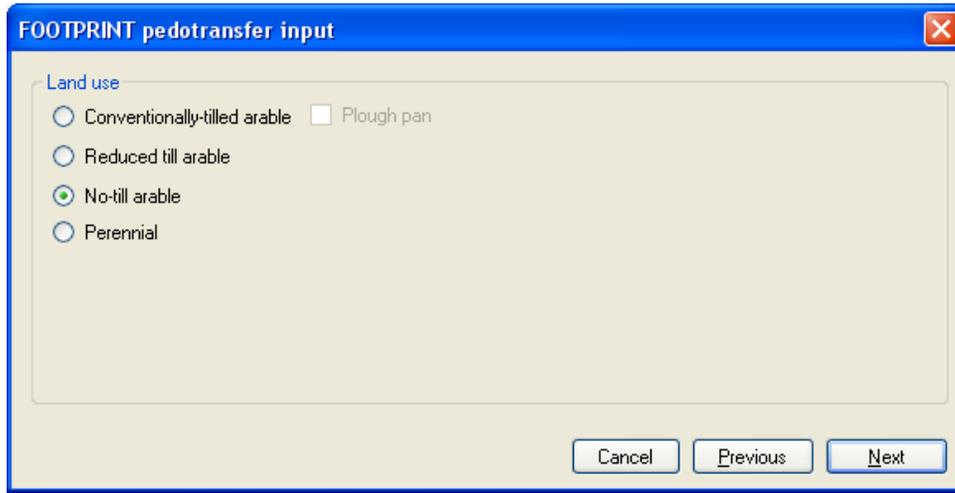
You then specify which parameters you would like to estimate, and for which horizons in the soil profile. These pedotransfer functions are described in Appendix 1. The calculations are made when you click on *O.K.*, and the estimated parameter values are then shown in red font.

If you select the FOOTPRINT pedotransfer functions, you will first get some general but important information about the use of the pedotransfer functions.



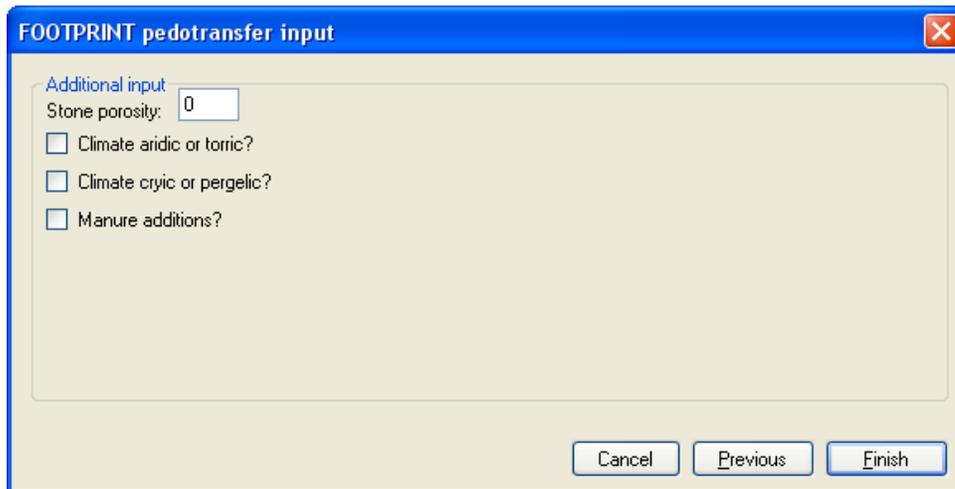
Following this step you need to supply information on the land use at the site. There are four options available: (i) conventionally-tilled arable, (ii) reduced till arable, (iii) no-till

arable, and (iv) perennial. The effects of these options on the estimated parameter values are described further in Appendix 1.



The screenshot shows a dialog box titled "FOOTPRINT pedotransfer input". The "Land use" section contains five radio button options: "Conventionally-tilled arable", "Reduced till arable", "No-till arable" (which is selected), "Perennial", and a checkbox for "Plough pan". At the bottom of the dialog are three buttons: "Cancel", "Previous", and "Next".

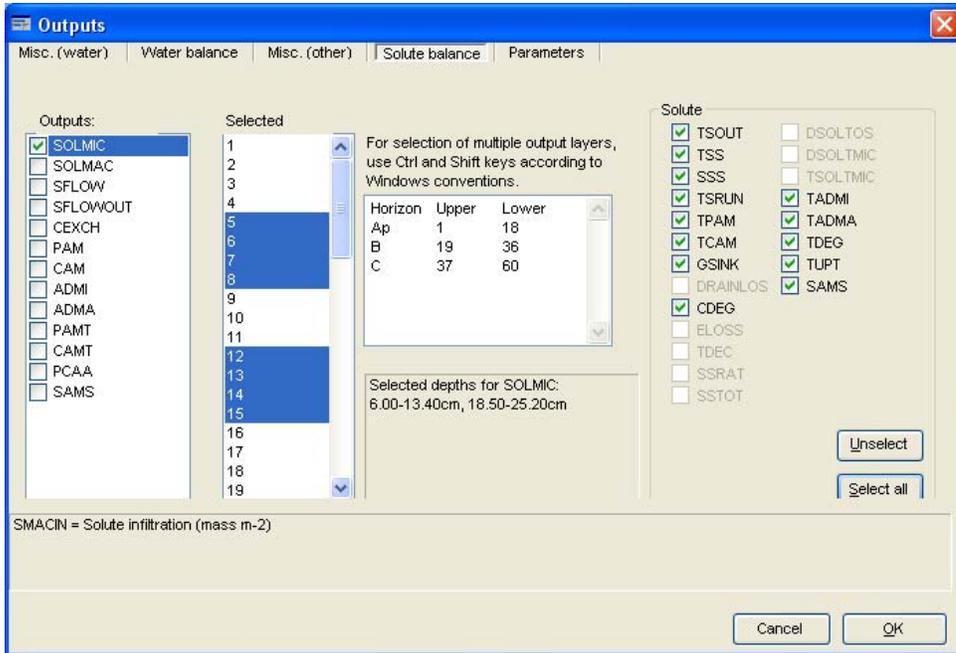
Finally you should supply some additional information related to the climate at the site and whether manure additions take place. As above, the effects of these options on the estimated parameter values are described further in Appendix 1.



The screenshot shows the same dialog box, but now the "Additional input" section is active. It includes a text box for "Stone porosity" with the value "0". Below this are three checkboxes: "Climate aridic or torric?", "Climate cryic or pergelic?", and "Manure additions?". At the bottom of the dialog are three buttons: "Cancel", "Previous", and "Finish".

### 3.5 Outputs

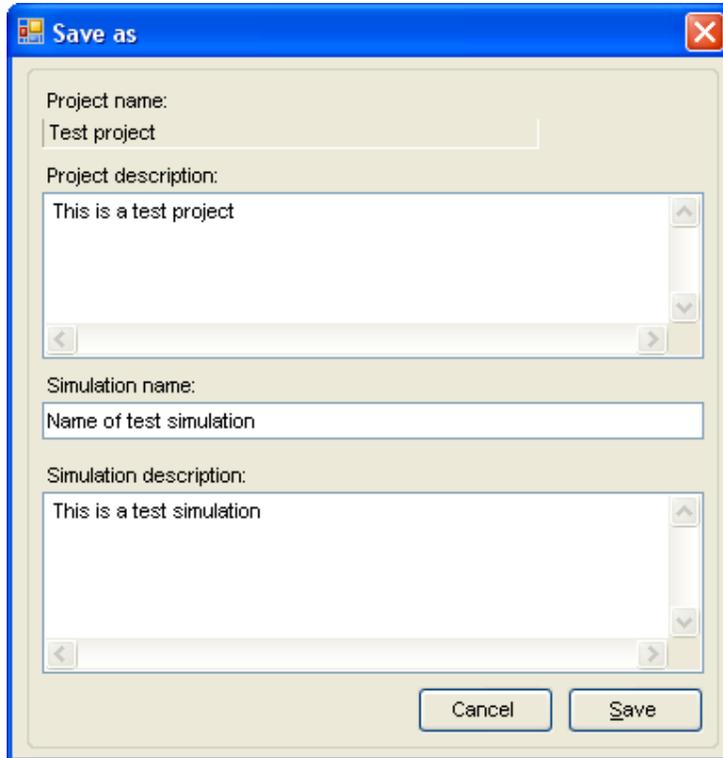
Here you can select output variables grouped under five headings: 'Water balance', 'Solute balance', 'Miscellaneous (water)', 'Miscellaneous (other)' and "Parameters".



For array output variables, you can select single or multiple numerical layers in the profile, using the Ctrl and Shift keys according to Windows conventions.

## 4 Saving a simulation

A MACRO simulation under the umbrella of the currently defined project can be defined at any time by clicking on “*File->Save As*”.



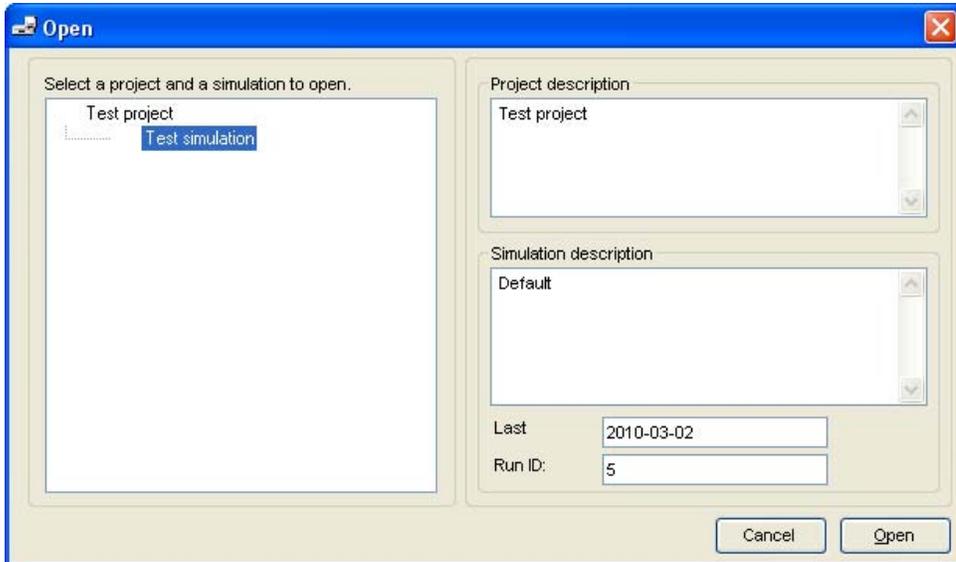
Give an appropriate name and description for the simulation and click on 'Save'. All current settings will be saved to the database under this name. You can save any further changes that you may make to the definition of the simulation by clicking on "File->Save" at any time.

#### **4.1 Save to parameter file**

By clicking on "File->Save to parameter file" it is possible to save the current MACRO simulation to a so called parameter file. The parameter file is a text file that completely defines the MACRO simulation and is meant to be used together with a specific tool to run simulations from parameter files outside the MACRO GUI. This tool is described in a separate document.

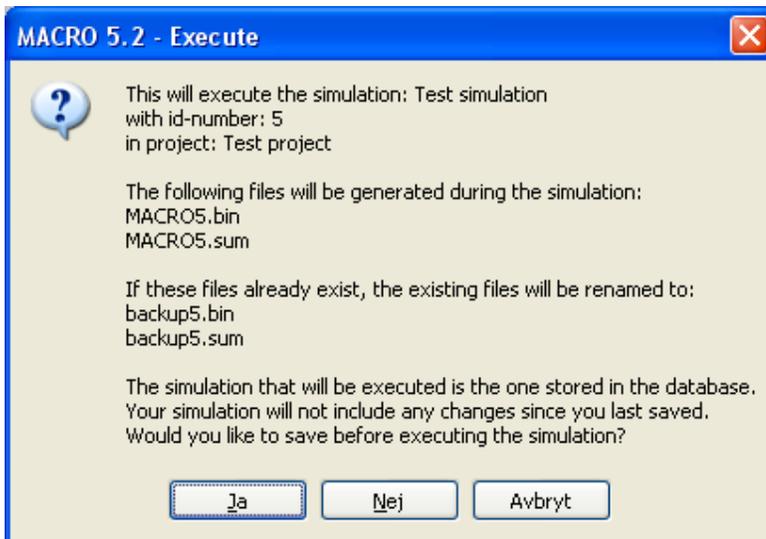
### **5 Opening an existing project and simulation**

You can open an existing project and previously defined simulation by clicking on "File->Open". You will find descriptions of each project and simulation, information on when each simulation was last modified and its unique Run ID.

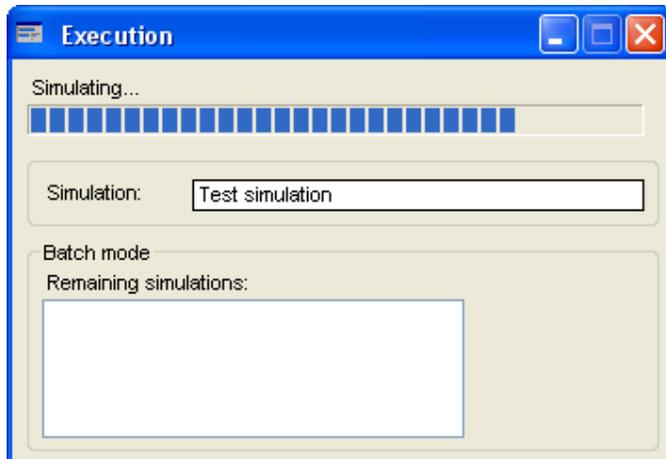


## 6 Executing simulations

The currently defined simulation can be executed at any time by clicking on “*Execute* > *Current*”. You will then get some information on the simulation name, project name, a Run ID for the simulation, and the output filenames. You can also save the current definition of the simulation to the database prior to running it.



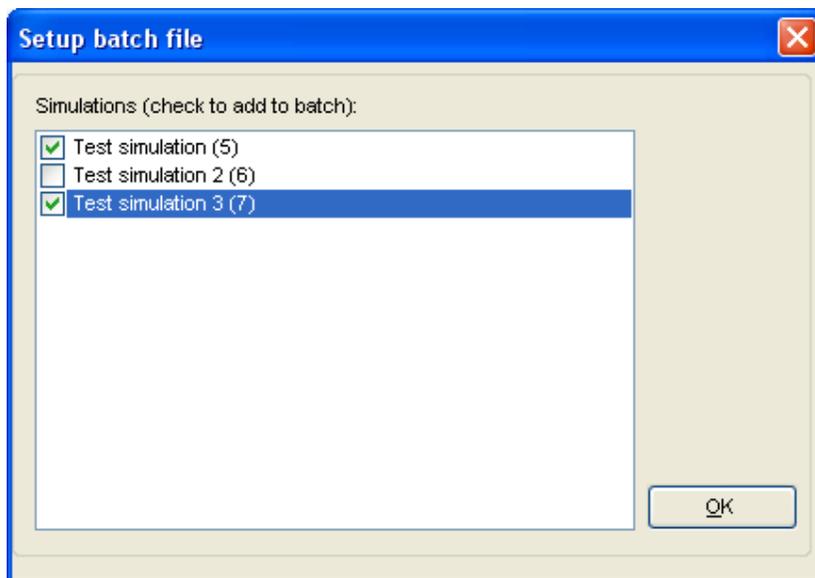
When the simulation starts, you will see the following screen.



On completion, you are provided with some information on how long the simulation took to run, and the mass balance errors for both water and solute.

## 6.1 Batch mode

Batch mode can be used to run a series of simulations. Each simulation must first be defined and saved to the database. You create the batch run under "*Tools->Setup batch file*". Add a simulation to the batch file by checking the simulation in the list of available simulations. Click *O.K.* to finish.



The batch file is run under "*Execute->Batch file*".

## 7 Tools

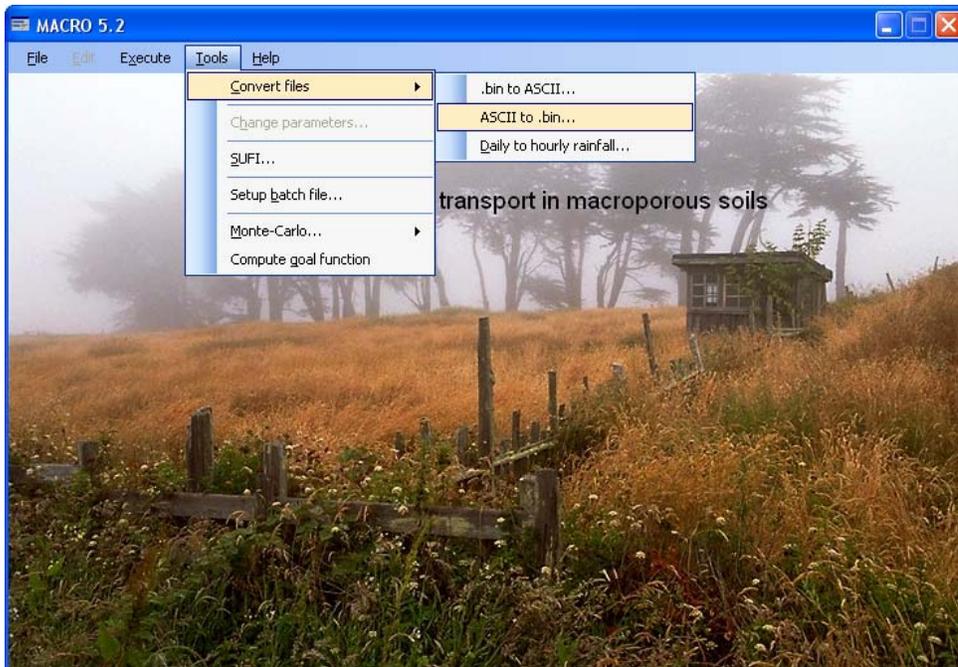
### 7.1 Converting files

#### 7.1.1 File conversion from binary format to ASCII format

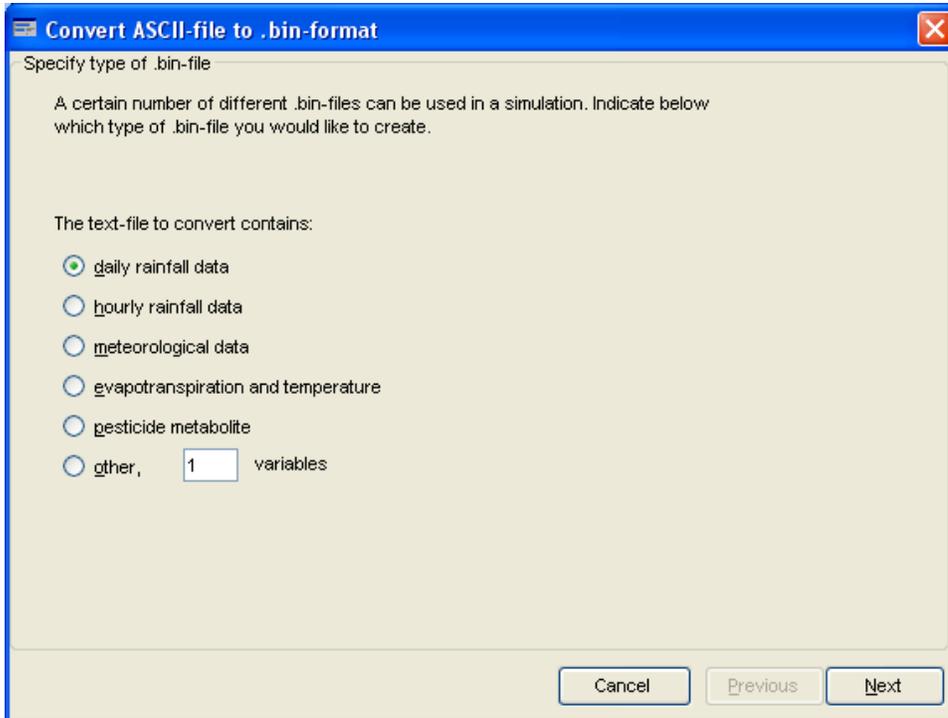
You can convert the \*.bin file to a text file using the file conversion wizard included in MACRO 5.2. Click on “*Tools->Convert files->.bin to ASCII*” (see figure below). You can then import the data into any graphics program you like (e.g. Excel). Note that the text file created by the wizard is ‘Tab delimited’.

#### 7.1.2 File conversion from ASCII to binary format

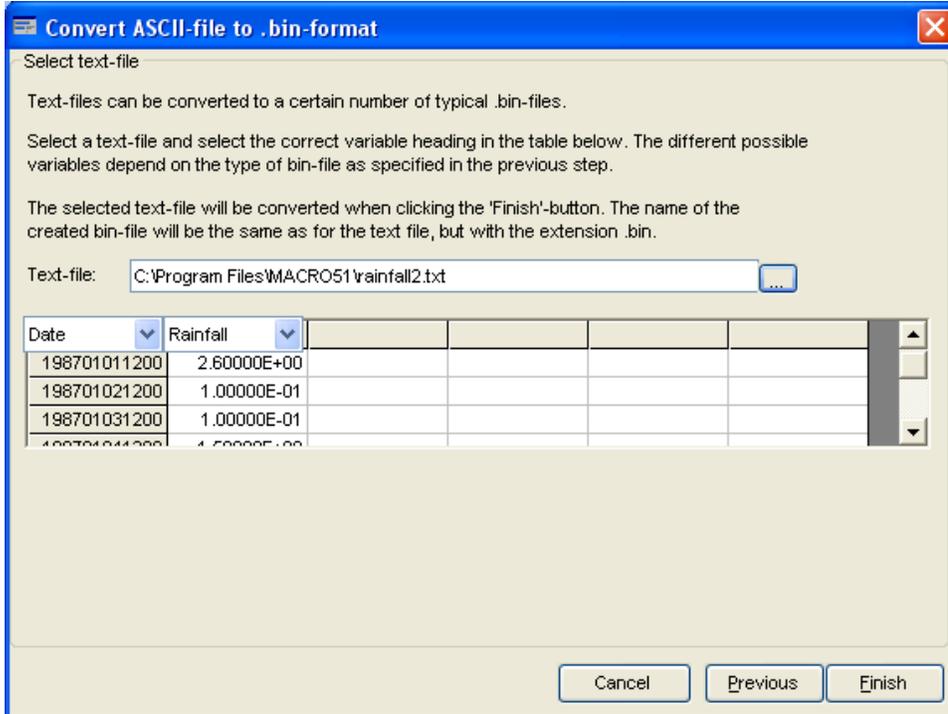
Text files can be converted into the .bin format required by MACRO using the file conversion wizard that you will find under “*Tools->Convert files->ASCII to .bin*”.



First, you specify the type of data in the file. If the file contains measured data, you should click on ‘*Other*’ and then fill in the correct number of variables (the date/time is not counted as a variable).

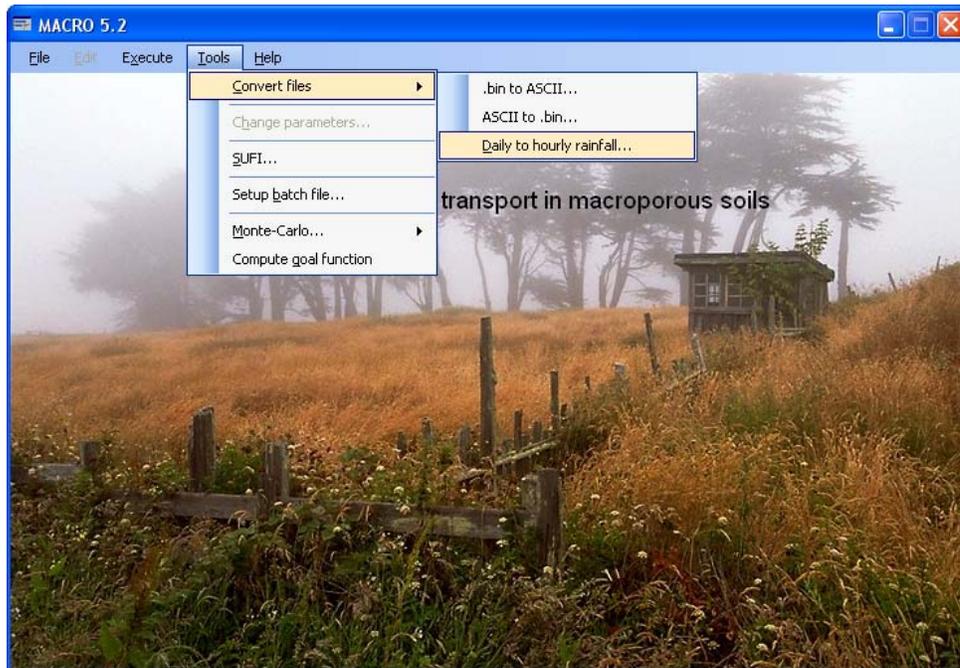


Click on 'Next', select the text file using the browse button, fill in appropriate column headings for each variable, and finally click on 'Finish'.



### 7.1.3 Daily to hourly rainfall

Hourly rainfall data files can be generated from daily rainfall data using an inbuilt stochastic disaggregation ‘cascade’ model, based on the temporal scale-invariance of rainfall patterns (Olsson, 1998). You will find this tool under “*Tools-> Convert files -> Daily to hourly rainfall*”.



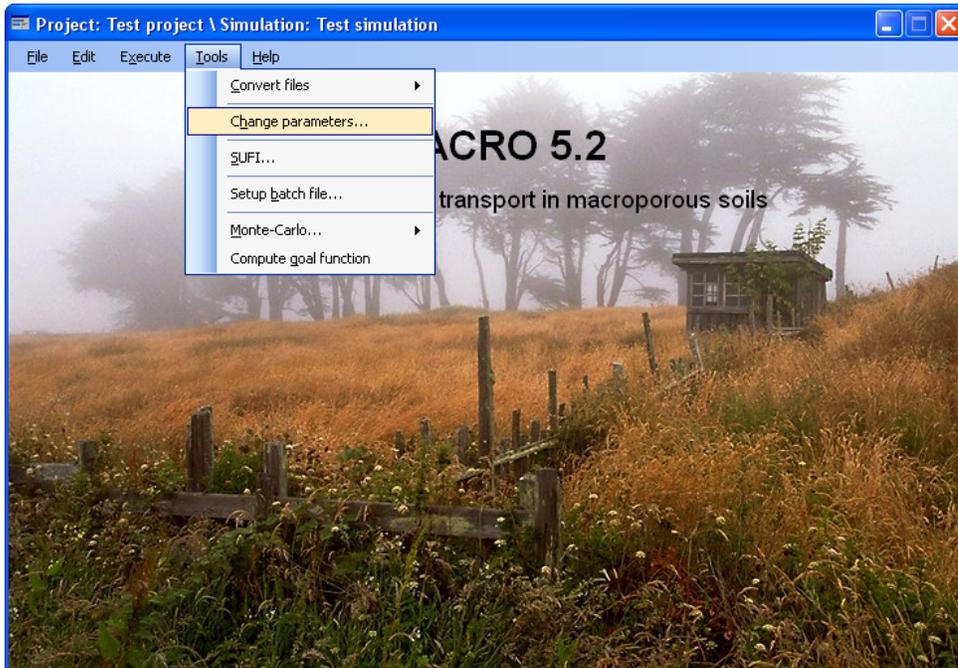
As model input, the disaggregation tool requires the user to input a series of probabilities that define the temporal distribution of rainfall as the time resolution changes from daily to hourly.

Use the browse button to select the .bin formatted file containing daily rainfall data. Define a name for the output file that will contain the hourly data. When you click on “*Disaggregate*”, the tool will make a stochastic conversion from daily to hourly rainfall data based on the probabilities given in the table on the left.

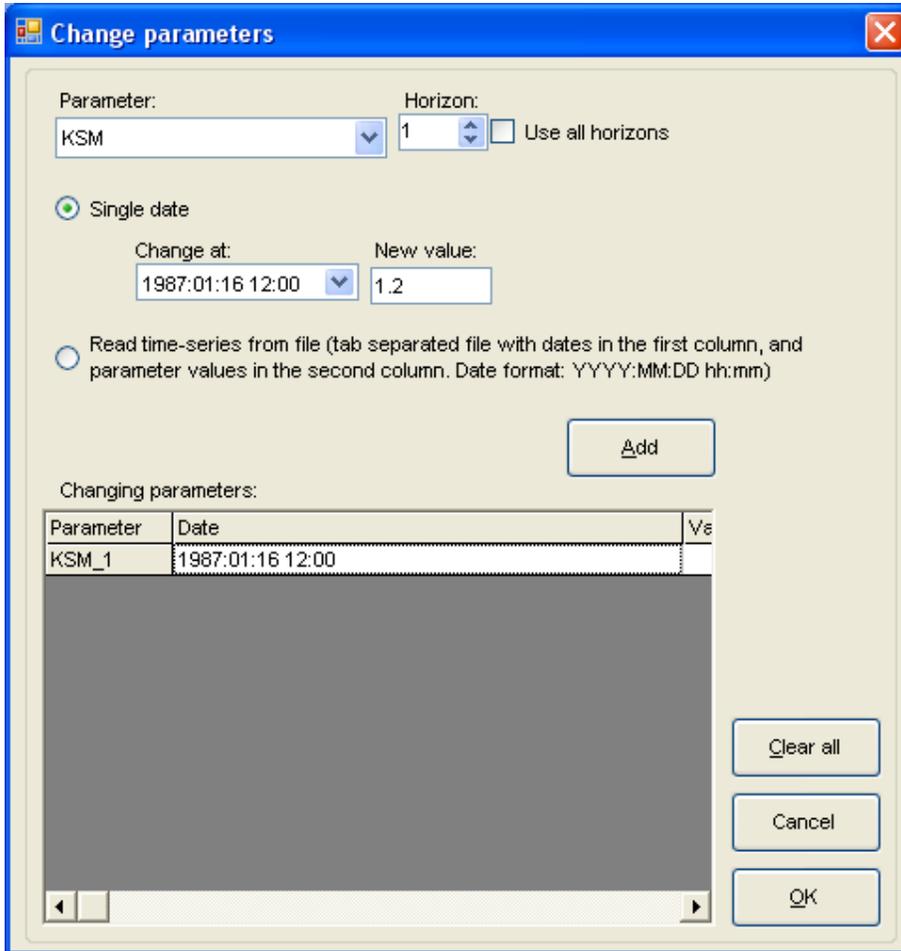
The model disaggregates rainfall data by doubling the time resolution for each disaggregation level. In MACRO 5.2, the daily rainfall series is first disaggregated to a series with a 45-minutes resolution. Each 45-minute interval is then further divided into three 15-minute intervals. The 15-minute intervals are added together four and four to create a series with an hourly time resolution. Rainfall ‘boxes’ are defined, classified into four different types depending on their position within a continuous rainfall period: ‘start’, ‘end’, ‘enclosed’ and ‘isolated’. These classes are further subdivided into ‘boxes’ that have higher or lower rainfall amounts than the observed mean (indicated with an ‘a’ for above and a ‘b’ for below in the tool). Each of these classes is associated with three probability parameters (wet/dry, dry/wet, wet/wet). Three default parameterisations are available in MACRO 5.2, based on analyses of rainfall patterns in southern Sweden, England, and Brazil (Olsson, 1998; Güntner et al., 2001).

## 7.2 Change parameters

It is possible to change model parameter during a simulation by using the tool “*Change parameters*”.

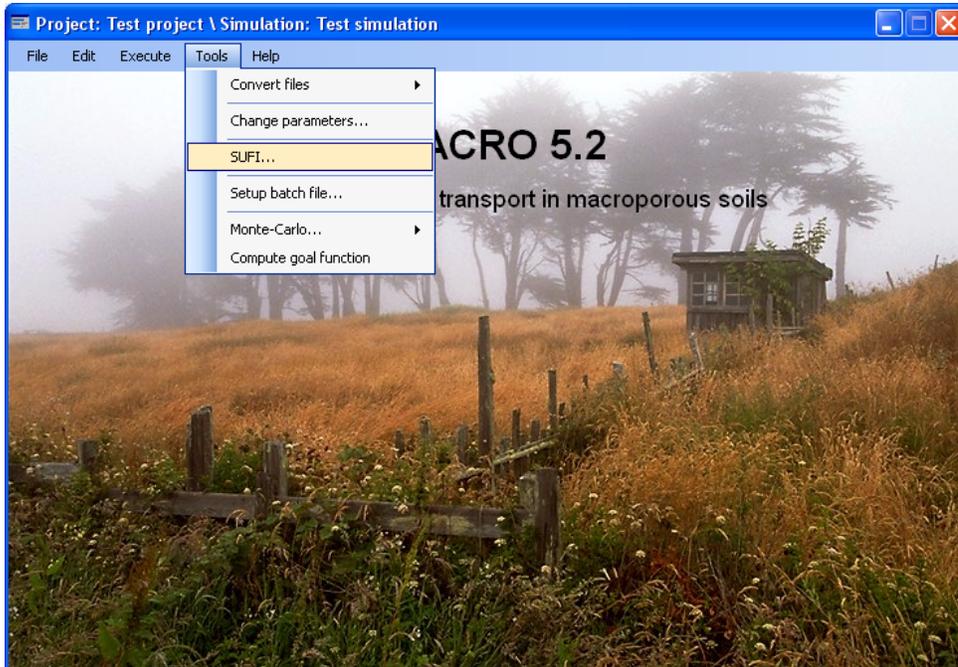


By clicking “*Tools->Change parameters*” you are able to specify which model parameters you wish to change and at what time during the simulation. You can choose to change a parameter at only a specific date and time, or read a time series from a separate file with the dates at which a change should occur and the new parameter value. By clicking “*Add*” the single parameter is added to the table or you are prompted to browse for the file with the time series from which the dates and parameter values are read and added to the table.



### 7.3 Parameter estimation with SUFI

The SUFI parameter estimation methodology (Abbaspour et al., 1997) has been incorporated into MACRO 5.2. You will find SUFI by clicking on “Tools->SUFI”.



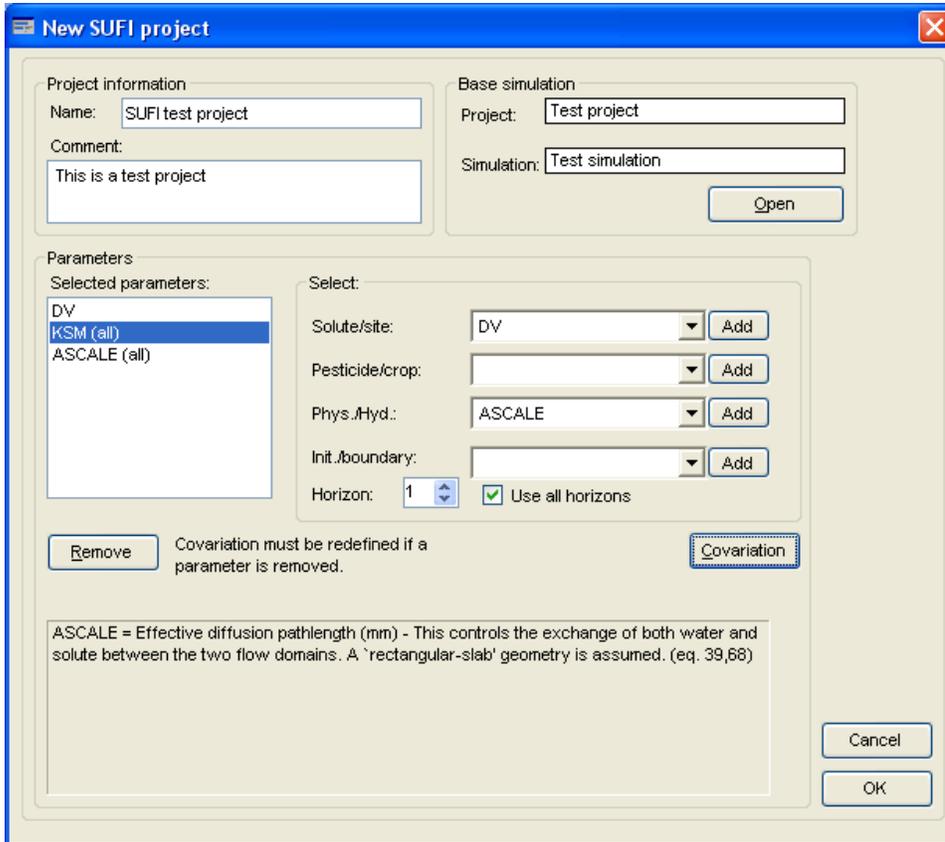
The work with SUFI is organised into projects and iterations. Each project is characterised by a name, a base simulation and a number of specified parameters that you would like to estimate.

Since the number of simulations to be run in any given calibration problem can be very large, you have the possibility with MACRO 5.2 to partition the simulations between any number of participating computers. One database, located on a server, then controls the execution of the project, with respect to identifying values of the parameters to be calibrated and partitioning of the simulations between computers. This database is specified at the base of the main SUFI screen (see below). The simulations are run on each individual computer, where a local database, defined under the main menu of MACRO 5.2, controls all other aspects of the simulation (e.g. driving data files, fixed parameter values).

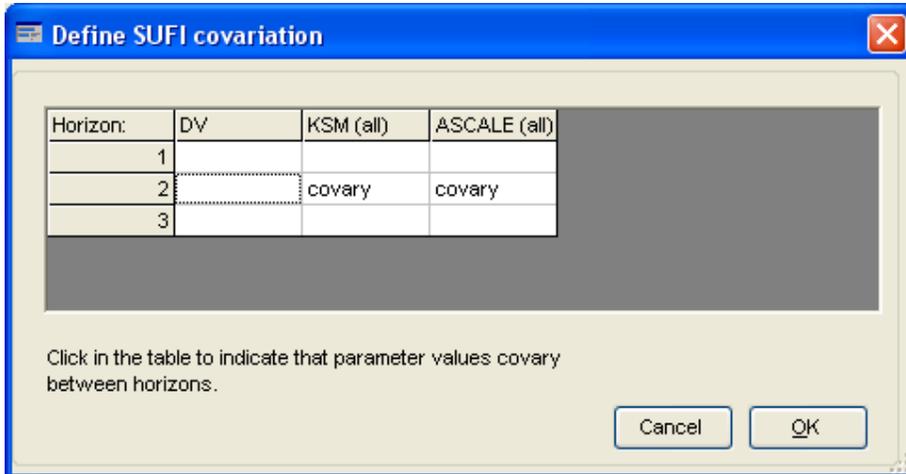
### **7.3.1 Creating a new project**

Click on “*File->New Project*” in order to create a new SUFI project.

Give the project a name and add a descriptive comment. Click “*Open*” to select the base simulation. This is a previously defined MACRO 5.2 simulation that defines all the parameters that are given constant values during the iteration.

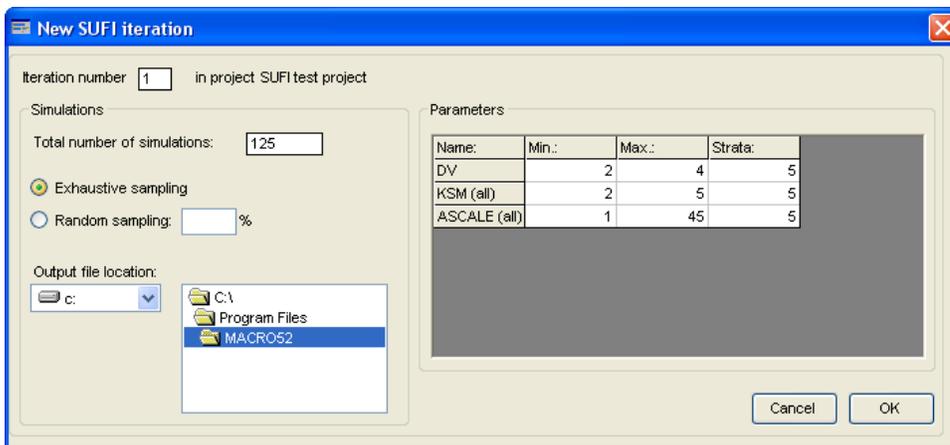


After selecting the base simulation, add the parameters that you want to include in the calibration. MACRO parameters describing solute, pesticide and physical- hydraulic properties can be selected in the three drop-down boxes. For parameters that vary with depth in the profile, you should specify the relevant horizon or choose to vary the parameter for all horizons. If you want a parameter to vary in the same way for two or more horizons, click “*Covariation*” to bring up the dialogue box shown below. By clicking on the table, you can toggle the text “*Covary*” on and off. In this example, ASCALE and KSM will always take the same values in horizons one and two, while the values in horizons 3 and 4 will be unaffected.



### 7.3.2 Creating a new iteration

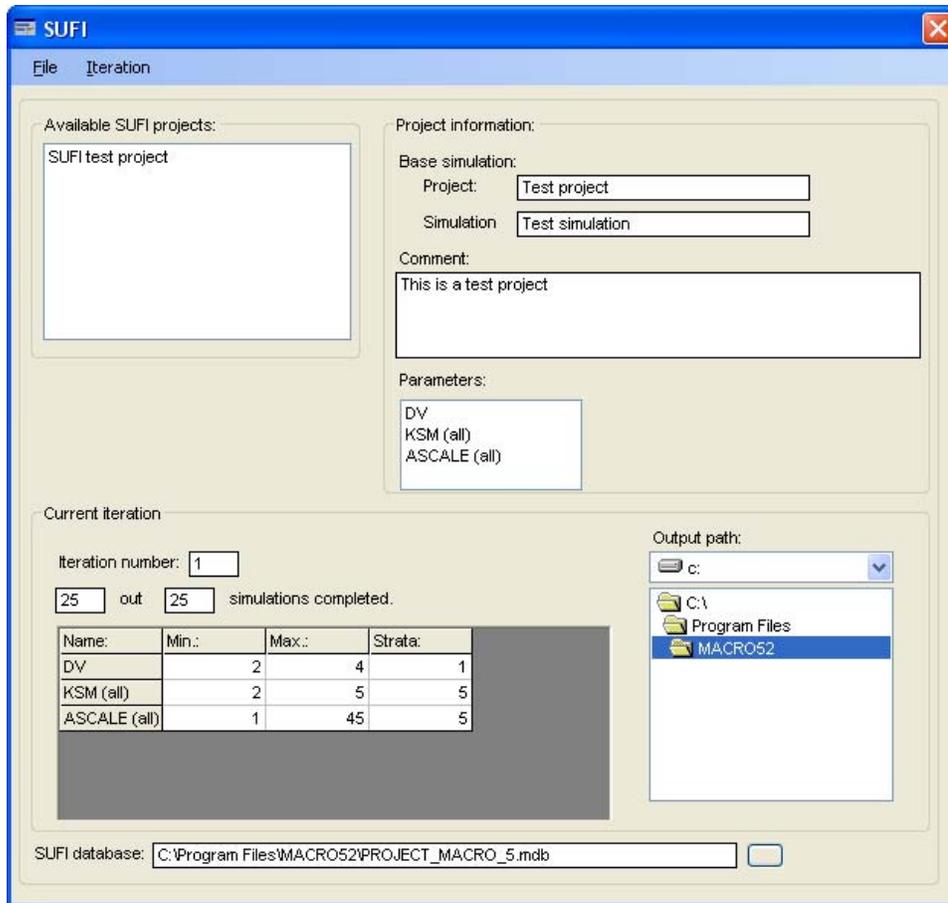
On returning to the main SUFI menu, you will find that all previously defined SUFI projects are shown in a list. Click on the project you want to work with in the list and then click on “*Iteration->New...*” to bring up the dialogue shown below.



The selected parameters are shown in the table to the right. Enter the minimum and maximum value and the number of strata for each parameter. The parameter range (i.e. maximum minus minimum) is divided into the specified number of (equal- sized) strata. You can choose either exhaustive or random sampling. Exhaustive sampling means that all combinations of parameter values are run, using the mid- point value of each strata as the parameter estimate. Random sampling means that a specified subset of all the strata combinations will be selected at random and propagated through the simulation model. The program calculates the total number of simulations that will be run. For exhaustive sampling, this is equal to the number of strata raised to the power of the number of parameters (if the same number of strata is specified for each parameter).

An output directory is also specified for the iteration. The output files from the MACRO simulation will be renamed and moved to this directory. The new file name will be “*iter\_[iteration number]\_[Run ID]\_[simulation number].bin*” (e.g. iter\_1\_5\_22.bin). In

order for the analysis tool to work, all output files should be located in the specified output file directory and they must not be renamed. Note that it is possible to change the output file directory after the iteration has been executed. This is useful if output files are moved from one computer to another. Note that the selected output times in the base simulation must correspond to the dates and times for the measured data in order to make the computation of the goal function possible. You create the iteration by clicking on “O.K.” You then return to the main SUFI menu, where the information that you just entered is displayed in the lower part of the screen. You can also change the output path at this point.



Even if you intend to execute the iteration simultaneously on several computers, the complete iteration must be defined on each computer, in exactly the same way. In practice this may be done by copying a “master” database to each individual computer.

### 7.3.3 Executing an iteration

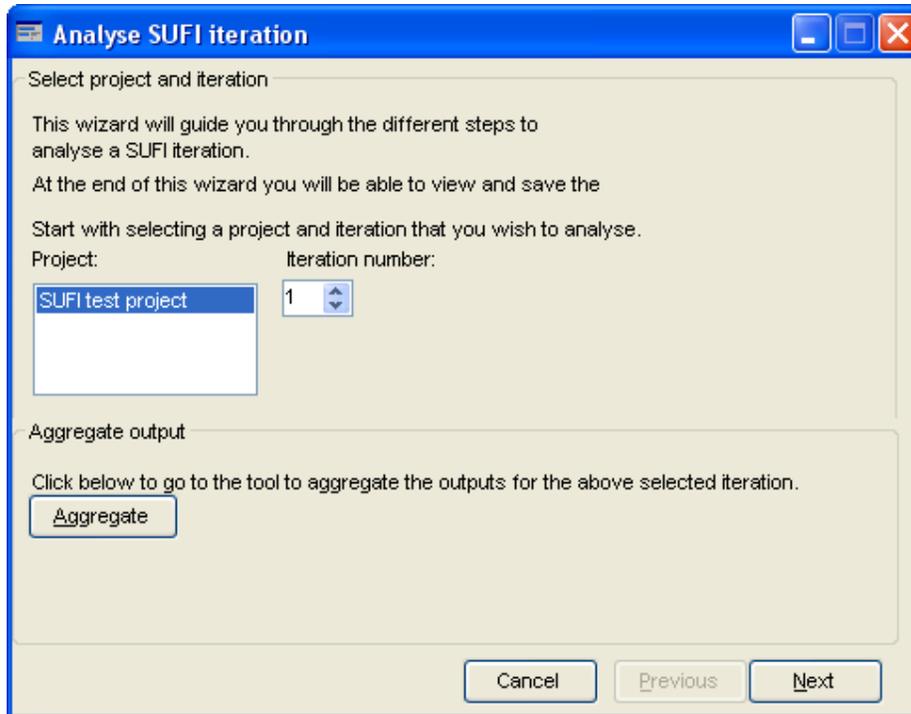
It is recommended that before executing a SUFI iteration, you first check that the selected base simulation runs properly in “manual” mode, making sure that the correct output variables have been selected.

After clicking “Iteration->Execute current” the program will execute all the simulations in the iteration. For parallel execution, you should give this command on each

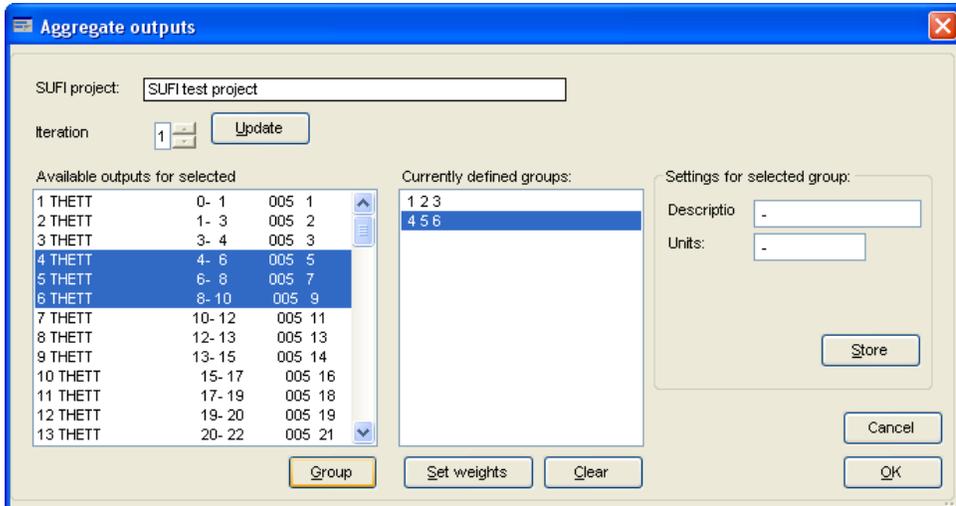
computer. The simulations will then be automatically divided up between the participating computers. It is possible to abort the execution and resume it later. You will receive a message when the iteration has been completed.

### 7.3.4 Analysing an iteration

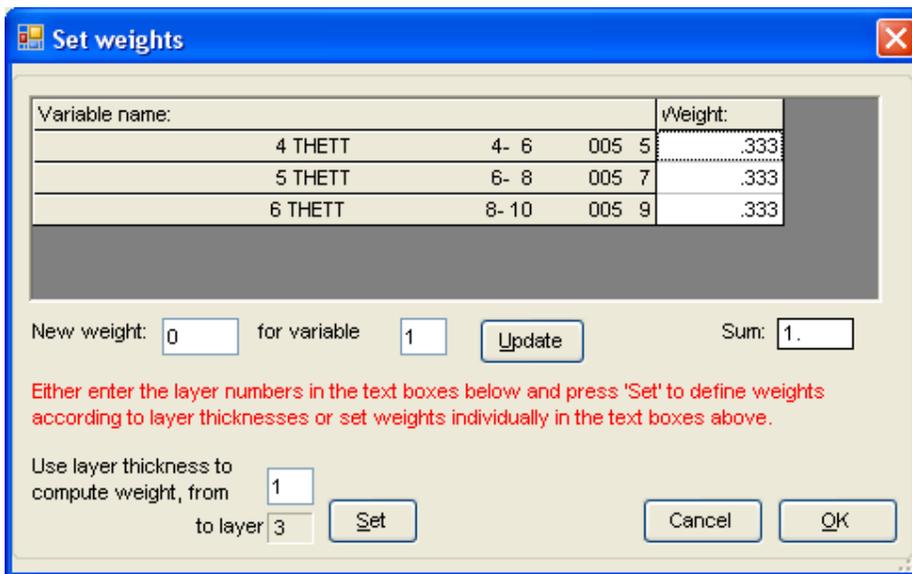
The analysis is performed in a step-wise manner with the help of an 'analysis wizard'. After clicking on “*Iteration->Analyse*”, you first select the project and iteration that you wish to analyse.



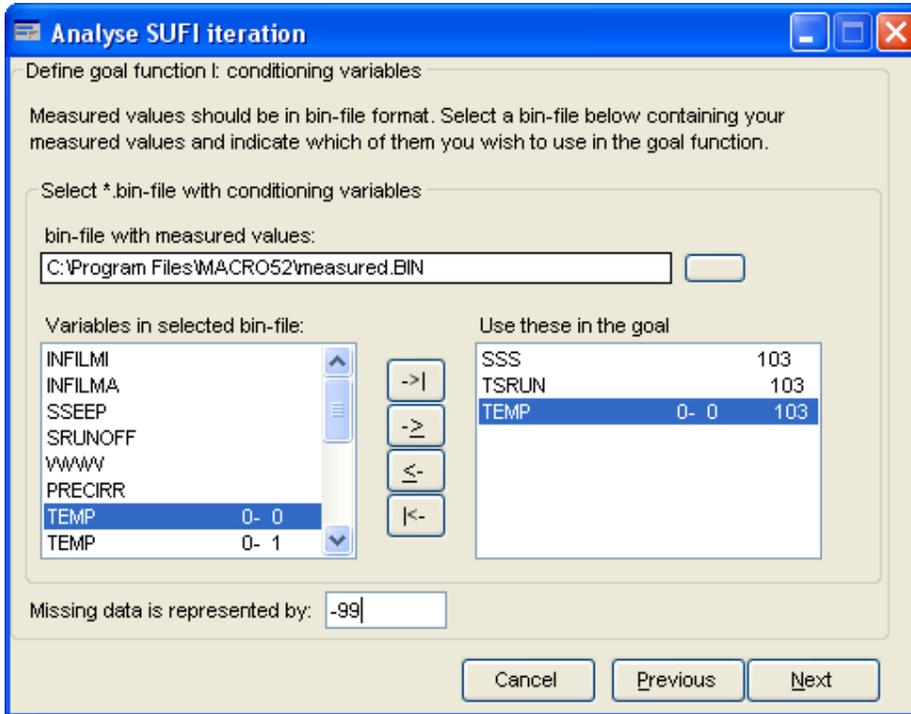
Measured data (e.g. for water contents or solute concentrations) often represents a value averaged over some depth interval in the soil, while the simulation output variable will usually be given for a larger number of smaller depth intervals corresponding to numerical layers in the model. By clicking on “*Aggregate*”, it is possible to calculate a weighted average representing the same depth interval as the measured data for the output variables in question (see below). It should be noted that the original .bin file is overwritten by this operation, so it is recommended to make a backup copy of the original file prior to aggregating outputs.



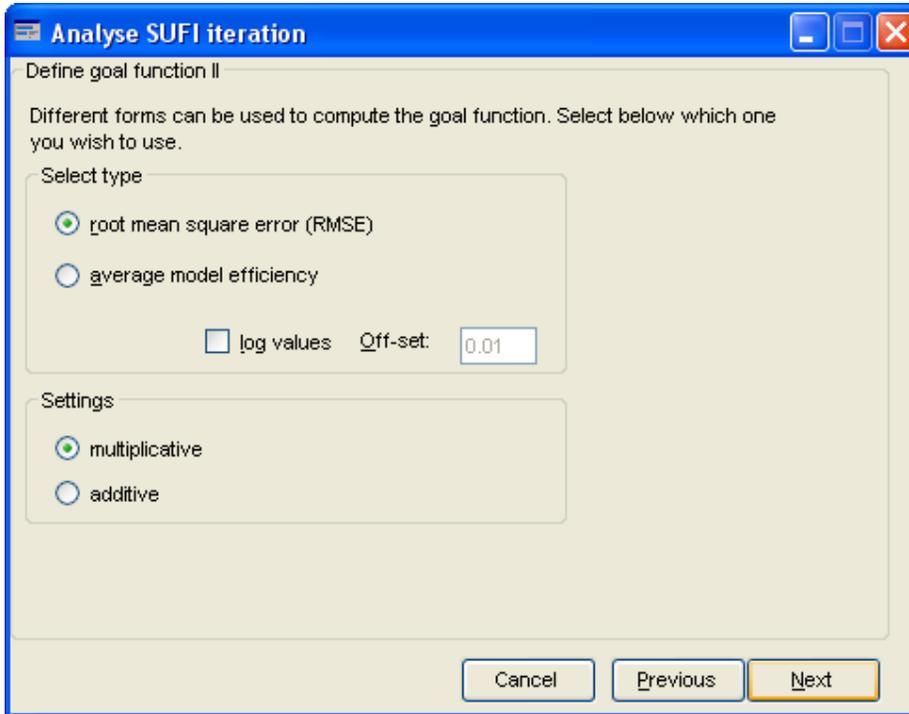
Select the variables in the left-hand list for which a weighted average will be calculated, and then click “*Group*”. Do this for each group that you wish to include. Note that all the variables used in the subsequent calibration must be selected and grouped, even variables that you do not want to modify in the weighting procedure. Click on each of the defined groups and provide the information in the textboxes to the right. To set the weights, click “*Set weights*”. Weights can be supplied either manually or calculated based on the thickness of the numerical layers.



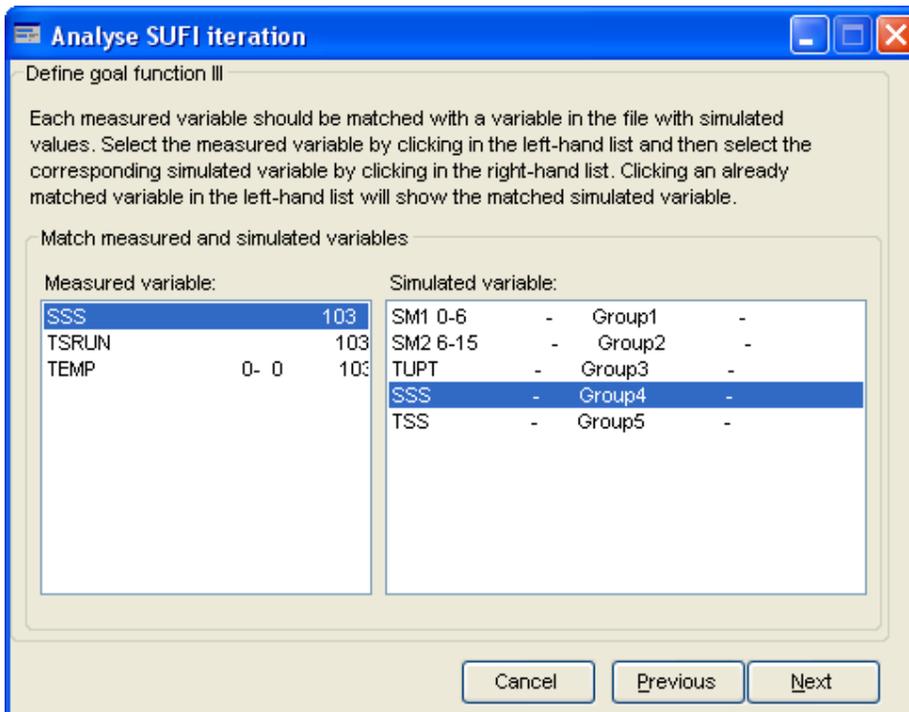
The goal function can then be defined by returning to the first screen of the wizard and clicking on “*Next*”. First, the file with the measured values is located and the conditioning variables (the measured values used for the calibration) are selected. The integer number representing missing data is also specified here.



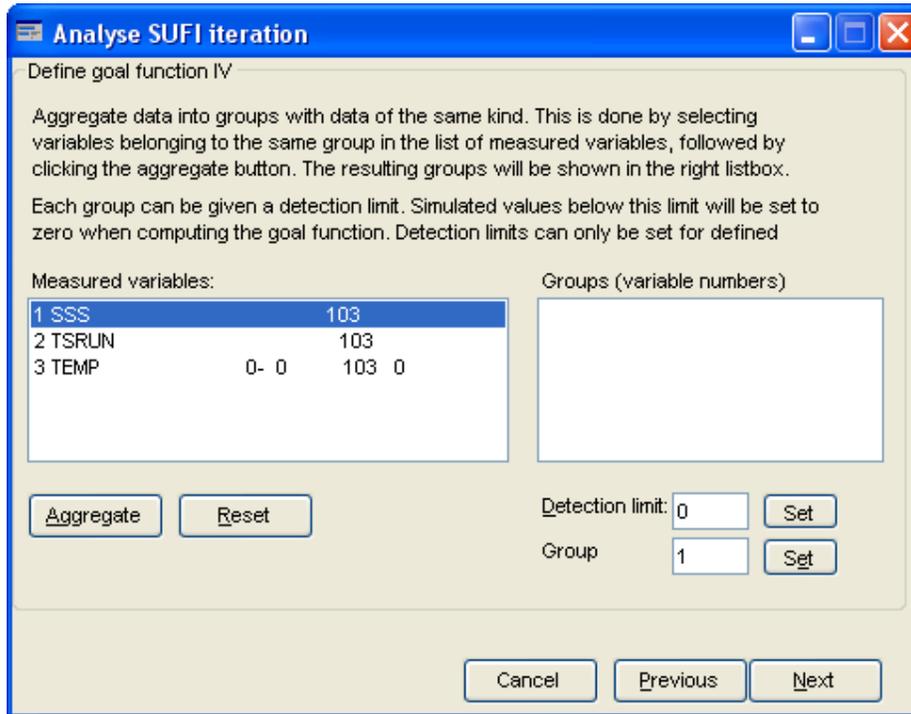
Secondly, either the root mean square error (RMSE) or the model efficiency (EF) is selected as the goal function. One value of the goal function is calculated for each ‘group’, representing different types of measured data. For the RMSE, the overall value of the goal function is obtained by either adding or multiplying the values for the different groups (select this under “Settings” as shown below). For EF, the average value is calculated.



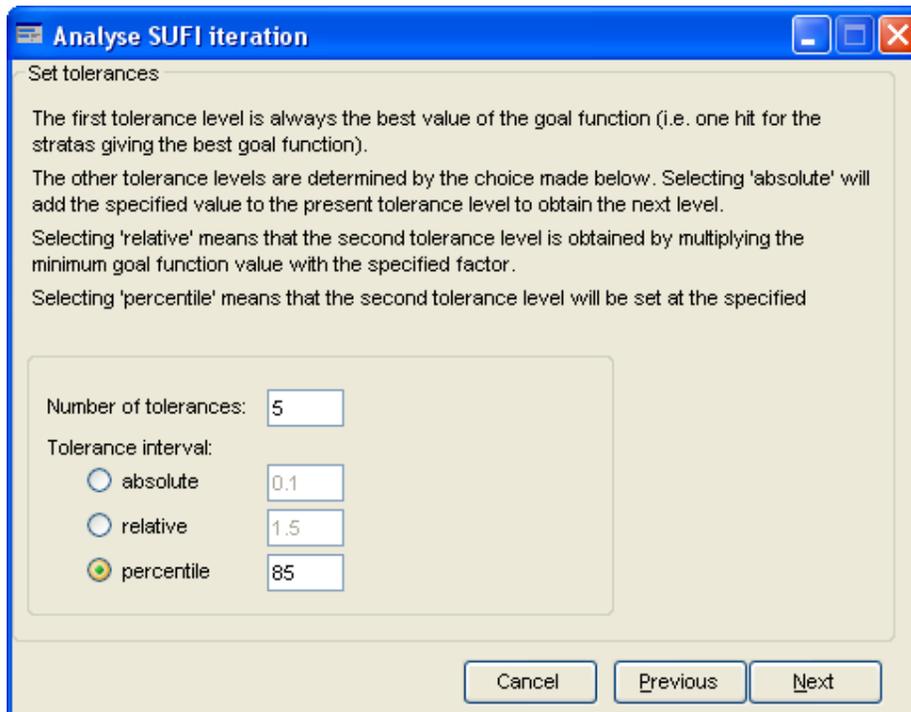
Next, the conditioning (i.e. measured) variables are matched to the appropriate simulated variables (see dialogue below). This is done by clicking on the measured variable in the left-hand list and then on the corresponding simulated variable in the right-hand list.



Next, you indicate how the measured variables should be grouped or aggregated. A detection limit for each group can be set. Each group can also be given a weight.



The number of tolerance levels is defined in the final step.

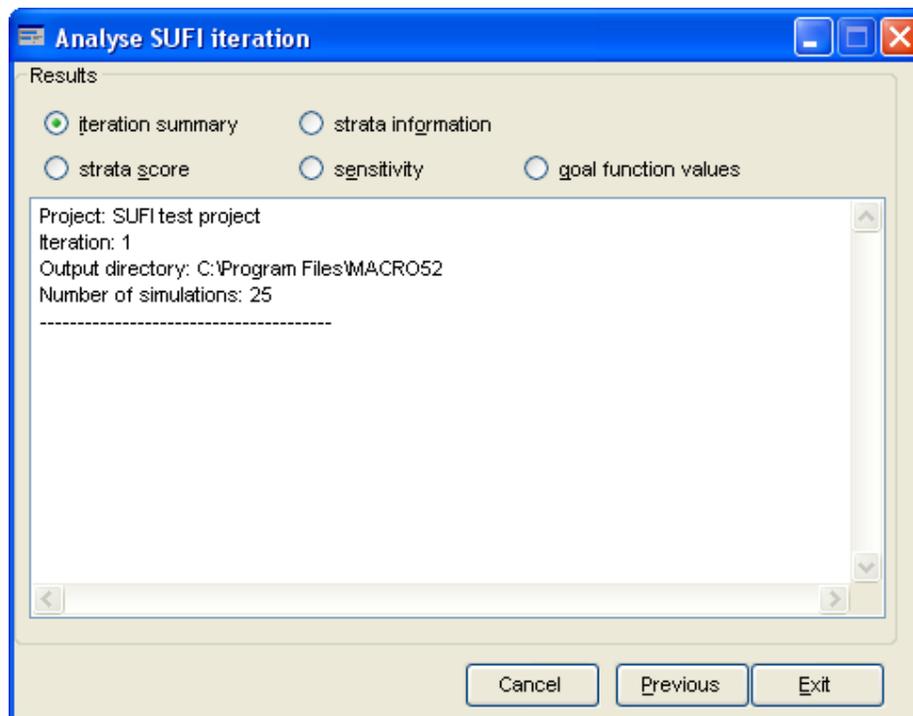


The first tolerance level is automatically set at the best value of the goal function, while subsequent levels are determined by either adding ('absolute') or multiplying ('relative') by a specified value. If the option "percentile" is selected, the second tolerance level will be set at the goal function value that is better than the selected percentile. Values that have matching dates in the measured data file and in the simulated output file are included in the goal function. This means that if a measured data point is not missing but has no corresponding date in the simulated file, it will not be included in the goal function. It is therefore necessary to make sure that the output dates are in agreement with the measured dates. The results are saved in a number of text files (Table 1).

Table 1. Output files from the analysis of a SUFI iteration.

File	Contents
StrataScores.txt	The strata score for each parameter as a function of tolerance level
StrataInformation.txt	Information for each parameter about the value for each strata
Sensitivity.txt	The average value of the goal function as a function of strata number for each parameter
Goalfunction.txt	Values of the goal function for all strata combinations (simulations)
IterationSummary.txt	Summary information about the iteration

These files are located in the iteration files output directory and can be viewed from within the wizard (see below) or opened in another program (e.g. Wordpad).



Under ‘*Strata information*’, you can find the actual parameter values used in the iteration (Table 1). Under ‘*Goal function values*’, you will find the individually calculated values of the goal function for each simulation (Table 1). ‘*Strata score*’ shows you the number of ‘hits’ for the defined tolerances for each parameter value. The information in these files forms the basis for the definition of the next iteration.

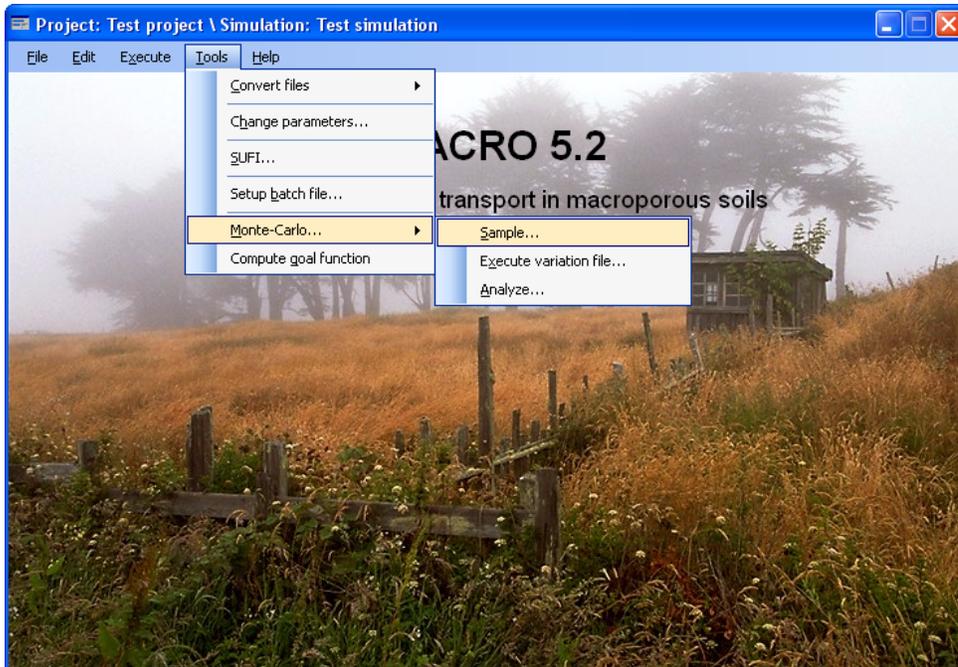
You should now decide on how to reduce the prior uncertainty domains for each parameter for the subsequent iteration, based on this information. Base your decision on a tolerance level that gives a good contrast in the number of hits between strata (i.e. only zeros and ones is not a good basis, and neither is the difference between 3 and 4 hits for example). You should not be too greedy!

### 7.3.5 Measured data

During the analysis phase, the measured data should be available in one single file, created from within the MACRO 5.2 shell. This file has the extension “.bin”. To prepare this file, follow the instructions given in section 1.1.2.

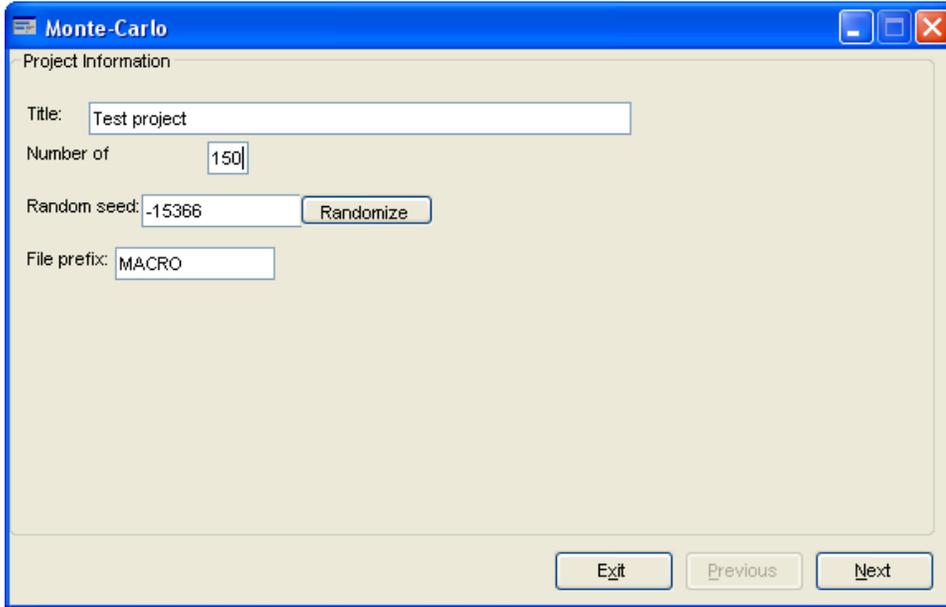
## 7.4 Monte Carlo simulation

MACRO 5.2 is loose-linked to the software package UNCSAM for Monte Carlo sensitivity and uncertainty analyses (Janssen et al., 1992). The UNCSAM programs for both ‘latin hypercube’ sampling of model parameter distributions and regression-correlation analysis of model outputs can be run from within the MACRO 5.2 GUI. These tools are accessed by clicking “*Tools->Monte Carlo*”.

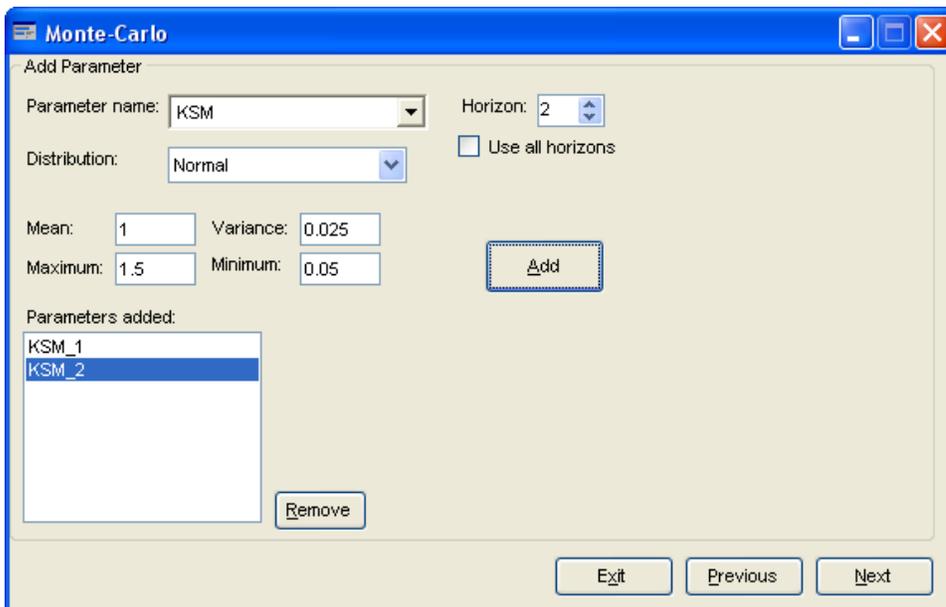


### 7.4.1 Sampling of parameter values

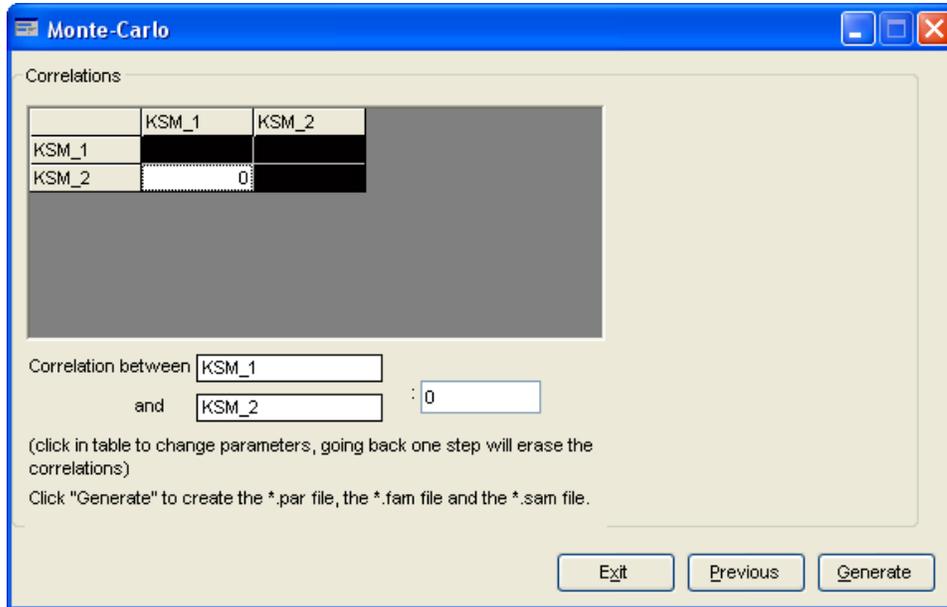
This is the first step in any Monte Carlo analysis. Click on “*Sample*” under “*Tools->Monte Carlo*”. First, some general information must be provided, including the number of simulations to be run, and the ‘seed number’ for the random number generator. It is thus possible to re-create sampled values by supplying the same seed number. All the UNCSAM files will use the specified file prefix.



In the next step, the parameters are selected and their distributions defined. You may choose between uniform, normal and log-normal distributions. For parameters that vary with depth, you can select which horizon to sample or specify that all horizons should be varied in the same way.



Next, correlation coefficients among the parameters are specified (see below). Finally, you create three UNCSAM formatted files (\*.fam, \*.sam and \*.par) by clicking on “Generate”.



The \*.fam file can be used to run all the generated MACRO parameter combinations in batch mode using the tool “Execute parameter variation file”.

#### 7.4.2 Execution of parameter variation file

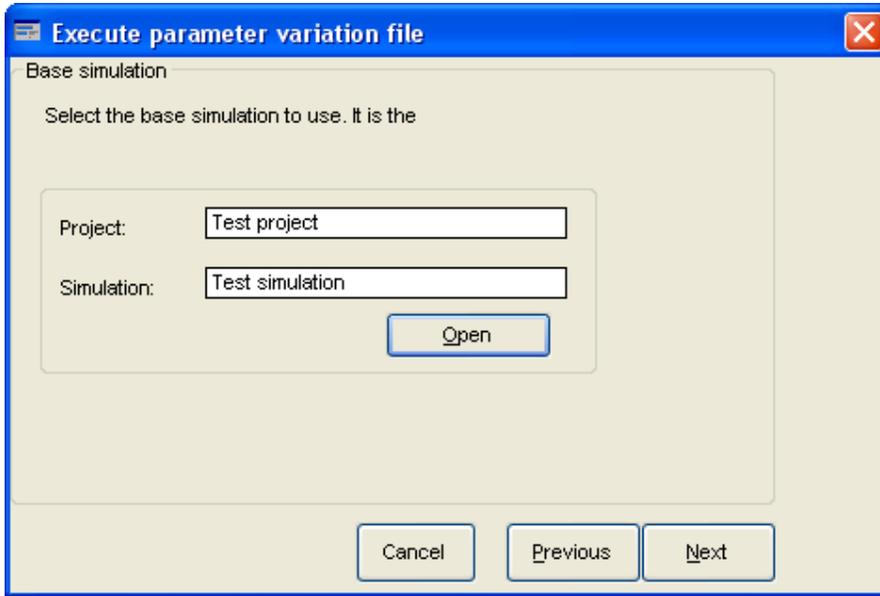
The next step is to run the Monte Carlo simulations. Click on “Tools-> Monte Carlo-> Execute variation file”. Using this wizard, you can execute a large number of parameter combinations in batch mode. Two file formats are supported:

- 1.) UNCSAM \*.fam file (Janssen et al., 1992) created with the in-built sampling tool (see above).
- 2.) MACRO default. This is a text file (\*.txt) where the first line contains a tab-separated list of the names of the parameters to vary, and the subsequent lines contain tab-separated parameter values. This file can either be created manually, or by an external program created by the user.

The parameter variation file defines a number of simulations (one for each row in the file) for which selected parameter values are varied. In addition to executing the defined simulations, the tool also creates a file that can be used for analysis of the outputs in UNCSAM (\*.sgn file).

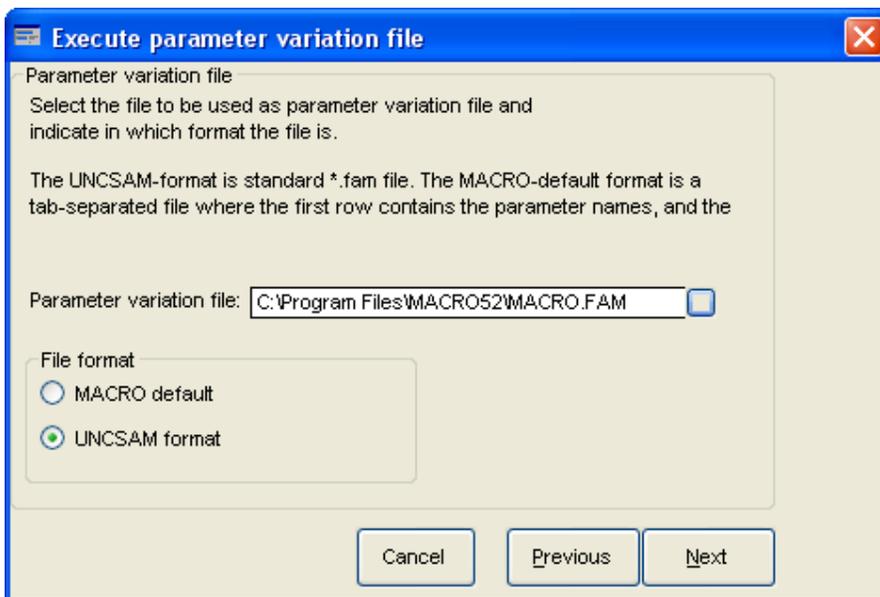
##### Step I

First, a base-simulation must be selected that defines the values of those parameters that are kept constant.



### Step II

The parameter variation file defining the simulations is then selected.



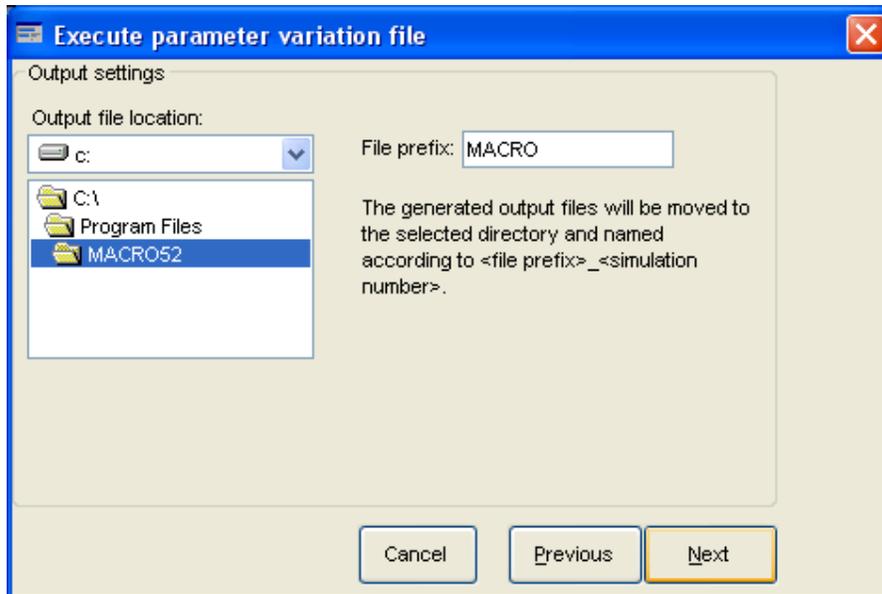
In order for the program to identify the correct parameter, the specified parameter names must have the following format: <MACRO parameter name>[\_<horizon number>]. If <horizon number> is specified as 'all', the parameter will be varied for all horizons (if this is possible). This applies to both file formats.

### Step III

The following output settings must be specified:

- 1.) the directory where the output files (\*.bin, \*.sum and \*.sgn) are stored.
- 2.) the output filename prefix.

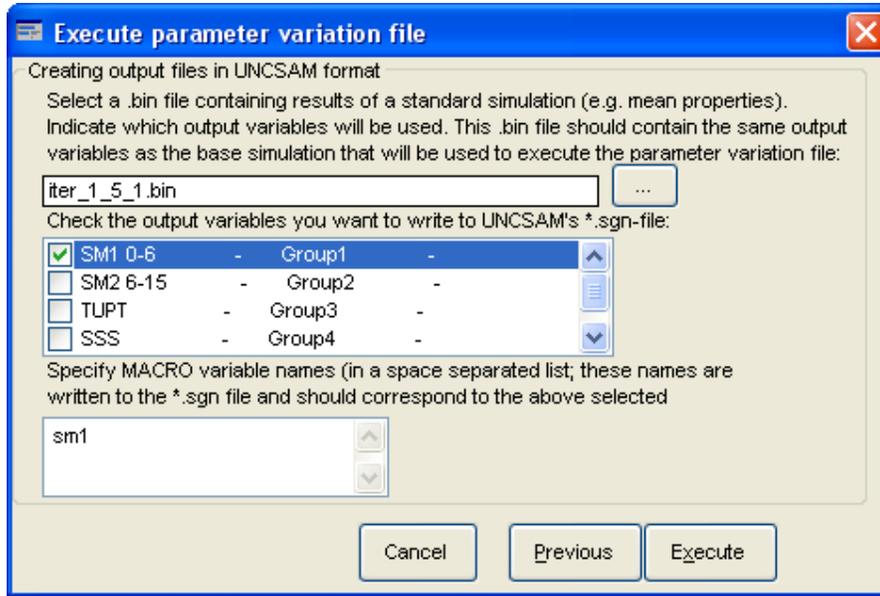
The output files will be moved to the specified directory and renamed according to “<filename prefix>\_<simulation number>”. The generated UNCSAM \*.sgn file will be named “<filename prefix>.sgn”.



#### Step IV

Step IV involves settings that define the UNCSAM input file, with extension \*.sgn. The output variables that you want to include in the sensitivity analysis are selected in the listbox. For each of the selected outputs, the parameter name that will be written to the \*.sgn file must also be specified. The result will be a space-separated list, with the same number of entries as the number of selected outputs.

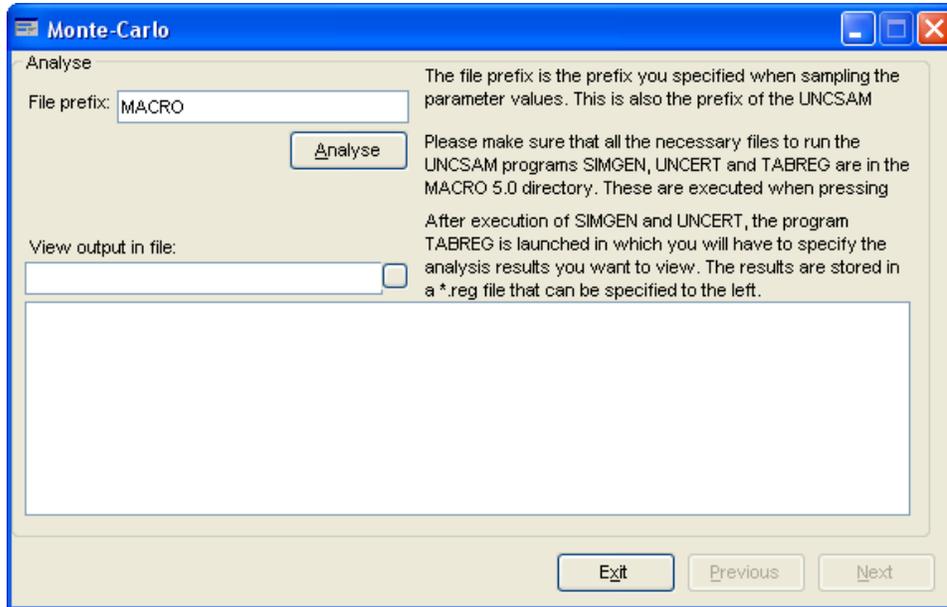
The selected output variables are read from the MACRO output file after the termination of each simulation, and an entry is written to the UNCSAM \*.sgn file. If the selected output is not an accumulated variable, the value of the variable for the last time-step of the simulation will be used. Note that this step must be carried out even if you do not intend to use UNCSAM for further analysis.



When you click on the button “Execute”, the simulations will be run in the order in which they are defined in the parameter variation file. After each simulation, the output files are moved to the specified directory and an entry is written to the \*.sgn file. If an error should occur during the simulation, so that no simulation results are available, an entry will be written to the file ‘parvar.log’ (in the program directory) with the error message, and the entry in the \*.sgn file will be set to ‘-99’ for each of the selected variables, to indicate missing values. If an error should occur when trying to access the simulation results in the \*.bin file, an entry describing the error will be written to the file ‘sgn.log’ (in the program directory).

### 7.4.3 Analysing the results of the Monte Carlo simulations

In order to access the analysis tool in UNCSAM, click on “Analyze” under “Tools-> Monte Carlo”. A file prefix needs to be specified. This is the same prefix specified when sampling the parameter values, which should also be the prefix of the UNCSAM \*.sgn file. Make sure that the UNCSAM programs SIMGEN, UNCERT and TABREG are in the MACRO 5.2 directory. These programs are normally placed in the program directory during installation and are executed when you click on ‘Analyze’. The \*.sgn (simulation results) and \*.sam (sampled parameter values) files must also be present in the program directory.



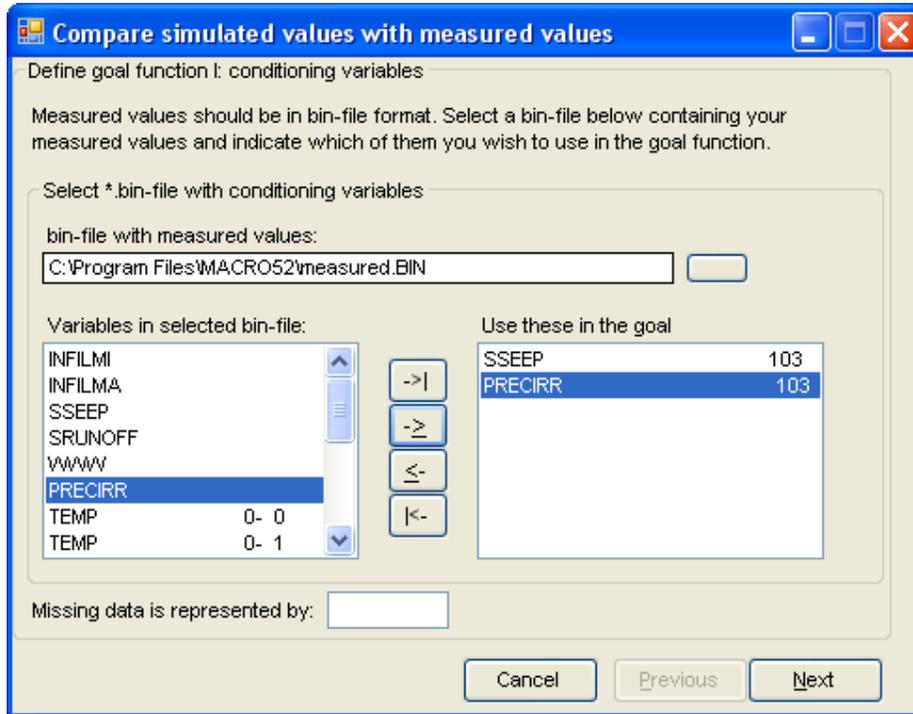
After execution of SIMGEN and UNCERT, the program TABREG is launched that requires some input from the user in order to determine what results to display.

The results are stored in a \*.reg file that can be viewed from within MACRO 5.22.

Note that the UNCSAM software package includes additional programs for analysis and presentation of the results of a Monte Carlo analysis. These can be run manually outside the MACRO shell.

## 7.5 Compute goal function

You can make a quick statistical comparison of the simulated results with measured data by clicking on “*Tools->Compute goal function*”. By clicking “*Compute goal function*”, you can calculate the goodness of fit of simulated variables with measured data, using either the root mean square error (RMSE) or model efficiency (EF) criteria.



## 7.6 The PG graphics program

The \*.bin file can be read directly using the auxiliary program PG that is supplied with MACRO 5.2. You first click twice on pg.exe to start the program, and then press the F2 key to list all the available \*.bin files. Use the up and down arrow keys to move on top of the file you wish to read in, and then press return twice. A list containing useful commands in PG is given in Appendix 2 to this report. It should be noted that PG can only handle a maximum of 400 output variables.

## 8 References

Abbaspour, K.C., van Genuchten, M.T., Schulin, R. & Schläppi, E., 1997. A sequential uncertainty domain inverse procedure for estimating subsurface flow and transport parameters. *Water Resources Research*, 33: 1879-1892.

Güntner, A., Olsson, J., Calver, A. & Gannon, B. 2001. Cascade-based disaggregation of continuous rainfall time series: the influence of climate. *Hydrology and Earth System Sciences*, 5, 145-164.

Hollis, J.M. & Woods, S.M. 1989. The measurement and estimation of saturated soil hydraulic conductivity. SSLRC Research Report, Cranfield Univ., U.K., 19 pp.

Janssen, P.H.M., P.S.C. Heuberger & R. Sanders. 1992. UNCSAM 1.1: a software package for sensitivity and uncertainty analysis. Report nr. 959101004, National Institute of Public Health and Environmental Protection (RIVM), Bilthoven, Netherlands.

Jarvis, N.J., Hollis, J.M., Nicholls, P.H., Mayr, T. & Evans, S.P. 1997. MACRO\_DB: a decision-support tool to assess the fate and mobility of pesticides in soils. *Environmental Modelling & Software*, 12, 251-265.

Jarvis, N.J., Zavattaro, L., Rajkai, K. Reynolds, W.D. Olsen, P-A., McGechan, M., Mecke, M., Mohanty, B., Leeds-Harrison, P.B. & Jacques, D. 2002. Indirect estimation of near-saturated hydraulic conductivity from readily available soil information. *Geoderma*, 108, 1-17.

Larsbo, M. & Jarvis, N.J. 2003. MACRO 5.2. A model of water flow and solute transport in macroporous soil. Technical description. *Studies in the Biogeophysical Environment, Emergo 2003:6*, Department of Soil Sciences, SLU, Uppsala, Sweden, 48 pp.

Olsson, J. 1998. Evaluation of a scaling cascade model for temporal rainfall disaggregation. *Hydrology and Earth System Sciences*, 2, 19-30.

Stenemo, F. & Jarvis, N.J. 2002. Guidance document and manual for the use of MACRO\_DB v.2.0. Unpublished report (<http://www.mv.slu.se/bgf/defeng.htm>).

Wösten, J.H.M., Lilly, A., Nemes, A., Le Bas, C. 1998. Using existing soil data to derive hydraulic parameters for simulation models in environmental studies and in land use planning. Report 156, Winand Staring Centre, SC-DLO, Wageningen, Netherlands, 106 pp.

## Appendix 1: Pedotransfer functions

### MACRO 5.0/ 5.1 pedotransfer functions

The saturated water content and the shape parameters ('ALPHA' and 'N') of the van Genuchten soil water retention curve are estimated using the pedotransfer functions developed for European soils by Wösten et al. (1998, Table 11, page 73). The pressure potential defining the boundary between micropores and macropores ('CTEN') is fixed at -10 cm. The water contents at pressure potentials of -10 cm ('XMPOR', saturated micropore water content) and -15000 cm ('WILT', wilting point) are calculated using the calculated values of the water retention parameters.

The effective diffusion pathlength ('ASCALE') is estimated from the FAO soil structural description using the method described in Jarvis et al. (1997, Table 8).

The saturated micropore hydraulic conductivity (i.e. 'KSM', the conductivity at a pressure potential of -10 cm) is calculated as a function of the geometric mean particle size (Jarvis et al., 2002, equation 6 in Table 2).

The total saturated hydraulic conductivity ('KSATMIN' in mm/h) is calculated from the texture, soil effective porosity and FAO structural development using a modified version of the algorithm proposed by Hollis and Woods (1989). This algorithm is also included in version 2 of MACRO\_DB (Stenemo and Jarvis, 2002). The code for this algorithm, which in this modified form is completely untested, is presented on the following page.

Algorithm used to calculate saturated hydraulic conductivity

ac = Saturated water content minus water content at a pressure head of -50 cm (%)  
clay = clay content (%)  
silt = silt content (%)  
s1 = strength of structural development (FAO)

```
If clay < 16 And silt + (clay * 2) < 31 Then
If ac < 7.5 Then
ksatmin = (0.4535 * ac ^ 1.03423) * (10 / 24#) Else
ksatmin = (8.03578 - (6.7707 * ac) + (0.833 * ac ^ 2)) * (10 / 24#) End If
Else
If s1 = "none" Then xfactor = 1
ElseIf s1 = "weak" Then
If ac > 10 Then xfactor = 10 ^ 0.5
Else
xfactor = 10 ^ (0.5 + (0.03 * (10 - ac))) End If
ElseIf s1 = "moderate" Then
If ac > 10 Then xfactor = 10 ^ 0.7
Else
xfactor = 10 ^ (0.7 + (0.05 * (10 - ac))) End If
```

```

Else
If ac > 10 Then xfactor = 10
Else
xfactor = 10 ^ (1 + (0.1 * (10 - ac))) End If
End If

If ac < 4# Then
ksatmin = (((0.14143 * Exp(0.46944 * ac)) * (10 / 24#)) * xfactor
Else
ksatmin(i) = ((5.8521 - (5.4125 * ac) + (1.05138 * ac ^ 2)) * (10 / 24#)) * xfactor
End If

```

## FOOTPRINT pedotransfer functions

MACRO parameters are given in capital letters.

The saturated water content and the shape parameters ('ALPHA' and 'N') of the van Genuchten soil water retention curve are estimated using the pedotransfer functions developed for European soils by Wösten et al. (1998, Table 11, page 73). The pressure potential defining the boundary between micropores and macropores ('CTEN') is fixed at -10 cm. The water contents at pressure potentials of -10 cm ('XMPOR', saturated micropore water content) and -15000 cm ('WILT', wilting point) are calculated using the calculated values of the water retention parameters.

The saturated hydraulic conductivity of the micropores, KSM (mm/h) is given by:

$$KSM = c \times XMPOR \times n^l$$

where  $c = 0.186$ ,  $l = 10.73$

The saturated hydraulic conductivity, KSATMIN (mm/h) is given by:

$$KSATMIN = \frac{6000 \times \frac{a}{100}}{ZN} \text{ where } a \text{ is the macroporosity}$$

The macroporosity is determined using the following function:

```

Public Function macroporosity(ByRef seq As Short, ByRef text_class As
String, ByRef desig As String, ByRef midpoint As Single, ByVal frm As
frmFOOTPRINTPedotransferInput) As Single
    If VB.Left(desig, 1) = "A" Then 'topsoil
        If sw_landuse = Switch.PerennialLanduse Or
frm.chkZeroTillOrManure.Checked Or desig = "A" Then 'perennial crops
            Return 5 'undisturbed
        ElseIf VB.Left(desig, 2) = "At" Then 'intensely tilled
            Return 5
        ElseIf VB.Left(desig, 2) = "Ap" Then 'Ap
            If text_class = "fine" Then
                Return 3
            ElseIf text_class = "coarse" Then
                Return 5
            Else
                Return 4
            End If

```

```

ElseIf VB.Left(desig, 1) = "A" Then
    If text_class = "fine" Then
        Return 2
    ElseIf text_class = "coarse" Then
        Return 4
    Else
        Return 3
    End If
ElseIf VB.Left(desig, 1) = "O" Or VB.Left(desig, 1) = "H" Then
    Return 5
Else
    Return 5
End If
'subsoil
Else
    If desig = "BC" Then
        If text_class = "fine" Then
            Return 0.2
        ElseIf text_class = "coarse" Then
            Return 4
        Else
            Return 0.4
        End If
    ElseIf (VB.Left(desig, 1) = "B" Or VB.Left(desig, 1) = "E")
And midpoint <= 50 Then
        If text_class = "fine" Then
            Return 1.6
        ElseIf text_class = "coarse" Then
            Return 5
        Else
            Return 1.6
        End If
    ElseIf (VB.Left(desig, 1) = "B" Or VB.Left(desig, 1) = "E")
And midpoint > 50 Then
        If text_class = "fine" Then
            Return 0.8
        ElseIf text_class = "coarse" Then
            Return 5
        Else
            Return 0.8
        End If
    ElseIf VB.Left(desig, 1) = "C" Then
        If text_class = "fine" Then
            Return 0.2
        ElseIf text_class = "coarse" Then
            Return 3
        Else
            Return 0.4
        End If
    ElseIf VB.Left(desig, 1) = "O" Or VB.Left(desig, 1) = "H" Then
        Return 5
    ElseIf desig = "Bpan" Then
        Return 0.1
    End If
End If
Return 0
End Function

```

The total porosity, TPORV (-), is then given as XMPOR + macroporosity.

The parameters ASCALE and ZN are determined by the following functions:

```
Public Function fp_ASCALE(ByRef structclass As Short) As Single
  If structclass = 1 Then
    fp_ASCALE = 1
  ElseIf structclass = 2 Then
    fp_ASCALE = 15
  ElseIf structclass = 3 Then
    fp_ASCALE = 50
  ElseIf structclass = 4 Then
    fp_ASCALE = 150
  End If
End Function
```

```
Public Function fp_ZN(ByRef StructureClass As Short) As Single
  If StructureClass = 1 Then
    fp_ZN = 6
  ElseIf StructureClass = 2 Then
    fp_ZN = 4
  ElseIf StructureClass = 3 Then
    fp_ZN = 3
  ElseIf StructureClass = 4 Then
    fp_ZN = 2
  End If
End Function
```

The parameter “structureclass” is determined by following the classification schemes described by Jarvis et al (1999).<sup>1</sup>

---

<sup>1</sup> Jarvis et al (2009). A Conceptual Model of Soil Susceptibility to macropore flow. *Vadoze Zone Journal* 8:902-901, 2009.

## Appendix 2: Useful commands in PG

L (= List)

L1 List the variable descriptions (with max. and min.values) L3 List the variable values for each output time step

P (= Plot)

P2 Time series

P5 Depth profile

C (= Create new files)

C1 Perform calculations and transformations

C6 Combine two files with different variables (i.e. simulation results and data) C7  
Combine two files with the same variables but different periods

G (= General utilities)

G4 Read in new .BIN file

G7 General settings (i.e. define plotting period)

E (= exit from program)

E2 log file deleted