

MFIT v1.0.0 – User Guide

Jacques Bodin, 2019

1. Introduction

MFIT is a Windows-based computer software package for the analytical modeling of concentration breakthrough curves (BTCs) from tracer experiments in fractured and karst aquifers. The software integrates four transport models built on a multiflow approach that assumes that the spatial heterogeneity of the aquifer can be approximated by a combination of independent one-dimensional channels. The four transport models are referred to as MDMi (multi-dispersion model, instantaneous injection), MDMed (multi-dispersion model, exponentially decaying injection), MDP-SFDM (multi-double porosity, single-fracture dispersion model), and MDP-2RNE (multi-double porosity, two-region nonequilibrium model). The conceptual and mathematical descriptions of these models are given in the companion article to this manual and are not represented here. The MDMi, MDMed, MDP-SFDM and MDP-2RNE models are implemented as independent codes that can be run either as console applications or using the MFIT program as a graphical user interface (GUI). MFIT provides a GUI for (i) importation and graphic visualization of user-provided BTC data, (ii) parameterization, direct running, and graphical output of the analytical transport models, (iii) inversion (automatic calibration) of model parameters for optimal curve fitting, and (iv) assessment of the uncertainty of calibrated parameter values. Both the optimization and uncertainty analysis of model parameters are carried out using routines from the PEST software package (<http://www.pesthomepage.org/>).

2. Direct (forward) transport simulations

2.1. Using the MDMi.exe, MDMed.exe, MDP_SFDM.exe, and MDP_2RNE.exe programs as console applications

The model input parameters are read from a file named « input.txt », which can be built using any text editor program. The format of the input file differs slightly depending on the model (see below).

Structure of the input parameter file (input.txt) for the MDMi.exe program

```
Lower limit of the simulation time range (Tmin), real number
Upper limit of the simulation time range (Tmax), real number
Number of time-concentration BTC points simulated by the model, integer
Total system flow rate (Q), real number
Number of channels, integer

Solute mass flowing through the first channel, real number
Mean transit time (T0) in the first channel, real number
Peclet number (Pe) of the first channel, real number

Solute mass flowing through the second channel, real number
Mean transit time (T0) in the second channel, real number
Peclet number (Pe) of the second channel, real number

etc.
```

Structure of the input parameter file (input.txt) for the MDMed.exe program

```
Lower limit of the simulation time range (Tmin), real number
Upper limit of the simulation time range (Tmax), real number
Number of time-concentration BTC points simulated by the model, integer
```

Initial/maximum injection concentration at the inflow boundary of the system (C_0), real number
 Number of channels, integer

Flowrate contribution ratio of the first channel, real number
 Mean transit time (T_0) in the first channel, real number
 Peclet number (Pe) of the first channel, real number
 Gamma coefficient of the first channel, real number

Flowrate contribution ratio of the first channel, real number
 Mean transit time (T_0) in the second channel, real number
 Peclet number (Pe) of the second channel, real number
 Gamma coefficient of the second channel, real number

etc.

Structure of the input parameter file (input.txt) for the MDP_SFDM.exe program

Lower limit of the simulation time range (T_{min}), real number
 Upper limit of the simulation time range (T_{max}), real number
 Number of time-concentration BTC points simulated by the model, integer
 Total system flow rate (Q), real number
 Number of channels, integer

Solute mass flowing through the first channel, real number
 Diffusion parameter (beta coefficient) of the first channel, real number
 Mean transit time (T_0) in the first channel, real number
 Peclet number (Pe) of the first channel, real number

Solute mass flowing through the second channel, real number
 Diffusion parameter (beta coefficient) of the second channel, real number
 Mean transit time (T_0) in the second channel, real number
 Peclet number (Pe) of the second channel, real number

etc.

Structure of the input parameter file (input.txt) for the MDP_2RNE.exe program

Lower limit of the simulation time range (T_{min}), real number
 Upper limit of the simulation time range (T_{max}), real number
 Number of time-concentration BTC points simulated by the model, integer
 Total system flow rate (Q), real number
 Number of channels, integer

Solute mass flowing through the first channel, real number
 Length of the first channel, real number
 Mean transit time (T_0) in the first channel, real number
 Peclet number (Pe) of the first channel, real number
 Psi coefficient of the first channel, real number
 Omega coefficient of the first channel, real number

Solute mass flowing through the second channel, real number
 Length of the second channel, real number
 Mean transit time (T_0) in the second channel, real number
 Peclet number (Pe) of the second channel, real number
 Psi coefficient of the second channel, real number
 Omega coefficient of the second channel, real number

etc.

By running the transport model code, a series of time-concentration values are computed and saved in a text file named « output.txt ». Care must be taken to use consistent units for all the variables in the input file, and these units also determine the time and concentration units of the output file (e.g., concentrations are in g/m^3 if the system flowrate is expressed in m^3/s , the solute mass in g, and the

mean transit times in s). Fig. 1 shows an example of an input file for a two-channel MDMi model. The related output file written by the MDMi program is shown in Fig. 2.

```

1 0
2 40
3 200
4 60.5
5 2
6
7 0.6
8 8.5
9 85
10
11 3.2
12 14.5
13 15.5

```

Fig. 1. Example of an input file for a two-channel MDMi model.

```

1 TIME VALUE ; CONCENTRATION VALUE
2 0;0
3 0.201005025125628;2.17632859649262e-118
4 0.402010050251256;3.65283057249872e-58
5 0.603015075376884;3.22421034980317e-38
6 0.804020100502513;2.59606856525294e-28
7 1.00502512562814;2.06823604182334e-22
8 1.20603015075377;1.66016490802163e-18
9 1.4070351758794;9.70214532407744e-16
10 1.60804020100503;1.10764392181961e-13
11 1.80904522613065;4.27019218081908e-12
12 2.01005025125628;7.71553821621159e-11
13 2.21105527638191;8.0460107467616e-10
14 2.41206030150754;5.56240301922299e-09
15 2.61306532663317;2.80538617581346e-08

```

Fig. 2. Output file written by the MDMi program from the input file shown in Fig. 1. Only the first lines of the output file (which contains 200 time-concentration lines) are displayed.

2.2. Using MFIT as a GUI for the MDMi.exe, MDMed.exe, MDP_SFDM.exe, and MDP_2RNE.exe programs

1. Select **File | New** from the main menu.
2. Select **File | Working Directory**, and pick a folder for the writing of the working (temporary) files and model output files.
3. Select **Transport models**, choose one of the four proposed transport models, and then select **Parameters**. A dialog will be displayed that allows you to set the simulation parameters. The example shown in Fig. 3 is similar to the two-channel MDMi model input.txt file shown in Fig. 1. The dialog can also be filled by importing a pre-existing input.txt file (or any other text file conforming to the same structure) using the Import button. When finished, click the OK button.

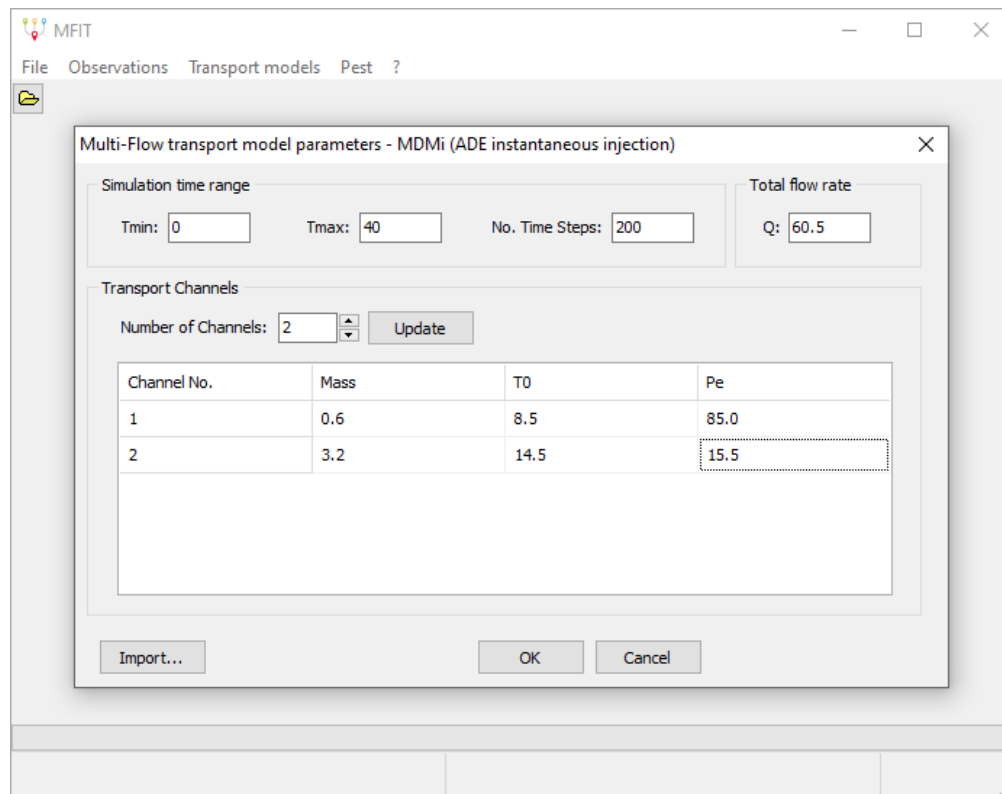


Fig. 3. GUI dialog for setting the MDMi model parameters.

4. Select **Transport models | Compute BTC**. The simulated BTC will be displayed (in blue, see Fig. 4). By clicking on the bottom- or left-axis black line, a dialog appears that allows changing the X- and Y-axis ranges.

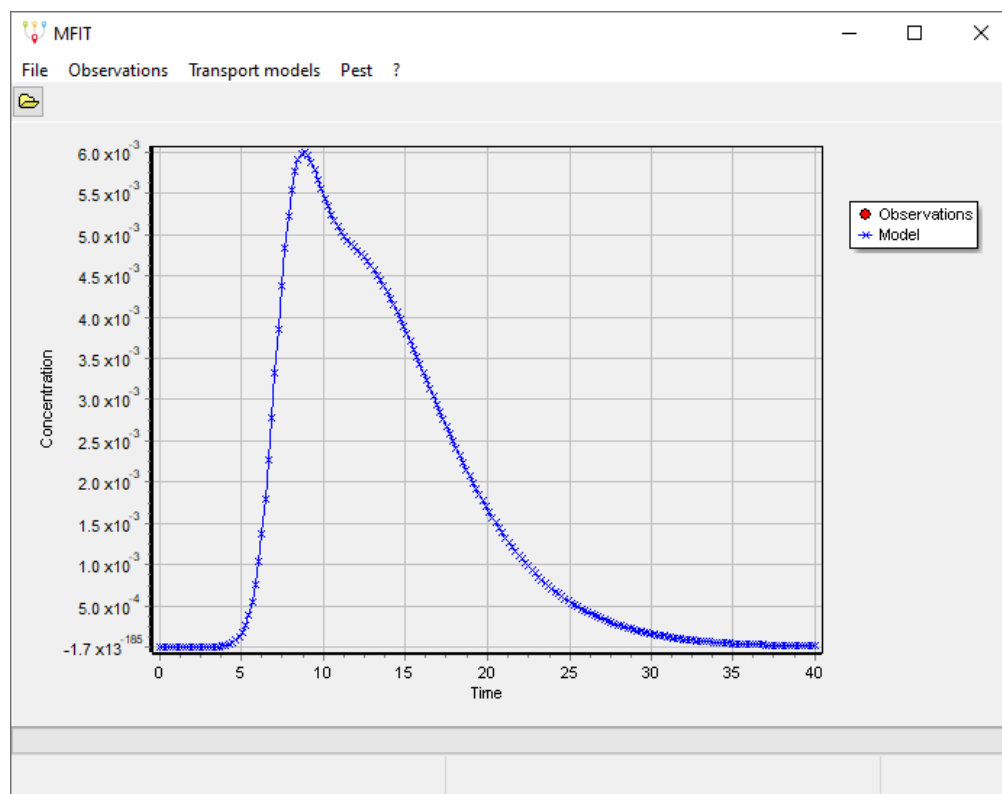


Fig. 4. Example of an MFIT-simulated BTC (two-channel MDMi model).

5. Optionally, select **File | Save As** to save the simulation parameters and the path of the working directory into an MFIT file (extension: .mfi). MFIT files can be edited with any text editor and can be reopened using the **File | Open** menu command.

3. Importing BTC data

1. Select **Observations** from the main menu, and fill the Time-Concentration-Weight table dialog. The « weight » value that must be assigned to each line is only relevant if PEST will be used for the automatic calibration of an MFIT transport model against the user-provided BTC data (see section 5.1 below). If not planning to use PEST, just apply any constant weight value to all the lines. The BTC data (including weights) can either be filled manually (use « Insert Row » or « Add Rows » for adding as many rows as needed), copied and pasted from a spreadsheet (e.g., Excel), or imported from a CSV text file (separator: comma).
2. Large BTC datasets can be resampled by applying a filter routine accessible from the « Filter » button. A general rule in BTC data analysis is to seek the best compromise between simplification and preservation of the information content of the observed BTC, especially in the case of multiple local concentration peaks. A number of time-concentration BTC points between 50 and 200 is appropriate in most cases.
3. Click on the OK button to display (in red) the observed BTC.
4. Optionally, select **File | Save As** to save the Time-Concentration-Weight table into an MFIT file (extension: .mfi). MFIT files can be edited with any text editor and can be reopened using the **File | Open** menu command.

4. Model curve fitting of the observed BTC

4.1. Manual trial-and-error procedure

The procedure described above in section 2.2 can be used for simulating a model BTC that will be plotted in the same graph as the observed BTC. Better agreement between the model and observed BTCs can be sought from a trial-and-error procedure that consists of changing the model parameters, again running the model, and evaluating (either by « eye » or by computing some error criteria between the simulated and observed BTCs) if the new set of model parameters is better (or worse) than the preceding one.

4.2. PEST model inversion

PEST is a model-independent optimizer that can be used to seek the set of model parameters that allows the « best fit » between the simulated and observed BTCs. As detailed in the PEST documentation (<http://www.pesthomepage.org/Downloads.php>), a number of settings may influence the way PEST operates. A decision was made to narrow the MFIT-PEST GUI dialog to the most important settings to maintain a simple (i.e., user-friendly) but effective program interface. Other PEST control variables were fixed in the MFIT code to the expected optimum settings based on recommendations from the PEST user manual and/or experience knowledge. A modified version of MFIT including additional PEST GUI dialogs can be made available upon request.

The objective function that is minimized by PEST may be expressed as a « measurement objective function » only or as the sum of a « measurement objective function » and a « regularization objective

function ». The measurement objective function is defined as the sum of the squared weighted residuals between the observed and simulated $c(t)$ values. The weight value assigned to each time-concentration line in the observation table (see section 3) has a direct influence on the computation of the measurement objective function and therefore on the inversion results. If the same weight value is set for each line, PEST will pay the same attention to each concentration value and will search for the best overall model curve fitting of the tracer BTC. If you prefer to promote the model fit on a specific part of the tracer BTC (e.g., the concentration peak), use higher relative weight values for the related lines (or lower relative weight values for the other lines). The regularization objective function acts as a penalty function for deviations from some preferred parameter conditions. Two options have been implemented in MFIT. The first option, referred to as « preferred homogeneity », promotes a solution of minimum variance for the model parameters pertaining to the different channels. In the second option, referred to as « preferred value », the inversion seeks the solution that is the closest to some prior estimates of the model parameters. The steps for optimizing an MFIT transport model are as follows:

1. Import the BTC data (see section 3).
2. Choose and run one of the four MFIT transport models, using trial-and-error as a first calibration method (see section 2.2).
3. Select **Pest | Options** from the main menu. A dialog will be displayed that allows setting the PEST options (see Fig. 5).

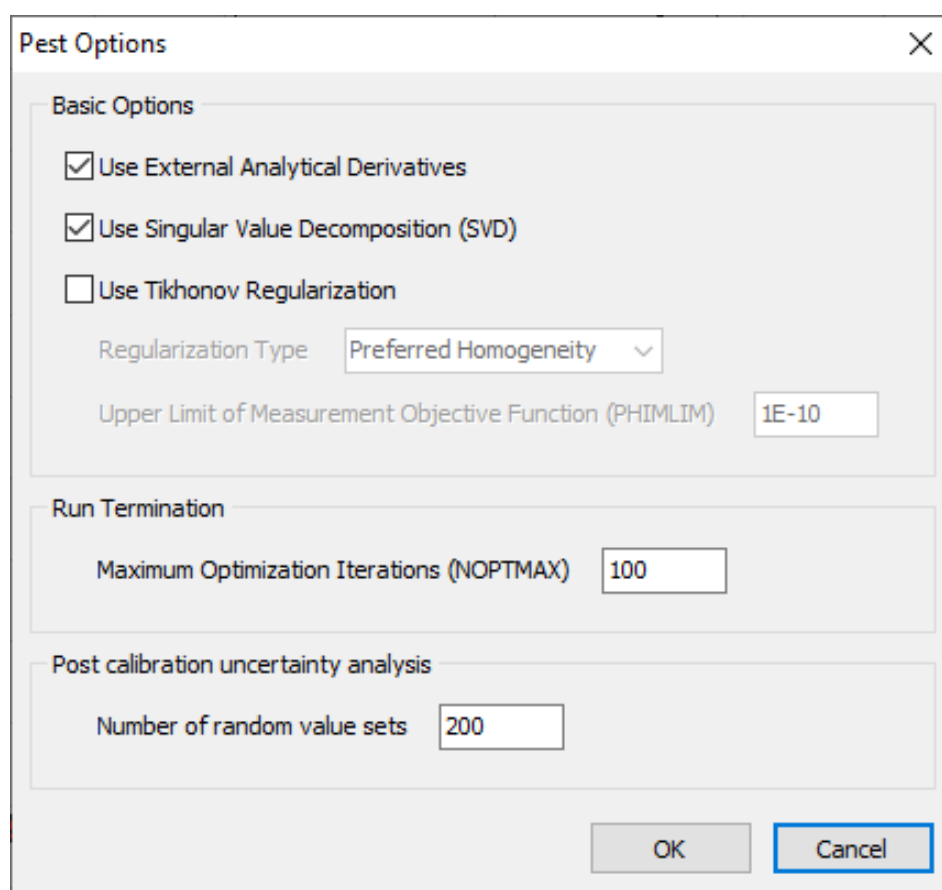


Fig. 5. GUI dialog of Pest options.

The first and second checkboxes should remain checked in most cases. The first checkbox is used to enable or disable the analytical computation of the Jacobian matrix derivatives. Unchecking this checkbox will force PEST to compute all the derivatives using finite differences, which is both time-

consuming and less accurate. The second checkbox activates (or deactivates) the use of the singular value decomposition (SVD) method for solving the inverse problem. Using SVD is recommended, as it guarantees the numerical stability of the inversion process.

The third checkbox allows adding some « preferred homogeneity » or « preferred value » Tikhonov regularization constraints to the inverse problem. If the second option is selected, PEST will use the parameter values specified in the model dialog (Fig. 3) as « preferred values ». As discussed below, two options are also available for the number of channels to be considered in the inversion procedure. The first option, referred to as « User-specified channels », considers a fixed number of channels as specified in the model dialog. For the second option, referred to as « Automatic multiple channels », a series of automatic tracer BTC fittings are performed for a decreasing number of channels ranging from N_{\max} (upper bound) to 1. If both the « preferred value » Tikhonov regularization and « Automatic multiple channels » options are selected and if several channels are specified in the model dialog, only the parameters pertaining to the first channel are used as preferred values and are applied to all the channels during the optimization.

The variable PHIMLIM controls the tradeoff between the model curve-fitting of the observed BTC and the regularization constraints, i.e., adherence to preferred homogeneity or preferred value conditions. As discussed in the PEST documentation, a suitable value for PHIMLIM can be determined by first running PEST without Tikhonov regularization, determining the measurement objective function value achieved by PEST, and then setting PHIMLIM 5 to 10 percent higher than this value.

The NOPTMAX variable sets the maximum allowed number of optimization iterations for each PEST run. A minimum value of 100 is recommended for this variable to ensure that PEST terminates its execution because a convergence criterion has been reached rather than because of the NOPTMAX limit.

The last edit control in the PEST option window dialog allows setting the number of parameter sets that will be used for assessing the postcalibration parameter uncertainty, if requested by the user after completion of a « User-specified channels » inversion (see section 4.3 below). This number should be set as high as possible within computational time constraints.

When the PEST options have been configured, click the OK button.

4. Select **Pest | Parameters** from the main menu, and then select either **User-Specified Channels** or **Automatic Multiple Channels**. Different dialogs (Figs. 6 and 7) will be displayed depending on the selected option and selected MFIT model. Each of these dialogs allows specifying the parameters to be estimated by PEST, along with their lower and upper bounds. The lower and upper parameter bounds must reflect as accurately as possible the range of prior parameter uncertainty, as these bounds are used both in the inversion process (internal parameter scaling « boundscale » option; see the PEST manual) and for the assessment of postcalibration parameter uncertainty (see section 4.3 below).

PEST Parameters - User Specified Channels

Pest Parameter No.	Type	Channel No.	Estimate (No=0/Yes=1)	Minimum	Maximum
1	Total Flowrate	All	0	1.0E-10	6.0E+01
2	Mass	1	1	1.0E-10	5.0E+00
3	T0	1	1	1.0E+00	4.0E+01
4	Pe	1	1	1.0E+00	1.0E+05
5	Mass	2	1	1.0E-10	5.0E+00
6	T0	2	1	1.0E+00	4.0E+01
7	Pe	2	1	1.0E+00	1.0E+05
8	None	1	0	1.0E-10	1.0E+10
9	Total Flowrate	1	0	1.0E-10	1.0E+10

Import... OK Cancel

Fig. 6. GUI dialog for the selection and setting of model parameters to be optimized with the « User-Specified Channels » option.

PEST Parameters - Automatic Multiple Channels

Parameter	Estimate (No=0/Yes=1)	Minimum	Maximum
Total Flowrate	0	1.0E-10	6.0E+01
Mass	1	1.0E-10	5.0E+00
T0	1	1.0E+00	4.0E+01
Pe	1	1.0E+00	1.0E+05

Maximum number of channels: 6 ☒ Enable Multiple Initial Condition Preconditioning

OK Cancel

Fig. 7. GUI dialog for the selection and setting of model parameters to be optimized with the « Automatic Multiple Channels » option.

The « Import » button in the lower-left corner of the User-Specified Channels PEST parameter dialog (Fig. 6) allows importing a CSV text file (separator: comma; see example Fig. 8) for implementing the PEST parameter table.

PEST_UserChanParam_example_file.csv

```

1 Total Flowrate,All,0,1.0E-10,60
2 Mass,1,1,1.0E-10,5
3 Mass,2,1,1.0E-10,5
4 T0,1,1,1,40
5 T0,2,1,1,40
6 Pe,1,1,1,1.0E+5
7 Pe,2,1,1,1.0E+5

```

Normal text file length: 131 lines: 7 Ln: 1 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS

Fig. 8. Example of a CSV text file that can be imported into the PEST parameter table shown in Fig. 6.

While the « User-Specified Channels » option considers a fixed number of channels as specified in the model dialog (Fig. 3), the selection of the « Automatic Multiple Channels » option leads to a series of PEST model calibrations using a decreasing number of channels ranging from N_{\max} (upper bound) to 1. The maximum number of channels, N_{\max} , can be specified using the edit control located below the PEST parameter table (see Fig. 7). The checkbox to the right of the N_{\max} edit

control activates (or deactivates) the use of a multistart optimization method that consists of multiplying the PEST runs using different sets of initial parameter values (see the companion article to this manual for related discussion). Despite the additional computational cost of this method, its use is recommended, as it significantly reduces the risk for PEST to be trapped in a local minimum of the parameter space during the optimization process.

When the settings of the PEST parameters are complete, click the OK button to validate.

5. Select **Pest | Create Datasets** from the main menu. Three PEST input files named MFIT.tpl, MFIT.ins, and MFIT.pst are written in the working directory. Each of these files can be edited with any text editor (see the PEST manual for full descriptions of the file formats and contents).
6. Optionally, select **Pest | Run PestChk** to check errors and/or inconsistencies in the PEST input dataset. This menu launches the PESTCHEK utility, and the resulting outputs are displayed in a text window. Ideally, the PESTCHEK report should be identical to Fig. 9, i.e., no error and a single warning reminding that the parameter bounds specified in the PEST parameter dialog should reflect as accurately as possible the range of prior parameter uncertainty. Any listed error (e.g., initial model parameter value outside the specified range) must be fixed prior to running PEST.

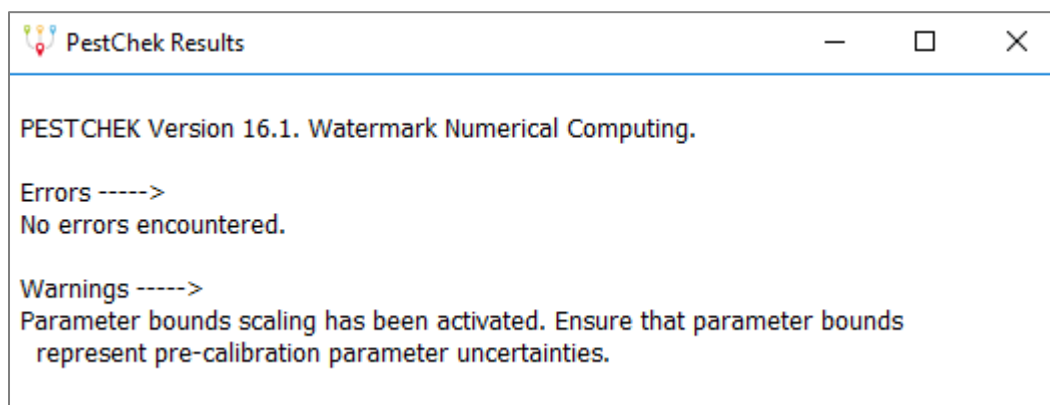


Fig. 9. PESTCHEK utility report.

7. Select **Pest | Run Pest** to start the inversion process. Depending on the problem settings (selected transport model, number of points in the simulated BTC, number of model parameters to estimate, User-Specified Channels / Automatic Multiple Channels / Multi-Start options, etc.), the computation time can range from a few seconds to several hours. Lengthy computations can be stopped at any time by selecting the **Pest | Abort optimization process** menu entry, or equivalently by clicking on the « Abort » shortcut button that is only visible during the optimization runs.

Once the model inversion is complete, the graphical output of MFIT depends on whether a « User-Specified Channels » inversion or an « Automatic Multiple Channels » inversion was performed. In the former case, MFIT will display the optimized model curve and a message box prompting whether the user wishes to import the related parameter values into the model parameter table (Fig. 10). In the second case, a graph is displayed on the right of the BTC plots and shows the minimum measurement objective function (PHI) value achieved by PEST for each channel-number scenario (Fig. 11). The numerical values in this graph are automatically saved in a text file named « Phi(N).txt » in the working directory. The up-down edit control located below allows navigation between the channel-number solutions, while the corresponding simulated BTC is displayed in the BTC graph. Clicking on the OK button will import the optimized parameters of the selected channel-number solution into the model parameter table.

8. Optionally, select **File | Save As** to save the optimized model into an MFIT file.

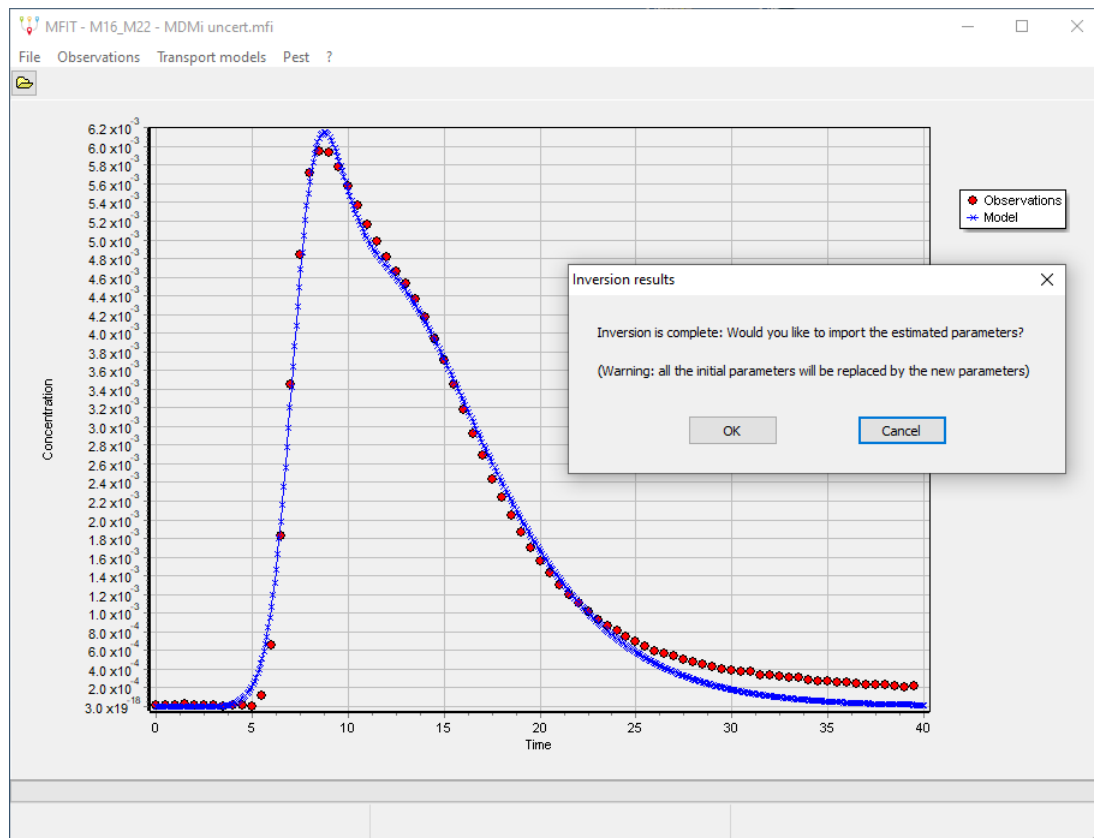


Fig. 10. Graphical display at the end of a User-Specified Channels inversion.

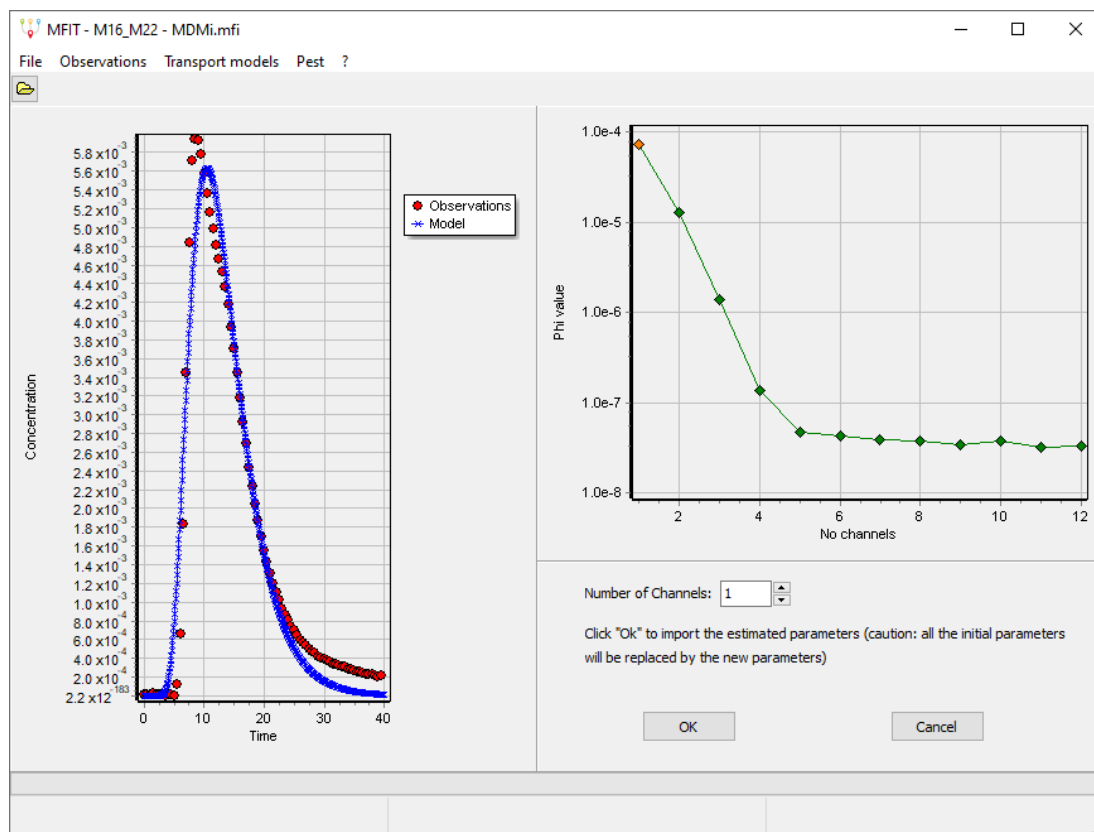


Fig. 11. Graphical display at the end of an Automatic Multiple Channels inversion.

4.3. Postcalibration uncertainty analysis

Because of the nonuniqueness of the inverse problem, some uncertainties may be associated with the PEST-optimized model parameter values. The postcalibration uncertainty of the model parameters can be assessed from random sampling and recalibration of a number of calibration-constrained parameter sets. The number of random parameter sets to be used for the uncertainty analysis can be specified by using the edit control at the bottom of the PEST option GUI dialog (Fig. 5). This number should be set as high as possible within computational time constraints. The processing steps are as follows:

1. Perform either a « User-Specified Channels » or « Automatic Multiple Channels » BTC inversion (see section 4.2). Particular attention must be paid to the lower and upper parameter bounds specified in the PEST parameter dialog (Figs. 6 and 7), as the uncertainty analysis method implemented in MFIT assumes that the bound difference represents the 95 percent prior confidence interval of the parameters.
2. At the end of the optimization process, click OK to import the PEST-optimized parameters into the model parameter table (if the « Automatic Multiple Channels » inversion was used, choose the channel-number solution for which you wish to assess the parameter uncertainty).
3. Select **Transport models | Compute BTC** to rerun the model with the set of optimized parameters.
4. If previously used, remove the Tikhonov regularization option in the PEST general settings.
5. Set up and run a new « User-Specified Channels » inversion based on the previously optimized parameter set. The PEST run should take only a few iterations. At the end of the optimization process, click CANCEL to close the dialog box.
6. Select **Pest | Run post-calibration uncertainty analysis**. Depending on the problem settings (selected transport model, number of random parameter sets, etc.), the computation time can range from a few minutes to several hours. Lengthy computations can be stopped at any time by selecting the **Pest | Abort optimization process** menu entry, or equivalently by clicking on the « Abort » shortcut button that is only visible during the optimization runs.

Once the uncertainty analysis process is complete, MFIT will display histograms of the recalibrated parameters (Fig. 12). The full set of recalibrated parameter values can be exported to a CSV text file using the button in the lower-right corner of the dialog.

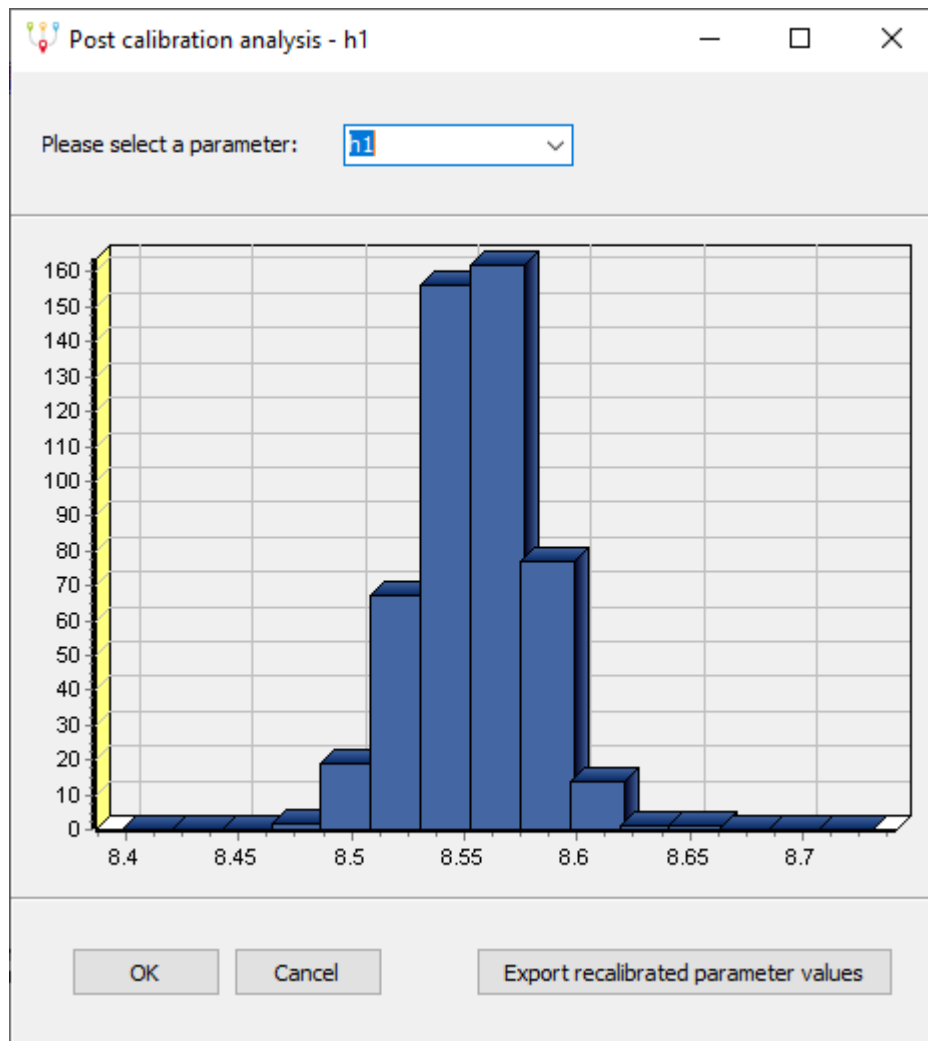


Fig. 12. Graphical display at the end of a postcalibration uncertainty analysis.

5. Contacts and technical support

If you have a question or problem, feel free to send me an email (jacques.bodin@univ-poitiers.fr). Any comment or suggestion related to improvement of the MFIT software or current manual will also be readily appreciated.