# MFIT v1.0.0 – Quick Start User Guide

### Jacques Bodin, 2019

## 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 multi-flow approach which 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 non-equilibrium model). These models were implemented as independent codes which can be run either as console applications, or using the MFIT program as a Graphical User Interface (GUI). MFIT provides a GUI for (i) the importation and graphic visualization of user-provided BTC data, (ii) the parameterization, direct run, and graphical output of the analytical transport models, (iii) the inversion (automatic calibration) of model parameters for optimal curve fitting, and (iv) the assessment of the uncertainty of calibrated parameter values. Both the optimization and uncertainty analysis of model parameters is carried out using routines from the PEST software package (<http://www.pesthomepage.org/>).

## Direct (forward) transport simulations

### Using the programs MDMi.exe, MDMed.exe, MDP\_SFDM.exe, and MDP\_2RNE.exe 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 format differs slightly depending on the model, see below. By running the transport model code, a series of time-concentration values are computed from the model input parameters and saved in a text file named « output.txt ».

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

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

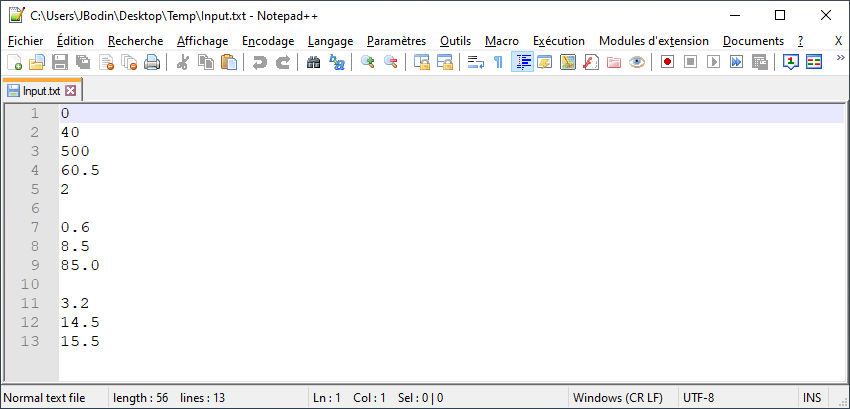
Solute mass flowing through the second channel, real number

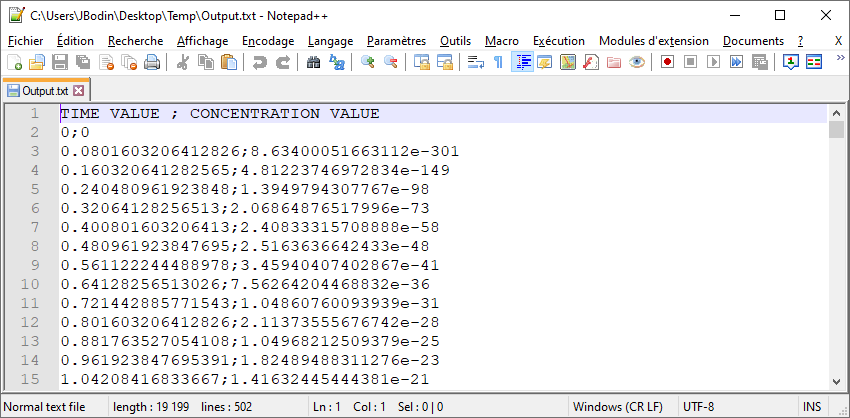
Mean transit time (T0) in the second channel, real number

Peclet number (Pe) of the second channel, real number

etc.

Care must be taken to use consistent units for all the variables in the input file, which also determine the time and concentration units of the output file (e.g., concentrations are in g/m3 if the the system flowrate has been expressed in m3/s, the solute-mass in g, and the mean transit times in s). Below are an example of input file for a two-channel MDMi model and the related output file written by the MDMi program. Only the first lines of the output file (which contains 500 time-concentration lines) are shown.





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

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 (C0), real number

Number of channels, integer

Flowrate contribution ratio of the first channel, real number

Mean transit time (T0) 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 (T0) 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 program MDP\_SFDM.exe

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

Diffusion parameter (Beta coefficient) of 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

Diffusion parameter (Beta coefficient) of 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 program MDP\_2RNE.exe

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

Length of the first channel, real number

Mean transit time (T0) 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 (T0) in the second channel, real number

Peclet number (Pe) of the second channel, real number

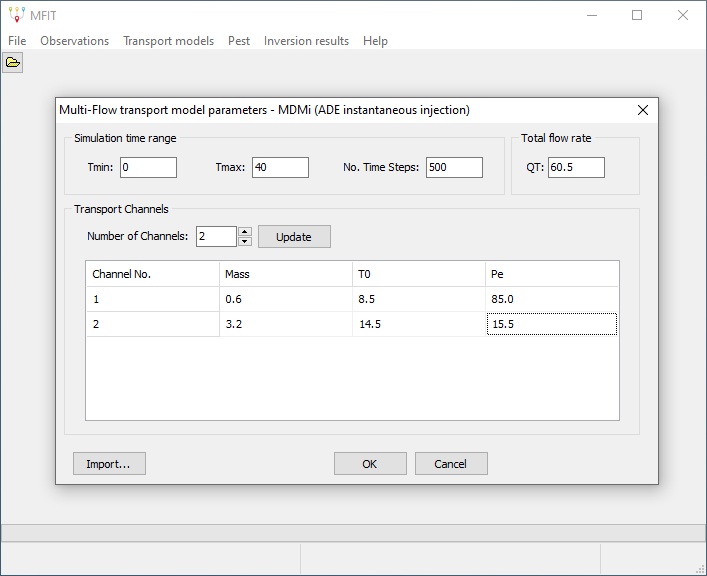
Psi coefficient of the second channel, real number

Omega coefficient of the second channel, real number

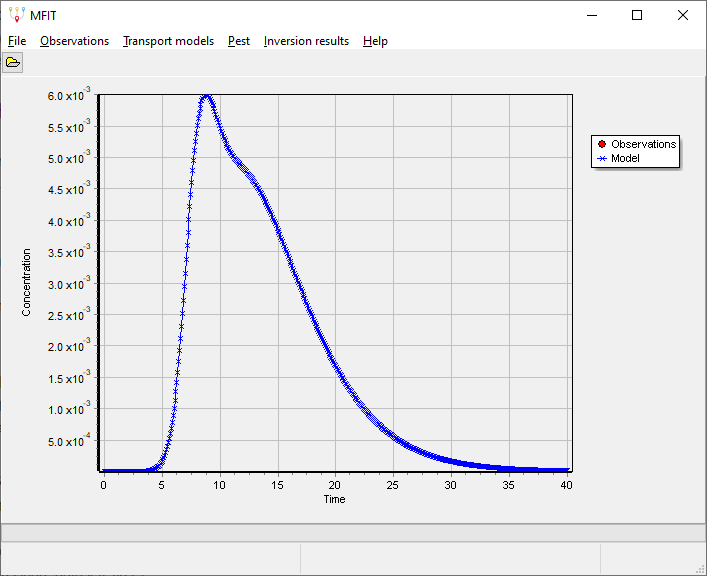
etc.

### Using MFIT as a GUI for the programs MDMi.exe, MDMed.exe, MDP\_SFDM.exe, and MDP\_2RNE.exe

1. Select **File | New** from the main menu.
2. Select **Transport models | 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 below is similar to the two-channel MDMi model input.txt file of Fig. #. 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.



1. Select **Transport models | Compute BTC**. The simulated BTC will be displayed (in blue). By clicking on the bottom- of left-axis black line, a dialog appears which allows to change the X- and Y-axis ranges.



1. (Optional) Select **File | Save As** for saving the simulation parameters and the path of the working directory into a MFIT file (extension: .mfi). The MFIT files can be edited with any text editor and can be re-opened using the **File | Open** menu command.

## Importing BTC data

If you already have some tracer-test data and if you want to use the MFIT software for modeling the observed concentration BTC, the first step is to import your BTC data in MFIT. The steps are as follows:

1. Start the MFIT software and select **File | New** from the main menu.
2. 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 is going to be used for the automatic calibration of a MFIT transport model against the user-provided BTC data (see section # below). If you don’t plan to use PEST, just apply any constant weight value to all the lines. The BTC data (including weights) can be either filled manually (use « Insert Row » or « Add Rows » for adding as many rows needed), or copy and pasted from a spreadsheet (e.g. Excel), or imported from a CSV text file (separator : comma).
3. 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 BTC, especially in the case of multiple local concentration peaks. A number of observations between 50 and 200 is appropriate in most cases.
4. Click on the Ok button will display (in red) the observed BTC.
5. (Optional) Select **File | Save As** for saving the Time-Concentration-Weight table into a MFIT file (extension: .mfi). The MFIT files can be edited with any text editor and can be re-opened using the **File | Open** menu command.

## Model-curve fitting of the observed BTC

### Manual trial-and-error procedure

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

### PEST model inversion

PEST is a model-independent optimizer that can be used for seeking the set of model parameters that allows the « best fit » between the simulated and observed BTCs. 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 squared weighted residuals between the observed and the 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 optimization results. If the same weight-value (e.g., 1.0) 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 allows to identify the solution that is the closest to some prior estimates of the model parameters. With the second option, the optimization process seeks a solution for which the model-parameter values for the different channels are as close as possible from each other (for a given parameter type). The steps for optimizing a MFIT transport model are as follows:

1. Import BTC data (see Section 3)
2. Choose and run one of the four MFIT transport problems, using trial-and-error as a first optimization method (see Section 2.2).
3. Select **Pest | Options** from the main menu. A tab dialog will be displayed that allows to set the PEST options. The default settings of the first tab are appropriate in most cases. For instance, unchecking the first check box will force PEST to compute all the Jacobian matrix derivatives using finite differences, which is both time consuming and less accurate. The second tab of the dialog allows to add some « preferred value » of « preferred homogeneity » Tikhonov regularization constraints to the optimization process. The variable PHIMLIM controls the trade-off 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 achieved by PEST, and then setting PHIMLIM 5 to 10 percent higher to this value.
4. is the target measurement objective function.
5. The reader is referred to the PEST documentation for more information about the recommended values for the PHIMLIM and PHIMACCEPT variables, as well as for more \* about the .
6. The third tab is for setting the termination criteria for the PEST optimization of model parameters. The fouth and last tab is that can be modified from the option window dialog.

## Contacts and technical support

If you have a question or problem, feel free to send me an email ([jacques.bodin@univ-poitiers.fr](mailto:jacques.bodin@univ-poitiers.fr))