SMIMfit Documentation

Cite: Volponi, S. (2024). *SMIMfit library* [MATLAB].https://doi.org/10.5281/zenodo.11147719

Introduction

This document gives a high-level description of the functionality of the SMIMfit library, which allows a user to fit transport parameters in the stochastic mobile-immobile model (SMIM) to breakthrough curve data. The library is an extension of the code developed by Mike Schmidt (Schmidt, 2020), incorporating a more robust curve fitting approach and calculates the standard error of fitted model parameters. The library is an adaptation of the methods outlined in Kelly et al. (2017) and Cortis and Berkowitz (2005), both of which can be consulted to better understand the methods employed within SMIMfit. The primary advance provided by this library is the ability to include first-order reaction rates for the mobile and immobile zones, allowing the user to fit reactive breakthrough curves. Although not yet incorporated, we plan to provide a future version of the SMIMfit with this capability included.

The parameters fit by our model include average channel velocity $[V, \frac{m}{s}]$ and dispersion $[D, \frac{m^2}{s}]$, subsurface exchange rate $[\Lambda, \frac{1}{s}]$, truncation start $[T_1, s]$ and end times $[T_1, s]$, and power law slope $[\beta]$.

Preparing the Data

Experimental data should be entered into the Excel spreadsheet *TemplateExample.xlsx*. Detailed instructions for data entry are found within the *Instructions* tab of the workbook, the following data is required:

1. Experimental concentration $[mg/L]$ and time [HH:MM:SS] data.
2. Release time [HH:MM:SS]
3. Release distance [m]
4. Injected tracer mass [mg]
5. The time [HH:MM:SS] and day [DD] at which the injected tracer first reaches the sensor.
6. The time [HH:MM:SS] and day [DD] at which measured concentrations return to pre-injection baselines after the injection.
7. The time between subsequent concentration measurements [s].
8. *Optional Parameter:* Discharge $[m^3/s]$. If unknown, SMIMfit will provide an estimate.

In addition to instructions, the workbook contains a blank template to be filled in and a pre-filled example (example_Release1). The example_Release1 sheet can be directly inputted into the model for exploration. Ensure that the filled template is in the same folder as the Matlab code files so that it can be read in.

Running the Model

A. Start the model

To run the model, open the Matlab code file *Lead_SMIM.m*. Running the code requires that the user specify the following information (lines 8 – 22):

1. filename – The name of the Excel template with data to be fitted.
2. params_upper – Upper bounds to be used in the optimization for the 6 parameters being fit, listed as: $[V,\ D,\ \Lambda,\ \log_{10}(T_1),\ \log_{10}(T_2),\ \beta]$.
3. params_lower – Lower bounds to be used in the optimization for the 6 parameters being fit, listed as: $[V,\ D,\ \Lambda,\ \log_{10}(T_1),\ \log_{10}(T_2),\ \beta]$.

In the subsequent section, the code loads in data from the filled Excel template. The template file needs to be in the same folder as the Matlab code for its data to be accessed.

B. Information on the SMIMfit function

The Matlab function *SMIMfi*t performs the optimization procedure. *SMIMfit* takes as an input a normalized concentration vector (i.e., sums to 1). Within the code, an experimental concentration vector is normalized using the function *cNorm*. More details on the inputs and outputs of *SMIMfit* and *cNorm* can be found within their corresponding Matlab files.

SMIM parameters are fit by performing a nonlinear least squares optimization with 60 random restarts. Starting positions are selected using Latin Hypercube sampling. The optimization problem is defined within the code file *runFit.m* and more details are provided within the code file.

SMIM parameters are optimized to minimize a weighted mean absolute error objective. See the file *objFunctionSMIM.m* for more details on the objective.

The output of *SMIMfit* is a cell array (*ModCon*) that contains the following information:

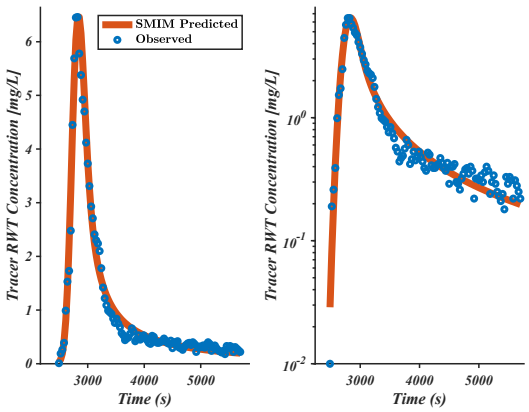
1. ModCon.params_fit – Array of optimized model parameters, listed as: $[V,\ D,\ \Lambda,\ \log_{10}(T_1),\ \log_{10}(T_2),\ \beta]$.
2. ModCon.tcfit – Vector of time values that normalized experimental data and SMIM predictions correspond to.

3. ModCon.cobs – Normalized, experimental concentration data.
4. ModCon.ccfit – SMIM predicted concentrations for each time in ModCon.tcfit, generated using the optimized parameter set.
5. ModCon.resid – Objective function evaluation for the optimized parameter set.
6. ModCon.resnorm – The squared norm of the residual for the optimized parameter set.
7. ModCon.lambda – Lagrange multiplier at the for the optimized parameter set.
8. ModCon.jacobian – Jacobian at the optimized for the optimized parameter set.

C. Model Outputs

After the optimization is complete, *Lead_SMIM.m* generates the following data summary products:

1. An excel file (*SMIMResults.xls*) containing model fit for all the tabs included in the Excel template. Reported information includes:
    a. Optimized parameter values.
    b. The standard error associated with the model fit for each optimized parameter (not to be confused with experimental error).
    c. The objective output for the optimized parameter set.
    d. The mass recovery fraction
    e. Volume flow rate, which is estimated if unknown at the outset.
2. A Matlab figure *for each* run tab comparing observed and SMIM predicted concentrations.



3. A Matlab workspace containing the final workspace produced *for each* tab in the Excel template file.

References

Cortis, A., & Berkowitz, B. (2005). Computing “anomalous” contaminant transport in porous media: The ctrw matlab toolbox. *Groundwater*, *43*(6), 947–950. https://doi.org/10.1111/j.1745-6584.2005.00045.x

Kelly, J. F., Bolster, D., Meerschaert, M. M., Drummond, J. D., & Packman, A. I. (2017). FracFit: A robust parameter estimation tool for fractional calculus models. *Water Resources Research*, *53*(3), 2559–2567. https://doi.org/10.1002/2016WR019748

Schmidt, M. (2020). *SMIMfit library* [MATLAB]. https://github.com/mjs271/SMIMfit (Original work published 2020)