Stochastic Assessment for Non-Point Source Contamination of Heterogeneous Aquifer: Instructions for Inputs and Outputs

Christopher V. Henri
chenri@ucdavis.edu

This document provides instructions to generate the input files of the different software used to model transport from a nonpoint source into a heterogeneous aquifer within a stochastic framework.

We refer to the manuscripts attached to this document if further information about the mathematical and conceptual background of the study is needed.

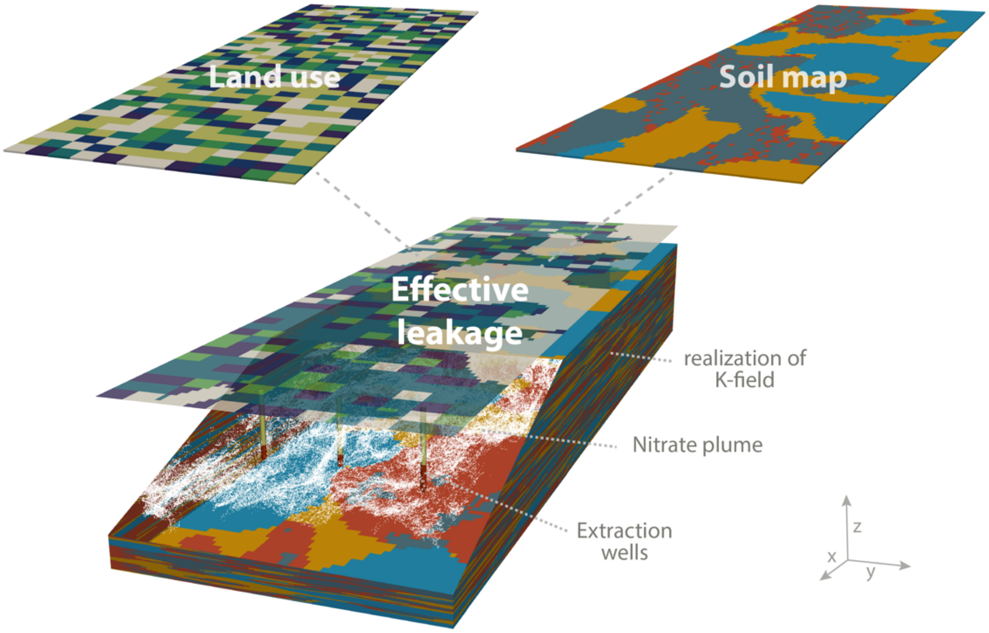

Figure 1: Illustration of the simulation setup

1. T-PROGS

The same T-PROGS model is used for all studies made in the project. In our study, we do not use any data to condition the geostatistical model, but T-PROGS is proposing this option. Two set of parameters have to be generated in order to run the program: MCMOD inputs and TSIM inputs. MCMOD and TSIM inputs were generated using the GUI available for the TPROGS software. T-PROGS manual provides the scientific background and the detailed instructions on how to generate the needed inputs. A set of example parameter files and the *tsim* executable in provided in the folder /1_TPROGS. The software *tsim* will generate a single file with the facies spatial distribution for all realizations (output file with extension "*.asc*").

The only post-process required here to proceed with our stochastic analysis is to split this output file into a file per realization and convert the facies index to a corresponding value of hydraulic conductivity. A Matlab script does this:

tsim_to_Kmat.mat: Generates hydraulic conductivity fields to be later used to generate the input of the Modflow model

2. MODFLOW

Some inputs are common to all realizations. It should be created once only, depending on the model main characteristics, and be used for any simulation. These files will be copied in the sub-folder used to run Modflow-2000 (MF2K) during the Monte Carlo simulation. These files, common to all realizations are: *BAS, DIS, GMG, MNWI, OC.* We refer to the Modflow online manual for details about these packages. An example of each file is provided in the folder /2_MF2K.

Some other input files depend on the hydraulic conductivity field, such as the recharge rate, the well package and the K-weighted fluxes leaving the domain and are, therefore, realization dependent. Some parameters used as input to compute the flow field is also dependent on the land use.

These provided Matlab scripts help generate the series (for all realization) of input files:

get_landuse_cons.m: Generates a series of *RCH* files, which are MF2K input files that mostly specify the spatial variability of the recharge rate; and generates the spatial distribution of the particle density used for the transport modeling.

In our study, the local recharge rate depends on the soil and the crop types. The soil type is defined as the first (top) layer of the hydraulic conductivity field previously generated.

The Matlab script generates a random spatial distribution of a series of crop over the domain, which is also used to generates the spatial distribution of the particle density used for the transport modeling.

The crop and soil type dependent recharge rate and contaminant mass flux are obtained by means of a series of Hydrus-1D simulations. We refer to the attached manuscript for more details.

KweightedFHB.m: Generates a series of *FHB* packages, which specify the spatial distribution of prescribed fluxes to be applied at the bottom of the domain in order to simulate non-represented extraction. The local flux is proportional to the local hydraulic conductivity.

mnw2_pack.m: Generates the *MNW2* packages for each realization with different pumping rate, screen length, and top depth. 3 extraction wells are implemented in each simulation. The well location is selected in order to always have 10 ft of gravel/sand for each 100 gpm of pumping rate. If this is not doable at the given well location, the algorithm changes this location until the criteria is fulfilled.

get_nam.m: Generates the name file (specifies MF2K input and output files names) for each realization.

3. RW3D

The transport is solved using the random-walk particle-tracking method. We use the code RW3D, which is provided (compiled executable and source code) in its last version. The code presents a high versatility in the processes that can be simulated: linear sorption (retardation), linear reaction network (first-order decay), multi-rate mass transfer, non-linear bi-molecular reaction network. We refer to the attached readme file for details about the specification of required inputs. In our Monte Carlo framework, all inputs are realization dependent and are generated using the following Matlab script:

get_nameRW3D.m: Generates the name file (specifies RW3D input and output files names) for each realization.

get_paramRW3D.m: Generates the main input file specifying all parameters for each realization. Among other, this file is calling the flow field previously generated by MF2K and the file specifying the spatial distribution of the original particle density generated by the script *get_landuse_cons.m.*

4. BATCH

The flow field (via MF2K) and the transport (via RW3D) are computed for all realizations using a Matlab script (/4_BATCH/batch.mat) organizing all necessary (and previously generated) inputs in a realization-specific folder and calling both software. Once all inputs are generated, only this script has to be ran. It allows both parallel and serial simulations.

5. OUTPUTS

The objective of our studies is to better understand the spatio-temporal behavior of a series of management metrics adapted to nonpoint source contamination under highly uncertain conditions. To do so, we designed a series of visual tools to easily and stochastically assess travel times, capture zone (or contributing area) spatial and temporal extension, and contaminant levels at extraction wells.

We analyze two main transport outputs from RW3D: the (cumulative) breakthrough curves to compute travel times and concentration statistics; the original location of particles reaching a well during the simulation to compute the spatio-temporal behavior of the contributing area.

Travel time statistics

Travel times (or arrival times), indicating the groundwater age mixed within well screens are here analyzed stochastically. Each realization simulate arrivals of mass in 3 extraction wells and all data are gathered in order to be stochastically analyzed. The process to visualize any output consists mainly in two steps: the gathering / preparation of the data and the plotting. In case of travel time statistics analysis, two Matlab scripts (located in /4_OUTPUTS/travel_times/) do these jobs:

Get_travel_times_pdfs.m: Reads and analyze the cumulative breakthrough for all realizations. The time of arrival of a given percentage of the total mass reaching the well is recorded for each realization, and pdfs are then computed. The pdfs of a series of characteristic travel times (for 5%, 10%, …, 95% of the total mass) are exported in an Excel file.

plot_travel_times_stats_and_pdfs.m: Plots the pdfs of a series of travel times (of 5%, 50% and 95% of the total mass in the script current form) as well as their lower moments (mean, variance or coefficient of variation). Requires a previously produced excel file with all statistics. The script is designed to produce plots comparing outputs from different simulation setting (different depths of the wells, or different pumping rates).

In our study, we also analyze characteristic arrival times, i.e., the time required for a solute to reach the top, the middle and the bottom of the screen (average elevations are used) in an equivalent homogeneous setup, without extraction. These characteristic times have to be previously evaluated (see the manuscript for more detail). An example of produced figure is shown below:

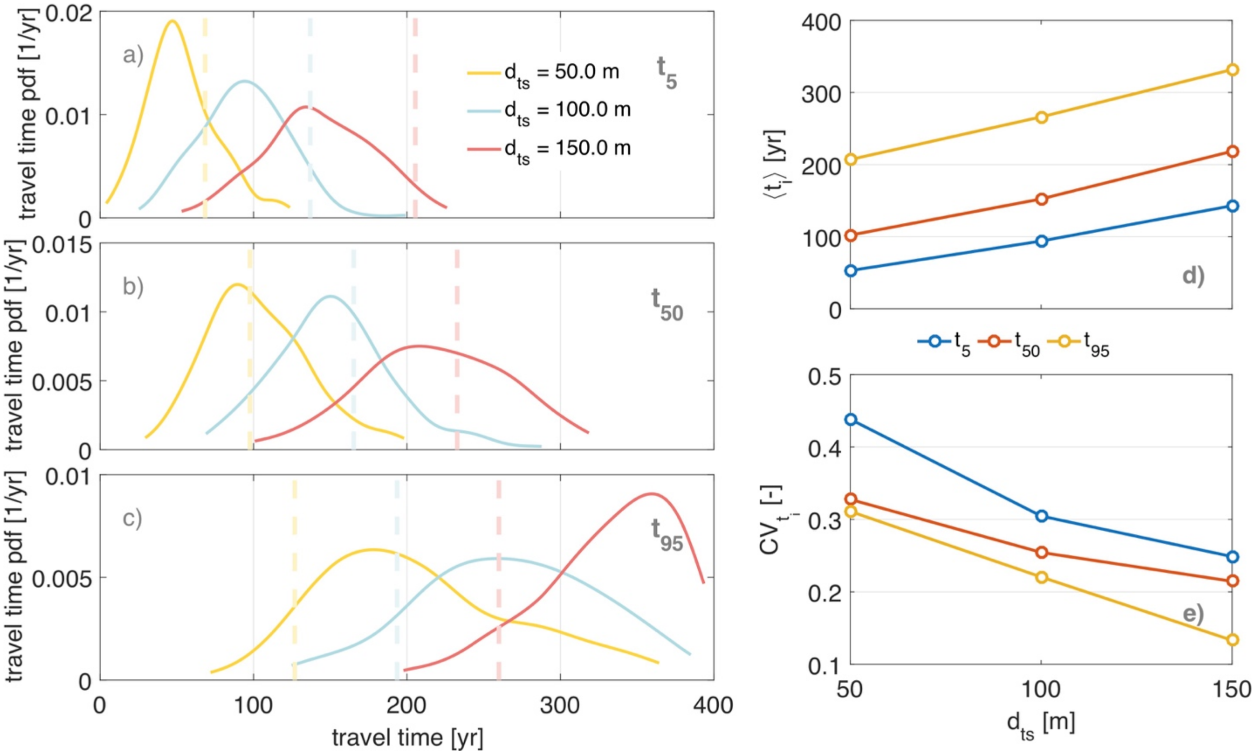

Figure 2: Illustration of potential output showing the travel time pdfs (a-c) and lower moments (d-e) for different scenarios (screen depth)

Contaminant level statistics

Breakthrough curves can also be used to analyze the contaminant levels temporal evolution predicted in wells. In a stochastic framework, concentrations can be analyzed in term of probabilities to observe a given contaminant level at a given time. For this metric too, two Matlab scripts (located in /4_OUTPUTS/concentrations/) do these jobs:

get_cbtc_db.m: Generates a database gathering the concentration signal recorded on all breakthrough curves (number of well per realization times number of realizations). The times are fixed (vector to be defined in script), and the simulated concentrations at any given time is linearly interpolated.

plot_cVSt.m: Creates a plot showing the probability to exceed a given concentration (in y-axis) at a given time (in x-axis). The database previously generated is used. Here again, our study analyzes the potential output in an equivalent homogeneous case, without extraction (see the manuscript for details). A potential illustration can be seen below:

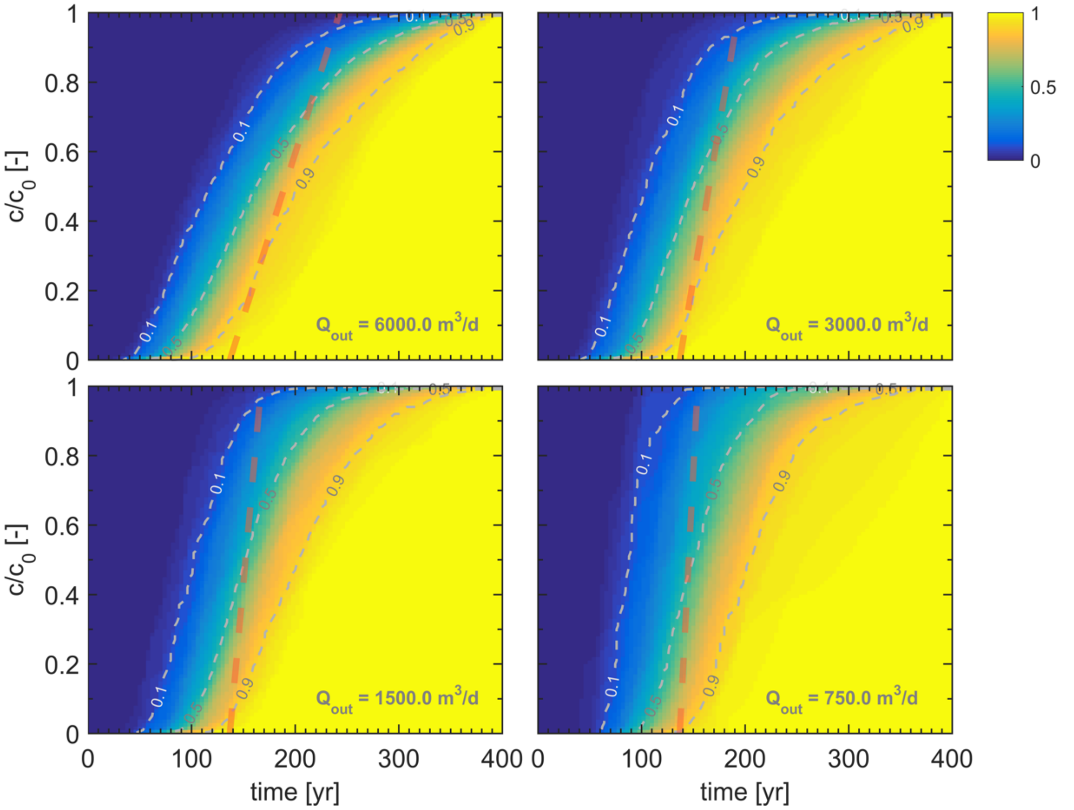

Figure 3: Illustration of potential output showing the probability to exceed a (normalized) contaminant level at a given time

Capture zone

For this metric too, two Matlab scripts (located in /4_OUTPUTS/contributing_areas/) do these jobs:

get_birthplace.m: Analyzes the particles birthplace (original location of particles reaching a well during the simulation) and the time at which each particle reached a well. On a predefined grid (x-y on the domain top edge), the script computes then the probability for a particle leaving a given grid cell to reach a well. The script also computes the average and the uncertainty (coefficient of variation or standard deviation) of the time required for a particle leaving a given grid cell to reach a well.

Generates finally an Excel file with all matrices (probability to reach well, mean travel time, travel time uncertainty).

plot_stat_maps.m: Creates a series of plots showing the probability for a particle leaving a given grid cell to reach a well, the average or the uncertainty (coefficient of variation or standard deviation) of the time required for a particle leaving a given grid cell to reach a well. The script allows a comparison between the different metric for different simulation setup (i.e., screen depths or pumping rates).

For comparison purpose, the upstream, middle, and downstream location of an equivalent contributing area in a homogeneous, non-pumping case can also be shown on the plot (see the manuscript for details). A potential illustration can be seen below:

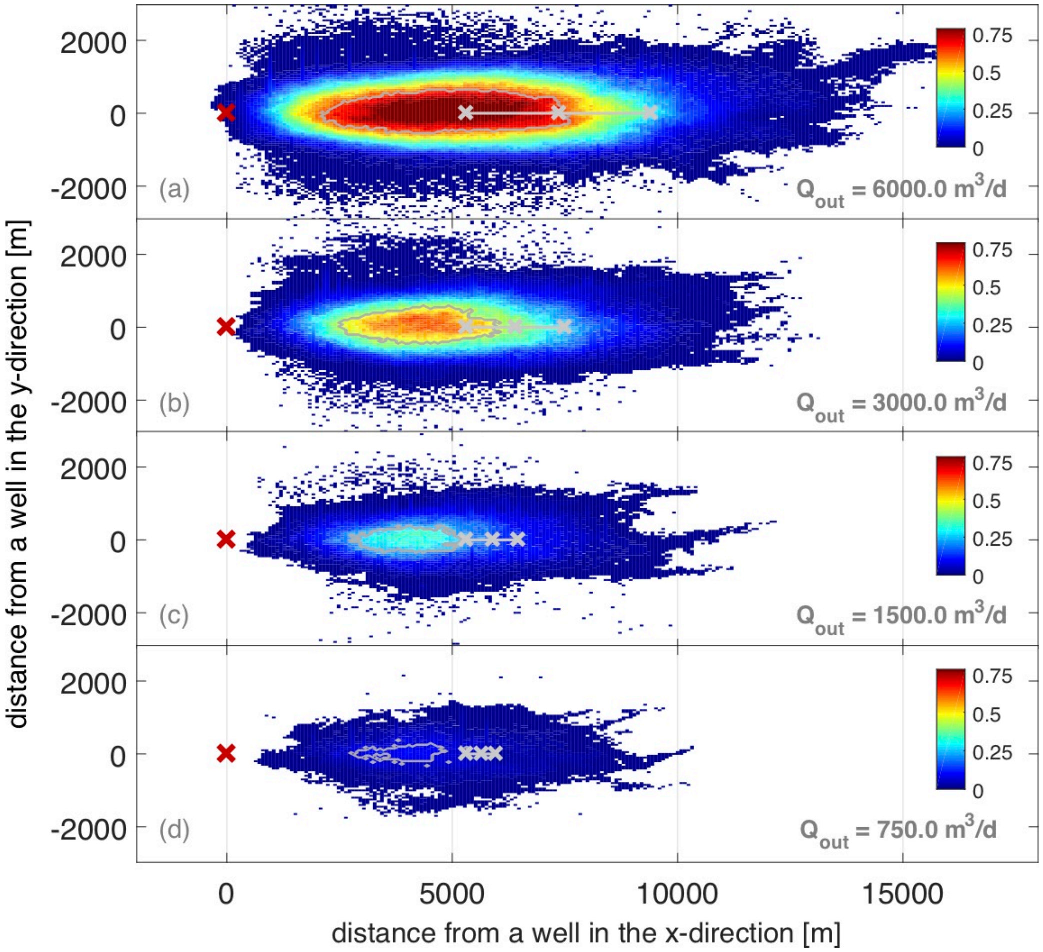

Figure 4: Illustration of potential output showing the probability for a particle leaving a given grid cell to reach a well

6. Adaptation to pesticide contamination

The simulation setting of this study was designed for non-reactive solute, i.e., no degradation or transformation are considered in the aquifer. Conceptually, simulating pesticide transport may require

(1) to adapt the effective leakage rates of water and contaminant simulated by Hydrus 1D;

(2) to simulate degradation in the groundwater;

(3) to simulation sorption in the groundwater.

Adding reactions in the solute transport modeling is an easy task using RW3D.

To add first order decay (degradation), edit the line 176 – 182 of the script *get_paramRW3D.m*.

To add retardation (linear instantaneous sorption), edit the line 165 – 168 of the script *get_paramRW3D.m*.