Steady-State Diffusion in a Spatially Varying Porous Medium: Data Sets

Authors: Peter T. Ferrero V, Brian D. Wood
Contact Info: brian.wood@oregonstate.edu

Abstract

This repository contains the data associated with the paper "Steady-State Diffusion in a Spatially Varying Porous Medium" to be published in Advances in Water Resources. These data include the information and program scripts required to reproduce the results reported in the publication. In particular, the data set includes (1) COMSOL and MATLAB files to reproduce the simulations for Cases 1-4, (2) the files required to calculate the closure problem source terms for all four cases, (3) mesh files in open-source format providing the mesh used in the computations associated with the paper, (4) the data used to produce the figures in the paper.

README

This document is meant to support the use of the files contain within this archive listing. Care has been taken to make the programming and file structure for each of the four simulations completely standardized. As a result, the contents of this file pertain to each simulation listed as Cases 1-4. The cases are listed as follows

1. **Case 1**: Steady-state diffusion in a periodic homogeneous porous medium.
2. **Case 2**: Steady-state diffusion in a quasi-periodic heterogeneous porous medium with a gradual porosity change.
3. **Case 3**: Steady-state diffusion in a quasi-periodic heterogeneous porous medium with a jump porosity change.
4. **Case 4**: Steady-state diffusion in a disordered heterogeneous porous medium with a jump porosity change.

System Requirements

To run these simulations, the user must have access to MATLAB (R2017b was used for this work) and the COMSOL v5.3 libraries. The preferred setup would be to link MATLAB to COMSOL with the "Livelink for MATLAB" server application.

Getting Started

If your goal is to reproduce the results in our publication. Simply open the folder of which case (Cases 1-4) results you would like to view. To reproduce the results and plots, simply run the MATLAB script "CaseX_Main.m" in the main folder of each case and the scripts will automatically set up the simulation and calculate the solutions presented in our publication. The figures shown in our publication will then be displayed to you. The MATLAB files for the Cases 1-4 folders (here denoted by 'CaseX') are listed as follows:

1. **CaseX_Main.m**: The main driver script that calls the other required main scripts.

2. **CaseX_DNS.m**: A MATLAB script that sets up the COMSOL model and directly solves the diffusion equation over the microscale domain. This script also calculates and saves the values of the averaged parameters such as the porosity, first moment, and second moment.

3. **CaseX_bField.m**: This script solves the closure problem and then uses the solution to calculate the effective diffusivity coefficients for models 0, 1, and 2.

4. **CaseX_Macroscale.m**: This script solves the macroscale average diffusion equation using the effective diffusivity coefficients calculated in "CaseX_bField.m" and then calculates the relative error compared to the DNS solution.

COMSOL Files

In the subdirectory "mph Files", the native COMSOL model formatted (".mph") are provided for Cases 1-4. The files are stored in COMSOL 5.3a format. These are readable with COMSOL version 5.3a or later, but are not reverse compatible with earlier versions of COMSOL. Additionally, the ".m" formatted files are provided and may be run in MATLAB if the appropriate COMSOL libraries are loaded. The ".m" formatted files are extensively commented and are provided with the hope that new users may further experiment with the results presented in our publication.

CSV Data

For cases 1-4 (here denoted by 'CaseX'), CSV data files are contained within the folder 'CSV Data Files'. These data files are produced from the MATLAB .m files: CaseX_DNS.m, CaseX_bField.m, and CaseX_Macroscale.m. Each data file is a plain-ASCII text file in comma separated value (CSV) format. The data contained in each of the .csv files is denoted by the title of the .csv file, for example 'Porosity.csv' contains the data relating to the porosity of the porous material.

CSV Data Format. The general format for the .csv files is as follows

---

```
% Header
% ...
% X, Y, Parameter
```

(+Data)

The following is the list of files stored in this folder where Y denotes models 0, 1, or 2:

1. **DeviationConcentrationSource.csv**: This file contains data for the nonlocal deviation concentration source term $\frac{1}{V}\int_{\mathbf{y}\in\mathscr{A}_{\gamma\kappa}(\mathbf{x})}\mathbf{n}_{\gamma\kappa}\tilde{c}_{A\gamma}w(\mathbf{y})\,dA(\mathbf{y})$, $[\frac{mol}{m^4}]$.

2. **DNS_Solution.csv**: This file contains the DNS concentration profile accepted as the true solution to the simulation, $[\frac{mol}{m^3}]$.

3. **DNSAverageConcentration.csv**: This file contains the data for the superficially averaged DNS concentration profile, $[\frac{mol}{m^3}]$.

4. **FirstMoment.csv**: This file contains the γ-phase first moment data, $[m]$.

5. **MacroscalePorosity.csv**: A file containing the porosity data used in "CaseX_Macroscale.m", [1].

6. **ModelY_EFfectiveDiffusivity_xx.csv**: A file containing the effective diffusivity coefficient used in the volume averaged model, $[\frac{m^2}{s}]$.

7. **ModelY_RelativeError.csv**: A file containing the relative error of Model X compared to the DNS solution, [1].

8. **ModelY_Solution.csv**: A file containing the concentration profile produced by volume averaging model X, $[\frac{mol}{m^3}]$.

9. **Porosity.csv**: A file containing the porosity date produced by "CaseX_DNS.m", [1].

10. **SecondMoment.csv**: A file containing the γ-phase second moment data produced by "CaseX_DNS.m", $[m^2]$.

Figures

The "Figure" directory contains the figures saved by the "TaylorSeriesPlotter.m" and "MacroscalePlots.m" MATLAB scripts in the "Utility Files" folder. The figures are saved as Enhanced MetaFiles. The plots produced are as follows:

1. **ConcentrationProfiles.emf**: This plot displays the concentration profiles produced by the DNS and volume averaged solutions. Four curves are plotted with the following line types, (1) the DNS with a dashed black line, (2) model 0 with a solid blue line, (3) model 1 with a solid green line, and (4) model 2 with a solid red line.

2. **ConcentrationRelativeError.emf**: This plot displays the relative error of each volume averaged model compared to the DNS solution. The color scheme used is model 0: solid blue line, model 1: solid green line, model 2: solid red line.

3. **EffectiveDiffusion.emf**: This plot displays the xx-component of the effective diffusivity tensor. The color scheme used is model 0: solid blue line, model 1: solid green line, model 2: solid red line.

4. **TaylorSeriesPlot.emf**: This plot displays the first three partial sums of the Taylor series approximation of the average molar flux. The color scheme used is (1) first term, solid blue line, (2) first two terms, solid orange line, (3) first three terms, solid yellow line.

Mesh Data

The mesh data in this repository is contained in the "Meshes" folder. The data appears in two formats. The first is exported as a NASTRAN file, one of the formats natively exported from COMSOL. The NASTRAN file (.nas) is an ASCII formatted field that can be read by many open-source mesh tools including the tool GMSH (http://gmsh.info//) which is available for most operating systems. The NASTRAN format itself is well-documented in many sources; the Wikipedia entry (https://en.wikipedia.org/wiki/Nastran) is one stable resource that contains a substantial description of the format. The original NASA source code was available (as of 1 January 2018) on GitHub (https://github.com/nasa/NASTRAN-93). This also provides documentation that can be used to interpret the NASTRAN format.

The second file format is the ".msh" format that is also an ASCII-formatted file. It represents the proprietary output format for the open-source mesh-handling software GMSH (http://gmsh.info//).

Text File Data

For cases 1-4 (here denoted by 'CaseX'), .txt data files are contained within the folder 'Text File Data'. These data files are produced from the MATLAB .m files: AveragedDataConverter.m and EffectiveDiffusivityDataConverter.m located in the "Utility Files" directory. The data contained in each of the .txt files is denoted by the title of the .txt file, for example 'Porosity.txt' contains the data relating to the porosity of the porous material.

TXT Data Format. The general format for the .txt files is as follows

```
x Parameter
```

(+Data)

The following is the list of files stored in this folder where Y denotes models 0, 1, or 2:

1. **CenterPositions.txt**: A file containing a list of the κ-phase squares or circles in the computation domain.
2. **DNSAverageConcentration.txt**: A file containing the DNS concentration profile. This is taken to be the true solution of the simulation, $[\frac{mol}{m^3}]$.
3. **FirstMoment_x.txt**: A file containing the first derivative with respect to x of the x-component of the first moment data of the γ-phase, [1].
4. **FirstMoment_xx.txt**: A file containing the second derivative with respect to x of the x-component of the first moment data of the γ-phase, $\frac{1}{m}$.
5. **ModelY_EffectiveDiffusivity_xx.txt**: A file containing the xx-component of the effective diffusivity tensor calculated for Model Y, $[\frac{m^2}{s}]$
6. **Parameters.txt**: A file containing the parameter data necessary to run the simulation.
7. **Porosity.txt**: A file containing the porosity data of the γ-phase, [1].
8. **Porosity_x.txt**: A file containing the first derivative with respect to x of the porosity of the γ-phase, $[\frac{1}{m}]$.

The Parameter File

Each simulation contains a file named "Parameters.txt" file in the "Text File Data" folder containing the basic parameter information needed to set up the model. This files contains the overall domain length and width, the period length, the formulas for calculating the κ-phase size lengths necessary for a target porosity, the microscale diffusion coefficient and the weighting function normalization coefficient. An example file is given below

---

```
L 14[mm] "Macroscale Domain Length, x-Direction"
W 8[mm] "Macroscale Domain Width, y-Direction"
Lp 1[mm] "Period Length"
epsilon_gamma 0.4 "Target Gamma-Phase Porosity"
r sqrt((Lp^2)*(1-epsilon_gamma)) "Kappa-Phase Square Side Length"
Da 1e-9[m^2/s] "Microscale Diffusion Coefficient"
A 1 "Weighting Function Coefficient"
```

κ-Phase Center Placement Files

Each simulation contains a "CenterPositions.txt" file in the "Text File Data" folder that lists the positions and side length (or radii) of the κ-phase center positions. The general format for this file is as follows

```
x Position, y Position, Side Length/Radius
```

(+Data)

Note: For Case 4, two such files exist. One for the "Case4_DNS.m" file and one for the "Case4_bField.m" file. Two separate files are needed since computation domain size is reduced by half of the weighting function window size along the edges of the "Case4_DNS.m" domain when moving to the "Case4_bField.m" MATLAB script. This is because we neglect the effect of the edges of the domain in our analysis for this work. As a result, the two κ-phase center positions files are titled "CenterPositionsDNS.txt" and "CenterPositionsBField.txt".

Utility Files

The Utility Files folder contains the following MATLAB scripts:

1. **AveragedDataConverter.m**: Converts the averaged parameters produced by "CaseX_DNS.m" from .csv to .txt files. Also calls the "Derivative.m" subroutine to perform numerical derivatives on the averaged parameters to produce the source terms needed for "CaseX_bField.m". These source terms are also saved as .txt files.

2. **CenteredDerivativeCoeff.m**: Calculates the weighting coefficients for the centered derivative approximations for "Derivative.m".

3. **Derivative.m**: Calculates the n^{th}-order derivative with a m^{th}-order accurate finite difference method. Additional files belonging to this subroutine are "DerivativeMatrixAssembler.m", "CenteredDerivativeCoeff.m", and "OneSidedDerivativeCoeff.m".

4. **DerivativeMatrixAssembler.m**: Creates the differentiation matrix which contains indexed coefficients for the one-sided and centered derivative approximations. This matrix is then used to calculate the numerical derivatives of the data inputted into the function call.

5. **EffectiveDiffusivityDataConverter.m**: converts the components of the effective diffusivity tensor calculated in "CaseX_bField.m" from .csv to .txt files.

6. **MacroscalePlots.m**: Displays the volume averaged solution, relative error, and xx-component of the effective diffusivity tensor plots for Models 0, 1, and 2 to the user.

7. **OneSidedDerivativeCoeff.m**: Calculates the weighting coefficients for the one-sided derivative approximations for "Derivative.m".

8. **TaylorSeriesPlotter.m**: Produces the Taylor series plot shows the convergence of the average flux as higher order moment terms are summed together. The plot is shown to the user when called.

Weighting Function

The MATLAB script "ConvolutionMain.m" will produce the mollified triangle function .txt file used by the COMSOL models. For Cases 1-3, a 1D mollified triangle function is used while a 2D version is created for Case 4. To create a mollified triangle function, simply execute a function call to "ConvolutionMain.m" from the MATLAB command line. For Cases 1-3, the function call is

Cases 1-3

```
ConvolutionMain(xMin, xMax, step, ep)
```

Case 4

```
ConvolutionMain(xMin, xMax, yMin, yMax, step, ep)
```

where xMin, xMax, yMin, yMax, step, and ep are defined for the user in the header of "ConvolutionMain.m".

Note: A default weighting function .txt file is already included in this package for each case so the user does not need to change the weighting function if they do not desire to do so. The file is contained in the "Weighting Function" folder and is named "WeightingFunction.txt".

The Weighting Function folder contains the following MATLAB scripts:

1. **BumpFunction.m**: A subfunction that creates the bump function used to mollify the initial unsmooth weighting function.

2. **ConvolutionMain.m**: The main function file that creates the mollified weighting function and then saves it under "WeightingFunction.txt".

3. **Rectangle.m**: An optional unsmooth weighting function. May be swapped out with the "Triangle.m" function to create mollified rectangle weighting functions.

4. **Triangle.m**: The default unsmooth weighting function. Creates an unsmoothed triangle weighting function.