Comput Geosci (2006) 10:343–359
DOI 10.1007/s10596-006-9028-4

JHomogenizer: a computational tool for upscaling permeability for flow in heterogeneous porous media

Brahim Amaziane · Joe Koebbe

Received: 4 September 2004 / Accepted: 23 June 2006 / Published online: 20 October 2006

Abstract This paper presents an object-oriented programming approach for the design of numerical homogenization programs, called JHomogenizer. It currently includes five functional modules to compute effective permeability and simple codes for computing solutions for flow in porous media. Examples with graphical output are shown to illustrate some functionalities of the program. A series of numerical examples demonstrates the effectiveness of the methodology for two-phase flow in heterogeneous reservoirs. The software is freely available, and the open architecture of the program facilitates further development and can adapt to suit specific needs easily and quickly.

Keywords heterogeneous porous media · numerical homogenization · two-phase flow · upscaling

Mathematics Subject Classifications (2000) 65N30 · 65Y15 · 68U01 · 68U20 · 74Q99 · 76T99 · 76M50

1 Introduction

Homogenization is a fundamental tool for the modeling of various phenomena taking place in a heterogeneous medium, in particular for flow and transport in porous media, in oil reservoir simulation or hydrogeology. There is an extensive literature on this subject. We will not attempt a literature review here, but merely mention a few references. In this paper, we restrict ourselves to the mathematical homogenization method as described in [21] and [29] for flow in porous media. For recent reviews on other upscaling methods, see, for instance, [20] and references therein.

The objective of homogenization is to replace the governing equations by a simpler set of equations for which the solution can be resolved on a reasonable coarse-scale mesh and approximates the average behavior of the solution of the governing equations. In its simplest form, one replaces the coefficients of the governing equations with effective or macroscopic coefficients. For a more advanced presentation of homogenization, the reader is referred to the classical books [8, 11, 19, 22, 32]. An extensive collection of applications to porous media can be found in [21].

In the present paper, we will focus on the computation of effective permeability for moderately heterogeneous media obtained via homogenization within a user-friendly computational tool JHomogenizer [26]. Each homogenization method leads to the definition of a global or effective model of a homogeneous reservoir defined by the computed effective coefficients. Homogenization methods allow the determination of these effective coefficients from knowledge of the geometrical structure of a basic cell and its heterogeneities by solving appropriate local problems. The technique is based on numerics. We assume that data given on a fine grid fully represent the important physical scales and that a practical computational grid must be somewhat coarser.

B. Amaziane (✉)
Université de Pau, LMA, CNRS-UMR 5142,
Av. de l'Université, 64000 Pau, France
e-mail: brahim.amaziane@univ-pau.fr

J. Koebbe
Department of Mathematics and Statistics,
Utah State University, Logan, UT 84322-3900, USA
e-mail: koebbe@math.usu.edu

In the homogenization methods described and implemented in this work, we use conforming and mixed finite elements to compute approximate solutions of the local problems used in the calculation of the effective permeability. One of the significant requirements in the design of a finite element structural analysis program is the ability to store, retrieve, and process data that may be complex and varied. To users of such programs, it is important not only to have powerful numerical homogenization solvers, but also to work with a 'user-friendly' graphical interface. On the other hand, realistic problems have grown in size and complexity. Consequently, the size and complexity of computer codes have also grown. The development of reusable, modular, platform-independent computer code is prudent in any programming strategy.

The primary goal of this paper is to illustrate the practical application of an Object-Oriented Programming (OOP) approach in the development and implementation of upscaling techniques. Development of user-friendly and flexible scientific programs is a key to their usage, extension, and maintenance. This paper presents an OOP-based computational environment for upscaling programs. General organization of the developed software system, called JHomogenizer, is given, which includes the solver and the pre/postprocessors with a user-friendly graphical user interface (GUI). Examples with graphical visualization of results are used to illustrate functionality of the program. The software is freely available for research and educational purposes. The open architecture of the program facilitates further developments and adapts to suit specific needs easily and quickly. Moreover, the proposed user interface has proved to be satisfactory and flexible. This paper presents a brief description of the JHomogenizer application. It includes five functional modules to compute effective permeability and a simple set of objects for simulation of flow in porous media.

The outline of the rest of the paper is as follows. In the next section, we present a short description of the methods used for computing the effective permeability of a heterogeneous region. The methods require the numerical solution of partial differential equations. The first three upscaling techniques are based on solving a linear second-order elliptic equation in divergence form (Darcy's equation) and subject to some specific boundary conditions. The first of these involves periodic boundary conditions, the second linear boundary conditions, and the third confined boundary conditions. The effective permeability tensor is itself symmetric and positive definite, just as in the fine scale. Even if the original permeability tensor is diagonal, the effective permeability is, in general, a full tensor. The off-diagonal terms of the effective permeability tensor are significant in many cases. The fourth upscaling technique deals with the so-called double porosity model for describing single-phase flows in fractured porous media (see, e.g., [4]). The approximation of the homogenized tensor require the numerical resolution of local problems in the fracture domain with periodic boundary conditions on the exterior boundary and a no-flow Neumann condition on the interface between the matrix and the fracture blocks. A standard conforming finite element method [18] is used for the solutions of the first and the fourth methods. We use conforming and mixed finite elements (see, e.g., [17, 18, 31]) for the calculation of the effective coefficients obtained in the second and third methods. The fifth approach implemented in the software involves the use of a wavelet characterization for the computation of effective parameters as described in [24, 27, 28]. The fast wavelet transforms developed make the computation more efficient. Section 3 contains a brief description of the JHomogenizer tool. We explain the basic structure of the software and how new solvers/models can be added to the interface. Although large parts of JHomogenizer are documented here, this paper is not a manual or tutorial but rather a description of the choices we made to build the software. We would like to point out that one of our purposes is to explore the use of Java for providing an efficient open source library for education and research. A selection of numerical examples are presented in Section 4. We give the results of computation of the effective permeability by various methods and numerical simulations for miscible and immiscible two-phase flow comparing fine-grid simulations in heterogeneous media with simulations in their homogeneous (effective) counterparts. Finally, conclusions and future developments are discussed in Section 5.

2 Numerical homogenization techniques for flow in porous media

In this section, we will present five numerical homogenization techniques that are implemented in JHomogenizer. In Sections 2.1, 2.2, and 2.3, we consider first incompressible two-phase flow, and in Sections 2.4 and 2.5, one-phase flow in a fractured medium.

We consider first the model problem of incompressible two-phase flow in moderately heterogeneous media. For instance, in [12], using homogenization theory, the case where the phase permeabilities and the capillary pressures are identical in all parts of the medium, i.e., a single rock-type model, was investigated. It was then shown that the homogenized model has the same form as the initial model and then defining effective parameters makes sense. These methods are based on computing the average of either the energy or the flux on a representative elementary volume (REV) with some boundary conditions, e.g., periodic, Dirichlet, or Neumann boundary conditions. These approximation procedures are widely used in the engineering literature (see for instance [30]). We refer to [16] for a rigorous mathematical justification for the convergence of these methods used to approximate the effective tensor of random stationary media. The estimate of the rates of these approximations is also given. The methods differ mostly by the choice of the conditions given on the boundary of the domain to be homogenized.

2.1 Periodic boundary conditions

Let $\Omega \subset I\!R^d$, $d = 1, 2$, or 3, be a bounded domain with a periodic structure. More precisely, we shall scale this periodic structure by a parameter ε that represents the ratio of the cell size to the size of the whole region Ω, and we assume that $0 < \varepsilon \ll 1$ in a decreasing sequence tending to zero. Let $Y = \prod_{i=1}^{d}]0, y_i[$ represent the microscopic domain of the basic cell. Assume that, in such a configuration, the absolute permeability tensor depends only on the microscopic variable $y = x/\varepsilon$ where x is the variable in the macroscopic scale. Namely, $K^\varepsilon(x) = K(x/\varepsilon)$ with K a Y-periodic function in y. Assume that K is a symmetric, strictly positive definite, tensor. Then K_p^*, the effective permeability, is given by

$$(K_p^*)_{ij} = \frac{1}{|Y|} \int_Y K(y) \left[\nabla w_i + \vec{\mathbf{e}}_i\right] \cdot \left[\nabla w_j + \vec{\mathbf{e}}_j\right] dy \qquad 1 \le i, j \le d \tag{2.1}$$

with w_j, $j = 1, ..., d$, the solution of the so-called local or cell problem defined by:

$$\begin{cases} w_j \in H_p^1(Y) / I\!R \\ -\nabla \cdot \left[K(y)(\nabla w_j + \vec{\mathbf{e}}_j)\right] = 0 \quad \text{in} \quad Y \end{cases} \tag{2.2}$$

Here $\vec{\mathbf{e}}_j$ is the j^{th} standard basis vector of $I\!R^d$. We denote by $C_{\text{p}}^\infty(Y)$ the space of infinitely differentiable functions in $I\!R^d$ that are periodic of period Y. Then $H_{\text{p}}^1(Y)$ is the completion of $C_{\text{p}}^\infty(Y)$ for the norm of $H^1(Y)$. A variational formulation of problem (2.2) is

$$\begin{cases} w_j \in H_p^1(Y) / I\!R \\ \int_Y K(y) \nabla w_j \cdot \nabla v \, dy = -\int_Y K(y) \vec{\mathbf{e}}_j \cdot \nabla v \, dy \\ \qquad\qquad \forall v \in H_p^1(Y) / I\!R. \end{cases} \tag{2.3}$$

This problem has a unique solution (cf. [11]). In JHomogenizer, the computation of w_j is performed by a conforming finite element method. A classical Q_1 interpolation yields an approximation to the space $H_p^1(Y) / I\!R$ (cf. [18]); for more details, see [1]. The problem reduces to $d = 2, 3$ computational linear algebra problems with the same matrix; only the right-hand sides differ in the linear systems.

2.2 Linear boundary conditions

In this section, we outline the homogenization method used for the determination of the effective permeability of heterogeneous reservoir regions without any periodic assumption on the microstructure. In the engineering literature, this method is known as the REV method [10] and could be seen as the stochastic homogenization of a stationary and ergodic random field (see [14] and the references therein). The limit as the "volume of homogenization" (coarse grid block) tends to infinity is exactly the homogenized limit in a stationary and ergodic random field (see [6, 7]). In the multi-dimensional case, to compute the effective permeability tensor, K_l^*, we have to solve the local problems for $j = 1, ..., d$

$$\begin{cases} -\nabla \cdot \left[K(y) \nabla p_j\right] = 0 \quad \text{in} \quad Y \\ p_j = y_j \quad \text{on} \quad \partial Y \end{cases} \tag{2.4}$$

where y_j is the j^{th} coordinate.

Solving these local problems gives the following expression for the coefficients of the tensor K_l^*:

$$(K_l^*)_{ij} = \frac{1}{|Y|} \int_Y K(y) \nabla p_i \cdot \nabla p_j \, dy \qquad 1 \le i, j \le d. \tag{2.5}$$

In JHomogenizer, we use both conforming and mixed finite elements to solve the local problems (2.4) and compute approximations from above and below of the effective permeability, respectively (cf. [3]). Linear boundary conditions are convenient, particularly in complicated geometries with nonrectangular coarse grid cells.

Remark 2.1 Note that the equation governing the local problems is the same in the periodic and linear boundary homogenization method, but the boundary conditions are different. The local problem (2.4) could be written in the form:

$$\begin{cases} -\nabla \cdot \left[K(y) \left(\nabla w_j + \vec{\mathbf{e}}_j\right)\right] = 0 \quad \text{in} \quad Y \\ w_j = 0 \quad \text{on} \quad \partial Y \end{cases} \tag{2.6}$$

where $w_j = p_j - y_j$. Then

$$(K_l^*)_{ij} = \frac{1}{|Y|}\int_Y K(y)\left[\nabla w_i + \vec{\mathbf{e}}_i\right]\cdot\left[\nabla w_j + \vec{\mathbf{e}}_j\right]dy \quad 1 \le i, j \le d. \tag{2.7}$$

The approach used here has the advantage of being applicable to realistic complex heterogeneous reservoirs with unstructured grids.

2.3 Confined boundary conditions

This technique considers each coarse grid cell separately and performs three independent flow problems with no-flow boundary conditions on four sides of the cell and constant pressure conditions on two opposing faces. Consider a domain containing fine-scale microscopic grid blocks, as shown in figure 1. The effective permeability tensor, K_c^*, is given for $i = 1, ..., d$ by:

$$(K_c^*)_{ii} = \frac{1}{|Y|}\int_Y \mathbf{K}(y)\nabla w_i \cdot \vec{\mathbf{e}}_i\, dy,\ 1 \le i \le d, \tag{2.8}$$

with $w_i,\ i = 1, ..., d$, the solution of the local problem defined by:

$$\begin{cases} -\nabla\cdot\left[K(y)\nabla w_i\right] = 0 & \text{in} \quad Y \\ K(y)\nabla w_i \cdot \vec{\nu} = 0 & \text{on} \quad \mathcal{S}_i \\ w_i = y_i & \text{on} \quad \partial Y\backslash\mathcal{S}_i. \end{cases} \tag{2.9}$$

where $\mathcal{S}_i$ is a union of faces of the block Y parallel to y_i axis and $\vec{\nu}$ is the outward normal to $\partial\mathcal{S}_i$. Again, both conforming and mixed finite elements methods may be used to solve the local problems (2.9) with JHomogenizer. Note that this technique leads to a diagonal effective permeability tensor.

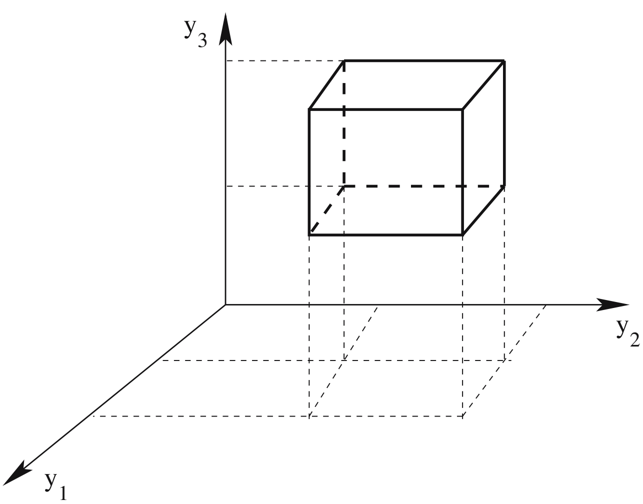

Figure 1 A representative volume element Y.

Remark 2.2 It is well known that the effective permeability is between the harmonic and the arithmetic averages. Furthermore, using the minimization problems associated to the variational formulation of the local problems (2.2), (2.4), and (2.9), it is easy to show, see for instance [3], that we have, for $i = 1, ..., d$: $(K_c^*)_{ii} \le (K_l^*)_{ii}$ and $(K_p^*)_{ii} \le (K_l^*)_{ii}$.

2.4 A fractured porous medium

This section is devoted to computing effective permeability for a double-porosity model describing single-phase flows in a fractured porous medium. We consider a periodic porous medium where the rescaled unit cell Y is made of two complementary parts, the matrix block Y_m and the fracture set Y_f. The matrix block is assumed to be completely surrounded by the fracture set, i.e., Y_m is strictly included in Y (see figure 4). One of the main features of such a model is to have a high contrast ratio between the permeability tensor in the matrix region and in the fissures system through/around the matrix, leading to a high contrast for the corresponding characteristic times. The homogenized global model is also of parabolic type with a nonlocal term which could be seen as a source term or as a time delay (see for instance [4, 5]) with effective permeability defined by:

$$(K_f^*)_{ij} = \frac{1}{|Y|}\int_{Y_f} K(y)\left[\nabla w_i + \vec{\mathbf{e}}_i\right]\cdot\left[\nabla w_j + \vec{\mathbf{e}}_j\right]dy \quad 1 \le i, j \le d \tag{2.10}$$

where $w_j,\ \ j = 1, ..., d$, is the unique solution of the following cell problem:

$$\begin{cases} w_j \in H_p^1(Y)\,/\,I\!R \\ -\nabla\cdot\left[K(y)(\nabla w_j + \vec{\mathbf{e}}_j)\right] = 0 & \text{in} \quad Y_f \\ \left[K(y)(\nabla w_j + \vec{\mathbf{e}}_j)\right]\cdot\vec{\nu} \quad = 0 & \text{in} \quad \partial Y_f. \end{cases} \tag{2.11}$$

As in the previous section, the effective permeability tensor is determined by solving the local problems (2.11) via a conforming finite elements method.

2.5 Wavelets and homogenization

In this section, a brief description of an analogy between homogenization and wavelet representation will be given. More details of the analogy are given in [24, 27, 28]. These papers show how to characterize standard homogenization via wavelet representation. The wavelet-based methods currently included with the JHomogenizer tool are based on the wavelet analogy described here and in the references.

The analogy will be illustrated in one dimension for ease of presentation. Assume that the coefficient $k(x)$ for the elliptic problem

$$\frac{d}{dx}k(x)\frac{dh}{dx} = f, \quad x \in]0,1[$$

with appropriate boundary conditions is a piecewise constant function. Also assume that the coefficient function is defined on 2^m equally sized subintervals of the entire domain. One might imagine that some function $k(x)$ is sampled at 2^m equally spaced points. The idea is to develop a transform method that can be used to compute the correct homogenized value for $k(x)$.

The next step is to compute the solution of a local problem using two neighboring samples of $k(x)$; for example, we may choose to solve for $j = 1, 2, \ldots, 2^{m-1}$ the local problems

$$\frac{d}{dy}k_j(y)\frac{dw_j}{dy} = -\frac{d}{dy}k_j(y)$$

with

$$k_j(y) = \begin{cases} k_{2j} & , \quad 0 \leq y \leq 1/2 \\ k_{2j+1} & , \quad 1/2 \leq y \leq 1 \\ 0 & , \quad \text{otherwise} \end{cases}$$

and periodic boundary conditions, $w_j(0) = w_j(1) = 0$. This definition gives a total of 2^{m-1} local problems to solve. Once the problems have been solved, the homogenized value for a pair can be computed using

$$k_j^{\#} = \int_0^1 k_j(y)\left(1 + \frac{dw_j}{dy}\right)dy.$$

It pays to define level estimates

$$k_{l,j}(y) = \begin{cases} k_{l,2j} & , \quad 0 \leq y \leq 1/2 \\ k_{l,2j+1} & , \quad 1/2 \leq y \leq 1 \\ 0 & , \quad \text{otherwise} \end{cases}$$

and

$$k_{l-1,j} = \int_0^1 k_{l,j}(y)\left(1 + \frac{dw_{l,j}}{dy}\right)dy.$$

With these definitions, it is not a difficult task to develop a fast wavelet-based transform for computing a homogenized value for the entire region as defined in [24, 27, 28].

To do this in a computationally effective way, we would need to know the solutions of the local problems. Fortunately, in one dimension, the local problem defined above admits a solution of the form

$$w_{l,j}(y) = \frac{k_{l,2j+1} - k_{l,2j}}{k_{l,2j} + k_{l,2j+1}} \begin{cases} y & , \quad 0 \leq y \leq 1/2 \\ 1 - y & , \quad 1/2 \leq y \leq 1 \\ 0 & , \quad \text{otherwise} \end{cases}$$

with piecewise derivative given by

$$\frac{d}{dy}w_{l,j}(y) = \frac{k_{l,2j+1} - k_{l,2j}}{k_{l,2j} + k_{l,2j+1}} \begin{cases} 1 & , \quad 0 \leq y \leq 1/2 \\ -1 & , \quad 1/2 \leq y \leq 1 \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{2.12}$$

For those familiar with wavelets, it is easy to see that the derivative of the solution (a piecewise constant function) is a scaled Haar wavelet. The scaling is a nonlinear combination of the two neighboring sample values or homogenized values from the previous level.

One should note that the wavelet characterization is based on the solution of the local problem that results from performing the perturbation analysis in the homogenization procedure. Thus, the wavelet characterization will change as the homogenization method changes.

The extension to multiple dimensions is conceptually very easy. Instead of solving the local problem defined on two cells in one dimension, the local problem must be defined on multidimensional analogs of the one-dimensional work shown above. For the two-dimensional case, an analytic solution for a local problem defined on a 2×2 set of cells must be computed. Once the analytic solution of this problem is obtained, a fast wavelet transform can be created using the analogous averaging formula for multidimensional problems. Details for the multidimensional cases are contained in [24, 28].

3 A brief description of the JHomogenizer tool

The JHomogenizer tool provides a computational environment for the comparison of homogenization methods. The GUI is shown in figure 2. The GUI is written in Java to provide some level of platform independence and interacts with Java objects that implement different methods of homogenization. Some averaging methods are implemented directly into Java objects, whereas other methods are implemented in native codes (e.g, f77, f90, and C) and connected to Java objects through objects that spawn external processes that run the native code. The averaging objects are passed a data set and then turned loose on the data to compute an average value based on the homogenization method employed by the object. Using multithreading within Java, the averaging object spawns a thread to compute the effective value in parallel to other computations being done within the GUI.

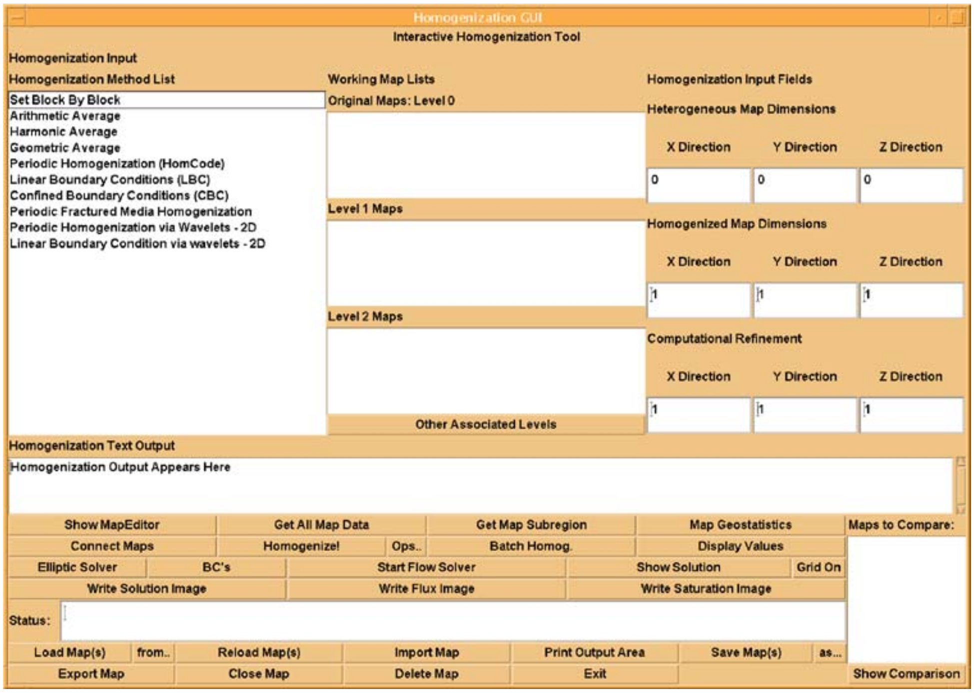

Figure 2 The main JHomogenizer graphical user interface as it appears on the screen.

For the most part, the package is self-contained. The software can be downloaded (see below), compiled, and run with a minor amount of work. There are some parts of the package (codes for some averaging methods) of the distribution that are written in various native language (e.g., Fortran 77, Fortran 90) that need to be compiled if the user wants to use the homogenization method implemented in the native code. This means that to use these methods, the user must have appropriate compilers on their local machines to compile the code to create executables that can be accessed by the interface. A suite of executables is being assembled for various platforms, such as Linux and Windows platforms. These can be downloaded and tried, but there is no guarantee that these will function on every computer. Native codes should be compiled into a static image as discussed in [26].

3.1 Methods of homogenization implemented

The tool has a number of homogenization methods implemented in either Java or some native programming language (e.g., Fortran 77 and C/C++) that can be used directly in the tool. When the interface is started, the methods that are available will appear in a list in the upper left-hand portion of the application as shown in figure 2. The list of methods at the time of the writing of this paper are contained in table 1.

Each of the methods has been tested to make sure that the appropriate input and output works. Results from these tests can be found at [25]. Over time, more methods will be incorporated into the package for comparison purposes. It is hoped that contributors will submit their own methods for inclusion in the package in future releases of the package.

Table 1 Methods currently implemented in the JHomogenizer package.

Method name	Language	Description
Arithmetic average	Java	
Harmonic average	Java	
Geometric average	Java	
Periodic homogenization (HomCode)	Fortran 77	Section 2.1
Linear boundary conditions (LBC)	Fortran 90	Section 2.2
Confined boundary conditions (CBC)	Fortran 90	Section 2.3
Periodic fractured media homogenization	Fortran 77	Section 2.4
Periodic homogenization via wavelets - 2d	Java	Section 2.5
Linear boundary condition via wavelets - 2d	Java	Section 2.5

3.1.1 Adding averaging objects

Users of the JHomogenizer tool can add averaging objects to the JHomogenizer tool that are written in Java or in other languages. The details of how to add averaging objects are presented in the JHomogenizer User Guide [26] that can be downloaded from the Web site [25].

If a user would like to implement a homogenization method in a Java averaging object, the work is not too difficult as long as the user is familiar with the C syntax. An abstract object called AverageObject has been created and is the abstract interface needed to create an official JHomogenizer object. However, it is much easier to make a copy of an existing averaging object (e.g., ArithmeticAverageObject in the file ArithmeticAverageObject.java) and make appropriate modifications of the copy. The name of the file must match the name of the new averaging object. For example, if one implemented a new method named BestFitMethod, then the file name should be BestFitMethod.java. One could copy ArithmeticAverageObject.java changed to BestFitMethod.java using the appropriate command on their computer. The main chore after this is to modify the computeValue() method in the new file so that the desired homogenization method is implemented. Note that if this method is chosen, the object can be compiled and executed on a test problem as a standalone java application before the method is included in the JHomogenizer tool.

To implement a native code object, the work is slightly more complicated. The user will need to match the input and output structure as described in [26]. The interface will produce a file in a specific format and expects the output from the native code in the same format. Once the input/output formats match, a java object must be created to communicate between the native code and the interface. The details of this construction are given in [26].

3.2 MapEditor tool

There are two basic means for getting data sets into the interface. The first is accomplished by importing a map using the interface. The map must adhere to the input specifications as described in [26]. The other method is to create a map using the MapEditor tool. This tool allows the user to specify parameters that define a map such as number of blocks in the coordinate directions, the background map value, and so on. A few tools for creating “realistic” permeability maps have been included in the JHomogenizer tool (e.g, a Kriging option exists).

3.3 Map hierarchy

The interface implements a tree-like hierarchy in dealing with the maps. When a new map is loaded into the interface or created by the MapEditor tools, it is treated as a level 0 map and will appear in the topmost list of maps in the interface. Any maps created via homogenization or using the elliptic solver embedded in the interface are considered associated maps and appear in lists below the level 0 maps. For example, if an original heterogeneous map is imported into the interface and a homogenization is performed, the output of the homogenization will be a map that is associated with the original. The new map will appear in the level 1 list of maps in the interface.

If the user saves the work performed on a map, the original map and all associated maps are saved. The next time the original map is loaded into the interface, all associated maps are also loaded into the interface and appear in the map lists in the interface. This allows the user the ability to go back and continue working on all maps in the data structure at a later time. In particular, as more homogenization methods are added to the interface, a user will be able to go back to previous work and compare new methods to prior results without recomputing previous results.

3.4 Performing homogenization on a map

Once a map has been loaded into the interface, the steps to compute a homogenized version of the data are few. First, one clicks on the name of the map in the lists that appear in the interface. Next, the user specifies the number of homogenized blocks that are desired in the output file. This refers to the number of blocks in each coordinate direction the user wants in the output (homogenized/coarse) map. For example, if the initial map is $100 \times 100 \times 2$ and the user wants a coarser scale map with dimensions $10 \times 10 \times 1$, then the user needs to enter the last set of dimensions into the interface in the given text fields. These are the text fields below the “Homogeneous Map Dimensions” label. Finally, the user clicks on a method of homogenization in the list of available methods in the upper left-hand corner of the GUI and then clicks on **Homogenize!** button.

As the calculations progress, output will be sent to a text window on the screen. If the homogenization is successful, the user will be prompted for a new map name for the resulting homogenized map, and the homogenized map will be added as a map associated with the original finer scale map. This map is, in turn, available for analysis. In addition, the new map can be exported to a file to be used as input to other programs.

3.5 Elliptic solution comparison and flow simulation

It is important to have some idea of how various homogenization methods produce different maps. A simple set of objects for computing solutions for flow in porous media has been implemented. The first object, EllipticRectMFEObject, computes the solution of the flow equation (Darcy's law) where the permeability is defined by the values in any map file currently in the GUI. To compute the approximation solution of this equation for a given map, all one needs to do is to select the map from the lists of available maps and click on the **Elliptic Solver** button. The equation is solved approximately via a lowest order mixed finite element method as implemented in [23].

Once the elliptic problem has been solved, transport through porous media can be simulated. A second object, UpwindSaturationObject, that uses simple upwinding to approximate the solution of the transport equation is available. One may think of this as a simulation of a linear flood in a two-phase immiscible displacement problem. The upwinding produces some level of numerical diffusion. However, the simple simulation gives some idea of the effect of the homogenization on a typical flow problem, albeit very simple.

3.6 Other useful features

During the development of the JHomogenizer tool, a number of useful features have been added to the GUI. Some of these features have been added to the JHomogenizer GUI, and some have been added to the MapEditor tool. The following lists document some, but not all, of these additional features.

1. The elliptic solver allows for the user to select from point source forcing and a selection of simple boundary conditions. This allows the user to visualize flow through a porous medium in several directions.
2. Maps of appropriate dimension can be connected together. This has been helpful in computing homogenized values in subregions and then creating domains that connect the homogenized values in the subregions together again. Examples that use this feature can be found in [25].
3. For those who would like to upscale a number of files of the same size, a batch homogenization feature was added. Problems that require homogenization of a number of stochastic realizations of the same map would certainly benefit from this feature. All the user needs to do is to identify the folder where the realizations reside, and the JHomogenizer tool will do the rest.
4. The GUI allows the user to save images into files. Currently, this is done in a PPM format. However, in the near future, a GIF89 encoding will be included.
5. The GUI allows the user to compare up to three different flow results at once. This allows for the comparison of up to three homogenization methods at the same time.
6. In most cases, people will want to document results. To this end, a feature has been added to the MapEditor tool that will write the map file data in HTML and latex formats.
7. The MapEditor tool has a number of canned methods for generating specific maps. There are options to create uniform, normal, log-normal, and fracture maps with user-defined properties. In addition, options for kriging and other stochastically generated maps are included.

It is not practical to include all details of the features included in the JHomogenizer tool. The details can be found in [26].

4 Numerical examples

In this section, a number of standard idealized and realistic benchmark problems are presented to show how the JHomogenizer tool can be used in testing and comparing homogenization methods. Each section describes results from a single test problem along with the results from homogenization methods. The first two sections contain standard periodic test problems using a symmetric cell and the inverted-L cell (see [2]) and are included for completeness. The rest of the sections give results for more realistic problems from porous media flow and one example of a heat diffusion problem wherein results related to optimal design for an optimized heating element for a microarray.

Each section gives a selection of results associated with using various homogenization methods on the test problems. It is not practical to present all results for all problems. Therefore, a set of results will be given to demonstrate how the JHomogenizer tool can be used to understand the use of homogenization methods. The steps used within the JHomogenizer tool can be found at [25]. Results found on this Web site include

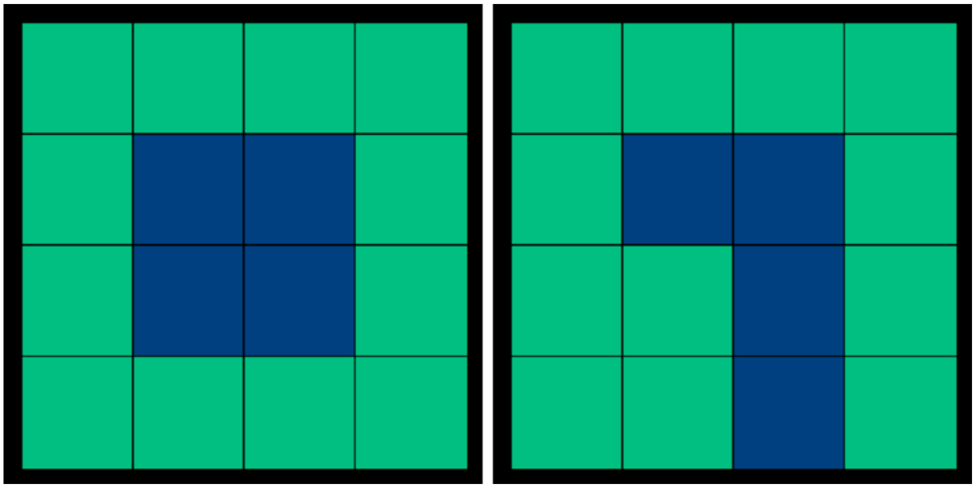

Figure 3 Two examples of local cells used to test homogenization methods. On the left is a simple symmetric cell that should produce an isotropic homogenized tensor. On the right is the so-called 'inverted-L' pattern and is about the simplest example that results in an anisotropic homogenized tensor.

map files that can be used as input to various flow simulators the user might want to use. A more thorough documentation of the results presented in this paper can also be found at [25].

Examples of the simulations of simple two-phase flow are given for some of the example problems. The simulations are approximations of simple flow problems using standard upwind methods. The graphics include plots of the solution of the elliptic problems and flow problems where appropriate. These results give some idea of the effect of the homogenization on flow problems in porous media.

It should be noted that all of the results presented here can be found in [25] along with all the associated data files and software. The instructions for how to create and homogenize the maps to create the results in this paper can also be found there. The JHomogenizer User's Guide also can be used to learn how to use the software.

4.1 Simple periodic examples

In testing any homogenization method, it pays to make sure that the method is behaving correctly. For example, any computational method must be able to return the constant input map value if the heterogeneous map value is constant. As another simple test, any method should produce the harmonic average of the heterogeneous parameter in one dimension. In two (or more) dimensions, any method should also produce a diagonal tensor for stratified maps with the diagonal entries associated with directions parallel to the stratification being equal to the arithmetic average and the diagonal entries associated with coordinate directions perpendicular to the stratification equal to the harmonic average. One such case is documented in [25]. Moving to more complex problems, one should make sure that simple periodically repeated patterns are being averaged correctly. This section describes the

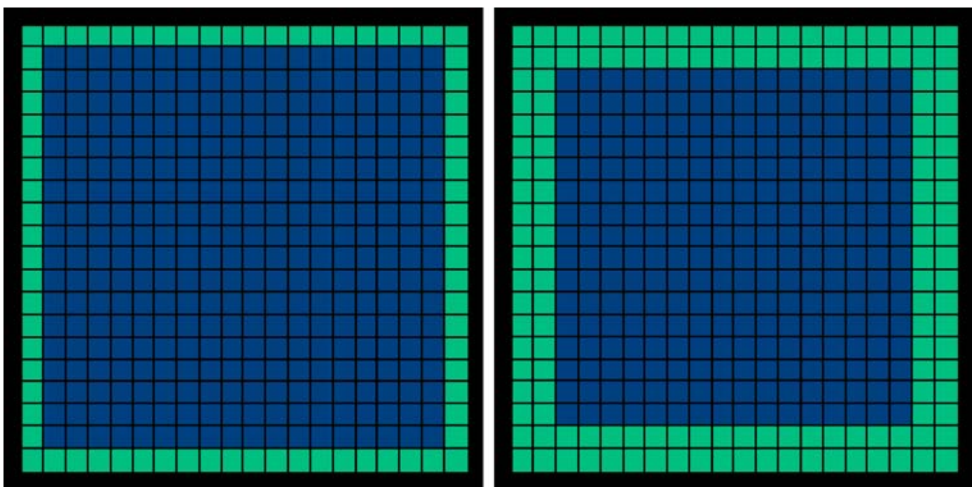

Figure 4 Two examples of idealized fractures. On the left is a rectangular model of a fracture where the fracture zone around the border is 5% of the total size of the cell. The figure on the right is an idealization with a fracture width of 10% of total cell size.

Table 2 Effective permeability values computed by the different upscaling methods for the symmetric cell example with coefficient ratio of 10:1.

Computational method	K_{xx}	K_{xy}	K_{yy}
Arithmetic average	7.75	0.00	7.75
LBC homogenization **CFE**	7.29	0.00	7.29
LBC homogenization **MFE**	7.28	0.00	7.28
Periodic homogenization **CFE**	6.52	0.00	6.52
Confined boundary conditions **CFE**	6.48	0.00	6.48
Confined boundary conditions **MFE**	6.47	0.00	6.47
Geometric average	5.62	0.00	5.62
Harmonic average	3.08	0.00	3.08

Table 3 Effective permeability values computed by the different upscaling methods for the inverted-L example with coefficient ratio of 10:1.

Computational method	K_{xx}	K_{xy}	K_{yy}
Arithmetic average	7.750	0.000	7.750
LBC homogenization **CFE**	5.752	-0.343	6.834
LBC homogenization **MFE**	5.733	-0.343	6.817
Periodic homogenization **CFE**	5.332	-0.286	6.761
Confined boundary conditions **CFE**	4.888	0.000	6.792
Confined boundary conditions **MFE**	4.876	0.000	6.777
Geometric average	5.620	0.000	5.620
Harmonic average	3.077	0.000	3.077

homogenization of two simple examples: one that is geometrically symmetric and another example that is not geometrically symmetric.

4.1.1 Periodic symmetric cell

The symmetric cell is depicted in figure 3 (left). This problem provides a sanity test for any homogenization method including those implemented in the application. One can easily compute the arithmetic and harmonic averages for the symmetric cell and use these as upper and lower bounds, respectively, for the homogenization results. In addition, the output tensor for the symmetric cell should be a diagonal tensor, unlike the inverted-L cell discussed next and also shown in figure 3. The reader should note that the cells depicted in figure 3 are assumed to be periodically repeated many times in the domain defining the porous medium. In fact, the assumption is that the size of the cell containing the symmetric pattern shown in figure 3 is much smaller than the size of the domain of interest.

Ratios of the coefficient values of 10:1, 100:1, and 1,000:1 were chosen with the larger coefficient value in the border region of the cell and the lower value in the interior region. The results for the various homogenization methods for the 10:1 ratio are given in table 2. These results and the results for the other ratios can be found in [2]. It is not practical to include all the results here. Instead, we refer the reader to [25] for a more thorough set of results. We note CFE (MFE) when the conforming (mixed) finite element method is used to solve local problems.

*4.1.2 Periodic inverted-*L *cell*

The inverted-L pattern shown in figure 3 (right) is about the simplest nonsymmetric test problem one can think of. The reader should note that the cell is repeated periodically with the cell in figure 3 much smaller than the domain of interest. The homogenization methods are applied to the local problem on a single cell. In this case, the output of homogenization may produce a symmetric tensor with nonzero entries off the main diagonal (see table 3). As in the symmetric cell example discussed previously, results are presented only for the 10:1 ratio. The coefficient value in the inverted-L-shaped region is lower than in the rest of the cell.

The results of the various homogenization methods are given in table 3. As can be seen, several methods produce tensors with nonzero entries off the diagonal. More results can be found in [25].

4.2 Fracture model cell

The MapEditor tool available in the JHomogenizer interface allows users to set up some specialized maps such as maps with uniformly distributed values. One of the specialized maps involves a model of a fracture via a rectangular region. This looks similar to the periodic symmetric cell previously discussed. However, the difference is that the width of the border around the interior can be varied. The border region models a fracture, whereas the interior region models the matrix. Some of the homogenization methods will produce more reliable results than the others.

Table 4 Effective permeability values computed by the different upscaling methods for the fracture on the left with coefficient ratio of 10:1.

Computational method	K_{xx}	K_{xy}	K_{yy}
Arithmetic average	2.710	0.000	2.710
LBC homogenization	2.031	0.000	2.031
Periodic homogenization	2.021	0.000	2.021
Geometric average	1.549	0.000	1.549
Harmonic average	1.206	0.000	1.206

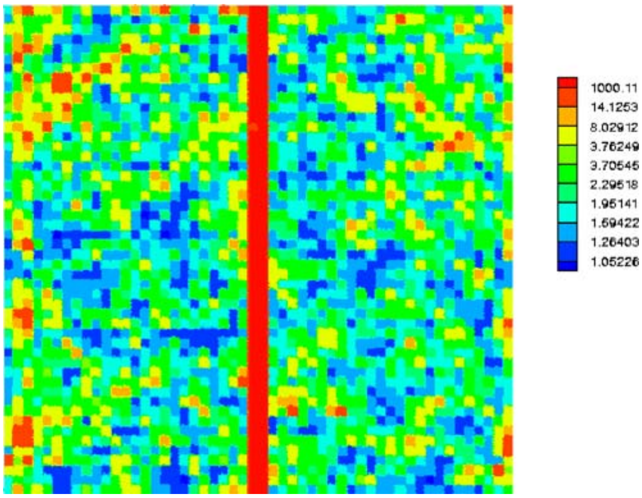

Figure 5 Permeability distribution for a more realistic fractured medium. The vertical band models a fracture zone bounded on the left and right by two lower permeability regions.

Homogenized tensors are presented in table 4 for some of the homogenization methods available in the JHomogenizer tool. The results show a large variation in tensor values.

4.3 A realistic fractured porous media example

In this section, results for a problem developed by Alain Bourgeat (http://www.gdrmomas.org/pageperso/Bourgeat/index.htm) are presented. The original absolute permeability map is shown in figure 5. It is easy to see that a fracture runs vertically through the middle of the region. The regions to the left and right of the fracture have absolute permeability of order 1, and the values in the fracture zone are of order 10^3. Thus, there is a significant contrast between the three regions. In addition, the three regions themselves are heterogeneous.

Part of the problem here is to determine reasonable strategies for computing homogenized coefficient values. One could compute a single coefficient defined on the entire domain. However, the results are disastrous. Saturation contours are shown for the heterogeneous and homogenized maps in figure 6. It is easy to see that the effect of the fracture has been dramatically washed out in the process. All of the local influence of the fracture is lost.

To illustrate a more reasonable approach, homogenization can be performed in the three subregions of the original heterogeneous map. The process divides the original region into three subregions: the fracture region, a region to the left of the fracture, and a region to the right of the fracture. A homogenization method is applied to each of the three regions separately to find a single coefficient value in each region. Finally, the three homogenizer regions are connected back together. Homogenized values for these regions are given in table 5 for various methods. Two saturation contours corresponding to two different types of boundary conditions for homogenization are shown in figure 7. They are almost identical.

An extension of this process is to treat the problem a bit like a boundary layer problem. The idea suggested by A. Bourgeat is to expand the fracture zone to include one (or more) map value(s) from the matrix. In figure 5, the fracture zone includes two grid blocks in the horizontal direction. The grid dimension of the original fracture zone is 2×50. By including one additional grid block to the left and right of the original fracture zone, the modified fractured zone now has a grid dimension of 4×50. Using the modified regions, the left and right matrices can be homogenized while

Figure 6 Left: Saturation contours for a two-phase flow with a permeability map shown in figure 5. Right: Saturation contours obtained with the homogenized map. In this case, the entire map was homogenized using periodic homogenization. The influence of the fracture is lost in the homogenization.

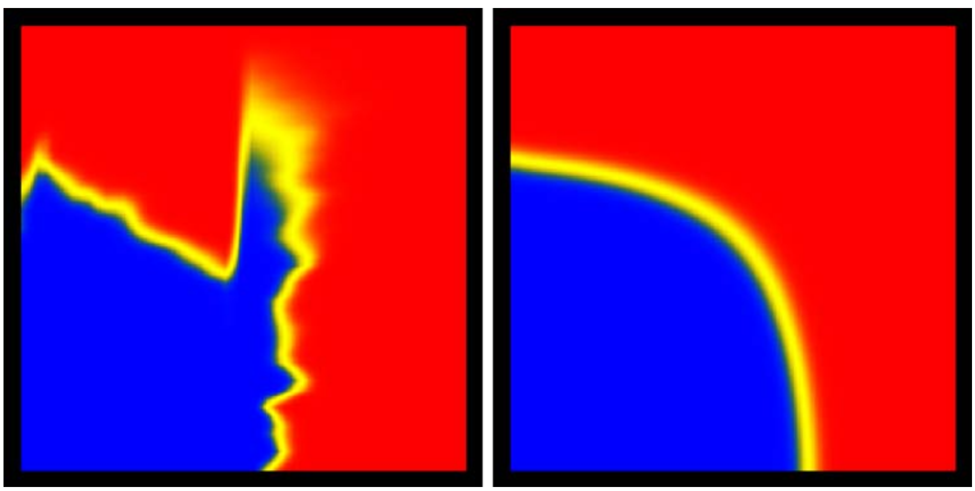

Table 5 Homogenization results for the three regions in a realistic fracture example.

Region homogenized	Homogenization method	Homogenization tensor
Entire region		
	Arithmetic average	$\begin{bmatrix} 41.4974 & 0.0000 \\ 0.0000 & 41.4974 \end{bmatrix}$
	Harmonic average	$\begin{bmatrix} 2.1267 & 0.0000 \\ 0.0000 & 2.1267 \end{bmatrix}$
	Periodic homogenization	$\begin{bmatrix} 2.4707 & 0.0134 \\ 0.0134 & 40.8541 \end{bmatrix}$
	Linear BC homogenization	$\begin{bmatrix} 3.5035 & 0.0059 \\ 0.0059 & 40.9342 \end{bmatrix}$
Left region		
	Arithmetic average	$\begin{bmatrix} 3.3827 & 0.0000 \\ 0.0000 & 3.3827 \end{bmatrix}$
	Harmonic average	$\begin{bmatrix} 2.0822 & 0.0000 \\ 0.0000 & 2.0822 \end{bmatrix}$
	Periodic homogenization	$\begin{bmatrix} 2.5898 & 0.0089 \\ 0.0089 & 2.4213 \end{bmatrix}$
	Linear BC homogenization	$\begin{bmatrix} 2.7398 & 0.0209 \\ 0.0209 & 2.4798 \end{bmatrix}$
Fracture zone		
	Arithmetic average	$\begin{bmatrix} 1001.18 & 0.0000 \\ 0.0000 & 1001.18 \end{bmatrix}$
	Harmonic average	$\begin{bmatrix} 1001.18 & 0.0000 \\ 0.0000 & 1001.18 \end{bmatrix}$
	Periodic homogenization	$\begin{bmatrix} 1001.18 & 0.0000 \\ 0.0000 & 1001.18 \end{bmatrix}$
	Linear BC homogenization	$\begin{bmatrix} 1001.18 & 0.0000 \\ 0.0000 & 1001.18 \end{bmatrix}$
Right region		
	Arithmetic average	$\begin{bmatrix} 2.6072 & 0.0000 \\ 0.0000 & 2.6072 \end{bmatrix}$
	Harmonic average	$\begin{bmatrix} 2.2112 & 0.0000 \\ 0.0000 & 2.2112 \end{bmatrix}$
	Periodic homogenization	$\begin{bmatrix} 2.3185 & -0.0109 \\ -0.0109 & 2.1502 \end{bmatrix}$
	Linear BC homogenization	$\begin{bmatrix} 2.3555 & -0.0109 \\ -0.0109 & 2.1803 \end{bmatrix}$

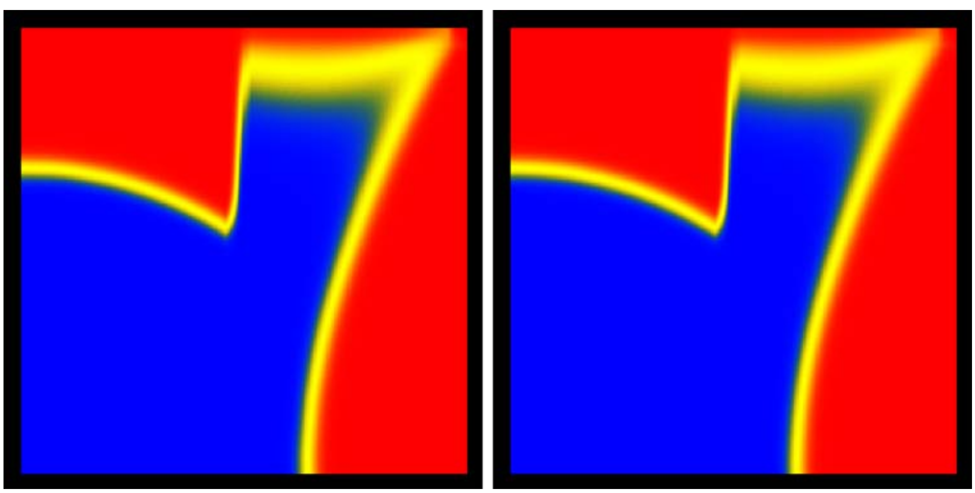

Figure 7 Saturation contours obtained with a homogenized map such that the three subregions are homogenized separately and connected back together. The figure on the left shows results when periodic homogenization method was used, and the figure on the right shows results when the linear boundary condition homogenization method was used.

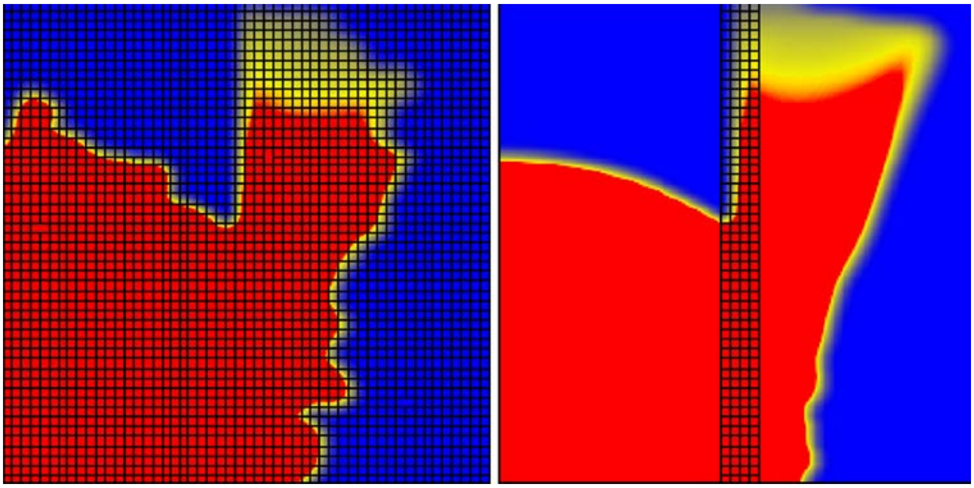

Figure 8 Left: Saturation contours for a two-phase flow with the heterogeneous permeability map shown in figure 5. Right: Saturation contours obtained with a homogenized map where the three subregions are homogenized separately and connected back together.

leaving the fracture zone heterogeneous. The benefit is that the important heterogeneities are maintained, whereas the heterogeneities of lesser importance in the left and right regions are averaged.

Figure 8 shows the flow using this approach for homogenizing the domain. The figure showing the flow through the porous medium resulting from the homogenization shows the grid associated with the modified fracture zone. The advantage of this approach is that a single value can be used in the modified matrix zones leading to a simpler linear algebra problem.

The reader should understand that all of the steps described in the section can easily be performed in the JHomogenizer tool. In addition, the steps for doing this work are outlined in [25]. The JHomogenizer tool allows the user to extract subregions, homogenize these subregions, create appropriate homogenized maps from the homogenization results, and then connect the pieces back together. The output from the various homogenization techniques can be found at the Web site [25].

4.4 Total data example

Total, Pau, France provided the data for the permeability map shown in figure 9 for testing the effect of homogenization in a realistic situation. The map contains three distinct coefficient values of 1.0 (darkest shaded regions), 10.0, and 100.0 (lightest shaded regions) and has higher permeability paths through the domain that influence the flow of fluids in the porous medium. The map is highly heterogeneous. In this test problem, the original heterogeneous map is defined on 30×30 grids. The test involves obtaining a coarser grid map defined on 6×6 grids. The JHomogenizer tool allows the user to set the homogenized or coarse grid size in the interface. This results in the solution with a 6×6 grid of local problems.

As an example of the output, one might expect from the JHomogenizer tool, table 6 shows the results from applying periodic homogenization to the 6×6 grid of subregions defined in the problem. The 36 local problems are solved with periodic boundary conditions. These and other results are documented with more details in [25].

To visualize the effect of the homogenization method on the porous medium, a simulation of two-phase displacement was performed. These simulations

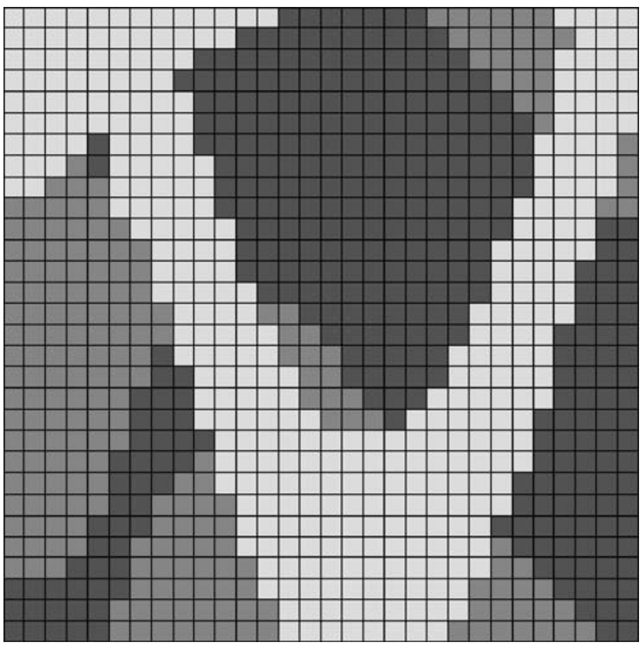

Figure 9 The fine-scale permeability field. The structure is defined on a 30×30 grid of blocks. The homogenized values use 5×5 blocks to end up with a 6×6 coarse scale permeability field.

Table 6 Equivalent permeability values computed for the periodic homogenization method using the conformal finite element method on 5 × 5 blocks of cells in the heterogeneous field. The values for the tensor are K_{xx}, $K_{xy} = K_{yx}$, and K_{yy}, from top to bottom, respectively, for the appropriate cell.

	1	2	3	4	5	6
	10.0000	6.1073	0.1753	0.1000	0.4497	3.4402
6	0.0000	-0.0001	0.0110	0.0000	-0.0416	0.0000
	10.0000	7.2762	0.1391	0.1000	0.4496	7.4172
	3.8292	5.8735	0.1000	0.1000	0.1092	4.7480
5	0.1912	0.3386	0.0000	0.0000	0.0000	0.1795
	2.7202	7.8420	0.1000	0.1000	0.1092	6.9637
	1.0000	2.5889	0.2432	0.1000	0.2015	0.2179
4	0.0000	-0.0787	-0.0101	0.0000	0.0120	0.0001
	1.0000	5.7037	2.3542	0.1000	3.0033	4.3325
	1.0000	0.6353	5.2078	0.2080	5.8155	0.1230
3	0.0000	0.0064	-0.2286	-0.0032	0.4584	0.0000
	1.0000	2.5881	7.1734	0.2126	7.2970	0.2463
	0.9347	0.2081	8.3336	10.0000	5.8156	0.1000
2	0.0000	0.0140	0.0000	0.0000	0.4585	0.0000
	0.9346	0.2735	8.9698	10.0000	7.2969	0.1000
	0.2060	0.7137	2.5058	10.0000	1.6822	0.1597
1	0.0047	0.0000	-0.0754	0.0000	0.0479	-0.0101
	0.1466	0.8640	5.3132	10.0000	3.9697	0.1597

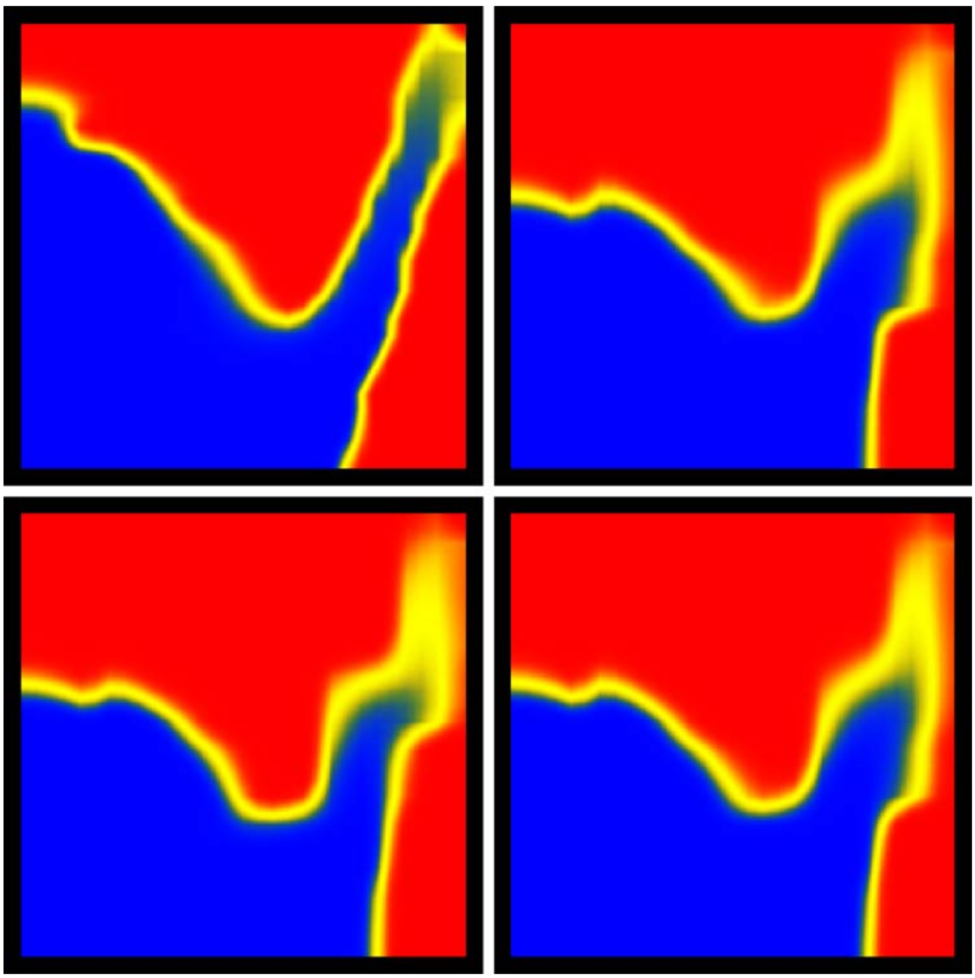

Figure 10 Saturation contours for the Total map data with the original heterogeneous map (upper left-hand corner) and with the medium homogenized using three methods: confined upscaling (upper right), periodic homogenization (lower left), and linearly boundary condition (lower right).

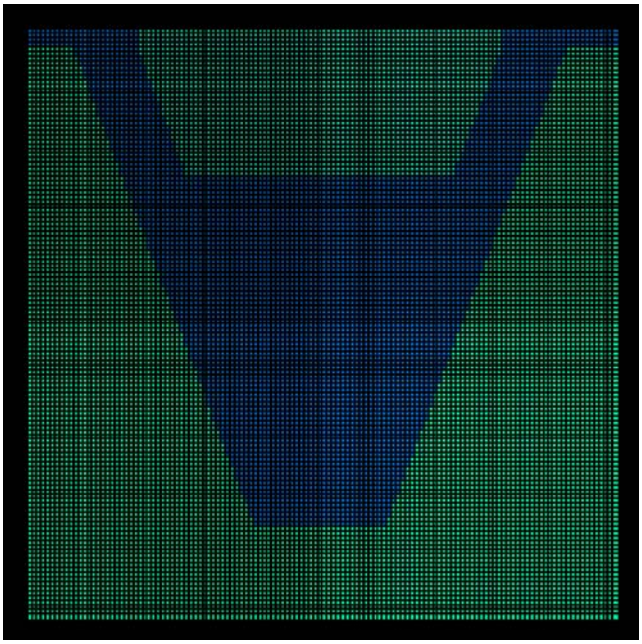

Figure 11 The map data above show the contrasts in the thermal diffusivity in a single microarray cell that might be used in the replication of DNA collected for a variety of applications.

were performed using the simple simulation features in the JHomogenizer. In figure 10, a comparison of the simulations for the original heterogeneous map and homogenized maps are shown. It is clear that the results are not exactly the same, but the general configuration of the saturation contours is the same. There are differences, and the JHomogenizer tool gives users the opportunity to investigate these differences by analyzing data in localized regions.

4.5 Heating element design example

The original purpose for developing the JHomogenizer tool was to test various homogenization methods related to porous media flow simulation. However, the methodology is applicable to any elliptic problem with a heterogeneous coefficient. To demonstrate this, we consider the design of a heating element for a microarray used in DNA replication given (figure 11).

Polymerase chain reaction is a process whereby small amounts of DNA can be duplicated through a series of heating and cooling cycles. A tray of samples is created

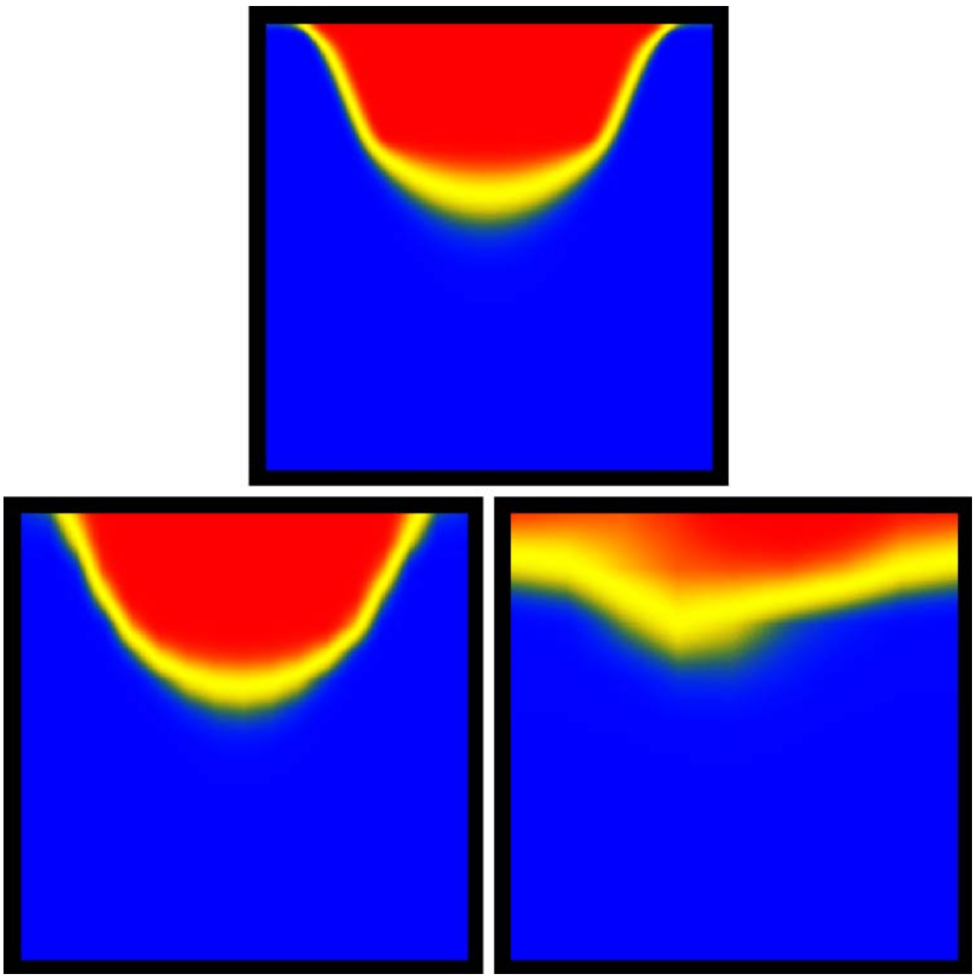

Figure 12 An approximation of the solution of the elliptic problem for the heterogeneous map defined on a 128×128 grid and homogenized maps using periodic homogenization with 16×16 (lower left) and 8×8 (lower right) coarse grids.

in the lab. In turn, the tray is placed in a device that heats the samples within the tray. The heating process should ensure that all samples are heated uniformly and each of the samples in the tray is heated in exactly the same manner. The heat source is contained as a circuit located beneath the sample tray. The heating element only contacts a subset of the lower surface of the sample tray.

One way to proceed would be to define the heat equation on a domain that includes the tray of samples and the device housing the samples during the heating process. This is certainly a heterogeneous media where various parts of the device and tray have different coefficients of thermal diffusivity. To design an appropriate heating element, an optimization problem was defined for the heating process. The criterion to minimize measures the deviation from a mean temperature measured at the center of each sample. The optimization problem necessitated multiple solutions of the heat equation to find an optimal design. If a fine computational mesh were used, the computational effort would be enormous. In addition, finding a solution in the "ball park" would suffice for the company (Idaho Technology, Salt Lake City, UT). Thus, homogenization became a reasonable mean to reduce the computational cost in this project.

Results, extracted from Daniel Balls' PhD thesis [9] and obtained with JHomogenizer, are shown in figure 12. On top of this figure, we show the solution calculated on the original map, the left that on a 16×16 homogenized map, and the right that on a 8×8 homogenized map. Clearly, this last grid is not sufficient for a reasonable calculation.

5 Concluding remarks

In this paper, we presented the program JHomogenizer for numerical homogenization and several examples of problems solved with it. Its open architecture facilitates further developments for specific needs. Moreover, the proposed user interface has proved to be very friendly and flexible. The platform still needs to be improved in several areas such as upscaling permeability for fractured reservoirs, upscaling relative permeabilities and capillary pressure (cf. [13]), and upscaling dispersion in saturated porous media (cf. [15]). Efforts addressing both of these important issues are currently underway.

Acknowledgements The authors would like to thank Prof. Alain Bourgeat for useful discussions. This work has been partially supported from the GdR MoMaS 2439 CNRS ANDRA BRGM CEA EDF whose support is gratefully acknowledged.

References

1. Amaziane, B.: Global behavior of compressible three-phase flow in heterogeneous porous media. Transp. Porous Media **10**, 43–56 (1993)
2. Amaziane, B., Bourgeat, A., Koebbe, J.: Numerical simulation and homogenization of two-phase flow in heterogeneous porous media. Transp. Porous Media **6**, 519–547 (1991)
3. Amaziane, B., Hontans, T., Koebbe, J.: Equivalent permeability and simulation of two-phase flow in heterogeneous porous media. Comput. Geosci. **5**, 279–300 (2001)
4. Arbogast, T., Douglas, J., Hornung, U.: Derivation of the double porosity model of single phase flow via homogenization theory. SIAM J. Math. Anal. **21**, 823–826 (1990)
5. Arbogast, T.: Computational aspects of dual-porosity models. In: Hornung, U. (ed.) Homogenization and Porous Media, pp. 203–223. Springer, Berlin Heidelberg New York (1997)
6. Badea, A., Bourgeat, A.: Homogenization of two phase flow through randomly heterogeneous porous media. In: Bourgeat, A. et al. (eds.) Proceedings of the Conference Mathematical Modelling of Flow Through Porous Media, pp. 44–58. World Scientific, Singapore (1995)
7. Badea, A., Bourgeat, A.: Numerical simulations by homogenization of two-phase flow through randomly heterogeneous porous media. In: Helmig, R. et al. (eds.) Notes on Numerical Fluid Mechanics, Modeling and Computation in Environmental Sciences, vol. 59, pp. 13–24. Vieweg, Wiesbaden (1997)
8. Bakhvalov, N., Panasenko, G.: Homogenisation: Averaging Processes in Periodic Media. Kluwer Academic Publishers, Dordrecht (1989)
9. Balls, D.: Numerical methods and simulation applied to optimize heating designs, Master's Thesis, Utah State University, Logan, Utah (2004)
10. Bear, J., Bachmat, Y.: Introduction to Modeling of Transport Phenomena in Porous Media. Kluwer Academic Publishers, London (1991)
11. Bensoussan, A., Lions, J.L., Papanicolaou, G.: Asymptotic Analysis for Periodic Structures. North-Holland, Amsterdam (1978)
12. Bourgeat, A.: Two-phase flow. In: Hornung, U. (ed.) Homogenization and Porous Media, pp. 95-128. Springer, Berlin Heidelberg New York (1997)
13. Bourgeat, A., Hidani, A.: Effective model of two-phase flow in a porous medium made of different rock types. Appl. Anal. **58**, 1–29 (1995)
14. Bourgeat, A., Kozlov, S.M., Mikelic, A.: Effective equations of two-phase flow in random media. Calc. Var. **3**, 385–406 (1995)
15. Bourgeat, A., Jurak, M., Piatnitski, A.: Averaging a transport equation with small diffusion and oscillating velocity. Math. Methods Appl. Sci. **26**, 95–117 (2003)
16. Bourgeat, A., Piatnitski, A.: Approximation of effective coefficients in stochastic homogenization, Ann. Inst. Henri Poincaré B **40**, 153–165 (2004)
17. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Element Methods. Springer, Berlin Heidelberg New York (1991)
18. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. North-Holland, Amsterdam (1978)
19. Cioranescu, D., Donato, P.: An Introduction to Homogenization. Oxford University Press, Oxford (1999)
20. Farmer, C.L.: Upscaling: A review. Int. J. Numer. Meth. Fluids **40**, 63–78 (2002)
21. Hornung, U.: Homogenization and Porous Media. Interdisciplinary Applied Mathematics. Springer, Berlin Heidelberg New York (1997)

22. Jikov, V.V., Kozlov, S.M., Oleinik, O.A.: Homogenization of Differential Operators and Integral Functionals. Springer, Berlin Heidelberg New York (1994)
23. Koebbe, J.: A computationally efficient modification of mixed finite element methods for flows problems with full transmissivity tensors. Numer. Methods Partial Differ. Equ. **9**, 339–355 (1993)
24. Koebbe, J.: Homogenization-wavelet reconstruction methods for elliptic problems. Numer. Methods Partial Differ. Equ. **100**, 1–35 (2002)
25. Koebbe, J.: Computational Homogenization and Multiscale Research Web Site. (2004) Available online at http://www.math.usu.edu/~koebbe/wwwHomog/
26. Koebbe, J.: JHomogenizer User's Guide. (2004) Available at http://www.math.usu.edu/~koebbe/wwwHomog/
27. Koebbe, J., Thomas, R.: Wavelet construction based on homogenization. In: Proceedings of the XIII International Conference on Computational Methods in Water Resources, Calgary, Alberta, Canada 25–29 June 2000
28. Koebbe, J., Watkins, L., Thomas, R.: Characterization and upscaling of sedimentary depositional formations using archetypal analysis and homogenization. In: Proceedings of the Third International IMACS, Jackson, Wyoming, USA, 9–12 July 1997
29. Panfilov, M.: Macroscale Models for Flow Through Highly Heterogeneous Porous Media. Kluwer Academic, Dordrecht (2000)
30. Renard, P., DeMarsily, G.: Calculating equivalent permeability: A review. Adv. Water Resour. **20**, 253–278 (1997)
31. Roberts, J.E., Thomas, J.M.: Mixed and hybrid methods. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, Finite Element Methods (Part 1), Vol. II, pp. 524–639, North-Holland, Amsterdam (1991)
32. Sanchez-Palencia, E.: Non-Homogeneous Media and Vibration Theory. Springer, Berlin Heidelberg New York (1980)