The Thesis Committee for Anshul Goyal
certifies that this is the approved version of the following thesis:

Parallel Computing using The Multiscale Finite Element Method for Sub-Surface Flow Models

APPROVED BY
SUPERVISING COMMITTEE:

Clint N. Dawson, Supervisor

Howard Liljestrand, Co-Supervisor

Steven Mattis

Parallel Computing using The Multiscale Finite Element Method for Sub-Surface Flow Models

by

Anshul Goyal, B.Tech.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my Mother

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Clint Dawson for his guidance, encouragement, and useful discussions. I enjoyed every bit of the opportunity given to me to work with his excellent research group. I thank, Dr. Howard Liljestrand and Dr. Steven Mattis, for serving on this committee and giving thoughtful advice. My sincere thanks to Dr. Michael Presho for sharing his research and long discussion sessions. I enjoyed working with him and glad that he always encouraged my ideas. I am also thankful to Dr. Arash Fathi, the postdoctoral scholar in Prof. Dawson's group, for his valuable help in running simulations on Lonestar. I acknowledge TACC HPC consulting staff for their invaluable help and resources. I thank Mr. Udit Agarwal, the doctoral student in Dept. of Computer Science at UT Austin, for discussions related to computer programming.

I want to thank my parents, sister and friends for their love and support. Last but not the least, I thank the Almighty for giving me strength and patience to pursue all my endeavors.

Abstract

Parallel Computing using The Multiscale Finite Element Method for Sub-Surface Flow Models

Anshul Goyal, M.S.E.
The University of Texas at Austin, 2016

Supervisor: Clint N. Dawson
Co-Supervisor: Howard Liljestrand

Subsurface flows, occurring in groundwater movement and production of hydrocarbons in the petroleum industry, are affected by the heterogeneity of the medium varying over large scales. In this thesis, we have used state of the art multiscale methods to solve one such flow model, influenced by high contrast permeability field. The focus is on the elliptic pressure equation which is solved on both fine and coarse scale for comparison purposes. Shared memory parallelism has been achieved for generating basis functions, which is computationally the most expensive portion of the multiscale implementation. Parallel systematic spectral enrichment using the GMsFEM (Generalized Multiscale Finite Element Method) is the key feature of the current work and has been compared with the MsFEM (Multiscale Finite Element Method). Efficiency of algorithmic implementation has been first tabulated for a two-dimensional finite element code using the MATLAB parallel computing toolbox and also has

been given a more generalized form using a three-dimensional finite element code written in OpenMP (Open Multiprocessing) and C++. The timing comparison shows a significant decline in the execution time for the algorithms. It indicates that a higher level of enrichment and desired accuracy is achievable for large scale problems. Computational time gain and fewer memory requirements are two key features achieved in this work. Distributed parallel computing can further be implemented to achieve mass parallelism through which one can solve large problems accurately and efficiently when compared to benchmark fine scale solutions where global system solver, memory requirements and execution time become significant issues.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background

1.1 Motivation

Many problems encountered in science and engineering are multiscale in nature. It is this multiscale feature which dominates the simulation efforts while dealing with problems of this nature. A typical example is the analysis of groundwater transport, where the difficulty lies in the heterogeneous subsurface spanning over many scales. A similar problem arises while analyzing composite materials, flow through porous media and turbulent transport in high Reynolds number flow. Even with modern supercomputing facilities, the direct numerical solution of these problems is complicated and requires a tremendous amount of computer memory and CPU time. Parallel computing may be the obvious choice to tackle this, but the size of the discrete problem is not reduced. The workload remains the same, but now being shared by more processors and memory. No doubt, direct solutions provide a complete analysis of the physical process at all scales, however in an engineering context, it is desirable to predict the macroscopic properties of the multiscale systems. These may be effective conductivity and elastic modulus for composite materials and permeability for flow through porous media or groundwater transport.

Such complexities have led to the development of multiscale methods which can capture the fine scale effects on the larger scale without ever fully resolving all the fine scale features. The approach constructs coarse scale solutions accurately and efficiently without solving a global fine scale system. The main idea is to construct the finite element basis functions which capture the fine scale information within each element.

1.2 Scope

While implementing multiscale methods, a significant amount of computational time is required to construct the basis functions. However, the computation of basis functions per coarse node is independent of each other. This allows parallel computing, which distributes the computation and reduces CPU time as more processors are used. Though the operation count of the multiscale method is almost twice that of conventional finite element method, the present work aims to demonstrate the application of parallel computing to reduce overall computation time. Parallel versions of the GMsFEM and the MsFEM have been developed, which have not yet been presented elsewhere as a full comparison study.

Shared memory parallelism is achieved using MATLAB's built-in parallel toolbox for a two-dimensional finite element code. It is subsequently generalized by solving a three-dimensional finite element code using OpenMP in C++. The timing comparisons indicate a declining trend which is encouraging for using these methods to solve large scale problems. Spectral enrichment

using parallel GMsFEM gives an added advantage to achieve the desired level of accuracy which is one of the key features of the current work.

1.3 Thesis Organization

The thesis consists of five chapters. **Chapter 2** provides an introduction to the multiscale methods. A brief literature review is done on earlier proposed methods for solving problems which are multiscale in nature. A short discussion of several variants of multiscale methods is also presented. The chapter ends with a discussion on GMsFEM. **Chapter 3** describes the physical problem and mathematical details. Notable features include the variational statement of the problem and the governing equations for solving the problem numerically using a Galerkin formulation on the fine scale and the MsFEM and the GMsFEM on the coarse scale. **Chapter 4** presents the numerical results which include the solution of an elliptic pressure equation, computing the fluxes and comparing CPU time at fine and coarse scale. **Chapter 5** summarizes the study objectives and highlights the key results and conclusions of this study.

Chapter 2

Introduction

In subsurface flow, the permeability of the porous media may vary over several scales and can be represented as a high contrast heterogeneous coefficient. Such kinds of multiscale problems pose challenges in mathematical modeling and simulation. A direct numerical solution is difficult because of the tremendous amount of computation involved. For instance, in a typical reservoir simulation, even the most detailed models have grid blocks with dimensions of the order of tens of meters, whereas the raw permeability and porosity data are measured on a scale of a few centimeters [29]. Therefore, the first major issue is to upscale the information from the core to the grid blocks for obtaining flow properties. The coarse-grained permeabilities also depend on the pressure boundary conditions on the coarse blocks. Direct numerical solutions are more informative since one can get the complete information about the physical process at all scales. Some of the recent direct solutions for flow and transport in porous media are reported in [2, 32]. However, the goal here is to avoid the computational complexity presented by direct solutions and to obtain accurately coarse scale solutions without resolving the fine scale details completely. This is done by capturing the multiscale structure of the solution via localized basis functions. The essential multiscale information at

the fine level is embedded in basis functions, which is coupled through the global formulation to obtain an approximate solution at the coarse grid.

There are two types of multiscale processes. The first one has scale separation, where small-scale information is captured via local multiscale basis functions within local regions (coarse-scale grid blocks) and the second one does not have clear scale separation; therefore, information at different scales is used for generating multiscale basis function. In the present work, we have used the first approach by constructing coarse grids. In most of the cases, a Representative Volume Element (RVE) is used when the input information about the process or material properties is not available everywhere. In such cases, the RVE contains the essential information about the heterogeneities. Thus, the fine scale information within each RVE can be ingrained in the basis functions. Based on the information available within each RVE there are two approaches to solve the macroscopic system. The first one is fine to coarse, where coarse-scale equations are not formulated explicitly and the characteristic fine scale information is carried throughout the simulation. The other approach is coarse to fine, where coarse scale equations are developed and solved explicitly to obtain coarse scale parameters [20]. The present work uses fine to coarse scale approach.

The next critical component in multiscale problems may be the uncertainty associated with media properties. These uncertainties are usually parameterized and give rise to a large set of permeability fields with multiscale nature. It is yet another motivation to use multiscale methods as direct solu-

tions cannot take into account large number of multiscale permeability fields. In such complex problems, multiscale basis functions contain both spatio-temporal scale information and the associated uncertainties. Thus, from the point of view of uncertainty quantification, the multiscale approach is a feasible choice [9].

2.1 Literature Review

Many numerical techniques have been developed to solve problems with multiple scales. Some of the earlier attempts focused on obtaining the effective properties of the medium using statistical averaging, homogenization theory [8, 11] and upscaling [10, 33]. The techniques for upscaling range from simple averaging of the heterogeneous values within the block to sophisticated inversions. Wen and Hernandez [33] provide a comprehensive overview of the recent studies for upscaling hydraulic conductivity in heterogeneous media. Dykaar and Kitanidis [8] used a numerical spectral technique known as the Fourier Galerkin method to determine the effective hydraulic conductivity in a rigid saturated porous medium. It was also shown that the finite difference method required about four times more nodes than the spectral method to give the same accuracy. Methods based on homogenization have been successfully applied for determining the effective properties in porous media but are difficult in problems containing separate scales since the cost of computation grows exponentially [25].

MsFEM was first introduced in [27] and has similarities with upscaling

methods. The idea behind upscaling is to form coarse scale equations analytically, which may differ from the underlying fine scale equations. However, in multiscale methods, coarse equations are not expressed analytically, but rather formed and solved numerically. One of the earliest descriptions of the multiscale methods can be found in [3, 4], where the authors proposed the use of multiscale basis functions for elliptic equations with a special multiscale coefficient, which is the product of one-dimensional fields. This led to the introduction of the generalized finite element method. The approach was extended in the work of Hou and Wu [25], where the authors implemented the multiscale method for solving elliptic PDEs arising in composite materials and flow through porous media. They also showed that boundary conditions play a significant role in constructing the basis functions and is important to consider for the overall accuracy of the method. It has also been shown that the construction of basis functions per coarse node is independent of each other and has been implemented in parallel. CPU time for a large number of processors was plotted, depicting a declining trend, which points to the scalability of the multiscale methods for solving large-scale problems. Effendiev and Hou [21] extended the idea for solving nonlinear problems. MsFEMs also share similarity with the variational multiscale methods [26]. Once the multiscale basis function/multiscale map is obtained, a global finite element formulation is used to solve the problem.

Application of MsFEM for solving two-phase flows has been studied extensively by [18, 19]. The authors use a global formulation which allows

capturing long-range effects more accurately than the traditional multiscale method, which uses local information for constructing the basis functions. Two different approaches have been considered for solving the transport equation. Moreover, application of MsFEM for solving inverse problems arising in subsurface characterization has also been addressed. One of the recent development of the multiscale methods is the use of limited global information. The idea is to use simplified surrogate models to extract non-local multiscale behavior of physical processes. The surrogate models are precomputed in an offline step which allows me to compute effective parameters, which are useful for solving dynamic problems with varying sources and boundary conditions. Chen and Durlofsky, [6] showed the use of single phase information for accurate upscaling of two-phase flow and transport. The global single phase problem is solved several times to compute upscaled permeabilities which are then used in the simulation of two-phase flow and transport on the coarse grid. Similar to this approach based on upscaling, multiscale methods using limited global information and model reduction techniques are presented in [1] and [31] respectively.

Several other variants of multiscale methods exist in the literature which is used for solving a wide variety of physical problems involving issues related to scaling [20]. The present work discusses these variations in the context of problems related to subsurface flow and its applications. Effendiev et al. [14] implemented another approach called the GMsFEM for two-phase flow problems. GMsFEM provides a flexible framework by systematically enrich-

ing the coarse space, taking into account the small scale information through a high dimensional input space. The author proposed that GMsFEM framework overcomes the limitation of the reduced solution space which needs to be generated for different forcing or boundary conditions. Section 2.2.2 covers further details on this. Presho et al. [5] proposed the method for construction of locally conservative flux fields used in the two-phase flow models. It has also been shown that the addition of more basis functions to the enriched multiscale space produces solutions which more accurately captures the behavior and converges faster to the full fine scale model. Effendiev et al. [30] developed and investigated oversampling strategies to improve the accuracy of multiscale methods. In general, oversampling allows the use of larger coarse blocks for basis function generation. The following subsection provides an overview of different finite element techniques, implementation, and variation for addressing problems related to subsurface flow.

2.2 Finite Element Techniques: Overview

2.2.1 Multiscale Methods

Implementation of multiscale methods involves two major components, the first being the basis function generation to capture the fine scale information and the second is the global formulation which combines the basis function formulation with a variety of available techniques like finite element, finite volume, and mixed finite element methods to solve the physical problem. For example, in the case of MsFEM, the basis functions are constructed

from the solution of the leading-order homogeneous elliptic equation on each coarse element with some specified boundary conditions. The domain of interest for the multiscale basis function can even be chosen to be smaller than the coarse grid if the RVE is used to represent the local heterogeneities within the coarse-grid block. It is necessary to mention that the basis function generation involves solving the homogenized equations on the fine scale and therefore is computationally the most expensive portion of the whole process.

Once the basis functions are generated, global coupling procedures project the solution on the coarse dimensional space. The choice of the global coupling scheme is affected by the problem one is solving. In subsurface applications, mass conservative schemes play a central role, and therefore it becomes necessary to consider methodologies which can provide conservative approximations for flux. In the case of the Multiscale Finite Volume Method (MsFV), finite volume global formulation with multiscale basis functions are used to obtain a mass conservative velocity field on a coarse grid [28]. In the Multiscale Finite Volume Element Method (MsFVEM), mass conservative equations are set on the dual grid. Both these methods are different from conventional MsFEM since the velocity field obtained by both methods is conservative at the coarse grid level.

For multiphase flow and transport simulations, one often needs the conservative velocity at the fine scale level to solve the transport part of the equation. Chen et al. [7] implemented the mixed MsFEM for reconstructing conservative velocity fields at the fine scale. The general implementation

framework of the multiscale methods can be summarized as:

- Set the coarse mesh configuration from the fine scale mesh information.
- For every node of each of the coarse grid blocks, compute the basis function satisfying the boundary condition and the homogenized equation.
- Assemble the stiffness matrix on the coarse mesh. This may differ based on the strategy used for global coupling. The specific formulation will govern the assembly step.
- Assemble the external force on the coarse matrix.
- Solve the system at the coarse grid level.

2.2.2 Generalized Multiscale Finite Element Method

In subsurface flow problems the input space consists of permeability and source terms which take values from a large parameter space. It is because we are dealing with problems where the permeability field is high contrast and heterogeneous. Therefore, to make accurate predictions on the global level, one may have to solve a large number of forward problems to account for the variability in permeability and source terms. The computational cost associated with this would be tremendous, and therefore, model reduction techniques are sought which can reduce the solution space to a small dimensional space. However, in the presence of multiscale features, it is hard to obtain this reduced dimensional space with the existing techniques, which was one of the motivations to develop the GMsFEM [14].

Model reduction techniques can be grouped into two classes, global and local. Global reduction techniques approximate the solution space by global fields (global snapshots) obtained by solving global problems for many input parameters. The approximation space is further reduced using the spectral decomposition approach. However, this method is sensitive to boundary data and the source terms, changing one or both leads to re-computation.

When dealing with problems of multiple scales in heterogeneous porous media, model reduction techniques are often needed. As discussed above, in such cases, multiscale type methods attempt to approximate the fine scale effects on the coarse scale using the localized basis functions. This methodology is known as local model reduction approach, where the local effects are transmitted at the coarse grid level. An offline space is first constructed, which is used to obtain the multiscale basis functions at the online stage. Local approaches are useful as they avoid computation of the global snapshots. Another advantage is that in global approach one needs to compute a large number of global snapshots and all of these may not contribute to the solution. Such redundant values of the parameters are identified in the local approaches quickly and inexpensively at the coarse level.

GMsFEM is a local reduced order technique which systematically enriches the coarse space such that the desired level of accuracy is achieved, and the solution converges to fine scale as the coarser space is enriched more and more. This approach divides the computation into two stages: offline and online. In the offline stage, a small dimensional space is constructed and is

used to construct basis functions in the online stage. Substantial reduction in computational time is obtained since these basis functions can be re-used for any input parameter for solving the problem on the coarse grid. An outline is presented below,

Offline Computation:

- Configuring the coarse grid structure which contains the underlying network of interconnected fine grid blocks.
- Computing local snapshots by solving local problems for various choices of input parameter within each coarse grid.
- Construction of offline space via spectral decomposition of the local snapshot space. This step involves solving the eigenvalue problems and selecting eigenvector corresponding to the largest eigenvalues.

Online Computation:

- Computation of the multiscale basis function for each of the parameter.
- Solution of the coarse grid problem for any given force term and boundary conditions.

This computed online coarse space is used within the finite element framework such as Galerkin or Petrov-Galerkin coupling to solve the global problem. It is important to emphasize that this approach is different from the earlier described ones as it provides a procedure to enrich the local space systematically to converge to the fine grid solution.

2.3 Summary

This chapter gives a brief overview starting from the development of methods to solve problems in subsurface flow governed by high contrast heterogeneous permeability field. Common finite element techniques like MsFEM and GMsFEM are reviewed, which are the focus of the current work. Several variants of the Multiscale method such as MsFEM, MsFV, MsFVEM and mixed MsFEM are also discussed. Finally, general algorithmic steps are given from an implementation point of view. The next chapter will describe the physical problem in more detail and relevant mathematical details of the above-discussed methodologies used in our work.

Chapter 3

Problem Statement

We discuss the problem in the context of hydrocarbons flowing through a porous medium. The problem consists of a heterogeneous oil reservoir confined in a domain Ω. The reservoir is equipped with an injection well from which water is discharged to push the trapped oil towards the production wells situated on the perimeter of the domain. Certain assumptions are made to model this physical process mathematically. First, the system is two phase with oil, o, and water, w considered incompressible fluids. Second, capillary pressure is not included in the model. Further assumptions are the gravity free environment and the two fluids fill pore space completely. Darcy's law combined with the conservation of mass allows us to write the governing equations as

$$\nabla \boldsymbol{v} = q \tag{3.1}$$

$$\boldsymbol{v} = -\lambda(S)k(\boldsymbol{x})\nabla p \tag{3.2}$$

$$\frac{\partial s}{\partial t} + \nabla \cdot (f(S)\boldsymbol{v}) = q_w, \tag{3.3}$$

where $\boldsymbol{v}$ is the velocity, S is the water saturation and $k(\boldsymbol{x})$ is the permeability coefficient. The total mobility $\lambda(S)$ and the flux function $f(S)$ are respectively given as

$$\lambda(S) = \frac{k_{rw}(S)}{\mu_w} + \frac{k_{ro}(S)}{\mu_o} \tag{3.4}$$

$$f(S) = \frac{k_{rw}/\mu_w}{\lambda(S)}, \tag{3.5}$$

where $k_{rj}, j = w, o$ is the relative permeability of the phase j.

In this present work, the focus is on the elliptic pressure equation, which will be solved on the fine scale using a continuous Galerkin formulation and on the coarse scale using MsFEM and GMsFEM to obtain the pressure at fine and coarse scales respectively. For this purpose, the elliptic part of the problem is rewritten as

$$\begin{aligned} -\nabla.(\lambda k(\boldsymbol{x})\nabla p) = q \quad &\text{in} \quad \Omega \subset R^2 \\ p = p_D \quad &on \quad \Gamma_D \\ -\lambda k(\boldsymbol{x})\nabla p.\boldsymbol{n} = g_N \quad &on \quad \Gamma_N. \end{aligned} \tag{3.6}$$

The variational formulation of Eq. 3.6 is to find $p \in H^1(\Omega)$ with $(p - p_D) \in H^1_D = w \in H^1(\Omega) : w|\Gamma_D = 0$ and such that

$$a(p, w) = F(w) - \langle g_N, w \rangle_{\Gamma_N} \ \forall \ w \ \in H^1_D. \tag{3.7}$$

Here p and $v \in H^1(\Omega)$. The bi-linear form of a is defined by

$$a(p, w) = \int_\Omega \lambda k(\boldsymbol{x}) \nabla p \cdot \nabla w \ d\boldsymbol{x}. \tag{3.8}$$

The functional $F(w)$ is defined as

$$(q, w) = \int_\Omega q \ w \ d\boldsymbol{x}, \tag{3.9}$$

and the boundary condition is given by

$$\langle g_N, w \rangle_{\Gamma_N} = \int_{\Gamma_N} g_N \ w \ dl. \tag{3.10}$$

The approximate solution of p is sought in some finite dimensional subspace of H^1_D, $\mathcal{V}_h$, that satisfies Eq. 3.7. The domain Ω is discretized into rectangular subdomains, τ_h, such that $\bigcup \tau_h = \Omega$. The approximate solution is found to satisfy $(p_h - p_{D,h})$ and

$$a(p_h, w_h) = (q, w_h) - \langle g_N, w_h \rangle_{\Gamma_N} \ \forall \ w_h \in \mathcal{V}_h. \tag{3.11}$$

The above formulation leads to a linear system such that $\boldsymbol{Ap} = \boldsymbol{f}$, where $\boldsymbol{A}$ is a square matrix $(N_f \times N_f)$ whose entries are given as $a(\phi_i, \phi_j)$, $\boldsymbol{f}$ is the vector of entries (q, ϕ_i) for inside nodes and $(q, \phi_i) - \langle g_N, \phi_i \rangle$ for nodes on the Neumann boundary. Here N_f is the size of the fine scale matrix.

3.1 Multiscale Finite Element Method

MsFEM solves Eq 3.7 in the finite dimensional subspace of H^1_D. The distinctive feature of the method is that it obtains the solution without directly resolving the fine scale details of $k(\boldsymbol{x})$. The size of the grid chosen for computation is larger than the characteristic scale of $k(\boldsymbol{x})$ resulting in less computer memory requirements as a consequence of the smaller global system. The information about the fine scale is ingrained within the multiscale basis functions that characterize the finite-dimensional subspace. The governing equations for obtaining the multiscale basis functions are obtained by solving Eq 3.12 and enforcing the boundary conditions such that the multiscale basis functions coincide with the standard bilinear finite element basis functions at the boundaries.

$$\begin{aligned} -\nabla.(k(\boldsymbol{x})\nabla\chi_{i,\tau}) = 0 \quad \text{in} \quad \tau \\ \chi_{i,\tau}(\boldsymbol{x}) = \phi_{i,\tau}(\boldsymbol{x}) \quad \text{on} \quad \partial\tau. \end{aligned} \tag{3.12}$$

The multiscle finite dimensional subspace is defined as

$$\mathcal{V}_{ms,h} = span\ (\chi_i)\ \subset\ H^1_D \tag{3.13}$$

Consistent to the above definition, the multiscale method seeks to find $p_{ms,h}$ with $(p_{ms,h} - p_{D,h})\ \in\ \mathcal{V}_{ms,h}$ such that

$$a(p_{ms,h}, w_h) = (q, w_h) - \langle g_N, w_h \rangle_{\Gamma_N} \ \forall \ w_h \ \in \ \mathcal{V}_{ms,h}. \tag{3.14}$$

Such a treatment yields system of equations $\boldsymbol{A_{ms}}\ \boldsymbol{p} = \boldsymbol{f_{ms}}$. Here $\boldsymbol{A_{ms}}$ is a square matrix of size $N_c \times N_c$ with entries $a(\chi_i, \chi_j)$, $\boldsymbol{f_{ms}}$ is a vector of size $N_c \times 1$ with entries (q, χ_i) for inside nodes and $(q, \chi_i) - \langle g_N, \chi_\tau \rangle$ for nodes lying on the Neumann Boundary.

From the above discussion, it is evident that the crux of the multiscale method lies in the fact that if one wishes to solve Eq 3.6 using finite element formulation as described in Eq 3.11, it is required to choose a mesh resolution comparable to the scale of heterogeneity in $k(\boldsymbol{x})$, which would ultimately yield a very big system whose inversion may pose a significant challenge. As well, issues related to the storage of such a large system may further add to the problems. On the other hand, multiscale method avoids this drawback as there is no need to pose Eq 3.14 on a mesh having the resolution comparable to $k(\boldsymbol{x})$. It is because the fine scale information is ingrained into the finite dimensional subspace $\mathcal{V}_{ms,h}$ spanned by the multiscale basis function, χ.

3.2 Generalized Multiscale Finite Element Method

As evident from the above methodology, the aim of the present work is to reduce systematically the dimension of the system that results from the fully-resolved (fine scale) solution. A local model reduction technique is applied which constructs localized basis functions that span the coarse solu-

tion space. Fig 3.1, provides a representation of the model reduction approach where the coarse grid is shown for representative and notation purposes. $\{y_i\}_{i=1}^{N_c}$ denote the vertices of the coarse scale mesh τ^H, where $N_c << N_f$. The neighborhood of the node y_i is represented by ω_i such that

$$\omega_i = \bigcup \{K_j \in \tau^H; \quad y_i \in \overline{K}_j\} \tag{3.15}$$

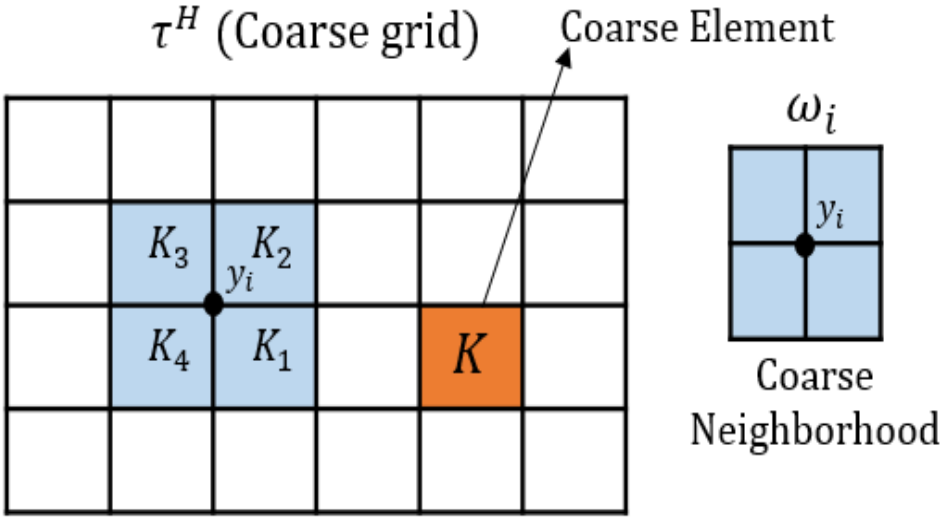

Figure 3.1: Illustration of Coarse Neighborhood, Ghommen et al. [24]

Using this coarse mesh τ^H, we start with an initial coarse space $V_o^{initial} = span\,\{\chi_i\}_{i=1}^{N_c}$, where χ_i are standard (MsFEM) partition of unity functions satisfying Eq 3.12. A solution computed within $V_o^{initial}$ is the standard MsFEM solution. However, the coarse space may be systematically enriched by using

generalized approaches for multiscale model reduction [16, 22, 23].The coarse space is enriched by multiplying the localized eigenvalue problems to the initial partition of unity. It produces a result such that the solution error decreases with respect to the localized eigenvalue behavior [16]. The first step in this process involves computing pointwise energy of the original basis functions by setting

$$\widetilde{k} = kH^2 \sum_{i=1}^{N_c} \mid \bigtriangledown \chi_i \mid^2 \tag{3.16}$$

where H is the coarse mesh size. Once $\widetilde{k}$ is available, we solve homogeneous Neumann eigenvalue problem given by Eq 3.17, on each coarse block neighborhood ω_i.

$$- \bigtriangledown \cdot (k \ \bigtriangledown \psi_l) = \lambda_l \ \widetilde{k} \ \psi_l \tag{3.17}$$

The eigenvalues and eigenvectors are denoted by $\{\lambda_l^{\omega_i}\}$ and $\{\psi_l^{\omega_i}\}$ respectively. For a zero Neumann problem, the first eigenpair is represented as $\{\lambda_l^{\omega_i}\} \ = \ 0$ and $\{\psi_l^{\omega_i}\} \ = \ 1$. The eigenvectors corresponding to small, asymptotically vanishing eigenvalues are used for the construction of the coarse space V_o. The updated basis functions are defined by Eq 3.18

$$\phi_{i,l} = \chi_i \ \psi_l^{\omega_i} \quad 1 \leq i \leq N_c \ ; \ 1 \leq l \leq L_i \tag{3.18}$$

where L_i denotes the number of eigenvectors that will be chosen for each

node i. With the updated basis functions computed above, the local spectral multiscale space is defined by Eq 3.19

$$V_o = span\left\{\phi_{i,l} \ : \ 1 \leq i \leq N_c \ ; 1 \leq l \leq L_i\right\} \tag{3.19}$$

This enriched space is used for obtaining the approximate solution for the pressure equation. For rigorous error estimates, refer to [17].

3.3 Summary

The first section of the chapter presents the governing equations involved in the fluid flow. The subsequent sections discuss the Finite Element formulation of the problem. The problem has been solved on both fine and coarse grids for the comparison purposes. The initial formulation is on the fine scale where we use the standard bilinear basis functions. Later on, coarse scale techniques are formulated where multiscale finite element basis functions are generated. The last section deals with the enrichment of coarse space after solving the corresponding eigenvalue problem on each of the coarse grid neighborhood. The choice of the number of basis functions governs the accuracy and the computation time. Numerical results are presented in the next section to verify the claims. Implementation of parallel computing for independent calculation of basis functions as well coarse space enrichment is also shown.

Chapter 4

Numerical Results

This chapter contains the numerical results based on the governing equations developed in the earlier chapter. The objective of this section is to compare the coarse mesh results with the fine scale solutions and to improve the computational efficiency of the coarse scale solutions by implementing parallel computing. Shared memory parallel programming is explored, where each processor accesses the same memory. Initially, the fine scale problem is solved to set up the benchmark solution and the CPU time associated with the serial run of the program. These benchmark results are later compared, after solving the problem on the coarse scale using the standard multiscale method and spectral enrichment.

The first section consists of a set of results for a 2D Finite Element Code made parallel using the MATLAB parallel computing toolbox. The parallel computation starts by creating a pool of workers. The pool consists of a set of MATLAB workers in a cluster or desktop as shown in Fig 4.1. The program is run on Lonestar Linux cluster, which allows a maximum of twelve workers for shared parallel operations on a single node, using MATLAB-2013a version. The comparison results for CPU time are thus restricted to a maxi-

mum of twelve workers. Within the MATLAB, the parallel computing toolbox supports several programming models like Data parallel, Distributed Memory (Message Passing), Single Program Multiple Data (SPMD) and Multiple Program Multiple Data (MPMD). For shared memory applications, available ones are SPMD blocks and PARFOR loops. PARFOR is chosen because the computation of basis functions per coarse node is independent of each other, which implies no dependencies in computation. PARFOR divides the loop iterations into groups so that each worker executes some portion of the total number of iterations. Since the problem is two-dimensional, it involves an inner and outer loop for looping through the entire grid. MATLAB does not allow the use of nested PARFOR and therefore, either the outer or the inner loop can be made in parallel. Following three portions of the program are made parallel, to calculate the CPU time as more processors are used.

- Standard Multiscale basis function generation (one basis function per coarse node)
- Generalized Multiscale basis function generation (spectral enrichment with more than one basis functions per coarse node)
- Downscaling fluxes to fine scale.

The rest of the program which typically involves assembling the global system, setting up the boundary conditions and global system solver are still in serial. This way, we are comparing the overhead time in solving the coarse system, which is essentially due to the basis function generation.

A similar exercise is repeated for a more intensive 3D Finite Element code using OpenMP in C++. OpenMP uses a fork-join model of parallel execution as shown in Fig 4.2. When a thread encounters a parallel construct, it creates a team composed of itself and some additional number of threads. The encountering thread becomes the master, and the other threads are called slave of the team. When a thread finishes its work within the parallel construct, it waits at the implicit barrier at the end of the parallel construct. When all the team members have arrived at the barrier, the threads can leave the barrier. The master thread continues execution of user code beyond the end of the parallel construct. The slave threads become free and are available for the next parallel construct. This model offers a lot more flexibility than the MATLAB parallel computing toolbox. For example, unlike PARFOR, OpenMP allows nested parallel loops. However, the efficiency depends on the complexity of the code and may involve several trials before fixing the level of nesting in the program. In the present case, no parallel nesting is used, and all the slave threads work at the outermost loop. The maximum number of threads used in the program also depends on the run-time efficiency. Using a higher number of threads may not always give the best results. In this study, the best results are obtained for twelve number of threads. The following sections discuss the numerical results for pressure, velocity, relative L^2 and H^1 norm and execution time associated with a different number of processors/threads for two-dimensional and three-dimensional problem.

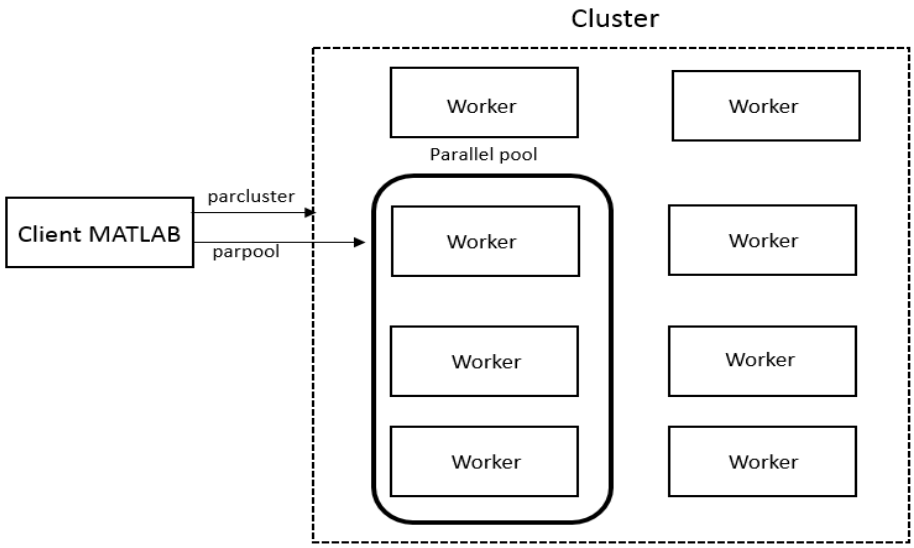

Figure 4.1: Schematic for Parallel Computing in MATLAB

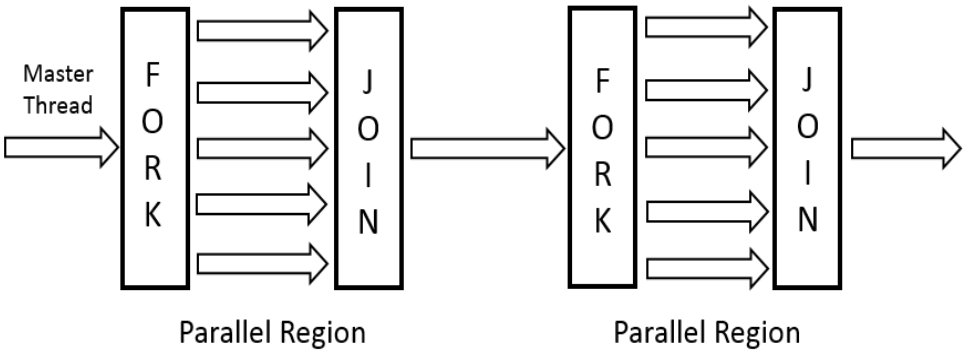

Figure 4.2: Fork-Join Model used in OpenMP

4.1 2D Finite Element Results

The domain is represented by $\Omega = [0,1] \times [0,1]$. To solve the pressure Eq. 3.6, the Dirichlet boundary conditions used are $p_L = 1$ and $p_R = 0$ at the left and right boundaries of Ω respectively. No flow (zero Neumann) conditions are specified at the top and bottom boundaries. There is no external forcing such that, $q = 0$ and $\lambda(S) = 1$ is assumed to further simply the problem. Fig 4.3 represents the deterministic high contrast permeability field, $k(\mathbf{x})$, used in the problem. This permeability is constructed on a fine mesh containing 100×100 elements. The coarse scale mesh is chosen to be 10×10. The total number of nodes in the fine scale mesh are 10201 whereas the total number of coarse nodes are 121. The summary of the two-dimensional finite element code is given in Table 4.1. The fine scale solution is considered as a benchmark and the performance of MsFEM and GMsFEM is compared against this. The global system solved using the multiscale methods is on the coarser grid, and hence, the memory requirements and the computational time required is significantly less than the fine solution. The time taken to solve the coarse system is 0.017 sec which is smaller than the time needed for fine scale solve, 0.35 sec. This difference becomes significant when large-scale complex problems are solved. Table 4.2 shows the relative L^2 and H^1 norm for the pressure solution obtained after solving the system on the coarse scale. It can be observed that as the number of basis functions per coarse node is increased, the solution tends to converge towards fine scale solution. Table 4.3 shows the relative L^2 and H^1 norm for flux obtained at the fine scale after downscaling.

Fig 4.4, 4.5 and 4.6 shows the pressure solution obtained after solving the system of equations at the fine scale and the coarse scale respectively. Here also, it is evident that systematic spectral enrichment reduces the error and converges towards the fine scale solution. A similar trend is obtained for velocity flux along the x direction as shown in Fig 4.7, 4.8 and 4.9.

Table 4.1: Summary: 2D Finite Element Code

Total number of coarse elements in each direction	10
Total number of elements within each coarse element	10
Total number of elements in each direction	100
Total number of nodes in each direction	101
Total number of global nodes	10201

The next set of results focuses on the CPU time associated with solving the problem on the fine and coarse scale. Fig 4.10 shows the decline plot obtained when MsFEM and GMsFEM are implemented in parallel. Because of the restriction with the MATLABs parallel computing toolbox, only twelve number of processors could be used. It is one of the motivations to shift towards a more flexible OpenMP based shared memory parallel computing. Three components of the algorithm are made parallel: the **MsFEM** basis function generation, the **GMsFEM** spectral enrichment and the **downscaling** method to obtain fluxes on fine scale. Fig 4.11 gives a comparison of the overhead time required when solving the problem on the coarse scale using the multiscale method. The overhead time is obtained by adding up time needed for the basis function generation and the coarse system solver. Though the

CPU time required for the multiscale method is higher, the declining trend suggests that the method is scalable and can be effectively used for solving large scale problems where memory requirements and global system solver are significant issues. Table 4.4 tabulates the CPU time for a different number of processors.

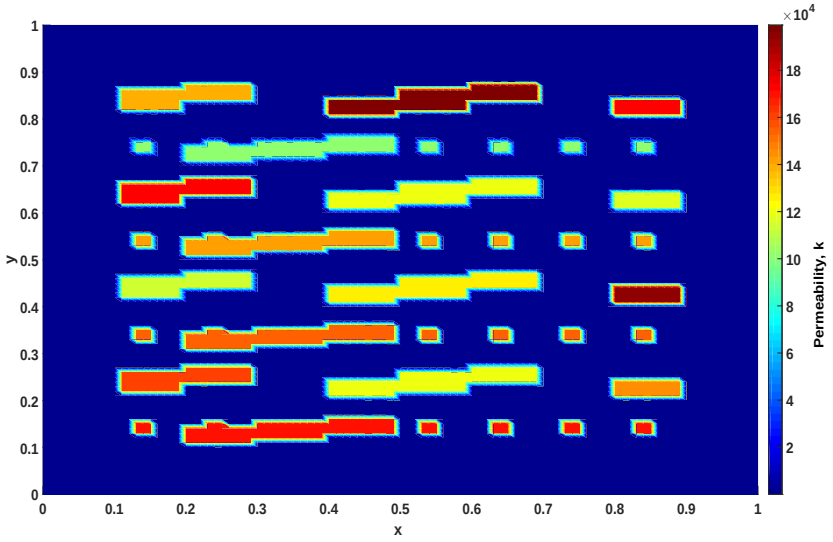

Figure 4.3: Deterministic High Contrast Permeability Field

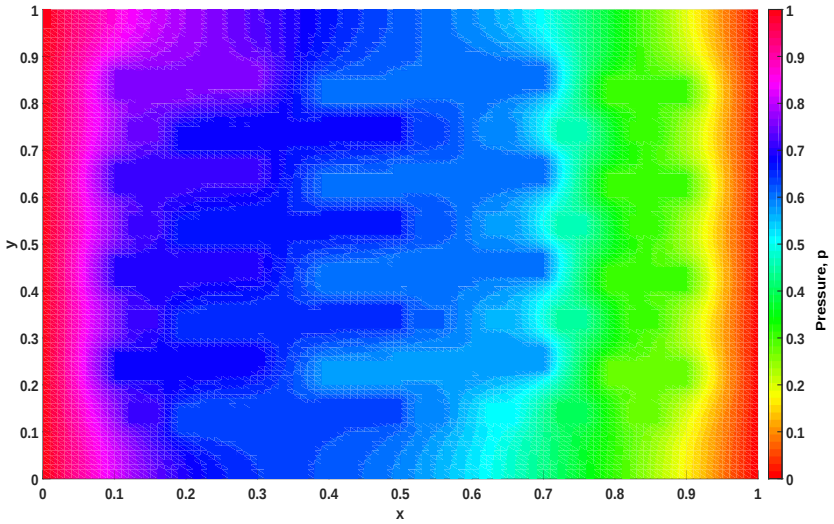

Figure 4.4: Fine Scale: Pressure Solve

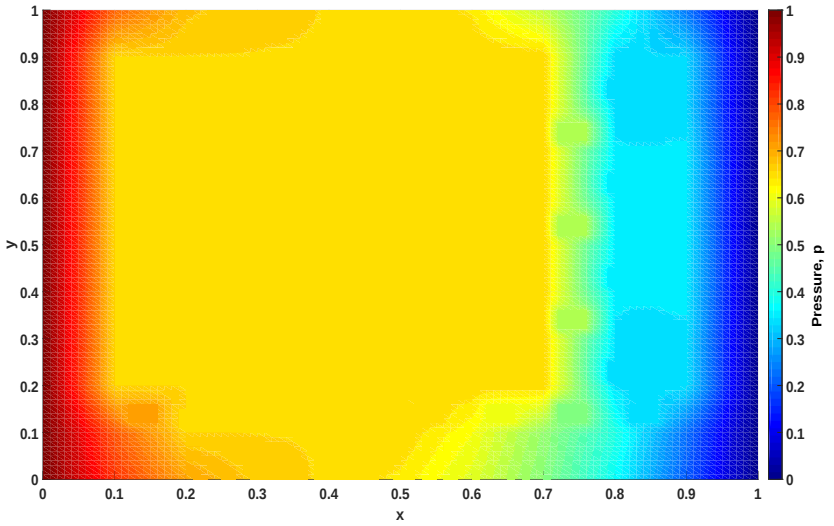

Figure 4.5: MsFEM: Pressure Solve

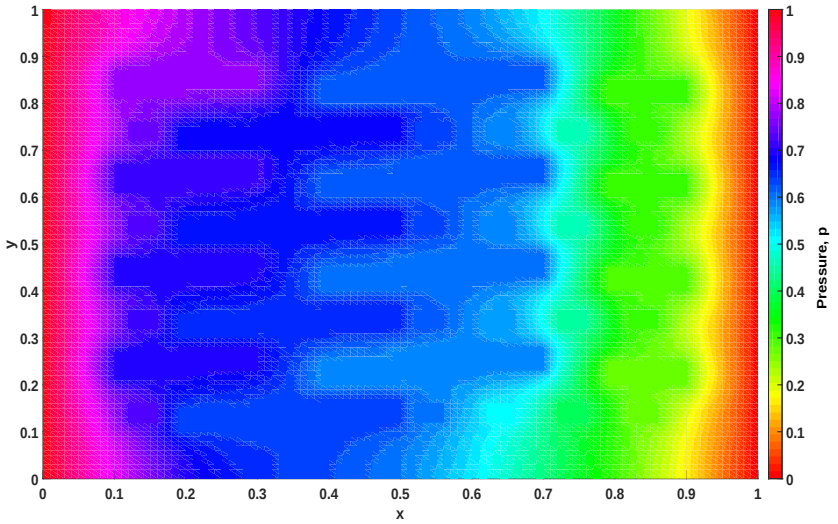

Figure 4.6: GMsFEM: Pressure Solve

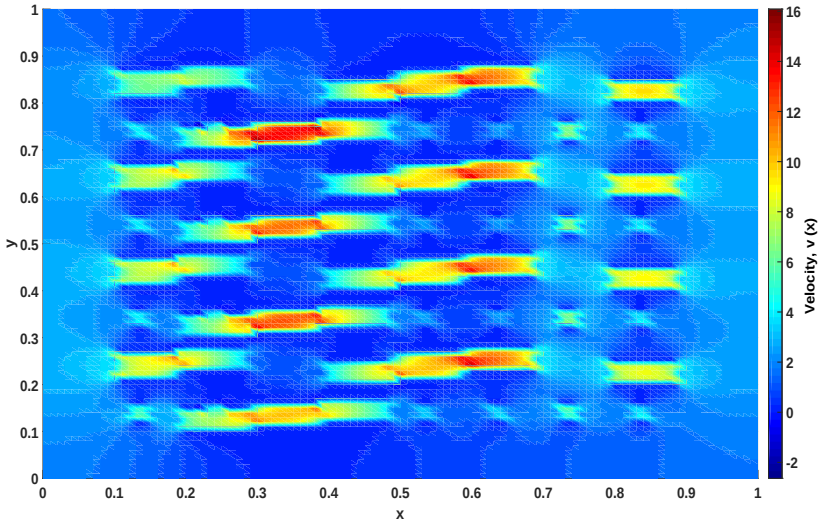

Figure 4.7: Fine Scale: Velocity along x direction

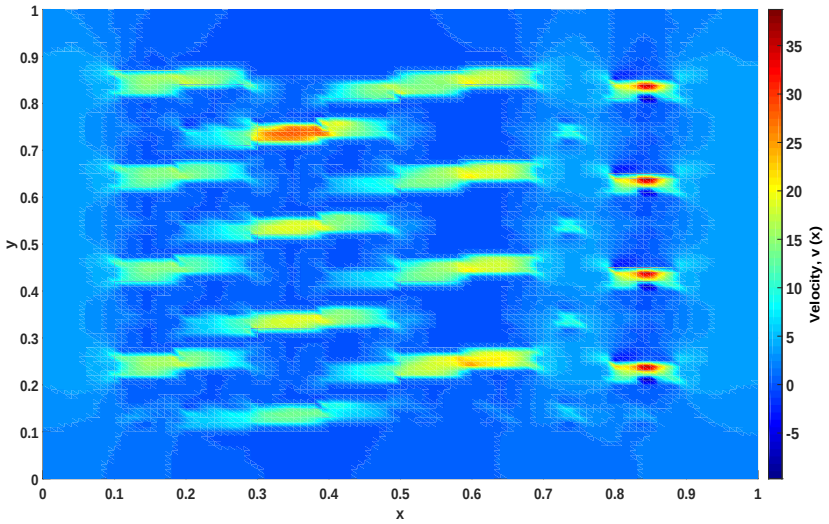

Figure 4.8: MsFEM: Velocity along x direction

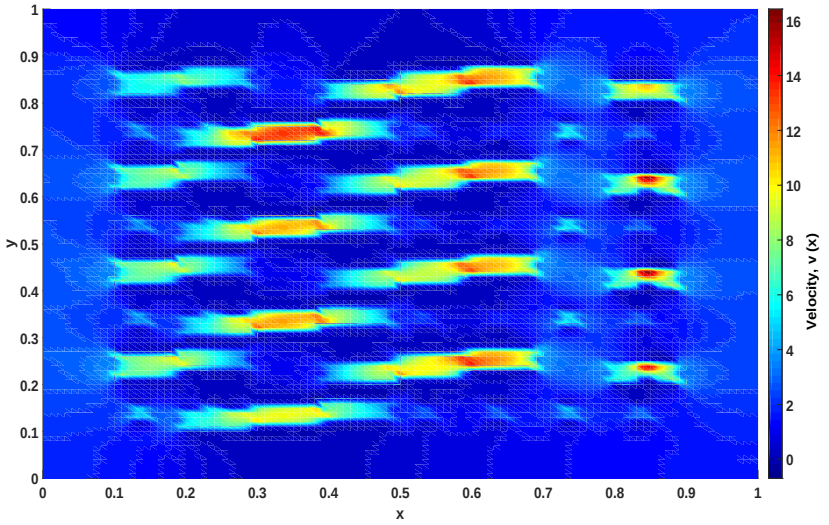

Figure 4.9: GMsFEM: Velocity along x direction

Table 4.2: Relative L^2 and H^1 norm for different levels of enrichment: Pressure

	L^2 (%)	H^1 (%)
$N_z = 1$	8.425	480.09
$N_z = 2$	7.389	39.859
$N_z = 4$	1.180	14.971
$N_z = 8$	0.897	11.098

Table 4.3: Relative L^2 and H^1 norm for different levels of enrichment: Flux

	L^2 (%)	H^1 (%)
$N_z = 1$	68.907	109.01
$N_z = 2$	33.331	52.481
$N_z = 4$	8.724	21.345
$N_z = 8$	8.190	19.346

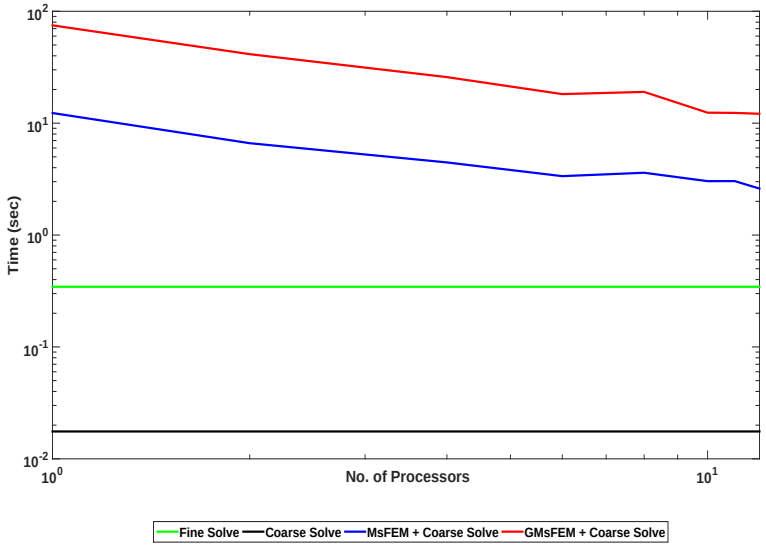

Figure 4.10: CPU Time: Coarse Solve + MsFEM/GMsFEM

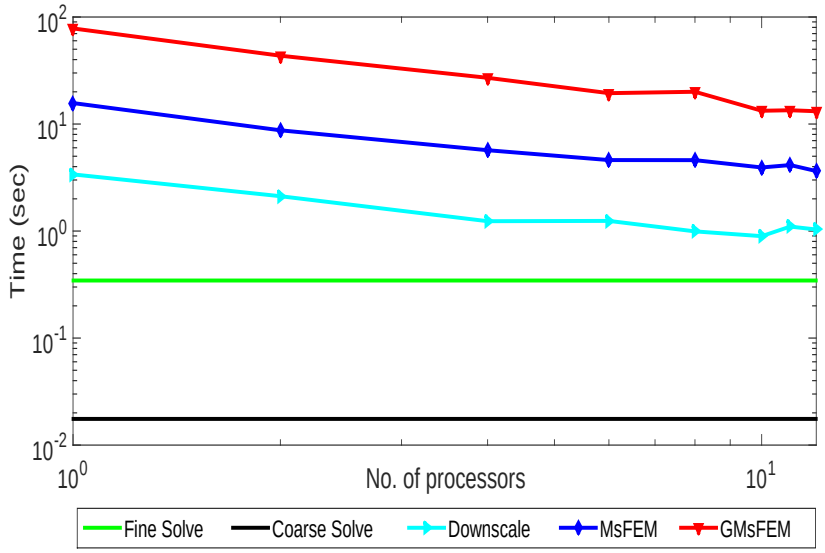

Figure 4.11: CPU Time: Coarse Solve + MsFEM/GMsFEM + Downscaling

Table 4.4: CPU Time (in seconds): 2D Finite Element

Processors #	Multiscale Basis	Spectral Enrichment	Downscaling
1	12.304	74.826	3.384
2	6.606	41.370	2.117
4	4.440	25.781	1.236
6	3.344	18.169	1.244
8	3.587	19.012	0.993
10	3.014	12.391	0.896
11	3.018	12.335	1.101
12	2.585	12.117	1.034

4.2 3D Finite Element Results

After attempting to solve the problem using a two-dimensional finite element code, a similar exercise is repeated with a three-dimensional finite element code. The results obtained after solving the two-dimensional problem are encouraging and led to the motivation to solve a three-dimensional problem and use OpenMP for parallel computing.

The domain is represented by $\Omega = [0,1] \times [0,1] \times [0,1]$. For simplicity, all the boundaries have Dirichlet boundary conditions such that all the six faces have $p = 0$. The forcing condition chosen is, $q = 1$. Fig 4.12 shows that the permeability field is log-normal which is generated by simulating random numbers from a standard normal distribution and raising to the exponent. The summary of the finite element code is given in Table 4.5. The Biconjugate Gradient Stabilized **(Bi-CGSTAB)** method is used for solving the system of

equations obtained as a result of the finite element formulation.

Table 4.5: Summary: 3D Finite Element Code

Total number of coarse elements in each direction	5
Total number of elements within each coarse element	5
Total number of elements in each direction	25
Total number of nodes in each direction	26
Total number of global nodes	17576

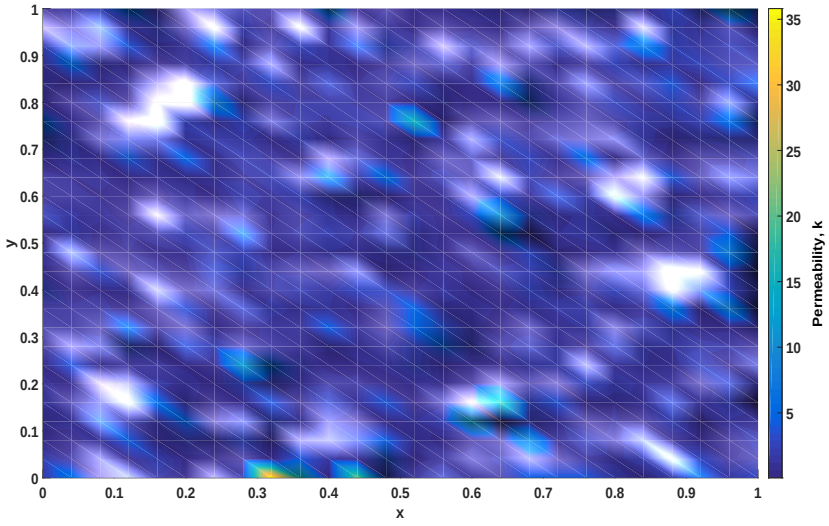

Figure 4.12: Log-normal Permeability Field

Fig 4.13 shows the benchmark fine scale pressure solution. Fig 4.14 and Fig 4.15 show the coarse scale solution obtained after using MsFEM and GMsFEM respectively. The number of basis functions used for spectral enrichment is fifteen. It can be seen clearly that the accuracy of the solution increases by

spectral enrichment. The visualization of the pressure solution is done using Paraview 4.0, and a slice at $x = 0.5$ is chosen for this purpose. The relative percentage error using the coarse scale techniques is given in Table 4.6.

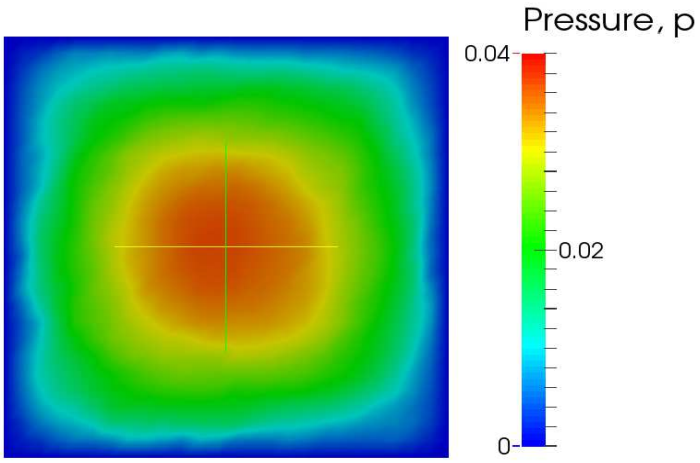

Figure 4.13: Pressure Solution: Fine Scale

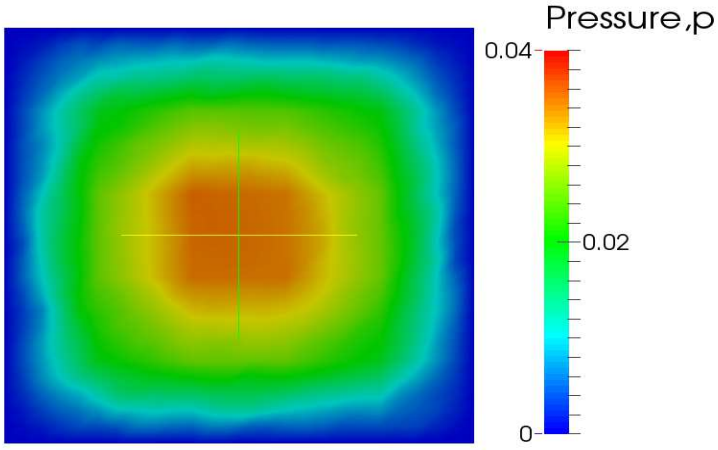

Figure 4.14: Pressure Solution: MsFEM

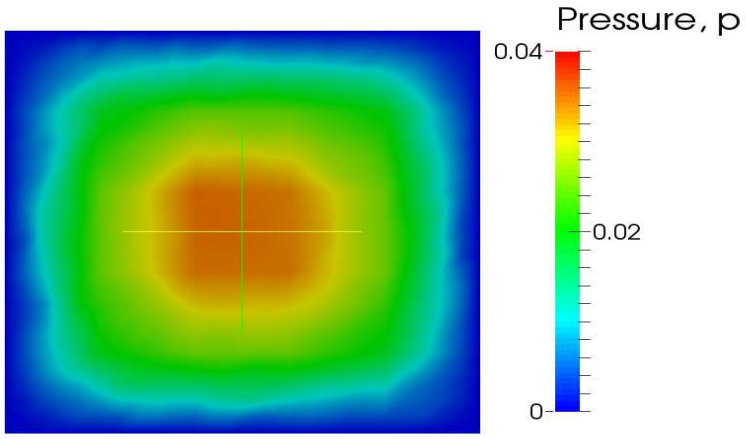

Figure 4.15: Pressure Solution: GMsFEM

Table 4.6: Relative L^2 and H^1 norms: Pressure

	L^2 (%)	H^1 (%)
MsFEM	9.465	32.483
GMsFEM	3.572	19.415

The next set of results focuses on the scalability of the algorithms when run in parallel. Shared memory parallelism is achieved here using OpenMP in C++. Table 4.7 shows the CPU time taken by different number of threads for MsFEM and GMsFEM. Fig 4.16 shows the graphical comparison of CPU time with the benchmark results obtained using fine scale method. The results indicate that the overhead time required for multiscale method drops below the fine scale solution. The time requirement for coarse solve is negligible and

therefore, the effective time is whatever required for the basis function generation. It is worthwhile to mention here the importance of compiler optimization flags used while compiling the programs in parallel. Without any optimization option, the compiler's goal is to reduce the cost of compilation and to make debugging produce the expected results. However, turning on optimization flags makes the compiler attempt to improve the performance. Several optimization flags such as -O1, -O2, -O3, etc. are supported by g++ compiler. The CPU time reported in Table 4.7 is obtained after turning on -O2 optimization flag which enables speed optimization and gives the best results.

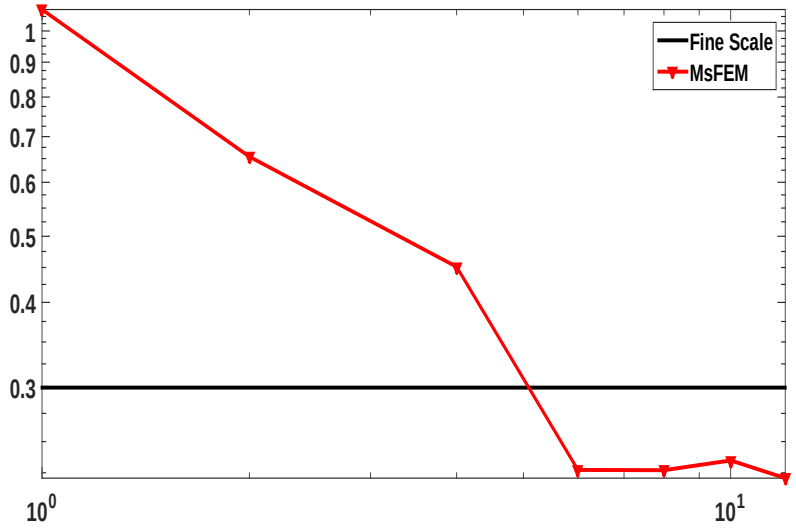

Figure 4.16: CPU Time: Coarse Solve + MsFEM

Table 4.7: CPU Time (in seconds): 3D Finite Element

Number of Threads	MsFEM	GMsFEM
1	1.075	3751.46
2	0.6542	1875.5
4	0.451	1047.7
6	0.2272	1047.64
8	0.227	1037.5
10	0.2346	1036.46
12	0.221	1029.04

The results, therefore, indicate that it is possible to beat the fine scale solution at the expense of some approximations used in the standard multiscale method. OpenMP provides more flexibility and the use of g++ compiler further enables optimization flags one can use for faster execution of the program. The spectral enrichment allows reducing the relative error but increases the computation time as shown in Table 4.7. The declining trend suggests the need of mass parallelism to make spectral enrichment further faster. It also depends on the type of problem one is solving and the level of accuracy desired. Distributed computing may further be explored to make the enrichment process faster.

Chapter 5

Conclusions and Future Work

The objective of the study was to solve the elliptic pressure Eq 3.6, where the permeability of the medium is high contrast with several jumps and discontinuities. Two separate studies have been conducted using a two-dimensional and a three-dimensional finite element code to solve numerically the elliptic pressure equation. Solving the equation on fine scale requires tremendous computational effort and storage demands. It may not always be feasible and required to solve such a problem on the fine scale, and therefore, multiscale techniques were developed which provide a suitable framework to handle the complexity. A desired accuracy and convergence to fine scale solution is demonstrated by using spectral enrichment technique, GMsFEM. It is worthwhile to mention here, that GMsFEM is same as standard MsFEM when only one basis function per coarse node is used for spectral enrichment.

5.1 Results

- Implementation of MsFEM and GMsFEM for solving the problem on the coarse grid is demonstrated. The pressure and the flux obtained using both the methods are compared against the benchmark fine scale

solutions. Two-dimensional results are presented first, where Figs 4.4, 4.5 and 4.6 graphically demonstrate that the relative error with MsFEM, which uses only one basis function per coarse node is more than GMsFEM where six basis functions per coarse node are used. This systematic enrichment of the coarse space after using more number of basis functions per coarse node, allows us to obtain the higher level of accuracy. Table 4.2 shows the numerical value of the relative error percentage, as a larger number of basis functions are used. Similar results are obtained for the three-dimensional problem, where the accuracy of the pressure solution increases when fifteen number of basis functions are used. However, the gain in accuracy leads to greater computational time, which leads to parallel computing.

- The size of the actual fine scale problem is chosen smaller for the demonstration purposes only. It is expected that the computation time for solving the global system and memory requirements will be tremendous for larger complex systems, and one would need multiscale methods. Fig 4.10 and 4.11 show that it is possible to implement these methods in parallel and reduce the CPU time associated with the basis function generation, spectral enrichment and downscaling. Though the computational time required for solving the fine scale system is lesser, much because of the smaller size of the problem chosen here, the time needed solving the coarse system drops down significantly indicating that the algorithms are scalable.

- The implementation has been made more general and flexible using OpenMP in C++ for solving a three-dimensional finite element code. It gives more flexibility for optimizing the program when compared to the MATLAB's parallel computing toolbox. Table 4.7 indicates a higher CPU time efficiency is achieved by using the **-O2** optimization scheme. A significant result is shown in Fig 4.16, where the CPU time required for generating basis functions using standard MsFEM drops below fine scale solve time. It is yet another claim that parallel implementation makes these algorithms scalable. Spectral enrichment still takes much longer, but the decline is significant when more threads are used.

5.2 Future Work

It can be concluded that the results of parallel computing for solving subsurface flow using multiscale methods are very encouraging. The next step can be to implement distributed parallel computing using Message Passing Interface (MPI). It can particularly be done for the loop involving basis function generation during the spectral enrichment process, which is computationally the most difficult step. It will provide more opportunities to enhance further the efficiency of the algorithms.

Another application may be in uncertainty quantification where permeability field is random and noisy. Researchers in the Computational Hydraulics Group are currently focused on solving the inversion problem which requires using the forward model as a "black box" iteratively. Parallel computing using

multiscale methods will make the computation more efficient.

Bibliography

[1] Jorg E Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Modeling and Simulation*, 2(3):421–439, 2004.

[2] Rachid Ababou, Dennis McLaughlin, Lynn W Gelhar, and Andrew FB Tompson. Numerical simulation of three-dimensional saturated flow in randomly heterogeneous porous media. *Transport in Porous Media*, 4(6):549–565, 1989.

[3] Ivo Babuška, Gabriel Caloz, and John E Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31(4):945–981, 1994.

[4] Ivo Babuška and John E Osborn. Generalized finite element methods: their performance and their relation to mixed methods. *SIAM Journal on Numerical Analysis*, 20(3):510–536, 1983.

[5] Lawrence Bush, Victor Ginting, and Michael Presho. Application of a conservative, generalized multiscale finite element method to flow models. *Journal of Computational and Applied Mathematics*, 260:395–409, 2014.

[6] Yuguang Chen and Louis J Durlofsky. Adaptive local–global upscaling for general flow scenarios in heterogeneous formations. *Transport in Porous*

Media, 62(2):157–185, 2006.

[7] Zhiming Chen and Thomas Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, 72(242):541–576, 2003.

[8] Manuel E Cruz and Anthony T Patera. A parallel monte-carlo finite-element procedure for the analysis of multicomponent random media. *International Journal for Numerical Methods in Engineering*, 38(7):1087–1121, 1995.

[9] Paul Dostert, Yalchin Efendiev, and Thomas Y Hou. Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification. *Computer Methods in Applied Mechanics and Engineering*, 197(43):3445–3455, 2008.

[10] LJ Durfolsky. Numerical calculation of equivalent grid block permeability tensors of heterogeneous porous media: Water resour res v27, n5, may 1991, p299–708. In *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, volume 28, page A350. Pergamon, 1991.

[11] Bruce B Dykaar and Peter K Kitanidis. Determination of the effective hydraulic conductivity for heterogeneous porous media using a numerical spectral approach: 1. method. *Water Resources Research*, 28(4):1155–1166, 1992.

[12] Yalchin Efendiev and Juan Galvis. Coarse-grid multiscale model reduction techniques for flows in heterogeneous media and applications. In *Numerical Analysis of Multiscale Problems*, pages 97–125. Springer, 2012.

[13] Yalchin Efendiev, Juan Galvis, and Eduardo Gildin. Local–global multiscale model reduction for flows in high-contrast heterogeneous media. *Journal of Computational Physics*, 231(24):8100–8113, 2012.

[14] Yalchin Efendiev, Juan Galvis, and Thomas Y Hou. Generalized multiscale finite element methods (gmsfem). *Journal of Computational Physics*, 251:116–135, 2013.

[15] Yalchin Efendiev, Juan Galvis, Guanglian Li, and Michael Presho. Generalized multiscale finite element methods: Oversampling strategies. *International Journal for Multiscale Computational Engineering*, 12(6), 2014.

[16] Yalchin Efendiev, Juan Galvis, and Xiao-Hui Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, 230(4):937–955, 2011.

[17] Yalchin Efendiev, Juan Galvis, and Xiao-Hui Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, 230(4):937–955, 2011.

[18] Yalchin Efendiev, Victor Ginting, T Hou, and R Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *Journal of Computational Physics*, 220(1):155–174, 2006.

[19] Yalchin Efendiev and T Hou. Multiscale finite element methods for porous media flows and their applications. *Applied Numerical Mathematics*, 57(5):577–596, 2007.

[20] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science and Business Media, 2009.

[21] Yalchin Efendiev, Thomas Y Hou, Victor Ginting, et al. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):553–589, 2004.

[22] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling and Simulation*, 8(4):1461–1483, 2010.

[23] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: Reduced dimension coarse spaces. *Multiscale Modeling and Simulation*, 8(5):1621–1644, 2010.

[24] Mehdi Ghommem, Michael Presho, Victor M Calo, and Yalchin Efendiev. Mode decomposition methods for flows in high-contrast porous media. part ii. local-global approach. *arXiv preprint arXiv:1301.5742*, 2013.

[25] Thomas Y. Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169 – 189, 1997.

[26] Thomas JR Hughes. Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1):387–401, 1995.

[27] Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1):3–24, 1998.

[28] P Jenny, SH Lee, and HA Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187(1):47–67, 2003.

[29] JF McCarthy. Comparison of fast algorithms for estimating large-scale permeabilities of heterogeneous media. *Transport in Porous Media*, 19(2):123–137, 1995.

[30] Michael Presho and Juan Galvis. A mass conservative generalized multiscale finite element method applied to two-phase flow in heterogeneous porous media. *arXiv preprint arXiv:1504.02033*, 2015.

[31] Christophe Prudhomme, Dimitrios V Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, and Gabriel Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[32] Andrew FB Tompson. Numerical simulation of chemical migration in physically and chemically heterogeneous porous media. *Water Resources Research*, 29(11):3709–3726, 1993.

[33] Xian-Huan Wen and J Jaime Gómez-Hernández. Upscaling hydraulic conductivities in heterogeneous media: An overview. *Journal of Hydrology*, 183(1):ix–xxxii, 1996.

Vita

Anshul Goyal received the Bachelor of Technology degree in Civil Engineering from the Indian Institute of Technology, Guwahati in July 2014. He was accepted and started graduate studies in the Department of Civil, Architectural and Environmental Engineering at the University of Texas at Austin in 2014. He joined the Computational Hydraulics Group at Institute for Computational Sciences and Engineering for Master's Thesis in May 2015.

Email Address: anshul.goyal@utexas.edu

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.