MULTISCALE METHODS FOR STOCHASTIC COLLOCATION OF MIXED FINITE ELEMENTS FOR FLOW IN POROUS MEDIA

by

Benjamin Ganis

B.S. in Mathematics and Computer Science

University of Pittsburgh, 2004

Submitted to the Graduate Faculty of

the Department of Mathematics in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

MATHEMATICS DEPARTMENT

This dissertation was presented

by

Benjamin Ganis

It was defended on

April 8, 2010

and approved by

Professor Ivan Yotov, Department of Mathematics, University of Pittsburgh

Professor William Layton, Department of Mathematics, University of Pittsburgh

Professor Catalin Trenchea, Department of Mathematics, University of Pittsburgh

Professor Noel Walkington, Department of Mathematical Sciences, Carnegie Mellon University

Dissertation Director: Professor Ivan Yotov, Department of Mathematics, University of Pittsburgh

MULTISCALE METHODS FOR STOCHASTIC COLLOCATION OF MIXED FINITE ELEMENTS FOR FLOW IN POROUS MEDIA

Benjamin Ganis, PhD

University of Pittsburgh, 2010

This thesis contains methods for uncertainty quantification of flow in porous media through stochastic modeling. New parallel algorithms are described for both deterministic and stochastic model problems, and are shown to be computationally more efficient than existing approaches in many cases.

First, we present a method that combines a mixed finite element spatial discretization with collocation in stochastic dimensions on a tensor product grid. The governing equations are based on Darcy's Law with stochastic permeability. A known covariance function is used to approximate the log permeability as a truncated Karhunen-Loève expansion. *A priori* error analysis is performed and numerically verified.

Second, we present a new implementation of a multiscale mortar mixed finite element method. The original algorithm uses non-overlapping domain decomposition to reformulate a fine scale problem as a coarse scale mortar interface problem. This system is then solved in parallel with an iterative method, requiring the solution to local subdomain problems on every interface iteration. Our modified implementation instead forms a Multiscale Flux Basis consisting of mortar functions that represent individual flux responses for each mortar degree of freedom, on each subdomain independently. We show this approach yields the same solution as the original method, and compare the computational workload with a balancing preconditioner.

Third, we extend and combine the previous works as follows. Multiple rock types are modeled as nonstationary media with a sum of Karhunen-Loève expansions. Very hetero-

geneous noise is handled via collocation on a sparse grid in high dimensions. Uncertainty quantification is parallelized by coupling a multiscale mortar mixed finite element discretization with stochastic collocation. We give three new algorithms to solve the resulting system. They use the original implementation, a deterministic Multiscale Flux Basis, and a stochastic Multiscale Flux Basis. Multiscale *a priori* error analysis is performed and numerically verified for single-phase flow.

Fourth, we present a concurrent approach that uses the Multiscale Flux Basis as an interface preconditioner. We show the preconditioner significantly reduces the number of interface iterations, and describe how it can be used for stochastic collocation as well as two-phase flow simulations in both fully-implicit and IMPES models.

Keywords: Porous Media Flow, Mixed Finite Element, Mortar Finite Element, Multiscale Basis, Uncertainty Quantification, Stochastic Collocation.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Ivan Yotov for his invaluable guidance and support. He have been a great role model for me, and I am privileged to consider him both a mentor and a friend. This project was only possible due to his patient, steady direction, and I have thoroughly enjoyed the time we spent developing these new ideas.

Thank you to the many friends and colleagues at my home for the past five years, the Department of Mathematics at the University of Pittsburgh. I have benefited from the knowledge and wisdom of all my professors, including my committee members William Layton, Catalin Trenchea, and Noel Walkington. Thanks also to Mike Sussman and Beatrice Riviere for their help in beginning my career in scientific computation.

Thank you to Mary Wheeler at ICES for generously inviting me to visit on so many occasions, and involving me in the impressive work that is done at the Center for Subsurface Modeling. I am very excited to continue my career at CSM, beginning this summer.

Thank you to everyone with whom I have had the privilege of collaboration at Pitt, including Jeffrey Connors, Danail Vassilev, and Ming Zhong. Thank you to Gergina Pencheva, Sunil Thomas, and Tim Wildey for the wonderful collaborations I have had during my visits to CSM. Thank you to John Burkardt and Dongxiao Zhang for the willingness to share your ideas, and the integration of computer code into this research project.

Thank you to my family for inspiring me at a young age to get interested in science, mathematics, and technology. Their unconditional love and encouragement have allowed me pursue a meaningful career and achieve my dreams.

Finally, I wish to dedicate this dissertation to my wife Beth. I owe her everything for sacrificing with me these past years as a graduate student. I look forward to our future of infinite possibilities together.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 MODEL PROBLEMS FOR FLOW IN POROUS MEDIA 2

1.2 THE KARHUNEN-LOÈVE EXPANSION 4

1.3 UNCERTAINTY QUANTIFICATION 7

1.4 MULTISCALE MORTAR MIXED FINITE ELEMENT METHOD 8

1.5 THESIS OUTLINE 10

2.0 STOCHASTIC COLLOCATION WITH MIXED FINITE ELEMENTS 11

2.1 STATISTICALLY STATIONARY RANDOM POROUS MEDIA 11

2.2 FORMULATION OF THE STOCHASTIC MIXED METHOD 13

2.2.1 Tensor Product Grid Collocation 16

2.3 ERROR ANALYSIS 19

2.4 NUMERICAL EXAMPLES FOR INCOMPRESSIBLE FLOW 23

2.4.1 Left-To-Right Example 24

2.4.2 Quarter Five-Spot Example 24

2.4.3 Discontinuous Permeability Example 28

2.5 NUMERICAL EXAMPLES FOR SLIGHTLY COMPRESSIBLE FLOW . . 31

2.5.1 Single-Phase Example 31

2.5.2 Two-Phase Example 32

3.0 IMPLEMENTATION OF A MORTAR MIXED FINITE ELEMENT METHOD USING A MULTISCALE FLUX BASIS 37

3.1 FORMULATION OF THE MORTAR MIXED FINITE ELEMENT METHOD 38

3.1.1 Interface Formulation and Iteration 42

3.2 THE MULTISCALE FLUX BASIS IMPLEMENTATION 44
3.3 NUMERICAL EXAMPLES 48
3.3.1 2-D Smooth Solution Example 49
3.3.2 2-D Rough Heterogeneous Permeability Example 54
3.3.3 3-D Smooth Solution Example 57
3.3.4 Mesh Adaptivity Example 60
4.0 STOCHASTIC COLLOCATION WITH MORTAR FINITE ELEMENTS USING MULTISCALE FLUX BASIS 62
4.1 STATISTICALLY NON-STATIONARY RANDOM POROUS MEDIA . . . 63
4.2 FORMULATION OF STOCHATIC MORTAR MIXED METHOD 65
4.2.1 Tensor Product and Sparse Grid Collocation 67
4.3 ERROR ANALYSIS 71
4.4 THREE COLLOCATION-MMMFEM ALGORITHMS 79
4.4.1 Collocation with Traditional MMMFEM 79
4.4.2 Collocation with a Deterministic Multiscale Flux Basis 80
4.4.3 Collocation with a Stochastic Multiscale Flux Basis 81
4.5 NUMERICAL EXAMPLES 83
4.5.1 L-Shape Example 85
4.5.2 Checkerboard Example 87
4.5.3 SPE10 Benchmark Example 90
4.5.4 Convergence Example 94
5.0 MULTISCALE PRECONDITIONER AND TWO-PHASE FLOWS . . 100
5.1 MULTISCALE PRECONDITIONER 101
5.2 FULLY-IMPLICIT MODEL 103
5.3 IMPES MODEL 105
5.4 NUMERICAL EXAMPLES 107
5.4.1 Fully-Implicit Example, SPE6 Permeability 108
5.4.2 IMPES Example, Constant Permeability 113
BIBLIOGRAPHY 115

LIST OF TABLES

2.1 Stochastic convergence results for Example 2.4.1. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses. 26

2.2 Deterministic convergence results for Example 2.4.1. Absolute errors reported against 160×160 grid; convergence ratios given in parentheses. 26

2.3 Stochastic convergence results for Example 2.4.2. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses. 28

2.4 Stochastic convergence results for Example 2.4.3. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses. 30

2.5 Deterministic Convergence results for Example 2.4.3. Absolute errors reported against 160×160 grid; convergence ratios given in parentheses. 30

3.1 Computational cost of Example 3.3.1 using continuous linear mortars with 3 elements per interface. 50

3.2 Computational cost of Example 3.3.1 using continuous quadratic mortars with 2 elements per interface. 51

3.3 Relative errors and computational cost for the fine scale solution in Example 3.3.1. 52

3.4 Relative error and computational cost for the multiscale solution using Method D3 with a single linear mortar per interface in Example 3.3.1. 53

3.5 Computational cost of Example 3.3.2 using continuous linear mortars with 2 elements per interface. 56

3.6 Computational cost of Example 3.3.2 using continuous quadratic mortars with 2 elements per interface. 56

3.7 Computational cost of Example 3.3.3 using linear mortars with one element per interface. 58
3.8 Computational cost of Example 3.3.3 using quadratic mortars with one element per interface. Relative residual CG tolerance = $1E-06$. 59
3.9 Computational cost of Example 3.3.3 using quadratic mortars with one element per interface. Relative residual CG tolerance = $1E-09$. 59
4.1 Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.1 with $N_{\text{term}} = 11$ and $N_{\text{term}} = 200$. Parenthesis denotes precomputation loop. 87
4.2 Comparison of runtime and linear systems across refinement levels 1-4 with the three collocation algorithms for Example 4.5.2. Parenthesis denotes precomputation loop. 91
4.3 Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.3 with $N_\Omega = 2$ KL Regions. Parenthesis denotes precomputation loop. 95
4.4 Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.3(c) with $N_\Omega = 20$ KL Regions. Parenthesis denotes precomputation loop. 95
4.5 Table of convergence in physical space for Example 4.5.4. Relative errors reported against finest grid level; convergence ratios given in parentheses. . . 98
5.1 Computational cost of Example 5.4.1 with Methods P1, P2, and P3 in the Fully-Implicit Model. 112
5.2 Computational cost of Example 5.4.2 with Methods P1 and P2 in the IMPES Model. 114

LIST OF FIGURES

2.1 A Monte-Carlo realization of the permeability field (left) and its corresponding solution (right) for Example 2.4.1 with 6 KL terms. 25

2.2 Expectation of solution (left), and variance of the pressure (right) for Example 2.4.1 with 4^6 collocation points. 25

2.3 Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.1 with 4^6 collocation points. 26

2.4 Expectation of solution (left), and variance of the pressure (right) for Example 2.4.2 with 5^6 collocation points. 27

2.5 Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.2 with 5^6 collocation points. 27

2.6 Expectation of solution (left), and variance of the pressure (right) for Example 2.4.3 with 5^6 collocation points. 29

2.7 Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.3 with 5^6 collocation points. 29

2.8 Upscaled SPE10 mean log-permeability field. 31

2.9 Mean pressure field and streamlines for Example 2.5.1 using the mean permeability (top-left), 100 Monte Carlo simulations (top-right), 2^4 collocation points (bottom-left), and 3^4 collocation points (bottom-right). 32

2.10 Standard deviation of pressure field for Example 2.5.1 using 100 Monte Carlo simulations (left), and 3^4 collocation points (right). 33

2.11 Mean oil pressure and streamlines for Example 2.5.2 using 100 Monte Carlo simulations (top-left) and 2^4 collocation points (top-right). Standard deviation of oil pressure using 100 Monte Carlo simulations (bottom-left) and 2^4 collocation points (bottom-right). 34
2.12 Mean water saturation for Example 2.5.2 using 100 Monte Carlo simulations (top-left) and 2^4 collocation points (top-right). Standard deviation of water saturation using 100 Monte Carlo simulations (bottom-left) and 2^4 collocation points (bottom-right). 35
2.13 The mean (left) and standard deviation (right) for cumulative oil production of Example 2.5.2 using 100 Monte Carlo simulations and 2^4 collocation points. 35
2.14 Mean pressure field and streamlines of Example 2.5.2 with 3^4 collocation points using $\sigma_Y = 0.25$ (top-left) and $\sigma_Y = 3$ (top-right). Standard deviation of pressure field using $\sigma_Y = 0.25$ (bottom-left) and $\sigma_Y = 3$ (bottom-right). . . . 36
2.15 Mean pressure field and streamlines of Example 2.5.2 using 3^4 collocation points with correlation lengths $\eta_1 = 0.16$, $\eta_2 = 0.23$ (left), and $\eta_1 = \eta_2 = 0.08$ (right). 36
3.1 Illustration of the Multiscale Flux Basis approach. 45
3.2 Computed pressure (color) and velocity (arrows) in Example 3.3.1: fine scale solution (left) and multiscale solution with a single linear mortar per interface (right). 53
3.3 Pressure error (left) and magnitude of velocity error (right) for the multiscale solution with a single linear mortar per interface for Example 3.3.1. 54
3.4 Permeability field (left), fine scale solution (middle), and multiscale solution with 3×5 subdomains and a single linear mortar per interface (right) for Example 3.3.2. 55
3.5 Computed multiscale solution (left) and its error (right) on $4 \times 4 \times 4$ subdomains with a single linear mortar per interface for Example 3.3.3. 58
3.6 Permeability field for Example 3.3.4 on mesh refinement level 4 (left) and its corresponding solution (right). Numbers indicate the total number of subdomain solves using the Multiscale Flux Basis implementation. 61

4.1 A Gauss-Hermite sparse grid (left) versus a Gauss-Hermite tensor grid (right) with a comparable number of points on each axis. 68

4.2 Subdomain and KL region layouts for Examples 4.5.1–4.5.4. Dashed lines represent subdomain boundaries and shading distinguishes between KL regions. 85

4.3 Realization of permeability (top-left), mean pressure (top-middle), mean velocity magnitude (top-right), variance of pressure (bottom-left), variance of horizontal velocity (bottom-middle), and variance of vertical velocity (bottom-right) for tensor product collocation in Example 4.5.1 with $N_{\text{term}} = 11$. 88

4.4 Realization of permeability (top-left), mean pressure (top-middle), mean velocity magnitude (top-right), variance of pressure (bottom-left), variance of horizontal velocity (bottom-middle), and variance of vertical velocity (bottom-right) for level $\ell_{\max} = 1$ sparse grid collocation in Example 4.5.1 with $N_{\text{term}} = 200$. 89

4.5 Spatial grids for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2. 91

4.6 Mean pressure (top) and pressure variance (bottom) for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2. 92

4.7 Mean velocity magnitude (top) and velocity magnitude variance (bottom) for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2. 92

4.8 Realization of permeability (left) and its corresponding solution (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions. 96

4.9 Mean solution (left) and Pressure Variance (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions. 96

4.10 X-Velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions. 96

4.11 Y-velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions. 97

4.12 Z-velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions. 97

4.13 Log-log plot of convergence in stochastic space for Example 4.5.4. Different types of sampling methods are shown in absolute L^2-error for pressure with $1E-6$ tolerance (left) and $1E-15$ tolerance (right). 99
4.14 Log-log plot of convergence in stochastic space for Example 4.5.4. Different types of sampling methods are shown in absolute $H(div)$-error for velocity with $1E-6$ tolerance (left) and $1E-15$ tolerance (right). 99
5.1 Flow charts for Methods P1, P2, and P3 in the Fully-Implicit Model. Right columns include steps that may be parallelized. 108
5.2 X-component of SPE6 Permeability data in millidarcies for Example 5.4.1. . 109
5.3 Oil relative permeability (left), water relative permeability (middle), and capillary pressure (right) versus water saturation for Example 5.4.1. 109
5.4 Plots for Example 5.4.1: Water Pres. at $t = 100$ (top-left), Oil Conc. at $t = 100$ (top-right), Water Pres. at $t = 200$ (bottom-left), Oil Conc. at $t = 200$ (bottom-right). 110
5.5 Interface-GMRES iterations versus time for Example 5.4.1 with Methods P1, P2, and P3 in the Fully-Implicit Model. 112
5.6 Interface-GMRES iterations versus time for Example 5.4.2 with Methods P1 and P2 in the IMPES Model. 114

LIST OF ALGORITHMS

3.1 Original MMMFEM Interface Iteration 44

3.2 Formation of a Multiscale Flux Basis 46

3.3 Interface Iteration with Multiscale Flux Basis 47

4.1 A natural ordering for sparse grid points 70

4.2 Method S1. Collocation without a Multiscale Flux Basis 80

4.3 Method S2. Collocation with a Deterministic Multiscale Flux Basis 81

4.4 Method S3. Collocation with a Stochastic Multiscale Flux Basis 82

4.5 Global to local index conversion for a tensor product grid 83

4.6 Global to local index conversion for a sparse grid 84

1.0 INTRODUCTION

A porous medium is a solid matrix with a large amount of microscopic pores and throats, in which there are narrow tubes that fluid can pass through. Example solids include sand, soil, and rock, and example fluids include oil, water, and gas. Darcy's Law refers to the basic constitutive relationship for the conservation of momentum of fluids in porous media. It is a potential theory discovered in 1856 that states fluids flow from areas of high pressure to areas of low pressure [28]. Today's modern mathematical models and numerical methods for porous media flow exist in increasingly more complex varieties, designed to handle multiple phases, multiple components, solute transport, and coupling with other physical processes [84, 21]. Applications involving flow through porous media include hydrologists studying underground aquifers for such purposes as groundwater remediation, and petroleum engineers simulating reservoirs for petroleum production. As we progress through the 21st century, matters of energy and the environment are of great importance to humanity.

The work in this thesis has twofold motivation related to the difficulties in the numerical simulation of these physical processes. First, both the difficulty in describing the media itself and the phenomena describing the fluids occupying the void spaces lead to the continuum approach for modeling flow through porous media [13]. Inherent to this approach are uncertainties associated with both natural randomness and incomplete knowledge of physical properties in model input. As computational power increases, there is great interest in the forward propagation of stochastic noise to model output, via techniques known as Uncertainty Quantification (UQ) methods. Second, for large reservoir simulation, about 80-90% of total runtime is spent on the solution of linear systems [21]. Specific emphasis needs to be placed on the parallel solvers to make them as efficient as possible. Realistic deterministic simulations are computationally intensive, and adding stochastic parameters exponentially

increases these demands.

1.1 MODEL PROBLEMS FOR FLOW IN POROUS MEDIA

We consider two deterministic mathematical models involving Darcy's Law for the conservation of momentum coupled with a conservation of mass. These models will also be extended to allow the possibility of stochastic permeability. Let $D \subset \mathbb{R}^d$ be a bounded domain in $d = 2$ or 3 spatial dimensions, with Lipschitz boundary and outer unit normal $\mathbf{n}$.

In the following, C denotes a generic positive constant independent of the discretization parameters. For a subspace $G \subseteq D$, the $L^2(G)$ inner product and norm are denoted $(\cdot,\cdot)_G$ and $\|\cdot\|_G$, respectively. Let $\|\cdot\|_{m,p,G}$ denote the norm of the Sobolev space $W^{m,p}(G)$, and in the case of Hilbert space $H^m(G) = W^{m,2}(G)$ the norm is denoted by $\|\cdot\|_{m,G}$. Define the space $H(div;G) = \{\mathbf{v} \in (L^2(G))^d \mid \nabla\cdot\mathbf{v} \in L^2(G)\}$. We may omit G in any subscript if $G = D$. For a section of the domain or element boundary $S \subset \mathbb{R}^{d-1}$ we write $\langle\cdot,\cdot\rangle_S$ and $\|\cdot\|_S$ for the $L^2(S)$ inner product (or duality pairing) and norm, respectively. Dual spaces are denoted by $(\cdot)^*$.

The first model is for single-phase, incompressible, non-gravitational flow. It is a linear, steady-state, second-order, elliptic equation written in mixed form. We consider both Dirichlet and Neumann boundary conditions with $\partial D = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. The system is:

$$\mathbf{u} = -K\nabla p, \qquad \text{in } D, \tag{1.1}$$

$$\nabla\cdot\mathbf{u} = f, \qquad \text{in } D, \tag{1.2}$$

$$p = g_D, \qquad \text{on } \Gamma_D, \tag{1.3}$$

$$\mathbf{u}\cdot\mathbf{n} = g_N, \qquad \text{on } \Gamma_N. \tag{1.4}$$

The two unknowns are the fluid pressure $p(\mathbf{x})$ (hydraulic head), and the Darcy velocity $\mathbf{u}(\mathbf{x})$. Here $K(\mathbf{x})$ represents the absolute permeability (hydraulic conductivity) divided by the fluid viscosity, and can either be a positive scalar-valued function, or a uniformly symmetric

positive definite 2-tensor with components in $L^\infty(D)$. Our typical regularity assumptions are that $f(\mathbf{x}) \in L^2(D)$, $g_D(\mathbf{x}) \in H^{1/2}(\Gamma_D)$, and $g_N(\mathbf{x}) \in L^2(\Gamma_N)$.

The second model is for unsteady, two-phase, immiscible, incompressible or slightly-compressible flow. It is a highly nonlinear, transient system which can be reformulated as a coupled system of a hyperbolic equation and a degenerate parabolic equation. Denote fluid phases by $\alpha = o$ (oil) and $\alpha = w$ (water), assume Dirichlet boundary conditions, and denote the time interval $J = [0, T]$. For each phase, the system is:

$$\mathbf{u}_\alpha = \frac{-K\ k_\alpha(S_\alpha)}{\mu_\alpha} \left(\nabla p_\alpha - \rho_\alpha g \nabla z\right), \qquad \text{in } D \times J, \tag{1.5}$$

$$\phi \frac{\partial (S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = \rho_\alpha q_\alpha, \qquad \text{in } D \times J, \tag{1.6}$$

$$p_\alpha = g_{\alpha,D}, \qquad \text{on } \partial D \times J, \tag{1.7}$$

$$p_\alpha = g_{\alpha,0}, \qquad \text{on } D \times \{t = 0\}. \tag{1.8}$$

The unknowns are the phase pressures $p_\alpha(\mathbf{x}, t)$, the Darcy velocities $\mathbf{u}_\alpha(\mathbf{x}, t)$, and the phase saturations $S_\alpha(\mathbf{x}, t)$. Known data includes the absolute permeability $K(\mathbf{x})$, the phase relative permeabilities $k_\alpha(S_\alpha)$, the porosity $\phi(\mathbf{x})$, the phase densities ρ_α, the phase viscosities μ_α, and the gravitational constant g and depth z. The source functions $q_\alpha(\mathbf{x}, t)$ represent well injection and production rates, defined using the Peaceman model [68] extended to multiphase flow. In immiscible flow, the presence of one fluid (its saturation) will displace the other and inhibit its ability to flow. Moreover, a known capillary pressure measures the surface tensions between the two fluids when they occupy the same pore space. Therefore this system of four equations in six unknowns is closed with constitutive constraints for the balance of volume and the capillary pressure relation

$$S_o + S_w = 1, \tag{1.9}$$

$$p_c(S_w) = p_o - p_w. \tag{1.10}$$

In the case of two-phase, slightly compressible flow, the densities are modeled by the equations of state for $\alpha = o, w$

$$\rho_\alpha = \rho_\alpha^{ref} \exp\left[c_\alpha (p_\alpha - p_\alpha^{ref})\right], \tag{1.11}$$

where the ρ_α^{ref} are reference densities, the p_α^{ref} are reference pressures, and c_α are compressibility coefficients. In the case of single-phase, slightly compressible flow, the oil saturation $S_o = 0$, so that equations (1.5)-(1.8) reduce to

$$\mathbf{u} = \frac{-K}{\mu}\left(\nabla p - \rho_w g \nabla z\right), \qquad \text{in } D \times J, \tag{1.12}$$

$$\phi\frac{\partial(\rho_w)}{\partial t} + \nabla \cdot (\rho_w \mathbf{u}_\alpha) = \rho_w q, \qquad \text{in } D \times J, \tag{1.13}$$

$$p = g_D, \qquad \text{on } \partial D \times J, \tag{1.14}$$

$$p = g_0, \qquad \text{on } D \times \{t = 0\}. \tag{1.15}$$

We retain the subscript w on water density to prevent conflict with future notation on probability density functions. In this model we assume that there exists some positive constant α such that $\phi \in L^\infty(D)$ and $1/\alpha \leq \phi(\mathbf{x}) \leq \alpha$. Furthermore, we assume $\rho_w(\mathbf{x}) \in W^{2,\infty}(\mathbb{R})$ and $1/\alpha \leq \rho_w, \rho_w', \rho_w'' \leq \alpha$. Note that if we neglect the gravity term and assume incompressibility, we return to the structure of the first model.

1.2 THE KARHUNEN-LOÈVE EXPANSION

In general, random noise may be added to any type of input in a deterministic model, such as boundary conditions, geometry, parameters, or functions. When more than one random input exists, they may even be mutually correlated. In this work however, we merely allow the permeability coefficient K to be the sole stochastic input in both models.

More specifically, let $(\Omega, \mathcal{F}, P)$ be a complete probability space, where Ω is a set of outcomes, $\mathcal{F} \subset 2^\Omega$ is a σ-algebra of events, and $P : \mathcal{F} \to [0, 1]$ is a probability measure. In our stochastic models, the permeability $K : D \times \Omega \to \mathbb{R}$ is a spatially correlated random field. It is scalar-valued, as opposed to our deterministic models when it can be a full 2-tensor.

A stochastic input will lead to a stochastic output, so the goal is called Uncertainty Quantification. This refers to the computation of the stochastic solution's statistical moments such as its expectation and variance. A random variable $\xi : \Omega \to \mathbb{R}$ is defined as an

$\mathcal{F}$-measurable function with a Probability Distribution Function (PDF) $\rho(y)$. It's expectation and variance are defined by

$$E[\xi] = \int_\Omega \xi(\omega) dP(\omega) = \int_{\mathbb{R}} y\rho(y) dy, \tag{1.16}$$

$$\mathrm{Var}[\xi] = E[\xi^2] - E[\xi]^2. \tag{1.17}$$

The calculation of variance gives an important measure of confidence in the expectation, and identifies the specific areas of the solution that are most sensitive to the stochastic input parameters. To ensure the positive definiteness of the permeability coefficient almost surely, we give random expressions for its mean-removed logarithm Y' satisfying the relations

$$Y' = Y - E[Y], \qquad Y = \ln(K), \qquad K = \exp(Y). \tag{1.18}$$

The main tool that we use to represent the stochastic permeability is a spectral series known as a Karhunen-Loève (KL) expansion [57]. It has the form

$$Y'(\mathbf{x}, \omega) = \sum_{j=1}^{\infty} \xi_j(\omega) \sqrt{\lambda_j} f_j(\mathbf{x}), \tag{1.19}$$

where $\{\xi_j\}$ is a set of independent identically distributed (iid) random variables, $\{\lambda_j\}$ is a set of eigenvalues, and $\{f_j\}$ is an orthonormal set of deterministic eigenfunctions. This expansion can be thought of as the stochastic equivalent of a Fourier series, where the coefficients are the random variables, and the orthogonal basis functions are determined by the covariance of the random process. It is our choice to use this expression, and it should be noted that there exist other possibilities such as the Polynomial Chaos expansion [41, 90]. A useful comparison is made between both types of representations in [32].

Our stochastic models will assume that we are given one (or more) known covariance functions

$$C_Y(\mathbf{x}, \bar{\mathbf{x}}) = E[Y'(\mathbf{x}, \omega) Y'(\bar{\mathbf{x}}, \omega)], \tag{1.20}$$

which are by definition bounded, symmetric, and positive definite. When this function is regarded as the resolvent kernel of the Fredholm integral equation

$$\int_D C_Y(\mathbf{x}, \bar{\mathbf{x}}) f(\mathbf{x}) d\mathbf{x} = \lambda f(\bar{\mathbf{x}}), \tag{1.21}$$

it can be decomposed into the series expansion

$$C_Y(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{j=1}^{\infty} \lambda_j f_j(\mathbf{x}) f_j(\bar{\mathbf{x}}). \tag{1.22}$$

Indeed, as a consequence of Mercer's Theorem [73, p.245], there exists a set of orthonormalized eigenfunctions such that $\int_D f_i(\mathbf{x}) f_j(\mathbf{x}) d\mathbf{x} = \delta_{ij}$, they form a complete spanning set, and the convergence is uniform and absolute. (See, also [25]). Using this fact, one can verify the expression for (1.19) is correct by substituting it into expression (1.20) to obtain (1.22).

It is known that the eigenvalues in this type of series typically decay quite rapidly, which motivates one to truncate the KL series after a finite number of terms. This is a modeling error, and the goal of this work is not to answer the question of how many terms one will need in the series *a priori*. This question has been addressed in the adaptive methods of such works as [35, 64, 63, 61]. Instead, we focus on the discretization error in stochastic and physical space, as well as an optimized parallel implementation of such an approach. We have found that increasing the number of terms will increase the permeability's heterogeneity. The images of the remaining random variables form a finite-dimensional vector space, so that the stochastic system is transformed into a higher-dimensional deterministic system, with the extra dimensions coming from the number of terms in the KL series.

This work also explores the possibility of statistically nonstationary random porous media. By stationary random porous media, we mean that there is a single global probability space and a single global KL expansion. By nonstationary random porous media, we mean that there are several independent probability spaces, each of which has its own independent KL expansion. We refer to these statistically independent regions as KL Regions. The eigenfunctions in each KL expansion are extended by zero in the other KL Regions, so that a global KL expansion may be written as a sum of the other series. This approach is motivated by modeling porous media with multiple rock types.

1.3 UNCERTAINTY QUANTIFICATION

UQ methods can be classified in three major groups: (1) sampling methods [33, 55], (2) moment/perturbation methods [52, 59, 95, 50, 74] and, (3) non-perturbative methods, either based on polynomial chaos expansions [41, 40, 90, 91, 92, 45] or stochastic finite element methods [41, 29, 12]. A brief survey of these methods can be found in [80], where an extensive reference list is given. In such order, we can say that these methods range from being non-intrusive to very intrusive in terms of modification to the deterministic simulation model. In certain situations these techniques may even be combined together to form hybrid approaches [24]. The best known sampling method is Monte Carlo simulation (MCS) [33], which involves repeated generation of random samplings (realizations) of input parameters followed by the application of the simulation model in a "black box" fashion to generate the corresponding set of stochastic responses. These responses are further analyzed to yield statistical moments or distributions. The major drawback of MCS is the high computational cost due to the need to generate valid representative statistics from a large number of realizations at a high resolution level.

Moment/perturbation and finite element stochastic methods fall into the category of non-sampling methods. These methods are suitable for systems with relatively small dimensions of random inputs. However, despite the apparent accuracy and mild cost with respect to MCS, these methods also present some limitations that have prevented them from being widely used. The problem is that their semi-intrusive or fully intrusive character may greatly complicate the formulation, discretization and solution of the model equations, even in the case of linear and stationary input distributions. There is also a high computational cost associated with these methods since the number of terms needed to accurately represent the propagation of uncertainties grows significantly with respect to the degree of variability of the system. It is still not clear how these methods may be formulated in the event of high nonlinearities due to complex flow and chemical reactions over arbitrary geometries.

On the other hand, stochastic finite elements exhibit fast convergence through the use of generalized polynomial chaos representations of random processes. These generalizations of the Wiener-Hermite polynomial chaos expansion can include a wider class of random

processes by means of global polynomial expansions, piecewise polynomial expansions and wavelet basis expansions, see *e.g.* [12, 81]. However, besides their very intrusive feature, the dimensionality of the discretized stochastic finite element equations can be dramatically larger than the dimensionality of the base case deterministic model.

This thesis utilizes a UQ technique called the stochastic collocation method [11, 89, 88], which is a very promising approach for improving the efficiency of non-sampling methods. It combines a finite element discretization in physical space with a collocation at specially chosen points in probability space. As a result a sequence of uncoupled deterministic problems need to be solved, just like in MCS. However, the stochastic collocation method shares the approximation properties of the stochastic finite element method, making it more efficient than MCS. Choices of collocation points include tensor product of zeros of orthogonal polynomials [11, 89], sparse grid approximations [35, 61, 64, 76, 89], and probabilistic collocation [56]. The last two provide approaches to reduce the number of collocation points needed to obtain a given level of approximation, leading to very efficient algorithms. This thesis considers collocation on both tensor product and sparse grids.

1.4 MULTISCALE MORTAR MIXED FINITE ELEMENT METHOD

The use of Mixed Finite Element (MFE) methods is advantageous for its simultaneous high-order approximation of both the primary variable and a second variable of physical interest. Since the 1970's, a robust theory has been developed to produce stable schemes for subsurface flow, as well as applications in surface flow, electromagnetism, and elasticity [14]. Moreover these methods provide physical fidelity via the element-wise conservation of mass, a property that standard Galerkin finite element methods do not possess. However, difficulties arise in porous media flow applications, where the domain is quite large and the permeability tensor varies on a fine scale. Resolving the solution on the fine scale is often computationally infeasible, necessitating the use of multiscale approximations.

The Multiscale Mortar Mixed Finite Element Method (MMMFEM) was proposed in [9] as an alternative to existing multiscale methods, such as the variational multiscale method

[48, 49, 6, 1, 7, 4] and multiscale finite elements [46, 47, 31, 20, 51, 3]. The latter two approaches are closely related [7]. In all three methods the domain is decomposed into a series of small subdomains (coarse grid), and the solution is resolved globally on the coarse grid and locally (on each coarse element) on a fine grid. Non-overlapping domain decomposition has a long history that predates digital computers, when in 1870 Schwarz proved the convergence of his alternating method for elliptic equations [77]. All three multiscale methods are based on a divide and conquer approach: solving relatively small fine scale subdomain problems that are only coupled together through a reduced number (coarse scale) degrees of freedom.

The variational multiscale method and multiscale finite elements both compute a multiscale basis by solving local fine scale problems with boundary conditions or a source term corresponding to the coarse scale degrees of freedom. This basis is then used to solve the coarse scale problem. The MMMFEM uses a non-overlapping domain decomposition algorithm which introduces a Lagrange multiplier space on the subdomain interfaces to weakly impose certain continuity conditions. By eliminating the subdomain unknowns the global fine scale problem is reduced to an interface problem, which is solved using an iterative method. The domain decomposition algorithm was originally developed for the case of matching grids [42] and then extended to the case of non-matching grids using mortar finite elements [93, 8]. This generalization allows for extremely flexible finite element partitions, as both the fine scale elements across subdomain interfaces and the subdomains themselves (i.e., the coarse grid) may be spatially non-conforming. Moreover, one has the ability to vary the interface degrees of freedom [86, 70, 9]. If only a single mortar grid element is used per interface, the resulting approximation is comparable to the one in the variational multiscale method or multiscale finite elements. In the MMMFEM framework, *a posteriori* error estimators [87] can be employed to adaptively refine the mortar grids where necessary to improve the global accuracy. Furthermore, higher order mortar approximation can be used to compensate for the coarseness of the mortar grid and obtain fine scale convergence of the error [9]. Thus, the MMMFEM is more flexible than the variational multiscale method and multiscale finite elements. Another observation is that the MMMFEM resolves the flux through the coarse interfaces on the fine scale, which is not the case for the other two approaches.

Both the variational multiscale method and multiscale finite elements have recently been

applied to Uncertainty Quantification in such works as [10, 10, 35] and [30, 2], respectively. Much of the work done for this thesis [36, 39] marks the first time that the Multiscale Mortar Mixed Finite Element Method has been extended to UQ in an analogous way.

1.5 THESIS OUTLINE

The rest of this thesis is organized as follows: In Chapter 2 we formulate a method for problems with stationary stochastic permeability, which couples a MFE spatial discretization with stochastic collocation on a tensor product grid. In Chapter 3 we develop an algorithm called the Multiscale Flux Basis implementation to solve the interface system associated with the MMMFEM, in a deterministic setting. In Chapter 4 we both extend and combine the previous works by coupling the MMMFEM with stochastic collocation on both tensor product and sparse grids, allowing nonstationary stochastic permeability, and developing algorithms to solve the resulting system using Multiscale Flux Basis ideas. In Chapter 5 we present a concurrent approach that uses the Multiscale Flux Basis as an interface preconditioner, and describe its application to stochastic collocation as well as two-phase flow simulations in both fully-implicit and IMPES models.

2.0 STOCHASTIC COLLOCATION WITH MIXED FINITE ELEMENTS

This chapter contains material that was published in [36], in which methods were formulated to combine Mixed Finite Element discretizations in physical space with the stochastic collocation method. We consider both the incompressible single-phase model (1.1)–(1.4) and the slightly compressible single (1.12)–(1.15) and two-phase model (1.5)–(1.8), coupled with stochastic collocation on tensor product grids. Convergence analysis for the pressure and the velocity is presented, which follows the approach in [11] where standard Galerkin discretizations are studied. We show that the total error can be decomposed into the sum of deterministic and stochastic errors. Optimal convergence rates and superconvergence for the pressure are established for the deterministic error. The stochastic error converges exponentially with respect to the number of the collocation points. Numerical experiments confirm theoretical convergence rates, and demonstrate the efficiency to our approach compared to MCS. A model for statistically stationary random porous media is discussed in Section 2.1, the MFE stochastic collocation method is formulated in Section 2.2, it is analyzed in Section 2.3, and some numerical experiments are presented in Sections 2.4 and 2.5.

2.1 STATISTICALLY STATIONARY RANDOM POROUS MEDIA

Assume that the KL expansion for Y' is determined by the following covariance function, given in $d = 2$ or 3 spatial dimensions by

$$C_Y(\mathbf{x}, \bar{\mathbf{x}}) = \sigma_Y^2 \prod_{i=1}^{d} \exp\left[\frac{-|x_i - \bar{x}_i|}{\eta_i}\right]. \tag{2.1}$$

Here σ_Y is a variance, and η_i are correlation lengths for the physical dimensions. This covariance has been considered in such works as [96]. This type of random porous media is statistically stationary, meaning that the covariance between any two points in the domain depends on their distance only, rather than the actual locations of these two points. We also assume that Y' is a Gaussian process, so that the $\{\xi_j\}$ are Normal $N(0,1)$ iid random variables, meaning $E[\xi_j] = 0,\ E[\xi_i \xi_j] = \delta_{ij}$, and each has the PDF

$$\rho_j(y) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{y^2}{2}\right]. \tag{2.2}$$

In this chapter, we require the Lipschitz domain D to be a product of intervals. This fact, along with the separability of the covariance function, allows one-dimensional eigenvalues and eigenfunctions to be obtained in each spatial dimension $i = 1, \ldots, d$ separately. They are computed by solving the Fredholm integral equations (1.21) on intervals $[0, L_i]$. We employ the procedure that solves for these analytically as presented in Appendix A of [96], in which they are found to be

$$\lambda_j^{(i)} = \frac{2\eta_i \sigma_Y^2}{\eta_i^2 w_j^2 + 1} \quad \text{and} \quad f_j^{(i)}(x_i) = \frac{\eta_i w_j \cos(w_j x) + \sin(w_j x)}{\sqrt{(\eta_i^2 w_j^2 + 1) L_i/2 + \eta_i}}, \tag{2.3}$$

where w_j are the positive roots of the characteristic equation

$$(\eta_i^2 w^2 - 1)\sin(wL_i) = 2\eta_i w \cos(wL_i). \tag{2.4}$$

The one-dimensional eigenvalues and eigenfunctions are then multiplied across all spatial dimensions, and then sorted in such a way to form a nonincreasing series in λ_j. We note that not all types of covariance functions give closed-form expressions, and that efficient methods for numerically computing the KL expansion are reported in [76].

As is typically done at this point, we commit a modeling error that replaces the stochastic problem by a higher dimensional deterministic approximation.

Assumption 2.1.1. *(Finite Dimensional Noise Assumption). The KL Expansion for Y' is truncated after N_{term} terms, which allows us to approximate (1.19) by*

$$Y'(\mathbf{x}, \omega) \approx \sum_{j=1}^{N_{term}} \xi_j(\omega) \sqrt{\lambda_j} f_j(\mathbf{x}). \tag{2.5}$$

This is feasible to do as the eigenvalues λ_j in (2.3) decay at a rate which is asymptotic to $O(1/j^2)$ [96]. This allows us to write $Y(\mathbf{x},\omega) \approx Y(\mathbf{x},\xi_1(\omega),\ldots,\xi_{N_{\text{term}}}(\omega))$. The images of the random variables $\mathbb{S}_j = \xi_j(\Omega)$ make up the finite dimensional vector space

$$\mathbb{S} = \prod_{i=1}^{N_{\text{term}}} \mathbb{S}_j \subseteq \mathbb{R}^{N_{\text{term}}}. \tag{2.6}$$

If ρ_j corresponds to the PDF of each ξ_j, then the joint PDF for the random vector $(\xi_1,\ldots,\xi_{N_{\text{term}}})$ is defined to be $\rho = \prod_{j=1}^{N_{\text{term}}} \rho_j$. Finally, we may write $Y(\mathbf{x},\omega) \approx Y(\mathbf{x},\mathbf{y})$, where $\mathbf{y} = (y_1,\ldots,y_{N_{\text{term}}})$ and $y_i = \xi_i(\omega)$.

2.2 FORMULATION OF THE STOCHASTIC MIXED METHOD

Consider the single-phase model (1.1)-(1.4) with stochastic permeability $K = K(\mathbf{x},\omega)$ defined by covariance (2.1). The stochastic problem is to find $\mathbf{u}$ and p that satisfy this system for P-almost every (*a.e.*) $\omega \in \Omega$

$$\begin{aligned}
\mathbf{u} &= -K(\mathbf{x},\omega)\nabla p, && \text{in } D, && (2.7)\\
\nabla\cdot\mathbf{u} &= f, && \text{in } D, && (2.8)\\
p &= g_D, && \text{on } \Gamma_D, && (2.9)\\
\mathbf{v}\cdot\mathbf{n} &= g_N, && \text{on } \Gamma_N. && (2.10)
\end{aligned}$$

By the Doob-Dynkin Lemma [65, Lemma 2.1.2], the solution $(\mathbf{u},p)$ is $\mathcal{F}$-measurable, so we can write $\mathbf{u} = \mathbf{u}(\mathbf{x},\omega)$ and $p = p(\mathbf{x},\omega)$. Next, admit Assumption 2.1.1 so that $K(\mathbf{x},\omega) \approx K(\mathbf{x},\mathbf{y})$. It follows that we can describe the solution using the same random approximation, *i.e.*

$$\begin{aligned}
\mathbf{u}(\mathbf{x},\omega) &\approx \mathbf{u}(\mathbf{x},\xi_1(\omega),\ldots,\xi_{N_{\text{term}}}(\omega)) = \mathbf{u}(\mathbf{x},y_1,\ldots,y_{N_{\text{term}}}) \quad \text{and}\\
p(\mathbf{x},\omega) &\approx p(\mathbf{x},\xi_1(\omega),\ldots,\xi_{N_{\text{term}}}(\omega)) = p(\mathbf{x},y_1,\ldots,y_{N_{\text{term}}}).
\end{aligned}$$

In a mixed method, the Neumann boundary condition (1.4) is imposed essentially into the velocity space. We define the deterministic pressure and velocity spaces as

$$W(D) = L^2(D) \quad \text{and} \quad \mathbf{V}^\gamma(D) = \{\mathbf{v} \in H(div; D) \mid \mathbf{v} \cdot \mathbf{n} = \gamma \text{ on } \partial D \cap \Gamma_N\}, \tag{2.11}$$

where $\gamma \in L^2(\Gamma_N)$. Note that the condition $\mathbf{v} \cdot \mathbf{n} = \gamma$ requires slightly higher regularity than the usual normal traces of functions in $H(div; D)$. Since our goal is to compute stochastic solutions with finite second moment, we define the space

$$L^2_\rho(\mathbb{S}) = \left\{ \mathbf{v} : \mathbb{S} \to \mathbb{R}^d \mid \left(\int_{\mathbb{S}} \|\mathbf{v}(\mathbf{y})\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} < \infty \right\}, \tag{2.12}$$

and take its vector product with the aforementioned deterministic spaces to form the stochastic Sobolev spaces

$$W(D, \mathbb{S}) = W(D) \otimes L^2_\rho(\mathbb{S}) \quad \text{and} \quad \mathbf{V}^\gamma(D, \mathbb{S}) = \mathbf{V}^\gamma(D) \otimes L^2_\rho(\mathbb{S}).$$

Whenever the explicit dependence in parentheses is omitted, it is implied that we mean the stochastic spaces, *e.g.* $W = W(D, \mathbb{S})$. We equip the stochastic pressure and velocity spaces with norms

$$\|\mathbf{v}\|^2_V = \int_{\mathbb{S}} \left(\int_D \left(|\mathbf{v}|^2 + |\nabla \cdot \mathbf{v}|^2 \right) d\mathbf{x} \right) \rho(\mathbf{y}) d\mathbf{y} = E\left[\|\mathbf{v}\|^2_{H(div;D)} \right] \quad \text{and} \tag{2.13}$$

$$\|w\|^2_W = \int_{\mathbb{S}} \left(\int_D w^2 d\mathbf{x} \right) \rho(\mathbf{y}) d\mathbf{y} = E\left[\|w\|^2_{L^2(D)} \right]. \tag{2.14}$$

Multiplication by test functions and application of Green's Identity leads to the stochastic equivalent of a standard dual mixed weak formulation.[1] That is to find $\mathbf{u} \in \mathbf{V}^{g_N}$ and $p \in W$ such that

$$\int_{\mathbb{S}} (K(\mathbf{x}, \mathbf{y})^{-1} \mathbf{u}, \mathbf{v}) \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \left[(p, \nabla \cdot \mathbf{v}) - \langle g_D, \mathbf{v} \cdot \mathbf{n} \rangle_{\Gamma_D} \right] \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall \mathbf{v} \in \mathbf{V}^0, \tag{2.15}$$

$$\int_{\mathbb{S}} (\nabla \cdot \mathbf{u}, w) \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} (f, w) \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall w \in W. \tag{2.16}$$

Next we employ a mixed finite element discretization in physical dimensions to pose a semidiscrete stochastic formulation. Let $\mathcal{T}_h$ be a shape-regular affine finite element partition

[1] Here, "dual" refers to the duality $(grad)^* = (div)$, as opposed to a primal H^1-mixed form. We say this, because recently "dual" may also refer to three variable mixed methods with two-fold saddle point problems.

of the spatial domain D [23]. A mixed finite element discretization $\mathbf{V}_h(D) \times W_h(D) \subset H(div, D) \times L^2(D)$ is chosen to satisfy a discrete *inf-sup* condition [18]. The semidiscrete formulation of (2.15)-(2.16) will be to find $\mathbf{u}_h : \mathbb{S} \to \mathbf{V}_h(D)$ and $p_h : \mathbb{S} \to W_h(D)$ such that for *a.e.* $\mathbf{y} \in \mathbb{S}$,

$$\begin{aligned} (K(\mathbf{x},\mathbf{y})^{-1}\mathbf{u}_h, \mathbf{v}_h) &= (p_h, \nabla \cdot \mathbf{v}_h) - \langle g_D, \mathbf{v}_h \cdot \mathbf{n} \rangle_{\Gamma_D}, \quad \forall \mathbf{v}_h \in \mathbf{V}_h(D), \qquad (2.17) \\ (\nabla \cdot \mathbf{u}_h, w_h) &= (f, w_h), \quad \forall w_h \in W_h(D). \qquad (2.18) \end{aligned}$$

By the general saddle point problem theory [18], a solution to the mixed problem exists and it is unique for a.e. $\mathbf{y} \in \mathbb{S}$. This can be extended to the stochastic problem by the argument in [75], which analogously uses the Lax-Milgram Lemma for the non-mixed elliptic problem.

Remark 2.2.1. *In this work, we may consider any of the usual mixed finite element spaces, such as the RTN spaces [72, 62], BDM spaces [17], BDFM spaces [16], BDDF spaces [15], CD spaces [19], or those listed in [14]. Let the space of polynomials of total degree k be denoted $\mathbb{P}_k = \{p \in \mathbb{P} \mid p = \sum_{0 \le i_1+\ldots+i_d \le k} \alpha_{i_1,\ldots,i_d} x_i^{i_1} \cdots x_d^{i_d}\}$. On each element $E \in \mathcal{T}_h$, the velocity space contains vector polynomials of total degree $r \ge 0$, $(\mathbb{P}_r(E))^d \subset \mathbf{V}_h(E)$. Velocity normal components on each edge (face) contain scalar polynomials of total degree r, $\mathbb{P}_r(\gamma) \subset \mathbf{V}_h(E) \cdot \mathbf{n}$. The pressure space contains scalar polynomials of total degree s, $\mathbb{P}_s(E) \subset W_h(E)$. In all of the above mixed spaces, $s = r$, or $s = r - 1$ when $r \ge 1$. Our numerical experiments use lowest order RTN space on logically rectangular grids.*

Similarly, we also consider the two-phase model problem (1.5)-(1.8) with the same stochastic permeability. Define the time-dependent Stochastic Sobolev spaces

$$\begin{aligned} \mathbf{V}_J(D,\mathbb{S}) &= H(div; D) \otimes L^p(J) \otimes L^2_\rho(\mathbb{S}), \\ W_J(D,\mathbb{S}) &= L^2(D) \otimes L^p(J) \otimes L^2_\rho(\mathbb{S}). \end{aligned}$$

with norms

$$\begin{aligned} \|\mathbf{v}\|^2_{V_J} &= \int_{\mathbb{S}} \left(\int_J \left(\int_D \left(|\mathbf{v}|^2 + |\nabla \cdot \mathbf{v}|^2 \right) d\mathbf{x} \right)^p dt \right)^{1/p} \rho(\mathbf{y}) d\mathbf{y} \quad \text{and} \\ \|w\|^2_{W_J} &= \int_{\mathbb{S}} \left(\int_J \left(\int_D w^2 d\mathbf{x} \right)^p dt \right)^{1/p} \rho(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

The stochastic variational formulation is to find $\mathbf{u} \in \mathbf{V}_J$ and $p \in W_J$ such that

$$\int_{\mathbb{S}} \left(\left(\frac{K(\mathbf{x},\mathbf{y}) k_i(S_w)}{\mu_i} \right)^{-1} \mathbf{u}_i, \mathbf{v} \right) \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \Big[(p_i, \nabla \cdot \mathbf{v}) - (\rho_i g \nabla z, \mathbf{v}) - \langle g_{i,D}, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial D} \Big] \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall \mathbf{v} \in \mathbf{V}_J, \tag{2.19}$$

$$\int_{\mathbb{S}} \left[\left(\phi_i \frac{\partial (S_i \rho_i)}{\partial t}, w \right) + (\nabla \cdot (\rho_i \mathbf{u}_i), w) \right] \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} (f_i, w) \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall w \in W_J. \tag{2.20}$$

In the single-phase, slightly compressible case, this once again reduces to finding $\mathbf{u} \in \mathbf{V}_J$ and $p \in W_J$ such that

$$\int_{\mathbb{S}} \left(\left(\frac{K(\mathbf{x},\mathbf{y})}{\mu} \right)^{-1} \mathbf{u}, \mathbf{v} \right) \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \Big[(p, \nabla \cdot \mathbf{v}) - (\rho_w g \nabla z, \mathbf{v}) - \langle g_D, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial D} \Big] \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall \mathbf{v} \in \mathbf{V}_J, \tag{2.21}$$

$$\int_{\mathbb{S}} \left[\left(\phi \frac{\partial \rho_w}{\partial t}, w \right) + (\nabla \cdot (\rho_w \mathbf{u}), w) \right] \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} (f, w) \rho(\mathbf{y}) d\mathbf{y}, \qquad \forall w \in W_J. \tag{2.22}$$

The semi-discrete formulations for the slightly compressible single and two-phase models are given by restriction to a finite dimensional MFE subspace in the spatial dimensions, along with the use of an appropriate time stepping scheme in the temporal dimension.

2.2.1 Tensor Product Grid Collocation

Next we employ the stochastic collocation method, using a tensor product Gauss-Hermite quadrature rule in stochastic dimensions, to form the fully discrete solution. The reason for this choice is that Hermite polynomals are the orthogonal family of polynomials that is associated with Normal random variables in the Wiener-Askey scheme of hypergeometric polynomials [32]. If other types of random variables are used instead, then other orthogonal polynomials can be substituted. As described in Section 1.3 this non-intrusive approach decouples the $(d + N_{\text{term}})$-dimensional stochastic problem into a sequence of independent d-dimensional deterministic problems, which are realizations in stochastic space and function evaluations in the quadrature rule.

Let m (or $\mathbf{m}$) be a scalar (or multi-index) indicating the desired polynomial degree of accuracy in the stochastic dimensions. The stochastic collocation method approximates the semidiscrete solution by an interpolant $\mathcal{I}_m$ in the stochastic dimensions. It is uniquely formed on a set of N_{real} stochastic points $\{\mathbf{y}_k\}$ that form a Haar set in $\mathbb{S}$, where N_{real} is a function of m. More precisely the fully discrete solution is

$$\mathbf{u}_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m \mathbf{u}_h(\mathbf{x},\mathbf{y}), \quad p_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m p_h(\mathbf{x},\mathbf{y}).$$

Let $\{L_m^{\{k\}}(\mathbf{y})\}$ be the Lagrange basis satifying $\{L_m^{\{k\}}(\mathbf{y}_j)\} = \delta_{kj}$. Then the fully discrete solution has the Lagrange representation

$$(\mathbf{u}_{h,m}, p_{h,m})(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{N_{\text{real}}} (\mathbf{u}_h^{\{k\}}, p_h^{\{k\}})(\mathbf{x}) L_m^{\{k\}}(\mathbf{y}),$$

where $(\mathbf{u}_h^{\{k\}}, p_h^{\{k\}})$ is the evaluation of semidiscrete solution $(\mathbf{u}_h, p_h)$ with the permeability realization

$$K^{\{k\}}(\mathbf{x}) = K(\mathbf{x}, \mathbf{y}_k).$$

In practice, the Lagrange representation of the interpolant is plugged into the expectation integral (1.16) to form a numerical quadrature rule. For example, the pressure expectation computed by

$$\begin{aligned} E[p_{h,m}](\mathbf{x}) &= \int_{\mathbb{S}} p_{h,m}(\mathbf{x},\mathbf{y})\rho(\mathbf{y})d\mathbf{y} \approx \int_{\mathbb{S}} \mathcal{I}_m p_h(\mathbf{x},\mathbf{y})\rho(\mathbf{y})d\mathbf{y} \\ &= \int_{\mathbb{S}} \sum_{k=1}^{N_{\text{real}}} p_h^{\{k\}}(\mathbf{x}) L_m^{\{k\}}(\mathbf{y})\rho(\mathbf{y})d\mathbf{y} = \sum_{k=1}^{N_{\text{real}}} w_m^{\{k\}} p_h^{\{k\}}(\mathbf{x}), \end{aligned} \tag{2.23}$$

where the weights are given by

$$w_m^{\{k\}} = \int_{\mathbb{S}} L_m^{\{k\}}(\mathbf{y})\rho(\mathbf{y})d\mathbf{y}. \tag{2.24}$$

The choice of collocation points $\{\mathbf{y}_k\}$, *i.e.* the type of quadrature rule, produces different types of stochastic collocation methods. Tensor product grids are described in this chapter, and sparse grids will be described in Chapter 4. Both types of grids are constructed from one-dimensional rules, where the points in each stochastic dimension are the zeros of orthogonal

polynomials with respect to the $L^2_\rho(\mathbb{S}_j)$-inner-product. Since we are using Gaussian random variables, we choose the zeros of the "probabilist" $N(0,1)$ Hermite polynomials

$$H_m(y) = m! \sum_{k=0}^{[m/2]} (-1)^k \frac{(2y)^{m-2k}}{k!(m-2k)!}. \tag{2.25}$$

Denote the sets of one-dimensional weights and abscissae for $H_m(y)$ by

$$\mathcal{W}(m) = \{w_m^1, \ldots, w_m^m\} \quad \text{and} \quad \mathcal{H}(m) = \{h_m^1, \ldots, h_m^m\}, \tag{2.26}$$

and notice that when $m = 2k-1$ is odd, the point h_m^k is the origin. These weights and abscissae can easily be computed with a symbolic manipulation software package. Alternatively, one may convert a table of rules for the "physicist" $N(0,1/2)$ Hermite polynomials listed in [5] by dividing the weights by factor of $\sqrt{\pi}$ and multiplying the abscissae by a factor of $\sqrt{2}$.

In tensor product collocation, the polynomial accuracy is prescribed in terms of *component* degree, *i.e.* independently in each stochastic dimension. This allows for very easy construction of anisotropic rules, accurate to different polynomial degrees in different stochastic dimensions. Unfortunately, the number of points in tensor product rules grow exponentially with both the polynomial accuracy and the number of dimensions. This is commonly referred to as the "curse of dimensionality". Therefore, this inherently limits their usage to problems with a relatively low number of stochastic dimensions, *i.e.* about a dozen or less.

If we choose $N_{\text{coll}}(j)$ collocation points in stochastic dimension j, then $\mathbf{m} = (N_{\text{coll}}(j))_j$ is the N_{term}-dimensional multi-index indicating the desired component degree of the interpolant in the stochastic space $\mathbb{S}$. The corresponding anisotropic tensor product Gauss-Hermite interpolant in N_{term}-dimensions is defined by

$$\begin{aligned} \mathcal{I}^{\text{TG}}_{\mathbf{m}} f(\mathbf{y}) &= (\mathcal{I}_{\mathbf{m}(1)} \otimes \cdots \otimes \mathcal{I}_{\mathbf{m}(N_{\text{term}})}) f(\mathbf{y}) \\ &= \sum_{k_1=1}^{\mathbf{m}(1)} \cdots \sum_{k_{N_{\text{term}}}=1}^{\mathbf{m}(N_{\text{term}})} f(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})}) L^{k_1}_{\mathbf{m}(1)}(y_1) \cdots L^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})}(y_{N_{\text{term}}}). \end{aligned} \tag{2.27}$$

The set of abscissae for this rule is

$$\mathcal{T}(\mathbf{m}) = \bigotimes_{k=1}^{N_{\text{term}}} \mathcal{H}(\mathbf{m}(k)), \tag{2.28}$$

which interpolates the semi-discrete solution into the polynomial space $\mathbb{P}_{\mathbf{m}} = \prod_k \mathbb{P}_{\mathbf{m}(k)}$ in the stochastic dimensions. By formula (2.24), the tensor product weight for the point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$ is given by

$$w(\mathbf{k}) = \prod_{i=1}^{N_{\text{term}}} w^{k_i}_{\mathbf{m}(i)}.$$

In a fixed stochastic dimension, the one dimensional Gauss-Hermite quadrature rules are accurate to degree $2m-1$.

2.3 ERROR ANALYSIS

The error between the true stochastic velocity $\mathbf{u}$ and the approximate fully discrete velocity $\mathbf{u}_{h,m}$ may be decomposed by adding and subtracting the semidiscrete velocity $\mathbf{u}_h$

$$\|\mathbf{u} - \mathbf{u}_{h,m}\|_V \leq \|\mathbf{u} - \mathbf{u}_h\|_V + \|\mathbf{u}_h - \mathbf{u}_{h,m}\|_V = \|\mathbf{u} - \mathbf{u}_h\|_V + \|\mathbf{u}_h - \mathcal{I}_m \mathbf{u}_h\|_V.$$

A similar decomposition holds for $\|p - p_{h,m}\|_W$. An *a priori* bound on the first term follows, assuming enough smoothness of the solution, from standard deterministic mixed FEM error analysis [18]

$$\begin{aligned}
&\|\mathbf{u} - \mathbf{u}_h\|^2_V + \|p - p_h\|^2_W \\
&\quad = \int_{\mathbb{S}} \left(\|\mathbf{u} - \mathbf{u}_h\|^2_{H(\text{div};D)} + \|p - p_h\|^2_{L^2(D)} \right) \rho(\mathbf{y}) d\mathbf{y} \\
&\quad \leq C \int_{\mathbb{S}} \left(h^{2r+2} \|\mathbf{u}\|^2_{H^{r+1}(D)} + h^{2s+2} \|\nabla \cdot \mathbf{u}\|^2_{H^{s+1}(D)} + h^{2s+2} \|p\|^2_{H^{s+1}(D)} \right) \rho(\mathbf{y}) d\mathbf{y} \\
&\quad = C \left(h^{2r+2} \|\mathbf{u}\|^2_{H^{r+1}(D) \otimes L^2(\mathbb{S})} + h^{2s+2} \|\nabla \cdot \mathbf{u}\|^2_{H^{s+1}(D) \otimes L^2(\mathbb{S})} \right. \\
&\qquad\qquad \left. + h^{2s+2} \|p\|^2_{H^{s+1}(D) \otimes L^2(\mathbb{S})} \right).
\end{aligned}$$

For the second term, an interpolation bound on $\mathbb{S}$ has recently been found in [11] to be

$$\|\mathbf{u}_h - I_m \mathbf{u}_h\|_V + \|p_h - I_m p_h\|_W \leq C \sum_{i=1}^{N_{\text{term}}} e^{-c_i \sqrt{m_i}},$$

where $c_i > 0$ are defined in [11, Theorem 4.1]. In particular, if K is smooth enough in $\mathbb{S}$, then the solution admits an analytic extension in a region of the complex plane containing $\mathbb{S}_i$ for $i = 1, \ldots, N_{\text{term}}$, and that c_i depends on the distance between $\mathbb{S}_i$ and the nearest singularity in the complex plane. The KL expansion (1.19) satisfies the smoothness assumption in [11]. As a result we have the following theorem.

Theorem 2.3.1. *Assume that* $\mathbf{u} \in H^{r+1}(D) \otimes L^2_\rho(\mathbb{S})$, $\nabla \cdot \mathbf{u} \in H^{s+1}(D) \otimes L^2_\rho(\mathbb{S})$, *and* $p \in H^{s+1}(D) \otimes L^2_\rho(\mathbb{S})$. *Then there exists a constant* C *independent of* h *and* M *such that*

$$\|\mathbf{u} - \mathbf{u}_{h,m}\|_V + \|p - p_{h,m}\|_W \leq C \left(h^{r+1} + h^{s+1} + \sum_{i=1}^{N_{term}} e^{-c_i \sqrt{m_i}} \right).$$

We next establish a superconvergence bound for the pressure. For $\varphi \in L^2(D)$, denote with $\hat{\varphi}$ its L^2-projection in W_h satisfying

$$(\varphi - \hat{\varphi}, w_h)_{L^2(D)} = 0 \quad \forall\, w_h \in W_h, \tag{2.29}$$

$$\|\varphi - \hat{\varphi}\|_{L^2(D)} \leq C h^l \|\varphi\|_{H^l(D)}, \quad 0 \leq l \leq s+1. \tag{2.30}$$

Let $\Pi : (H^1(D))^d \to \mathbf{V}_h(D)$ be the mixed finite element projection operator satisfying

$$(\nabla \cdot (\mathbf{u} - \Pi\mathbf{u}), w_h)_{L^2(D)} = 0 \quad \forall\, w_h \in W_h, \tag{2.31}$$

$$\|\mathbf{u} - \Pi\mathbf{u}\|_{(L^2(D))^d} \leq C h^l \|\mathbf{u}\|_{(H^l(D))^d}, \quad 1 \leq l \leq r+1. \tag{2.32}$$

Theorem 2.3.2. *Assume that problem (2.7)–(2.9) is* H^2*-elliptic regular. Under the assumptions of Theorem 2.3.1, there exists a constant* C *independent of* h *and* M *such that*

$$\|\hat{p} - p_{h,m}\|_W \leq C(h\|\mathbf{u} - \mathbf{u}_h\|_V + \|p_h - I_m p_h\|_W).$$

Proof. The proof is based on a duality argument. Taking $\mathbf{v} = \mathbf{v}_h$ and $w = w_h$ in the weak formulation (2.15)-(2.16) and subtracting the semidiscrete formulation (2.17)-(2.18) gives the error equations for *a.e.* $\mathbf{y} \in \mathbb{S}$

$$(K^{-1}(\mathbf{u} - \mathbf{u}_h), \mathbf{v}_h)_{L^2(D)} = (\hat{p} - p_h, \nabla \cdot \mathbf{v}_h)_{L^2(D)}, \quad \forall \mathbf{v}_h \in \mathbf{V}_h(D), \tag{2.33}$$

$$(\nabla \cdot (\mathbf{u} - \mathbf{u}_h), w_h)_{L^2(D)} = 0, \quad \forall w_h \in W_h(D). \tag{2.34}$$

Now consider the following auxiliary problem in mixed form:

$$\begin{aligned} \boldsymbol{\psi}(\cdot, \mathbf{y}) &= -K(\cdot, \mathbf{y}) \nabla \varphi(\cdot, \mathbf{y}) && \text{in } D, \\ \nabla \cdot \boldsymbol{\psi}(\cdot, \mathbf{y}) &= \hat{p} - p_{h,m} && \text{in } D, \\ \varphi(\cdot, \mathbf{y}) &= 0 && \text{on } \partial D. \end{aligned}$$

The elliptic regularity implies

$$\|\varphi(\cdot, \mathbf{y})\|_{H^2(D)} \leq C \|\hat{p} - p_{h,m}\|_{L^2(D)}. \tag{2.35}$$

Therefore,

$$\begin{aligned} \|\hat{p} - p_{h,m}\|_W^2 &= \int_{\mathbb{S}} (\hat{p} - p_{h,m}, \hat{p} - p_{h,m})_{L^2(D)} \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} (\nabla \cdot \boldsymbol{\psi}, \hat{p} - p_{h,m})_{L^2(D)} \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \left((\nabla \cdot \boldsymbol{\psi}, \hat{p} - p_h)_{L^2(D)} + (\nabla \cdot \boldsymbol{\psi}, p_h - I_m p_h)_{L^2(D)} \right) \rho(\mathbf{y}) d\mathbf{y} \\ &= I + II. \end{aligned}$$

Applying the Cauchy-Schwarz inequality, we have

$$\begin{aligned} |II| &\leq \left(\int_{\mathbb{S}} \|\nabla \cdot \boldsymbol{\psi}\|_{L^2(D)}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|p_h - I_m p_h\|_{L^2(D)}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &= \left(\int_{\mathbb{S}} \|\hat{p} - p_{h,m}\|_{L^2(D)}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \|p_h - I_m p_h\|_W \\ &= \|\hat{p} - p_{h,m}\|_W \|p_h - I_m p_h\|_W. \end{aligned}$$

Using (2.31) and (2.33) with $\mathbf{v}_h = \Pi\boldsymbol{\psi}$,

$$\begin{aligned} I &= \int_{\mathbb{S}} (K^{-1}(\mathbf{u}-\mathbf{u}_h), \Pi\boldsymbol{\psi})_{L^2(D)}\rho(\mathbf{y})d\mathbf{y} \\ &= \int_{\mathbb{S}} \left((K^{-1}(\mathbf{u}-\mathbf{u}_h), \Pi\boldsymbol{\psi}-\boldsymbol{\psi})_{L^2(D)} - (\mathbf{u}-\mathbf{u}_h, \nabla\varphi)_{L^2(D)}\right)\rho(\mathbf{y})d\mathbf{y} \\ &= I_1 + I_2. \end{aligned}$$

The Cauchy-Schwarz inequality, (2.32), and (4.39) imply

$$\begin{aligned} |I_1| &\leq C\left(\int_{\mathbb{S}} \|\mathbf{u}-\mathbf{u}_h\|^2_{L^2(D)}\rho(\mathbf{y})d\mathbf{y}\right)^{1/2}\left(\int_{\mathbb{S}} \|\Pi\boldsymbol{\psi}-\boldsymbol{\psi}\|^2_{L^2(D)}\rho(\mathbf{y})d\mathbf{y}\right)^{1/2} \\ &\leq C\|\mathbf{u}-\mathbf{u}_h\|_V\ h\left(\int_{\mathbb{S}} \|\boldsymbol{\psi}\|^2_{(H^1(D))^d}\rho(\mathbf{y})d\mathbf{y}\right)^{1/2} \\ &\leq Ch\|\mathbf{u}-\mathbf{u}_h\|_V\|\hat{p}-p_{h,m}\|_W. \end{aligned}$$

Using (2.34), (2.30), and (4.39), we have

$$\begin{aligned} |I_2| &= \left|\int_{\mathbb{S}} (\nabla\cdot(\mathbf{u}-\mathbf{u}_h), \varphi - w_h)_{L^2(D)}\rho(\mathbf{y})d\mathbf{y}\right| \\ &\leq C\|\mathbf{u}-\mathbf{u}_h\|_V h\|\hat{p}-p_{h,m}\|_W. \end{aligned}$$

A combination of the above estimates completes the proof of the theorem. □

Corollary 2.3.1. *Under the assumptions of Theorem 2.3.2, there exists a constant C independent of h and M such that*

$$\|\hat{p}-p_{h,m}\|_W \leq C\left(h^{r+2} + h^{s+2} + \sum_{i=1}^{N_{term}} e^{-c_i\sqrt{m_i}}\right).$$

Next, we describe the extension of these theoretical results to slightly compressible flows. We do not give theoretical results for two-phase flow, as few *a priori* estimates are known. As before, we add and subtract the semidiscrete velocity, splitting the error into

$$\|\mathbf{u}-\mathbf{u}_{h,m}\|_{V_J} = \|\mathbf{u}-\mathbf{u}_h\|_{V_J} + \|\mathbf{u}-I_m\mathbf{u}_h\|_{V_J},$$

which represents a deterministic discretization error and a stochastic error. Similar decomposition holds for $\|p-p_{h,m}\|_{W_J}$. Using the deterministic error bounds [53, 54, 67]

$$\begin{aligned} \|\mathbf{u}-\mathbf{u}_h\|_{H(\mathrm{div};D)\otimes L^p(J)} + \|p-p_h\|_{L^2(D)\otimes L^p(J)} &\leq C(h^{r+1}\|\mathbf{u}\|_{H^{r+1}(D)\otimes L^p(J)} \\ + h^{s+1}\|\nabla\cdot\mathbf{u}\|_{H^{s+1}(D)\otimes L^p(J)} + h^{s+1}\|p\|_{H^{s+1}(D)\otimes L^p(J)}), \end{aligned}$$

and the argument for the proof of Theorem 2.3.1, we obtain the following result.

Theorem 2.3.3. *Assume that* $\mathbf{u} \in H^{r+1}(D) \otimes L^p(J) \otimes L^2_\rho(\mathbb{S})$, $\nabla \cdot \mathbf{u} \in H^{s+1}(D) \otimes L^p(J) \otimes L^2_\rho(\mathbb{S})$, *and* $p \in H^{s+1}(D) \otimes L^p(J) \otimes L^2_\rho(\mathbb{S})$. *Then there exists a constant* C *independent of* h *and* M *such that*

$$\|\mathbf{u} - \mathbf{u}_{h,m}\|_{V_J} + \|p - p_{h,m}\|_{W_J} \leq C \left(h^{r+1} + h^{s+1} + \sum_{i=1}^{N_{term}} e^{-c_i \sqrt{m_i}} \right).$$

2.4 NUMERICAL EXAMPLES FOR INCOMPRESSIBLE FLOW

To perform the numerical experiments for Single-Phase Incompressible Flow, the framework to support uncertainty quantification methods and stochastic permeability representation with KL expansion was coded and integrated into the PARCEL simulator. This is a parallel mixed finite element research code [27], which is written in FORTRAN and parallelized using the MPI Library. It divides the problem into four non-overlapping subdomains and reformulates the problem in terms of interfaces variables, using the original implementation of the MMMFEM that will be described in Chapter 3. In a matching grid configuration, this is equivalent to a standard MFE discretization, and with lowest order Raviart-Thomas RT_0 space on a uniform mesh of rectangular 2-D elements, this is also equivalent to a finite difference scheme.

The covariance function (2.1) was used to generate the KL expansion of an isotropic permeability field by the procedure described in Section 2.1. Implementation of the stochastic collocation method was achieved by adding a loop around the deterministic solver and supplying it with permeability realizations at each stochastic collocation point. The solutions for both stochastic pressure and velocity are then averaged together using the collocation weights in order to compute their expectation and variance.

The numerical experiments are solved on the square domain $D = (0, 1)^2$. Each test assumes the same KL expansion for mean removed log permeability Y' with variance $\sigma_Y = 1$, and correlation lengths $\eta_1 = 0.20, \eta_2 = 0.125$. The series is truncated after $N_{\text{term}} = 6$ terms.

Reported pressure errors are computed with a discrete L^2 norm at the cell centers, velocity errors at the midpoints of the edges, and flux errors at the midpoints of the edges.

The stochastic convergence is computed on a fixed 80×80 spatial mesh. The expected solutions on stochastic tensor product grids made up of $N_{\text{coll}} = 2, 3, 4$ collocation points in $N_{\text{term}} = 6$ stochastic dimensions are compared to the mean solution using $N_{\text{coll}} = 5$ collocation points. The deterministic convergence is computed using a fixed stochastic tensor product grid of with $N_{\text{coll}} = 4$ collocation points in $N_{\text{term}} = 6$ stochastic dimensions. The spatial mesh is refined from a 10×10 grid to an 80×80 grid, and error is computed against the numerical solution on a 160×160 grid.

We consider three examples: 2.4.1 tests flow induced from left-to-right, 2.4.2 tests a quarter five-spot well distribution, and 2.4.3 tests a discontinuous permeability field.

2.4.1 Left-To-Right Example

Let Dirichlet boundary conditions $p = 1$ on $\{x_1 = 0\}$, $p = 0$ on $\{x_1 = 1\}$ and Neumann boundary conditions $\mathbf{u} \cdot \mathbf{n} = 0$ specified on both $\{x_2 = 0\}, \{x_2 = 1\}$. The source function is $f = 0$. The log permeability Y has zero mean.

Figure 2.1 shows a typical Monte-Carlo realization of the isotropic permeability field, and its corresponding solution. Figures 2.2 and 2.3 show the expectation and variance of the stochastic solution. The pressure variance is largest in a vertical strip in the middle of the domain, away from the Dirichlet boundary edges. The velocity variance is smallest along the Neumann edges and it is affected by the direction of the flow. Table 2.1 shows the stochastic convergence. We note that exponential convergence is observed for the stochastic error. Table 2.2 shows the deterministic convergence. The numbers in parenthesis are the ratios between the errors on successive levels of refinement. Superconvergence of the deterministic error is observed for both the pressure and the velocity, confirming the theory.

2.4.2 Quarter Five-Spot Example

Assume no-flow boundary conditions $\mathbf{u} \cdot \mathbf{n} = 0$ on ∂D. The spatial mesh is made up of 80×80 elements. The source function has a source $f = 100$ in the upper left element and a sink $f = -100$ in the lower right element, and is everywhere else $f = 0$. The log permeability Y has zero mean.

Figure 2.1: A Monte-Carlo realization of the permeability field (left) and its corresponding solution (right) for Example 2.4.1 with 6 KL terms.

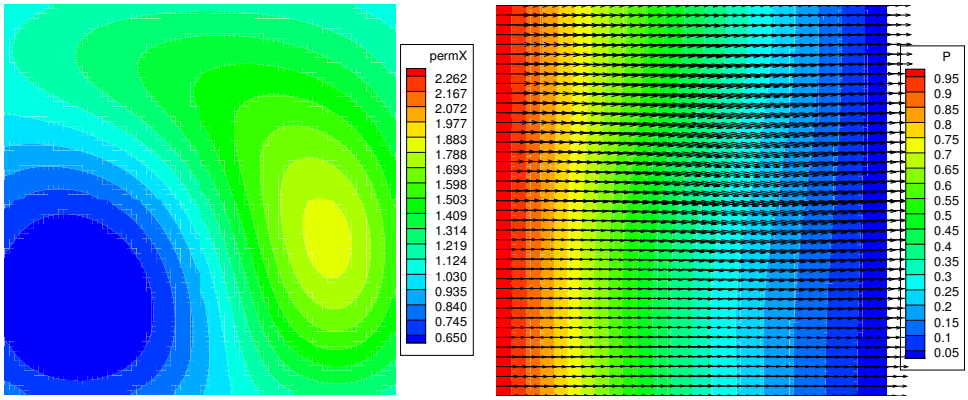


Figure 2.2: Expectation of solution (left), and variance of the pressure (right) for Example 2.4.1 with 4^6 collocation points.

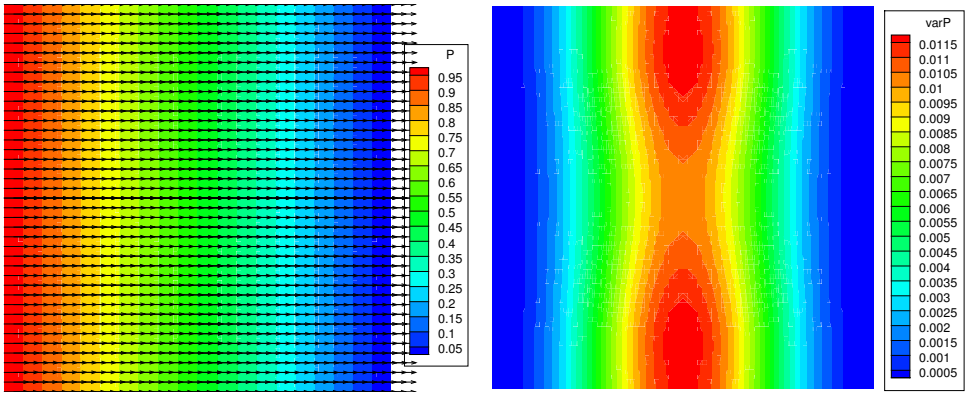


Figure 2.3: Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.1 with 4^6 collocation points.

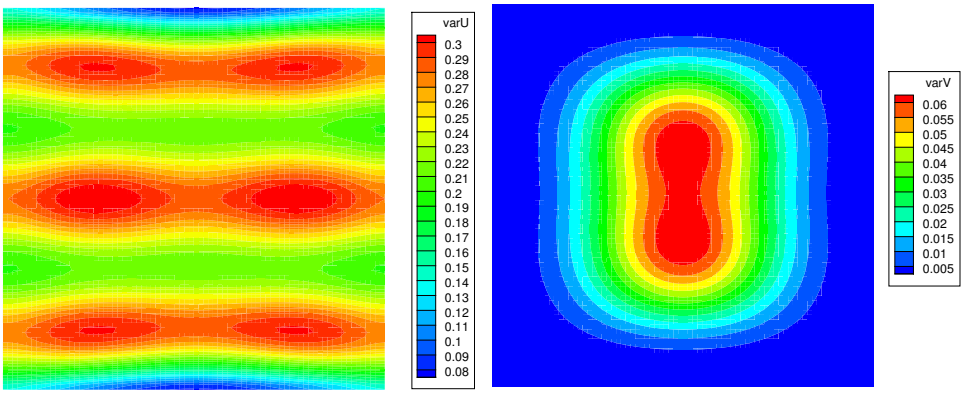


Table 2.1: Stochastic convergence results for Example 2.4.1. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses.

Coll. Points	Flux L2 Error	Pressure L2 Error	Velocity L2 Error
$2^6 = 64$	2.3472E-03	1.8826E-05	1.6338E-03
$3^6 = 729$	5.6240E-05 (41.74)	1.2013E-06 (15.67)	3.8667E-05 (42.25)
$4^6 = 4096$	3.8503E-06 (14.61)	1.0064E-07 (11.94)	2.6241E-06 (14.74)

Table 2.2: Deterministic convergence results for Example 2.4.1. Absolute errors reported against 160×160 grid; convergence ratios given in parentheses.

Grid	Flux L2 Error	Pressure L2 Error	Velocity L2 Error
10×10	9.2214E-04	1.1149E-04	9.2673E-04
20×20	2.3358E-04 (3.95)	2.7343E-05 (4.08)	2.4653E-04 (3.76)
40×40	5.5987E-05 (4.17)	6.5060E-06 (4.20)	5.9988E-05 (4.11)
80×80	1.1776E-05 (4.75)	1.3030E-06 (4.99)	1.2159E-05 (4.93)

Figure 2.4: Expectation of solution (left), and variance of the pressure (right) for Example 2.4.2 with 5^6 collocation points.

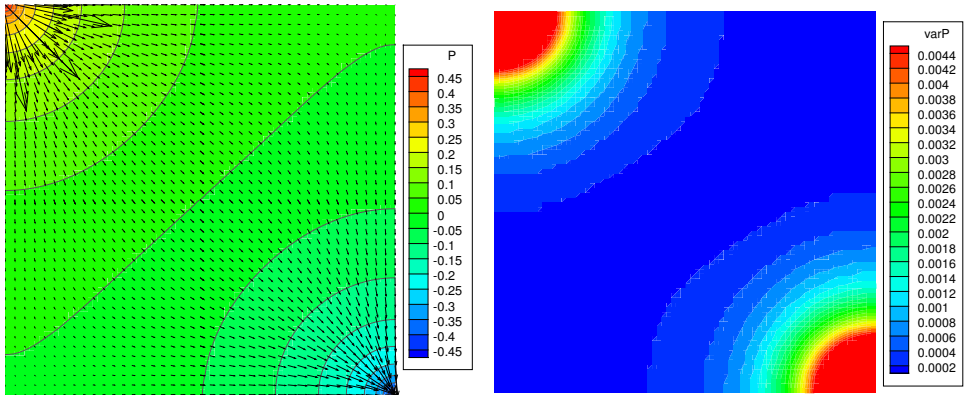


Figure 2.5: Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.2 with 5^6 collocation points.

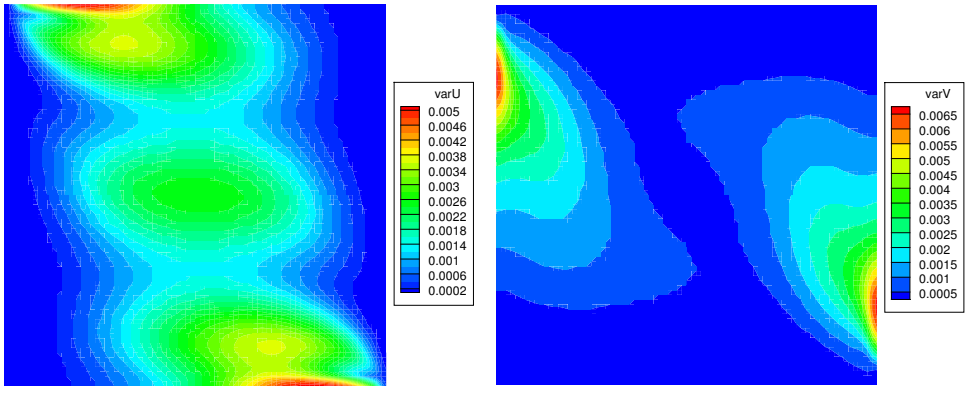


Table 2.3: Stochastic convergence results for Example 2.4.2. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses.

Coll. Points	Flux L2 Error	Pressure L2 Error	Velocity L2 Error
$2^6 = 64$	6.3070E-05	6.4144E-05	3.1526E-05
$3^6 = 729$	2.3975E-06 (26.31)	4.7729E-07 (134.3)	1.1400E-06 (27.65)
$4^6 = 4096$	1.3597E-07 (17.63)	3.4527E-08 (13.82)	6.4159E-08 (17.77)

Figures 2.4 and 2.5 show the expectation and variance of the stochastic solution for Example 2.4.2. The pressure variance is is largest at the wells and so is the velocity variance. However, the velocity variance is also affected by the no flow boundary conditions. Table 2.3 shows the stochastic convergence. We again observe exponential convergence.

2.4.3 Discontinuous Permeability Example

Take the same boundary conditions and source function as Example 2.4.1. The log permeability Y has a mean of 4.6 in lower-left and upper-right subdomains, and zero mean in upper-left and lower-right subdomains.

Figures 2.6 and 2.7 show the expectation and variance of the stochastic solution to example 2.4.3. The pressure variance is largest in the regions where the pressure changes the most. The velocity variance is largest at the cross-point, where the solution is singular and the true velocity is infinite. Table 2.4 shows the stochastic convergence. Despite the singularity in physical space, the solution preserves it smoothness in stochastic space, and exponential convergence is observed. Table 2.5 shows the deterministic convergence. Due to the singularity at the cross-point, the convergence rates have deteriorated, but appear to be approaching first order for both the pressure and velocity.

Figure 2.6: Expectation of solution (left), and variance of the pressure (right) for Example 2.4.3 with 5^6 collocation points.

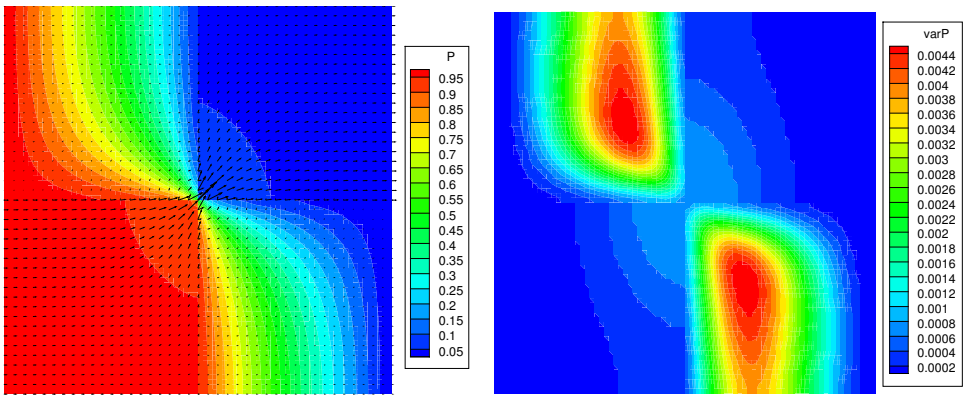


Figure 2.7: Variance of the x-velocity component (left), and variance of the y-velocity component (right) for Example 2.4.3 with 5^6 collocation points.

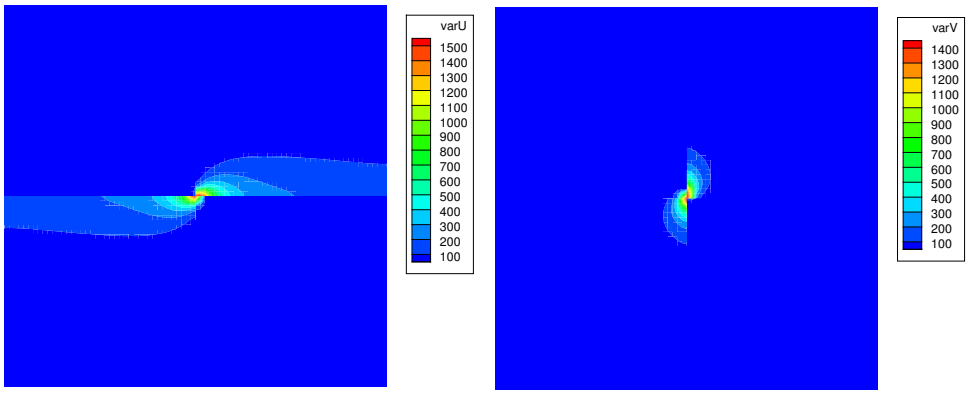


Table 2.4: Stochastic convergence results for Example 2.4.3. Absolute errors reported against 5^6 collocation points; convergence ratios given in parentheses.

Coll. Points	Flux L2 Error	Pressure L2 Error	Velocity L2 Error
$2^6 = 64$	4.1409E-02	1.6391E-05	1.6094E-02
$3^6 = 729$	5.0458E-04 (82.06)	9.5194E-07 (17.22)	2.5807E-04 (62.36)
$4^6 = 4096$	1.9238E-05 (26.23)	1.2567E-08 (75.75)	6.8610E-06 (37.61)

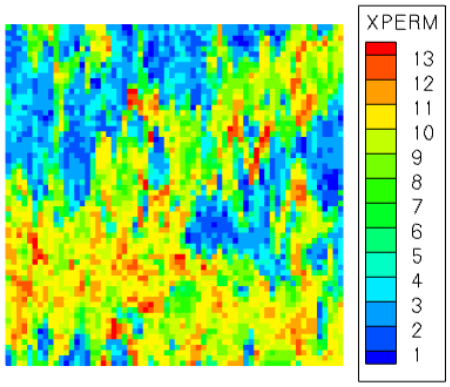
Table 2.5: Deterministic Convergence results for Example 2.4.3. Absolute errors reported against 160×160 grid; convergence ratios given in parentheses.

Grid	Flux L2 Error	Pressure L2 Error	Velocity L2 Error
10×10	12.2307	1.1630E-02	5.5089
20×20	12.7644 (0.95)	8.7942E-03 (1.32)	4.3481 (1.26)
40×40	11.9079 (1.07)	5.7352E-03 (1.53)	2.9959 (1.45)
80×80	8.2836 (1.43)	2.7497E-03 (2.08)	1.5245 (1.96)

2.5 NUMERICAL EXAMPLES FOR SLIGHTLY COMPRESSIBLE FLOW

The numerical experiments for slightly compressible single and two-phase flow were programmed using a reservoir simulator known as IPARS [70, 82, 66]. This code also uses lowest order Raviart-Thomas elements on rectangles in 2-D. We consider a two dimensional reservoir 1280×1280 $[ft^2]$ with a mean permeability field with upscaled data from the Tenth SPE Comparative Solution Project [22], as shown in Figure 2.8.

Figure 2.8: Upscaled SPE10 mean log-permeability field.



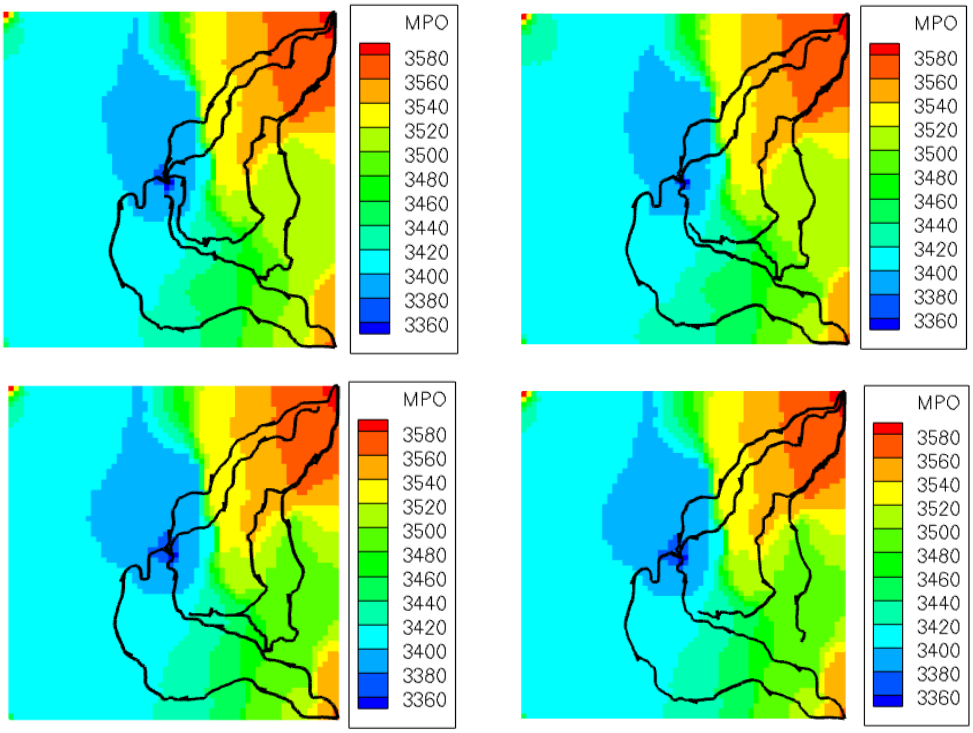
We consider two examples for slightly compressible flow: 2.5.1 tests single-phase flow, and 2.5.2 tests two-phase flow. They have the following parameters: The initial pressure is set at 3550 [psi]. Injection wells are placed in each corner with a pressure of 3600 [psi], and a production well is placed in the center with a pressure of 3000 [psi]. The numerical grid is 64×64 and the simulations run for T=50 days with variable time stepping. We assume $\sigma_Y = 1$, correlation lengths $\eta_1 = \eta_2 = 0.78$, and truncate the KL expansion after four terms.

2.5.1 Single-Phase Example

In Figure 2.9 we plot the mean pressure fields at $t = 50$ for the Monte Carlo and stochastic collocation simulations respectively. We include streamlines to indicate the direction of the flow. In Figure 2.10 we plot the standard deviation of the pressure fields at $t = 50$ for the Monte Carlo and stochastic collocation simulations, respectively. In each case, we see that the

stochastic collocation provides results comparable to the Monte Carlo while requiring fewer simulations. We note that the scale of the standard deviation of the pressure is significantly less than that of the mean.

Figure 2.9: Mean pressure field and streamlines for Example 2.5.1 using the mean permeability (top-left), 100 Monte Carlo simulations (top-right), 2^4 collocation points (bottom-left), and 3^4 collocation points (bottom-right).

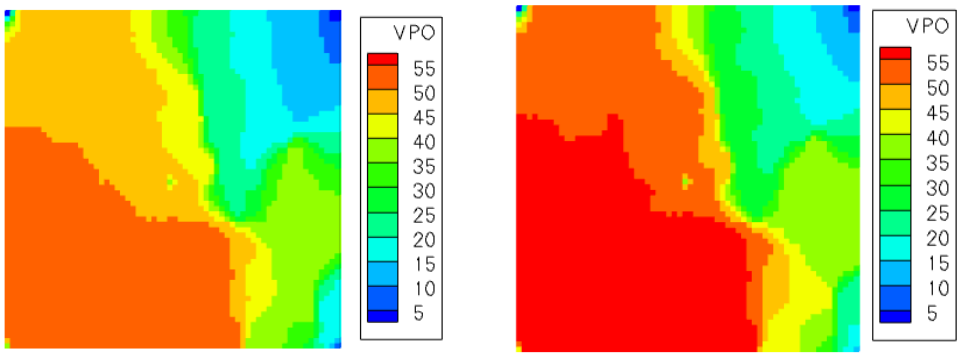


2.5.2 Two-Phase Example

Next, we compare some numerical results for two phase (oil and water) slightly compressible flow to the numerical results for single phase slightly compressible flow computed above. The mean permeability, the porosity, and the well models are same as the previous example. We also use the same KL expansion and collocation points. The initial oil pressure is set at 3550 [psi] and the initial water saturation is 0.2763.

In Figures 2.11, 2.12 and 2.13, we plot the mean and the standard deviation of the oil pressure, the water saturation, and the cumulative oil production respectively using 100

Figure 2.10: Standard deviation of pressure field for Example 2.5.1 using 100 Monte Carlo simulations (left), and 3^4 collocation points (right).



Monte Carlo simulations and 2^4 collocation points. The statistics computed using collocation are comparable to the Monte Carlo simulations while requiring less computational effort.

Comparing Figures 2.9 and 2.11, we see that the mean and the standard deviation of the pressure fields have similar structure. This indicates that we may be able to use the single phase solver, which is less expensive, to determine the appropriate number of terms in the KL expansion, to select the number of collocation points, or to design effective preconditioners.

To investigate the effect that changing σ_Y in (2.1) has on the output statistics, we repeat the above simulations for two phase flow with $\sigma_Y = 0.25$ and with $\sigma_Y = 3$. In each case we apply the collocation method with 3^4 points. In Figure 2.14 we plot the mean and the standard deviation of the oil pressure. Comparing with Figure 2.11, we see that varying the standard deviation of the permeability field affects the scale of the uncertainty but not the overall structure of the pressure field.

Finally, we compare the mean pressure field in Figure 2.11 (b) with the mean pressure computed using different correlation lengths in the covariance function (2.1). In Figure 2.15, we plot the mean pressure using $\eta_1 = 0.16$ and $\eta_2 = 0.23$ as well as the mean pressure using $\eta_1 = \eta_2 = 0.08$. We notice that the mean pressure differs only slightly in each case, despite using different correlation lengths. Theoretically, more terms in the KL expansion are required to accurately represent the output statistics for shorter correlation lengths.

Figure 2.11: Mean oil pressure and streamlines for Example 2.5.2 using 100 Monte Carlo simulations (top-left) and 2^4 collocation points (top-right). Standard deviation of oil pressure using 100 Monte Carlo simulations (bottom-left) and 2^4 collocation points (bottom-right).

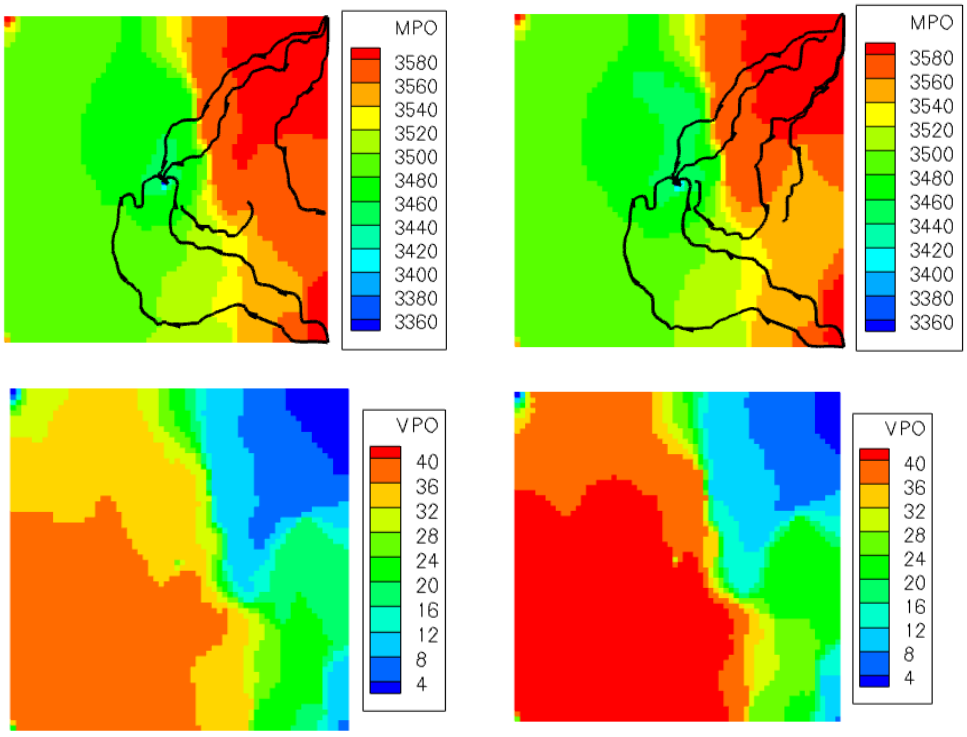


Figure 2.12: Mean water saturation for Example 2.5.2 using 100 Monte Carlo simulations (top-left) and 2^4 collocation points (top-right). Standard deviation of water saturation using 100 Monte Carlo simulations (bottom-left) and 2^4 collocation points (bottom-right).

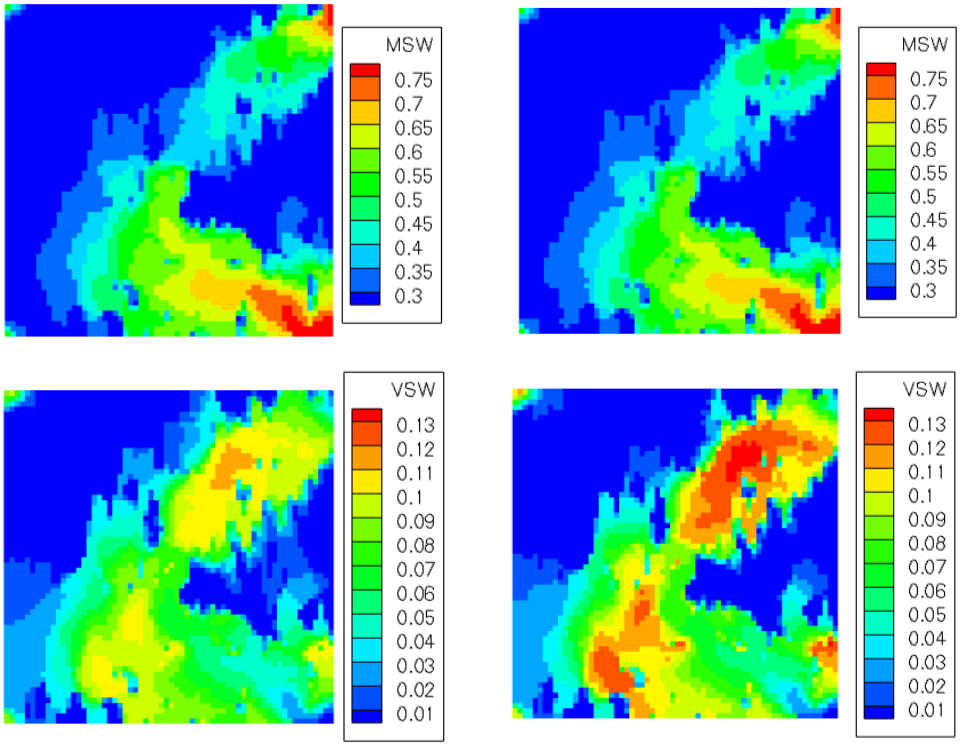


Figure 2.13: The mean (left) and standard deviation (right) for cumulative oil production of Example 2.5.2 using 100 Monte Carlo simulations and 2^4 collocation points.

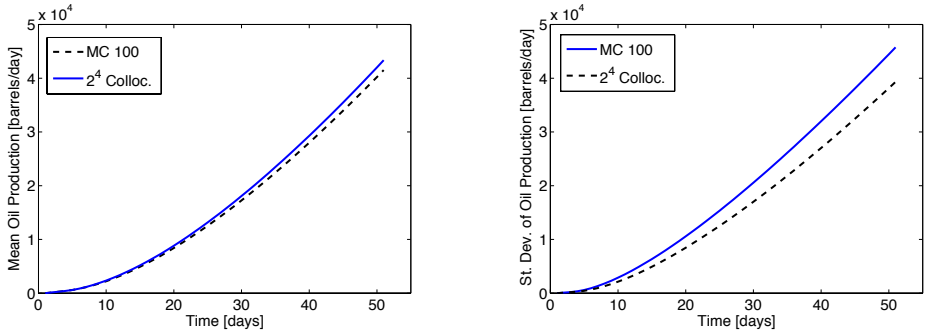


Figure 2.14: Mean pressure field and streamlines of Example 2.5.2 with 3^4 collocation points using $\sigma_Y = 0.25$ (top-left) and $\sigma_Y = 3$ (top-right). Standard deviation of pressure field using $\sigma_Y = 0.25$ (bottom-left) and $\sigma_Y = 3$ (bottom-right).

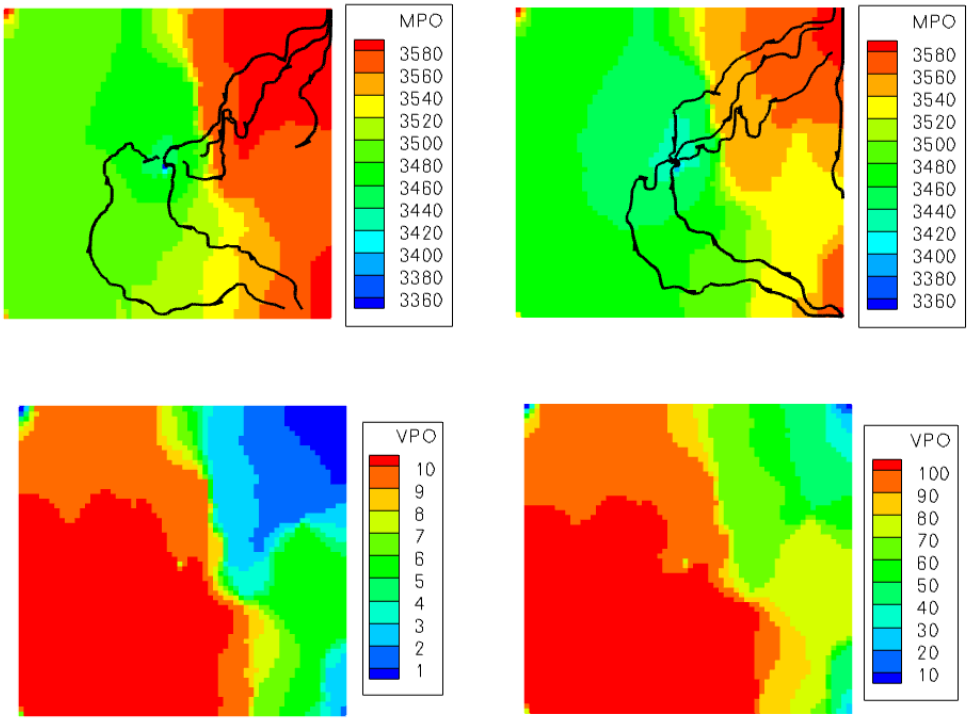
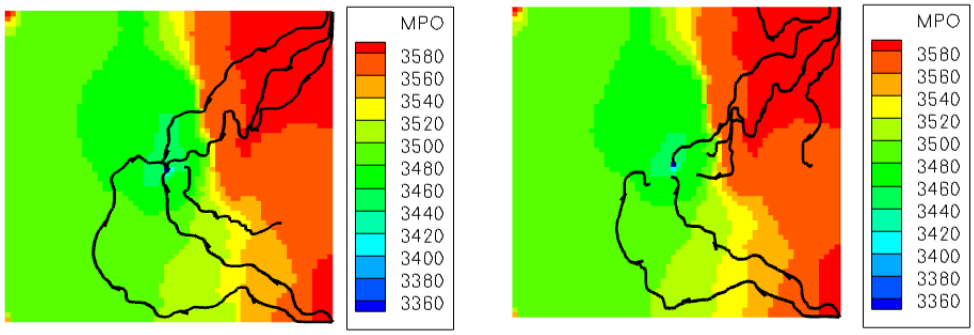


Figure 2.15: Mean pressure field and streamlines of Example 2.5.2 using 3^4 collocation points with correlation lengths $\eta_1 = 0.16$, $\eta_2 = 0.23$ (left), and $\eta_1 = \eta_2 = 0.08$ (right).



3.0 IMPLEMENTATION OF A MORTAR MIXED FINITE ELEMENT METHOD USING A MULTISCALE FLUX BASIS

This chapter contains material which was published in [38], in which a new implementation is described for the Multicale Mortar Mixed Finite Element Method in a purely deterministic setting. The original implementation of the MMMFEM in [9] requires solving one Dirichlet fine scale subdomain problem per interface iteration. As a result the number of subdomain solves increases with the dimension of the coarse space, making it difficult to compare the computational efficiency of the method to other existing multiscale methods. We alter this implementation by forming what we call a Multiscale Flux Basis, before the interface iteration begins.

This basis consists of mortar functions representing the individual flux responses from each mortar degree of freedom, on each subdomain independently. The basis functions may also be described as traces of the discrete Green's functions corresponding to the mortar degrees of freedom along the subdomain interfaces. The computation of these basis functions requires solving a fixed number of Dirichlet subdomain problems. Taking linear combinations of the Multiscale Flux Basis functions replaces the need to solve any Dirichlet subdomain problems during the interface iteration. This new implementation yields the same solution as the original implementation and makes the MMMFEM comparable to the variational multiscale method and multiscale finite elements in terms of computational efficiency.

In our numerical experiments we compare the computational cost of the new implementation to the one for the original implementation with and without preconditioning of the interface problem. If no preconditioning is used, the Multiscale Flux Basis implementation is computationally more efficient in cases where the number of mortar degrees of freedom per subdomain is less than the number of interface iterations. If balancing preconditioning is

used [26, 69], the number of iterations is reduced, but each interface iteration requires three subdomain solves. In this case the Multiscale Flux Basis implementation is more efficient if the number of mortar degrees of freedom per subdomain is less than three times the number of interface iterations.

The format of this chapter is as follows: Section 3.1 introduces the MMMFEM and its original implementation, Section 3.2 describes the new implementation using a Multiscale Flux Basis, and Section 3.3 provides several numerical examples that illustrate its computational efficiency.

3.1 FORMULATION OF THE MORTAR MIXED FINITE ELEMENT METHOD

Consider the deterministic incompressible single-phase model (1.1)-(1.4). Recall the weak pressure and velocity spaces for the global dual mixed problem from definition (2.11), and denote them in this chapter by $\tilde{W} = W(D)$ and $\tilde{\mathbf{V}}^{\gamma} = \mathbf{V}^{\gamma}(D)$. The corresponding variational formulation is the deterministic analog of system (2.15)-(2.16). That is to find $\mathbf{u} \in \tilde{\mathbf{V}}^{g_N}$ and $p \in \tilde{W}$ such that

$$
\begin{aligned}
(K^{-1}\mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) &= -\langle \mathbf{v} \cdot \mathbf{n}, g_D \rangle_{\Gamma_D} \quad \forall \mathbf{v} \in \tilde{\mathbf{V}}^0, && (3.1)\\
(\nabla \cdot \mathbf{u}, w) &= (f, w) \quad \forall w \in \tilde{W}. && (3.2)
\end{aligned}
$$

The first step in formulating the MMMFEM is to use the substructuring approach described in [42] to restrict the model problem into subdomains. Let the domain D be divided into N_D non-overlapping subdomains D_i, $i = 1, \ldots, N_D$. They are allowed to be spatially non-conforming, so we have $\overline{D} = \bigcup_{i=1}^{N_D} \overline{D_i}$ and $D_i \cap D_j = \emptyset$ for $i \neq j$. Denote the single interface between subdomains D_i and D_j by $\Gamma_{i,j} = \partial D_i \cap \partial D_j$, all interfaces that touch subdomain D_i by $\Gamma_i = \partial D_i \setminus \partial D$, and the union of all interfaces by $\Gamma = \bigcup_{i \neq j} \Gamma_{i,j}$. The domain decomposition can be viewed as a coarse grid on D.

The original system (1.1)-(1.4) holds within each subdomain D_i. The pressure and the normal components of the velocity must be continuous across the interfaces. Equivalently, we seek $(\mathbf{u}_i, p_i)$ such that for $i = 1, \ldots, N_D$,

$$\begin{aligned}
\nabla \cdot \mathbf{u}_i &= f, \quad \text{in } D_i, && (3.3)\\
\mathbf{u}_i &= -K\nabla p_i, \quad \text{in } D_i, && (3.4)\\
p_i &= g_D, \quad \text{on } \partial D_i \cap \Gamma_D, && (3.5)\\
\mathbf{u}_i \cdot \mathbf{n} &= g_N, \quad \text{on } \partial D_i \cap \Gamma_N, && (3.6)\\
p_i &= p_j, \quad \text{on } \Gamma_{i,j} \quad i \neq j, && (3.7)\\
\mathbf{u}_i \cdot \mathbf{n}_i + \mathbf{u}_j \cdot \mathbf{n}_j &= 0, \quad \text{on } \Gamma_{i,j}, \quad i \neq j, && (3.8)
\end{aligned}$$

Define the weak spaces for each subdomain D_i by

$$W_i = L^2(D_i), \quad \mathbf{V}_i = H(\text{div}; D_i), \quad \mathbf{V}_i^\gamma = \{\mathbf{v} \in \mathbf{V}_i \mid \mathbf{v} \cdot \mathbf{n} = \gamma \text{ on } \partial D_i \cap \Gamma_N\}. \tag{3.9}$$

The weak spaces for the domain decomposition problem (3.3)-(3.8) are

$$W = \bigoplus_{i=1}^{N_D} W_i, \quad \mathbf{V}^\gamma = \bigoplus_{i=1}^{N_D} \mathbf{V}_i^\gamma. \tag{3.10}$$

Note that no continuity is imposed across the interfaces. On Γ we introduce a Lagrange multiplier space that has a physical meaning of pressure and is used to weakly impose continuity of the normal velocities:

$$M = \{\mu \in H^{1/2}(\Gamma) \mid \mu|_{\Gamma_i} \in (\mathbf{V}_i \cdot \mathbf{n}_i)^*, \, i = 1, \ldots, N_D\}. \tag{3.11}$$

The corresponding domain decomposition variational formulation is to find $\mathbf{u} \in \mathbf{V}^{g_N}$, $p \in W$, and $\lambda \in M$ such that for $i = 1, \ldots, N_D$,

$$\begin{aligned}
(K^{-1}\mathbf{u}, \mathbf{v})_{D_i} - (p, \nabla \cdot \mathbf{v})_{D_i} &= -\langle \mathbf{v} \cdot \mathbf{n}_i, \lambda \rangle_{\partial D_i \cap \Gamma} - \langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \quad \forall \mathbf{v} \in \mathbf{V}_i^0, && (3.12)\\
(\nabla \cdot \mathbf{u}, w)_{D_i} &= (f, w)_{D_i} \quad \forall w \in W_i, && (3.13)\\
\sum_{i=1}^{N_D} \langle \mathbf{u}_i \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} &= 0 \quad \forall \mu \in M. && (3.14)
\end{aligned}$$

Since $\mathbf{V}^\gamma \neq \tilde{\mathbf{V}}^\gamma$, the extra condition (3.14) is needed to weakly enforce the flux continuity lost across the interfaces in the domain decomposition. The following equivalence result has been shown in [71].

Lemma 3.1.1. *If the solution* $(\mathbf{u}, p)$ *to* (3.1)–(3.2) *satisfies* (1.1)–(1.4) *in a distributional sense, then* $(\mathbf{u}, p, p|_\Gamma)$ *solves* (3.12)–(3.14). *Conversely, if* $(\mathbf{u}, p, \lambda)$ *solves* (3.12)–(3.14), *then* $(\mathbf{u}, p)$ *solves* (3.1)–(3.2).

The multiscale approach to the mortar mixed finite element method combines a local fine scale discretization within each subdomain with a global coarse scale discretization across subdomain interfaces. [9]

First, independently partition each subdomain D_i into its own local d-dimensional quasi-uniform affine mesh $\mathcal{T}_{h,i}$. The faces (or edges) of these meshes are spatially conforming within each subdomain, but are allowed to be non-conforming along subdomain interfaces. Let the maximal element diameter of this fine mesh be h_i, and let the global characteristic fine scale diameter be $h = \max_{i=1}^{N_D} h_i$. Denote the global fine mesh by $\mathcal{T}_h = \bigcup_{i=1}^{N_D} \mathcal{T}_{h,i}$. Let $\mathbf{V}_{h,i} \times W_{h,i} \subset \mathbf{V}_i \times W_i$ be a mixed finite element space (*cf.* Remark 2.2.1) on the mesh $\mathcal{T}_{h,i}$ such that $\mathbf{V}_{h,i}$ contains piecewise polynomials of degree k and $W_{h,i}$ contains piecewise polynomials of degree l. Globally, the discrete pressure and velocity spaces for this method are $W_h = \bigoplus_{i=1}^{N_D} W_{h,i}$ and $\mathbf{V}_h = \bigoplus_{i=1}^{N_D} \mathbf{V}_{h,i}$.

Second, we partition each subdomain interface $\Gamma_{i,j}$ with a $(d-1)$-dimensional quasi-uniform affine mesh denoted $\mathcal{T}_{H,i,j}$. This mesh will be the mortar space that weakly enforces continuity of normal fluxes for the discrete velocities across the non-matching grids. Let the maximal element diameter of this coarse mesh be $H_{i,j}$, and let the global characteristic coarse scale diameter be $H = \max_{1 \le i < j \le N_D} H_{i,j}$. Denote the global coarse mesh by $\mathcal{T}_H = \bigcup_{1 \le i < j \le N_D} \mathcal{T}_{H,i,j}$. Let $M_{H,i,j} \subset L^2(\Gamma_{i,j})$ be the mortar space containing continuous or discontinuous piecewise polynomials of degree m where $m \ge k+1$. Globally, the mortar space for this method is $M_H = \bigoplus_{1 \le i < j \le N_D} M_{H,i,j}$. Notice that this is a nonconforming approximation, as $M_H \not\subseteq M$.

With these finite dimensional subspaces, the multiscale mortar mixed finite element approximation of (3.12)–(3.14) is to find $\mathbf{u}_h \in \mathbf{V}_h^{g_N}$, $p_h \in W_h$, and $\lambda_H \in M_H$ such that for

$i = 1, \ldots, N_D,$

$$(K^{-1}\mathbf{u}_h, \mathbf{v})_{D_i} - (p_h, \nabla \cdot \mathbf{v})_{D_i} = -\langle \mathbf{v} \cdot \mathbf{n}_i, \lambda_H \rangle_{\partial D_i \cap \Gamma} - \langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \; \forall \mathbf{v} \in \mathbf{V}^0_{h,i}, \tag{3.15}$$

$$(\nabla \cdot \mathbf{u}_h, w)_{D_i} = (f, w)_{D_i} \quad \forall w \in W_{h,i}, \tag{3.16}$$

$$\sum_{i=1}^{N_D} \langle \mathbf{u}_{h,i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} = 0 \quad \forall \mu \in M_H. \tag{3.17}$$

In this formulation the pressure continuity (3.7) is modeled via the mortar pressure function λ_H, while the flux continuity (3.8) is imposed weakly on the coarse scale via (3.17). For the above method to be well posed, the two scales must be chosen such that the mortar space is not too rich compared to the normal traces of the subdomain velocity spaces.

Assumption 3.1.1. *Assume there exists a constant C independent of h and H such that*

$$\|\mu\|_{\Gamma_{i,j}} \leq C(\|\mathcal{Q}_{h,i}\mu\|_{\Gamma_{i,j}} + \|\mathcal{Q}_{h,j}\mu\|_{\Gamma_{i,j}}), \quad \forall \mu \in M_H, \quad 1 \leq i < j \leq N_D,$$

where $\mathcal{Q}_{h,i} : L^2(\Gamma_i) \to \mathbf{V}_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i}$ is the L^2-projection operator from the mortar space onto the normal trace of the velocity space on subdomain i, i.e. *for any $\phi \in L^2(\Gamma_i)$,*

$$\langle \phi - \mathcal{Q}_{h,i}\phi, \mathbf{v} \cdot \mathbf{n}_i \rangle_{\Gamma_i} = 0, \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}. \tag{3.18}$$

This condition can be easily satisfied in practice by restricting the size of H from below, see for example [93, 8, 69]. Under the above assumption, method (3.15)-(3.17) is solvable, stable, and accurate [9]. The following result has been shown in [9].

Theorem 3.1.1. *If Assumption 3.1.1 holds, then method (3.15)-(3.17) has a unique solution and there exists a positive constant C, independent of h and H, such that*

$$\|\nabla \cdot (\mathbf{u} - \mathbf{u}_h)\| \leq C \sum_{i=1}^{N_D} \|\nabla \cdot \mathbf{u}\|_{r,D_i} h^r, \quad 0 \leq r \leq l+1, \tag{3.19}$$

$$\|\mathbf{u} - \mathbf{u}_h\| \leq C \sum_{i=1}^{N_D} \big(\|p\|_{s+1/2,D_i} H^{s-1/2} + \|\mathbf{u}\|_{r,D_i} h^r + \|\mathbf{u}\|_{r+1/2,D_i} h^r H^{1/2} \big), \quad 1 \leq r \leq k+1, \; 0 < s \leq m+1. \tag{3.20}$$

Furthermore, if the problem on D is H^2-regular, then

$$\|\hat{p} - p_h\| \leq C \sum_{i=1}^{N_D} \big(\|p\|_{s+1/2,D_i} H^{s+1/2} + \|\nabla \cdot \mathbf{u}\|_{t,D_i} h^t H \tag{3.21}$$
$$+ \|\mathbf{u}\|_{r,D_i} h^r H + \|\mathbf{u}\|_{r+1/2,D_i} h^r H^{3/2} \big),$$
$$\|p - p_h\| \leq C \sum_{i=1}^{N_D} \|p\|_{t,D_i} h^t + \|\hat{p} - p_h\|, \tag{3.22}$$

where $1 \leq r \leq k+1$, $0 < s \leq m+1$, *and* $0 \leq t \leq l+1$ *and* $\hat{p}$ *is the* L^2*-projection of* p *onto* W_h.

3.1.1 Interface Formulation and Iteration

Following [42], we formulate (3.15)-(3.17) as an interface problem for the mortar pressure. We decompose the solution to (3.15)-(3.17) into two parts: $\mathbf{u}_h = \mathbf{u}_h^*(\lambda_H) + \overline{\mathbf{u}}_h$ and $p_h = p_h^*(\lambda_H) + \overline{p}_h$. The first component $(\mathbf{u}_h^*, p_h^*) \in \mathbf{V}_h^0 \times W_h$ solves subdomain problems with zero source and boundary conditions, and has λ_H as a Dirichlet boundary condition along Γ, *i.e.* for $i = 1, \ldots, N_D$

$$\left(K^{-1}\mathbf{u}_h^*, \mathbf{v}\right)_{D_i} - \left(p_h^*, \nabla \cdot \mathbf{v}\right)_{D_i} = -\left\langle \mathbf{v} \cdot \mathbf{n}_i, \lambda_H \right\rangle_{\partial D_i \cap \Gamma} \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0, \tag{3.23}$$
$$\left(\nabla \cdot \mathbf{u}_h^*, w\right)_{D_i} = 0 \quad \forall w \in W_{h,i}. \tag{3.24}$$

The second component $(\overline{\mathbf{u}}_h, \overline{p}_h) \in \mathbf{V}_h^{g_N} \times W_h$ solves subdomain problems with source f, boundary conditions g_D and g_N on ∂D, and zero Dirichlet boundary conditions along Γ, *i.e.* for $i = 1, \ldots, N_D$

$$(K^{-1}\overline{\mathbf{u}}_h, \mathbf{v})_{D_i} - (\overline{p}_h, \nabla \cdot \mathbf{v})_{D_i} = -\langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0, \tag{3.25}$$
$$(\nabla \cdot \overline{\mathbf{u}}_h, w)_{D_i} = (f, w)_{D_i} \quad \forall w \in W_{h,i}. \tag{3.26}$$

Since the sum of (3.23)–(3.24) and (3.25)–(3.26) gives (3.15)–(3.16), all that remains to do is enforce equation (3.17). Thus, the variational interface problem is to find $\lambda_H \in M_H$ such that

$$\sum_{i=1}^{N_D} \langle -\mathbf{u}_{h,i}^*(\lambda_H) \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} = \sum_{i=1}^{N_D} \langle \overline{\mathbf{u}}_{h,i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i}, \quad \forall \mu \in M_H. \tag{3.27}$$

Equivalently, we define bilinear forms $b_{H,i} : L^2(\Gamma_i) \times L^2(\Gamma_i) \to \mathbb{R}$ and $b_H : L^2(\Gamma) \times L^2(\Gamma) \to \mathbb{R}$ and a linear functional $g_H : L^2(\Gamma) \to \mathbb{R}$ by

$$b_{H,i}(\lambda_{H,i}, \mu) = \langle -\mathbf{u}^*_{h,i}(\lambda_{H,i}) \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i}, \tag{3.28}$$

$$b_H(\lambda_H, \mu) = \sum_{i=1}^{N_D} b_{H,i}(\lambda_{H,i}, \mu), \tag{3.29}$$

$$g_H(\mu) = \sum_{i=1}^{N_D} \langle \overline{\mathbf{u}}_{h,i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i}. \tag{3.30}$$

With these definitions, (3.27) is equivalent to finding $\lambda_H \in M_H$ such that $b_H(\lambda_H, \mu) = g_H(\mu)$, for all $\mu \in M_H$. The distinction is made between bilinear forms (3.28) and (3.29) because the former measures the total flux across interface Γ_i and requires no interprocessor communication, while the latter measures the total *jump* in flux across the set of all interfaces Γ and hence does require interprocessor communication.

It is easy to check that b_H is symmetric and positive semi-definite on $L^2(\Gamma)$. Moreover, it is positive definite on M_H if Assumption 3.1.1 holds and $\Gamma_D \neq \emptyset$ [8, 9]. Therefore we solve the resulting discrete system with a Conjugate Gradient (CG) algorithm. Define linear operators $B_{H,i} : M_{H,i} \to M_{H,i}$ and $B_H : M_H \to M_H$ and a vector $g_H \in M_H$ corresponding to equations (3.28), (3.29), and (3.30) by

$$\langle B_{H,i}\lambda_{H,i}, \mu \rangle_{\Gamma_i} = -\langle \mathbf{u}^*_{h,i}(\lambda_{H,i}) \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} \ \forall \mu \in M_{H,i}, \tag{3.31}$$

$$B_H\lambda_H = \sum_{i=1}^{N_D} B_{H,i}\lambda_{H,i}, \tag{3.32}$$

$$\langle g_H, \mu \rangle_\Gamma = \sum_{i=1}^{N_D} \langle \overline{\mathbf{u}}_{h,i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} \ \forall \mu \in M_H. \tag{3.33}$$

Let $\mathcal{Q}^T_{h,i} : \mathbf{V}_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i} \to M_{H,i}$ be the L^2-orthogonal projection from the normal trace of the velocity space onto the mortar space. Note that (3.31) and (3.33) imply, respectively,

$$B_{H,i} = -\mathcal{Q}^T_{h,i}\mathbf{u}^*_{h,i}(\lambda_{H,i}) \cdot \mathbf{n}_i, \quad g_H = \sum_{i=1}^{N_D} \mathcal{Q}^T_{h,i}\overline{\mathbf{u}}_{h,i} \cdot \mathbf{n}_i. \tag{3.34}$$

Using this notation, the interface formulation is to find $\lambda_H \in M_H$ such that

$$B_H\lambda_H = g_H. \tag{3.35}$$

The operator B_H is known as the Steklov-Poincaré operator [71].

Starting from an initial guess, we iterate on the value of λ_H using the CG algorithm. On each CG iteration, we must evaluate the action of B_H on λ_H. This is process is summarized in Algorithm 3.1. Lines 3-5 evaluate the action of the flux operator $B_{H,i}$ as in (3.34) and can be done by every subdomain in parallel (thus, eliminating the for-loop). Line 7 evaluates the action of the jump operator B_H as in (3.32) and requires interprocessor communication across every subdomain interface.

Algorithm 3.1 Original MMMFEM Interface Iteration.

1: **while** (CG_resid $<$ TOL) **do**

2: **for** $i = 1, \ldots, N_D$ **do**

3: Project mortar data onto subdomain boundaries: $\lambda_{H,i} \overset{\mathcal{Q}_{h,i}}{\rightarrow} \gamma_i$.

4: Solve the set of subdomain problems (3.23)–(3.24) with Dirichlet boundary data γ_i.

5: Project the resulting fluxes back onto the mortar space:

$$-\mathbf{u}_h^*(\gamma_i) \cdot \mathbf{n}_i \overset{\mathcal{Q}_{h,i}^T}{\rightarrow} -\mathcal{Q}_{h,i}^T \mathbf{u}_h^*(\gamma_i) \cdot \mathbf{n}_i.$$

6: **end for**

7: Compute flux jumps across all subdomain interfaces $\Gamma_{i,j}$:

$$B_H \lambda_H = -\sum_{i=1}^{N_D} \mathcal{Q}_{h,i}^T \mathbf{u}_h^*(\gamma_i) \cdot \mathbf{n}_i.$$

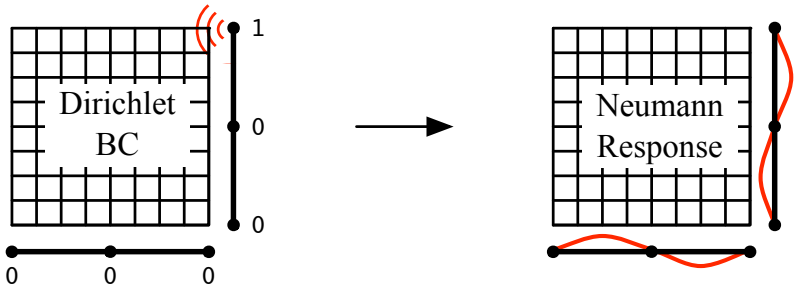
8: **end while**

3.2 THE MULTISCALE FLUX BASIS IMPLEMENTATION

The dominant computational cost in each CG iteration is in the evaluation of the flux operator $B_{H,i}$, which requires solving one Dirichlet subdomain problem per subdomain. One way to potentially reduce this computational cost is with the following approach:

Before the CG algorithm begins, compute and store the flux responses associated with each mortar degree of freedom, on each subdomain independently.

Figure 3.1: Illustration of the Multiscale Flux Basis approach.



This is what we call the Multiscale Flux Basis. Its assembly requires solving only a *fixed* number of Dirichlet subdomain problems (3.23)–(3.24), shown in Figure 3.1. After these solves are completed, the action of $B_{H,i}$ is reduced to taking a linear combination of Multiscale Flux Basis functions. Therefore, if the number of CG iterations exceeds the maximum number of mortar degrees of freedom on any subdomain, then the computational cost will be reduced by requiring fewer subdomain solves, and should yield faster runtime.

Let there be $N_{\text{dof}}(i)$ mortar degrees of freedom on subdomain D_i and define $\{\phi_{H,i}^{(k)}\}_{k=1}^{N_{\text{dof}}(i)}$ to be the mortar basis functions for $M_{H,i}$. Then for $\lambda_{H,i} \in M_{H,i}$ we may express

$$\lambda_{H,i} = \sum_{k=1}^{N_{\text{dof}}(i)} \lambda_{H,i}^{(k)} \phi_{H,i}^{(k)}.$$

The Multiscale Flux Basis $\{\psi_{H,i}^{(k)}\}_{k=1}^{N_{\text{dof}}(i)}$ is formed by applying Steps 3-5 from Algorithm 3.1 to evaluate the action of the operator $B_{H,i}$ on each mortar basis function $\phi_{H,i}^{(k)}$, on each subdomain independently. The formation of the Multiscale Flux Basis is summarized in Algorithm 3.2.

There are several remarks to be made about this procedure. Note that each mortar basis function $\phi_{H,i}^{(k)}$ on interface $\Gamma_{i,j}$ corresponds to exactly two different Multiscale Flux Basis

Algorithm 3.2 Formation of a Multiscale Flux Basis.

1: **for** $i = 1, \ldots, N_D$ **do**

2: **for** $k = 1, \ldots, N_{\text{dof}}(i)$ **do**

3: Project mortar basis function onto subdomain boundary: $\mathcal{Q}_{h,i}\phi_{H,i}^{(k)} = \gamma_i^{(k)}$.

4: Solve subdomain problem (3.23)–(3.24) with Dirichlet data $\gamma_i^{(k)}$, *i.e.* find $\mathbf{u}_h^* = \mathbf{u}_h^*(\gamma_i^{(k)})$ and $p_h^* = p_h^*(\gamma_i^{(k)})$ such that

$$(K^{-1}\mathbf{u}_h^*, \mathbf{v})_{D_i} - (p_h^*, \nabla \cdot \mathbf{v})_{D_i} = -\langle \mathbf{v} \cdot \mathbf{n}_i, \gamma_i^{(k)} \rangle_{\partial D_i \cap \Gamma} \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0,$$
$$(\nabla \cdot \mathbf{u}_h^*, w)_{D_i} = 0 \quad \forall w \in W_{h,i}.$$

5: Project the resulting boundary flux back onto the mortar space:

$$\psi_{H,i}^{(k)} = -\mathcal{Q}_{h,i}^T \mathbf{u}_h^*(\gamma_i^{(k)}) \cdot \mathbf{n}_i.$$

6: **end for**

7: **end for**

functions, one for D_i and one for D_j. Whereas the original MMMFEM implementation requires each processor to perform the exact same number of subdomain solves during the interface iteration process, our implementation may have each processor perform a different number of subdomain solves in assembling its Multiscale Flux Basis. This is because there may be a varying number of degrees of freedom in each mortar $M_{H,i,j}$ and subdomains may share portions of their boundaries with Γ_D and Γ_N. The Multiscale Flux Basis in this method consists of coarse interface fluxes, as opposed to fine scale subdomain data. The extra storage cost associated with saving this basis is equal to the square of the size of the mortar space on each subdomain, $N_{\text{dof}}(i)^2$. More specifically, each multiscale flux basis function belongs to the mortar space $M_{H,i}$ of dimension $N_{\text{dof}}(i)$, and there are exactly $N_{\text{dof}}(i)$ basis functions to be computed. Therefore the storage cost for the Multiscale Flux Basis is significantly lower than the storage cost in the variational multiscale method and multiscale finite elements, where the basis functions are defined on the entire local fine grid.

To use the Multiscale Flux Basis in the interface iteration, we need only observe that

the flux operator $B_{H,i}$ is linear. Therefore,

$$B_{H,i}\left(\lambda_{H,i}\right) = B_{H,i}\left(\sum_{k=1}^{N_{\mathrm{dof}}(i)} \lambda_{H,i}^{(k)} \phi_{H,i}^{(k)}\right) = \sum_{k=1}^{N_{\mathrm{dof}}(i)} \lambda_{H,i}^{(k)} B_{H,i}\left(\phi_{H,i}^{(k)}\right) = \sum_{k=1}^{N_{\mathrm{dof}}(i)} \lambda_{H,i}^{(k)} \psi_{H,i}^{(k)}. \tag{3.37}$$

In other words, to compute the resulting flux on subdomain D_i from Dirichlet data $\lambda_{H,i}$, we simply take a linear combination of the Multiscale Flux Basis functions $\psi_{H,i}^{(k)}$ using the same scalars which express $\lambda_{H,i}$ in terms of its mortar basis functions $\phi_{H,i}^{(k)}$. This demonstrates the equivalence of the original MMMFEM implementation to our new Multiscale Flux Basis implementation. The process is summarized in Algorithm 3.3.

Algorithm 3.3 Interface Iteration with Multiscale Flux Basis.

1: Form Multiscale Flux Basis $\{\psi_{H,i}^{(k)}\}_{k=1}^{N_{\mathrm{dof}}(i)}$ with Algorithm 3.2.

2: **while** (CG_resid $<$ TOL) **do**

3: **for** $i = 1, \ldots, N_D$ **do**

4: Perform linear combination to evaluate subdomain flux

$$B_{H,i}\left(\lambda_{H,i}\right) = \sum_{k=1}^{N_{\mathrm{dof}}(i)} \lambda_{H,i}^{(k)} \psi_{H,i}^{(k)}.$$

5: **end for**

6: Compute flux jumps across all subdomain interfaces $\Gamma_{i,j}$

$$B_H \lambda_H = \sum_{i=1}^{N_D} B_{H,i}\left(\lambda_{H,i}\right).$$

7: **end while**

8: Perform additional subdomain problem (3.23)–(3.24) to recover fine scale variables.

In practice, one may store the Multiscale Flux Basis in a matrix and compute Step 4 with an optimized matrix-vector product, *e.g.* the one available in the popular BLAS library. In the original MMMFEM implementation, fine scale pressure and velocity variables may also be updated iteratively in the interface iteration. In the new Multiscale Flux Basis implementation, this convention should be dropped, because storing arrays of these fine scale variables for each mortar degree of freedom would be an unnecessary burden on memory. Instead, in Step 8 we perform one additional Dirichlet subdomain solve after the CG iteration has converged in order to recover the fine scale pressure and velocity.

3.3 NUMERICAL EXAMPLES

To perform the numerical experiments in this chapter, the Multiscale Flux Basis implementation was coded into the PARCEL simulator, and the simulations were run in parallel with one subdomain per processor. On the subdomain interfaces, the coarse grid is comprised of continuous or discontinuous, linear or quadratic mortar spaces. Unless otherwise noted, the tolerance for the relative residual in the CG algorithm is taken to be $1E-06$.

We compare the computational efficiency of the new Multiscale Flux Basis implementation to the original implementation. Since the number of interface iterations in the original implementation is directly related to the number of subdomain solves, we compare to both non-preconditioned and preconditioned methods. On the other hand, the Multiscale Flux Basis implementation shifts the workload from the number of interface iterations to the the number of interface degrees of freedom per subdomain; hence we do not employ a preconditioner in this method. More precisely, the following three numerical methods are compared:

- **Method D1**. Original MMMFEM implementation, no interface preconditioner.
- **Method D2**. Original MMMFEM implementation, balancing preconditioner.
- **Method D3**. New Multiscale Flux Basis implementation, no preconditioner.

The balancing preconditioner used in the tests has been described in [26, 69, 9]. It involves solving Neumann subdomain problems and a course problem which provides global exchange of information across subdomains. This causes the condition number of the interface problem to grow more modestly versus non-preconditioned CG as the grids are refined or the number of subdomains increases. The cost for one preconditioned iteration is three subdomain solves and two coarse solves.

Four example problems are considered: a 2-D problem with smooth permeability, a 2-D problem with a rough permeability, a 3-D problem with smooth permeability, and a 2-D problem with adaptive mesh refinement. In the first three examples we solve each problem using a fixed fine grid several times. Each time we increase the number of subdomains, *i.e.*, refine the coarse grid. It should be noted that under a fixed fine grid, as the number of subdomains is increased, the size of the local subdomain problems becomes smaller. This

causes the interface problem to become larger and more ill-conditioned, hence increasing the number of CG iterations. Tables are provided which compare both the number of CG iterations and maximum number of subdomain solves required by the three methods. In this way, the new multiscale flux basis implementation can be directly compared to the original MMMFEM implementation. No error norms are reported in these tests, because all three methods produce the same solution within roundoff error. For the first two examples we also provide tests comparing the accuracy and the cost of the MMMFEM solution to a fine scale solution. The fourth example involves adaptive mesh refinement and illustrates the greater flexibility of the MMMFEM compared to existing multiscale methods. It also shows that the gain in efficiency from the new implementation is increased when grid adaptivity is employed.

3.3.1 2-D Smooth Solution Example

This example is a 2-D problem on the domain $D = (0,1)^2$ with a fixed global fine grid of 120×120 elements. The solution is given by $p(x,y) = x^3y^4 + x^2 + \sin(xy)\cos(y)$, and the coefficient K is a smooth, full tensor defined by

$$K = \begin{pmatrix} (x+1)^2 + y^2 & \sin(xy) \\ \sin(xy) & (x+1)^2 \end{pmatrix}.$$

Boundaries $\{y = 0\}$ and $\{y = 1\}$ are Dirichlet type and boundaries $\{x = 0\}$ and $\{x = 1\}$ are Neumann type.

Table 3.1 shows results for Example 3.3.1 using continuous linear mortars with 3 elements per edge. Observe that the number of CG iterations increases with the number of subdomains, since the dimension of the interface problem grows. Recall that for all methods the dominant computational cost is measured by the number of subdomain solves. In Method D3 the number of subdomain solves does not depend on the number of interface iterations, only on the number of coarse scale mortar degrees of freedom per subdomain. As a result the number of subdomain solves does not change with increasing the number of subdomains (except for the 2×2 case where only two out of four edges of each subdomain have mortars). This is in contrast to the original implementation, Methods D1 and D2, where the number

Table 3.1: Computational cost of Example 3.3.1 using continuous linear mortars with 3 elements per interface.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 = 4$	14	17*	11	41	14	19
$3 \times 3 = 9$	29	32*	19	67	29	35
$4 \times 4 = 16$	42	45	24	82	42	35*
$5 \times 5 = 25$	54	57	26	88	54	35*
$6 \times 6 = 36$	65	68	27	91	64	35*
$7 \times 7 = 49$	75	78	26	88	77	35*
$8 \times 8 = 64$	86	89	26	88	86	35*

* - denotes fewest number of solves

of subdomain solves is directly related to the number of CG interface iterations. Method D1 requires one subdomain solve per iteration plus three additional subdomain solves. Method D2 requires three subdomain solves per iteration plus ten additional subdomain solves. The balancing preconditioner used in Method D2 causes the number of CG iterations to grow more modestly with the number of subdomains, but this method is still more costly in terms of subdomain solves. Method D1 performs the best of all three methods until we reach the 4×4 subdomain case. After this point Method D3 becomes the most efficient in terms of subdomain solves. This table demonstrates that as the number of subdomains is increased, there is a point after which Method D3 performs best. We found this to be the case for most tests we ran.

The Balancing preconditioner involves two additional coarse grid solves per CG iteration. Thus even in cases where Method D2 required fewer subdomain solves, Method D3 was more efficient in terms of CPU time, as the time for the coarse solves was not negligible. We do not report CPU times in this chapter, and note they depend on the particular implementation of the coarse solve in the Balancing preconditioner.

There is also a cost in runtime associated with the interprocessor communication for each interface iteration. The Multiscale Flux Basis implementation does not reduce the number of interface iterations necessary for flux matching, hence it does not have an effect on the communication overhead. This cost can be reduced by using the Multiscale Flux Basis implementation in conjunction with an efficient preconditioner.

Table 3.2: Computational cost of Example 3.3.1 using continuous quadratic mortars with 2 elements per interface.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 = 4$	16	19*	12	44	16	53
$3 \times 3 = 9$	35	38*	17	61	34	63
$4 \times 4 = 16$	51	54*	20	70	51	63
$5 \times 5 = 25$	65	68	21	73	65	63*
$6 \times 6 = 36$	78	81	22	76	78	63*
$7 \times 7 = 49$	91	94	21	73	91	63*
$8 \times 8 = 64$	103	107	21	73	103	63*

* - denotes fewest number of solves

In Table 3.2 we report results for Example 3.3.1 with continuous quadratic mortars with 2 elements per edge. This slightly increases the required work for Method D1 and slightly decreases the work for Method D2. However, for Method D3 this change nearly doubles the amount of subdomain solves required due to the increase in mortar degrees of freedom per subdomain. This means that initially our method solves more subdomain problems than the other two, and the computational efficiency of Method D3 is not observed until the 5×5 case. This difference versus the previous table shows that the number of mortar degrees of freedom per subdomain is an important parameter which determines the relative computationally efficiency of the Multiscale Flux Basis implementation.

To illustrate the accuracy of the MMMFEM and the efficiency of the proposed new implementation, we compare the quality and cost of the multiscale solution to these of the

Table 3.3: Relative errors and computational cost for the fine scale solution in Example 3.3.1.

Subdomains	pres-L2-err	vel-L2-err	CGIter	Solves
$2 \times 2 = 4$	7.1657E-05	7.1848E-05	58	123
$3 \times 3 = 9$	7.1955E-05	7.9269E-05	72	163
$4 \times 4 = 16$	7.1968E-05	8.7311E-05	85	123
$5 \times 5 = 25$	7.2211E-05	9.5265E-05	96	99
$6 \times 6 = 36$	7.2260E-05	1.0281E-04	107	83

fine scale solution. The latter is computed using the same domain decomposition algorithm with Method D3, but with fine scale Lagrange multipliers. In Table 3.3 we report the relative errors in pressure and velocity, and the cost of the interface iteration. This type of test is comparable to a standard mixed finite element algorithm without domain decomposition. Indeed, the recorded error norms remain nearly constant as the number of subdomains is increased.

In comparison, Table 3.4 shows results for the MMMFEM using Method D3 with linear mortars and a single element per interface. This subdomain configuration is very much akin to the variational multiscale methods and multiscale finite element methods mentioned in the introduction. We note that the MMMFEM requires significantly smaller number of subdomain solves, while at the same time resolves the flow very well, as seen in Figure 3.2 where a comparison of the plots of the computed fine scale and multiscale solutions with 5×5 subdomains is shown. The relative error norms reported in Table 3.4 indicate that the error is larger for the multiscale solution, but does decrease as the number of subdomain is increased. Figure 3.3 shows that the locations with greatest error in the multiscale solution are along the subdomain interfaces.

Table 3.4: Relative error and computational cost for the multiscale solution using Method D3 with a single linear mortar per interface in Example 3.3.1.

Subdomains	pres-L2-err	vel-L2-err	CGIter	Solves
$2 \times 2 = 4$	1.2966E-02	4.4386E-02	8	11
$3 \times 3 = 9$	7.1036E-03	3.6534E-02	22	19
$4 \times 4 = 16$	4.2496E-03	3.0038E-02	33	19
$5 \times 5 = 25$	2.7673E-03	2.5191E-02	42	19
$6 \times 6 = 36$	1.9159E-03	2.1527E-02	51	19

Figure 3.2: Computed pressure (color) and velocity (arrows) in Example 3.3.1: fine scale solution (left) and multiscale solution with a single linear mortar per interface (right).

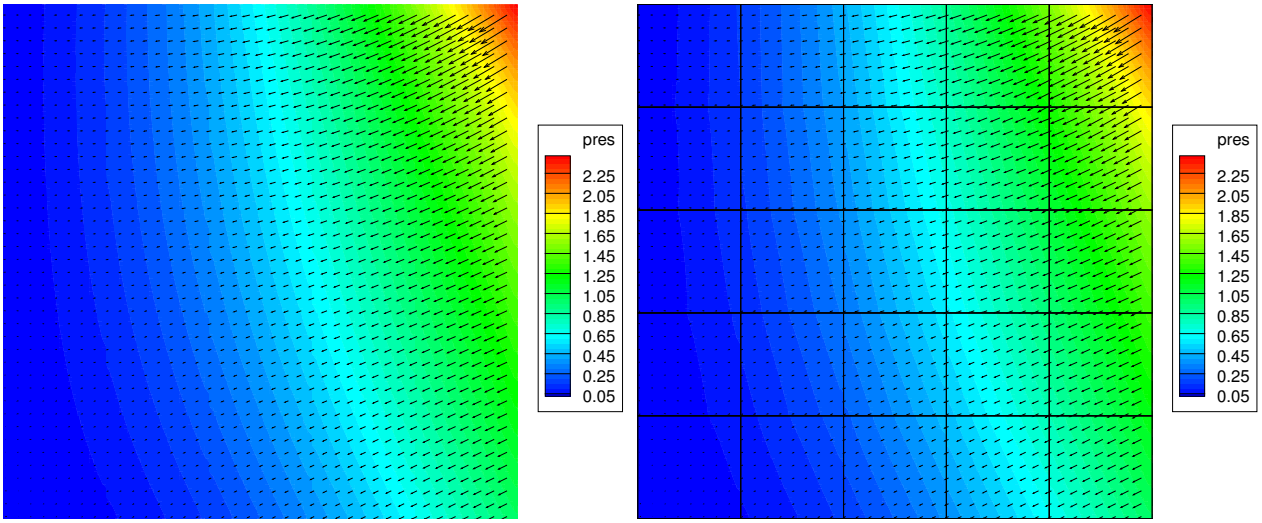
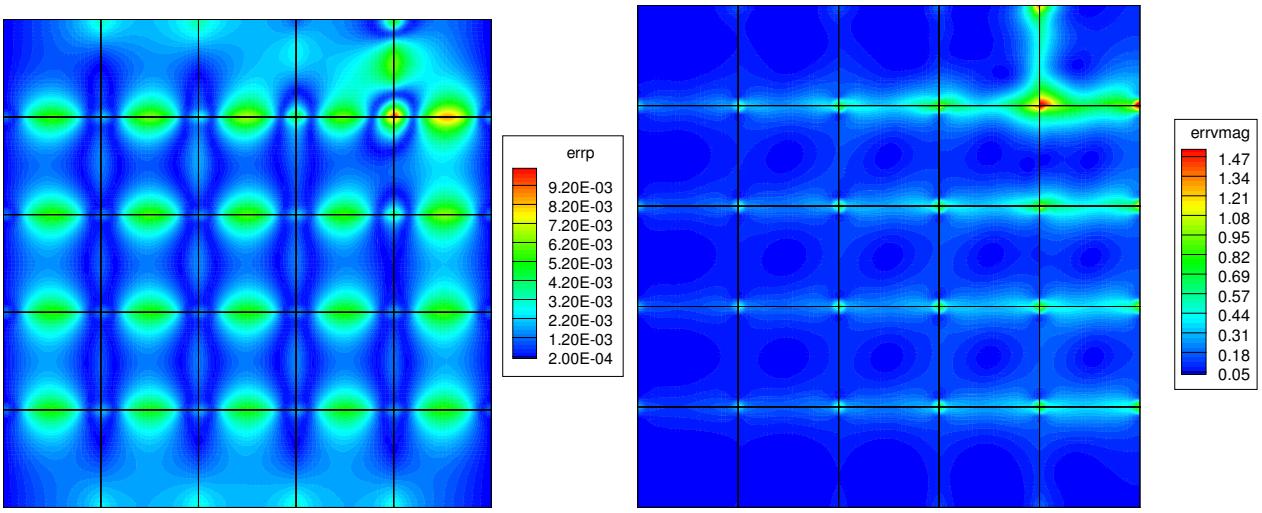


Figure 3.3: Pressure error (left) and magnitude of velocity error (right) for the multiscale solution with a single linear mortar per interface for Example 3.3.1.



3.3.2 2-D Rough Heterogeneous Permeability Example

This problem uses a 2-D heterogeneous permeability field, obtained from the Society of Petroleum Engineers (SPE) Comparative Solution Project [22]. The domain is $D = (0, 60) \times (0, 220)$ with a fixed global fine grid of 60×220 elements. Pressure values of one and zero are specified on the left and right boundaries, respectively. No flow is specified on the top and bottom boundaries. A plot of the permeability field is shown on the left in Figure 3.4.

Table 3.5 shows the results for Example 3.3.2 using continuous linear mortars with 2 elements per edge. Method D3 requires at most 26 solves per subdomain and is computationally more efficient than Methods D1 and D2 for all subdomain configurations. As the number of subdomains is increased, the improvement of Method D3 over Methods D1 and D2 becomes greater.

A comparison between the fine scale solution and the multiscale solution with 3×5 subdomains is presented in Figure 3.4. We observe a very good match between the two solutions. We note that the number of subdomain solves required by Method D3 for the multiscale solution, 26, is significantly less than Methods D1–D3 used for computing the fine

Figure 3.4: Permeability field (left), fine scale solution (middle), and multiscale solution with 3×5 subdomains and a single linear mortar per interface (right) for Example 3.3.2.

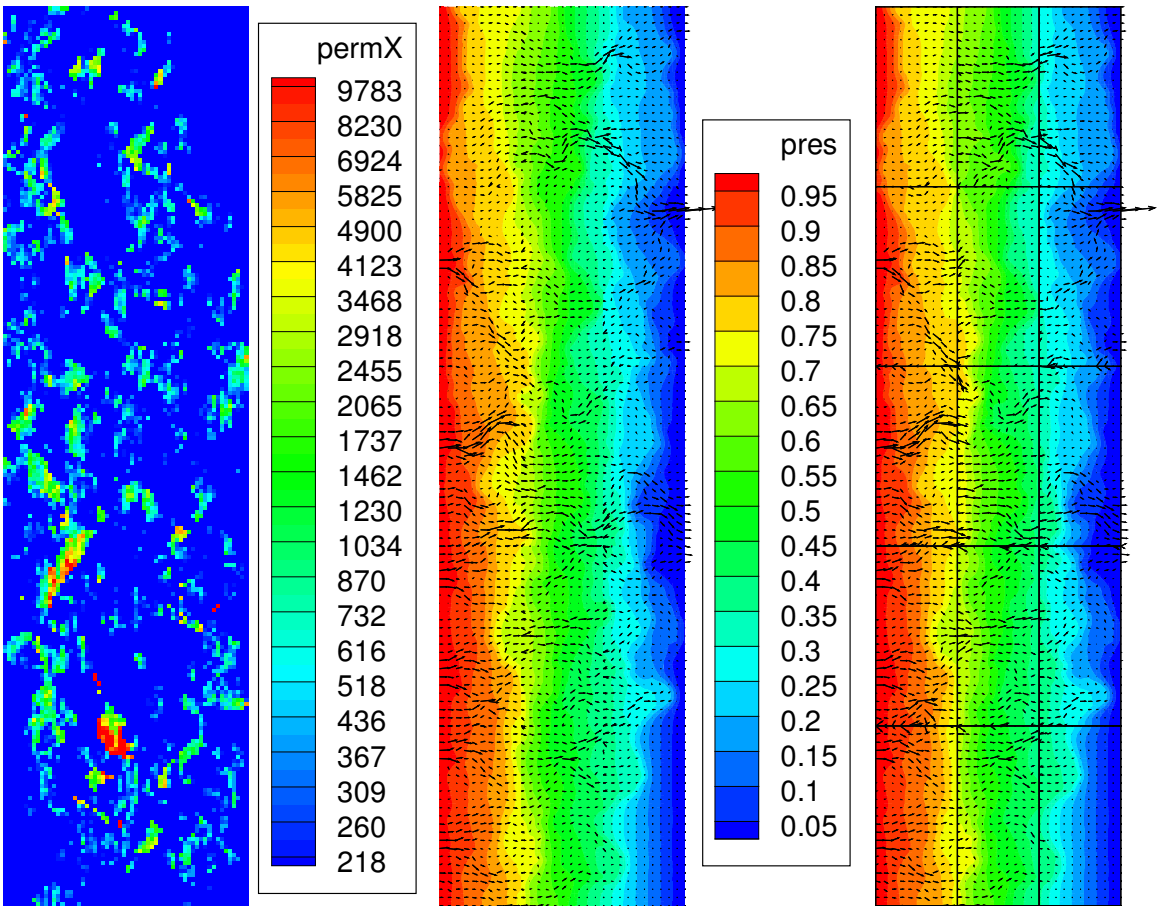


Table 3.5: Computational cost of Example 3.3.2 using continuous linear mortars with 2 elements per interface.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 = 4$	13	15	8	31	13	14*
$3 \times 2 = 6$	19	21	15	53	19	20*
$2 \times 4 = 8$	25	27	18	62	23	20*
$2 \times 5 = 10$	37	39	29	95	35	20*
$3 \times 4 = 12$	37	39	28	93	36	26*
$3 \times 5 = 15$	51	53	37	120	51	26*

* - denotes fewest number of solves

scale solution, which require 388, 84, and 130 subdomain solves, respectively.

Table 3.6: Computational cost of Example 3.3.2 using continuous quadratic mortars with 2 elements per interface.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 = 4$	17	19*	15	52	16	32
$3 \times 2 = 6$	30	32*	23	77	31	47
$2 \times 4 = 8$	39	41*	25	83	38	47
$2 \times 5 = 10$	56	58	39	125	56	47*
$3 \times 4 = 12$	53	55	33	108	52	62*
$3 \times 5 = 15$	92	94	46	147	92	62*

* - denotes fewest number of solves

Table 3.6 shows the results for Example 3.3.2 using continuous quadratic mortars with 2 elements per interface. Compared to the previous table, the increased number of mortar

degrees of freedom per interface leads to more subdomain solves for Method D3, the maximum number being 62. Nevertheless, Method D3 is still more computationally efficient than Methods D1 and D2 for 10 and more subdomains.

3.3.3 3-D Smooth Solution Example

This example is a 3-D problem on the domain $D = (0,1)^3$ with a fixed global fine grid of $48 \times 48 \times 48$ elements. The solution is given by $p(x,y,z) = x+y+z-1.5$, and the coefficient K is a smooth full tensor defined by

$$K = \begin{pmatrix} x^2+y^2+1 & 0 & 0 \\ 0 & z^2+1 & \sin(xy) \\ 0 & \sin(xy) & x^2y^2+1 \end{pmatrix}.$$

Boundaries $\{y=0\}$ and $\{y=1\}$ are Dirichlet type and the rest of the boundary is Neumann type.

Figure 3.5 shows the computed multiscale solution and its error for Example 3.3.3 with $4\times 4\times 4$ subdomains and a single linear mortar per interface. Table 3.7 shows the computational cost for Methods D1–D3 with various coarse grids. Method D3 requires at most 27 solves per subdomain and outperforms Methods D1 and D2 for all subdomain configurations.

Table 3.8 shows the results for Example 3.3.3 using quadratic mortars with one element per interface with the usual relative residual CG tolerance of $1E-06$. Method D3 requires at most 57 solves per subdomain. It is the fastest method on coarser domain decompositions, but Method D2 outperforms it slightly on 27 or more subdomains.

When a tighter tolerance is imposed on the CG on the interface, all three methods perform more CG iterations. Under Methods D1 and D2, this also requires performing more subdomain solves. For Method D3, however, the maximum number of solves per subdomain is unaffected by this change in tolerance. This is illustrated in Table 3.9, which shows the results for relative residual CG tolerance of $1E-09$. In this case Method D3 is the most computationally efficient for all subdomain configurations.

Figure 3.5: Computed multiscale solution (left) and its error (right) on $4 \times 4 \times 4$ subdomains with a single linear mortar per interface for Example 3.3.3.

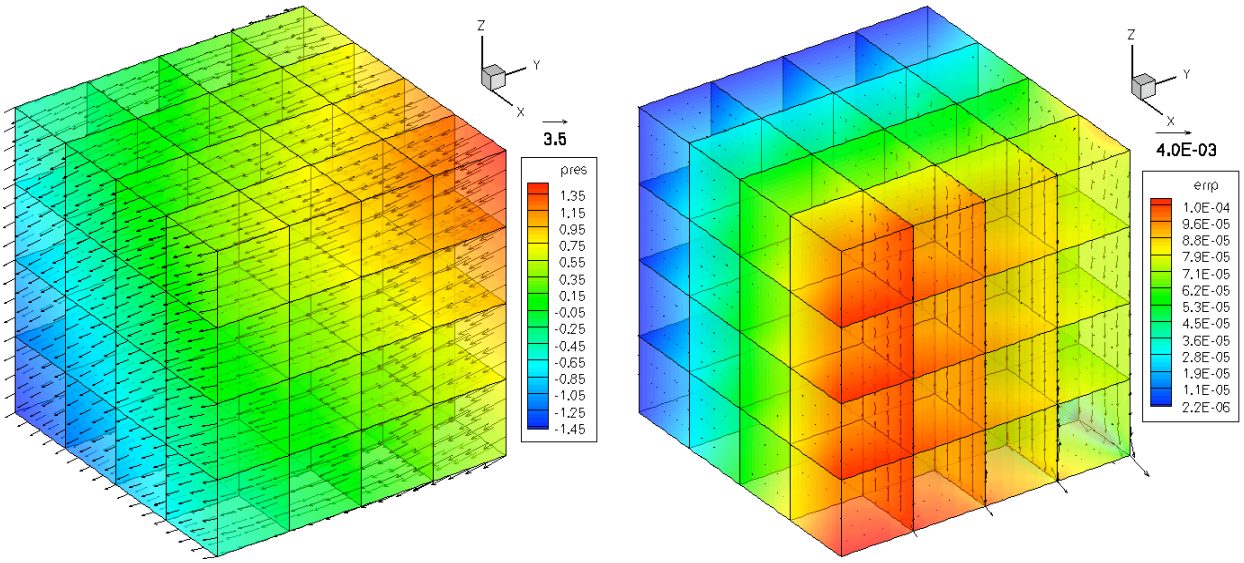


Table 3.7: Computational cost of Example 3.3.3 using linear mortars with one element per interface.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 \times 2 = 8$	28	31	11	42	28	15*
$2 \times 2 \times 3 = 12$	33	36	12	46	33	19*
$2 \times 3 \times 3 = 18$	37	40	13	50	37	23*
$3 \times 3 \times 3 = 27$	46	49	13	51	46	27*
$3 \times 3 \times 4 = 36$	50	53	13	51	50	27*
$3 \times 4 \times 4 = 48$	55	58	13	51	55	27*
$4 \times 4 \times 4 = 64$	60	63	13	51	60	27*

* - denotes fewest number of solves

Table 3.8: Computational cost of Example 3.3.3 using quadratic mortars with one element per interface. Relative residual CG tolerance $= 1E-06$.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 \times 2 = 8$	36	39	13	48	36	30*
$2 \times 2 \times 3 = 12$	41	44	13	49	41	39*
$2 \times 3 \times 3 = 18$	47	50	14	53	47	48*
$3 \times 3 \times 3 = 27$	56	59	14	54*	56	57
$3 \times 3 \times 4 = 36$	60	63	14	54*	60	57
$3 \times 4 \times 4 = 48$	64	67	14	54*	64	57
$4 \times 4 \times 4 = 64$	69	72	14	54*	69	57

* - denotes fewest number of solves

Table 3.9: Computational cost of Example 3.3.3 using quadratic mortars with one element per interface. Relative residual CG tolerance $= 1E-09$.

	Method D1		**Method D2**		**Method D3**	
Subdomains	CGIter	Solves	CGIter	Solves	CGIter	Solves
$2 \times 2 \times 2 = 8$	48	51	19	66	48	30*
$2 \times 2 \times 3 = 12$	56	59	19	67	56	39*
$2 \times 3 \times 3 = 18$	62	65	21	74	62	48*
$3 \times 3 \times 3 = 27$	74	77	20	72	74	57*
$3 \times 3 \times 4 = 36$	79	82	21	75	79	57*
$3 \times 4 \times 4 = 48$	84	87	21	75	85	57*
$4 \times 4 \times 4 = 64$	92	95	21	75	92	57*

* - denotes fewest number of solves

3.3.4 Mesh Adaptivity Example

The final example illustrates an increased computational benefit from the MMMFEM when adaptive mesh refinement is utilized. By using the multiscale flux basis implementation on each refinement level, the overall computational savings are compounded. In this example, residual based *a posteriori* error indicators are used to refine only those subdomains where the error is highest. Mortars that touch refined subdomains are also refined in order to maintain accuracy. This approach has been shown to be both efficient and reliable, see [87, 9] for details.

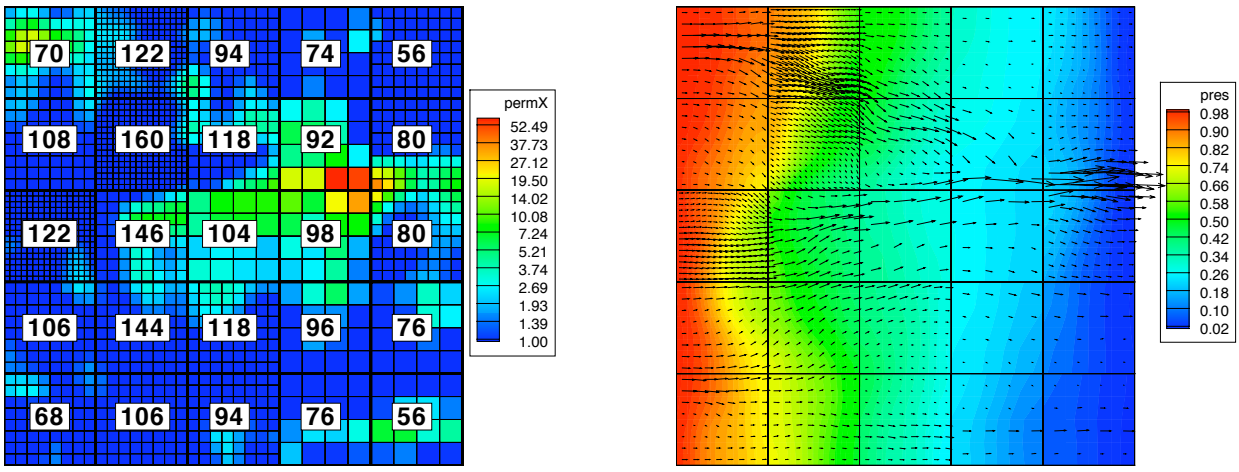
The permeability K is a single Monte Carlo realization of a stochastic permeability field on the domain $D = (0,1)^2$. A Karhunen-Loève expansion for the log permeability $Y = \ln(K)$ is computed from the covariance function (2.1). The parameters used for this test are correlation lengths $\eta_1 = 0.25$, $\eta_2 = 0.125$, and variance $\sigma_Y = 2.1$. The series was truncated after $N_{\text{term}} = 400$ terms. We specifically chose to generate the permeability in this example as a realization of a KL expansion, so that no upscaling or homogenization would be needed. On each level of mesh refinement, we are able to analytically evaluate a very heterogeneous permeability on an arbitrarily fine grid using the analytic procedure described in Section 2.1.

This test was performed on $5 \times 5 = 25$ subdomains, initially starting with 2×2 fine grids and continuous linear mortars with one element per edge. The permeability field and its corresponding solution on the fourth level of mesh refinement are shown in Figure 3.6. Note that this adaptive procedure leads to different scales being resolved on different subdomains, providing a truly multiscale approximation. After refinement, one can see the subdomains now have 4×4, 8×8, and 16×16 fine grids.

Using Method D1, each subdomain performed 283 subdomain solves, roughly one for each CG iteration on each of the 4 grid levels. Using Method D3, the number of subdomain solves after 4 levels of mesh refinement is shown in the figure on top of the permeability plot. The maximum number of subdomain solves is 160 and the minimum number is 56.

We can draw two conclusions from Example 3.3.4. First, since the computational savings of the multiscale flux basis implementation occurs on each level of adaptive mesh refinement,

Figure 3.6: Permeability field for Example 3.3.4 on mesh refinement level 4 (left) and its corresponding solution (right). Numbers indicate the total number of subdomain solves using the Multiscale Flux Basis implementation.



the overall savings after all levels are complete is amplified by the number of refinement levels. Second, the workload for each processor may become increasingly unbalanced due to a large variation in the number of mortar degrees of freedom per subdomain. Nevertheless, even if the algorithm is only as fast as its slowest processor, the Multiscale Flux Basis implementation is still faster than the original implementation. One can take full advantage of the computational efficiency of the new method in adaptive mesh refinement setting by implementing load balancing.

4.0 STOCHASTIC COLLOCATION WITH MORTAR FINITE ELEMENTS USING MULTISCALE FLUX BASIS

This chapter contains material that is to be published in [39], which both extends and combines the two previous chapters. We consider the single-phase incompressible model with stochastic permeability. We employ the Multiscale Mortar Mixed Finite Element Method coupled with the Stochastic Collocation Method on both tensor product and sparse grids. It is implemented in three possible ways using the Multiscale Flux Basis approach.

Additionally, we follow [60] in extending the random porous media from Section 2.1 to be statistically non-stationary, in which we are given several stationary covariance functions of type (2.1), each influencing different parts of the domain. We shall refer to these statistically independent zones as KL Regions, which are used to represent multiple rock types motivated by geologic features. In this framework, the covariance between any two points within a single KL region still depends on their distance only, but the covariance between any two points which lie in different KL regions is zero, *i.e.* they are uncorrelated.

The physical domain has two decompositions: KL regions for the statistical representation of the nonstationary random permeability, and subdomains for the domain decomposition of the MMMFEM. The former is a physical decomposition depending on geologic structure, and the latter is a computational decomposition depending on available computing resources. It is our choice in implementation that the subdomains conform to the KL regions, meaning that each subdomain belongs to a single KL region. Therefore the number of KL regions N_Ω is less than or equal to the number of subdomains N_D, and each KL region can be expressed as a union of one or more disjoint subdomains. This approach allows for utilizing more processors than the physically dependent number of KL regions.

We propose and analyze three algorithms that combine stochastic collocation and the

MMMFEM with varying degrees of the Multiscale Flux Basis Implementation. The first collocation algorithm uses the MMMFEM with its traditional implementation, requiring solving one subdomain problem per interface iteration, on every stochastic realization. The second collocation algorithm is a naive (but more efficient) approach which forms a *deterministic* multiscale basis to solve the MMMFEM on each stochastic realization. These bases are then discarded and then re-computed with new permeability data for each subsequent realization. The third collocation algorithm is a smarter approach that forms a *stochastic* multiscale basis across all realizations, containing all the necessary information to perform the collocation before it begins. With extra "book-keeping" in the nonstationary case, we can take advantage of the repeated local structure of the permeability realizations in both tensor and sparse grids.

The resulting collocation algorithms are more computationally efficient than the traditional implementation by orders of magnitude. By limiting the number of linear systems we have to solve via the computation of deterministic or stochastic multiscale bases, we demonstrate that we can lessen the burden of the curse of dimensionality in the stochastic collocation method. Some of the examples show how *a posteriori* error estimation and adaptivity for the MMMFEM can be employed in stochastic multiscale simulations. We also present numerical convergence studies that confirm the theoretical *a priori* error estimates.

4.1 STATISTICALLY NON-STATIONARY RANDOM POROUS MEDIA

Consider the single-phase incompressible model with stochastic permeability $K = \exp(Y)$ as given in system (2.7)-(2.10). Following [60], let D be a union of disjoint KL regions, $\overline{D} = \cup_{i=1}^{N_\Omega} \overline{D}_{KL}^{(i)}$. Strictly within each KL region, the porous medium is statistically stationary, meaning covariance between any two points depends only on their distance and not on their location. The covariance between any two points from different regions is zero. Therefore the medium is globally nonstationary. As a result the probability space Ω is a product of

N_Ω spaces $\Omega^{(i)}$. For each event $\omega \in \Omega$,

$$\omega = (\omega^{(1)}, \ldots, \omega^{(N_\Omega)}) \quad \text{and} \quad Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} Y^{(i)}(\mathbf{x}, \omega^{(i)}), \tag{4.1}$$

where $Y^{(i)}(\mathbf{x}, \omega^{(i)})$ has physical support in $D_{KL}^{(i)}$.

Each $Y^{(i)}$ in (4.1) is assumed have a separate covariance function $C_{Y^{(i)}}$ of type (2.1) for its KL region $D_{KL}^{(i)}$. Following the method discussed in Section 2.1, the eigenvalues and eigenfunctions can be computed for its KL expansion, with a slight modification: Each covariance function is first restricted to a (rectangular) subdomain in the KL region, and after computation of the eigenfunctions on that sumdomain, they joined with neighboring subdomains in the KL Region. In this way, Fredholm equations of type (1.21) are always solved analytically on one rectangular region at a time. The KL expansion for the log permeability can now be written as the sum

$$Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} \sum_{j=1}^{\infty} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}). \tag{4.2}$$

At this point we admit the Finite Dimensional Noise Assumption 2.1.1 so that each KL expansion $Y^{(i)}$ is truncated after $N_{\text{term}}(i)$ terms. In this way (4.2) becomes

$$Y'(\mathbf{x}, \omega) \approx \sum_{i=1}^{N_\Omega} \sum_{j=1}^{N_{\text{term}}(i)} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}). \tag{4.3}$$

Globally, this means that we have $N_{\text{term}} = \sum N_{\text{term}}(i)$ terms in Y'. A low number of terms leads to a smooth permeability in a KL region. Therefore to model very heterogeneous noise in a KL region, $N_{\text{term}}(i)$ should be increased. The images of the random variables $\mathbb{S}_j^{(i)} = \xi_j^{(i)}(\Omega^{(i)})$ make up the finite dimensional vector spaces

$$\mathbb{S}^{(i)} = \prod_{j=1}^{N_{\text{term}}(i)} \mathbb{S}_j^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}(i)} \quad \text{and} \quad \mathbb{S} = \prod_{i=1}^{N_\Omega} \mathbb{S}^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}},$$

which are local to each KL region and globally, respectively.

To simplify notation, we shall introduce a function κ that provides a natural ordering for the global number of stochastic dimensions. Let the j-th stochastic parameter of the i-th KL region have a global index in $\{1, \ldots, N_{\text{term}}\}$ by the function

$$\kappa(i,j) = \begin{cases} j, & \text{if } i = 1 \\ j + \sum_{k=1}^{i-1} N_{\text{term}}(k), & \text{if } i > 1. \end{cases}$$

For example, the random vector $\xi = \left(\xi_j^{(i)}\right)_{1 \le \kappa(i,j) \le N_{\text{term}}} = \left(\xi_j^{(i)}\right)_\kappa$ is by definition equal to

$$\left(\underbrace{\xi_1^{(1)}, \ldots, \xi_{N_{\text{term}}(1)}^{(1)}}_{\text{KL region 1}}, \underbrace{\xi_1^{(2)}, \ldots, \xi_{N_{\text{term}}(2)}^{(2)}}_{\text{KL region 2}}, \ \ldots, \ \xi_j^{(i)}, \ \ldots, \ \underbrace{\xi_1^{(N_\Omega)}, \ldots, \xi_{N_{\text{term}}(N_\Omega)}^{(N_\Omega)}}_{\text{KL region } N_\Omega} \right).$$

If $\rho_j^{(i)}$ is the PDF of each $\xi_j^{(i)}$, then joint PDF for ξ is defined to be $\rho = \prod_i \prod_j \rho_j^{(i)}$. Then we can write $Y(\mathbf{x}, \omega) \approx Y(\mathbf{x}, \mathbf{y})$, where $\mathbf{y} = \left(\xi_j^{(i)}(\omega^{(i)})\right)_\kappa$.

As was done in Chapter 2, we once again abuse notation by replacing $K(\mathbf{x}, \omega)$ with its finite dimensional spectral approximation $K(\mathbf{x}, \mathbf{y})$ given by equation (4.3). Furthermore we also identify each KL region $\Omega^{(i)}$ with its parameterization $\mathbb{S}^{(i)}$. Therefore the modeling error between the true stochastic solution and its finite dimensional approximation $\|\mathbf{u}(\mathbf{x}, \omega) - \mathbf{u}(\mathbf{x}, \mathbf{y})\|$ is neglected.

4.2 FORMULATION OF STOCHATIC MORTAR MIXED METHOD

Recall the domain decomposition notation from Section 3.1 with subdomains D_i and interfaces $\Gamma_{i,j}$. We make the extra restriction that the subdomains must conform to the KL regions, meaning each KL region is simply a union of subdomains. In domain decomposition variational formulation for the stochastic single-phase model, system (3.3)-(3.8) holds for ρ-*a.e.* $\mathbf{y} \in \mathbb{S}$. Recall the deterministic Sobolev spaces for subdomains $i = 1, \ldots, N_D$ from definition (3.9) and globally from definition (3.10). We denote them in this chapter by $W_i(D_i) = W_i$, $\mathbf{V}_i^\gamma(D) = \mathbf{V}_i^\gamma$, $W(D) = W$, and $\mathbf{V}^\gamma(D) = \mathbf{V}^\gamma$. Since are reusing the

notation from Chapters 2 and 3, we must stress that the global velocity space $\mathbf{V}^\gamma(D)$ is not the same as (2.11) because normal velocities are not continuous across subdomain interfaces Γ. Therefore we must also recall the deterministic mortar space (3.11) and denote this by $M(\Gamma) = M$.

Since our goal once again is to compute statistical moments, recall the definition of the space $L^2_\rho(\mathbb{S})$ from (2.12) and take its tensor product with the deterministic domain decomposition spaces to form the stochastic spaces

$$W(D,\mathbb{S}) = W(D) \times L^2_\rho(\mathbb{S}), \quad \mathbf{V}^\gamma(D,\mathbb{S}) = \mathbf{V}^\gamma(D) \times L^2_\rho(\mathbb{S}), \quad M(\Gamma,\mathbb{S}) = M(\Gamma) \times L^2_\rho(\mathbb{S}).$$

Once again, when the explicit dependence in parentheses is omitted, it is implied that we mean the stochastic spaces, *e.g.* $W = W(D,\mathbb{S})$. These spaces are equipped with the same norms as given in (2.13)-(2.14).

The stochastic domain decomposition variational formulation is to find $\mathbf{u} \in \mathbf{V}^{g_N}$, $p \in W$, and $\lambda \in M$ such that for $i = 1, \ldots, N_D$,

$$\begin{aligned}
\int_{\mathbb{S}} (K^{-1}\mathbf{u}, \mathbf{v})_{D_i} \rho(\mathbf{y}) d\mathbf{y} &= \int_{\mathbb{S}} \Big[(p, \nabla \cdot \mathbf{v})_{D_i} - \langle \mathbf{v} \cdot \mathbf{n}_i, \lambda \rangle_{\Gamma_i} \\
&\qquad - \langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \forall \mathbf{v} \in \mathbf{V}^0_i, \qquad (4.4) \\
\int_{\mathbb{S}} (\nabla \cdot \mathbf{u}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} &= \int_{\mathbb{S}} (f, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} \qquad \forall w \in W_i, \qquad (4.5) \\
\int_{\mathbb{S}} \sum_{i=1}^{N_D} \langle \mathbf{u}_i \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} \rho(\mathbf{y}) d\mathbf{y} &= 0 \qquad \forall \mu \in M. \qquad (4.6)
\end{aligned}$$

The discretization of this system is first performed in the spatial dimensions. Each subdomain and interface are independently partitioned into mixed finite element and mortar finite element spaces, as described in Section 3.1. The coarseness of the mortar space is carefully chosen to satisfy Assumption 3.1.1. This gives us the semidiscrete stochastic multiscale

mortar mixed finite element approximation: find $\mathbf{u}_h : \mathbb{S} \to \mathbf{V}_h^{g_N}(D)$, $p_h : \mathbb{S} \to W_h(D)$, and $\lambda_H : \mathbb{S} \to M_H(\Gamma)$ such that for $i = 1, \ldots, N_D$ and ρ-*a.e.* $\mathbf{y} \in \mathbb{S}$,

$$\begin{aligned}
(K^{-1}\mathbf{u}_h, \mathbf{v})_{D_i} &= (p_h, \nabla \cdot \mathbf{v})_{D_i} - \langle \mathbf{v} \cdot \mathbf{n}_i, \lambda_H \rangle_{\Gamma_i} \\
&\quad - \langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0(D), \qquad (4.7) \\
(\nabla \cdot \mathbf{u}_h, w)_{D_i} &= (f, w)_{D_i} \qquad \forall w \in W_{h,i}(D), \qquad (4.8) \\
\sum_{i=1}^{N_D} \langle \mathbf{u}_{h,i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} &= 0 \qquad \forall \mu \in M_H(D). \qquad (4.9)
\end{aligned}$$

4.2.1 Tensor Product and Sparse Grid Collocation

Recall the material on tensor product grid collocation in Section 2.2.1. We choose $N_{\text{coll}}(i,j)$ collocation points in stochastic dimension j of KL Region i, so that $\mathbf{m} = (N_{\text{coll}}(i,j))_\kappa$ is the N_{term}-dimensional multi-index indicating the desired component degree of the interpolant in the stochastic space $\mathbb{S}$. The formula for the corresponding anisotropic tensor product Gauss-Hermite interpolant (2.27) is the same. We make the following two remarks about how things change in the case of non-stationary random porous media with KL regions, which will be important to our collocation algorithms in Section 4.4.

First, with multiple KL Regions, the set of abscissae for a tensor product grid (2.28) becomes

$$\mathcal{T}(\mathbf{m}) = \bigotimes_{i=1}^{N_\Omega} \left(\bigotimes_{j=1}^{N_{\text{term}}(i)} \mathcal{H}(N_{\text{coll}}(i,j)) \right) \qquad (4.10)$$

In other words, the global N_{term}-dimensional tensor grid is the tensor product of N_Ω smaller tensor product grids of dimension $N_{\text{term}}(i)$. Therefore, the number of permeability realizations local to the KL region i and global to the entire domain are:

$$N_{\text{real}}(i) = \prod_{j=1}^{N_{\text{term}}(i)} N_{\text{coll}}(i,j) \quad \text{and} \quad N_{\text{real}} = \prod_{i=1}^{N_\Omega} N_{\text{real}}(i), \quad \text{respectively.} \qquad (4.11)$$

Moreover, in the case of isotropic tensor product collocation where each stochastic dimension $\mathbb{S}_j^{(i)}$ has the same polynomial accuracy $\mathbf{m} = (m, m, \ldots, m)$, the tensor grid points are

$$\mathcal{T}(\mathbf{m}) = \bigotimes_{k=1}^{N_{\text{term}}} \mathcal{H}(m).$$

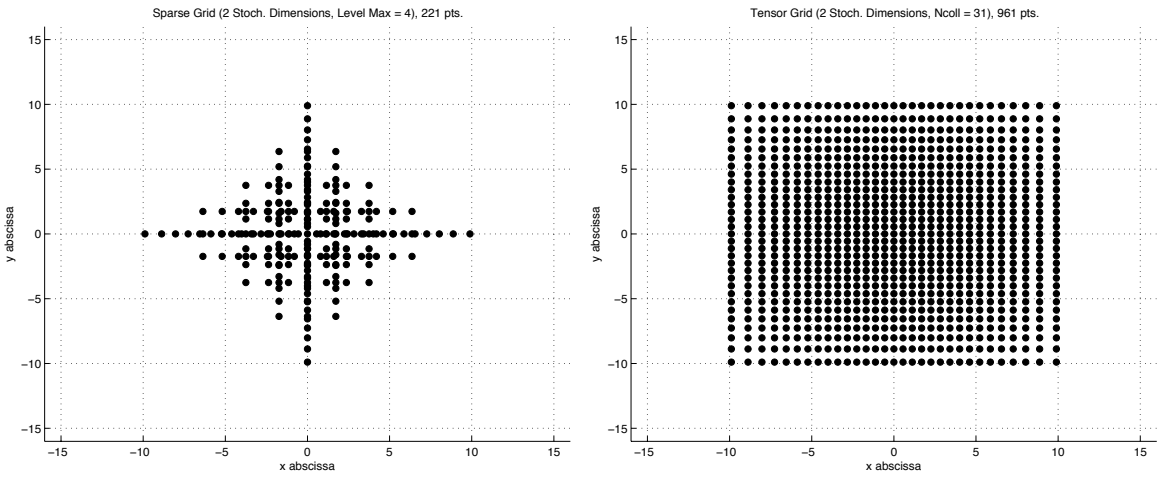
and the number of realizations reduces to $N_{\text{real}}(i) = m^{N_{\text{term}}(i)}$ and $N_{\text{real}} = m^{N_{\text{term}}}$.

Second, we choose to index the tensor product collocation points with a natural ordering. For the tensor grid point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$, its global collocation index $k \in \{1, \ldots, N_{\text{real}}\}$ is given by

$$k = k_1 + \sum_{i=2}^{N_{\text{term}}} k_i \prod_{j=1}^{i-1} \mathbf{m}(k). \tag{4.12}$$

Next we introduce the idea of a sparse grid. They were first used for high dimensional quadrature by Smolyak in 1963 [79] and have been applied to stochastic collocation in such works as [83, 30, 61]. In sparse grid collocation, the polynomial accuracy is prescribed in terms of *total* degree. We only consider isotropic sparse grids, meaning for polynomials up to total degree m, we exclude monomials such as $y_1^{\alpha_1} y_2^{\alpha_2}$ where $\alpha_1 + \alpha_2 > m$. Sparse grids rules are known to have the same asymptotic accuracy as tensor product rules, while requiring far fewer points as the dimension increases. This property is essential for coping with the curse of dimensionality. Therefore sparse grids are applicable for problems with higher dimensional noise, *i.e.* up to several hundred stochastic dimensions. A picture of comparable sparse grid and tensor grid rules is shown in Figure 4.1.

Figure 4.1: A Gauss-Hermite sparse grid (left) versus a Gauss-Hermite tensor grid (right) with a comparable number of points on each axis.



Sparse grid rules are linear combinations of tensor products on a family of nested one dimensional rules. They are constructed hierarchically to have the property that the total

polynomial degree is a constant independent of dimension. They are described in terms of a level $\ell_{\max}$, where the N_{term}-dimensional sparse grid quadrature rule of level $\ell_{\max}$ is accurate to degree $(2 \cdot \ell_{\max} + 1)$. We consider isotropic sparse grids but note they may be generalized to anisotropic accuracy, *cf.* [63].

Each level between $\ell_{\max}$ and $\ell_{\min} = \max\{0, \ell_{\max} - N_{\text{term}} + 1\}$ is an integer partitioned into N_{term} non-negative parts. These partitions form multi-indices $\mathbf{p} = (p_1, \ldots, p_{N_{\text{term}}})$, $|\mathbf{p}| = \sum p_i$, denoting the levels of one dimensional rules to use for each stochastic dimension. In this work, the one dimensional abscissae of level p_i are the Gauss-Hermite points $\mathcal{H}(2^{p_i+1} - 1)$. Level 0 starts with a single point, and the number of points doubles plus one on each subsequent level.

Let the multi-index $\mathbf{m} = 2^{\mathbf{p}+\mathbb{1}} - \mathbb{1}$ denote degree for each partition $\mathbf{p}$. The corresponding isotropic sparse grid Gauss-Hermite interpolant in N_{term}-dimensions is defined by

$$\mathcal{I}^{\text{SG}}_{\ell_{\max}} f(\mathbf{y}) = \sum_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} (-1)^{\ell_{\max} - |\mathbf{p}|} \cdot \binom{N_{\text{term}} - 1}{\ell_{\max} - |\mathbf{p}|} \cdot \mathcal{I}^{\text{TG}}_{\mathbf{m}} f(\mathbf{y}). \tag{4.13}$$

The set of abscissae for this rule is

$$\mathcal{S}(\ell_{\min}, \ell_{\max}, N_{\text{term}}) = \bigcup_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} \bigotimes_{i=1}^{N_{\text{term}}} \mathcal{H}(2^{p_i+1} - 1). \tag{4.14}$$

We remark that the number of local permeability realizations on a sparse grid for a particular KL region is given by the relationship

$$\mathcal{S}(\ell_{\min}, \ell_{\max}, N_{\text{term}}) \subsetneq \underbrace{\mathcal{S}(0, \ell_{\max}, N_{\text{term}}(1))}_{\text{Projection into } \mathbb{S}^{(1)}} \otimes \cdots \otimes \underbrace{\mathcal{S}(0, \ell_{\max}, N_{\text{term}}(N_\Omega))}_{\text{Projection into } \mathbb{S}^{(N_\Omega)}}. \tag{4.15}$$

Note that $\ell_{\min} = 0$ for all the local sparse grids, unlike the global sparse grid.

The points in (4.14) are weakly nested because the origin is the sole value which is repeated in each one dimensional rule. Taking tensor products of one dimensional rules produces many repeated points that contain the origin in one or more of its components. There are both pros and cons to skipping these repeated abscissae. On the one hand, fewer function evaluations in the quadrature rule means fewer realizations to solve. On the other hand, extra book-keeping is necessary for indexing the points and calculating their collocation

weights. In Algorithm 4.1, we give an efficient method that provides a natural ordering for the points in a Gauss-Hermite sparse grid, which skips repeated points.

Algorithm 4.1 A natural ordering for sparse grid points.

input: Global Index g

$j \leftarrow 0$

for $\ell = \ell_{\min}, \ldots, \ell_{\max}$ **do** {Loop over levels}

  for $i = 1, \ldots, \dfrac{(\ell + N_{\text{term}} - 1)!}{\ell!(N_{\text{term}} - 1)!}$ **do** {Loop over partitions}

    $part \leftarrow (p_1, \ldots, p_{N_{\text{term}}})$, $\mathbf{m} \leftarrow 2^{part+\mathbb{1}} - \mathbb{1}$ {The i-th multi-index}

    if $\ell = \ell_{\min}$ **then** add $part$ to $PartList$

    for $k = 1, \ldots, \prod_\alpha \mathbf{m}(\alpha)$ **do** {Loop over points}

      $point \leftarrow (h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$ {The j-th point using (4.12)}

      $\tilde{part} \leftarrow (\tilde{p}_1, \ldots, \tilde{p}_{N_{\text{term}}})$ where $\tilde{p}_i = \begin{cases} p_i, & \text{if } h^{k_i}_{\mathbf{m}(i)} \neq 0 \\ 0, & \text{if } h^{k_i}_{\mathbf{m}(i)} = 0 \end{cases}$

      if ($\ell = \ell_{\min}$ and $\tilde{part} \in PartList$) **then** {Repeated point; skip it}

      else if ($\ell > \ell_{\min}$ and $\tilde{part} \neq part$) **then** {Repeated point; skip it}

      else $j = j + 1$ {Unique point; count it}

      if ($j = g$) **then return** $point, part$

    end for

  end for

end for

Suppose that a sparse grid point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$ occurs in a set of partitions $\mathcal{P}$. If it is used in a single function evaluation with subsequent occurrences skipped by Algorithm 4.1, then its quadrature weight must be calculated by the formula

$$\sum_{\mathbf{p} \in \mathcal{P}} (-1)^{\ell_{\max} - |\mathbf{p}|} \cdot \binom{N_{\text{term}} - 1}{\ell_{\max} - |\mathbf{p}|} \prod_{i=1}^{N_{\text{term}}} w^{k_i}_{\mathbf{m}(i)}.$$

Note that in a sparse grid rule, quadrature weights may become negative.

4.3 ERROR ANALYSIS

In this section we present *a priori* error estimates for the solution to the stochastic MMM-FEM. As in previous stochastic collocation works, see *e.g.* [11, 89, 64, 36], the error is decomposed into deterministic and stochastic errors, see Theorem 4.3.1. Furthermore, we employ a duality argument to show superconvergence for the pressure, see Theorem 4.3.2.

Note that throughout this entire section, we tacitly assume that Assumption 3.1.1 holds. To avoid technical details for the approximation of the Neumann boundary condition, we further assume that $g_N \in \mathbf{V}_h(D) \cdot \mathbf{n}$.

We start with some definitions. We define the space of weakly continuous velocities by

$$\mathbf{V}_{h,0}^{\gamma}(D) = \left\{ \mathbf{v} \in \mathbf{V}_h^{\gamma}(D) \mid \sum_{i=1}^{N_D} \langle \mathbf{v}|_{D_i} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i} = 0 \quad \forall \mu \in M_H(D) \quad \text{for } \rho\text{-a.e. in } \mathbb{S} \right\}.$$

Recall that for any of the standard mixed spaces, $\nabla \cdot \mathbf{V}_{h,i}(D) = W_{h,i}(D)$. Let, for any $\varepsilon > 0$, $\Pi_i : (H^{\varepsilon}(D_i))^d \cap \mathbf{V}_i^{\gamma} \to \mathbf{V}_{h,i}^{\gamma}$, $\Pi \mathbf{q}|_{D_i} = \Pi_i \mathbf{q}$ be the standard MFE projection operators. A projection operator $\Pi_0 : (H^{1/2+\varepsilon}(D))^d \cap \mathbf{V}(D) \to \mathbf{V}_{h,0}(D)$ is defined in [8, 9], satisfying

$$(\nabla \cdot (\Pi_0 \mathbf{q} - \mathbf{q}), w)_{D_i} = 0, \quad \forall\, w \in W_{h,i}(D_i),$$

$$\|\Pi_0 \mathbf{q} - \Pi \mathbf{q}\| \leq C \sum_{i=1}^{N_D} \|\mathbf{q}\|_{r+1/2,D_i} h^r H^{1/2}, \quad 0 \leq r \leq k+1 \tag{4.16}$$

$$\|\Pi_0 \mathbf{q} - \mathbf{q}\| \leq C \sum_{i=1}^{N_D} \|\mathbf{q}\|_{r,D_i} h^{r-1/2} H^{1/2}, \quad 1 \leq r \leq k+1\,. \tag{4.17}$$

$$\left(\sum_{i=1}^{N_D} \|\Pi_0 \mathbf{q}\|_{H(div,D_i)}^2 \right)^{1/2} \leq C \sum_{i=1}^{N_D} \|\mathbf{q}\|_{1,D_i}. \tag{4.18}$$

Note that (4.18) is not explicitly stated in [8, 9], but follows easily from the results there.

For any $\varphi \in L^2(D)$, define its $L^2(D)$-projection $\hat{\varphi}$ onto $W_h(D)$ by

$$(\varphi - \hat{\varphi}, w) = 0, \quad \forall w \in W_h(D).$$

Similarly, let $\mathcal{P}_H$ denote the $L^2(\Gamma)$-projection onto $M_H(\Gamma)$. Let $\mathcal{I}_H^c$ be the nodal interpolant operator into the space $M_H^c(\Gamma)$ which is the subset of continuous functions in $M_H(\Gamma)$, where

we use the Scott-Zhang operator [78] to define the nodal values of ψ if it doesn't have poinwise values. We will make use of the following inequalities:

$$\|\psi - \mathcal{I}_H^c \psi\|_{t,\Gamma_i} \leq C\|\psi\|_{s,\Gamma_i} H^{s-t}, \qquad 0 \leq s \leq r+1, \quad 0 \leq t \leq 1, \tag{4.19}$$

$$\|\psi - \mathcal{P}_H \psi\|_{-t,\Gamma_i} \leq C\|\psi\|_{s,\Gamma_i} H^{s+t}, \qquad 0 \leq s \leq r+1, \quad 0 \leq t \leq 1, \tag{4.20}$$

$$\|\varphi - \hat{\varphi}\| \leq C\|\varphi\|_t h^t, \qquad 0 \leq t \leq l+1, \tag{4.21}$$

$$\|q\|_{t,\Gamma_i} \leq C\|q\|_{t+1/2,D_i}, \qquad 0 < t, \tag{4.22}$$

$$\|\mathbf{v}\cdot\mathbf{n}\|_{\Gamma_i} \leq Ch^{-1/2}\|\mathbf{v}\|_{D_i} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}(D_i), \tag{4.23}$$

$$\langle \psi, \mathbf{q}\cdot\mathbf{n}\rangle_{\Gamma_i} \leq C\|\psi\|_{1/2,\Gamma_i}\|\mathbf{q}\|_{H(div;D_i)}, \tag{4.24}$$

$$\|(\mathbf{q} - \Pi_i \mathbf{q})\cdot\mathbf{n}_i\|_{-t,\Gamma_i} \leq C\|\mathbf{q}\|_{s,\Gamma_i} h^{s+t}, \qquad 0 \leq s \leq k+1, \quad 0 \leq t \leq k+1, \tag{4.25}$$

where $\|\cdot\|_{-t}$ is the norm of H^{-t}, the dual of H^t (not H_0^t). Bound (4.19) is found in [78], the L^2-projection approximations (4.20), (4.21), and (4.25) are found in [23], the nonstandard trace theorem (4.22) is found in [43], the trace inequality (4.23) is found in [8], and the bound (4.24) follows from the normal trace inequality for $H(div; D_i)$-functions.

We eliminate λ_H from the semi-discrete formulation by restricting $\mathbf{V}_h^{g_N}(D)$ to $\mathbf{V}_{h,0}^{g_N}(D)$. In other words, (4.7)–(4.9) is equivalent to finding $\mathbf{u}_h : \mathbb{S} \to \mathbf{V}_{h,0}^{g_N}$ and $p_h : \mathbb{S} \to W_h$ such that for ρ-a.e. $\mathbf{y} \in \mathbb{S}$,

$$\sum_{i=1}^{N_D} (K^{-1}\mathbf{u}_h, \mathbf{v})_{D_i} = \sum_{i=1}^{N_D} (p_h, \nabla\cdot\mathbf{v})_{D_i} - \langle \mathbf{v}\cdot\mathbf{n}_i, g_D\rangle_{\Gamma_D} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,0}^0(D), \tag{4.26}$$

$$\sum_{i=1}^{N_D} (\nabla\cdot\mathbf{u}_h, w)_{D_i} = \sum_{i=1}^{N_D} (f, w)_{D_i} \qquad \forall w \in W_h(D). \tag{4.27}$$

We form error equations by integrating system (4.26) in $\mathbb{S}$ against the PDF, and subtracting it from system (4.4)–(4.6).

$$\int_{\mathbb{S}} \sum_{i=1}^{N_D} (K^{-1}(\mathbf{u} - \mathbf{u}_h), \mathbf{v})_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \Big[\sum_{i=1}^{N_D} (\hat{p} - p_h, \nabla\cdot\mathbf{v})_{D_i} - \langle p, \mathbf{v}\cdot\mathbf{n}_i\rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,0}^0, \tag{4.28}$$

$$\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot(\mathbf{u} - \mathbf{u}_h), w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} = 0 \qquad \forall w \in W_h. \tag{4.29}$$

Recall that k, l, r, m denote the polynomial degrees of approximation for the velocity space, pressure space, mortar space, and collocation interpolant, respectively. In all mixed methods, $l = k$ or $l = k-1$. The first results follows from the deterministic multiscale bound on velocity, which is proved in Theorem 4.1 in [9].

Lemma 4.3.1. *There exists a positive constant C independent of h and H such that for all $0 \le q \le l+1$, $1 \le t \le k+1$, and $0 < s \le r+1$,*

$$\begin{aligned}
\|\mathbf{u} - \mathbf{u}_h\|_{L^2(D)\otimes L^2(\mathbb{S})} &\le C(\|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2} + \|\mathbf{u}\|_{H^t(D)\otimes L^2(\mathbb{S})} h^t \\
&\quad + \|\mathbf{u}\|_{H^{t+1/2}(D)\otimes L^2(\mathbb{S})} h^t H^{1/2}), \\
\|\nabla\cdot(\mathbf{u} - \mathbf{u}_h)\|_{L^2(D_i)\otimes L^2(\mathbb{S})} &\le C\|\nabla\cdot\mathbf{u}\|_{H^q(D_i)\otimes L^2(\mathbb{S})} h^q, \quad 1 \le i \le N_D.
\end{aligned}$$

For the pressure bound we need the following inf-sup condition.

Lemma 4.3.2. *There exists a positive constant γ independent of h and H such that for all $w : \mathbb{S} \to W_h(D)$,*

$$\sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}_{h,0}^0(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \ge \gamma \|w\|_W.$$

Proof. Let $w : \mathbb{S} \to W_h(D)$. Consider the auxiliary problem, for ρ-almost every $\mathbf{y} \in \mathbb{S}$

$$\nabla\cdot\boldsymbol{\psi}(\cdot,\mathbf{y}) = w(\cdot,\mathbf{y}) \ \text{ in } D, \quad \boldsymbol{\psi}(\cdot,\mathbf{y}) = \mathbf{g}(\cdot,\mathbf{y}) \ \text{ on } \partial D,$$

where $\mathbf{g} \in (H^{1/2}(\partial D))^d$ is constructed to satisfy $\int_{\partial D} \mathbf{g}\cdot\mathbf{n} = \int_D w$ and $\mathbf{g}\cdot\mathbf{n} = 0$ on Γ_N. More precisely, we take $\mathbf{g} = \left(\int_D w\right)\varphi\mathbf{n}$, where $\varphi \in C^0(\partial D)$ is such that $\int_{\partial D}\varphi = 1$ and $\varphi = 0$ on Γ_N. Clearly $\|\mathbf{g}\|_{1/2,\partial D} \le C\|w\|$. It is known [34] that the above problem has a solution satisfying

$$\|\boldsymbol{\psi}\|_1 \le C(\|w\| + \|\mathbf{g}\|_{1/2,\partial D}) \le C\|w\|. \tag{4.30}$$

Then

$$\sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}_{h,0}^0(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \ge \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\Pi_0\boldsymbol{\psi}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\Pi_0\boldsymbol{\psi}\|_V} \ge \gamma\|w\|_W,$$

where we have used (4.3), (4.18), and (4.30) for the last inequality. □

From Lemma 4.3.2, we can derive a multiscale bound on the semi-discrete pressure.

Lemma 4.3.3. *There exists a positive constant C independent of h and H such that for $1 \le t \le k+1$, and $0 < s \le r+1$,*

$$\|p - p_h\|_W \le C(\|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2} + \|\mathbf{u}\|_{H^t(D)\otimes L^2(\mathbb{S})} h^t + \|\mathbf{u}\|_{H^{t+1/2}(D)\otimes L^2(\mathbb{S})} h^t H^{1/2}).$$

Proof. Taking $w = \hat{p} - p_h$ in Lemma 4.3.2 and using (4.28) gives

$$\begin{aligned}
\|\hat{p} - p_h\|_W &\le \frac{1}{\gamma} \sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, \hat{p} - p_h)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \\
&= \frac{1}{\gamma} \sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} \left[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \mathbf{v})_{D_i} + \langle p - \mathcal{I}^c_H p, \mathbf{v}\cdot\mathbf{n}_i\rangle_{\Gamma_i}\right] \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \\
&\le C \left(\|\mathbf{u} - \mathbf{u}_h\|_{L^2(D)\otimes L^2(\mathbb{S})} + \sum_{i=1}^{N_D} H^{s-1/2} \|p\|_{H^s(\Gamma_i)\otimes L^2(\mathbb{S})} \right) \\
&\le C(\|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2} + \|\mathbf{u}\|_{H^t(D)\otimes L^2(\mathbb{S})} h^t + \|\mathbf{u}\|_{H^{t+1/2}(D)\otimes L^2(\mathbb{S})} h^t H^{1/2}),
\end{aligned}$$

where we have used (4.19), (4.22), and Lemma 4.3.1 in the last two inequalities. The proof is completed using the triangle inequality and (4.21). □

Theorem 4.3.1. *Assume that the solution $(\mathbf{u}, p)$ to (2.7)–(2.10) is sufficiently smooth, so that the norms that appear in Lemma 4.3.1 are well defined. Then there exists a positive constant C independent of h and H such that for $0 \le q \le l+1$, $1 \le t \le k+1$, and $0 < s \le r+1$,*

$$\|\mathbf{u} - \mathbf{u}_{h,m}\|_V + \|p - p_{h,m}\|_W \le C(H^{s-1/2} + h^q + h^t + h^t H^{1/2}) + \eta. \tag{4.31}$$

For tensor product grid collocation,

$$\eta \le C \sum_{i=1}^{N_\Omega} \exp\left(-c_i \sqrt{m_i}\right), \tag{4.32}$$

For sparse grid collocation,

$$\eta \le C(\sigma) \begin{cases} \exp(-\sigma N_{term} 2^{\ell_{max}/N_{term}}), & \text{for large } \ell_{max} \\ \exp(-\sigma e \log_2(\ell_{max})), & \text{for large } N_{term}\ . \end{cases} \tag{4.33}$$

In the above c_i and σ are positive constants that depend on the smoothness of K in $\mathbb{S}$.

Proof. The left hand side of (4.31) is decomposed into deterministic and stochastic errors with the triangle inequality by adding and subtracting the semidiscrete solution

$$\begin{aligned}\|\mathbf{u}-\mathbf{u}_{h,m}\|_V &+ \|p-p_{h,m}\|_W \\ &\leq (\|\mathbf{u}-\mathbf{u}_h\|_V + \|p-p_h\|_W) + (\|\mathbf{u}_h - \mathcal{I}_m\mathbf{u}_h\|_V + \|p_h - \mathcal{I}_m p_h\|_W) \\ &= (\|\mathbf{u}-\mathbf{u}_h\|_V + \|p-p_h\|_W) + \eta.\end{aligned}$$

Assuming K is smooth enough in $\mathbb{S}$, which is true for the KL expansion, the estimate of the stochastic error η in the case of tensor product grid collocation (4.32) can be found in [11, Theorem 4.1], and in the case of sparse grid collocation (4.33) can be found in [64, Theorem 3.18].

The remaining estimate of the deterministic error follows by integrating the equations in Lemmas 4.3.1 and 4.3.3 against the PDF $\rho(\mathbf{y})$ in $\mathbb{S}$.

$$\begin{aligned}\|\mathbf{u}-\mathbf{u}_h\|_V^2 &+ \|p-p_h\|_W^2 = \int_{\mathbb{S}} \left(\|\mathbf{u}-\mathbf{u}_h\|_{H(\mathrm{div};D)}^2 + \|p-p_h\|^2\right)\rho(\mathbf{y})d\mathbf{y} \\ &\leq C\int_{\mathbb{S}}\left[\left(\sum_{i=1}^{N_D}\left(\|p\|_{s+1/2,D_i}H^{s-1/2} + \|\mathbf{u}\|_{t,D_i}h^t + \|\mathbf{u}\|_{t+1/2,D_i}h^t H^{1/2}\right)\right)^2\right. \\ &\qquad \left. + \left(\sum_{i=1}^{N_D}\|\nabla\cdot\mathbf{u}\|_{q,D_i}h^q\right)^2\right]\rho(\mathbf{y})d\mathbf{y} \\ &\leq C\int_{\mathbb{S}}\left[\left(\sum_{i=1}^{N_D}\|p\|_{s+1/2,D_i}\right)^2 H^{2s-1} + \left(\sum_{i=1}^{N_D}\|\mathbf{u}\|_{t,D_i}\right)^2 h^{2t}\right. \\ &\qquad \left. + \left(\sum_{i=1}^{N_D}\|\mathbf{u}\|_{t+1/2,D_i}\right)^2 h^{2t}H + \left(\sum_{i=1}^{N_D}\|\nabla\cdot\mathbf{u}\|_{q,D_i}\right)^2 h^{2q}\right]\rho(\mathbf{y})d\mathbf{y}\end{aligned}$$

□

In the next theorem we establish superconvergence for the pressure.

Theorem 4.3.2. *Assume that the problem (2.7)–(2.10) is H^2-elliptic regular. Under the assumptions of Theorem 4.3.1, there exists a positive constant C independent of h and H such that for $0 \leq q \leq l+1$, $1 \leq t \leq k+1$, and $0 < s \leq r+1$,*

$$\|\hat{p} - p_{h,m}\|_W \leq C(H^{s+1/2} + h^q H + h^t H + h^t H^{3/2}) + \eta, \tag{4.34}$$

where η is defined in Theorem 4.3.1.

Proof. Consider the following auxiliary problem in mixed form. For ρ-a.e. $\mathbf{y} \in \mathbb{S}$,

$$\boldsymbol{\psi}(\cdot, \mathbf{y}) = -K(\cdot, \mathbf{y}) \nabla \varphi(\cdot, \mathbf{y}) \qquad \text{in } D, \tag{4.35}$$

$$\nabla \cdot \boldsymbol{\psi}(\cdot, \mathbf{y}) = \hat{p} - p_{h,m} \qquad \text{in } D, \tag{4.36}$$

$$\varphi(\cdot, \mathbf{y}) = 0 \qquad \text{on } \Gamma_D \tag{4.37}$$

$$\boldsymbol{\psi}(\cdot, \mathbf{y}) \cdot \mathbf{n} = 0 \qquad \text{on } \Gamma_N \ . \tag{4.38}$$

The H^2-elliptic regularity implies

$$\|\varphi(\cdot, \mathbf{y})\|_2 \leq C \|\hat{p} - p_{h,m}\|. \tag{4.39}$$

Therefore,

$$\begin{aligned} \|\hat{p} - p_{h,m}\|_W^2 &= \int_{\mathbb{S}} (\hat{p} - p_{h,m}, \hat{p} - p_{h,m}) \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \left[(\nabla \cdot \boldsymbol{\psi}, \hat{p} - p_h) + (\hat{p} - p_{h,m}, p_h - \mathcal{I}_m p_h) \right] \rho(\mathbf{y}) d\mathbf{y} \\ &= I + II. \end{aligned}$$

Applying the Cauchy-Schwarz inequality,

$$\begin{aligned} |II| &\leq \left(\int_{\mathbb{S}} \|\hat{p} - p_{h,m}\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|p_h - \mathcal{I}_m p_h\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &= \|\hat{p} - p_{h,m}\|_W \|p_h - \mathcal{I}_m p_h\|_W = \|\hat{p} - p_{h,m}\|_W \ \eta. \end{aligned}$$

Taking $\mathbf{v} = \Pi_0 \boldsymbol{\psi} \in \mathbf{V}_{h,0}^0(D)$ in (4.28), we have

$$\begin{aligned} I &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla \cdot \Pi_0 \boldsymbol{\psi}, \hat{p} - p_h)_{D_i} \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \left[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \Pi_0 \boldsymbol{\psi})_{D_i} + \langle p - \mathcal{P}_H p, \Pi_0 \boldsymbol{\psi} \cdot n_i \rangle_{\Gamma_i} \right] \rho(\mathbf{y}) d\mathbf{y} \qquad \text{by (4.29)} \\ &= I_1 + I_2. \end{aligned}$$

We can break up I_1 into three terms by

$$\begin{aligned} I_1 &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[(K^{-1}(\mathbf{u}-\mathbf{u}_h), \Pi_0\boldsymbol{\psi}-\boldsymbol{\psi})_{D_i} - (\mathbf{u}-\mathbf{u}_h, \nabla\varphi)_{D_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[(K^{-1}(\mathbf{u}-\mathbf{u}_h), \Pi_0\boldsymbol{\psi}-\boldsymbol{\psi})_{D_i} + (\nabla\cdot(\mathbf{u}-\mathbf{u}_h), \varphi-\hat{\varphi})_{D_i} \\ &\qquad -\langle (\mathbf{u}-\mathbf{u}_h)\cdot\mathbf{n}_i, \varphi - \mathcal{I}_H^c\varphi\rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\ &= I_{11} + I_{12} - I_{13}. \end{aligned}$$

Upper bounds for I_{11}, I_{12}, I_{13} can be obtained using the Cauchy-Schwarz inequality

$$\begin{aligned} I_{11} &= \int_{\mathbb{S}} \left[\sum_{i=1}^{N_D} (K^{-1}(\mathbf{u}-\mathbf{u}_h), \Pi_0\boldsymbol{\psi}-\boldsymbol{\psi})_{D_i} \right] \rho(\mathbf{y}) d\mathbf{y} \\ &\le C \sum_{i=1}^{N_D} \left(\int_{\mathbb{S}} \|\mathbf{u}-\mathbf{u}_h\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|\Pi_0\boldsymbol{\psi}-\boldsymbol{\psi}\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le C \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\mathbf{u}-\mathbf{u}_h\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\Pi_0\boldsymbol{\psi}-\boldsymbol{\psi}\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le C \|\mathbf{u}-\mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \|\Pi_0\boldsymbol{\psi}-\boldsymbol{\psi}\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le C\sqrt{hH}\ \|\mathbf{u}-\mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \|\boldsymbol{\psi}\|_1^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \qquad \text{by (4.17)} \\ &\le C\sqrt{hH}\ \|\mathbf{u}-\mathbf{u}_h\|_V\ \|\hat{p}-p_{h,m}\|_W \qquad \text{by (4.39), (4.35).} \end{aligned}$$

$$\begin{aligned} I_{12} &= \int_{\mathbb{S}} \left[\sum_{i=1}^{N_D} (\nabla\cdot(\mathbf{u}-\mathbf{u}_h), \varphi-\hat{\varphi})_{D_i} \right] \rho(\mathbf{y}) d\mathbf{y} \\ &\le C \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\nabla\cdot(\mathbf{u}-\mathbf{u}_h)\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|\varphi-\hat{\varphi}\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le Ch \|\mathbf{u}-\mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \|\varphi\|_1^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \qquad \text{by (4.21)} \\ &\le Ch \|\mathbf{u}-\mathbf{u}_h\|_V \|\hat{p}-p_{h,m}\|_W \qquad \text{by (4.39).} \end{aligned}$$

$$
\begin{aligned}
I_{13} &= \int_{\mathbb{S}} \left[\sum_{i=1}^{N_D} \langle (\mathbf{u}-\mathbf{u}_h)\cdot n_i, \varphi - \mathcal{I}_H^c \varphi \rangle_{\Gamma_i} \right] \rho(\mathbf{y}) d\mathbf{y} \\
&\leq C \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\mathbf{u}-\mathbf{u}_h\|^2_{H(div,D_i)} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\varphi - \mathcal{I}_H^c \varphi\|^2_{1/2,\Gamma_i} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \quad \text{by (4.24)} \\
&\leq CH \|\mathbf{u}-\mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\varphi\|^2_{3/2,\Gamma_i} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \quad \text{by (4.19)} \\
&\leq CH \|\mathbf{u}-\mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\varphi\|^2_{2,D_i} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \quad \text{by (4.22)} \\
&\leq CH \|\mathbf{u}-\mathbf{u}_h\|_V \|\hat{p} - p_{h,m}\|_W \quad \text{by (4.39).}
\end{aligned}
$$

Since $h \leq H$, it follows that $|I_1| \leq CH\|\mathbf{u}-\mathbf{u}_h\|_V \|\hat{p}-p_{h,m}\|_W$. We bound I_2 as follows:

$$
\begin{aligned}
I_2 &= \int_{\mathbb{S}} \left[\sum_{i=1}^{N_D} \langle p - \mathcal{P}_H p, \Pi_0 \psi \cdot n_i \rangle_{\Gamma_i} \right] \rho(\mathbf{y}) d\mathbf{y} \\
&= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\langle p - \mathcal{P}_H p, (\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}) \cdot \mathbf{n}_i + (\Pi_i \boldsymbol{\psi} - \boldsymbol{\psi}) \cdot \mathbf{n}_i + \boldsymbol{\psi} \cdot \mathbf{n}_i \rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\
&\leq \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p - \mathcal{P}_H p\|_{\Gamma_i} \Big(\|(\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}) \cdot \mathbf{n}_i\|_{\Gamma_i} + \|(\Pi_i \boldsymbol{\psi} - \boldsymbol{\psi}) \cdot \mathbf{n}_i\|_{\Gamma_i} \Big) \\
&\qquad + \|p - \mathcal{P}_H p\|_{-1/2,\Gamma_i} \|\boldsymbol{\psi} \cdot \mathbf{n}_i\|_{1/2,\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \quad \text{by (4.24)} \\
&\leq C \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p\|_{s,\Gamma_i} H^s \Big(\|\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}\|_{D_i} h^{-1/2} + \|\boldsymbol{\psi}\|_{1/2,\Gamma_i} h^{1/2} \Big) \\
&\qquad + \|p\|_{s,\Gamma_i} H^{s+1/2} \|\boldsymbol{\psi}\|_{1/2,\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \quad \text{by (4.20), (4.23), (4.25)} \\
&\leq C \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p\|_{s+1/2,D_i} H^s \Big(\|\boldsymbol{\psi}\|_{1,D_i} H^{1/2} + \|\boldsymbol{\psi}\|_{1,D_i} h^{1/2} \Big) \\
&\qquad + \|p\|_{s+1/2,D_i} H^{s+1/2} \|\boldsymbol{\psi}\|_{1,D_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \quad \text{by (4.22), (4.16)} \\
&\leq CH^{s+1/2} \int_{\mathbb{S}} \sum_{i=1}^{N_D} \big[\|p\|_{s+1/2,D_i} \|\hat{p} - p_{h,m}\|_{D_i} \big] \rho(\mathbf{y}) d\mathbf{y} \\
&\leq CH^{s+1/2} \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|p\|^2_{s+1/2,D_i} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \|\hat{p} - p_{h,m}\|_W .
\end{aligned}
$$

The proof is completed using Theorem 4.3.1 together with these estimates. □

4.4 THREE COLLOCATION-MMMFEM ALGORITHMS

To form the fully discrete stochastic solution to (4.7)–(4.9), we employ the parallel substructuring algorithm of Section 3.1.1 for each permeability realization $K^{\{k\}} = K(\mathbf{x}, \mathbf{y}_k)$, $k = 1, \ldots, N_{\text{real}}$. Therefore we have the following sequence of linear interface problems to solve: find $\lambda_H^{\{k\}} \in M_H(\Gamma)$ such that for $k = 1, \ldots, N_{\text{real}}$,

$$B_H^{\{k\}} \lambda_H^{\{k\}} = g_H^{\{k\}}, \tag{4.40}$$

where the linear operators $B_H^{\{k\}} : M_H(\Gamma) \to M_H(\Gamma)$, $B_{H,i}^{\{k\}} : M_{H,i}(\Gamma_i) \to M_{H,i}(\Gamma_i)$, and the vector $g_H^{\{k\}} \in M_H(\Gamma)$ are defined by

$$B_H^{\{k\}} \lambda = \sum_{i=1}^{N_D} B_{H,i}^{\{k\}} \lambda, \quad \langle B_{H,i}^{\{k\}} \lambda, \mu \rangle_{\Gamma_i} = b_{H,i}^{\{k\}}(\lambda, \mu) \ \forall \mu \in M_{H,i}, \quad \langle g_H^{\{k\}}, \mu \rangle_{\Gamma} = g_H^{\{k\}}(\mu) \ \forall \mu \in M_H.$$

In this section we describe three Collocation-MMMFEM algorithms. We will measure their computational effort in terms of the number of these subdomain solves, as they are dominant computational cost of the MMMFEM.

4.4.1 Collocation with Traditional MMMFEM

First we apply the traditional implementation of the MMMFEM, given by Algorithm 3.1, to each global realization of the stochastic collocation method. Recall that this algorithm requires solving one Dirichlet-to-Neumann subdomain solve per interface CG iteration, and this number grows with the condition number of the problem. When this method is coupled with the stochastic collocation method, this cost is multiplied by the number of realizations. We call this Method S1 and it is summarized in Algorithm 4.2.

Theorem 4.4.1. *The computational cost for each subdomain D_i for Method S1 is given by*

$$\begin{pmatrix} \textit{Number of Solves for} \\ \textit{Method S1} \end{pmatrix} = \sum_{k=1}^{N_{real}} (N_{iter}(k) + 2). \tag{4.41}$$

Algorithm 4.2 (Method S1) - Collocation without a Multiscale Flux Basis.

1: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
2: Generate permeability realization $K^{\{k\}}$ corresponding to global index k
3: Solve interface problem (4.40) using Algorithm 3.1
4: Multiply solution by collocation weight and sum to statistical moments
5: **end for**

4.4.2 Collocation with a Deterministic Multiscale Flux Basis

The second method uses the Multiscale Flux Basis implementation, given in Algorithm 3.3, to solve each global realization of the stochastic collocation method.

A subdomain problem is solved for each mortar degree of freedom before the interface iteration begins, forming a basis of coarse scale flux responses containing all the necessary information to solve one deterministic problem. No interprocess communication is required in the formation of the basis. Linear combinations of the basis are used to evaluate the flux operators during the interface iteration so that no additional subdomain solves are necessary, except one or more additional solves at the conclusion of the iteration to recover the fine scale solution. The computational cost is a fixed and controllable quantity, and therefore does not worsen with the condition number of the problem. Indeed, it was shown to be more efficient than the traditional implementation in most cases for deterministic problems. This gain in computational efficiency increases with the number of subdomains, and also in cases where a basis can be computed once and then reused many times.

This approach can be coupled to stochastic collocation method in a straightforward way by forming a new deterministic multiscale basis for each realization. We call this Method S2, which is summarized by Algorithm 4.3.

Theorem 4.4.2. *The computational cost for each subdomain D_i for Method S2 is given by*

$$\begin{pmatrix} \textit{Number of Solves for} \\ \textit{Method S2} \end{pmatrix} = (N_{dof}(i) + 2) * (N_{real}). \tag{4.42}$$

Note that each subdomain performs a different number of solves because they may have a different number of mortar degrees of freedom.

Algorithm 4.3 (Method S2) - Collocation with a Deterministic Multiscale Flux Basis.

1: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
2: Generate permeability realization $K^{\{k\}}$ corresponding to global index k
3: Compute Multiscale Flux Basis for global index k with Algorithm 3.2
4: Solve interface problem (4.40) with Algorithm 3.3 using the basis from line 3
5: Multiply solution by collocation weight and sum to statistical moments
6: **end for**

On line 3, we emphasize that the formation of a Deterministic Multiscale Flux Basis $\{\psi_{H,i}^{(j),\{k\}}\}_{j=1}^{N_{\text{dof}}(i)}$ is for a single global permeability realization k, on each subdomain D_i. For each subdomain D_i, $i = 1, \ldots, N_D$, let $\{\phi_{H,i}^{(j)}\}_{j=1}^{N_{\text{dof}}(i)}$ denote a mortar basis for $M_{H,i}(\Gamma_i)$. Individual flux responses for realization k are computed by evaluating the action of the operator $B_{H,i}^{\{k\}}$ on these functions. On line 4, each CG iteration of the interface iteration uses a linear combination of Multiscale Flux Basis functions to replace the need to solve additional subdomain problems. They are discarded and recalculated for realization $k+1$.

4.4.3 Collocation with a Stochastic Multiscale Flux Basis

The main idea behind the Multiscale Flux Basis Implementation is to form a basis containing all the necessary information to solve the interface problem by solving as few linear systems as possible. In Method S2, if each realization saves just a few solves, then when performing several thousand realizations the overall savings will be great compared to Method S1. Moreover, in certain situations it is even possible to perform even fewer solves than Method S2.

In the setting of nonstationary random porous media with localized KL Regions throughout the domain, it is possible to get even greater computational savings with the formation of a Stochastic Multiscale Flux Basis. Recalling equations (4.11) and (4.15), both tensor product and sparse grids have a repeated local structure in the KL Regions. A Stochastic Multiscale Flux Basis can be formed by looping over all local realizations of a subdomain's KL Region in a precomputation loop before the stochastic collocation begins. We call this

Method S3, and is summarized by Algorithm 4.4.

Theorem 4.4.3. *The computational cost for each subdomain D_i for Method S3 is given by*

$$\begin{pmatrix} \textit{Number of Solves for} \\ \textit{Method S3} \end{pmatrix} = (N_{dof}(i) * N_{real}(j)) + (2 * N_{real}). \tag{4.43}$$

Each subdomain performs a different number of solves because they may have a different number of mortar degrees of freedom and may belong to different KL regions with different numbers of local realizations.

Algorithm 4.4 (Method S3) - Collocation with a Stochastic Multiscale Flux Basis.

1: **for** $k' = 1, \ldots, N_{\text{real}}(j)$ **do** {Precomputation Loop}
2: Generate permeability realization corresponding to local index k'
3: Compute and store Multiscale Flux Basis for local index k'
4: **end for**
5: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
6: Generate permeability realization corresponding to global index k
7: Convert global index k to local index k' {Using Algorithm 4.5 or 4.6}
8: Solve interface problem (4.40) with MMMFEM using the basis with local index k' from Precomputation Loop
9: Multiply solution by collocation weight and sum to statistical moments
10: **end for**

On line 7, the global to local collocation index conversion is the critical step in being able to perform Method S3. Any algorithms developed for this purpose will be dependent on the ordering of the points that was used. For a tensor product grid, recall from (4.12) that we chose to follow the natural ordering by 1-D point first, local dimension next, and KL Region last. In Algorithm 4.5, we give a global to local index conversion algorithm for this ordering. It is very similar to the algorithm one uses to convert an integer from one base into another, with the modification that each digit has a different base.

For a sparse grid, the indexing of the points is far more complicated than a tensor product grid due to its hierarchical construction and skipping of repeated points. Nevertheless, it is

still possible to formulate a global to local index conversion that is more efficient than a brute force approach. Recall from Algorithm 4.1 that we chose to follow the natural ordering by level first, followed by a partition of that level into N_Ω parts, followed by 1-D point, and global dimension last. In Algorithm 4.6, we give the global to local index conversion algorithm for this ordering. It uses formula (4.12), where the global point and partition are truncated, and the indexing scheme is applied to the local dimensions.

Algorithm 4.5 Global to Local Index Conversion for a Tensor Product Grid.

input: Global Index g, KL Region r
$remainder \leftarrow g$
for $i = 1, \ldots, N_\Omega$ **do**
  $modulus \leftarrow 1$
  for $j = 1, \ldots, N_\Omega - i - 1$ **do**
    $modulus \leftarrow modulus * N_{\text{real}}(j)$
  end for
  if $(N_\Omega - i + 1 = r)$ **then return** $remainder/modulus$ {Return Local Index}
  $remainder \leftarrow \mathbf{mod}(remainder, modulus)$
end for

4.5 NUMERICAL EXAMPLES

The following numerical examples compare the relative computational efficiency of Methods S1, S2, and S3, by presenting tables showing the maximum number of required linear systems and the maximum total runtime per processor. As shown in Figure 4.2, there are four examples, and in each case we test both tensor product and sparse grid collocations. Example 4.5.1 is a 2-D "L-shape" with two KL regions and is a relatively small problem that is fast to compute. Example 4.5.2 is a 2-D "checkerboard" with four KL regions that demonstrates a procedure for adaptive mesh refinement in the spatial grid. Example 4.5.3 is a 3-D "SPE10 benchmark" test with either two or twenty KL regions, and is a much more computationally intensive problem to solve than Examples 4.5.1 and 4.5.2. Finally, Example 4.5.4 is a 2-D

Algorithm 4.6 Global to Local Index Conversion for a Sparse Grid.

input: Global Index g, KL Region r

$point \leftarrow (h_1, \ldots, h_{N_{\text{term}}})$

$part \leftarrow (p_1, \ldots, p_{N_{\text{term}}})$ {Using Algorithm 1 with index g}

$subpoint \leftarrow (h_{\kappa(r,1)}, \ldots, h_{\kappa(r,N_{\text{term}}(r))})$

$subpart \leftarrow (p_{\kappa(r,1)}, \ldots, p_{\kappa(r,N_{\text{term}}(r))})$ {Truncate to local dimensions}

$l \leftarrow 1$

if $(subpoint = 0)$ **then return** l {Special case for 0 partition}

for $\ell = 1, \ldots, \ell_{\max}$ **do** {Loop over sub-levels}

  for $i = 1, \ldots, \dfrac{(\ell + N_{\text{term}}(r) - 1)!}{\ell!(N_{\text{term}}(r) - 1)!}$ **do** {Loop over sub-partitions}

    $newpart \leftarrow (q_1, \ldots, q_{N_{\text{term}}(r)})$ {The i-th multi-index}

    $\mathbf{m} = 2^{newpart + \mathbb{1}} - \mathbb{1}$

    for $j = 1, \ldots, \prod_\alpha \mathbf{m}(\alpha)$ **do** {Loop over sub-points}

      $newpoint \leftarrow (k^{j_1}_{\mathbf{m}(1)}, \ldots, k^{j_{N_{\text{term}}(r)}}_{\mathbf{m}(N_{\text{term}})})$ {The j-th point using (4.12)}

      $\tilde{newpart} \leftarrow (\tilde{p}_1, \ldots, \tilde{p}_{N_{\text{term}}(r)})$ where $\tilde{p}_i = \begin{cases} p_{\kappa(r,i)}, & \text{if } k^{j_i}_{\mathbf{m}(i)} \neq 0 \\ 0, & \text{if } k^{j_i}_{\mathbf{m}(i)} = 0 \end{cases}$

      if $(\tilde{newpart} = newpart)$ **then**

        $l \leftarrow l + 1$

        if $(newpoint = subpoint)$ **then return** l {Return Local Index}

      else

        {Repeated point; skip it}

      end if

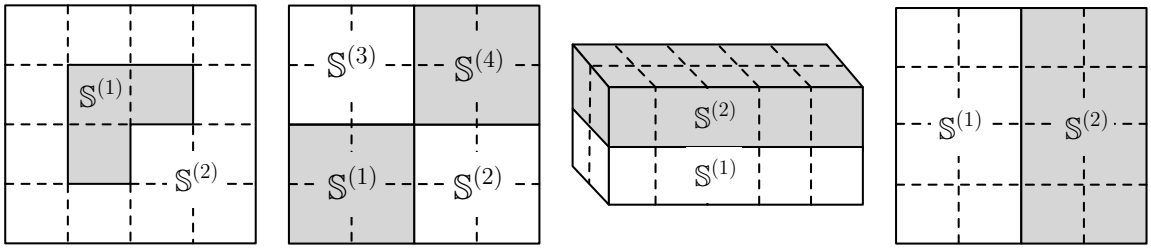
    end for

  end for

end for

"convergence test" with two equally sized KL regions, meant to empirically test convergence rates in both stochastic and physical space.

Figure 4.2: Subdomain and KL region layouts for Examples 4.5.1–4.5.4. Dashed lines represent subdomain boundaries and shading distinguishes between KL regions.



To perform these tests, Methods S1, S2, and S3 were coded into the PARCEL simulator. The runtimes were recorded by compiling the code without optimization using Intel's ifort compiler and MKL library, and run with MVAPICH2 on a parallel cluster of Intel Xeon E5430 2.66GHz processors.

4.5.1 L-Shape Example

Description. This example has $N_\Omega = 2$ KL Regions on the domain $[0,1]^2$. We test a low number of terms with an isotropic tensor product grid and a large number of terms with a level $\ell_{\max} = 1$ sparse grid. KL Region $\mathbb{S}_1$ is an L-shaped inclusion with a mean value of $E[Y] = 3.0$, $N_{\text{term}}(1) = 3 \times 3 = 9$ (with tensor grid) and $14 \times 14 = 196$ terms (with sparse grid), $\eta = 0.01$ correlation lengths, and $\sigma^2 = 1.0$ variance. KL Region $\mathbb{S}_2$ is the remainder of the domain with a mean value of $E[Y] = 0.0$, $N_{\text{term}}(2) = 2 \times 1 = 2$ (with tensor grid) and $N_{\text{term}}(2) = 2 \times 2 = 4$ (with sparse grid) terms, $\eta = 0.10$ correlation lengths, and $\sigma^2 = 1.0$ variance. Flow is induced from left-to-right with Dirichlet boundary conditions $g_D = 1$ on face $\{x = 0\}$, $g_D = 0$ on face $\{x = 1\}$, and no-flow homogeneous Neumann boundary conditions on the other two edges.

The domain for Example 4.5.1 is divided into $N_D = 4 \times 4 = 16$ subdomains. Tensor product collocation will use a uniform spatial grid, with all subdomains containing 25×25 elements. The interfaces use continuous linear mortars with 10 elements. Sparse grid

collocation will use a non-uniform spatial grid such that subdomains in KL Region $\mathbb{S}_1$ have 20×20 elements, and subdomains in KL Region $\mathbb{S}_2$ have 4×4 elements. The interfaces use continuous linear mortars, with the number of elements on $\mathbb{S}_1 - \mathbb{S}_1$, $\mathbb{S}_1 - \mathbb{S}_2$, and $\mathbb{S}_2 - \mathbb{S}_2$ interfaces being 10, 4, and 2 elements, respectively.

Discussion. First we try isotropic tensor product collocation with a low number of terms. Using $N_{\text{coll}} = 2$ collocation points in $N_{\text{term}} = 9 + 2 = 11$ stochastic dimensions requires a total of $N_{\text{real}} = 2^{11} = 2048$ global realizations and a maximum number of $N_{\text{real}}(i) = 2^9 = 512$ local realizations, giving a global to local ratio of 4.0. Table 4.1 shows that the number of linear systems was reduced by 61% with a deterministic multiscale basis and by 90% with a stochastic multiscale basis. However, in practice the runtime was only reduced by 33% and 45% respectively. This is because the use of a multiscale basis in Methods S2 and S3 does nothing to reduce the interprocess communication necessary to converge the CG iterations during each realization of the stochastic collocation loop. Notice that Method S3 took only 11.5 seconds to complete nearly all of the linear systems necessary to complete the problem, because these systems are relatively small and easy to solve. One way to reduce the time spent on communication would be to use a preconditioner to improve the condition number of the interface problem, which could be done in conjunction with the multiscale basis implementation. Plots are shown of the calculated statistics in Figure 4.3 with $N_{\text{coll}} = 2$ collocation points.

Next we try sparse grid collocation with a large number of terms. Using a level $\ell_{\max} = 1$ sparse grid in $N_{\text{term}} = 196 + 4 = 200$ stochastic dimensions requires a total of $N_{\text{real}} = 401$ global realizations and a maximum number of $N_{\text{real}}(i) = 393$ local realizations, giving the much smaller global to local ratio of 1.02. This time the number of linear systems was reduced by 25% with a deterministic multiscale basis and by 26% with a stochastic multiscale basis. The runtimes, however, remained nearly constant, giving evidence that in practice there is not much benefit to using multiscale basis techniques with sparse grid collocation on small problems. Plots are shown of the calculated statistics in Figure 4.4 with a level $\ell_{\max} = 1$ sparse grid. Notice that third order accuracy with a tensor grid on 10 stochastic dimensions required 2048 realizations, while third order accuracy with a sparse grid on 200 stochastic

dimensions only required 401 realizations. It would not be possible to perform tensor product collocation in 200 dimensions because it would require over 1.6E60 realizations.

Table 4.1: Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.1 with $N_{\text{term}} = 11$ and $N_{\text{term}} = 200$. Parenthesis denotes precomputation loop.

$N_{term} = 11$, Tensor Product Collocation, $N_{coll} = 2$ (2048 realizations)

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	542,498	208,896	55,296 (51,200)
Runtime in Seconds	301.8	202.5	166.6 (11.5)

$N_{term} = 200$, Sparse Grid Collocation, $\ell_{max} = 1$ (401 realizations)

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	35,082	26,466	25,954 (25,152)
Runtime in Seconds	34.7	33.4	32.5 (5.5)

4.5.2 Checkerboard Example

Description. This example is meant to demonstrate an adaptive procedure which is used to refine the spatial grid. There are $N_\Omega = 4$ KL Regions on the domain $[0,1]^2$. The bottom-left and upper-right KL Regions $\mathbb{S}^{(i)}, i = 1, 4$ each have a mean value of $E[Y] = 4.6$, $N_{\text{term}}(i) = 2 \times 1 = 2$ terms, $\eta = 0.1$ correlation lengths, and $\sigma^2 = 1.0$ variance. The top-left and bottom-right KL Regions $\mathbb{S}^{(i)}, i = 2, 3$ each have a mean value of $E[Y] = 0.0$, $N_{\text{term}}(i) = 2 \times 2 = 4$ terms, $\eta = 0.01$ correlation lengths, and $\sigma^2 = 100.0$ variance. Tensor product collocation with $N_{\text{coll}} = 2$ and sparse grid collocation with level $\ell_{\text{max}} = 2$ is performed in $N_{\text{term}} = 2+4+4+2 = 12$ stochastic dimensions, requiring 4096 and 361 global realizations, 16 and 57 maximum local realizations, giving global to local ratios of 256.0 and 6.33, per mesh adaptation respectively. Flow is induced from left-to-right with the same boundary conditions as Example 4.5.1.

Figure 4.3: Realization of permeability (top-left), mean pressure (top-middle), mean velocity magnitude (top-right), variance of pressure (bottom-left), variance of horizontal velocity (bottom-middle), and variance of vertical velocity (bottom-right) for tensor product collocation in Example 4.5.1 with $N_{\text{term}} = 11$.

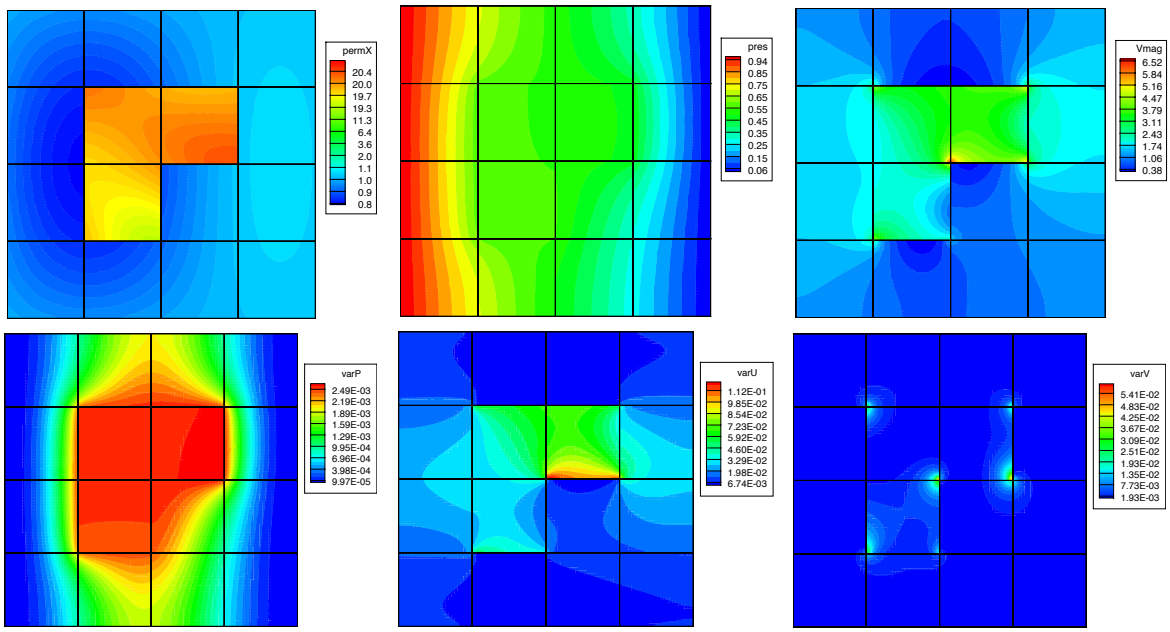
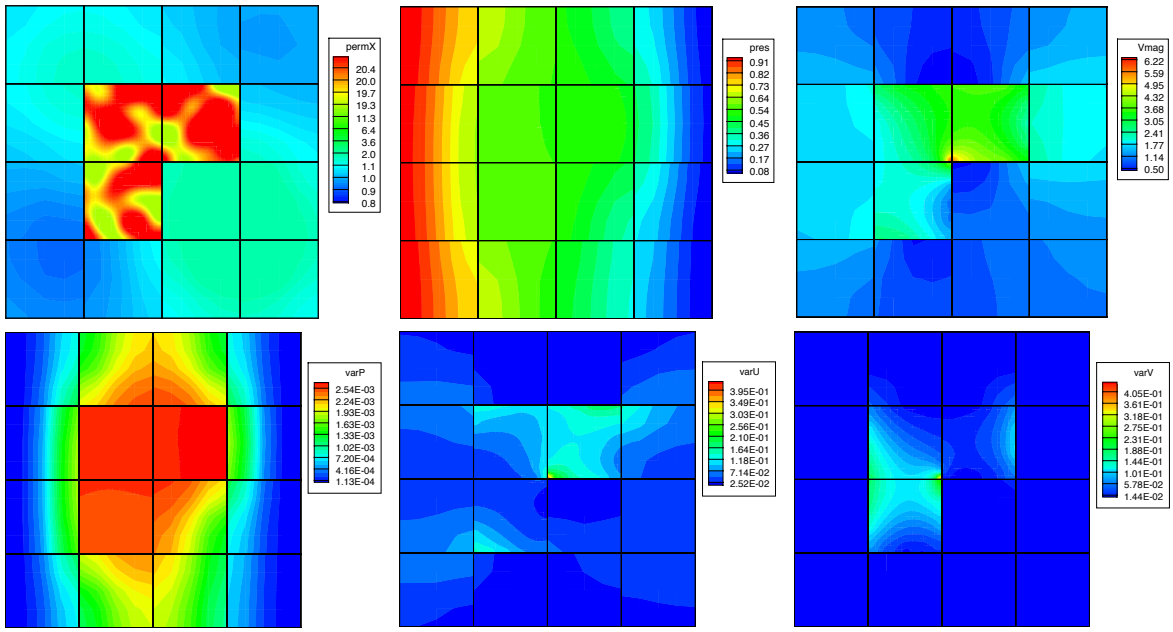


Figure 4.4: Realization of permeability (top-left), mean pressure (top-middle), mean velocity magnitude (top-right), variance of pressure (bottom-left), variance of horizontal velocity (bottom-middle), and variance of vertical velocity (bottom-right) for level $\ell_{\max} = 1$ sparse grid collocation in Example 4.5.1 with $N_{\text{term}} = 200$.



The adaptive procedure can be described as follows: The domain is divided into $N_D = 8 \times 8 = 64$ subdomains, and all interfaces use continuous linear mortars. Each subdomain begins with a 2×2 local grid, and each mortar begins with a single element. The stochastic collocation method is performed with Methods S1, S2, or S3 using this spatial grid. Upon completion of the collocaiton, a residual-based *a posteriori* error indicator is computed using the expectation of the pressure together with the mean permeability, which can be found in [87, 9]. The spatial grids of subdomains that contain errors beyond a given tolerance are refined, as well as the mortars that touch those refined subdomains. At this point, the entire collocation is performed again using the new spatial grid, and the procedure stops when no new subdomains need refinement. Note that this is merely a heuristic procedure, and is given for illustrative purposes only. In the future we anticipate the formulation of a rigorous, residual-based *a posteriori* error indicator using the same norm as given in Section 4.3.

Discussion. Since the number of random dimensions was kept low, Table 4.2 shows the amount of work necessary for both tensor product and sparse grid collocation on the same test. Method S2 showed an 82-83% decrease and Method S3 showed a 93-99% decrease in the number of linear systems required to solve the collocation on refinement levels 1-4 when compared to Method S1. Once again, the runtimes remained constant, because the small size of the linear systems keep the problems communication bound.

Figure 4.5 shows how the adaptive procedure refined the spatial grid, in a similar way as one expects for a deterministic problem with the given mean permeability. On each subsequent refinement, an entire new stochastic collocation was performed. Figures 4.6 and 4.7 demonstrate how both expectation and variance of pressure and velocity magnitude are improved on progressively finer spatial grids. They are presented for the level 2 sparse grid collocation, which is estimated to be accurate to total degree 5.

4.5.3 SPE10 Benchmark Example

Description. The mean permeability in the third example is a 3-dimensional scalar field of actual geological measurements, obtained from the x-component of the Society of Petroleum Engineers' (SPE) Comparative Solution Project [22]. It is a challenging benchmark problem

Table 4.2: Comparison of runtime and linear systems across refinement levels 1-4 with the three collocation algorithms for Example 4.5.2. Parenthesis denotes precomputation loop.

$N_{term} = 12$, *Tensor Product Collocation,* $N_{coll} = 2$ *(4096 realizations)*

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	3,722,250	655,360	35,200 (1,152)
Runtime in Seconds	5,353	5,409	5,280 (0.2)

$N_{term} = 12$, *Sparse Grid Collocation,* $\ell_{max} = 2$ *(361 realizations)*

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	341,836	57,760	11,552 (4,104)
Runtime in Seconds	493.7	493.4	487.0 (0.5)

Figure 4.5: Spatial grids for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2.

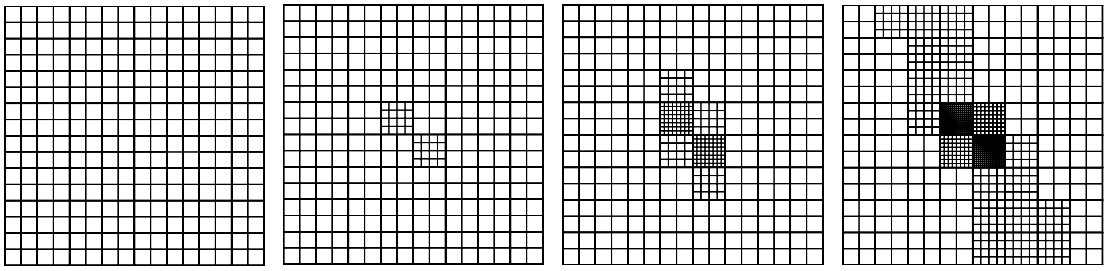


Figure 4.6: Mean pressure (top) and pressure variance (bottom) for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2.

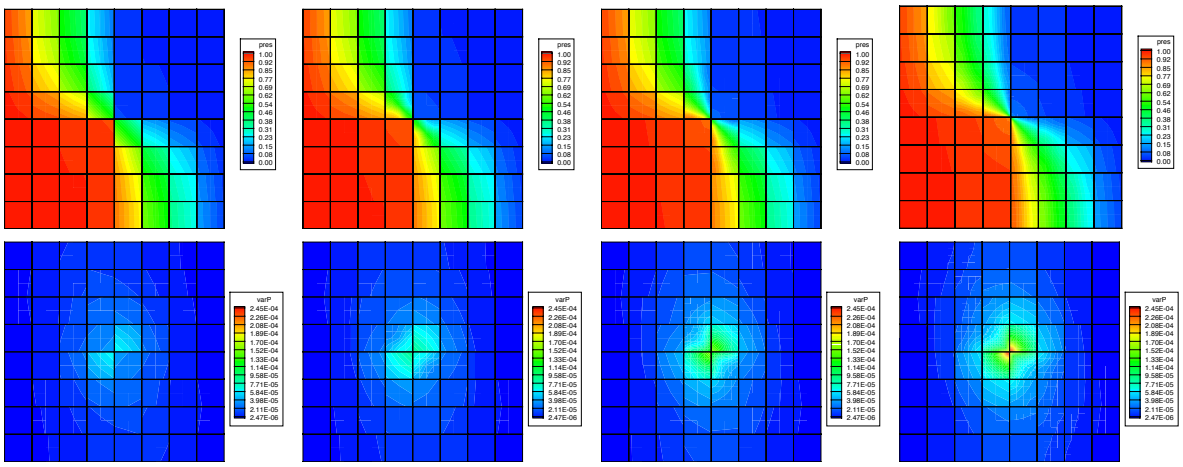
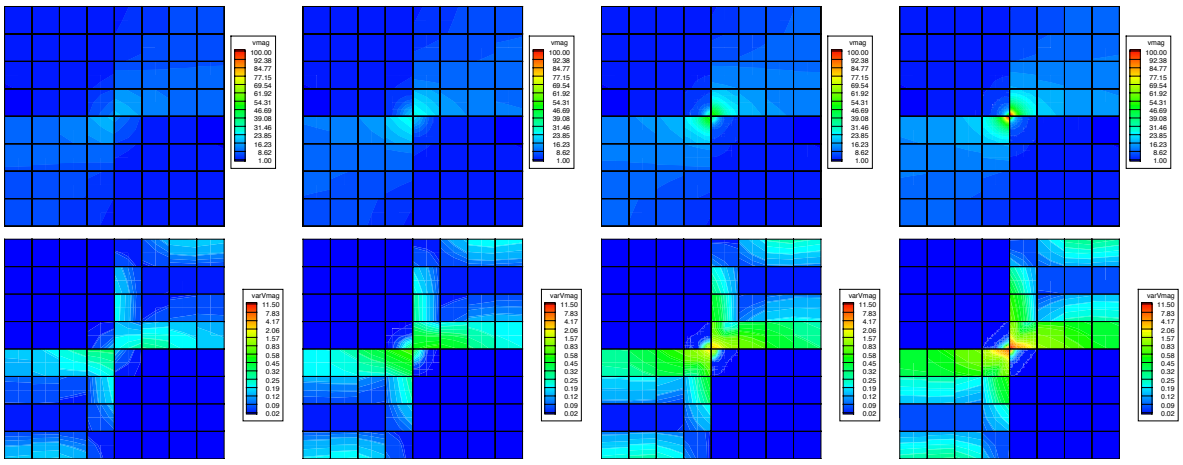


Figure 4.7: Mean velocity magnitude (top) and velocity magnitude variance (bottom) for refinement levels 1-4 with $\ell_{\max} = 2$ sparse grid on Example 4.5.2.



with a cartesian grid of $60 \times 220 \times 85$, giving a total of $1,122,000$ finite elements. This dataset is part of a Brent sequence, with the lower 35 layers representing a prograding Tarbert formation, and the top 50 layers representing a fluvial Upper Ness formation. A flow is induced from front-to-back with Dirichlet boundary conditions $g_D = 1$ on face $\{y = 0\}$, $g_D = 0$ on face $\{y = 220\}$, and no-flow homogeneous Neumann boundary conditions on the other four faces.

For the discretization, the fine scale grid is broken up into $N_D = 2 \times 5 \times 2 = 20$ subdomains of nearly equal size. On the interfaces, the mortar space is comprised of faces with discontinuous linear mortars with a single 1×1 element. Unlike the previous two examples, the size of these subdomains is sufficiently large so that the time spent solving a typical linear system dominates the time needed to perform both interprocess communication and a multiscale basis linear combination.

In Example 4.5.3(a), we perform tensor product collocation with noise being added to $N_\Omega = 2$ statistically independent KL Regions, roughly coinciding with the two geologic formations of the deterministic data. The first region is the lower 10 subdomains and is described by the parameters: $N_{\text{term}}(1) = 1 \times 4 = 4$ terms, $\eta = 6.0$ correlation lengths, $\sigma^2 = 1.7$ variance. The second region is the upper 10 subdomains and is described by the parameters: $N_{\text{term}}(2) = 1 \times 6 = 6$ terms, $\eta = 10.0$ correlation lengths, $\sigma^2 = 1.2$ variance. In Example 4.5.3(b) we switch to sparse grid collocation, and increase the number of terms in the bottom and top KL Regions to $N_{\text{term}}(1) = 4 \times 4 \times 4 = 64$ and $N_{\text{term}}(2) = 5 \times 5 \times 5 = 125$ respectively. In Example 4.5.3(c) we increase the number of KL Regions to $N_\Omega = 20$, one in each subdomain with $N_{\text{term}}(i) = 2 \times 3 \times 2 = 12$ terms.

Discussion. In this 3-D benchmark problem, the size of the subdomain problems is sufficiently large so that the time spent solving a linear system dominates the time needed to perform the interprocess communication in the CG iteration of each realization. Tensor product collocation with $N_{\text{coll}} = 2$ in $N_{\text{term}} = 4 + 6 = 10$ stochastic dimensions requires a total of $N_{\text{real}} = 2^{10} = 1024$ global realizations, $N_{\text{real}}(i) = 64$ maximum local realizations, giving a global to local ratio of 16.0. Table 4.3 shows the linear systems was reduced by 92% with a deterministic multiscale basis and 99% with a stochastic multiscale basis. Due to the

sheer size of the subdomain problems, the runtime was also dramatically reduced by 85% and 89%, respectively. Figures 4.8–4.12 show the results of the computations.

In Example 4.5.3(b), sparse grid collocation with $\ell_{\max} = 1$ in $N_{\text{term}} = 64 + 125 = 189$ stochastic dimensions requires a total of $N_{\text{real}} = 379$ global realizations, $N_{\text{real}}(i) = 251$ maximum local realizations, giving a global to local ratio of 1.51. The number of linear systems was reduced by 91% and 94%, and the runtime was reduced by 83% with deterministic multiscale basis, but was slightly worse with a stochastic multiscale basis with a reduction of only 80%. This still makes Method S3 unquestionably superior to Method S1, but only better than Method S2 in some cases. The reason Method S3 is faster than Method S2 in the tensor grid case but not in the sparse grid case is due to the global to local ratio. When this ratio is smaller, more runtime is spent forming the stochastic multiscale basis. For instance, in Example 4.5.3(b) the precomputation loop is over 45% of the total runtime, while in Example 4.5.3(a) it is only 7%. The structure of a tensor product grid causes this ratio to remain very large, even when the difference between global and local dimension is small.

Recall that the main benefit to using a sparse grid is that the number of points grows more modestly than a tensor grid as dimension increases. Unfortunately this means the global to local ratio are smaller, so in practice Method S3 will only be faster than Method S2 when the difference between global and local dimension is large. Indeed, in Example 4.5.3(c) we constuct a third SPE10 test to show this effect, using $N_\Omega = 20$ KL regions each having $N_{\text{term}}(i) = 12$ dimensions. Sparse grid collocation with $\ell_{\max} = 1$ in $N_{\text{term}} = 12 * 20 = 240$ stochastic dimensions requires a total of $N_{\text{real}} = 481$ global realizations, $nreal(i) = 41$ maximum local realizations, giving a global to local ratio of 19.0. The results are given in Table 4.4, and as one can see in this case Method S3 shows an improvement in runtime over Method S2. Both multiscale methods are still far superior to the traditional implementation.

4.5.4 Convergence Example

Description. This example empirically tests convergence rates in both stochastic and physical space. There are $N_\Omega = 2$ KL Regions on the domain $[0,1]^2$ with $N_D = 4 \times 4 = 16$ subdomains. Throughout the domain, a mean value of $E[Y] = 5000(1 - \sin(20x)\sin(20y))$

Table 4.3: Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.3 with $N_\Omega = 2$ KL Regions. Parenthesis denotes precomputation loop.

$N_{term} = 10$, *Tensor Product Collocation,* $N_{coll} = 2$ *(1024 realizations)*

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	236,964	18,432	3,072 (1,024)
Runtime in Hours	110.34	16.95	11.72 (0.82)

$N_{term} = 189$, *Sparse Grid Collocation,* $\ell_{max} = 1$ *(379 Realizations)*

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	79,047	6,822	4,774 (4,016)
Runtime in Hours	37.16	6.27	7.33 (3.32)

Table 4.4: Comparison of runtime and linear systems with the three collocation algorithms for Example 4.5.3(c) with $N_\Omega = 20$ KL Regions. Parenthesis denotes precomputation loop.

$N_{term} = 240$, *Sparse Grid Collocation,* $\ell_{max} = 1$ *(481 realizations)*

	Method S1	**Method S2**	**Method S3**
Max. Linear Systems	101,826	8,658	1,362 (400)
Runtime in Hours	47.8	7.97	5.50 (0.38)

Figure 4.8: Realization of permeability (left) and its corresponding solution (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions.

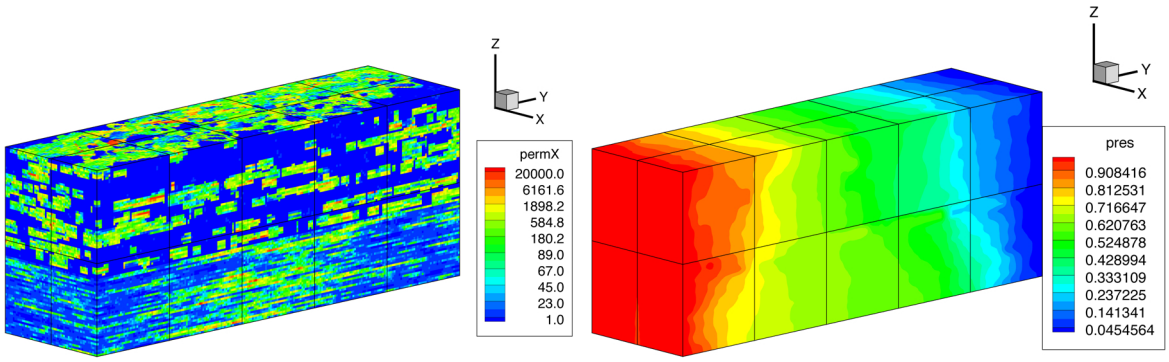


Figure 4.9: Mean solution (left) and Pressure Variance (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions.

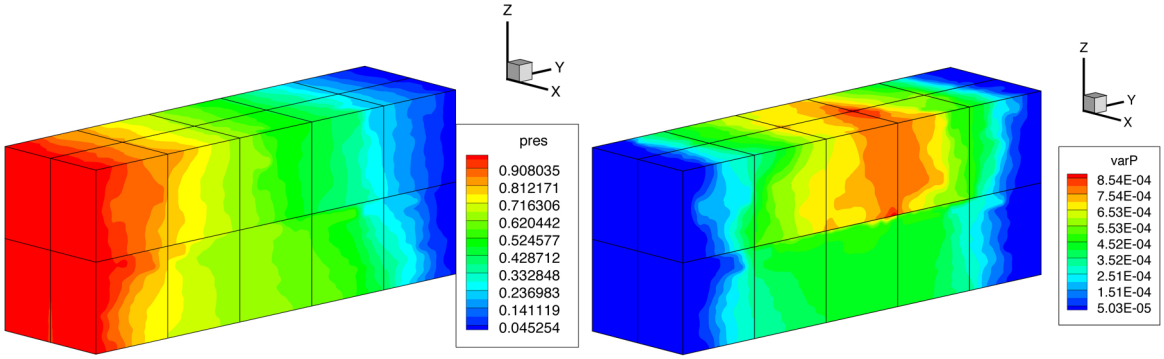


Figure 4.10: X-Velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_\Omega = 2$ KL Regions.

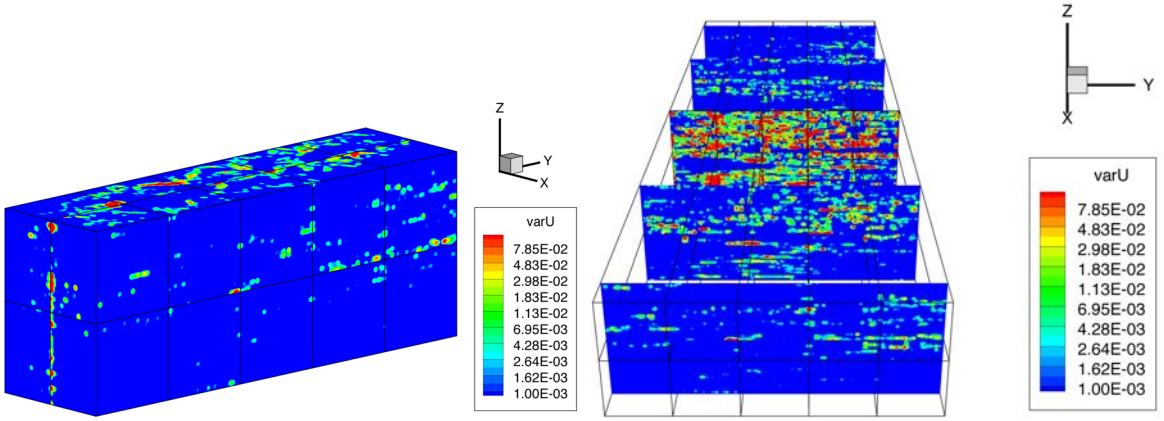


Figure 4.11: Y-velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_{\Omega} = 2$ KL Regions.

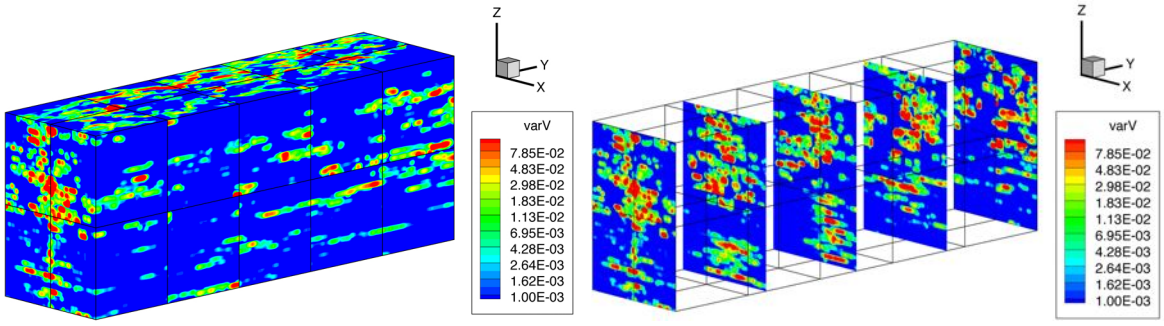
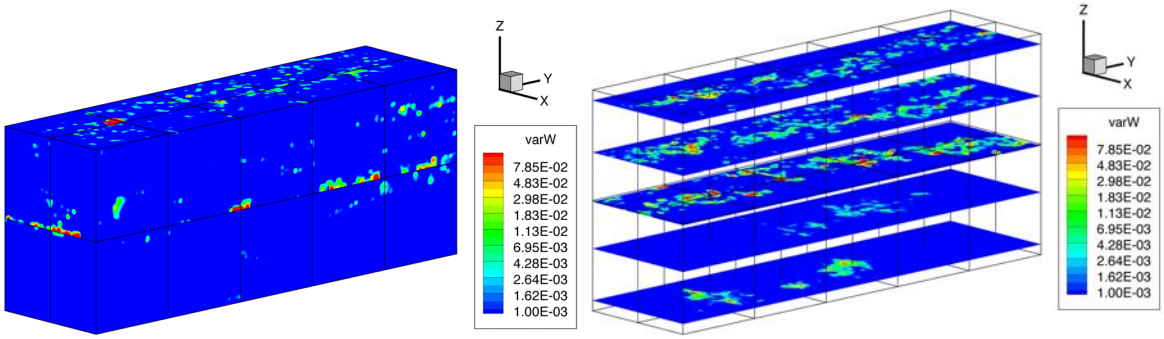


Figure 4.12: Z-velocity variance (left) with several cross-sections (right) for Example 4.5.3 with $N_{\text{term}} = 10$ terms and $N_{\Omega} = 2$ KL Regions.



is used throughout. KL Region $\mathbb{S}^{(1)}$ is the left half of the domain with $N_{\text{term}} = 2 \times 1 = 2$ terms, correlation length $\eta = 0.13$, and variance $\sigma = 1.0$. KL Region $\mathbb{S}^{(2)}$ is the right half of the domain with $N_{\text{term}} = 2 \times 2 = 4$ terms, correlation length $\eta = 0.09$, and variance $\sigma = 1.1$.

Table 4.5: Table of convergence in physical space for Example 4.5.4. Relative errors reported against finest grid level; convergence ratios given in parentheses.

$\mathbb{S}^{(1)}$ Grid	Mort.	$\mathbb{S}^{(2)}$ Grid	$\frac{\|E[p-p_{true}]\|}{\|E[p_{true}]\|}$	$\frac{\|E[\mathbf{u}-\mathbf{u}_{true}]\|}{\|E[\mathbf{u}_{true}]\|}$	$\frac{\|E[\nabla\cdot(\mathbf{u}-\mathbf{u}_{true})]\|}{\|E[\nabla\cdot\mathbf{u}_{true}]\|}$
4×5	2	3×7	1.04E-02	1.76E-01	2.56E-01
8×10	4	6×14	3.93E-03 (2.64)	6.35E-02 (2.77)	1.02E-01 (2.51)
16×20	8	12×28	1.39E-03 (2.82)	1.71E-02 (3.71)	3.30E-02 (3.09)
32×40	16	24×56	3.74E-04 (3.72)	3.80E-03 (4.50)	1.14E-02 (2.89)
64×80	32	48×112	–	–	–

Discussion. Figures 4.13 and 4.14 show convergence rates in stochastic space, wherein all tests have a fixed spatial grid. Subdomains have 20×20 grids in the first region, 17×15 grids in the second region, and continuous linear mortars with 10 elements are used on all interfaces. The true stochastic solution is estimated by a level 6 sparse grid, and the lines represent different types of sampling methods with increasingly more realizations. Due to the use of a CG iteration in the interface problem, the slopes can be seen tapering off as the error reaches a threshold with a tolerance of $1E-6$. The tests were rerun with a tolerance of $1E-15$ to mitigate this effect, but this can be contrasted with the steeper slopes as seen in a single domain solver in such works as [83].

Table 4.5 shows convergence rates in physical space. A level $\ell_{\max} = 3$ sparse grid rule was used in stochastic dimensions, but we also note that these deterministic results were within round-off for an isotropic tensor grid rule with $N_{\text{coll}} = 3$. The first three columns indicate how different spatial grids were used in each KL region, and how they were refined on each subsequent row. The convergence ratios appear to confirm the theory.

Figure 4.13: Log-log plot of convergence in stochastic space for Example 4.5.4. Different types of sampling methods are shown in absolute L^2-error for pressure with $1E-6$ tolerance (left) and $1E-15$ tolerance (right).

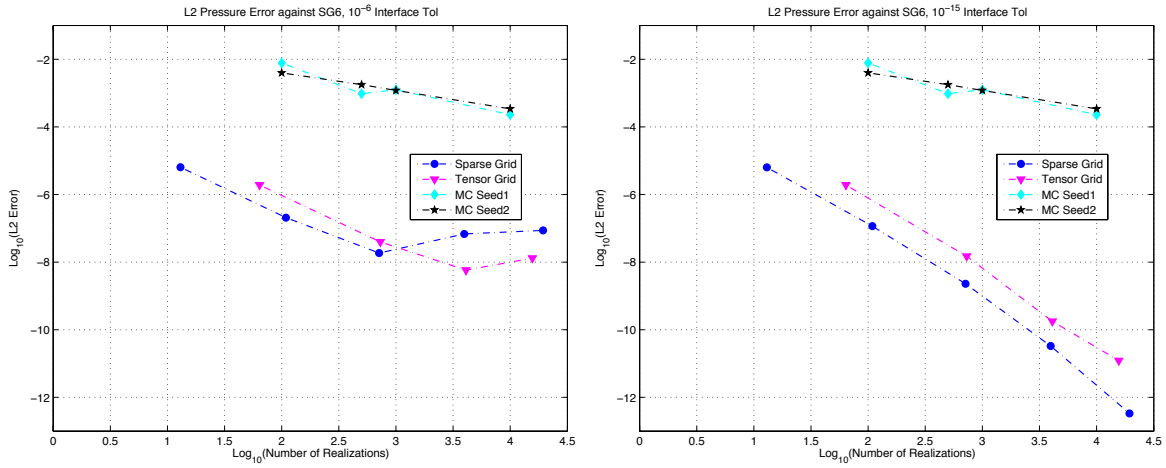
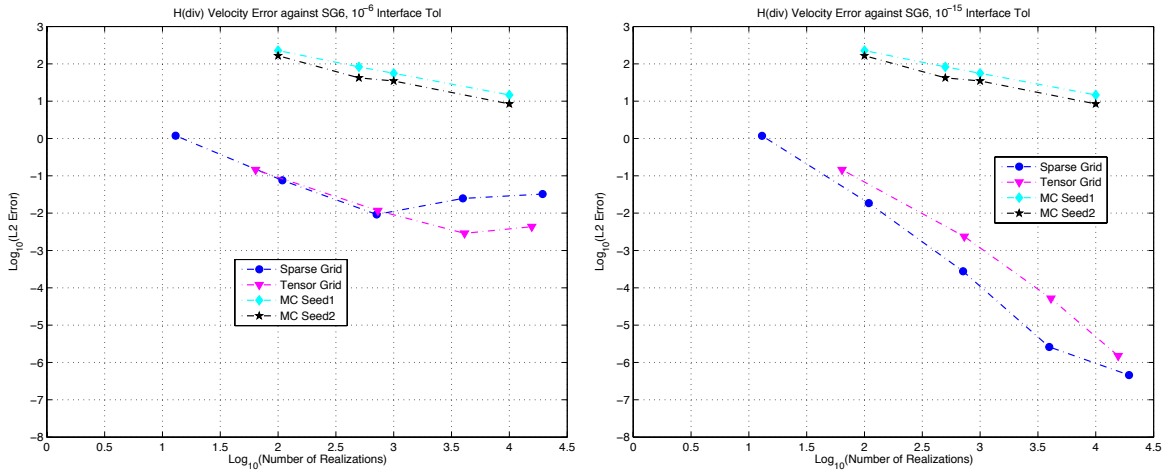


Figure 4.14: Log-log plot of convergence in stochastic space for Example 4.5.4. Different types of sampling methods are shown in absolute $H(div)$-error for velocity with $1E-6$ tolerance (left) and $1E-15$ tolerance (right).



5.0 MULTISCALE PRECONDITIONER AND TWO-PHASE FLOWS

This chapter describes and demonstrates the use of the Multiscale Flux Basis implementation (briefly, multiscale basis) as a preconditioner for interface problems, following the work of Wheeler, Wildey, and Yotov [85]. This previous work applied the preconditioner to uncertainty quantification for the linear interface problem in the single-phase model. The author of this thesis has helped contribute to the implementation of the multiscale basis approach and preconditioner into the IPARS simulator [70, 82, 66]. In this chapter, we summarize the forthcoming work that explores the use of the multiscale basis preconditioner in deterministic two-phase flows with non-linear interface problems [37].

Recall that in Chapter 3, the multiscale basis was introduced for a deterministic single-phase flow, which is an elliptic problem with a single linear interface problem (3.35). Contrast this with the stochastic single-phase problem in Chapter 4, which involved solving a sequence of linear interface problems (4.40). We described solving these problems with both deterministic (Method S2) and stochastic (Method S3) multiscale basis algorithms. Both of these methods required the calculation and re-calculation of *many* multiscale bases for each subdomain across multiple realizations.

In the original implementation of the MMMFEM, the coarse scale interface problems are solved in parallel via an iterative Krylov method such as CG or GMRES, and each iteration requires the solution to fine scale subdomain problems. The multiscale basis implementation shifts the computational burden to the formation of such a basis. One it is formed, its usage is very cheap. However, if it must be re-computed many times, this can be costly. Nevertheless, in Chapter 4 this approach still succeeded in reducing both theoretical cost and actual runtime for uncertainty quantification, especially in non-stationary media.

To avoid the re-calculation of a multiscale basis in a sequence of linear interface problems,

it can be computed *once* for a single permeability that captures the main characteristics of the porous media, called a training permeability. For uncertainty quantification, the obvious choice is the mean permeability. As described in [85], the single multiscale basis for the training permeability can be used as an efficient physics-based preconditioner. The cost of applying the preconditioner is the need to solve another interface problem.

Numerical methods for deterministic two-phase flow simulations in the MMMFEM framework involve non-linear interface problems. When an Inexact Newton's method is used with a finite difference approximation of the Jacobian, it can be reduced to a sequence of linear problems to solve on each time step [94]. The training permeability for the multiscale basis preconditioner may be taken as the initial permeability, and used throughout the simulation. As expected, its effectiveness as a preconditioner will degrade in time. Alternatively, the multiscale basis preconditioner may be re-computed *sparingly* at regular time intervals, or until some tolerance is reached. In these ways, the multiscale basis implementation for a steady-state model with linear interface problem is extended to handle the unsteady model with non-linear interface problem. We will consider two modern schemes for two-phase reservior simulation: Fully-Implicit [20, 58] and IMPES (IMplicit Pressure Explicit Saturation) [58, 44] models.

The Multiscale Preconditioner idea is described in Section 5.1, it is applied to the Fully-Implicit Model in Section 5.2, it is applied to the IMPES Model in Section 5.3, and numerical examples are given in Section 5.4.

5.1 MULTISCALE PRECONDITIONER

Denote by $N_{\text{dof}}(i)$ the number of mortar degrees of freedom associated with the mortar space $M_{H,i}$ on subdomain D_i, and by N_{dof} the number of global mortar degrees of freedom in M_H. For $m = 1, \ldots, M$, consider a sequence of variational interface problems with permeabilities $K^{\{m\}}$ to be solved with the MMMFEM framework

$$b_H^{\{m\}}(\lambda_H, \mu) = g_H^{\{m\}}(\mu), \quad \forall \mu \in M_H. \tag{5.1}$$

For example, take those associated with stochastic realizations in an uncertainty quantification problem, or those associated with an Inexact Newton-Krylov iteration in an explicit or implicit time stepping scheme in a deterministic two-phase flow problem. Note that the former sequence can be solved in an arbitrary order, while the latter cannot.

For $m = 1, \ldots, M$, define a real $N_{\text{dof}} \times N_{\text{dof}}$ matrix $B_H^{\{m\}}$ satisfying

$$\left[B_H^{\{m\}}\boldsymbol{\lambda}, \boldsymbol{\mu}\right] = b_H^{\{m\}}(\lambda_H, \mu), \quad \forall \lambda, \mu \in M_H, \tag{5.2}$$

where $[\cdot, \cdot]$ is the Euclidean scalar product in $\mathbb{R}^{N_{\text{dof}}}$, and $\boldsymbol{\mu}$ denotes a vector of values for each $\lambda_H \in M_H$ at N_{dof} degrees of freedom. Recall that $B_H^{\{m\}} = \sum_{i=1}^{N_D} B_{H,i}^{\{m\}}$. When the permeabilities are uniformly positive definite, there exist positive constants $\hat{c}^{\{m\}}, \hat{C}^{\{m\}}, \alpha_i$ such that

$$\hat{c}^{\{m\}}\alpha_i \xi^T \xi \leq \xi^T K^{\{m\}}(\mathbf{x})\xi \leq \hat{C}^{\{m\}}\alpha_i \xi^T \xi, \quad \forall \xi \in \mathbb{R}^d, \ \mathbf{x} \in D. \tag{5.3}$$

Moreover as shown in [93, 26, 69], there exist positive constants $c^{\{m\}}$ and $C^{\{m\}}$ such that

$$c^{\{m\}} \sum_{i=1}^{N_D} \alpha_i |\mathcal{I}^{\partial D_i} \mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i} \leq \left[B_H^{\{m\}}\boldsymbol{\mu}, \boldsymbol{\mu}\right] \leq C^{\{m\}} \sum_{i=1}^{N_D} \alpha_i |\mathcal{I}^{\partial D_i} \mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i}, \tag{5.4}$$

for all $\mu \in M_H$, where $\mathcal{I}^{\partial D_i}$ is a continuous piecewise linear interpolant. The constants $c^{\{m\}}$ and $C^{\{m\}}$ do not depend on h or H, and depend only mildly on $K^{\{m\}}$.

Let $\overline{K}$ denote a uniformly positive definite training permeability that captures the main characteristics of the media. Recall Algorithm 3.2 for the generation of a multiscale basis and Algorithm 3.3 for the action of operators $B_{H,i}$ as linear combinations of multiscale basis functions. To avoid recomputing a new multiscale basis for every m, each subdomain D_i constructs a single multiscale basis $\{\overline{\psi}_{H,i}^{\{k\}}\}_{k=1}^{N_{\text{dof}}(i)}$ for the training operator $\overline{B}_{H,i}$. Then for $m = 1, \ldots, M$, we solve the preconditioned system

$$\overline{B}_H^{-1} B_H^{\{m\}} \lambda_H^{\{m\}} = \overline{B}_H^{-1} g_H^{\{m\}} \tag{5.5}$$

using CG in the symmetric case or GMRES in the non-symmetric case. In this approach, the action of $B_H^{\{m\}}$ is computed by solving subdomain problems with permeability $K^{\{m\}}$, while the action of $\overline{B}_H^{-1}$ is computed using the multiscale basis.

The following result has been shown in [85], which shows that the condition number of (5.5) is independent of h, H, and N_D. However, it does depend on how close the training operator is to each permeability.

Theorem 5.1.1. *Assume that Assumption 3.1.1 holds and that $\overline{B}_H$ satisfies a bound similar to (5.4) with constants $\overline{c}$ and $\overline{C}$. Then $\overline{B}_H$ and $B_H^{\{m\}}$ are uniformly spectrally equivalent,* i.e. *for $m = 1, \ldots, M$,*

$$cond(\overline{B}_H^{-1} B_H^{\{m\}}) \leq \frac{\overline{C} C^{\{m\}}}{\overline{c} c^{\{m\}}}.$$

The preconditioner $\overline{B}_H^{-1}$ is not explicitly constructed. The cost of applying the preconditioner is the need to solve secondary interface problem. Another iterative algorithm is used, so that the action of the training operator is a linear combination on every subdomain. We note that although this preconditioner bounds the number of subdomain solves for each permeability, the cost of applying the preconditioner itself may grow as the number of subdomains increases.

5.2 FULLY-IMPLICIT MODEL

In the Fully-Implicit Model (see [20], the Simultaneous Solution method), the coupled nonlinear two-phase system (1.5)–(1.8) is solved for all variables simultaneously with an implicit time-stepping scheme. Two of the unknowns are immediately eliminated using (1.9) and (1.10). A common practice is to choose the primary variables oil pressure $P = p_o$ and water saturation $S = S_w$. Using these constraints, the system can be rewritten as

$$\nabla \cdot \left(\frac{\rho_w K k_w}{\mu_w} \left(\nabla p_o - \frac{dp_c}{dS_w} \nabla S_w - \rho_w g \nabla z \right) \right) = \phi \frac{\partial (\rho_w S_w)}{\partial t} - q_w \tag{5.6}$$

$$\nabla \cdot \left(\frac{\rho_o K k_o}{\mu_w} \left(\nabla p_o - \rho_o g \nabla z \right) \right) = \phi \frac{\partial (\rho_o (1 - S_w))}{\partial t} - q_o. \tag{5.7}$$

Next carry out time differentiation, divide (5.6) by ρ_w, divide (5.7) by ρ_o, and add the equations to obtain

$$\begin{aligned} &\frac{1}{\rho_w} \nabla \cdot \left(\frac{\rho_w K k_w}{\mu_w} \left(\nabla p_o - \frac{dp_c}{dS_w} \nabla S_w - \rho_w g \nabla z \right) \right) + \\ &\qquad \frac{1}{\rho_o} \nabla \cdot \left(\frac{\rho_o K k_o}{\mu_w} \left(\nabla p_o - \rho_o g \nabla z \right) \right) \\ &\qquad = \frac{S_w}{\rho_w} \phi \frac{\partial \rho_w}{\partial t} + \frac{1 - S_w}{\rho_o} \phi \frac{\partial \rho_o}{\partial t} - \frac{q_w}{\rho_w} - \frac{q_w}{\rho_o}. \end{aligned} \tag{5.8}$$

Let $0 = t_0 < t_1 < \ldots < t_N = T$ be a partition of time interval $J = [0, T]$. Following [94], the Backward Euler multiblock expanded mixed finite element method is: for subdomains $1 \leq i < j \leq N_D$ and time steps $n = 1, \ldots, N$, we seek $\mathbf{V}^n_{h,\alpha}|_{D_i} \in \mathbf{V}_{h_i}$, $\tilde{\mathbf{V}}^n_{h,\alpha}|_{D_i} \in \tilde{\mathbf{V}}_{h_i}$, $P^n_{h,\alpha}|_{D_i} \in W_{h_i}$, $S^n_{h,\alpha}|_{D_i} \in W_{h_i}$, and $P^{M,n}_{h,\alpha}|_{\Gamma_{ij}} \in M_{H,i,j}$ such that, for $\alpha = o$ and w,

$$\int_{D_i} \frac{(\phi\rho_{h,\alpha}S_{h,\alpha})^n - (\phi\rho_{h,\alpha}S_{h,\alpha})^{n-1}}{\Delta t^n} w\, dx + \int_{D_i} \nabla \cdot \mathbf{V}^n_{h,\alpha}\, w\, dx = \int_{D_i} q_\alpha\, w\, dx,\ w \in W_{h_i}, \tag{5.9}$$

$$\int_{D_i} \tilde{\mathbf{V}}^n_{h,\alpha} \cdot \mathbf{v}\, dx = \int_{D_i} P^n_{h,\alpha} \nabla \cdot \mathbf{v}\, dx - \int_{\partial D_i \setminus \partial D} P^{M,n}_{h,\alpha} \mathbf{v} \cdot \nu_i\, d\sigma,\ \mathbf{v} \in \mathbf{V}_{h_i}, \tag{5.10}$$

$$\int_{D_i} \mathbf{V}^n_{h,\alpha} \cdot \tilde{\mathbf{v}}\, dx = \int_{D_i} \frac{k^n_{h,\alpha} K}{\mu_{h,\alpha}} \rho^n_{h,\alpha} (\tilde{\mathbf{V}}^n_{h,\alpha} + \rho^n_{h,\alpha} g \nabla D) \cdot \tilde{\mathbf{v}}\, dx,\ \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{h_i}, \tag{5.11}$$

$$\int_{\Gamma_{ij}} [\mathbf{V}^n_{h,\alpha} \cdot \nu]_{ij}\, \mu\, d\sigma = 0,\ \mu \in M_{H,i,j}. \tag{5.12}$$

Here $k^n_{h,\alpha}$ and $\rho^n_{h,\alpha} \in W_{h,i}$ are given functions of the primary subdomain variables P^n_h and S^n_h. The mortar functions $P^{M,n}_{H,\alpha}$ can be computed using (1.9) and (1.10), given the primary mortar variables $P^{M,n}_H$ and $S^{M,n}_H$.

Using the substructuring domain decomposition algorithm, the discrete system (5.9)–(5.12) is reduced to an interface problem in the coarse mortar space. Let $\mathcal{M}_H = M_H \times M_H$ be the product space of mortar primary variables. We define the non-linear interface bivariate form $b^n : \mathcal{M}_H \times \mathcal{M}_H \to \mathbb{R}$ at time $t = t_n$ as follows. For $\psi = (P^{M,n}_H, S^{M,n}_H) \in \mathcal{M}_H$ and $\mu = (\mu_w, \mu_n) \in \mathcal{M}_H$, define

$$b^n(\psi, \mu) = \sum_{1 \leq i < j \leq N_D} \int_{\Gamma_{i,j}} \left([\mathbf{V}^n_{h,w} \cdot \mathbf{n}]\mu_w + [\mathbf{V}^n_{h,n} \cdot \mathbf{n}]\mu_n \right) d\sigma, \tag{5.13}$$

where $(S^n_h(\psi), \mathbf{V}^n_{h,\alpha}(\psi))$ are solutions to the time series of subdomain problems (5.9)–(5.12) with Dirichlet boundary data $P^{M,n}_{H,\alpha}$. Using this definition, also define the non-linear interface operator $B^n : \mathcal{M}_H \to \mathcal{M}_H$ by

$$\langle B^n \psi, \mu \rangle_{\mathcal{M}_H} = b^n(\psi, \mu), \quad \forall \mu \in \mathcal{M}_H.$$

It can be shown that $(\psi, S^n_H(\psi), \mathbf{V}^N_{h,\alpha}(\psi))$ is the solution to (5.9)–(5.12), where $\psi \in \mathcal{M}_H$ solves

$$B^n(\psi) = 0. \tag{5.14}$$

The system of nonlinear interface equations in (5.14) is solved with an Inexact Newton method, in which each Newton step

$$(B^n)'(\psi)s = -B^n(\psi) \tag{5.15}$$

is computed by a forward difference GMRES iteration. The action of the Jacobian is approximated by the forward difference for some $\delta > 0$,

$$D_\delta\ b^n(\psi,\mu) = \begin{cases} 0, & \mu = 0 \\ \|\mu\| \frac{\mathcal{B}(\psi+\delta\mu/\|\mu\|)-\mathcal{B}(\psi)}{\delta\|\psi\|}, & \mu \neq 0, \psi \neq 0 \\ \|\mu\| \frac{\mathcal{B}(\delta\mu/\|\mu\|)-\mathcal{B}(\psi)}{\delta}, & \mu \neq 0, \psi = 0 \end{cases} \tag{5.16}$$

requiring one evaluation of B^n on each Interface-GMRES iteration. This, in turn, requires the solution of block (subdomain) problems in parallel, which are also non-linear and are solved with an Inexact Newton-GMRES method.

Let $\{\hat{e}_j\}$ be the standard basis for the product space of primary mortar variables $\mathcal{M}_H$. This space has dimension N_{dof}^2. In order to form a multiscale basis for the Interface Newton-GMRES problem (5.15), one should compute subdomain problems with $\mu = \hat{e}_j$, for $j = 1, \ldots, N_{\text{dof}}^2$ in (5.16).

5.3 IMPES MODEL

In the IMPES scheme [20, 58], the coupled two-phase system with constraints (1.5)–(1.10) is split into a pressure equation that will be solved implicitly, and a saturation equation that will be solved explicitly, with respect to time discretization. The primary variables will be oil pressure $p = p_o$ and water saturation $S = S_w$. First we make the following definitions: the phase mobility functions are

$$\lambda_\alpha(\mathbf{x}, S_\alpha) = \frac{k_\alpha}{\mu_\alpha}, \quad \alpha = o, w, \tag{5.17}$$

the total mobility is

$$\lambda_{total}(\mathbf{x}, S) = \lambda_w + \lambda_o, \tag{5.18}$$

the fractional flow functions are

$$f_\alpha(\mathbf{x}, S) = \frac{\lambda_\alpha}{\lambda_{total}}, \quad \alpha = o, w. \tag{5.19}$$

and the total velocity is

$$\mathbf{u}_{total} = \mathbf{u}_w + \mathbf{u}_o. \tag{5.20}$$

For simplicity in the presentation, we assume incompressibility for both fluids, although the results will be slightly compressible. First add equations (1.6) for $\alpha = o$ and $\alpha = w$, and use (5.20) and (1.9) to obtain

$$\nabla \cdot \mathbf{u}_{total} = q(p, S) \equiv q_w(p, S) + q_o(p, S). \tag{5.21}$$

Next add equation (1.5) for $\alpha = o$ with $\alpha = w$, and use (1.10) and (5.20) to obtain

$$\mathbf{u}_{total} = -K\Big(\lambda_{total}(s)\nabla p - \lambda_w(S)\nabla p_c - (\lambda_w\rho_w + \lambda_o\rho_o)g\nabla z\Big). \tag{5.22}$$

The pressure equation is derived by substituting (5.22) into (5.21), which gives

$$-\nabla \cdot (K\lambda_{total}\nabla p) = q - \nabla \cdot \Big(K(\lambda_w\nabla p_c + (\lambda_w\rho_w + \lambda_o\rho_o)g\nabla z\Big). \tag{5.23}$$

The saturation equation is derived by applying (1.10), (5.20), and (5.22) to equations (1.5) and (1.6) with $\alpha = w$, which gives

$$\phi\frac{\partial S}{\partial t} = q_w(p, S) - \nabla \cdot \left[Kf_w(S)\lambda_o(S)\Big(\frac{dp_c}{dS}\nabla S + (\rho_o - \rho_w)g\nabla z\Big) + f_w(S)\mathbf{u}_{total}\right]. \tag{5.24}$$

In the standard IMPES method, the saturation S is known, and the pressure equation (5.23) is solved implicitly for p. This means for time steps $n = 0, 1, \ldots, N$ the pressure p satisfies the elliptic problem

$$-\nabla \cdot (K\lambda_{total}(S^n)\nabla p^n) = F(p^n, S^n), \tag{5.25}$$

where $F(p, S)$ is the RHS of (5.23) and S^n is given. (We note in the slightly compressible case, this will be a degenerate parabolic equation). The saturation equation (5.24) is solved

explicitly for S. This means for $n = 0, 1, \ldots, N-1$ the saturation S^{n+1} satisfies the parabolic problem

$$\phi \frac{\partial S^{n+1}}{\partial t} = G(p^n, \mathbf{u}^n, S^n), \tag{5.26}$$

where $G(p, \mathbf{u}, S)$ is the RHS of (5.24).

The multiblock formulation and interface iteration for the IMPES model are performed similarly to the derivation in Section 5.2. The difference here is that a mortar space M_H is constructed only for the implicit pressure solve. Therefore this model has a non-linear interface bivariate form $b^n : M_H \times M_H$, which is defined by

$$b^n(\psi, \mu) = \sum_{1 \leq i < j \leq N_D} \int_{\Gamma_{i,j}} [\mathbf{V}^n_{h,w} \cdot \mathbf{n}] \mu d\sigma. \tag{5.27}$$

The interface operator and Inexact Newton-GMRES iteration are formulated using this definition.

Let $\{\hat{e}_j\}$ be the standard basis for the mortar space M_H. This space has dimension N_{dof}. In order to form a multiscale basis for the Interface Newton-GMRES problem with the IMPES model, one should compute subdomain problems with $\mu = \hat{e}_j$, for $j = 1, \ldots, N_{\text{dof}}$ in (5.16).

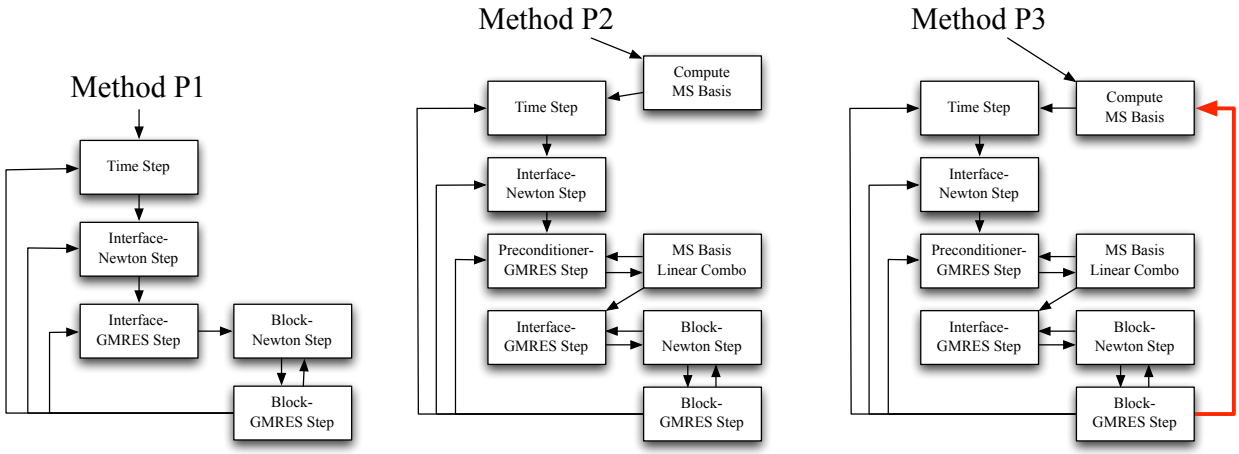
5.4 NUMERICAL EXAMPLES

We seek to make a comparison of both the number of Interface-GMRES iterations per time step and the total simulation runtime for the following three methods:

- **METHOD P1** - No Preconditioner.
- **METHOD P2** - A Single Multiscale Basis Preconditioner, computed at $t = 0$.
- **METHOD P3** - Multiscale Basis Preconditioner with re-calculation, triggered after 15 Interface-GMRES iterations are reached on a single time step.

To show the steps that each of these methods require, flow charts are given for the Fully-Implicit Model in Figure 5.1. These methods will be tested for a deterministic, slightly compressible, two-phase flow problem, with both the Fully-Implicit and IMPES numerical models.

Figure 5.1: Flow charts for Methods P1, P2, and P3 in the Fully-Implicit Model. Right columns include steps that may be parallelized.



To perform these tests, the multiscale basis implementation and preconditioner were programmed into the IPARS reservoir simulator, in both Fully-Implicit and IMPES models. The runtimes were recorded by compiling the code with optimization using Intel's ifort compiler and MKL library, and run in serial on an Intel Core 2 Quad CPU Q9650 3.0GHz system.

5.4.1 Fully-Implicit Example, SPE6 Permeability

The numerical example consists of a simulation of oil-water immiscible displacement in a 3-D slice of a highly heterogeneous reservoir. Permeability data from the SPE Comparative Solution Project [22] was used for the diagonal tensor K, shown in Figure 5.2. Note in this simulation, the x-coordinate represents the depth. The domain $D = (0, 25) \times (0, 420) \times (0, 420)$ [ft] is divided into $1 \times 2 \times 2$ equal blocks (subdomains), each having fine grids of $6 \times 8 \times 8$ finite elements. The four block interfaces are made up of continuous linear mortars with 3×4 elements. All boundary conditions are no-flow. Both fluids are assumed to be

Figure 5.2: X-component of SPE6 Permeability data in millidarcies for Example 5.4.1.

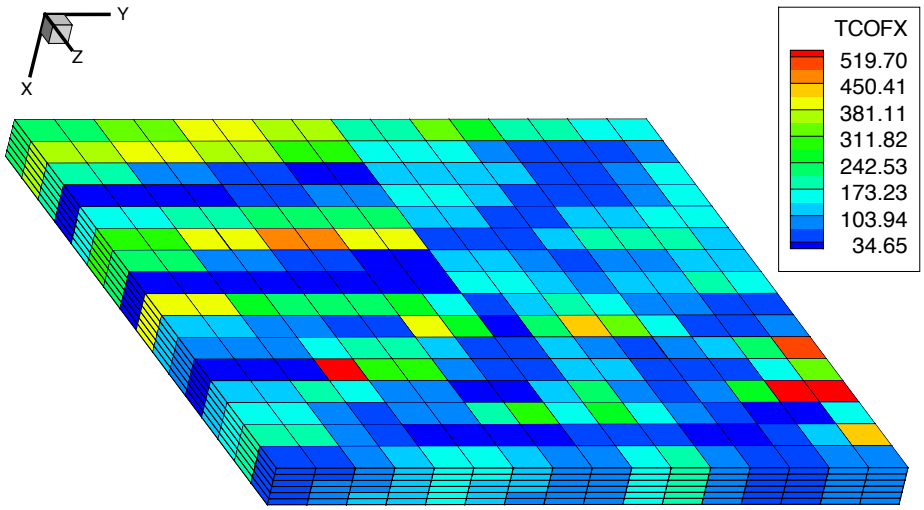


Figure 5.3: Oil relative permeability (left), water relative permeability (middle), and capillary pressure (right) versus water saturation for Example 5.4.1.

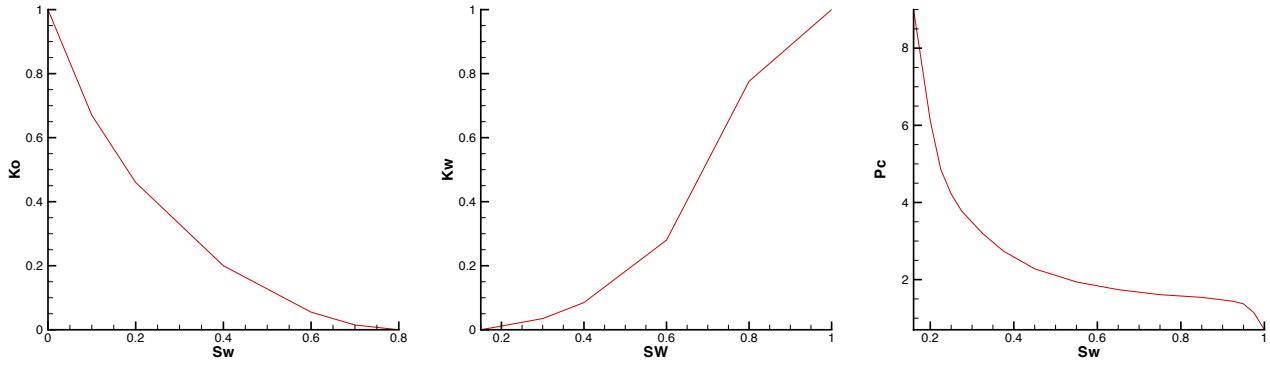
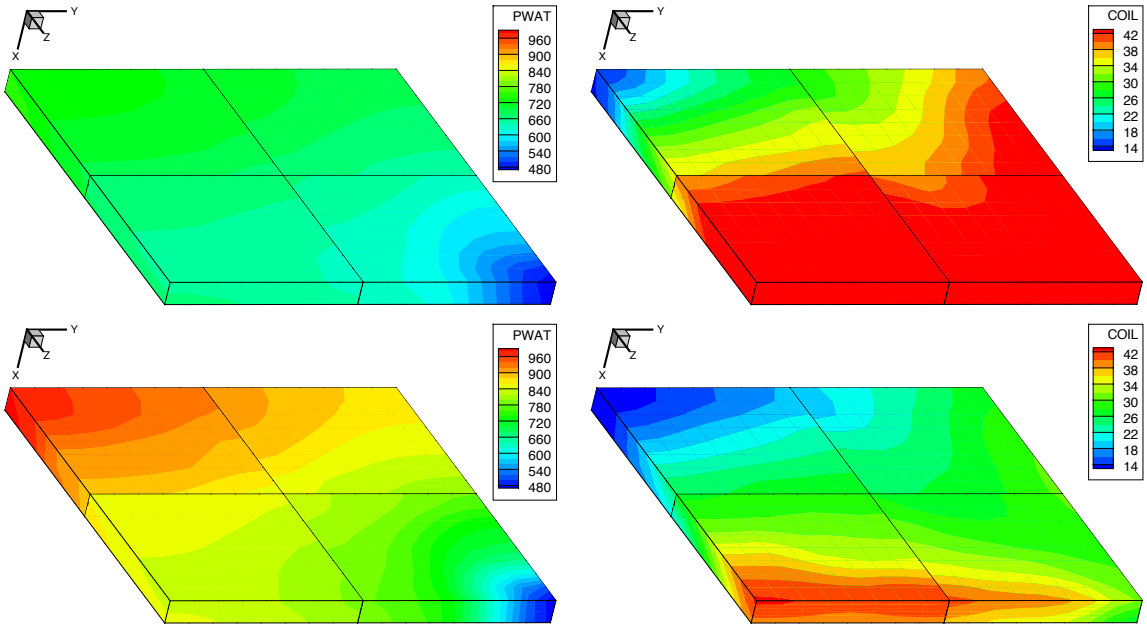


Figure 5.4: Plots for Example 5.4.1: Water Pres. at $t = 100$ (top-left), Oil Conc. at $t = 100$ (top-right), Water Pres. at $t = 200$ (bottom-left), Oil Conc. at $t = 200$ (bottom-right).



slightly compressible, with exponential coefficients c_o = 4.0E-5 and c_w = 3.3E-6 [1/psi] in the equations of state (1.11). Initial oil pressure is set at $p_{o,0} = 500$ [psi], and initial water saturation is $S_{w,0} = 0.22$. Water viscocity is $\mu_w = 0.5$ [cp] and oil viscocity is $\mu_o = 2.0$ [cp]. Relative permeabilities and capillary pressure are shown in Figure 5.3.

From an aerial perspective, water is injected at a well in upper-left corner, and oil is produced at a well in the lower-right corner, for a quarter five spot test. The length of the simulation is $T = 200$ [days], with a constant time step $\Delta t = 0.1$. Water injection bottom hole pressure (BHP) starts 505 [psi] and increases linearly until it reaches 1000 [psi] at the final time. Oil production BHP starts at 480 [psi] and decreases linearly until it reaches 350 [psi] at the final time.

The Fully-Implicit model was first used to test Methods P1, P2, and P3. Figure 5.4 shows the results of the computation at times $t = 100$ and $t = 200$. As water pressure increases in the injection well, water pressure continually increases in the domain and mass enters the domain at an increasing rate. This causes the water saturation front to sweep

across the domain, and oil concentration decreases accordingly as this phase is displaced.

The model tolerances for the convergences of each step are: 1E-4 for Interface-GMRES, 1E-5 for Interface-Newton, 1E-6 for Jacobian Forward Difference, 1E-7 for Block-Newton, 1E-14 for Block-GMRES, and 1E-3 for Preconditioner-GMRES.

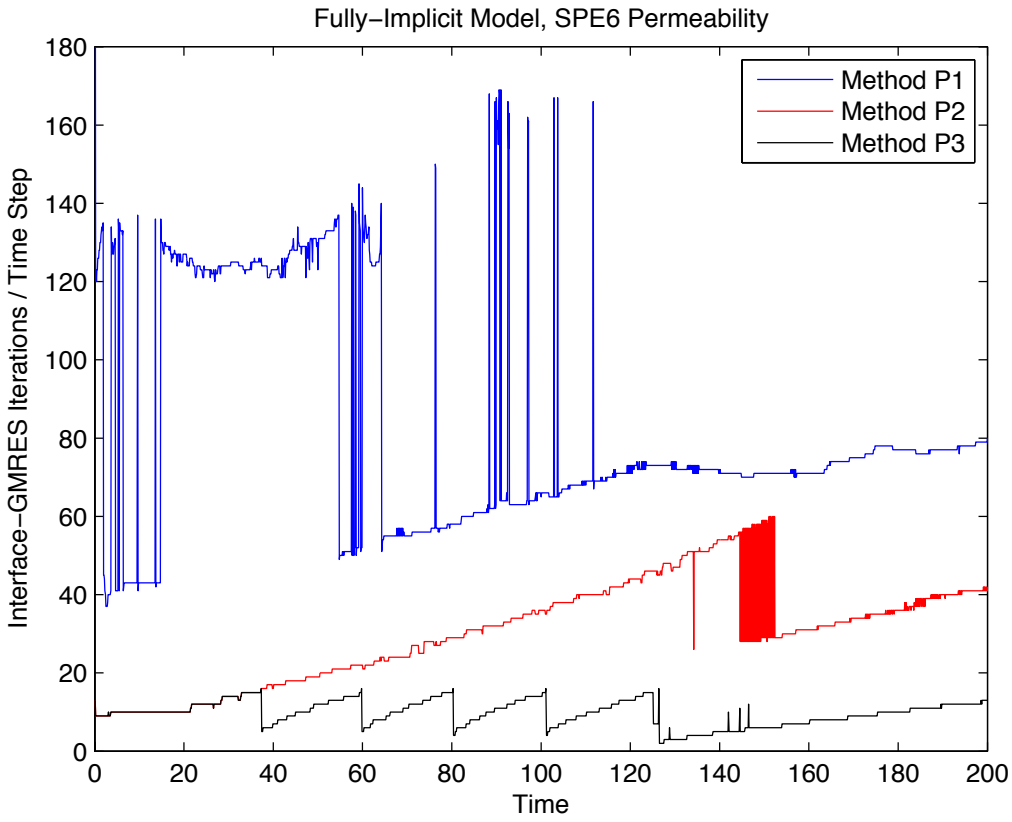
The computational cost for Methods P1, P2, and P3 are shown in Table 5.1. The time averaged number of Interface-GMRES iterations was successfully decreased by 63% with a single preconditioner, and decreased by 88% when the multiscale basis was recomputed after a tolerance of 15 Interface-GMRES iterations was reached on a single time step. Figure 5.5 shows how the interface iteration becomes more difficult in time, and how the use of the multiscale preconditioner with recomputation can help to bound the condition number.

Note that there is an overhead cost of Block-Interface problems to solve each time a multiscale basis is recomputed, which is not reflected in these numbers. However this cost is indeed reflected in the overall runtime, which was decreased from approximately 5 1/2 hours with Method P1 to just over 1 hour with Method P3. This makes a very compelling argument for the utility of the multiscale basis approach at increasing tractability for two-phase nonlinear MMMFEM interface problems with the fully-implicit model.

Table 5.1: Computational cost of Example 5.4.1 with Methods P1, P2, and P3 in the Fully-Implicit Model.

	METHOD P1	**METHOD P2**	**METHOD P3**
Avg. Iter. / Time Step	82.8	30.2	9.7
Runtime in Minutes	322.2	184.2	67.3

Figure 5.5: Interface-GMRES iterations versus time for Example 5.4.1 with Methods P1, P2, and P3 in the Fully-Implicit Model.



5.4.2 IMPES Example, Constant Permeability

The second numerical example uses the IMPES model for oil-water immiscible displacement. The domain and discretization are the same as the previous test. Instead of the heterogeneous diagonal tensor, the permeability in this test is a constant $K = 100$ [md]. All other physical quantities remain the same as the previous test, including well data.

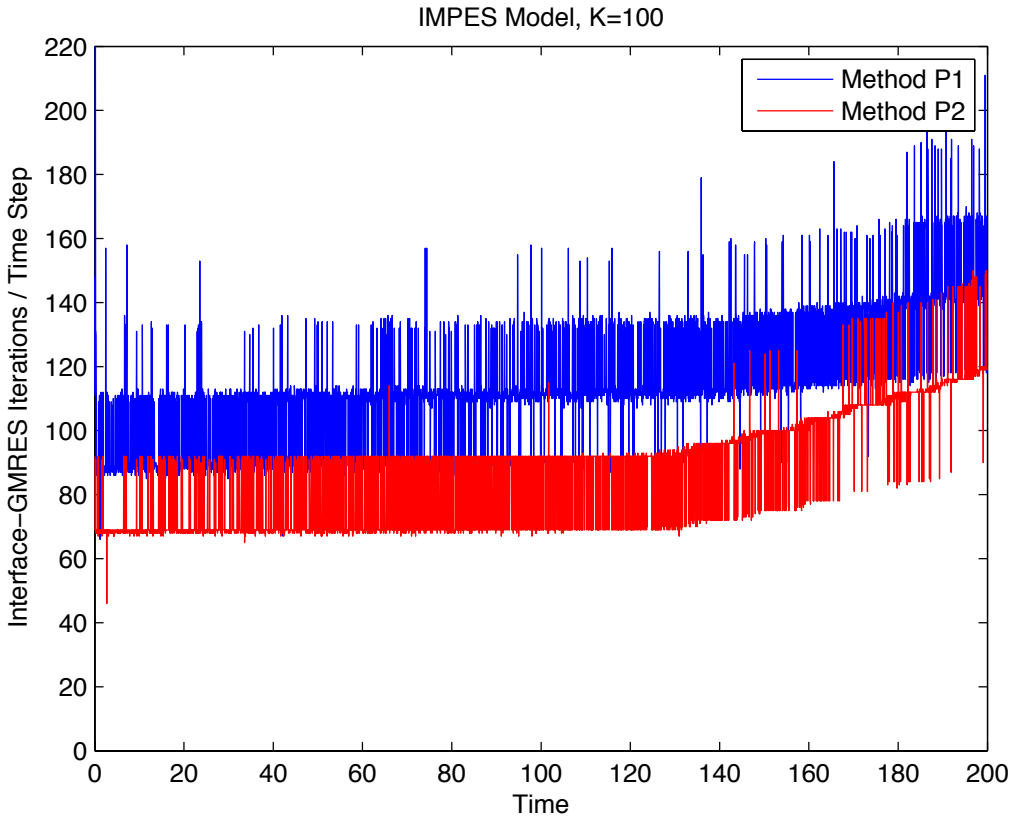
The implicit pressure time step is halved from the previous test at a constant $\Delta t = 0.05$. For further stability, the explicit saturation step is further partitioned, taking 5 saturation steps for each pressure step. The model tolerances for the convergences of each step are: 1E-10 for Interface-GMRES, 1E-4 for Interface-Newton, 1E-5 for Pressure Jacobian Forward Difference, 1E-6 for Saturation Jacobian Forward Difference, 1E-7 for Block-Newton, 1E-14 for Block-GMRES, and 1E-14 for Preconditioner-GMRES.

Currently, the multiscale preconditioner is not as effective at reducing the computational cost of the IMPES model, as it was in the Fully-Implicit model. In Table 5.2, the average number of Interface-GMRES iterations per time step was indeed reduced by 25% by Method P2. However, since the Block-Interface problem is much easier to solve than in the previous test, the runtime actually became worse by 63%. We believe the effectiveness of the preconditioner will improve as the Block-Problems become more difficult. Figure 5.6 shows how the difficulty of interface iteration is relatively flat in time for the IMPES model. We attribute this to the total mobility (5.18) changing relatively slowly in time. As such, Method P3 is not shown because there would be little benefit to periodic recomputation of a multiscale basis.

Table 5.2: Computational cost of Example 5.4.2 with Methods P1 and P2 in the IMPES Model.

	METHOD P1	**METHOD P2**
Avg. Iter. / Time Step	115.6	86.7
Runtime in Minutes	116.2	189.1

Figure 5.6: Interface-GMRES iterations versus time for Example 5.4.2 with Methods P1 and P2 in the IMPES Model.



BIBLIOGRAPHY

[1] J.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 Model. and Simul.*, 2(3):421–439, 2004.

[2] J.E. Aarnes and Y. Efendiev. Mixed multiscale finite element methods for stochastic porous media flows. *SIAM J. Sci. Comput.*, 30(5):2319–2339, 2008.

[3] J.E. Aarnes, Y. Efendiev, and L. Jiang. Mixed multiscale finite element methods using limited global information. *Multiscale Model. Simul.*, 7(2):655–676, 2008.

[4] J.E. Aarnes, S. Krogstad, and K.-A. Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Modeling and Simulation*, 5(2):337–363, 2007.

[5] M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,*, chapter Orthogonal Polynomials (N. 22), pages 771–802. Dover, 9th printing edition, 1972.

[6] T. Arbogast. Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems. *SIAM J. Numer. Anal.*, 42:576–598, 2004.

[7] T. Arbogast and K.J. Boyd. Subgrid upscaling and mixed multiscale finite elements. *SIAM Journal on Numerical Analysis*, 44(3):1150–1171, 2007.

[8] T. Arbogast, L.C. Cowsar, M.F. Wheeler, and I. Yotov. Mixed finite element methods on nonmatching multiblock grids. *SIAM J. Numer. Anal.*, 37:1295–1315, 2000.

[9] T. Arbogast, G. Pencheva, M.F. Wheeler, and I. Yotov. A Multiscale Mortar Mixed Finite Element Method. *Multiscale Modeling and Simulation*, 6(1):319, 2007.

[10] B.V. Asokan and N. Zabaras. A stochastic variational multiscale method for diffusion in heterogeneous random media. *Journal of Computational Physics*, 218(2):654–676, 2006.

[11] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[12] I. Babuška, R. Tempone, and G.E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.

[13] J. Bear. *Dynamics of fluids in porous media*. Dover Publications, 1972.

[14] D. Boffi, F. Brezzi, L.F. Demkowicz, R.G. Durán, R.S. Falk, and M. Fortin. *Mixed finite elements, compatibility conditions, and applications*. Springer-Verlag Berlin Heidelberg, 2008.

[15] F. Brezzi, J. Douglas, R. Durán, and M. Fortin. Mixed finite elements for second order elliptic problems in three variables. *Numerische Mathematik*, 51(2):237–250, 1987.

[16] F. Brezzi, J. Douglas, M. Fortin, and L.D. Marini. Efficient rectangular mixed finite elements in two and three space variables. *Modélisation mathématique et analyse numérique(Print)*, 21(4):581–604, 1987.

[17] F. Brezzi, J. Douglas, and L.D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985.

[18] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag, 1991.

[19] Z. Chen and J. Douglas. Prismatic mixed finite elements for second order elliptic problems. *Calcolo*, 26(2):135–148, 1989.

[20] Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, 72:541–576, 2003.

[21] Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media*. Society for Industrial Mathematics, 2006.

[22] M.A. Christie and M.J. Blunt. Tenth SPE comparative solution project: a comparison of upscaling techniques. *SPE Reservoir Evaluation & Engineering*, 4(4):308–317, 2001. http://www.spe.org/web/csp/.

[23] P.G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, 1978.

[24] P.G. Constantine, A. Doostan, and G. Iaccarino. A hybrid collocation/Galerkin scheme for convective heat transfer problems with stochastic boundary conditions. *International Journal for Numerical Methods in Engineering*, 80, 2009.

[25] R. Courant and D. Hilbert. *Methods of mathematical physics. Vol. 1*. Wiley, 1953.

[26] L.C. Cowsar, J. Mandel, and M.F. Wheeler. Balancing domain decomposition for mixed finite elements. *Math. Comp.*, 64:989–1015, 1995.

[27] L.C. Cowsar, C. Woodward, and I. Yotov. PARCEL v1.04 user guide. Technical Report 96-28, TICAM, The University of Texas at Austin, 1996.

[28] H. Darcy. *Les fontaines publiques de la ville de Dijon (The water supply of Dijon).* Victor Dalmont, Paris, 1856.

[29] M.K. Deb, I.M. Babuška, and J.T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.

[30] P. Dostert, Y. Efendiev, and T.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(43-44):3445–3455, 2008.

[31] Y.R. Efendiev, T.Y. Hou, and X.H. Wu. Convergence of a nonconforming multiscale finite element method. *SIAM J. Numer. Anal.*, 37(3):888–910 (electronic), 2000.

[32] M.S. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. *AIAA,(2009-0976)*, 2009.

[33] G.S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer, Berlin, 1996.

[34] G.P. Galdi. *An introduction to the mathematical theory of the Navier-Stokes equations. Vol. I. Linearized steady problems.* Springer-Verlag, New York, 1994.

[35] B. Ganapathysubramanian and N. Zabaras. Modeling diffusion in random heterogeneous media: Data-driven models, stochastic collocation and the variational multiscale method. *Journal of Computational Physics*, 226(1):326–353, 2007.

[36] B. Ganis, H. Klie, M.F. Wheeler, T. Wildey, I. Yotov, and D. Zhang. Stochastic collocation and mixed finite elements for flow in porous media. *Computer Methods in Applied Mechanics and Engineering*, 197(43-44):3547–3559, 2008.

[37] B. Ganis, G. Pencheva, S. Thomas, M.F. Wheeler, T. Wildey, and I. Yotov. A multiscale flux basis preconditioner for multiphase and multiphysics flow and transport in porous media. In preparation, 2010.

[38] B. Ganis and I. Yotov. Implementation of a Mortar Mixed Finite Element Method using a Multiscale Flux Basis. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3989–3998, 2009.

[39] B. Ganis, I. Yotov, and M. Zhong. A Stochastic Mortar Mixed Finite Element Method for Flow in Porous Media with Multiple Rock Types. Submitted, 2010.

[40] R.G. Ghanem. Scales of fluctuation and the propagation of uncertainty in random porous media. *Water Resources Research*, 34(9):2123–2136, 1998.

[41] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, New York, 1991.

[42] R. Glowinski and M.F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA*, 1988.

[43] P. Grisvard. Elliptic problems in nonsmooth domains. *Monographs and studies in Mathematics*, 24, 1985.

[44] H. Hoteit and A. Firoozabadi. Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures. *Advances in Water Resources*, 31(1):56–73, 2008.

[45] T.Y. Hou, W. Luo, B. Rozovskii, and H.M. Zhou. Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *Journal of Computational Physics*, 216(2):687–706, 2006.

[46] T.Y. Hou and X.H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169–189, 1997.

[47] T.Y. Hou, X.H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comp.*, 68:913–943, 1999.

[48] T.J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput. Methods Appl. Mech. Engrg.*, 127:387–401, 1995.

[49] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.B. Quincy. The variational multiscale method–a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166(1-2):3–24, 1998.

[50] K.D. Jarman and T.F. Russell. Analysis of 1-D moment equations for immiscible flow. In *Fluid flow and transport in porous media, mathematical and numerical treatment: proceedings of an AMS-IMS-SIAM Joint Summer Research Conference on Fluid Flow and Transport in Porous Media, Mathematical and Numerical Treatment, June 17-21, 2001, Mount Holyoke College, South Hadley, Massachusetts*, page 293. AMS Bookstore, 2002.

[51] P. Jenny, S.H. Lee, and H.A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comp. Phys.*, 187:47–67, 2003.

[52] M. Kaminski and G.F. Carey. Stochastic perturbation-based finite element approach to fluid flow problems. *International Journal of Numerical Methods for Heat and Fluid Flow*, 15(7):671, 2005.

[53] M.Y. Kim, F.A. Milner, and E.J. Park. Some observations on mixed methods for fully nonlinear parabolic problems in divergence form. *Applied Mathematics Letters*, 9(1):75–82, 1996.

[54] M.Y. Kim, E.J. Park, S.G. Thomas, and M.F. Wheeler. A multiscale mortar mixed finite element method for slightly compressible flows in porous media. *J. Korean Math. Soc*, 44(5):1103–1119, 2007.

[55] D.P. Landau and K. Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge University Press, 2005.

[56] H. Li and D. Zhang. Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resour. Res*, 43:44–48, 2007.

[57] M. Loève. *Probability Theory, Volume I*. Springer-Verlag, 1977.

[58] B. Lu. *Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media*. PhD thesis, The University of Texas at Austin, 2008.

[59] Z. Lu and D. Zhang. Conditional simulations of flow in randomly heterogeneous porous media using a KL-based moment-equation approach. *Advances in Water Resources*, 27(9):859–874, 2004.

[60] Z. Lu and D. Zhang. Stochastic simulations for flow in nonstationary randomly heterogeneous porous media using a KL-based moment-equation approach. *Multiscale Modeling and Simulation*, 6(1):228–245, 2008.

[61] X. Ma and N. Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics*, 228(8):3084–3113, 2009.

[62] J.C. Nedelec. Mixed finite elements in $\mathbf{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

[63] F. Nobile, R. Tempone, and C.G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[64] F. Nobile, R. Tempone, and C.G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[65] B. Oksendal. *Stochastic differential equations*. Springer Berlin, 1998.

[66] M. Parashar, J.A. Wheeler, J.C. Brown, G. Pope, K. Wang, and P. Wang. A New Generation EOS Compositional Reservoir Simulator: Part II - Framework and Multiprocessing. In *SPE Reservoir Simulation Symposium*. Houston, TX, SPE 37977, 1997.

[67] E.J. Park. Mixed finite element methods for generalized Forchheimer flow in porous media. *Numerical Methods for Partial Differential Equations*, 21(2):213–228, 2005.

[68] D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *Old SPE Journal*, 23(3):531–543, 1983.

[69] G. Pencheva and I. Yotov. Balancing domain decomposition for mortar mixed finite element methods. *Numer. Linear Algebra Appl*, 10:159–180, 2003.

[70] M. Peszyńska, M.F. Wheeler, and I. Yotov. Mortar upscaling for multiphase flow in porous media. *Computational Geosciences*, 6(1):73–100, 2002.

[71] A.M. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer, 1994.

[72] R.A. Raviart and J.M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics*, volume 606, pages 292–315. Springer-Verlag, New York, 1977.

[73] F. Riesz and B. Sz-Nagy. *Functional Analysis*. Dover, 1990.

[74] T.F. Russell and K.D. Jarman. Eulerian Moment Equations for 2-D Stochastic Immiscible Flow, 2003.

[75] C. Schwab and R.A. Todor. Sparse finite elements for elliptic problems with stochastic loading. *Numerische Mathematik*, 95(4):707–734, 2003.

[76] C. Schwab and R.A. Todor. Karhunen-Loeve approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):122, 2006.

[77] H.A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. Springer Berlin, 1890. First published in *Vierteljahrsschrift der Naturforschenden Gesellschaft* in Zürich, 15 (1870), pp. 272–286.

[78] L.R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990.

[79] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl. Akad. Nauk SSSR*, number 240-243 in 4, page 12, 1963.

[80] G. Stefanou. The stochastic finite element method: Past, present and future. *Computer Methods in Applied Mechanics and Engineering*, 198(9-12):1031–1051, 2009.

[81] X. Wan and G. E. Karniadakis. Beyond Wiener–Askey Expansions: Handling Arbitrary PDFs. *Journal of Scientific Computing*, 27(1):455–464, 2006.

[82] P. Wang, I. Yotov, M.F. Wheeler, T. Arbogast, C. Dawson, M. Parashar, and K. Sephernoori. A New Generation EOS Compositional Reservoir Simulator: Part I - Formulation and Discretization. In *SPE Reservoir Simulation Symposium*. Houston, TX, SPE 37979, 1997.

[83] C.G. Webster. *Sparse grid stochastic collocation techniques for the numerical solution of partial differential equations with random input data*. PhD thesis, The Florida State University, 2007.

[84] M.F. Wheeler, C. Dawson, and C. Celentano. *Mathematics of Finite Elements and Applications: Highlights 1996*, chapter Multicomponent, multiphase flow and transport in porous media, pages 223–224. John Wiley & Sons, 1997. J. R. Whiteman, editor.

[85] M.F. Wheeler, T. Wildey, and I. Yotov. A multiscale preconditioner for stochastic mortar mixed finite elements. Submitted, 2010.

[86] M.F. Wheeler and I. Yotov. Physical and computational domain decompositions for modeling subsurface flows. In Jan Mandel et al., editors, *Tenth International Conference on Domain Decomposition Methods, Contemporary Mathematics, vol 218*, pages 217–228. American Mathematical Society, 1998.

[87] M.F. Wheeler and I. Yotov. A posteriori error estimates for the mortar mixed finite element method. *SIAM J. Numer. Anal.*, 43(3):1021–1042, 2005.

[88] D. Xiu. Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys*, 2(2):293–309, 2007.

[89] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27:1118, 2005.

[90] D. Xiu and G.E. Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4927–4948, 2002.

[91] D. Xiu and D.M. Tartakovsky. A two-scale nonperturbative approach to uncertainty analysis of diffusion in random composites. *Multiscale Model. Simul*, 2(4):662–674, 2004.

[92] D. Xiu and D.M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM Journal on Scientific Computing*, 28(3):1167–1185, 2007.

[93] I. Yotov. *Mixed Finite Element Methods for Flow in Porous Media*. PhD thesis, Rice University, 1996.

[94] I. Yotov. Interface solvers and preconditioners of domain decomposition type for multiphase flow in multiblock porous media. *Advances in Computation: Theory and Practice*, 7:157–167, 2001.

[95] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*. Academic Press, San Diego, 2002.

[96] D. Zhang and Z. Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions. *Journal of Computational Physics*, 194(2):773–794, 2004.