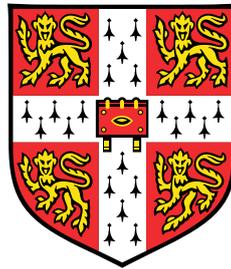


# Uncertainty quantification of engineering systems using the multilevel Monte Carlo method



**Helena Juliette Thomasin Unwin**

Supervised by Dr. Garth N. Wells

Department of Engineering,  
University of Cambridge

This dissertation is submitted for the degree of  
*Doctor of Philosophy*



## **Abstract**

This thesis examines the quantification of uncertainty in real-world engineering systems using the multilevel Monte Carlo method. It is often infeasible to use the traditional Monte Carlo method to investigate the impact of uncertainty because computationally it can be prohibitively expensive for complex systems. Therefore, the newer multilevel method is investigated and the cost of this method is analysed in the finite element framework.

The Monte Carlo and multilevel Monte Carlo methods are compared for two prototypical examples: structural vibrations and buoyancy driven flows through porous media. In the first example, the impact of random mass density is quantified for structural vibration problems in several dimensions using the multilevel Monte Carlo method. Comparable eigenvalues and energy density approximations are found for the traditional Monte Carlo method and the multilevel Monte Carlo method, but for certain problems the expectation and variance of the quantities of interest can be computed over 100 times faster using the multilevel Monte Carlo method. It is also tractable to use the multilevel method for three dimensional structures, where the traditional Monte Carlo method is often prohibitively expensive.

In the second example, the impact of uncertainty in buoyancy driven flows through porous media is quantified using the multilevel Monte Carlo method. Again, comparable results are obtained from the two methods for diffusion dominated flows and the multilevel method is orders of magnitude cheaper. The finite element models for this investigation are formulated carefully to ensure that spurious numerical artefacts are not added to the solution and are compared to an analytical model describing the long term sequestration of CO<sub>2</sub> in the presence of a background flow.

Additional cost reductions are achieved by solving the individual independent samples in parallel using the new podS library. This library schedules the Monte Carlo and multilevel Monte Carlo methods in parallel across different computer architectures for the two examples considered in this thesis. Nearly linear cost reductions are obtained as the number of processes is increased.



## **Declaration**

I hereby declare that except where specific reference is made to the work of others, the content of this thesis is original and has not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other, university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This thesis contains fewer than 33 500 words including the appendices and bibliography, and has 223 equations, 11 tables and 57 figures.

Helena Juliette Thomasin Unwin

May 2018



## **Acknowledgements**

On completing this thesis, it is obvious to me that it would not have been possible without the help and support of numerous people. In particular, I would like to thank my supervisor Dr. Garth Wells for his advice and direction during my PhD. I would also like to thank Prof. Andy Woods for his insights into porous media flows and for his collaboration on the analytical model in Chapter 6 and Dr. Nate Sime for his help identifying the eigenvalue solvers in Chapter 4 and for always being on hand to answer my questions. I am grateful to Dr. Chris Richardson for his help managing the intricacies of FEniCS and Hugo Hadfield for his contributions to podS.

None of the research in this thesis would be possible without the support of my family and friends. My father was the first academic I knew. He and my mother encouraged me from a young age, along with Jenny and Jake, to be curious and challenge the world around me. Then Jeremy Minton, Jasmynne Bushrod, Sarah Kaewert and Steph Diepeveen provided me continual support, encouragement and distraction throughout my time in Cambridge.

I would also like to acknowledge EPSRC for funding my doctoral training studentship and the Department of Engineering, Jesus College, SIAM and GAMM/WIAS for funding my attendance at national and international conferences and workshops.



# Table of contents

<b>List of figures</b>	<b>xiii</b>
<b>List of tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Methods for uncertainty quantification . . . . .	2
1.1.1 Monte Carlo based methods . . . . .	3
1.1.2 Stochastic Galerkin methods . . . . .	4
1.1.3 Multilevel methods . . . . .	5
1.1.4 Statistical energy analysis . . . . .	6
1.2 Overview . . . . .	6
<b>2 Mathematical preliminaries</b>	<b>9</b>
2.1 The finite element method . . . . .	9
2.2 <i>A priori</i> error estimates . . . . .	10
2.3 The generalised eigenvalue problem . . . . .	11
2.4 Linear system solvers . . . . .	11
2.5 Point masses for vibration problems . . . . .	13
2.6 Karhunen-Loève expansions . . . . .	14
2.6.1 Numerical eigenpair approximation . . . . .	15
2.6.2 Analytical eigenpair expressions . . . . .	16
2.6.3 Comparisons of methods for computing eigenvalues . . . . .	18
<b>3 Monte Carlo based methods for uncertainty quantification</b>	<b>21</b>
3.1 The Monte Carlo method . . . . .	21
3.1.1 Practical implications . . . . .	24
3.1.2 Implementation . . . . .	25
3.2 Control variate approach for accelerating the Monte Carlo method . . . . .	26
3.3 The multilevel Monte Carlo method . . . . .	26
3.3.1 Multilevel estimator . . . . .	27
3.3.2 Determining the fidelity at level $L$ . . . . .	28

3.3.3	Determining the number of samples at each level . . . . .	28
3.3.4	Cost estimates . . . . .	29
3.3.5	Practical implications . . . . .	30
3.3.6	Implementation . . . . .	30
3.4	Software library implementation of Monte Carlo based methods . . . . .	32
3.4.1	Module description . . . . .	32
3.4.2	Outcomes . . . . .	33
<b>4</b>	<b>Structural vibration problems under uncertainty</b>	<b>35</b>
4.1	Finite element formulations . . . . .	36
4.1.1	Wave equation . . . . .	36
4.1.2	Kirchhoff–Love plate equation . . . . .	36
4.1.3	Linearised elastodynamics equation . . . . .	37
4.2	Eigensolvers . . . . .	37
4.3	Eigenvalues of structures . . . . .	38
4.3.1	Wave equation . . . . .	39
4.3.2	Kirchhoff–Love plate equation . . . . .	44
4.3.3	Linearised elastodynamics equation . . . . .	47
4.4	Energy density of Kirchhoff–Love plate . . . . .	54
4.4.1	Comparison of the frequency response . . . . .	55
4.4.2	Comparison of time cost . . . . .	56
4.5	Conclusions . . . . .	56
<b>5</b>	<b>Modelling of buoyancy driven flows</b>	<b>59</b>
5.1	Spurious buoyancy driven flows . . . . .	59
5.2	General formulation . . . . .	60
5.3	Finite element formulation for buoyancy driven flows . . . . .	61
5.3.1	Darcy equations . . . . .	61
5.3.2	Scalar transport equation . . . . .	62
5.4	<i>A priori</i> error estimates for Darcy equation formulations . . . . .	63
5.5	Numerical examples of compatible function spaces for Darcy flow . . . . .	64
5.6	Numerical examples of compatible function spaces for Stokes flow . . . . .	68
5.6.1	Finite element formulations for Stokes flow . . . . .	69
5.6.2	Numerical examples . . . . .	70
5.7	Conclusions . . . . .	71
<b>6</b>	<b>Long term dissolution of CO<sub>2</sub> in background flow</b>	<b>73</b>
6.1	CO <sub>2</sub> sequestration . . . . .	73
6.2	Non-dimensional equations . . . . .	74
6.3	Analytical model . . . . .	76

6.3.1	Gravity intrusion regime . . . . .	76
6.3.2	Buoyancy-driven shear dispersion regimes . . . . .	78
6.3.3	Regime differentiation . . . . .	83
6.4	Numerical model . . . . .	83
6.5	Model comparison . . . . .	85
6.5.1	Weak background flow . . . . .	85
6.5.2	Strong background flow . . . . .	90
6.6	Conclusions . . . . .	92
<b>7</b>	<b>Quantifying uncertainty in buoyancy driven flows</b>	<b>95</b>
7.1	Buoyancy driven fingering . . . . .	95
7.2	Model system . . . . .	98
7.3	Challenges of quantifying uncertainty in density driven fingering . . . . .	100
7.4	Impact of uncertainty on diffusion dominated fingering . . . . .	101
7.5	Conclusions . . . . .	104
<b>8</b>	<b>Conclusions</b>	<b>107</b>
8.1	Structural vibration under uncertainty . . . . .	108
8.2	Buoyancy driven flows under uncertainty . . . . .	108
8.3	Further research . . . . .	108
8.3.1	MLMC implementation . . . . .	109
8.3.2	Application specific developments . . . . .	109
8.3.3	Wider applicability of the MLMC method . . . . .	110
	<b>References</b>	<b>111</b>
	<b>Appendix A Derivation of analytical eigenpairs for Karhunen Loéve expansion</b>	<b>121</b>
A.1	Domain one . . . . .	121
A.2	Domain two . . . . .	125
	<b>Appendix B PodS module description</b>	<b>127</b>
B.1	Statistics module . . . . .	127
B.2	Uncertainty module . . . . .	128
B.2.1	Probability space . . . . .	128
B.2.2	Stochastic function . . . . .	129
B.3	Mesh module . . . . .	130
B.3.1	Create mesh . . . . .	130
B.3.2	Read mesh . . . . .	131
B.4	Scheduling module . . . . .	131
B.5	Pod module . . . . .	131

B.6	Log module . . . . .	131
B.7	Distribution . . . . .	132
<b>Appendix C Convergence rates for eigenvalue problems</b>		<b>133</b>
C.1	Wave equation . . . . .	133
C.2	Kirchhoff–Love plate . . . . .	134
C.3	Three dimensional elastodynamics . . . . .	134
<b>Appendix D Derivation of kinetic energy density of the Kirchhoff–Love plate</b>		<b>137</b>

# List of figures

1.1	Examples of sampling distributions for MC based methods. . . . .	3
2.1	<i>A priori</i> error estimates for the Poisson equation with a singular forcing term. . . . .	14
2.2	Eigenvalue decay of separable covariance kernel for different dimensions and length scales. . . . .	15
2.3	An example of a KL expansion. . . . .	16
2.4	Comparison of computational cost for KL expansions. . . . .	18
3.1	Illustration of practical implications of the MC method. . . . .	25
3.2	Illustration of practical implications of the MLMC method. . . . .	30
3.3	PodS module schematic. . . . .	32
3.4	Schematic for current scheduler. . . . .	33
3.5	PodS run time for the MC and MLMC simulations in parallel on varying numbers of cores for a fixed $\epsilon$ . . . . .	34
4.1	Comparison of the Krylov–Schur and Jacobi–Davidson eigenvalue solvers. . . . .	38
4.2	Comparison of the wave equation MC and MLMC probability distributions. . . . .	40
4.3	MLMC convergence rates for the first stable eigenvalue of the wave equation. . . . .	41
4.4	Comparison of time cost for the MC and MLMC simulations for the wave equation for different tolerances. . . . .	42
4.5	Comparison of the MC and MLMC probability distribution functions for the 25th eigenvalue of the wave equation. . . . .	44
4.6	Comparison of the Kirchhoff–Love MC and MLMC probability distributions. . . . .	45
4.7	MLMC convergence rates for the first eigenvalue of the Kirchhoff–Love plate equation for $\bar{\epsilon} = 10^{-5}$ . . . . .	46
4.8	Comparison of time taken for the MC and MLMC simulations for the Kirchhoff–Love plate equation for different relative tolerances. . . . .	47
4.9	Level two slender plate-like structure mesh. . . . .	48
4.10	The MLMC probability distribution functions for the first eigenvalue of a three dimensional slender plate-like structure for $\bar{\epsilon} = 7 \times 10^{-7}$ and $\bar{\epsilon} = 8 \times 10^{-7}$ . . . . .	48

4.11	MLMC convergence rates for the first eigenvalue of the slender three dimensional plate-like structure for $\bar{\epsilon} = 8 \times 10^{-7}$ .	49
4.12	Comparison of time taken for the MC and MLMC simulations for the slender three dimensional plate-like structure for different relative tolerances.	50
4.13	Level two curved plate-like structure mesh.	51
4.14	The MLMC probability distribution functions for the first eigenvalue of the curved three dimensional plate-like structure for varying tolerances.	52
4.15	MLMC convergence rates for the first eigenvalue of the curved three dimensional plate-like structure for $\bar{\epsilon} = 8 \times 10^{-5}$ .	53
4.16	Comparison of time taken for the MC and MLMC simulations for the curved three dimensional plate-like structure for different relative tolerances.	54
4.17	Time averaged energy density frequency response for different forcing frequencies.	55
5.1	Boundaries of domain for compatible function space investigation.	64
5.2	Velocity profile of coupled Darcy and scalar transport equation flow for different finite element formulations with polynomial forcing.	65
5.3	Darcy flow concentration profile along y-axis at $x = 0.5$ for polynomial forcing.	66
5.4	Velocity profile of coupled Darcy and scalar transport equation flow for different finite element formulations with sinusoidal forcing.	67
5.5	Darcy flow concentration profile along y-axis at $x = 0.5$ with sinusoidal forcing.	68
5.6	Stokes flow concentration profile along y-axis at $x = 0.5$ for different forcing terms.	71
6.1	Diagrams of CO <sub>2</sub> sequestration.	77
6.2	Illustration of the distance that CO <sub>2</sub> extends upstream.	83
6.3	Illustration of different regimes.	84
6.4	Concentration contours for $u_B = 0.1$ and different values of $Ra$ .	86
6.5	Computed and analytical variation of $\bar{c}$ with $x$ for different values of $Ra$ when $u_B = 8 \times 10^{-4}$ .	87
6.6	Concentration as a function of position in the domain for the gravity intrusion regime, with $u_B = 0.1$ and $Ra = 3000$ .	88
6.7	Normalised $L^1$ error norm for the computed two dimensional numeric and analytical values of $\bar{c}$ .	88
6.8	Typical streamlines of the flow for $u_B = 0.01$ and $Ra = 100, 10$ and $1$ .	89
6.9	Scaled velocity profiles at different points along the domain for different values of $\alpha$ with $u_B = 8 \times 10^{-4}$ .	90
6.10	Concentration fields for $u_B = 1.0$ and different values of $Ra$ .	91
6.11	Concentration profiles for $u_B \gtrsim 1$ .	92
7.1	Diagram indicating fingering behaviour in porous media flows.	96

---

7.2	Elder benchmark contours indicating 20% of the maximum concentration computed using FEniCS. . . . .	97
7.3	Concentration profiles indicating difference in flow behaviour due to varying permeability fields. . . . .	98
7.4	Domain for pollution scenario. . . . .	99
7.5	Concentration profiles for one realisation of a random field with different fidelity meshes. . . . .	100
7.6	Diverging convergence rates for the integral of the concentration at $y = 0.4$ for $Ra = 1000$ and time step 50. . . . .	101
7.7	Concentration profile for one realisation of a random field with for $Ra = 100$ at time step 50. . . . .	102
7.8	Comparison of the MC and MLMC probability distribution functions for the integral of concentration at $y = 0.4$ and time step 50. . . . .	102
7.9	MLMC convergence rates for the integral of concentration at $y = 0.4$ and time step 50.	103
7.10	Comparison of time taken for the MC and MLMC simulations for the integral of the concentration at $y = 0.4$ . . . . .	104
C.1	Convergence rates of the error in the first non zero eigenvalue of the wave equation. .	134
C.2	Convergence rates of the error in the first non zero eigenvalue of the Kirchhoff–Love plate equation. . . . .	135
C.3	Convergence rates of the error in the first non zero eigenvalue of the elastodynamics equation. . . . .	135



# List of tables

4.1	Number of elements and samples required for the MC simulation to ensure convergence of the wave equation to a specified relative tolerance. . . . .	42
4.2	Number of samples computed at each level for MLMC simulations of wave equation for specified relative tolerances. . . . .	43
4.3	Number of samples computed for the 25th eigenvalue of the wave equation using the MC and MLMC method. . . . .	43
4.4	Number of elements and samples required for the MC simulation to ensure convergence of the Kirchhoff–Love plate equation to a specified relative tolerance. . . . .	45
4.5	Number of samples computed at each level for MLMC simulations of Kirchhoff–Love plate for specified relative tolerances. . . . .	46
4.6	Number of elements and degrees of freedom in the meshes for each level in the slender three dimensional plate-like MLMC simulations. . . . .	48
4.7	Number of samples computed at each level for MLMC simulations of the slender three dimensional plate-like structure for specified tolerances. . . . .	50
4.8	Number of samples computed at each level for MLMC simulations of the curved three dimensional plate-like structure for specified tolerances. . . . .	52
4.9	Time taken to compute the MLMC and MC simulations of the time averaged energy density for different tolerances. . . . .	56
4.10	Number of samples taken to compute the time averaged energy density for different tolerances. . . . .	56
7.1	Number of samples taken at each level for MLMC simulations of the integral of the concentration at $y = 0.4$ . . . . .	104



# Chapter 1

## Introduction

*"When one admits that nothing is certain one must, I think, also add that some things are more nearly certain than others."*

Bertrand Russell, *Am I An Atheist Or An Agnostic?* 1947

Uncertainty quantification is the modelling and characterisation of uncertainties within a system. Assumptions and approximations are necessary to model real-world engineering systems mathematically; the more uncertainty there is in these assumptions, the less certainty there is in understanding. Therefore, it is imperative to understand the impact of uncertainty on specific quantities of interest so that robust systems can be designed that perform as expected.

Uncertainty in real-world engineering systems can be classified into two categories: epistemic and aleatoric [32]. The impact of both these types of uncertainty needs to be considered to comprehensively understand system behaviour. Epistemic, or systematic, uncertainty results from limited knowledge of the system and the data. This uncertainty can be reduced by taking more, or better, measurements. For example, it is hard to predict the average value rolled by a six-sided dice from only four observations, but a better prediction could be obtained by gathering several hundred observations. In contrast, aleatoric, or statistical, uncertainty is intrinsic to the model of the system and causes variations in the results each time a computation is repeated. It is characterised by a probability distribution and can be used to model a random event, which is not considered in the model system otherwise. Unlike epistemic uncertainty, the impact of aleatoric uncertainty is not reduced by better measurements. For example, the uncertainty in the expectation of the next roll of a six-sided fair dice is aleatoric. The roller is no more likely to be able to predict the next outcome by rolling a dice more times; however, the roller can be confident that the probability of rolling each number is one sixth.

Epistemic and aleatoric uncertainties are inherent in many engineering systems, such as groundwater pollution, earthquake resistance of buildings, and weather forecasting. These three examples highlight the importance of quantifying uncertainty in real-world problems. If uncertainties are not considered in the first example, ill-informed pollution management strategies can result in the unknown contamination of groundwater resources. The transport of pollution in aquifers in this system can be modelled as a flow through porous media that is subject to epistemic uncertainties, such

as the permeability of the aquifer, and aleatoric uncertainties, such as the weather. This can cause the pollutant to travel at different speeds and pool in different areas of the aquifer than predicted with a deterministic model.

For the second real-world example, it is important not just to design a building that will withstand an idealised earthquake, but to ensure that the structure is robust to all likely earthquakes. High frequency waves radiate from the epicentre of an earthquake and can excite resonant frequencies of buildings, which may result in catastrophic failure. However, these waves are uncertain, hard to measure, can radiate from different epicentres and have varying magnitudes. Therefore, to increase the likelihood of a building surviving, the impacts of uncertainties need to be accounted for to inform the design, so that the buildings do not resonate at the frequencies of the earthquakes they experience.

Since the 1960s, uncertainty quantification has dominated the third example, weather forecasting, because the non-linear chaotic weather system is subject to numerous uncertainties [86]. In addition to the aleatoric uncertainty in the numerical models, epistemic uncertainties in the initial state variables of the atmosphere, for example temperature, humidity, winds and pressure, can result in diverging weather forecasts [110]. Therefore, it is important to be able to quantify these uncertainties to understand long term weather trends and make useful predictions.

Consideration of uncertainty is not a new idea, but recent developments in computer science, engineering and mathematics have now made advances in the field possible. Despite being able to solve larger and more complex problems, quantifying the impact of uncertainties in real-world systems is still often considered too expensive without making model simplifications. So, much research considers deterministic cases, which may, or may not, provide a good indicator of real-world performance. The research in this thesis develops and applies a new method to enable uncertainty to be quantified in real-world engineering systems in feasible time frames. In the remainder of this chapter, a review of uncertainty quantification methods is presented in Section 1.1, and in Section 1.2 an overview of the work in this thesis is given.

## 1.1 Methods for uncertainty quantification

The canonical approach for quantifying uncertainty in engineering systems is the Monte Carlo (MC) method. This method is inherently very simple: the expectation of a quantity of interest (QoI),  $Q$ , is estimated by computing the average value of the QoI over  $N$  samples,

$$\mathbb{E}[Q] = \frac{1}{N} \sum_{i=1}^N Q(\omega_i), \quad (1.1)$$

where  $\omega_i$  is the  $i^{\text{th}}$  sample from a probability space [85, 89].

The MC method was used in 1873 to experimentally determine the value of  $\pi$  [56] by throwing matchsticks. However, it was not until the 1940s that this method was further developed and used to compute probabilistic problems in operational research and nuclear physics [57]. Now, MC

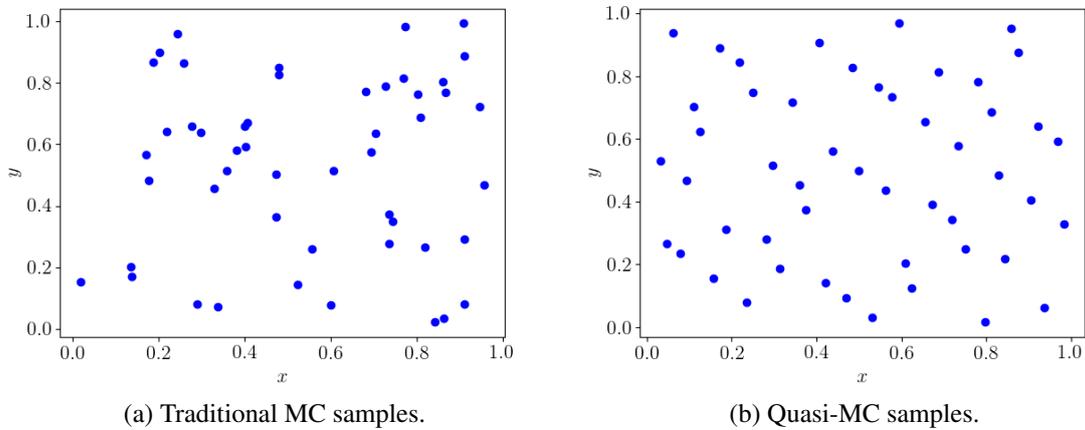


Figure 1.1: Examples of sampling distributions for MC based methods. Fig (a) shows an example of a uniform distribution and Fig (b) shows a Sobol sequence.

based methods have a much wider application in modern engineering, since stochastic problems are commonplace.

The main limitation of the MC method is that it is slow to converge. Consequently, it can be prohibitively expensive to sample enough of the probability space to accurately quantify the QoI when solving a single instance could take hours or days. Alternative methods have been developed to overcome this cost, which are discussed below: MC based methods, stochastic methods, multilevel methods, and a tailored method for vibration problems known as statistical energy analysis.

### 1.1.1 Monte Carlo based methods

Two widely used approaches to accelerate the traditional MC method are the quasi-Monte Carlo method and variance reduction techniques. The quasi-MC method accelerates convergence of the expectation of the QoI by using specially chosen samples, but unlike the MC method makes assumptions about the smoothness of the problem [91]. In the traditional MC method, the samples might be clustered because the individual samples are independent, see Figure 1.1a, but this method can still be advantageous because the error is independent of the stochastic dimension. In contrast, the clustering of samples is avoided in the quasi-MC method by choosing a correlated set of samples, for example a Sobol sequence (Figure 1.1b). This can improve the convergence rate of the method [19], but in practice only makes a difference if the number of samples is large and the number of stochastic dimensions is small.

Variance reduction methods decrease the error in the MC estimator for a given computational effort. Such methods include importance sampling, stratified sampling and using control variates.

- *Importance sampling*: The sampling distribution is distorted by sampling the important regions in the sample space more frequently, which reduces the variance of the estimator [66]. The

expectation of the QoI from a target probability space is approximated by a weighted average of random draws from a similar space [58]. The approximation accuracy depends on the choice of the second space; the closer the spaces, the fewer samples that are needed [117]. This method is limited because it requires a good approximation of the target probability space to be known *a priori*.

- *Stratified sampling*: The sample space is stratified and estimates of the QoI are computed from each stratum to optimise the sampling of a fixed number of stratum and samples. An adaptive method was formulated where more samples were taken from regions of the probability space, which were found to have largest variation. This method has better convergence rates for the expectation of the QoI than the traditional stratified sampling method, but is limited if the sample space cannot be split easily into different stratum [81].
- *Control variate method*: Information about the errors in estimates of known quantities are exploited in this method to reduce the error in an estimate of an unknown quantity. The control variate method replaces the problem of interest with two parts. The first part, the control variate, is an estimator of the problem, which can be known analytically or is computationally cheap to compute. The second part considers the difference between the control variate and exact problem. If the control variate is a good representation of the exact problem, the variance will be small and few expensive samples will be required.

### 1.1.2 Stochastic Galerkin methods

Stochastic methods have been developed as an alternative to the MC method for quantifying the impact of uncertainty on engineering systems. In these methods, the solution to a stochastic partial differential equation (SPDE) is represented using a polynomial chaos expansion, which can be formed from polynomials with unknown coefficients [131]. However, the accuracy in the solution of is reduced when the probability density function of the polynomial chaos expansion does not match the distribution of the stochastic variable it is representing [41]. These infinite expansions are also truncated in practice, which adds further error to the solution. High-order expansions are more accurate but are computationally more expensive [31].

The coefficients of polynomial chaos expansions can be found with the following two methods: stochastic Galerkin and stochastic collocation methods. In the stochastic Galerkin method, the error in the truncated expansion is minimised by applying a Galerkin projection to find the unknown expansion coefficients from the coupled equations [43, 136]; this error is bound with the estimates in Babuška et al. [6]. The computational cost of this method can be reduced by using more efficient solvers such as multigrid preconditioning [101].

Galerkin projection has three limitations. First, the system of complex, high dimensional, independent equations is typically larger than the deterministic system, which can make it difficult or impossible to find the solution with efficient and robust solvers [30, 136]. Second, computationally

expensive high order expansions are required when the solution varies rapidly over the domain or if a singularity exists in the random space [47, 49, 136]. Third, since the resulting equations are coupled, a deterministic solver code cannot be re-used immediately.

The stochastic collocation method overcomes some of the limitations of the stochastic Galerkin method. In the collocation method, the expansion is interpolated onto polynomial basis functions, which is then evaluated at compatible collocation points. Similar to the MC method, deterministic solvers can be used because the equations are uncoupled [7]. For multivariate problems, tensor products of univariate quadrature approximations can be used [7, 135].

Sparse tensor grid approximations can be used to lower the cost of the method by reducing the number of collocation points required [8, 92]. However, these methods are still expensive for quantifying high dimensional uncertainty, so the MC method may still be cheaper when the number of random parameters is high.

### 1.1.3 Multilevel methods

The convergence rate of the MC and stochastic methods have been increased by generalising the control variate approach to use a sequence of models with increasing fidelity. Kebaier [73] computed the expectation of a QoI for a time dependent stochastic finance problem by sampling the system many times for a coarse time step and a few additional times for a fine time step. The coarse representation acts as the control variate and is used to increase the convergence of the method, since it is computationally cheaper to compute than the fine representation. Heinrich [60] introduced the multilevel idea as a generalisation of the control variate method for solving the Fredholm integral, whereby, the value of the integral can be estimated using a number of different levels.

Giles [50] popularised the multilevel approaches from Heinrich and Kebaier as the multilevel Monte Carlo (MLMC) method. This method uses the convergence rates of the numerical error, cost and variance to determine the optimal number of levels and samples at each level. The MLMC method has been further analysed for different SPDEs, see Barth et al. [11], Charrier et al. [20], Cliffe et al. [26] and Teckentrup et al. [116]

The MLMC method has been further generalised, and applied to the stochastic collocation method, to reduce the cost of quantifying the impact of uncertainty in the following methods:

- The multi-index Monte Carlo method [55] uses multi-dimensional levels and high-order differences to further generalise the MLMC method. This method achieves optimal convergence rates for a wider range of problems than the MLMC method, but is more computationally expensive.
- The multifidelity Monte Carlo method [95] optimises a fixed computational budget for a set of differing fidelity model evaluations, for example from different surrogate models, to further generalise the MLMC method. This is beneficial when the convergence of the numerical error, cost and variance are not well defined. However, this method requires the highest fidelity method used to be known *a priori*, instead of selecting the highest fidelity model to satisfy some numerical error tolerance.

- The multilevel stochastic collocation method [59] uses a hierarchy of spatial and stochastic quadrature approximations to reduce the cost of stochastic collocation methods. Convergence rates for this method are given in Teckentrup et al. [115].

### 1.1.4 Statistical energy analysis

Statistical energy analysis [87] is a specific uncertainty quantification method that converges faster than the stochastic and MC based methods for computing the energy of random vibration problems, which is an application considered later in this thesis. In this method, the flow and storage of vibrational energy is calculated in complicated stochastic systems by dividing the system into independent subsystems. The total energy of the system is then computed from the time-average total of each of the probabilistic subsystems [88]. This method is inexpensive to solve because only a relatively coarse representation of the system is required [40].

Statistical energy analysis has been combined with finite element methods to solve larger systems [108]. In this hybrid method, some subsystems are chosen to be fully deterministic and modelled with the finite element method, and others are chosen to be highly random and modelled with statistical energy analysis. Non-parametric randomness, or randomness that cannot be parametrised by a finite number of parameters, is employed in the random subsections. This results in mode shapes and natural frequencies that conform to a Gaussian orthogonal ensemble and therefore, the mean and variance of the vibrational energy can be predicted without any knowledge of the physical uncertainty [88].

This hybrid finite element and statistical energy method was further developed by Cicirello and Langley [24, 25] to allow different uncertainties to be considered in the model. This was achieved by relaxing the statistical assumption that allow only non-parametric uncertainty to be considered. Despite this method being known as the hybrid finite element method, methods other than the finite element method can be used to solve the deterministic equations.

Statistical energy analysis is especially useful for mid to high frequency problems where the traditional MC method is too expensive [108]. This is because the underlying assumptions in structural energy analysis allow simplifications to the model that make the computations tractable in this frequency range.

## 1.2 Overview

This thesis examines uncertainty quantification in real-world engineering systems using the MLMC method, since this method shows potential for investigating systems with many dimensions of uncertainty. Uncertainty is quantified for two prototypical applications: vibration of structures and buoyancy driven flows through random porous media. These are chosen because they are two examples where uncertainty has a great impact on response. The impact of a random mass density field is quantified for frequencies of vibration and energy in the vibration examples, and the impact of a random permeability field is quantified for the porous media example.

The finite element method is introduced in Chapter 2, bounds on the approximation error are presented and different methods to solve the matrix equations that are formed are suggested. Methods for representing the uncertainty in the engineering systems are also considered, in particular point masses and Karhunen Loève expansions.

The MLMC method is derived in Chapter 3, with particular emphasis on using the finite element method. Cost complexity estimates are presented for both the MC and MLMC methods to highlight the computational cost benefit and time saving of the MLMC method. The new open source library, podS, developed for this research is also evaluated.

The cost benefit of using the MLMC method, instead of the traditional MC method, is illustrated in Chapter 4 for structural vibration problems. Theoretical cost estimates are shown to hold for eigenvalue problems and with the MLMC method, uncertainty can be quantified orders of magnitude quicker. This shows that uncertainty quantification of three dimensional structures can be tractable for random vibration problems.

In Chapter 5, different finite element formulations that can be used to model buoyancy driven flows through porous media are presented. These formulations are used to illustrate that spurious flows can be added to the solution if they cannot be represented on the function spaces point-wise.

The buoyancy driven flow formulations are then validated in Chapter 6 by considering the long term behaviour of CO<sub>2</sub> in an aquifer with background flow. The vertical averaged concentration profile of the two dimensional finite element model is compared with an analytical one dimensional model and a good match is obtained between the models in the parameter regime found in the aquifer. However, in other regimes, the flow is two dimensional so the one dimensional analytical model is a less good match.

In Chapter 7, the impact of uncertainties is considered for a pollution management example using the validated finite element model. This shows that the MLMC method can be applied to complex porous media systems. In this example, the CO<sub>2</sub> sequestration model is simplified with different boundary and initial conditions to model an unstable dense liquid on top of a lighter liquid. The impact of the uncertainty on the amount of the dense liquid that enters the lighter liquid is successfully quantified.

Finally, conclusions and further work are presented in Chapter 8. This chapter identifies two main areas where this research has contributed to scientific knowledge. First, the theoretical MLMC convergence rates are shown to hold for two real-world engineering systems. This means that it is now feasible to quantify uncertainty in certain circumstances orders of magnitude quicker than previously possible. Second, greater time savings can be obtained using the new software library podS. This specifically written library schedules the samples required for the MC and MLMC methods in parallel.

The appendix includes further derivations and analysis to support the research described above. Analytical expressions for Karhunen Loève expansion are derived in Appendix A. More information about podS, the library that was written to implement the Monte Carlo and multilevel Monte Carlo methods in parallel, is given in Appendix B. Then appendices specific to the vibration application are presented: the impact of using point masses on the regularity of the structural finite element

formulation is considered in Appendix C and the time averaged energy density of a Kirchhoff–Love plate is derived in Appendix D.

## Chapter 2

# Mathematical preliminaries

Stochasticity can be modelled in partial differential equation (PDE) governed systems using stochastic boundary conditions, initial parameters, model coefficients and geometries [51, 116]. Multiple deterministic samples of the stochastic PDE are required to quantify uncertainty in these systems using MC based methods, where each sample corresponds to one realisation of the system of interest. In this chapter, methods for solving deterministic samples of the system of interest and representing uncertainty with stochastic model coefficients are considered.

Various numerical methods, such as finite difference, finite volume and spectral methods, can be used to solve each deterministic sample of a PDE, but this research uses the finite element (FE) method. An outline of this chapter is as follows. The FE method is introduced in Section 2.1 and theoretical bounds on the approximation error are presented in Section 2.2. The FE method is then used to approximate the solution to an abstract generalised eigenvalue problem in Section 2.3 and the associated error bounds are presented. Common linear system solvers are discussed in Section 2.4 and two methods for representing uncertainty in the PDEs are presented: point masses in Section 2.5 and Karhunen Loéve expansions in Section 2.6.

### 2.1 The finite element method

Finite element methods can be used to approximate the solution to the weak form of a PDE in a domain  $D \in \mathbb{R}^d$ , where  $1 \leq d \leq 3$  is the spatial dimension of the domain [67]. The domain can be triangulated into non-overlapping simplices,  $K$ , where  $h_K$  is a measure of the size of the simplices. The boundary of the domain,  $\Gamma = \partial D$  can be partitioned into non-overlapping regions such that  $\Gamma_N \cup \Gamma_D = \partial D$  and  $\Gamma_N \cap \Gamma_D = \emptyset$ , where the  $N$  and  $D$  subscripts correspond to the two types of boundary conditions used in this thesis: Neumann and Dirichlet.

Consider finding the solution  $u \in V$  such that

$$a(u, v) = L(v) \quad \forall v \in V, \quad (2.1)$$

where  $V$  is a suitable function space,  $a(\cdot, \cdot)$  is a bilinear form on  $V \times V$ , and  $L(\cdot)$  is a linear functional on  $V$ .

If the Bubnov–Galerkin method is used to approximate the weak form in equation (2.1), the problem becomes: find  $u_h \in V_h$  such that

$$a(u_h, v_h) = L(v_h) \quad \forall v_h \in V_h, \quad (2.2)$$

where  $V_h \subset V$ .

After employing a suitable basis,  $\mathcal{B} = (\phi_1, \dots, \phi_n)$ , for  $V_h$ , equation (2.2) can be formulated in matrix form

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad (2.3)$$

where  $A_{i,j} = a(\phi_i, \phi_j)$  and  $b_i = L(\phi_i)$ . Assuming that  $\mathbf{A}$  can be inverted, the solution vector,  $\mathbf{u}$ , can then be computed.

## 2.2 *A priori* error estimates

Two norms of interest are considered to analyse how accurate the finite element solutions are compared to the exact value of the solution. First, the  $L_2$  norm, which is defined as the square root of the integral of the solution squared,

$$\|u\|_{L_2} = \left( \int_D u^2 dx \right)^{1/2}. \quad (2.4)$$

Then the  $H_1$  norm, which also considers the smoothness of the solution by considering the first derivatives of the solution,

$$\|u\|_{H_1} = \left( \int_D u^2 + \nabla u \cdot \nabla u dx \right)^{1/2}. \quad (2.5)$$

Assuming  $a(\cdot, \cdot)$  is uniformly elliptic and  $u \in H^{p+1}$ , the error in the FE solution can be bound by *a priori* estimates. The error satisfies

$$\|u - u_h\|_{H^s} \leq Ch^\alpha |u|_{H^{p+1}}, \quad (2.6)$$

where  $h$  is the maximum of  $h_K$ ;  $s$  is the norm of interest;  $p$  is the polynomial order of the basis functions;  $C$  is a problem dependent constant that is independent of  $h$  and  $u$ ; and

$$\alpha = \min(p + 1 - s, 2(p + 1 - m)), \quad (2.7)$$

where  $m$  is the highest order derivative in the weak form [67].

From equation (2.6), the  $L_2$  error estimate with first order derivatives in the weak form is

$$\|u - u_h\|_{L_2} \leq Ch^{p+1} |u|_{H^{p+1}}. \quad (2.8)$$

This means that the solution converges with  $O(h^2)$  for linear polynomial basis functions,  $p = 1$ . The  $H_1$  error estimate with first order derivatives in the weak form is

$$\|u - u_h\|_{H_1} \leq Ch^p |u|_{H^{p+1}}, \quad (2.9)$$

so the solution converges with  $O(h)$  for linear finite basis functions.

## 2.3 The generalised eigenvalue problem

Generalised eigenvalue problems are of interest for a range of problems, including structural vibration. Consider the following variational formulation of an eigenvalue problem with multiple solutions: find the  $k^{\text{th}}$  solution  $\lambda_k \in \mathbb{R}$  and  $u_k \neq 0 \in V$  such that

$$a(u_k, v_k) = \lambda_k b(u_k, v_k) \quad \forall v_k \in V, \quad (2.10)$$

where  $a(\cdot, \cdot)$  and  $b(\cdot, \cdot)$  are bilinear forms on  $V \times V$  and  $(\lambda_k, u_k)$  are listed in order of increasing size of eigenvalue, with the smallest eigenvalue pairs being of most interest.

The Galerkin method can be used to find an approximate solution to equation (2.10). The problem becomes: find  $\lambda_{k,h} \in \mathbb{R}$  and  $u_{k,h} \neq 0 \in V_h$  such that

$$a(u_{k,h}, v_{k,h}) = \lambda_{k,h} b(u_{k,h}, v_{k,h}) \quad \forall v_{k,h} \in V_h. \quad (2.11)$$

The eigenvalue problem in equation (2.11) can be formulated in matrix form using the basis functions  $\mathcal{B} = (\phi_1, \dots, \phi_n)$

$$\mathbf{A}\mathbf{u} = \boldsymbol{\lambda}\mathbf{M}\mathbf{u}, \quad (2.12)$$

where  $A_{i,j} = a(\phi_i, \phi_j)$  commonly defines the stiffness matrix,  $M_{i,j} = b(\phi_i, \phi_j)$  commonly defines the mass matrix,  $\mathbf{u}$  is a vector of eigenfunctions and  $\boldsymbol{\lambda}$  is a vector of eigenvalues.

The *a priori* error estimate of the  $k^{\text{th}}$  eigenvalue [67] is

$$\lambda_k \leq \lambda_{h,k} \leq \lambda_k + Ch^{2(p+1-m)} \lambda_k^{(p+1)/m}, \quad (2.13)$$

where  $C$  is a different constant independent of  $h$  and  $\lambda_k$ ; and similar to equation (2.6),  $p$  is the polynomial order of the basis functions and  $m$  is the highest derivative in the weak form. This means that the eigenvalues should converge with  $O(h^2)$  for linear basis functions, and  $O(h^4)$  for quadratic basis functions, if the weak form only has first order derivatives.

## 2.4 Linear system solvers

The solution to the finite element matrix problems in equations (2.3) and (2.12) can be computed using either direct or iterative solvers, which have different algorithmic complexities and memory

requirements. A discussion of solver complexity is presented here because it forms an important part of the MLMC analysis.

As the name suggests, direct solvers compute the solution to equations (2.3) and (2.12) for a pre-determined amount of work. As the size of the  $n \times n$  system increases, the time taken to compute the solution is  $O(n^3)$  if the matrices are dense [46]. The computation time is less for sparse matrices, but it is dimension dependent: the complexity of sparse direct methods is  $O(n^{3/2})$  in two dimensions and  $O(n^2)$  in three dimensions [72]. LU decomposition is the most common type of direct solver used and is a modified form of Gaussian elimination. If the matrices are symmetric and positive definite Cholesky factorisation can also be used, which is more efficient version of the LU decomposition method.

If  $n$  is large, direct methods can be expensive and the required matrices may not fit in the available memory. Instead, it may be possible to use iterative solvers, which generally require less memory because smaller matrices are stored. Unlike direct methods, Krylov subspace iterative methods approximate the solution to the matrix equations by repeatedly performing matrix-vector multiplications, see Saad [102]. If the matrices are symmetric, the conjugate gradient (CG) method [63, 102] can be used, whereas if the matrices are non-symmetric, the generalised minimum residual (GMRES) [103] can be used instead.

Although Saad [102] shows that Krylov subspace iterative methods are guaranteed to converge in the absence of round-off errors, their convergence rates can be very slow. Preconditioners can lead to more rapid convergence. Rather than solving the matrix equations in (2.3) and (2.12), they approximate the solution to

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{u} = \mathbf{P}^{-1}\mathbf{b}, \quad (2.14)$$

and

$$\mathbf{P}^{-1}(\mathbf{A} - \lambda\mathbf{M})\mathbf{u} = 0, \quad (2.15)$$

where  $\mathbf{P}$  is the preconditioner. The closer the preconditioner is to  $\mathbf{A}$ , or  $\mathbf{A} - \lambda\mathbf{M}$ , the faster the solver converges to the respective solution. Simple preconditioners are the Jacobi and Gauss–Seidel methods [102].

Multigrid techniques have been developed to overcome the increase in cost of iterative solvers that sometimes occurs with increases in matrix size. These techniques exploit different mesh discretisations to obtain optimal convergence rates for iterative methods that, are in theory, independent of matrix size [102, 130]. These techniques can be used as solvers themselves or to precondition other iterative methods.

The beginning of this chapter consider using FE methods to solve deterministic PDEs. When quantifying uncertainty, inherent stochasticity in the system needs to be captured and included in the model. The following sections consider methods for representing stochastic variables in the model.

## 2.5 Point masses for vibration problems

Uncertainty can be represented in deterministic FE formulations using perturbations in stochastic variables. For each sample of the PDE, a different realisation of the perturbation is computed. Uncertainty in mass density can be represented by adding small amounts of mass at discrete points. However, it does not make sense to represent all stochastic variables as point masses, or sources, because the length scale over which the uncertain variables change is much longer than points.

Point masses can be represented mathematically by Dirac delta functions, which are added to the FE mass matrix in equation (2.12) at any location. For  $N$  point masses located at  $x_n \in D$ , the mass matrix is

$$M = \int_D \rho_0 u_h v_h + \sum_{n=1}^N \xi_n \delta(x_n) u_h v_h \, dx, \quad (2.16)$$

where  $\xi_n$  is the magnitude of the  $n^{\text{th}}$  point mass and  $\rho_0$  is the unperturbed density of the material. This discrete representation of the mass matrix is non-smooth and includes singularities.

Using point masses to represent a stochastic field is simple and is how uncertainty is commonly represented in the vibration literature, such as Choi et al. [21], and experimentally, such as Adhikari et al. [2]. However, it is disadvantageous to use point masses because, as shown below, they cause sub-optimal error convergence rates result. This is because the FE formulations are no longer smooth, so do not meet the regularity requirements of the error estimate. There is little published analysis into the impact that singularities in the mass matrix have on the convergence rate. Although Scott [106] shows that the convergence of the  $L_2$  error norm, with a singular forcing term,  $L(v)$ , and linear elements, is  $O(h)$  instead of  $O(h^2)$ , see equation (2.8).

In Figure 2.1, the sub-optimal convergence rate of the following Poisson equation with a singular forcing term is computed using the FEniCS libraries [4] for a unit square domain:

$$-\nabla^2 u = \delta(\mathbf{x}_0) \quad \text{on } D, \quad (2.17a)$$

$$u = 0 \quad \text{in } \partial D, \quad (2.17b)$$

where  $\mathbf{x}_0 = (0.37654, 0.27653)$  so that the singularity location does not coincide with a node for any of the meshes considered. The exact solution to equation (2.8) is unknown so a fine approximation of the solution is used instead. The  $L_2$  convergence rate is  $O(h)$  for linear elements, as expected from the analysis in Scott [106], and  $O(h)$  for quadratic elements. The observed convergence rate for the  $H^1$  norm with linear elements is  $\approx O(h^{0.3})$ . It is also noted that the tolerances of the eigenvalue solvers may need to be tightened to ensure that the convergence rate of the  $L_2$  error norm is at least  $O(h)$ . This can significantly increase the number of iterations required, and so increase the cost of the solver.

A second disadvantage of using point masses is that the mean mass density of the system changes. This is because experimentally it would be very difficult to remove significant mass from a point, whilst keeping it intact, and numerically if too much mass is removed, the density field could go negative, which is non-physical.

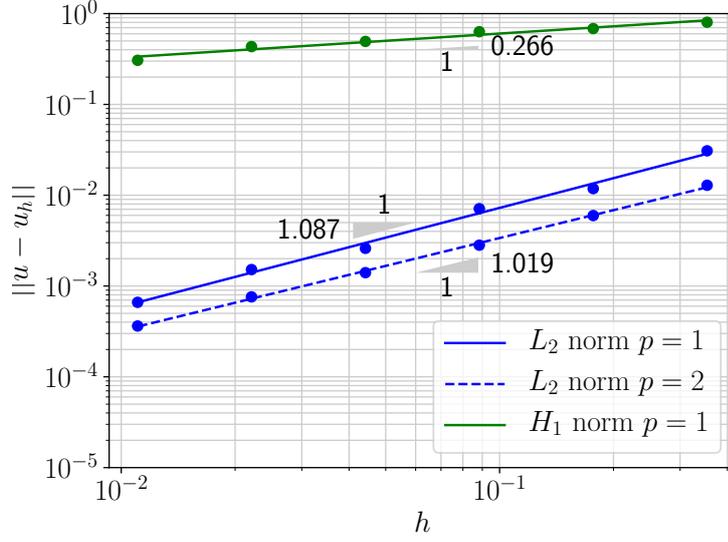


Figure 2.1: *A priori* error estimates for the Poisson equation with a singular forcing term.

## 2.6 Karhunen-Loève expansions

Karhunen–Loève (KL) expansions are an approach to representing variables as stochastic fields. These expansion are a linear combination of orthogonal functions [20, 48],

$$X = \mu + \sum_{j=1}^{\infty} \sqrt{v_j} \phi_j \xi_j(\omega), \quad (2.18)$$

where  $X(\mathbf{x}, \omega)$  is the stochastic field of interest,  $\omega$  is a sample of the probability space,  $\mu$  is the mean of the stochastic field,  $\xi_j(\omega) \sim N(0, 1)$  is a normally distributed random number, and  $\phi_j(\mathbf{x})$  and  $v_j$  are the eigenfunctions and eigenvalues of a covariance kernel respectively. The probability space is defined as  $(\Omega, \mathcal{F}, \mathbb{P})$ , where  $\Omega$  is a sample space, or the set of all possible outcomes;  $\mathcal{F}$  is a set of events, where each event contains a set of zero or more outcomes; and  $\mathbb{P}$  is a probability measure that assigns probabilities to events [85].

The eigenfunctions and eigenvectors of the covariance kernel can be found by solving the following Fredholm integral equation:

$$\int_D C(\mathbf{x}, \mathbf{x}') \phi_j(\mathbf{x}') d\mathbf{x}' = v_j \phi_j(\mathbf{x}), \quad (2.19)$$

where  $C(\mathbf{x}, \mathbf{x}')$  is a covariance kernel, which describes the spatial covariance of the random field. The  $j^{\text{th}}$  eigenpair,  $(v_j, \phi_j)$  can be found either numerically or analytically, as described in the following sections, and the eigenpairs are sorted into descending order of eigenvalues.

In practice, the expansion in equation (2.18) is truncated after the  $J^{\text{th}}$  largest eigenvalue,

$$X = \mu + \sum_{j=1}^J \sqrt{v_j} \phi_j \xi_j. \quad (2.20)$$

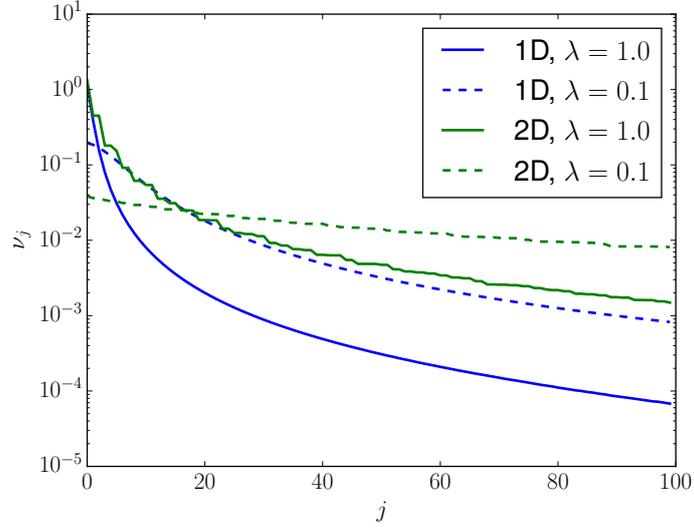


Figure 2.2: Eigenvalue decay of separable covariance kernel for different dimensions and length scales. For each dimension  $D = [-1.0, 1.0]$  and  $\sigma = 1.0$ .

The parameter  $J$  is chosen by considering the decay rate of  $\nu_j$ , from equation (2.19), for the desired accuracy [85]. Figure Figure 2.2 shows the decay of  $\nu_j$  for the one and two dimensional separable covariance kernel:

$$C(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{m=1}^d e^{-\frac{|\mathbf{x}_m - \mathbf{x}'_m|}{\lambda_m}} \quad \text{on } D, \quad (2.21)$$

where  $\sigma^2$  is the variance of the covariance kernel,  $\lambda$  is a length scale, and  $d$  is the number of spatial dimensions. An example of a KL expansion with the covariance kernel from equation (2.21) is shown in Figure 2.3.

The rate of the eigenvalue decay depends on the number of spatial dimensions and the length scale of the covariance kernel. The higher the number of spatial dimensions, the slower the rate of decay. This is because more eigenpairs are required to accurately represent higher spatial dimensions. In addition, the smaller the length scale, the slower the decay rate. So again, more eigenpairs are required because rougher fields can be represented on shorter length scales; however, more terms are required in the expansion to represent them accurately.

### 2.6.1 Numerical eigenpair approximation

The eigenpairs of equation (2.19) can be computed numerically using Galerkin projection. This method finds the approximate eigenfunctions  $\hat{\phi}_{h,j} \in V_h$  and eigenvectors  $\hat{\nu}_{h,j} \in \mathbb{R}$  such that

$$\int_D \left( \int_D C(\mathbf{x}, \mathbf{x}') \hat{\phi}_{h,j}(\mathbf{x}') d\mathbf{x}' \right) \tau_h d\mathbf{x} = \int_D \hat{\nu}_{h,j} \hat{\phi}_{h,j}(\mathbf{x}) \tau_h d\mathbf{x} \quad \forall \tau_h \in V_h. \quad (2.22)$$

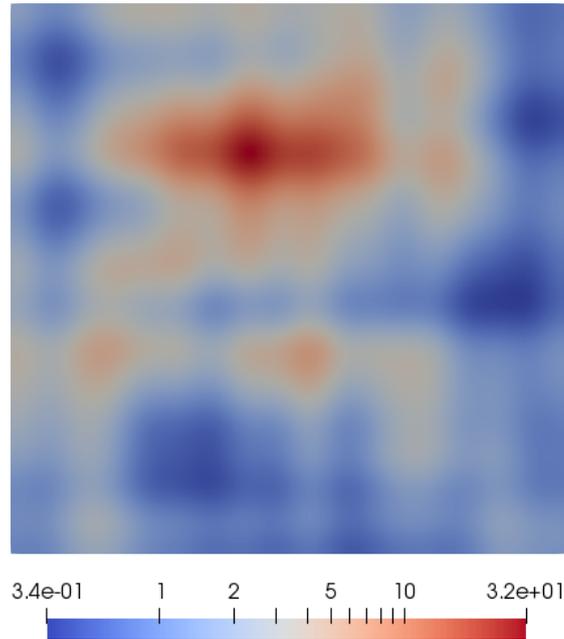


Figure 2.3: An example of the exponential of a KL expansion with a separable covariance kernel,  $\mu = 0.0$ ,  $\sigma = 1.0$  and  $\lambda = 0.25$ .

Numerical KL expansions are advantageous because they can be used to realise stochastic functions for domains with complicated shapes, and any covariance kernel can be used. However, this method is computationally expensive. Assembling the covariance operator and computing the eigenpairs of dense matrices are costly operations, even if the expansion is truncated and only a few eigenpairs are used.

To overcome the computational expense of dense matrices, Lindgren et al. [83] suggested that Gaussian Markov random fields could be used instead of Gaussian fields. Gaussian Markov random fields are discretely indexed and the Markov property ensures that the matrix is sparse so computationally cheaper algorithms can be used.

### 2.6.2 Analytical eigenpair expressions

Analytical expressions of the eigenpairs from equation (2.19) can be found for the separable exponential covariance kernel,

$$C(x, x') = \sigma^2 e^{-\frac{|x-x'|}{\lambda}} \quad \text{on } D, \quad (2.23)$$

for the domains:

1.  $D_1 = [-a, a]$ ,
2.  $D_2 = [0, a]$ ,

and multi-dimensional domains, which are the product of the one dimensional domains above. Using analytical eigenpairs enables significantly cheaper computation of the KL expansion for these restricted cases.

For  $D_1 = [-a, a]$ , the  $j^{\text{th}}$  eigenpair is

$$\phi_j(x) := \begin{cases} A_j \cos(\omega_j x), & j \text{ odd,} \\ B_j \sin(\omega_j x), & j \text{ even,} \end{cases} \quad v_j = \frac{2\sigma^2 \lambda}{\omega_j^2 \lambda^2 + 1}, \quad (2.24)$$

where

$$A_j = \frac{1}{\sqrt{a + \sin(2\omega_j a)/2\omega_j}}, \quad B_j = \frac{1}{\sqrt{a - \sin(2\omega_j a)/2\omega_j}}, \quad (2.25)$$

and

$$\omega_j := \begin{cases} \hat{\omega}_{\lceil j/2 \rceil}, & j \text{ odd,} \\ \tilde{\omega}_{j/2}, & j \text{ even.} \end{cases} \quad (2.26)$$

The parameters  $\hat{\omega}_j$  and  $\tilde{\omega}_j$  are the positive roots of the two transcendental equations:

$$\lambda^{-1} - \hat{\omega}_j \tan(\hat{\omega}_j a) = 0, \quad (2.27a)$$

$$\tilde{\omega}_j + \lambda^{-1} \tan(\tilde{\omega}_j a) = 0. \quad (2.27b)$$

The roots of equations (2.27a) and (2.27b) can be found numerically, for example using the Newton–Raphson method. For a full derivation of these eigenpairs see Appendix A.1.

For  $D_2 = [0, a]$ , the  $j^{\text{th}}$  eigenpair is:

$$\phi_j = B_j (\lambda \omega_j \cos(\omega_j x) + \sin(\omega_j x)), \quad v_j = \frac{2\sigma^2 \lambda}{\omega_j^2 \lambda^2 + 1}, \quad (2.28)$$

where

$$B_j = \sqrt{\frac{4\omega_j}{(\lambda^2 \omega_j^2 - 1) \sin(2\omega_j a)}}, \quad (2.29)$$

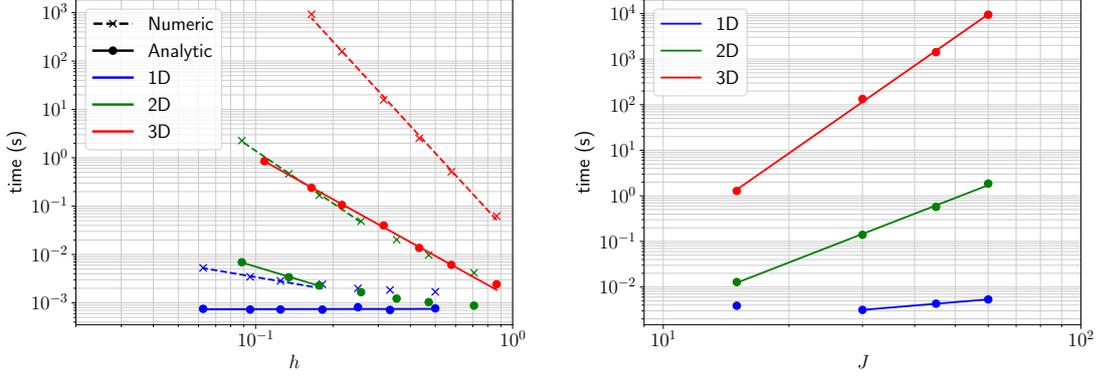
and  $\omega_j$  are the roots of

$$\tan(\omega_j a) = \frac{2\omega_j \lambda}{\omega_j^2 \lambda^2 - 1}. \quad (2.30)$$

For a full derivation of these eigenpairs see Appendix A.2.

Since the covariance kernel in equation (2.23) is separable, in higher dimensions the covariance kernel is the product of the one dimensional kernels,

$$C(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{m=1}^d e^{-\frac{|x_m - x'_m|}{\lambda_m}} \quad \text{on } D, \quad (2.31)$$



(a) Cost comparison of different fidelity meshes. (b) Cost comparison of different truncation points.

Figure 2.4: Comparison of computational cost for KL expansion fields with  $\mu = 0.0$  and  $\sigma = 1.0$ . Fig (a) shows the analytical and numerical cost of computing expansions for different fidelity meshes and Fig (b) shows the analytical cost of computing KL expansions that have been truncated after different numbers of eigenvalues.

where  $d$  is the number of spatial dimensions. Therefore, the eigenfunctions and eigenvalues for  $d = 3$  are

$$\phi_j = \phi_i^1(x_1)\phi_k^2(x_2)\phi_l^3(x_3), \quad v_j = v_i^1 v_k^2 v_l^3, \quad (2.32)$$

where the eigenpairs  $\{v_i^1, \phi_i^1\}$ ,  $\{v_k^2, \phi_k^2\}$  and  $\{v_l^3, \phi_l^3\}$  are the solutions to the following one dimensional problem:

$$\int_D \sigma^2 e^{-\frac{|x_m - x'_m|}{\lambda_m}} \phi^m(x') dx' = v^m \phi^m(x), \quad m = 1, 2, 3. \quad (2.33)$$

### 2.6.3 Comparisons of methods for computing eigenvalues

The cost of realising one sample of a stochastic field has been investigated in Figure 2.4a for expansions computed with the first four numeric and analytical eigenpairs on different fidelity meshes. One process of a desktop computer with one Intel i7-5820K processor and 32 GiB of RAM is used. It is much more expensive to compute numeric KL expansions in equation (2.22) using the LAPACK package [79] than using the analytical eigenpairs in Section 2.6.2. For example, it requires less than 1 s to compute the three dimensional KL expansion for  $h = 0.217$  using the analytical eigenpairs in equation (2.19), but approximately 150 s with the numeric eigenpairs. It is not possible to compute the three dimensional numeric KL expansion with a smaller cell size due to memory restrictions.

The cost of realising a stochastic field using different numbers of analytical eigenpairs is compared in Figure 2.4b for  $h = 0.02$ . The numerical eigenpairs have not been used here because the computation times were too great when more than four eigenpairs were used in the computation. As expected, the cost of computing the expansions increases as more eigenpairs are considered. For example, it takes approximately 1 s to compute a three dimensional expansion with 15 eigenpairs, but it takes approximately 2.7 hours to compute a three dimensional expansion with 60 eigenpairs.

---

It is, however, only possible to use analytical methods to compute the eigenpairs of the separable exponential covariance kernels with interval, rectangle or box domains. For other covariance kernels or geometries, numeric eigenpairs are required. Alternative methods, such as kriging and circular embedding, can be used to reduce the cost of computing KL expansions using numerical eigenpairs, but the computational cost is still higher than using analytical expansions [85].



## Chapter 3

# Monte Carlo based methods for uncertainty quantification

Three Monte Carlo (MC) based methods are presented in this chapter, their relative cost complexities are compared and practical implementations of the algorithms are given. In Section 3.1, the canonical method for uncertainty quantification, the traditional MC method, is presented. The MC method may be prohibitively expensive for real-world applications, so the control variate method is presented in Section 3.2. The control variate method is then generalised in Section 3.3 to form the multilevel Monte Carlo method (MLMC) [51]. This chapter concludes in Section 3.4 with a description of the library, which was specifically developed for this research, to parallelise the scheduling of the individual samples required in the MC and MLMC algorithms.

### 3.1 The Monte Carlo method

Uncertainty quantification involves computing the expectation, or expected value, and variance of a QoI,  $Q(\omega)$ , where  $\omega$  represents a sample from a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ . In this section, the traditional MC method is presented.

The expectation of the continuous QoI is defined as the following Lebesgue integral

$$\tilde{Q} = \mathbb{E}[Q(\omega)] = \int_D Q(\omega) d\mathbb{P}(\omega), \quad (3.1)$$

where  $\mathbb{E}[Q(\omega)]$  denotes the expectation of the QoI. Equation (3.1) can be approximated numerically using a MC estimator.

Assuming that the systems of interest are approximated using FE methods, the approximation of the QoI is denoted by  $Q_h$  where, similar to FE methods,  $h$  is a measure of the fidelity of the model.

Therefore, the MC estimator,  $\tilde{Q}_{h,N}$ , of the expectation of the QoI can be approximately defined as

$$\tilde{Q}_{h,N} = \frac{1}{N} \sum_{i=1}^N Q_h(\omega_i), \quad (3.2)$$

where  $N \in \mathbb{N}$  is the number of samples. As  $h \rightarrow 0$  and  $N \rightarrow \infty$ , the FE approximation and sampling errors tends to zero and  $Q_{h,N} \rightarrow \tilde{Q}$ :

$$\tilde{Q}_{h,N} \xrightarrow[N \rightarrow \infty]{h \rightarrow 0} \tilde{Q}. \quad (3.3)$$

Two useful properties of the expectation operator are as follows:

$$\mathbb{E}[\mathbb{E}[X]] = \mathbb{E}[X], \quad (3.4)$$

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]. \quad (3.5)$$

Using equation (3.4), the expectation of the MC estimator is equal to the numerical expectation of the QoI,  $\mathbb{E}[\tilde{Q}_{h,N}] = \mathbb{E}[Q_h]$ . The variance of the MC estimator  $\mathbb{V}[\tilde{Q}_{h,N}]$  is given by

$$\mathbb{V}[\tilde{Q}_{h,N}] = \frac{1}{N} \mathbb{V}[Q_h]. \quad (3.6)$$

There are two sources of error in the estimator in equation (3.2): the sampling error ( $e_V$ ) and the numerical error ( $e_h$ ). These become obvious from considering the root-mean-square error between the estimator  $Q_{h,N}$  and the exact expectation,

$$e^2 := \mathbb{E} \left[ \left( \tilde{Q}_{h,N} - \tilde{Q} \right)^2 \right] = e_V^2 + e_h^2. \quad (3.7)$$

Expanding equation (3.7) by adding and subtracting the expectation of the estimator  $\mathbb{E}[\tilde{Q}_{h,N}]$ ,

$$\begin{aligned} e^2 &= \mathbb{E} \left[ \left( \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right) + \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right) \right)^2 \right], \\ &= \mathbb{E} \left[ \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right)^2 + 2 \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right) \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right) + \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right)^2 \right]. \end{aligned} \quad (3.8)$$

Using the property in equation (3.5), equation (3.8) becomes

$$e^2 = \mathbb{E} \left[ \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right)^2 \right] + \mathbb{E} \left[ 2 \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right) \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right) \right] + \mathbb{E} \left[ \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right)^2 \right]. \quad (3.9)$$

Since  $\tilde{Q} = \mathbb{E}[Q]$ , after applying the properties from equation (3.4) and (3.5),  $\mathbb{E}[\mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q}] = \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q}$ . Therefore, equation (3.9) becomes

$$e = \mathbb{E} \left[ \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right)^2 \right] + 2 \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right) \mathbb{E} \left[ \left( \tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}] \right) \right] + \mathbb{E} \left[ \left( \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q} \right)^2 \right]. \quad (3.10)$$

Again from applying the properties in equation (3.4) and (3.5),  $\mathbb{E} \left[ (\mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q}) \right] = \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q}$  and  $\mathbb{E} \left[ \mathbb{E}[\tilde{Q}_{h,N}] \right] = \mathbb{E}[\tilde{Q}_{h,N}]$ . Therefore, equation (3.10) can be simplified as follows:

$$\begin{aligned} e^2 &= \mathbb{E} \left[ (\tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}])^2 \right] + 2(\mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q})(\mathbb{E}[\tilde{Q}_{h,N}] - \mathbb{E}[\tilde{Q}_{h,N}]) + (\mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q})^2, \\ &= \mathbb{E} \left[ (\tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}])^2 \right] + (\mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q})^2. \end{aligned} \quad (3.11)$$

Equation (3.11) can be further simplified because  $\mathbb{E} \left[ (\tilde{Q}_{h,N} - \mathbb{E}[\tilde{Q}_{h,N}])^2 \right]$  is the variance of the estimator and  $\mathbb{E}[\tilde{Q}_{h,N} - \tilde{Q}] = \mathbb{E}[\tilde{Q}_{h,N}] - \tilde{Q}$

$$e^2 = \mathbb{V}[\tilde{Q}_{h,N}] + \mathbb{E} \left[ \tilde{Q}_{h,N} - \tilde{Q} \right]^2. \quad (3.12)$$

Substituting equation (3.6) into equation (3.12),

$$e^2 = \underbrace{\frac{1}{N} \mathbb{V}[Q_{h,N}]}_{e_V^2} + \underbrace{\mathbb{E} \left[ \tilde{Q}_{h,N} - \tilde{Q} \right]^2}_{e_h^2}. \quad (3.13)$$

Equation (3.13) reduces to the classical Monte Carlo error estimate if the discretisation error,  $\mathbb{E} \left[ \tilde{Q}_{h,N} - Q \right]^2$ , is assumed to be zero. Convergence of the estimator is assumed if the error is below a certain tolerance,  $e^2 < \varepsilon^2$ , where  $\varepsilon$  is the tolerance. This tolerance is further decomposed into the sampling error tolerance  $\varepsilon_V$  and the numerical error tolerance  $\varepsilon_h$ , so  $\varepsilon^2 = \varepsilon_V^2 + \varepsilon_h^2$ .

The discretisation error,  $\mathbb{E} \left[ \tilde{Q}_{h,N} - Q \right]^2$ , can be bound by the usual *a priori* error estimates described in Section 2.2. This means that the expectation of the numerical error in the QoI is bound by

$$\mathbb{E} \left[ \tilde{Q}_{h,N} - Q \right] \leq c_\alpha h^\alpha, \quad (3.14)$$

where  $\alpha > 0$  is a constant and  $c_\alpha > 0$  is a constant that is independent of  $h$ . To satisfy  $e^2 \leq \varepsilon^2$ , the mesh cell size is determined by  $e_h^2 \leq \varepsilon_h^2$ , or equivalently

$$h \leq \left( \varepsilon_h c_\alpha^{-1} \right)^{\frac{1}{\alpha}}, \quad (3.15)$$

and the number of required samples is determined by  $e_V^2 \leq \varepsilon_V^2$ , leading to

$$N \geq \mathbb{V}[Q_h] \varepsilon_V^{-2}. \quad (3.16)$$

The cost of computing a single sample of the PDE is assumed to be

$$C_i(Q(\omega_i)) = c_\gamma h^{-\gamma d}, \quad (3.17)$$

where  $c_\gamma > 0$  and  $\gamma > 0$  are constants that do not depend on  $h$ , and  $d$  is the spatial dimension of the problem. Substituting equation (3.15) into equation (3.17) and assuming  $\eta = \frac{\gamma d}{\alpha}$ ,

$$C_i(Q(\omega_i)) = c_\gamma \varepsilon_h^{-\eta} c_\alpha^\eta. \quad (3.18)$$

The cost is described in terms of  $d$ , since the cost parameter  $\gamma$  typically only has a mild dependence on  $d$ .

The total cost of computing  $\tilde{Q}_h$  from  $N$  samples is bound by

$$C(\tilde{Q}_h) \leq N C_i, \quad (3.19)$$

where it is assumed that

$$N = \mathbb{V}[Q] \varepsilon_V^{-2} + 1, \quad (3.20)$$

to ensure the inequality in equation (3.16) holds. Substituting equation (3.18) into equation (3.19) results in

$$C \leq c_\gamma \varepsilon_h^{-\eta} c_\alpha^\eta \mathbb{V}[Q] \varepsilon_V^{-2} + c_\gamma \varepsilon_h^{-\eta} c_\alpha^\eta. \quad (3.21)$$

Further assuming the error tolerance is split such that  $\varepsilon_V^2 = (1 - \delta) \varepsilon^2$  and  $\varepsilon_h^2 = \delta \varepsilon^2$ , where  $\delta \in (0, 1)$ , the bound on the cost in equation (3.21) can be re-written as

$$C \leq (1 - \delta)^{-1} \delta^{-\frac{\eta}{2}} c_\gamma c_\alpha^\eta \mathbb{V}[Q] \varepsilon^{-2-\eta} + c_\gamma \delta^{-\frac{\eta}{2}} c_\alpha^\eta \varepsilon^{-\eta}. \quad (3.22)$$

The cost of the MC method is therefore  $O(\varepsilon^{-2-\eta})$  or  $O(\varepsilon^{-2-\frac{\gamma d}{\alpha}})$ . If the discretisation error is not considered, this simplifies to the classical MC estimate of  $O(\varepsilon^{-2})$ .

### 3.1.1 Practical implications

According to the cost bound in equation (3.22), the cost of computing high dimensional models with large  $d$ , and slow-to-converge models with small  $\alpha$ , is amplified. This amplification is shown through two examples:

- **High dimensional model:** The cost of computing a QoI with  $\alpha = 1$ , from a two dimensional ( $d = 2$ ) FE model, using sparse LU factorisation ( $\gamma = 3/2$ ) is  $O(\varepsilon^{-5})$ , whereas, in three dimensions ( $d = 3$ ), the cost is  $O(\varepsilon^{-8})$ , since  $\gamma = 2$  for sparse LU factorisation. This extra dimension causes the rate of convergence of the cost to increase by three. Figure 3.1a illustrates the impact of these cost convergence rates on the time taken to quantify uncertainty. If the simulation takes less than 10 s for the expectation of a QoI to converge to a tolerance of 0.1 for both problems, the two dimensional model would take six hours to converge to a tolerance of 0.01, whereas the three dimensional model would take 240 days.
- **Slow-to-converge model:** The method for adding uncertainty can cause the cost convergence rate to vary, see Section 2.5. Computing a QoI from a one dimensional ( $d = 1$ ) smooth FE

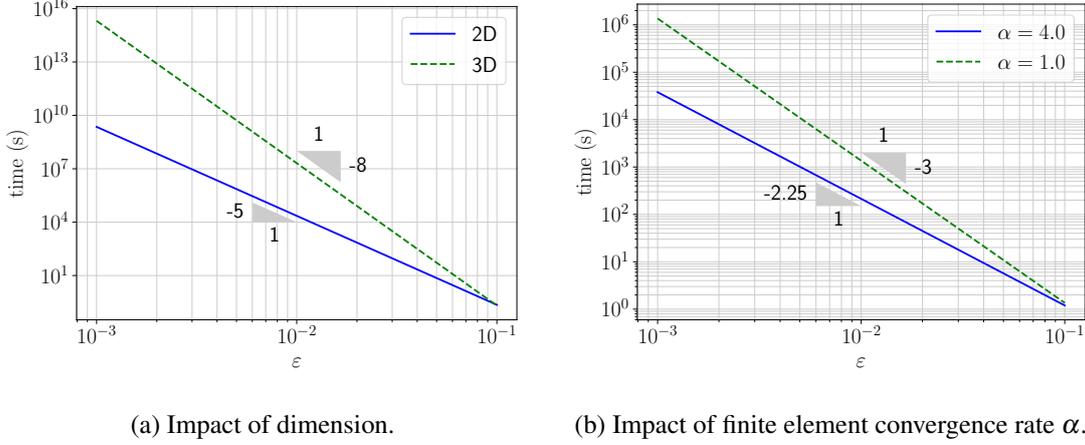


Figure 3.1: Illustration of practical implications of the MC method. Fig (a) shows the impact on cost of the dimensionality of the problem and Fig (b) shows the impact on cost of the finite element convergence rates.

formulation may have  $\alpha = 4$ . However, using a non-smooth formulation, for example caused by representing uncertainty with point masses, may result in  $\alpha = 1$ . Assuming  $\gamma = 1$ , the cost of the MC simulation changes from  $O(\varepsilon^{-2.25})$  to  $O(\varepsilon^{-3.0})$ . Figure 3.1b illustrates the impact of these cost convergence rates on the time taken to quantify uncertainty. If the simulation takes less than 10 s for the expectation of a QoI to converge to a tolerance of 0.1 for both models, the  $\alpha = 4$  model would take 4 minutes to converge to a tolerance of 0.01, whereas the  $\alpha = 1$  model would take 23 minutes.

### 3.1.2 Implementation

The implementation of the MC method in this thesis approximates the expectation  $\tilde{Q}_h$  and variance  $\mathbb{V}[\tilde{Q}_h]$  of the estimator of the QoI for an initial number of samples. This initial approximation is then used to compute the necessary number of samples from equation (3.20). Once the necessary samples have been computed, the size of the sampling error,  $e_V$  from equation (3.13), is checked to ensure  $e_V^2 \leq \varepsilon^2/2$ . If convergence is not obtained, a new approximation for the number of samples is computed using equation (3.20) and the new samples are obtained until the solution has converged.

This implementation assumes that the error tolerance is split equally between the sampling and discretisation error. For the MC method, the mesh fidelity is chosen *a priori* to ensure  $e_h^2 \leq \varepsilon^2/2$  (equation (3.15)). Pseudo-code for this implementation is included in Algorithm 1.

**Algorithm 1** MC implementation.

---

Choose mesh size,  $h$ , to ensure  $e_h^2 < \varepsilon^2/2$ .  
 Compute 10 initial samples.  
 Compute  $\tilde{Q}_{h,N}$  and  $\mathbb{V}[Q_h]$ .  
 Calculate  $e_V$  using equation (3.13).  
**if**  $e_V^2 > \varepsilon^2/2$  **then**  
   Compute new number of samples necessary using equation (3.20).  
   Compute additional samples required.  
   Compute new  $\tilde{Q}_{h,N}$  and  $\mathbb{V}[Q_h]$  from existing and extra samples.  
   Calculate new  $e_V$ .  
**end if**

---

**3.2 Control variate approach for accelerating the Monte Carlo method**

The MC method can be accelerated using control variates to reduce the variance of the estimator [52]. In the control variate method, an estimator  $Q_0(\omega_i)$  is introduced, which is close to  $Q_h(\omega_i)$ :

$$\tilde{Q}_{h,\text{diff}} = \frac{1}{N_0} \sum_{i=1}^{N_0} Q_0(\omega_i) + \frac{1}{N_1} \sum_{i=1}^{N_1} (Q_h(\omega_i) - Q_0(\omega_i)). \quad (3.23)$$

Clearly,

$$\mathbb{E}[\tilde{Q}_{h,\text{diff}}] = \mathbb{E}[Q_h]. \quad (3.24)$$

Following the root-mean-square error analysis in equation (3.13),

$$\begin{aligned} e^2 &= \mathbb{V}[\tilde{Q}_{h,\text{diff}}] + \mathbb{E}[\tilde{Q}_{h,\text{diff}} - Q]^2 \\ &= \underbrace{\frac{1}{N_0} \mathbb{V}[Q_0] + \frac{1}{N_1} \mathbb{V}[Q_h - Q_0]}_{e_V^2} + \underbrace{\mathbb{E}[\tilde{Q}_{h,\text{diff}} - Q]^2}_{e_h^2}. \end{aligned} \quad (3.25)$$

Again the error is split into two parts, the numerical error,  $e_h$ , and the sampling error,  $e_V$ . The fidelity of the model  $Q_h$  is chosen to achieve the desired numerical error, whereas, the control variate,  $Q_0$ , is used to reduce the number of the expensive models of  $Q_h$  that need to be solved. If  $\mathbb{V}[Q_h - Q_0]$  is small and samples of  $Q_0$  are inexpensive to compute, the overall cost of the estimator will be significantly lower than the traditional MC method because fewer samples of  $Q_h$  are required. An examples of control variates being used to reduce the computational expense of MC is Johannes [70].

**3.3 The multilevel Monte Carlo method**

The MLMC method [26, 51] generalises the two-level control variate approach to  $L$  levels. The presented derivation is close to that in Cliffe et al. [26].

### 3.3.1 Multilevel estimator

If the approximation of a QoI on a given level is denoted as  $Q_l$ , where  $l \in [0 \dots L]$  and indicates levels with decreasing  $h$ , the multilevel expectation can be expressed as

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{l=1}^L \mathbb{E}[Q_l - Q_{l-1}], \quad (3.26)$$

which is calculated by a coarse approximation of the expectation plus a telescopic sum of the differences in the expectation across the different levels. The expectation in equation (3.26) can be estimated by generalising equation (3.23):

$$\tilde{Q}_L = \frac{1}{N_0} \sum_{i=1}^{N_0} Q_0(\omega_i) + \sum_{l=1}^L \left( \frac{1}{N_l} \sum_{j=1}^{N_l} (Q_l(\omega_j) - Q_{l-1}(\omega_j)) \right). \quad (3.27)$$

It is important to ensure that  $\omega_j$ , or the realisation of the random field, is the same between the second and third terms in equation (3.27) to ensure that the difference in solution is only dependent on the numerical error and not from sampling two different distributions. Equation (3.27) is more compactly written as

$$\tilde{Q}_L = \sum_{l=0}^L \tilde{Q}_l, \quad (3.28)$$

where

$$\tilde{Q}_l = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} Q_0(\omega_i) & l = 0, \\ \sum_{l=1}^L \left( \frac{1}{N_l} \sum_{j=1}^{N_l} (Q_l(\omega_j) - Q_{l-1}(\omega_j)) \right) & l > 0. \end{cases} \quad (3.29)$$

The variance of the estimator in equation (3.28) is given by the sum of the variances across each level

$$\begin{aligned} \mathbb{V}[\tilde{Q}_L] &= \sum_{l=0}^L \mathbb{V}[\tilde{Q}_l], \\ &= \sum_{l=0}^L \frac{1}{N_l} V_l, \end{aligned} \quad (3.30)$$

where

$$V_l = \begin{cases} \mathbb{V}[Q_0] & l = 0, \\ \mathbb{V}[Q_l - Q_{l-1}] & l > 0. \end{cases} \quad (3.31)$$

The root-mean-squared error estimate of this estimator, from equation (3.7), is given by

$$e^2 = \underbrace{\sum_{l=0}^L \frac{1}{N_l} V_l}_{e_V^2} + \underbrace{\mathbb{E}[\tilde{Q}_L - Q]^2}_{e_L^2}, \quad (3.32)$$

where again the error is split between the sampling error,  $e_V$ , and the numerical error of the finest level,  $e_L$ . The error estimate in equation (3.32) is used to determine the fidelity of the meshes required and the number of samples at each level  $N_l$  that correspond to the minimum cost.

### 3.3.2 Determining the fidelity at level $L$

The fidelity of level  $L$  is determined by the numerical error

$$e_L^2 := \mathbb{E}[\tilde{Q}_L - Q]^2 \leq \epsilon_h^2. \quad (3.33)$$

Assuming the bound on  $h$  from equation (3.15) holds,

$$h_L \leq c_\alpha^{-\frac{1}{\alpha}} \epsilon_h^{\frac{1}{\alpha}}. \quad (3.34)$$

The constant  $c_\alpha$  may be known *a priori* or could be approximated during the computation from *a posteriori* estimates.

### 3.3.3 Determining the number of samples at each level

The number of samples at each level is determined by the sampling error. The cost of computing the multilevel estimator is

$$C = \sum_{l=0}^L N_l C_l, \quad (3.35)$$

where  $C_l$  is the cost of computing one sample of  $Q_l - Q_{l-1}$  for level  $l \geq 1$ , and  $C_0$  is the cost of computing one sample of  $Q_0$ .

To select the number of samples at each level, the total cost of the estimator is minimised for a given sampling error  $\epsilon_V$ . The functional of interest is

$$\mathcal{L}(N_0, \dots, N_L, \zeta^2) := \sum_{l=0}^L (N_l C_l + \zeta^2 N_l^{-1} V_l) - \zeta^2 \epsilon_V^2, \quad (3.36)$$

where  $\zeta \in \mathbb{R}$  is a Lagrange multiplier.

If  $N_l$  is treated as a continuous variable,  $\bar{N}_l$ , the optimum number of levels can be found by considering the directional derivative of equation (3.36):

$$\frac{\partial \mathcal{L}}{\partial \bar{N}_l} = C_l - \frac{\zeta^2 V_l}{\bar{N}_l^2} = 0. \quad (3.37)$$

Rearranging equation (3.37),

$$\bar{N}_l = \zeta V_l^{1/2} C_l^{-1/2}. \quad (3.38)$$

The Lagrange multiplier can then be found by substituting equation (3.38) into the sampling root-mean-squared error,  $\varepsilon_V$  from equation (3.32), for each level

$$\zeta = \varepsilon_V^{-2} \sum_{l=0}^L V_l^{1/2} C_l^{1/2}. \quad (3.39)$$

The optimum number of samples in the continuous setting is found by substituting equation (3.39) into equation (3.38)

$$\bar{N}_l = \varepsilon_V^{-2} \left( \sum_{l=0}^L V_l^{1/2} C_l^{1/2} \right) V_l^{1/2} C_l^{-1/2}. \quad (3.40)$$

Since the number of samples is a discrete value, equation (3.40) is bound by

$$N_l \geq c_N V_l^{1/2} C_l^{-1/2}, \quad (3.41)$$

where  $c_N$  is a constant, which is the same for all levels and chosen to ensure  $e_V^2 < \varepsilon_V^2$ . Detommaso et al. [33] has since undertaken further research on generalising the MLMC method for a setting where the level parameter can remain continuous.

Since equation (3.41) weights samples towards levels where the variance is largest, the greatest speed increases are for problems where the sampling error tolerance can be satisfied on coarse levels and the fine levels are only required to satisfy the discretisation error.

### 3.3.4 Cost estimates

The cost analysis requires a bound on the variance  $V_l$ . It is assumed that  $V_l$  decreases with increasing  $l$  such that

$$V_l \leq c_\beta h_l^\beta, \quad (3.42)$$

where  $\beta > 0$  and is constant and  $c_\beta$  is a constant that does not depend on  $h$ . Following Cliffe et al. [26], three different cost estimates can be obtained depending on the relationship between  $\beta$  and  $\gamma d$  and assuming that  $\alpha \geq 1/2 \min(\beta, \gamma)$ :

$$C(\tilde{Q}_h) \begin{cases} \leq c_1 \varepsilon^{-2} & \text{if } \beta > \gamma d, \\ \leq c_2 \varepsilon^{-2} (\log \varepsilon)^2 & \text{if } \beta = \gamma d, \\ \leq c_3 \varepsilon^{-2 - (\gamma d - \beta)/\alpha} & \text{if } \beta < \gamma d, \end{cases} \quad (3.43)$$

where  $c_1$ ,  $c_2$  and  $c_3$  are constants that do not depend on  $\varepsilon$ . This analysis demonstrates that the cost of the MLMC method is lower than the MC method, given in equation (3.17), for all three cases but is optimal if  $\beta > \gamma d$ . This corresponds to the variance converging faster than the cost of the solver.

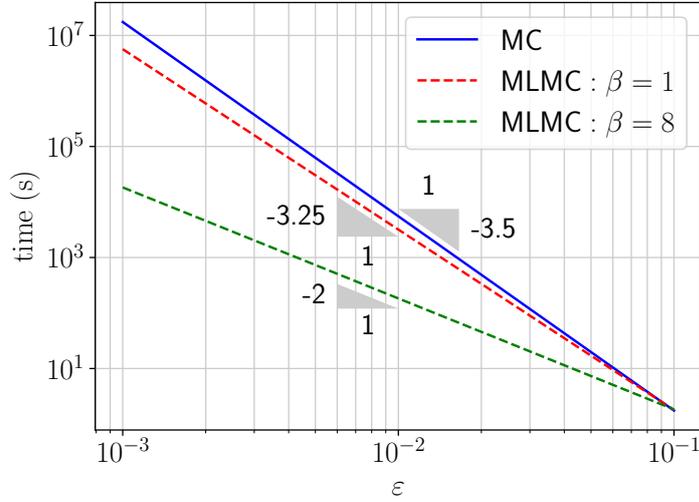


Figure 3.2: Illustration of practical implications of MLMC method for a three dimensional problem with  $\alpha = 4$  and  $\gamma = 2$ . The blue solid line shows the MC convergence rate and the red and green dashed lines show the MLMC convergence rates for  $\beta = 1$  and  $\beta = 8$ .

### 3.3.5 Practical implications

The bounds in equation (3.43) are illustrated for an idealised problem with  $d = 3$ ,  $\gamma = 2$  and  $\alpha = 4$ . From equation (3.22), the convergence rate of the MC method is  $O(\varepsilon^{-3.5})$  independent of  $\beta$ , whereas, the convergence rates of the MLMC method, from equation (3.43), depend on  $\beta$ . If  $\beta = 8$ , the order of convergence is  $O(\varepsilon^{-2})$ , since  $\beta > \gamma d$ . However, if  $\beta = 1$ , the order of convergence is  $O(\varepsilon^{-3.25})$ , since  $\beta < \gamma d$ . This means that if it takes approximately 1 s to compute all three examples for a tolerance of 0.1, it would take over 1.4 hours to converge to a tolerance of 0.01 with the MC method, 45 minutes to converge with the MLMC method for  $\beta = 1$ , but only 3 minutes with the MLMC method for  $\beta = 8$ .

A disadvantage of the MLMC method is that no information other than the expectation and variance is known about the distribution of the QoI. In contrast in the MC method, an expectation of the QoI is computed for each sample, so a histogram can be drawn that depicts the distribution.

### 3.3.6 Implementation

The implementation of the MLMC method in this thesis approximates  $\mathbb{E}[\tilde{Q}]$  for an initial number of samples and levels. The parameters  $V_l$ ,  $C_l$  and the quasi-optimal number of samples,  $N_l$  given in equation (3.41), are then approximated from this data initially assuming  $c_N = 1$ . Extra samples are computed until the number of samples is greater than or equal to  $N_l$ . If fewer samples are required than computed initially, this initialisation stage can miss represent the cost of the computation.

Once enough samples have been computed, the total error of the simulation is checked to ensure that it is smaller than the given tolerance. In this implementation, it is assumed that the error is equally distributed between the sampling and numerical error:  $\varepsilon_V^2 = \varepsilon_h^2 = \varepsilon^2/2$ .

- The numerical error convergence is checked to ensure that  $e_h^2 \leq \varepsilon^2/2$  and enough levels are used in the simulation. Then,  $\tilde{Q}^{l+1}$  is approximated using  $\alpha$ , which is estimated from the samples. If  $(\tilde{Q}^{l+1})^2 > \varepsilon^2/2$ , the numerical error is too large and an extra level is required. The parameter  $V_{l+1}$  and  $C_{l+1}$  are then estimated from the data, and the quasi-optimum number of samples at this new level is estimated. Once the quasi-optimum numbers of samples are obtained, convergence is checked again.
- If the numerical error has converged, the sampling error is checked to ensure  $e_V^2 \leq \varepsilon^2/2$ . If this criterion is not met, the value of  $c_N$  is increased. The quasi-optimal number of samples is estimated with the new  $c_N$ , and the extra samples are computed. Convergence is then checked again.

Once both the numerical and sampling errors have converged,  $\mathbb{E}[\tilde{Q}_L]$  and  $\mathbb{V}[\tilde{Q}_L]$  have converged for the given tolerance. Pseudo-code for this implementation is included in Algorithm 2.

---

**Algorithm 2** MLMC implementation.

---

Choose initial mesh,  $h_0$ , initial number of levels,  $L$ , maximum number of levels,  $L_{\max}$ , initial  $c_N$  and initial number of samples at each level,  $N_l$ .  
 Compute initial samples at each level.  
 Compute  $Q_l, V_l, C_l$ .  
 Calculate quasi-optimum number of samples  $N_l$ .  
**if** Number of samples computed at each level,  $\tilde{N}_l < N_l$  **then**  
   Compute extra samples.  
   Compute  $Q_l, V_l, C_l$ .  
   Calculate  $N_l$ .  
**end if**  
 Calculate  $\alpha, \beta$  and  $\gamma$  using linear regression.  
 Calculate  $e_h$  and  $e_V$   
**while**  $e_h^2 > \varepsilon^2/2$  and  $L < L_{\max}$  **do**  
   Add new level,  $L = L + 1$ .  
   Estimate  $V_L, C_L$  and  $N_L$  from previous data.  
   Compute extra samples.  
   Compute new  $Q_l, V_l, C_l, \alpha, \beta, \gamma$  and  $e_h$ .  
**end while**  
**while**  $e_h^2 > \varepsilon^2/2$  **do**  
   Increase  $c_N$ .  
   Calculate new  $N_l$ .  
   Compute extra samples.  
   Compute new  $Q_l, V_l, C_l, \alpha, \beta, \gamma$  and  $e_V$ .  
**end while**

---

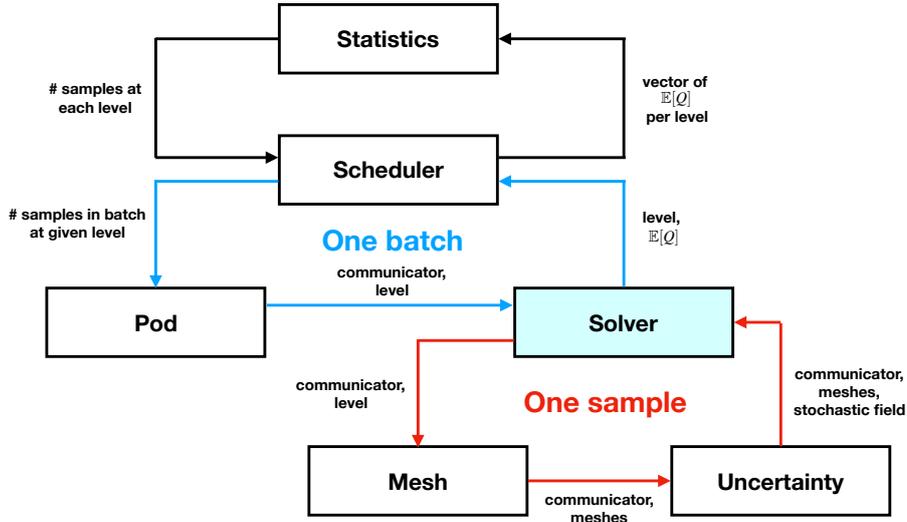


Figure 3.3: PodS module schematic.

### 3.4 Software library implementation of Monte Carlo based methods

A library, podS [124], has been specifically written to implement the MC and MLMC algorithms described in Sections 3.1.2 and 3.3.6. This is valuable because great reductions in computation time are possible from sampling the multiple independent realisations of the PDE in parallel since MC is an embarrassingly parallelisable problem.

PodS manages this parallelisation for the DOLFIN module of the FEniCS library [4], the chosen finite element package for this thesis. The name podS is a play on the name DOLFIN because a group of dolphins is called a pod and the 'S' indicates that stochastic computations are being solved. The majority of this library has been written by the author for this thesis, but the scheduling and pod module, discussed later in this section, was developed by Hadfield for a Masters project at the University of Cambridge [54]. The library is open source with a LGPLv3 license and is sufficiently generalised for other researchers to use it to implement the MC and MLMC methods efficiently for their applications.

The podS library is written in C++ for speed, portability and the availability of the MPI libraries [90]. It consists of seven different modules: statistics, solver, uncertainty, mesh, pod, scheduling and log. The implementation of the library is discussed in Section 3.4.1, with further detail provided in Appendix B. The outcomes of using this library are then presented in Section 3.4.2.

#### 3.4.1 Module description

Figure 3.3 shows the relationships between the modules in podS. The solver module is indicated in blue because this is the only module that includes user written code. This library can be simplified into three loops, with the log module interacting with all the other modules.

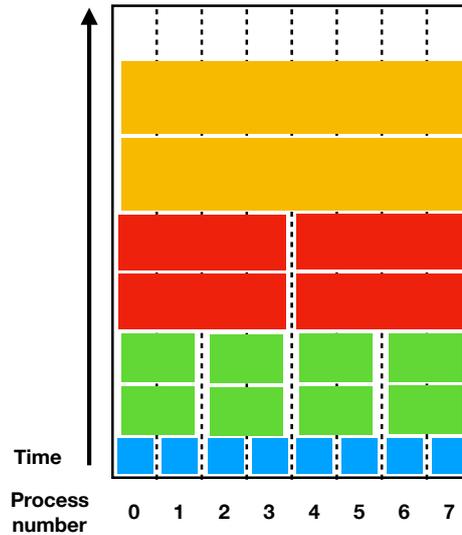


Figure 3.4: Schematic for current scheduler. The blue boxes correspond to level zero simulations of  $\mathbb{E}[Q]$ , green level one, red level two and orange level three.

The first loop, which is indicated with black arrows, contains the statistics and scheduler modules. This loop uses an initial estimate, or previous data, to determine how many samples should be computed at each level. The library then schedules these samples into batches. A simple scheduler is used, which runs batches of the problems with increasing numbers of degrees of freedom, until all the necessary samples have been computed; see Figure 3.4 for a schematic of the current scheduler. More complex schedules have been proposed by Gmeiner et al. [53].

The second loop, indicated with blue arrow, contains the scheduler, pod and solver modules. This loop is used to solve each scheduled batch, which contains multiple samples of the same level, in turn. It ensures that no more samples than will fit in the available memory are scheduled at the same time, which may result in more low fidelity modules being scheduled at the same time than high fidelity modules.

The third loop, indicated with red arrows, contains the solver, mesh and uncertainty modules. This loop solves an individual sample of the PDE in parallel using split MPI communicators. Each of the processes is seeded with its own random number generator to ensure that independent realisations of the random field are used in each sample.

### 3.4.2 Outcomes

PodS successfully enables samples of the MC and MLMC algorithm to be computed in parallel. This enables great savings in time, and so computational expense, to be obtained. Figure 3.5 shows the time taken to compute uncertainty in the one dimensional wave equation to the same tolerance using the MC and MLMC methods; here the solver is treated as a black box.

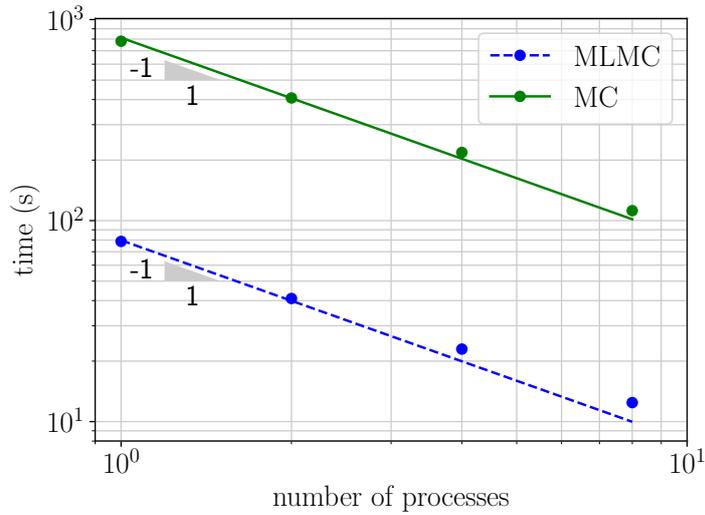


Figure 3.5: PodS run time for the MC and MLMC simulations in parallel on varying numbers of cores for a fixed  $\epsilon$ .

Two key features of the podS implementation of the MC and MLMC methods can be seen from Figure 3.5. First, the rate of convergence of the cost of the MLMC method is generally faster than the MC method. Once the number of samples computed is greater than the initial start up number, the reduced costs occur because the cost complexity of the MLMC method is lower, see Section 3.3.5. The exact speed up is dependent on the cost, variance and error convergence rates of the finite element models. In this example, it is approximately ten times quicker to compute the expectation of the QoI with the MLMC method instead of the MC method.

The second feature is that the simulation time is approximately halved when the number of processes is doubled. This is expected because each sample is solved independently of the others. The MLMC method does not quite achieve this optimal speed up when using more than four processes to quantify uncertainty in this black box problem. This is probably due to more processes being scheduled to solve some problems than necessary and the overhead of communicating the expectations between all processes.

## Chapter 4

# Structural vibration problems under uncertainty

Small perturbations in engineering structures can affect their natural frequencies, vibration modes and energy density. Unwanted resonance, resulting from small variations in the natural frequencies, can reduce performance life or in some cases cause catastrophic failure. However for others systems, it is necessary to design the structure to resonate with maximum energy. Therefore, it is important to quantify the impact of perturbations on system behaviour so that designs, which are robust to the realities of manufacture and operation, can be found.

Turbomachinery blade mistuning is one example where quantifying uncertainty is important for vibrating structures. Individual blades have small differences in structural properties due to manufacturing, material tolerances or in-service wear. These break the cyclic symmetry of the system and thus change its behaviour. This results in some blades experiencing deflections and stresses that are much larger than expected, which can cause failure before the design life of the blade is reached [12].

It is usually considered impractical to use traditional MC methods to quantify uncertainty in random vibration problems due to the computational cost [2, 21, 34, 45, 129]. Instead, probabilistic methods, such as statistical energy analysis, are often used. In this chapter, the MLMC method is demonstrated as a viable alternative method for quantifying uncertainty in large-scale structural vibration problems.

An outline of this chapter is as follows; the FE formulations for the wave, Kirchhoff–Love plate and linearised elastodynamics equations are presented in Section 4.1 and the eigensolvers used are discussed in Section 4.2. Then in Section 4.3, the expectation and variance of the first eigenvalues are computed for the three examples, and in Section 4.4, the expectation and variance in the energy density of the Kirchhoff–Love plate is computed. Conclusions are drawn in Section 4.5.

## 4.1 Finite element formulations

In this section, the FE formulations for the three models are presented: the wave equation, the Kirchhoff-Love plate equation and the linearised elasticity equation. In keeping with the notation in Section 2.1, the spatial domain for each problem is denoted by  $D \subset \mathbb{R}^d$ , where  $d$  is the number of spatial dimensions, and  $D$  is triangulated into non-overlapping cells,  $K$ . The boundary of  $D$  is denoted by  $\partial D$ .

### 4.1.1 Wave equation

The time-harmonic wave equation and associated boundary conditions in one spatial dimension read given  $\tau > 0$  and  $\rho \in W(D)$ , find the mode shapes  $u \neq 0$  and corresponding eigenvalues  $\lambda$  such that

$$-\frac{d}{dx} \tau \frac{du}{dx} = \lambda \rho u \quad \text{in } D, \quad (4.1a)$$

$$\tau \frac{du}{dn} = 0 \quad \text{on } \partial D, \quad (4.1b)$$

where the density field,  $\rho$ , is sampled from a space of random fields and  $W$  is an appropriate function space. The FE formulation of problem (4.1a) is to find  $u_h, \lambda_h \in V_h \times \mathbb{R}^+$  such that

$$\int_D \tau \frac{du_h}{dx} \frac{dv_h}{dx} dx = \lambda_h \int_D \rho u_h v_h dx \quad \forall v_h \in V_h, \quad (4.2)$$

where

$$V_h := \{u_h \in H^1(D), u_h|_K \in P_m(K) \forall K \in \mathcal{D}\}, \quad (4.3)$$

and  $P_m(K)$  is a standard Lagrange polynomial finite element space of degree  $m \geq 1$ .

### 4.1.2 Kirchhoff–Love plate equation

For a linear isotropic elastic Kirchhoff–Love plate, the problem and associated boundary conditions read, given a mass density field  $\rho \in W(D)$ , find the vibration modes  $u \neq 0$  and corresponding eigenvalues  $\lambda$  that satisfy

$$\nabla^4 u = \lambda \frac{2a\rho}{K} u \quad \text{in } D, \quad (4.4a)$$

$$\nabla u \cdot \mathbf{n} = 0 \quad \text{on } \partial D, \quad (4.4b)$$

$$\nabla(\nabla^2 u) \cdot \mathbf{n} = 0 \quad \text{on } \partial D. \quad (4.4c)$$

where  $2a$  is the plate thickness and  $K$  is the bending stiffness. Using the Ciarlet–Raviart method [23], the FE problem is formulated by splitting the operator in equation (4.4a) according to

$$w := -\nabla^2 u \quad (4.5)$$

$$-\nabla^2 w = 2a\lambda \frac{\rho}{K} u. \quad (4.6)$$

This leads to the problem, find  $u_h, w_h, \lambda_h \in V_h \times V_h \times \mathbb{R}^+$  such that

$$-\int_D w_h z_h \, dx + \int_D \nabla u_h \cdot \nabla z_h \, dx = 0 \quad \forall z_h \in V_h, \quad (4.7a)$$

$$\int_D \nabla w_h \cdot \nabla v_h \, dx = \frac{2a\lambda_h}{K} \int_D \rho u_h v_h \, dx \quad \forall v_h \in V_h, \quad (4.7b)$$

where  $V_h \in H^1(D)$ .

### 4.1.3 Linearised elastodynamics equation

The time harmonic formulation of the linear elastic wave equation and corresponding boundary condition read: given the mass density  $\rho \in W(D)$ , find mode shapes  $\mathbf{u} \neq \mathbf{0}$  and their corresponding eigenvalues  $\lambda$  such that

$$-\nabla \cdot \boldsymbol{\sigma} = \lambda \rho \mathbf{u} \quad \text{in } D, \quad (4.8)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \partial D, \quad (4.9)$$

where  $\boldsymbol{\sigma}(\mathbf{u}) = 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) + \Lambda \text{tr}(\boldsymbol{\varepsilon}(\mathbf{u})) \mathbf{I}$  is the tensor stress,  $\mu$  and  $\Lambda$  are the Lamé parameters, and  $\boldsymbol{\varepsilon}$  is the strain tensor. The FE formulation of problem (4.8) is to find  $\mathbf{u}_h, \lambda_h \in \mathbf{V}^h \times \mathbb{R}^+$  such that

$$\int_D \boldsymbol{\sigma}_h(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{v}_h) \, dx = \lambda_h \int_D \rho \mathbf{u}_h \cdot \mathbf{v}_h \, dx \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \quad (4.10)$$

where  $\mathbf{V}_h = [V_h]^d$  and  $V_h \in H^1(D)$ .

## 4.2 Eigensolvers

The choice of eigensolver is critical for performance, particularly in three dimensions, where the MC and MLMC algorithms demand multiple large solves of the matrix equation in (2.12). Two different eigenvalue solvers, which are used to solve the FE formulations above, are the preconditioned Jacobi–Davidson [109] method and the Krylov–Schur method [111]. The Jacobi–Davidson method uses the conjugate gradient iterative method with algebraic multigrid preconditioning. In contrast, the Krylov–Schur method applies shift-and-invert spectral transformation to the eigenvalue problem, which target specific spectral regions close to the natural frequency of interest. Then Cholesky factorisation is used to directly solve the eigenvalue problem.

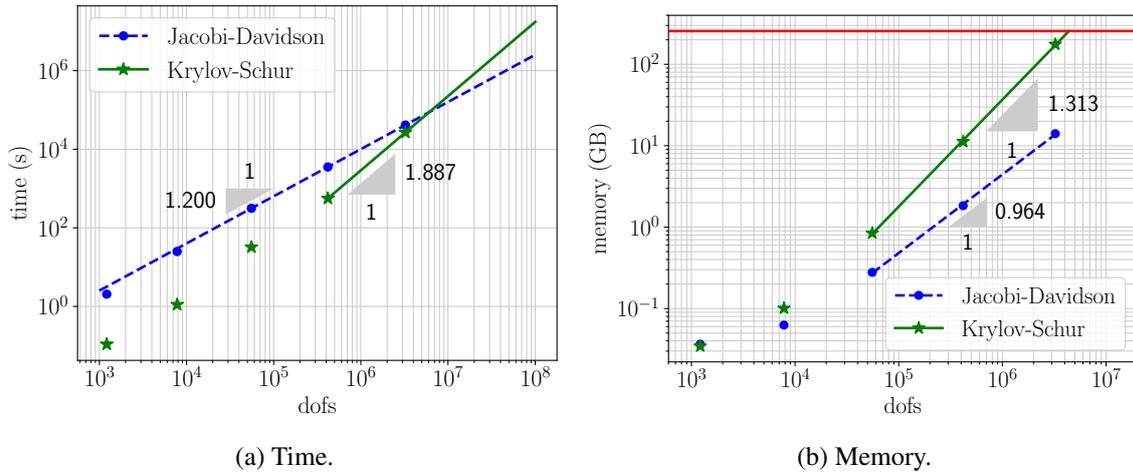


Figure 4.1: Comparison of computation times, Fig (a), and memory requirements, Fig (b), of the Krylov–Schur and Jacobi–Davidson eigenvalue solvers. The smallest 20 eigenvalues were computed for problems with increasing degrees of freedom. No parallelisation was used; all computations were conducted with one process.

The time and memory requirements for individual three dimensional eigenvalue problem solves, with different numbers of degrees of freedom, have been compared in Figure 4.1 for smallest 20 eigenvalues found using the Krylov–Schur and Jacobi–Davidson solvers in the SLEPc library [127]. The data in this chapter is computed using a workstation with two Intel Xeon CPU E5-2690 v4 processors with 14 cores each, a clock speed of 2.60 GHz and 256 GiB RAM. No parallelisation was used to give an indication for timings of computations on single processes.

Figure 4.1a indicates that the Krylov–Schur solver is faster than the Jacobi–Davidson solver for the problem sizes tested. However, the Krylov–Schur solver has higher complexity than the Jacobi–Davidson solver, which is showed by the steeper gradient of the Krylov–Schur line, meaning that the Krylov–Schur solver will be more expensive for larger problems. Figure 4.1b shows the memory usage of the two methods for solving the eigenvalue problem. The Jacobi–Davidson solver requires considerably less memory than the Krylov–Schur solver, but as previously stated, is more computationally expensive for smaller problems. Therefore, the Krylov–Schur method is used for the problems with smaller memory requirements (wave and Kirchhoff–Love plate), whereas the Jacobi–Davidson method is used for the larger problem (elastodynamics).

### 4.3 Eigenvalues of structures

So that the impact of the uncertain mass density can be quantified using the MC and MLMC methods implemented in parallel using podS [122], the three FE formulations described in Section 4.1 are coded in C++ using FEniCS [4], PETSc [10] and SLEPc [127]. The code for this chapter is available online [119].

The  $j^{\text{th}}$  eigenvalues,  $\lambda_j$ , of the structure is chosen as the first QoI. A relative tolerance  $\bar{\epsilon}$  is introduced to ensure that the error scales with the QoI. In the eigenvalue examples below, the total error is scaled by the expectation of the QoI:

$$\frac{e}{\mathbb{E}[\tilde{Q}_L]} < \bar{\epsilon}, \quad (4.11)$$

where  $e$  is the total error in the simulation from equation (3.13) and  $\mathbb{E}[\tilde{Q}_L]$  is the expectation of the MLMC estimator of the QoI.

### 4.3.1 Wave equation

In the first example, the impact of 30 uniformly randomly placed point masses is quantified for the first four eigenvalues of the one dimensional wave equation in (4.2). The domain is unit length,  $a = 1$ , with  $\rho = 8000\text{kg/m}^3$ ,  $\tau = 200\text{ N}$  and the maximum mass of each point is 10% of total mass of the unperturbed string. An FE space of polynomial degree two is used, so  $m = 2$  in equation (4.3). The coarsest mesh in the MLMC simulations has 30 elements, and the number of elements,  $n$ , doubles for each level of refinement. The value of  $h$  for each mesh is  $a/n$ . This resolution was chosen to ensure the eigenvalues were sufficiently accurate but not too fine to limit the benefit of the MLMC variance reduction. As mentioned in the MC implementation, the number of elements for the MC mesh used is chosen *a priori* to ensure that  $e_h < \epsilon_h$ .

Figure 4.2 shows the probability density distributions of the expectations for the first four non-zero eigenvalues, computed using both the MC and MLMC methods, for a relative error tolerance of  $\bar{\epsilon} = 10^{-3}$ . There is good agreement between the two methods for each eigenvalue, but the difference between the expectation and variance of the two methods is slightly larger for the higher eigenvalues. This is expected because, when the eigenvalues are converged to the same relative tolerance, the larger eigenvalues have a greater absolute error. As explained earlier, a histogram cannot be drawn from the MLMC results since only an expectation and variance are computed unlike MC where each individual sample QoI is known.

#### Prediction of theoretical complexities of the MC and MLMC method

The convergence parameters ( $\alpha, \beta$  and  $\gamma$ ), which were introduced in Section 3.3, can be estimated from the individual samples to predict the theoretical convergence rates of both the MC and MLMC method. The parameter  $\alpha$  is the finite element convergence rate,  $\beta$  is the convergence rate of the variance and  $\gamma$  is the complexity of the solver. These constants are shown in Figures 4.3a to 4.3c for the first stable eigenvalue of the wave equation, in (4.2), for  $\bar{\epsilon} = 10^{-4}$ . The data point from the largest elements is not used to compute the gradients in Figure 4.3 because samples for level zero only involve computations on the coarsest mesh instead of two different meshes, see equation (3.27).

It is immediately obvious from Figure 4.3a that the FE convergence rate,  $\alpha < 4$ , which is the expected *a priori* error estimate for quadratic elements with sufficiently regular data given in

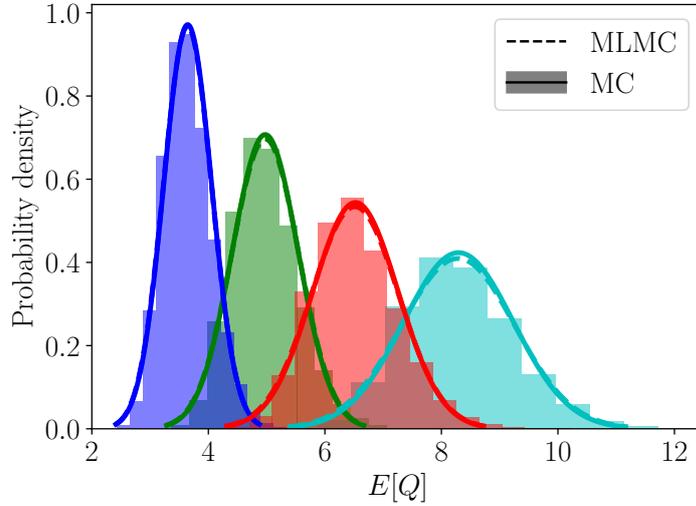


Figure 4.2: Comparison of the MC and MLMC probability distributions for the first four non-zero eigenvalues of the wave equation for  $\bar{\epsilon} = 10^{-3}$ . The mesh used in the MC simulation had 514 elements.

equation (2.8). As discussed in Section 2.5, this sub-optimal convergence rate results from reducing the regularity of the FE formulation by adding singular point masses. The impact of using point masses on the finite element error convergence rates of the three structural vibration examples is considered further in Appendix C.

It can also be seen from Figures 4.3b and 4.3c that  $\beta > \gamma d$ . This means that the MLMC samples should be weighted heavily towards the coarser meshes and the complexity of the method should be  $O(\epsilon^{-2})$ , see Section 3.3.4. In contrast, the MC complexity is  $O(\epsilon^{-2-\gamma d/\alpha})$ , from equation (3.22), or  $O(\epsilon^{-2.97})$  for this problem.

### Comparison of theoretical and computed MC and MLMC complexities

The theoretical and simulation complexities are compared in Figure 4.4 by computing the cost of the MC and MLMC simulations for varying relative tolerances using eight cores of the system described above. As expected, the gradients of the slopes match the theoretical complexity values very well. There is not much difference in the timings between the two methods for the large tolerances and it is noted that sometimes the MLMC method is more expensive than the MC method. This is due to the overhead of setting up the MLMC method, since initial samples are required from all four levels. The difference between the time taken for each method increases as the tolerance tightens due to the difference in complexities of the two methods. For  $\bar{\epsilon} = 10^{-4}$  there is nearly a two order of magnitude predicted difference in time. The MLMC method takes 38 minutes, whereas the MC method is predicted to take over one day.

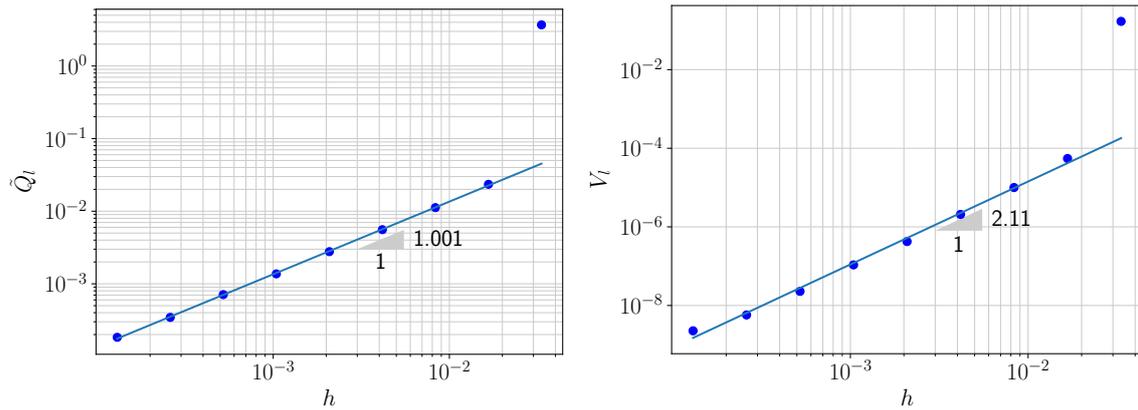
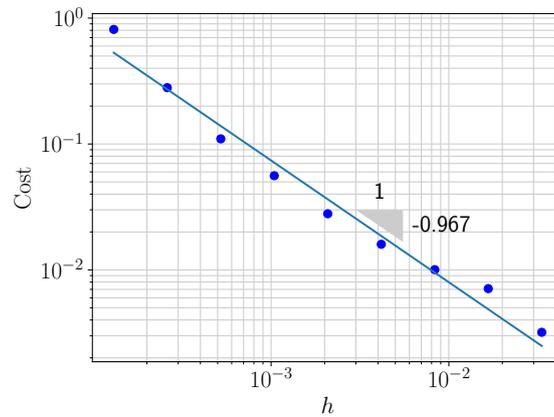
(a)  $\alpha$ , finite element convergence rate.(b)  $\beta$ , variance convergence rate.(c)  $\gamma$ , solver complexity.

Figure 4.3: MLMC convergence rates for the first stable eigenvalue of the wave equation. Fig (a) shows the finite element convergence rate  $\alpha = 1.001$ , Fig (b) shows the convergence rate of the variance  $\beta = 2.113$  and Fig (c) shows the convergence rate of the cost of solving each different level  $\gamma = 0.967$ , for  $\bar{\epsilon} = 10^{-4}$ .

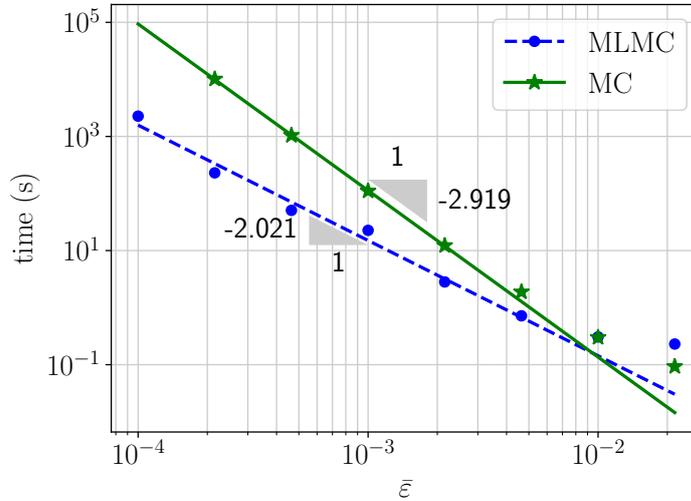


Figure 4.4: Comparison of time cost for the MC and MLMC simulations for the wave equation for different relative tolerances on eight cores.

The number of elements and samples necessary for each of the MC simulations in Figure 4.4 are shown in Table 4.1. The changes in order of magnitude are highlighted in grey. The number of samples increases with  $O(\epsilon^{-2})$ , as expected from the traditional MC convergence. This means that 100 times more samples are required for one order of magnitude decrease in the tolerance. The added complexity of the MC method comes from sampling a finer mesh, which is required to meet the numerical error constraints. This is shown in the theory from Section 3.1.

The number of samples computed at each level of the MLMC simulations in Figure 4.4 are shown in Table 4.2. No required samples at a given level is denoted by ‘-’. Initially 10 samples are computed at each level so convergence parameters can be calculated, even though theory may predict that fewer are needed. When the relative tolerance is decreased by an order of magnitude, there is an approximate

Table 4.1: Number of elements and samples required for the MC simulation to ensure convergence of the wave equation to a specified relative tolerance.

$\bar{\epsilon}$	Number of elements	Number of samples
$2.15 \times 10^{-2}$	25	100
$1.0 \times 10^{-2}$	53	457
$4.64 \times 10^{-3}$	113	2095
$2.15 \times 10^{-3}$	241	9693
$1.0 \times 10^{-3}$	514	44915
$4.64 \times 10^{-4}$	1097	208368
$2.15 \times 10^{-4}$	2341	966764
$1.0 \times 10^{-4}$	4995	4486988 <sup>1</sup>

<sup>1</sup> This value is extrapolated.

Table 4.2: Number of samples computed at each level for MLMC simulations of wave equation for specified relative tolerances.

$\bar{\epsilon}$	MLMC level index								
	0	1	2	3	4	5	6	7	8
$2.15 \times 10^{-2}$	34	10	10	10	-	-	-	-	-
$1.0 \times 10^{-2}$	191	10	10	10	-	-	-	-	-
$4.64 \times 10^{-3}$	1136	10	10	10	-	-	-	-	-
$2.15 \times 10^{-3}$	6100	76	29	10	3	-	-	-	-
$1.0 \times 10^{-3}$	53 187	668	199	48	11	-	-	-	-
$4.64 \times 10^{-4}$	123 894	1539	476	141	53	18	6	-	-
$2.15 \times 10^{-4}$	578 246	7326	2113	691	231	77	26	7	-
$1.0 \times 10^{-4}$	5 361 364	65 547	19 327	5 995	1 858	673	199	61	27

Table 4.3: Number of samples computed for the 25th eigenvalue of the wave equation using the MC and MLMC method.

$\bar{\epsilon}$	MLMC level index							MC	
	0	1	2	3	4	5	6		7
$1 \times 10^{-4}$	40 204	5 733	1 660	616	251	88	24	6	26 301

100 times increase in number of samples because the complexity is  $O(\bar{\epsilon}^2)$ . This is only noticeable when more than 10 initial samples are required. It is noted that the number of samples for  $\bar{\epsilon} = 10^{-2}$  is quite low. This is not of concern because the point does not lie on the complexity trend line and it is very quick to compute. This means that other stages, such as set-up time, could be dominating the timing instead of the solve time.

### Comparison of 25th eigenvalue

It is also possible to use the MLMC method to compute higher eigenvalues. The highest eigenvalue that can be computed depends on the fidelity of the coarsest mesh, or more specifically the number of degrees of freedom in the FE model.

The probability density functions for the 25th eigenvalue are shown in Figure 4.5 for both the MC and MLMC methods and  $\bar{\epsilon} = 10^{-4}$ . The number of elements in the coarsest MLMC mesh is increased to 100 to ensure accuracy in the higher eigenvalues and the MC method requires a mesh with 12 800 elements. It is noticeable that although there is good agreement between the expectation and variance of the 25th eigenvalue, they no longer exactly agree despite the solution converging to a relative tolerance an order of magnitude smaller than in Figure 4.2. This is because the absolute error is greater for the 25th eigenvalue than the first to fourth eigenvalues. The number of MLMC samples at each level and MC samples are shown in Table 4.3.

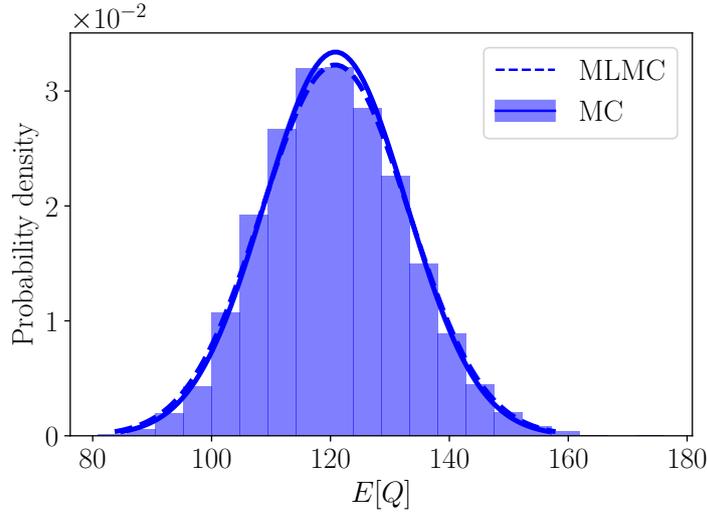


Figure 4.5: Comparison of the MC and MLMC probability distribution functions for the 25th eigenvalue of the wave equation for  $\bar{\epsilon} = 10^{-4}$ .

### 4.3.2 Kirchhoff–Love plate equation

In the second example, the impact of 30 uniformly randomly placed point masses is quantified for the two dimensional Kirchhoff–Love plate, described in equation (4.7). A rectangular domain, with dimensions of  $x = 0.1 \text{ m} \times y = 0.13 \text{ m}$ , is used to represent the plate. The maximum magnitude of each point mass is 1% of the total mass of the unperturbed plate,  $E = 70 \text{ GPa}$ ,  $\nu = 0.32$ ,  $\rho = 2700 \text{ kg/m}^3$  and  $2a = 0.001 \text{ m}$ . Second order polynomials are chosen for the function space so  $m = 2$  in equation (4.3). The coarsest mesh in the MLMC simulations has 50 elements and each subsequent level has four times as many elements. The value of  $h = \sqrt{(x/n_x)^2 + (y/n_y)^2}$ , where  $n_{x,y}$  describes the number of elements in each direction. The number of the elements in the MC samples again ensures that  $e_h < \epsilon_h$ .

The probability distributions for the expectation of the first and second eigenvalues are shown in Figure 4.6 for the MC and MLMC methods, which have converged to  $\bar{\epsilon} = 10^{-3}$ . These are not plotted on the same graph due to the different magnitude of the eigenvalues. The computed results cannot be distinguished since the solid and dashed lines lie on top of each other.

The MLMC convergence parameters are shown in Figure 4.7 for  $\bar{\epsilon} = 10^{-5}$ . It was possible to obtain tighter tolerances for the Kirchhoff–Love plate because  $\alpha$  is larger since equation (4.4a) is fourth order and, as shown in Appendix C, is less impacted by adding point masses. Again  $\beta > \gamma d$ , so most of the computational effort is on the coarser levels. This means that the MLMC method has a predicted complexity of  $O(\epsilon^{-2})$  and the MC method has a predicted complexity of  $O(\epsilon^{-2.68})$ .

The time taken to compute the expectation of the first two eigenvalues is compared in Figure 4.8 for the MLMC and MC methods, using 16 cores of the system described above. Again, the gradients of the lines match the theoretical complexities well and it is shown to be much quicker to compute samples using the MLMC method.

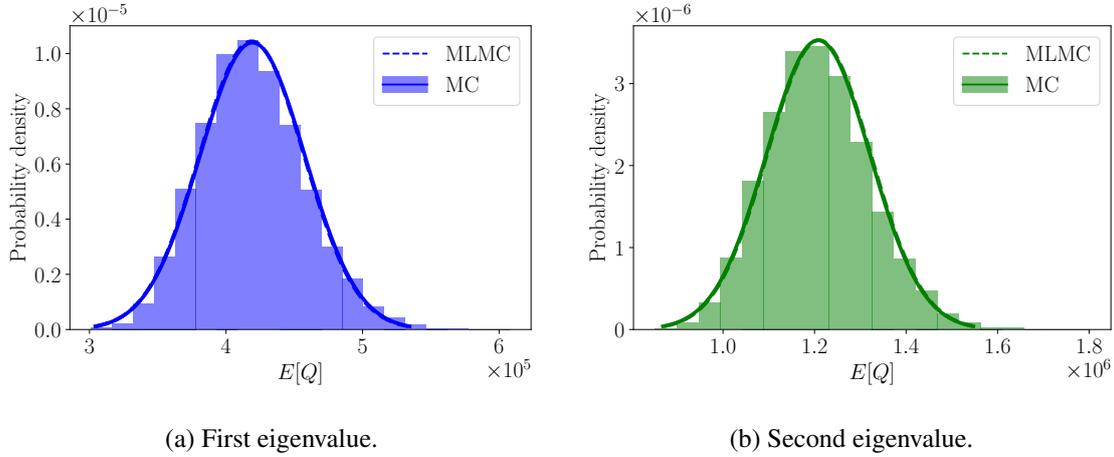


Figure 4.6: Comparison of the MLMC and MC probability distribution functions for the first and second eigenvalues of a two dimensional Kirchhoff–Love plate for  $\bar{\epsilon} = 10^{-3}$ . Fig (a) shows the first eigenvalue and Fig (b) shows the second eigenvalue. Here the two lines are indistinguishable because the dashed and solid line lie on top of each other.

Table 4.4: Number of elements and samples required for the MC simulation to ensure convergence of the Kirchhoff–Love plate equation to a specified relative tolerance.

$\bar{\epsilon}$	Number of elements	Number of samples
$1.0 \times 10^{-3}$	98	17 695
$4.64 \times 10^{-4}$	200	82 172
$2.15 \times 10^{-4}$	288	381 311
$1.0 \times 10^{-4}$	578	1 769 829
$4.64 \times 10^{-5}$	1058	8 214 572
$2.15 \times 10^{-5}$	2178	38 131 100 <sup>1</sup>
$1.0 \times 10^{-5}$	4232	176 982 900 <sup>1</sup>

<sup>1</sup> This value is extrapolated.

Using the MLMC method is one order of magnitude faster than using the MC method for  $\bar{\epsilon} = 10^{-4}$  and the MLMC method is predicted to be over one and a half orders of magnitude faster for  $\bar{\epsilon} = 10^{-5}$ . This corresponds to the MLMC method taking approximately 10 minutes to compute the result for  $\bar{\epsilon} = 10^{-4}$  instead of MC taking over 1.25 hours. For  $\bar{\epsilon} = 10^{-5}$ , the MLMC method takes approximately 17 hours, whereas MC is predicted to take approximately 28 days. This suggests that as it becomes prohibitively expensive to use the MC method, great savings in cost can be found by using the MLMC method.

The number of elements and samples required for each of the MC computations in Figure 4.8 are shown in Table 4.4 and the number of samples at each level computed for MLMC simulations are shown in Table 4.5. Again grey highlighting is used to indicate a change in order of magnitude. The number of samples is found to increased 100 fold when the tolerance is increased by an order of magnitude. Table 4.5 also illustrates the impact of large  $\beta$  values. This is because many more samples

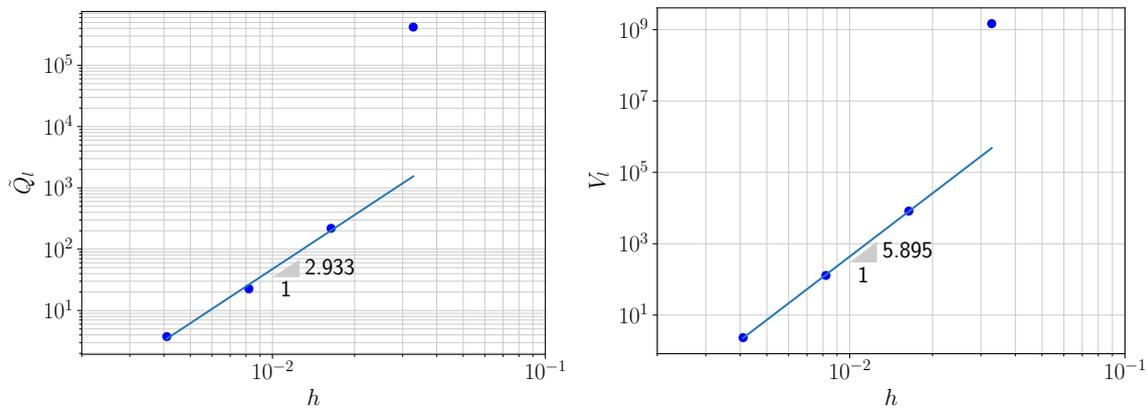
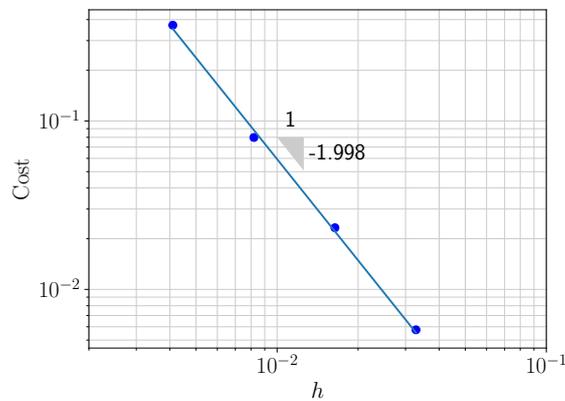
(a)  $\alpha$ , finite element convergence rate.(b)  $\beta$ , variance convergence rate.(c)  $\gamma_d$ , solver cost convergence rate.

Figure 4.7: MLMC convergence rates for the first eigenvalue of the Kirchhoff–Love plate equation for  $\bar{\epsilon} = 10^{-5}$ . Fig (a) shows the finite element convergence rate  $\alpha = 2.933$ , Fig (b) shows the convergence rate of the variance  $\beta = 5.895$  and Fig (c) shows the convergence rate of the cost of solving each different level  $\gamma = 1.998/2 = 0.999$ .

Table 4.5: Number of samples computed at each level for MLMC simulations of Kirchhoff–Love plate for specified relative tolerances.

$\bar{\epsilon}$	MLMC level index			
	0	1	2	3
$1.0 \times 10^{-3}$	16624	18	10	10
$4.64 \times 10^{-4}$	77826	102	10	10
$2.15 \times 10^{-4}$	368362	429	29	10
$1.0 \times 10^{-4}$	1687808	1937	92	10
$4.64 \times 10^{-5}$	7784733	9435	547	48
$2.15 \times 10^{-5}$	36757259	43172	2684	207
$1.0 \times 10^{-5}$	168316056	198827	12301	1041

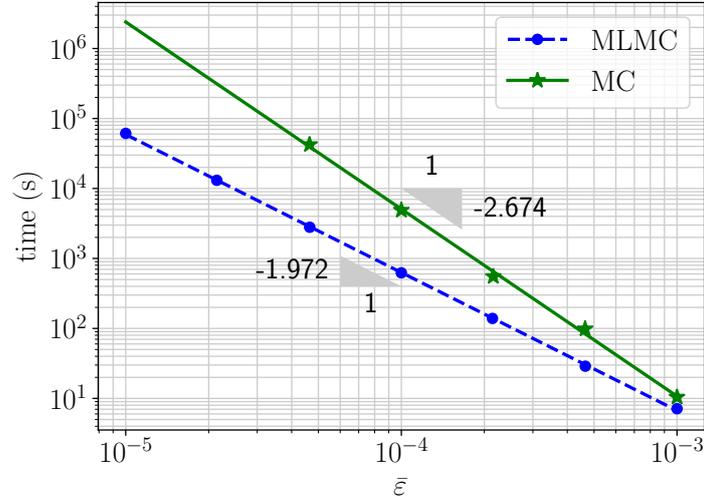


Figure 4.8: Comparison of time taken for the MC and MLMC simulations for the Kirchhoff–Love plate equation for different relative tolerances on 16 cores.

are computed on the cheaper coarse levels relative to the more expensive finer levels compared to Table 4.2, because the variance of the Kirchhoff–Love plate converges faster than the wave equation.

### 4.3.3 Linearised elastodynamics equation

The MLMC method can also be applied to three dimensional structures, where in the past it was often considered infeasible with traditional MC methods. In this third example, the first eigenvalue is computed for slender and curved plate-like structures. In both examples  $E = 70$  GPa,  $\nu = 0.32$  and  $\rho = 2700$  kg/m<sup>3</sup>. Third order polynomial function spaces are used, so  $m = 3$  in equation (4.3). This high order space was used to encourage fast convergence of error and reduce the impact that the point sources had on convergence.

#### Slender plate-like structure

The first eigenvalue for the structure shown in Figure 4.9, with dimensions  $5 \times 10^{-3}$  m  $\times$  0.1 m  $\times$  0.1 m, has been computed for different relative tolerances using the MLMC and MC methods. Twenty point masses are added at random locations in the domain and with maximum magnitude of  $10^{-5}$  of the mass of the unperturbed structure. The number of elements and degrees of freedom for the levels in the MLMC computations are given in Table 4.6 and, as previously, are computed by dividing the elements in each direction of the mesh by two.

The probability distributions, computed using the MLMC method, have converged for  $\bar{\epsilon} = 7 \times 10^{-7}$  and  $8 \times 10^{-7}$  in Figure 4.10. No comparison to the MC method have been made because, as shown later, it is too computationally expensive to use MC methods in three dimensions. The convergence parameters for the MLMC method are shown in Figure 4.11 for  $\bar{\epsilon} = 8 \times 10^{-7}$ . Since

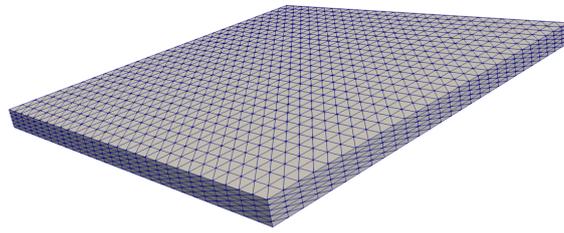
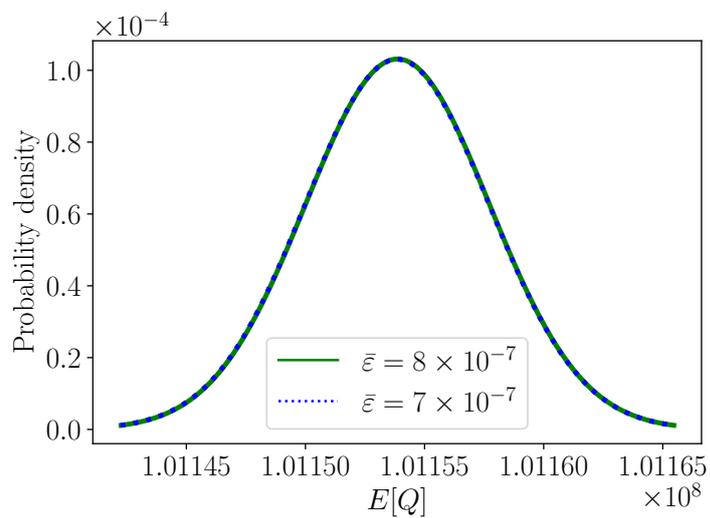


Figure 4.9: Level two slender plate-like structure mesh.

Table 4.6: Number of elements and degrees of freedom in the meshes for each level in the slender three dimensional plate-like MLMC simulations.

	MLMC level index				
	0	1	2	3	4
<b>Number of elements</b>	294	2352	23520	206976	1655808
<b>Number of degrees of freedom</b>	5808	38829	346800	2913222	22827369

Figure 4.10: The MLMC probability distribution functions for the first eigenvalue of a three dimensional slender plate-like structure for  $\bar{\epsilon} = 7 \times 10^{-7}$  and  $\bar{\epsilon} = 8 \times 10^{-7}$ .

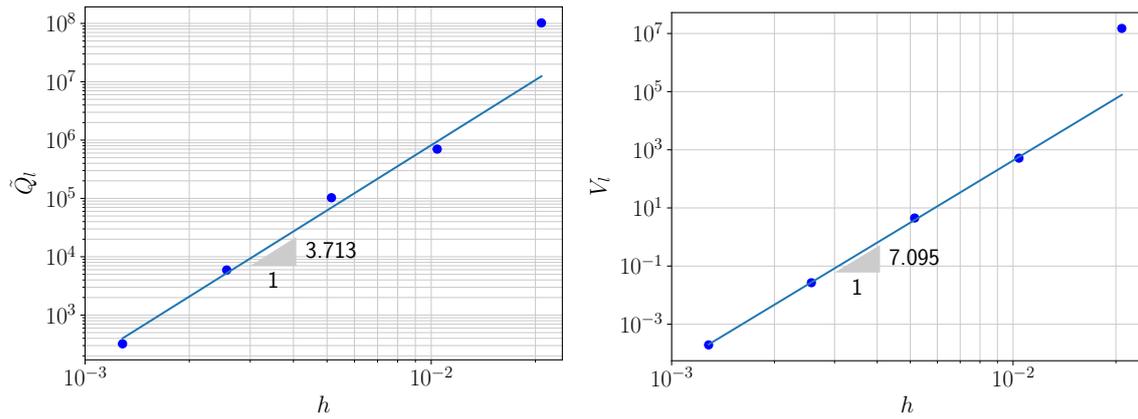
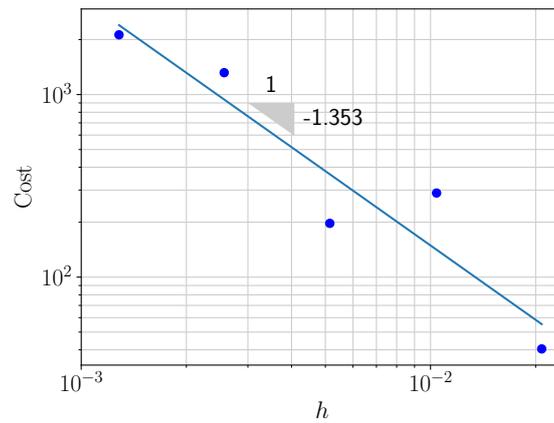
(a)  $\alpha$ , finite element convergence rate.(b)  $\beta$ , variance convergence rate.(c)  $\gamma_d$ , solver cost convergence rate.

Figure 4.11: MLMC convergence rates for the first eigenvalue of the slender three dimensional plate-like structure for  $\bar{\varepsilon} = 8 \times 10^{-8}$ . Fig (a) shows the finite element convergence rate  $\alpha = 3.713$ , Fig (b) shows the convergence rate of the variance  $\beta = 7.095$  and Fig (c) shows the convergence rate of the cost of solving each different level  $\gamma = 1.353/2 = 0.677$ .

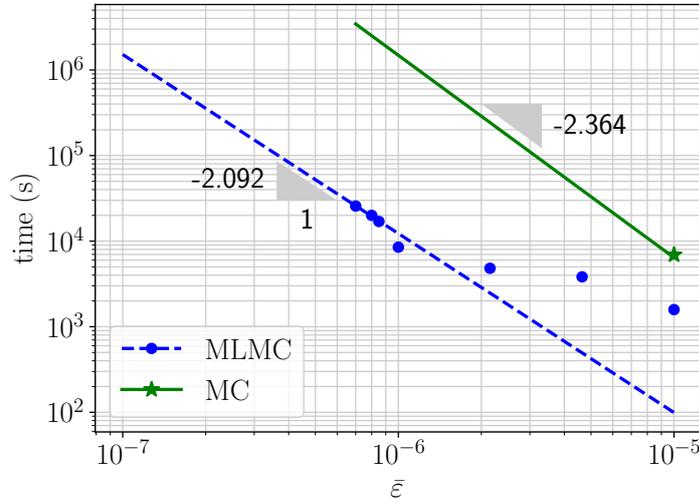


Figure 4.12: Comparison of time taken for the MC and MLMC simulations for the slender three dimensional plate-like structure for different relative tolerances on 24 cores. The green line shows predicted complexity because it was too expensive to compute the actual complexity.

$\beta > \gamma d$ , it is again expected that the MLMC method has a complexity of  $O(\varepsilon^{-2})$  and that the MC method has a complexity of  $O(\varepsilon^{-2.364})$ . The MC complexity seems lower than expected, but this is probably due to the slow convergence of the cost in Figure 4.11, and that the small point masses do not impact the FE error convergence very much since  $\alpha \approx 4$ .

The predicted complexity of the MLMC methods has been checked in Figure 4.12 using 24 cores. As the figure shows, the complexity of the MLMC simulation matches the theoretical prediction. However, due to machine memory restrictions, it was not possible to obtain data for any tighter tolerances. The number of samples at each level of the MLMC simulations is shown in Table 4.7.

It was not possible to check the complexity of the MC method for this slender plate-like structure due to the computational expense and memory requirement of the individual samples. The MC

Table 4.7: Number of samples computed at each level for MLMC simulations of the slender three dimensional plate-like structure for specified tolerances.

$\bar{\varepsilon}$	MLMC level index				
	0	1	2	3	4
$1.0 \times 10^{-5}$	27	4	4	4	-
$4.64 \times 10^{-6}$	117	4	4	4	1
$2.15 \times 10^{-6}$	694	4	4	4	1
$1.0 \times 10^{-6}$	2826	4	4	4	1
$8.5 \times 10^{-7}$	7786	8	4	4	1
$8.0 \times 10^{-7}$	9433	28	4	4	1
$7.0 \times 10^{-7}$	12376	24	4	4	1

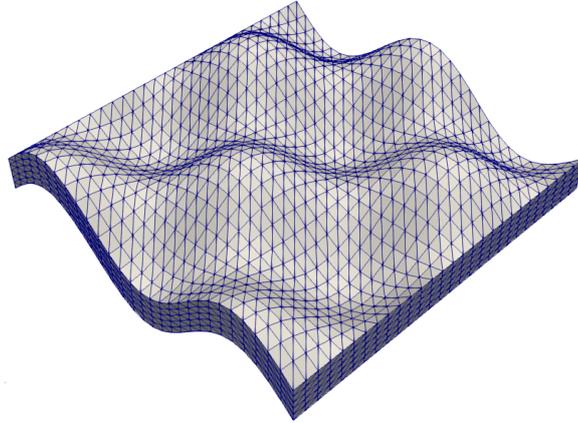


Figure 4.13: Level two curved plate-like structure mesh.

simulation takes approximately two hours for  $\bar{\epsilon} = 10^{-5}$  using the level three mesh with 2913222 degrees of freedom. However, it is predicted to take approximately 17 days to converge to  $\bar{\epsilon} = 10^{-6}$  using a mesh with at least 22827369 degrees of freedom. In reality, the time taken would be much longer because two or more of these large solves would not fit in parallel in the memory on the machine used. Therefore, only an estimate of the MC complexity has been plotted in Figure 4.12.

### Curved plate-like structure

One benefit of the MLMC method is that more complicated structures, such as the curved plate-like structure shown in Figure 4.13, can be considered easily without making model simplifications or assumptions. The dimensions of the plate-like structure are  $0.1\text{m} \times 0.1\text{m} \times 0.01\text{m}$ . The number of elements and degrees of freedom for the MLMC levels are the same as the slender plate and given in Table 4.6. Twenty point masses are added at random locations in the domain, with maximum magnitude of  $5 \times 10^{-3}$  the total mass of the unperturbed structure.

The probability distributions in Figure 4.14 indicate that the MLMC method is converging for  $\bar{\epsilon} = 4.56 \times 10^{-3}$  to  $\bar{\epsilon} = 2.15 \times 10^{-3}$ . Tighter tolerances cannot be obtained due to memory restrictions. Again no comparison is made with the MC method because it is prohibitively expensive to use the MC method in three dimensions. The convergence parameters for the MLMC method are shown in Figure 4.15. Again since  $\beta > \gamma d$ , the theoretical complexity of the MLMC method is  $O(\bar{\epsilon}^{-2})$ . If computed, the MC complexity should be  $O(\bar{\epsilon}^{-2.49})$ . It was noted that the cost of the level three sample looks cheaper than the level two sample. This is most likely due to poor estimation of cost since each solve of the larger meshes uses multiple cores.

The predicted complexities of the MLMC method for the curved plate-like structure has been checked in Figure 4.12 and found to agree with theoretical predictions. Again, since the MC method takes over 1 hour to converge to  $\bar{\epsilon} = 10^{-2}$  on a mesh with 206976 elements, and the predicted complexity is  $O(\bar{\epsilon}^{2.49})$ , it is predicted that the MC method would take 16 days for  $\bar{\epsilon} = 10^{-3}$ . The number of samples at each level of the MLMC simulations are shown in Table 4.8.

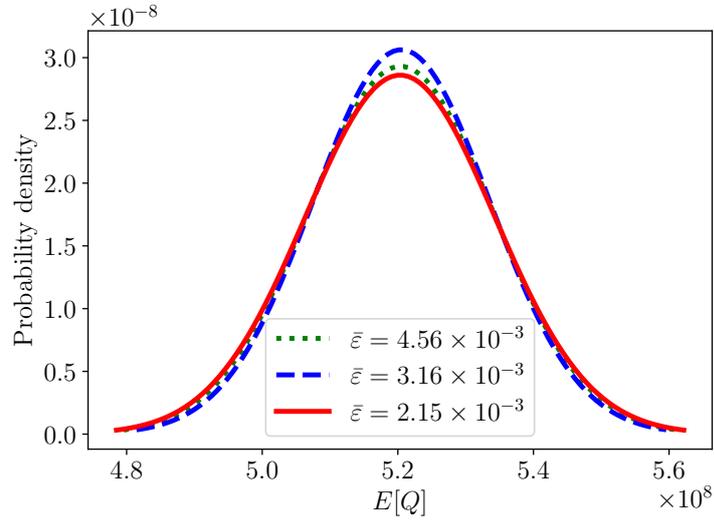


Figure 4.14: The MLMC probability distribution functions for the first eigenvalue of the curved three dimensional plate-like structure for varying tolerances.

Table 4.8: Number of samples computed at each level for MLMC simulations of the curved three dimensional plate-like structure for specified tolerances.

$\bar{\epsilon}$	MLMC level index				
	0	1	2	3	4
$1.0 \times 10^{-2}$	41	14	6	4	-
$6.81 \times 10^{-3}$	74	27	14	4	-
$4.64 \times 10^{-3}$	107	77	35	4	-
$3.16 \times 10^{-3}$	165	141	56	4	1
$2.15 \times 10^{-3}$	426	340	212	7	1

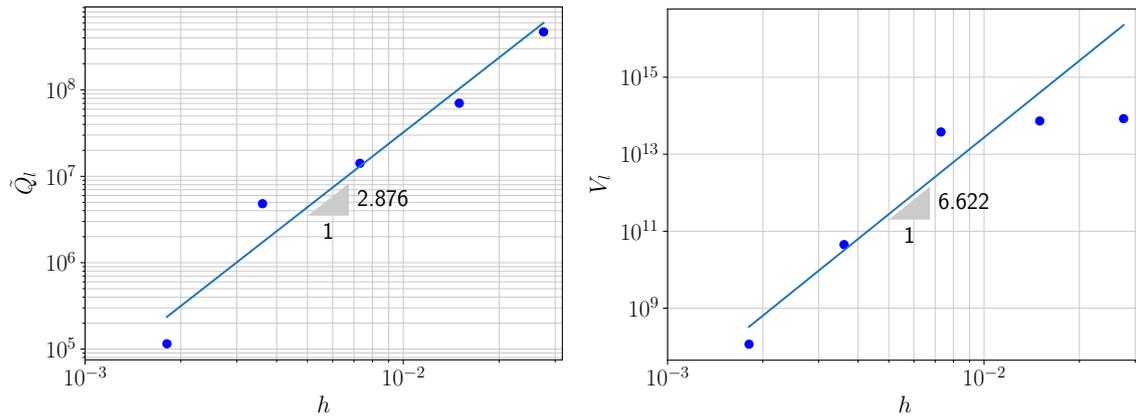
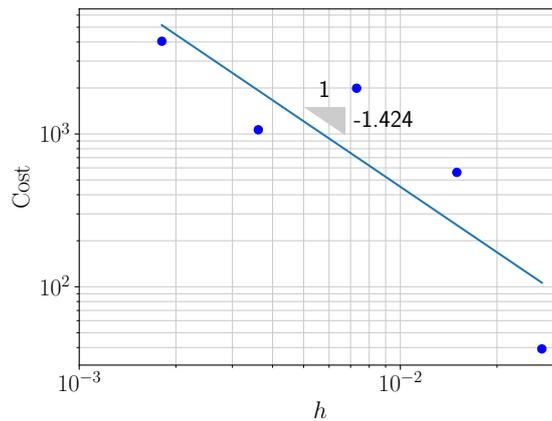
(a)  $\alpha$ , finite element convergence rate.(b)  $\beta$ , variance convergence rate.(c)  $\gamma d$ , solver cost convergence rate.

Figure 4.15: MLMC convergence rates for the first eigenvalue of the curved three dimensional plate-like structure equation  $\bar{\epsilon} = 3.16 \times 10^{-3}$ . Fig (a) shows the finite element convergence rate  $\alpha = 2.876$ , Fig (b) shows the convergence rate of the variance  $\beta = 6.622$  and Fig (c) shows the convergence rate of the cost of solving each different level  $\gamma = 1.424/2 = 0.712$ .

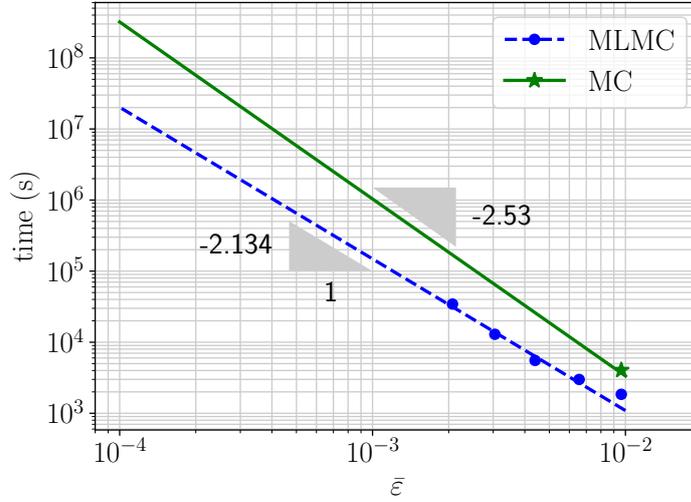


Figure 4.16: Comparison of time taken for the MC and MLMC simulations for the curved three dimensional plate-like structure for different relative tolerances on 24 cores.

#### 4.4 Energy density of Kirchhoff–Love plate

The energy density of the Kirchhoff–Love plate is chosen as the second QoI because this property is of particular interest to the vibrations community, for example Langley and Brown [78] and Cotoni et al. [27]. The time averaged kinetic energy density of the Kirchhoff–Love plate is given by

$$E(\zeta) = \frac{\zeta^2}{4R} \int_D \rho |\sigma(\zeta, \mathbf{x}_0, \mathbf{x})|^2 dx, \quad (4.12)$$

where  $\rho$  is the unperturbed mass density,  $R$  is the area of the plate and  $\sigma(\zeta, \mathbf{x}_0, \mathbf{x})$  is the transfer function between the response point  $\mathbf{x}$  and the drive point  $\mathbf{x}_0$ , at a frequency  $\zeta$ . This can be written as the following sum over the modes for a proportionally damped Kirchhoff–Love plate,

$$\sigma(\zeta, \mathbf{x}_0, \mathbf{x}) = \sum_{n=1}^N \frac{\phi_n(\mathbf{x}) \phi_n(\mathbf{x}_0)}{K(\zeta_n^2 - \zeta^2 + i\eta\zeta\zeta_n)}, \quad (4.13)$$

where  $\phi_n$  is the  $n$ th eigenfunction,  $\zeta_n$  is the  $n$ th natural frequency and  $\eta$  is the loss factor. See Appendix D for a derivation.

Since the eigenfunctions are orthogonal with respects to density

$$\int_D \rho \phi_n^2 = \frac{K}{2a}, \quad (4.14)$$

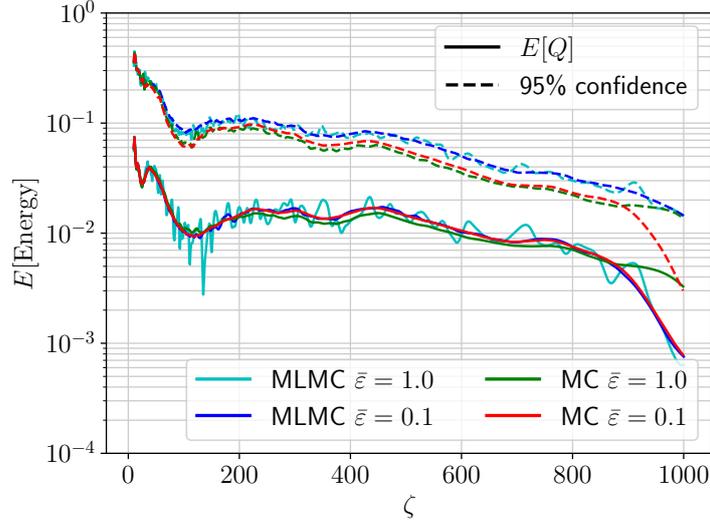


Figure 4.17: Time averaged energy density frequency response for different forcing frequencies.

the time averaged kinetic energy density of the plate is

$$E(\zeta) = \sum_{n=1}^N \frac{\zeta^2 \psi_n}{(\zeta_n^2 - \zeta^2)^2 + (\eta \zeta \zeta_n)^2}, \quad (4.15)$$

where

$$\psi_n = \frac{\phi_n^2(\mathbf{x}_0)}{4R(2aK)}. \quad (4.16)$$

#### 4.4.1 Comparison of the frequency response

The time averaged energy density is computed using equation (4.15) with the first 20 eigenvalues and eigenfunctions of the plate formulation, given in equation (4.7). The dimensions of the unperturbed plate are 1.0 m  $\times$  0.8 m with  $E = 70$  GPa,  $\nu = 0.32$ ,  $\rho = 2700$  kg/m<sup>3</sup> and  $2a = 0.003$  m. Seven point masses, totalling 10% of the total mass of the unperturbed plate, are added at random points in the domain and the loss factor is 0.05. The plate is forced at  $x = 0.5$  and  $y = 0.35$ . The coarsest mesh used for the MLMC simulation has 50 elements and the number of elements increased by four for each level. The mesh for the MC simulation for  $\bar{\varepsilon} = 1.0$  has 50 elements and  $\bar{\varepsilon} = 0.1$  has 450 elements.

The expectation of the energy from each forcing frequency is shown in Figure 4.17 for  $\bar{\varepsilon} = 1$  and  $\bar{\varepsilon} = 0.1$ . The energy is normalised by  $1 \times 10^{-2}$ . The solid lines correspond to the expectation of the energy for the given forcing frequency and the dashed lines correspond to the upper bound of the 95% confidence level. There is good agreement between the expectation of the energy for the MLMC and MC methods at  $\bar{\varepsilon} = 0.1$  and a reasonable agreement in the 95% confidence level for the given tolerances that the solution has converged to.

Table 4.9: Time taken to compute the MLMC and MC simulations of the time averaged energy density for different tolerances.

$\bar{\epsilon}$	MLMC time (s)	MC time (s)
1.0	52	95
0.1	4782	15206
0.01	478200 <sup>1</sup>	2704000 <sup>1</sup>

<sup>1</sup> This value is extrapolated.

Table 4.10: Number of samples taken to compute the time averaged energy density for different tolerances.

$\bar{\epsilon}$	MLMC level index				MC
	0	1	2	3	
1.0	4116	133	10	10	8113
0.1	411468	13308	912	477	929186

#### 4.4.2 Comparison of time cost

Table 4.9 shows that the expectation of the energies is found approximately twice as fast with the MLMC method, for  $\bar{\epsilon} = 0.1$ , than the MC method. The complexity of the MLMC method is  $O(\epsilon^{-2})$  as expected because a 100 fold difference in time taken occurs from an order of magnitude decrease in tolerance. The complexity of the MC method is predicted to be between  $O(\bar{\epsilon}^{-2.25})$  and  $O(\bar{\epsilon}^{-2.5})$  because  $\beta = 5 > \gamma d = 2$  and  $\alpha = [0.5, 1]$ . The value of  $\alpha$  varies since the energy from some forcing frequencies converges faster than others as the mesh is refined. These complexities have been used to predict the times taken for  $\bar{\epsilon} = 0.001$ . The MLMC method would take 5.5 days and MC would take over 31 days on 16 cores, so has not been computed. The number of samples for both methods is shown in Table 4.10 for the two relative tolerances.

## 4.5 Conclusions

In this chapter, the MLMC method is shown to be a viable alternative to the traditional MC method for quantifying uncertainty in structural vibrations problems subject to random density perturbations. The expectations and variances, that are computed using the MC and MLMC methods, are comparable for the four examples in Sections 4.3 and 4.4. However, the computational cost of the MLMC method in each case is much lower than the MC method. For example, for certain scenarios a two orders of magnitude speed up is obtained using the MLMC method for the wave equation, rather than the traditional MC method. Quantifying uncertainty is also now shown to be tractable in three dimensional plate-like structures.

The MLMC method is computationally cheaper than the MC method because the MLMC sampling error can be reduced by sampling a coarse representation of the models multiple times, which is cheaper to solve than the fine representation. Therefore, only a few samples of the expensive fine

models are required to satisfy the numerical error tolerance. In contrast, in the MC method, all the samples have to be computed using the fine model.

The expected theoretical complexities from Sections 3.1 and 3.3.4 are shown to hold for eigenvalue problems. The optimal MLMC method complexity of  $O(\varepsilon^{-2})$  is obtained in the four examples presented, since  $\beta > \gamma d$ . Where possible, the complexity of the MC method has also been shown to equal  $O(\varepsilon^{-2-\gamma d/\alpha})$ , as expected.

It is also noted that the regularity of the FE formulation is disturbed when using point masses. This results in sub-optimal convergence rates, which means the convergence of the QoI is slower using point masses than predicted with continuous formulations.



## Chapter 5

# Modelling of buoyancy driven flows

The second application considered in this thesis is buoyancy driven flows through porous media. These flows result from variations in buoyancy due to changes in parameters such as density or temperature and can be modelled by the Darcy equations coupled to a scalar transport equation. They are important in many real-world applications, such as CO<sub>2</sub> sequestration and pollution management.

In this chapter, an overview of modelling methodologies for buoyancy driven flows in porous media is given. In Section 5.1, a focused review of difficulties in modelling this type of flow is presented. Then in Section 5.2, the general equations for the Darcy equations and a scalar transport equation are given and in Section 5.3, different FE formulations are proposed. *A priori* error estimates are considered in Section 5.4 to explain why spurious flows form when function spaces are incompatible and numerical examples are presented in Section 5.5 and Section 5.6, which support this theory. Then conclusions are drawn in Section 5.7.

### 5.1 Spurious buoyancy driven flows

Segol et al. [107] used a primal FE formulation to solve the Darcy equations. In this formulation, the pressure field is computed directly, but the velocity field is computed diagnostically. They found that the resulting flows from the primal formulation may be inaccurate due to problems with representing the solution to the PDE accurately.

Voss and Souza [128] found that spurious velocities may result from using a primal formulation of the Darcy equations with linear piecewise continuous quadrilateral elements and a linear forcing term. They proposed using a significant modification to the primal formulation to ensure a consistent velocity approximation is obtained for quadrangle elements. This method has since been generalised for multiple elements [44, 74] and mesh independent approximations [3].

An alternative method to the consistent velocity approximations, for ensuring no spurious velocities are added to the primal formulation, is to raise the degree of the pressure function space [61, 128]. This may enable the gradient of the pressure to be represented accurately, but has not always been

possible to due to computer memory restrictions. Even today, some software packages may not offer higher order function spaces [35].

Solutions to density driven flows in porous media can also be obtained using mixed finite element methods, where both the pressure and velocity are computed directly. For example, an  $H(\text{div})$  function space could be used for the unknown velocity field and a discontinuous function space could be used for the unknown pressure field, where an  $H(\text{div})$  space has components in  $L_2$  such that  $\text{div}(H) \in L_2$ . As long as the inf-sup condition [15] is met, no spurious flows should result, even if the pressure and velocity function spaces cannot match the solution accurately [44, 74, 128].

Although spurious numerical artefacts have been shown in the literature to result from incompatibilities in representing the solution on function spaces, little clear explanation has been provided to explain why they form. In addition, the long term impact of incorrectly modelling density driven flows through porous media has not been investigated. Therefore, the following work aims to answer these questions.

## 5.2 General formulation

Similar to the notation in Section 2.1, the domain for these buoyancy driven flows is  $D \subset \mathbb{R}^d$ , where  $1 \leq d \leq 3$ , and  $D$  is triangulated into non-overlapping cells,  $K$ . The boundary is denoted by  $\Gamma = \partial D$ , and  $P$  is a point on the boundary. A time-dependent model is formulated on the time interval  $t \in I = [0, t_N]$ .

The Darcy equations and corresponding boundary conditions read: given the buoyancy driven forcing  $\mathbf{f}(\mathbf{x}, c)$ , which is dependent on the concentration of the fluid  $c$ , find the pressure field  $p(\mathbf{x})$  and velocity field  $\mathbf{u}(\mathbf{x})$  such that

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } D \times I, \quad (5.1a)$$

$$\mathbf{u} = \frac{\kappa}{\mu} (-\nabla p + \mathbf{f}) \quad \text{on } D \times I, \quad (5.1b)$$

$$p = p_D \quad \text{on } P \times I, \quad (5.1c)$$

$$\mathbf{u} \cdot \mathbf{n} = u_D \quad \text{on } \Gamma \times I, \quad (5.1d)$$

where the coefficients  $\kappa > 0$  and  $\mu > 0$  are the permeability and viscosity respectively,  $p_D$  is a prescribed pressure,  $u_D$  is a prescribed velocity and  $\mathbf{n}$  is the outward unit normal vector to the boundary.

The Darcy equations are coupled to a scalar transport equation through the forcing term,  $\mathbf{f}$ . The scalar transport equation and corresponding boundary conditions read: given  $\mathbf{u}(x, t)$ , find  $c(x, t)$  such

that

$$\frac{\partial c}{\partial t} + \nabla c \cdot \mathbf{u} - \nabla \cdot \mathcal{D} \nabla c = 0 \quad \text{on } D \times I, \quad (5.2a)$$

$$-\mathcal{D} \nabla c \cdot \mathbf{n} = c_{D,\Gamma} \quad \text{on } \Gamma \times I, \quad (5.2b)$$

$$\mathbf{c} \mathbf{u} \cdot \mathbf{n} = c_{U,\Gamma} \quad \text{on } \Gamma \times I, \quad (5.2c)$$

$$-\mathcal{D} \nabla c \cdot \mathbf{n} = c_{D,P} \quad \text{on } P \times I, \quad (5.2d)$$

$$\mathbf{c} \mathbf{u} \cdot \mathbf{n} = c_{U,P} \quad \text{on } P \times I, \quad (5.2e)$$

$$c(x, 0) = c_0 \quad \text{on } D, \quad (5.2f)$$

where  $\mathcal{D} \geq 0$  is the pore-scale diffusivity,  $c_0 \geq 0$  is the initial mass fraction of the solute, and  $c_{D,\Gamma}$ ,  $c_{D,P}$ ,  $c_{U,\Gamma}$  and  $c_{U,P}$  describe the transport of the solute on the boundary.

### 5.3 Finite element formulation for buoyancy driven flows

The Darcy and scalar transport equations can be solved with various FE formulations. However, the sharp fronts and jumps in concentration, which are inherent in problems with low diffusion, can be difficult to represent numerically. Therefore, a careful selection of the formulation is important. In this section, different FE formulations are discussed and presented.

#### 5.3.1 Darcy equations

It is possible to use either a primal or a mixed formulation to solve the Darcy equations. In the primal formulation, equation (5.1b) is substituted into equation (5.1a) and the formulation reads: given  $\mathbf{f}$ , find  $p_h \in Q_h$  such that

$$\int_D \nabla q_h \cdot \frac{\kappa}{\mu} \nabla p_h \, dx - \int_D \nabla q_h \cdot \frac{\kappa}{\mu} \mathbf{f} \, dx + \int_\Gamma u_D \cdot \mathbf{n} \, d\Gamma = 0 \quad \forall q_h \in Q_h, \quad (5.3)$$

where

$$Q_h := \{p_h \in H^1(D), p_h|_K \in P_m(K) \, \forall K \in \mathcal{D}\}, \quad (5.4)$$

and  $P(m)$  denotes a space of Lagrange polynomials on  $K$  of degree  $m > 0$ .

In contrast, the mixed formulation of the Darcy equations in (5.1a)-(5.1b) reads: given  $\mathbf{f}$ , find  $p_h \in Q_h$  and  $\mathbf{u}_h \in \mathbf{V}_h$  such that

$$\int_D q_h \nabla \cdot \mathbf{u}_h \, dx = 0 \quad \forall q_h \in Q_h, \quad (5.5a)$$

$$\int_D \mathbf{u}_h \cdot \mathbf{v}_h \, dx - \int_D \frac{\kappa}{\mu} p_h \nabla \cdot \mathbf{v}_h \, dx = \int_D \frac{\kappa}{\mu} \mathbf{f} \cdot \mathbf{v}_h \, dx - \int_{\Gamma_A} \frac{\kappa}{\mu} p_D \mathbf{v}_h \cdot \mathbf{n} \, d\Gamma \quad \forall \mathbf{v}_h \in \mathbf{V}_h. \quad (5.5b)$$

where

$$\mathcal{Q}_h := \left\{ p_h \in L^2(D), p_{h|K} \in P_m(K) \forall K \in \mathcal{D} \right\}, \quad (5.6)$$

$$\mathbf{V}_h := \left\{ \mathbf{u}_h \in H(\text{div}, D), \mathbf{u}_{h|K} \in \text{BDM}_{m+1}(K) \forall K \in \mathcal{D} \right\}, \quad (5.7)$$

form a mixed function space,  $\mathcal{Q}_h \times \mathbf{V}_h$ , where  $m > 0$  is the degree of the Lagrange polynomials and  $m + 1 > 0$  is the degree of the Brezzi-Douglas-Marini (BDM) elements [16].

BDM elements are used here because, as with the more common Raviart-Thomas elements [98], they form a conforming space of  $H(\text{div})$  that describes the continuous normal components of the solution across the cell facets and have similar error estimates. However, BDM elements require fewer degrees of freedom than the same degree Raviart Thomas elements [16].

### 5.3.2 Scalar transport equation

When diffusion is not the dominant transport mechanism in the flow, sharp fronts can propagate through the domain, which are hard to model numerically. One technique to overcome this is the Streamline-Upwind Petrov-Galerkin (SUPG) method [36]. This adds residual based artificial diffusion, which acts only in the flow direction, by modifying the standard Galerkin weight functions with a streamline upwind perturbation in the flow direction [18]. However, the smoothing is not always sufficient to stabilise the sharp front and numerical instabilities can still result.

Instead, a fully up-winded discontinuous Galerkin finite element method (DGFEM) can be implemented to ensure cell-wise mass conservation. Assuming a backward Euler time discretisation scheme, the formulation of the scalar transport equation (5.2a) reads: given  $\mathbf{u}_h \in \mathbf{V}_h$ , find  $c_h^{n+1} \in W_h$  such that

$$\begin{aligned} & \int_D \tau_h \frac{c_h^{n+1} - c_h^n}{\Delta t} dx - \int_D c_h^{n+1} \nabla \tau_h \cdot \mathbf{u}_h dx + \int_D \mathcal{D} \nabla \tau_h \cdot \nabla c_h^{n+1} dx \\ & \quad + \int_{\Gamma_C} \tau_h \left( \min(\mathbf{u}_h \cdot \mathbf{n}, 0) c_D + \max(\mathbf{u}_h \cdot \mathbf{n}, 0) c_h^{n+1} \right) d\Gamma \\ & \quad + \int_S \llbracket \tau_h \rrbracket \left( \max(\mathbf{u}_h \cdot \mathbf{n}, 0)^+ c_h^{n+1,+} - \max(\mathbf{u}_h \cdot \mathbf{n}, 0)^- c_h^{n+1,-} \right) dS \\ & + \int_S \mathcal{D} \frac{\gamma}{h^+} \llbracket \boldsymbol{\tau}_n \rrbracket \cdot \llbracket \mathbf{c}_n \rrbracket dS - \int_S \mathcal{D} \{ \nabla \tau_h \} \cdot \llbracket \mathbf{c}_n \rrbracket dS - \int_S \mathcal{D} \{ \nabla c_h^{n+1} \} \cdot \llbracket \boldsymbol{\tau}_n \rrbracket dS = 0 \quad \forall \tau_h \in W_h, \end{aligned} \quad (5.8)$$

where  $\gamma \geq 0$  is a constant and

$$\llbracket \boldsymbol{\tau}_h \rrbracket = \boldsymbol{\tau}_h^+ - \boldsymbol{\tau}_h^-, \quad (5.9a)$$

$$\llbracket \boldsymbol{\tau}_n \rrbracket = \boldsymbol{\tau}_h^+ \mathbf{n}^+ - \boldsymbol{\tau}_h^- \mathbf{n}^-, \quad (5.9b)$$

$$\llbracket \mathbf{c}_n \rrbracket = c_h^{n+1,+} \mathbf{n}^+ - c_h^{n+1,-} \mathbf{n}^-, \quad (5.9c)$$

$$\{\nabla \boldsymbol{\tau}_h\} = \frac{1}{2} (\nabla \boldsymbol{\tau}_h^+ + \nabla \boldsymbol{\tau}_h^-), \quad (5.9d)$$

$$\{\nabla c_h\} = \frac{1}{2} (\nabla c_h^{n+1,+} + \nabla c_h^{n+1,-}). \quad (5.9e)$$

The superscript ‘+’ and ‘-’ refer to opposite sides of the interior cell edge, denoted by  $S$ , and ‘ $\llbracket \cdot \rrbracket$ ’ refers to a jump and ‘ $\{\cdot\}$ ’ to an average across the interior cell edges. The function space  $W_h$  is defined as

$$W_h := \{c_h^{n+1} \in L^2(D), c_{h|K}^{n+1} \in P_k(K) \forall K \in D\}, \quad (5.10)$$

where  $P_k(K)$  denotes a space of Lagrange polynomials on  $K$  of degree  $k$  [17, 76].

## 5.4 *A priori* error estimates for Darcy equation formulations

The existence of the spurious buoyancy driven flows, which are identified in Section 5.1, can be explained by considering *a priori* error estimates. The standard *a priori* estimate for the primal formulation of the Darcy flow, in equation (5.3), is

$$\|p - p_h\|_1 \leq C \inf_{q \in Q_h} \|p - q\|_1, \quad (5.11)$$

where  $Q_h$  is the function space for the pressure in equation (5.6) and  $C$  is a constant that is independent of  $h$ ,  $p_h$  and  $p$ . This estimate means that even if the gradient space of  $p_h$  contains  $\mathbf{f}$ ,  $\nabla p_h$  will not necessarily balance  $\mathbf{f}$ . This balance between  $\nabla p_h$  and  $\mathbf{f}$  can only occur if the exact solution is contained in the finite element space. Otherwise, numerical errors can be added to the solution, which result in spurious velocities. Segol et al. [107] proposed that the buoyancy force,  $\mathbf{f}$  must at least come from the gradient space of  $p_h$  to capture  $\mathbf{u} = \mathbf{0}$  point-wise. From equation (5.11), this is an insufficient condition.

For the stable mixed formulation in equation (5.5), the pressure and velocity are bound independently; the pressure is bound by equation (5.11) and the velocity is bound by

$$\|\nabla \cdot (\mathbf{u} - \mathbf{u}_h)\| \leq C \inf_{\mathbf{v} \in \mathbf{V}_h} \|\nabla \cdot (\mathbf{u} - \mathbf{v})\|, \quad (5.12)$$

where  $\mathbf{V}_h$  is the function space for the velocity field in equation (5.7) and  $C$  is another constant. This implies that if a problem is in hydrostatic equilibrium,  $\mathbf{u} = \mathbf{0}$ , the finite element approximation of the flux  $\mathbf{u}_h$  will also be zero. This means that hydrostatic equilibrium can be satisfied point-wise for any functional form of the buoyancy term,  $\mathbf{f}$ , irrespective of the degree of the function spaces. Therefore,

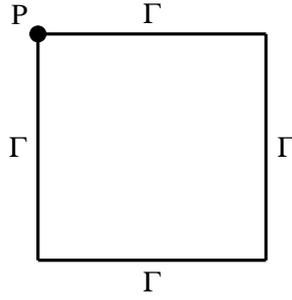


Figure 5.1: Boundaries of domain for compatible function space investigation.

no spurious velocities should be added to the solution from the mixed formulation, as in the primal formulation, due to errors in the pressure.

## 5.5 Numerical examples of compatible function spaces for Darcy flow

The coupled Darcy equations (5.1) and the scalar transport equation (5.2) are solved for two different forcing terms to illustrate the *a priori* error estimate explanation. The first forcing term is polynomial and the second sinusoidal.

Only advection terms are considered in the scalar transport equation to avoid artificial smoothing by diffusion; therefore  $\mathcal{D} = 0$  in equation (5.2). The problems are solved on the unit square domain in Figure 5.1, and discretised using a structured mesh. From equations (5.1) and (5.2), the constants  $p_D = 0$ ,  $u_N = 0$ ,  $c_{D,\Gamma} = 0$ ,  $c_{D,P} = 0$ ,  $c_{U,\Gamma} = 0$ ,  $c_{U,P} = 0$  and  $\kappa/\mu = 1$  are chosen to ensure hydrostatic equilibrium, and the time-step  $\Delta t = 0.001$ . The initial concentration field,

$$c_0 = \begin{cases} 1 & \text{if } 0 \leq y < 0.4, \\ 1 - 5 \left( y - \frac{2}{5} \right) & \text{if } 0.4 \leq y < 0.6, \\ 0 & \text{if } 0.6 \leq y \leq 1, \end{cases} \quad (5.13)$$

is chosen to represent a stable scenario, with a less dense layer of fluid on top of a more dense layer and a linear mixing region in between. This flow is in hydrostatic equilibrium so the velocity should be negligible everywhere and the concentration profile should not change.

The first forcing term,

$$\mathbf{f}_p = \mathbf{g}(1 + \beta c), \quad (5.14)$$

is a linear polynomial, where  $\mathbf{g} = (0.0, -1.0)$  is a gravity-like term and  $\beta = 0.01$  is an expansion coefficient. The velocity profiles for the primal and mixed formulations are depicted in Figure 5.2 for time  $t = 400\Delta t$ .

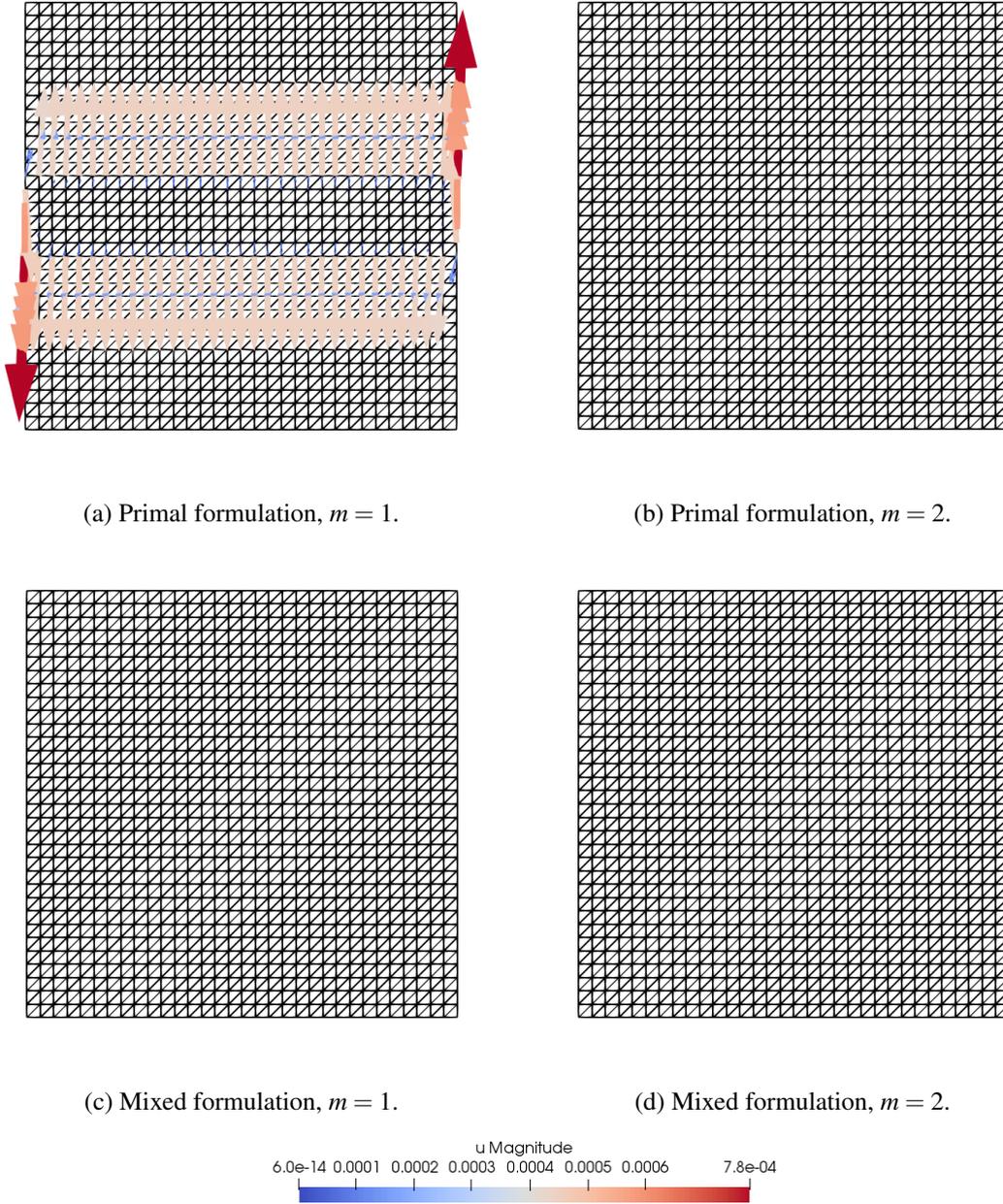


Figure 5.2: Velocity profile of coupled Darcy and scalar transport equation flow for different finite element formulations with polynomial forcing  $\mathbf{f}_p = \mathbf{g}(1 + \beta c)$  at time  $t = 400\Delta t$ . The flow velocity profile are depicted using arrows scaled by the magnitude of the velocity. Hydrostatic equilibrium is maintained when no arrows are visible.

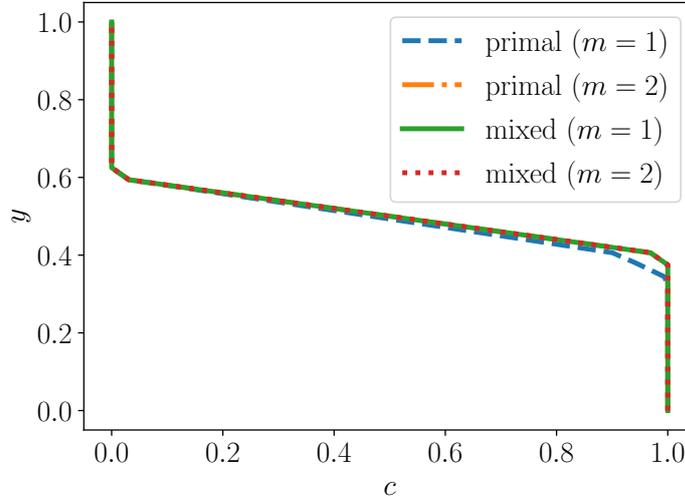


Figure 5.3: Concentration profile along  $y$ -axis at  $x = 0.5$  for  $\mathbf{f} = \mathbf{g}(1 + \beta c)$ . Note the dot-dashed, solid, and dotted lines are the same.

It is evident from Figure 5.2a that hydrostatic equilibrium is not maintained for the primal formulation when the degree of the pressure function space,  $m = 1$ , see equation (5.4). This is because, as expected from the error estimates in equation (5.11),  $\nabla p_h$  does not balance  $\mathbf{f}_p$  point-wise since the forcing term is also linear. In contrast, no flow occurs when the degree of the function space in equation (5.4) is increased to two, see Figure 5.2b. This is because  $\mathbf{f}_p$  can be matched point-wise by  $\nabla p_h$ , when  $p_h$  is represented by quadratic elements. No spurious flows are found in Figures 5.2c and 5.2d when either linear or quadratic degree mixed function spaces are used for the pressure, which corresponds to  $m = 1, 2$  in equation (5.6). This is because the error in the mixed method velocity approximate is bound by its own error estimate in equation (5.12), which holds irrespective of  $\mathbf{f}_p$ .

The impact of the spurious flows for the primal Darcy formulation is also shown in the concentration profile. If the flow remains in hydrostatic equilibrium, the concentration should remain the same as the initial condition because there is no diffusion. However, as shown in Figure 5.3, the concentration profile for the primal first order formulation at  $x = 0.5$  is not the same as the initial condition. This implies that spurious velocities have been generated. Again, the second order primal and both mixed formulations do not change showing that hydrostatic equilibrium is maintained.

The second forcing term

$$\mathbf{f}_s = \mathbf{g}(1 + \cos(c)) \quad (5.15)$$

is sinusoidal and the velocity fields for the different FE formulations are depicted in Figure 5.4. For this sinusoidal forcing, hydrostatic equilibrium does not remain for either of the primal formulations because the gradient of the pressure space does not contain sinusoidal terms. However, hydrostatic equilibrium is maintained by both degree mixed function spaces, for this example, because the velocity error estimate in equation (5.12) is independent of the forcing.

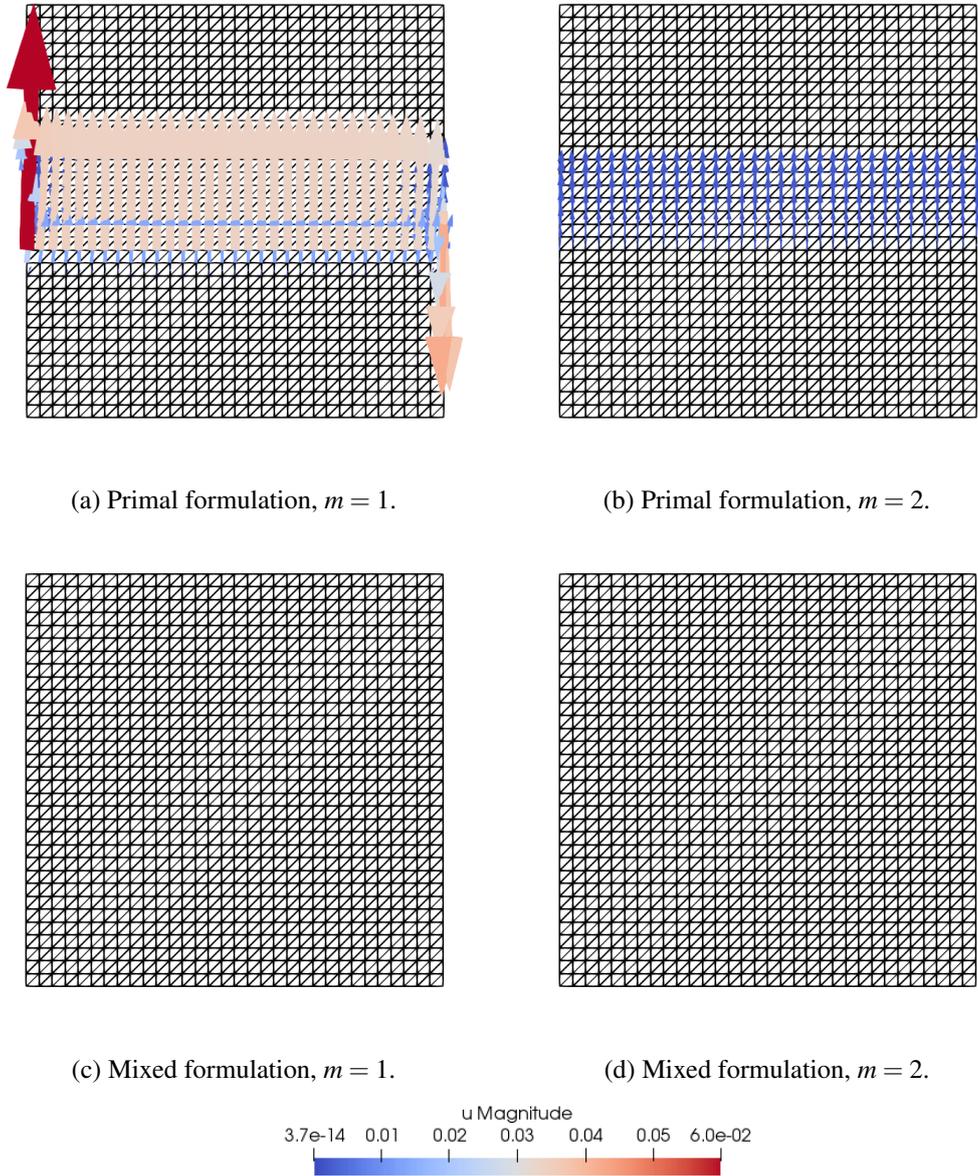


Figure 5.4: Velocity profile of coupled Darcy and scalar transport equation flow for different finite element formulations with sinusoidal forcing at  $t = 400\Delta t$  with  $\mathbf{f}_s = \mathbf{g}(1 + \cos(c))$ . The vectors in Fig (b) are scaled five times bigger than the arrows in Fig (a) so they can be seen. In Fig (c) and (d) there is no flow since the problem is in hydrostatic equilibrium.

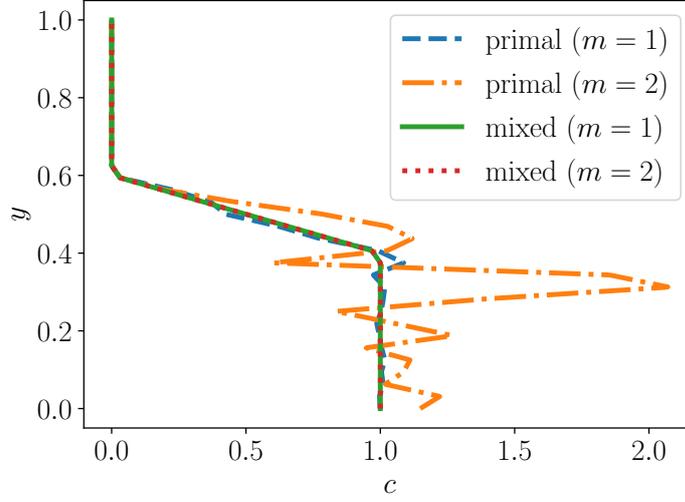


Figure 5.5: Darcy flow concentration profile along  $y$ -axis at  $x = 0.5$  for  $\mathbf{f}_s = \mathbf{g}(1 + \cos(c))$ . Note the dotted and solid lines are the same.

The concentration profiles in Figure 5.5 again show that only the primal formulations contain spurious velocities at time  $t = 400\Delta t$ . The profiles of the two primal formulations along  $x = 0.5$  no longer agree with the initial concentration profile in equation (5.13). However, hydrostatic equilibrium can be maintained by the mixed formulation, despite the non-polynomial forcing term.

## 5.6 Numerical examples of compatible function spaces for Stokes flow

Darcy flows through porous media are not the only type of flows that require compatible function spaces to ensure no spurious velocities are added to the solution. In this section, the impacts of different FE formulation are investigated for the Stokes equations coupled to an advection dominated scalar transport equation. This model is used to represent flows with small inertial forces compared to viscous forces.

For the domain in Figure 5.1, the Stokes equations and corresponding boundary conditions read: given the density driven forcing  $\mathbf{f}(\mathbf{x}, c)$ , which depends on the concentration of the fluid, find the velocity field  $\mathbf{u}(\mathbf{x})$  and pressure field  $p(x)$  such that

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } D \times I \quad (5.16a)$$

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{on } D \times I \quad (5.16b)$$

$$\boldsymbol{\sigma} = p\mathbf{I} - 2\nu\nabla^s \mathbf{u} \quad \text{on } D \times I \quad (5.16c)$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma \times I \quad (5.16d)$$

$$p = 0 \quad \text{on } P \times I \quad (5.16e)$$

where  $\boldsymbol{\sigma}$  is the momentum flux of the fluid,  $\mathbf{I}$  is the identity matrix,  $\nu$  is the viscosity of the fluid and  $\nabla^s \mathbf{u} = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$  is the symmetric gradient.

### 5.6.1 Finite element formulations for Stokes flow

Two FE formulations are identified to solve the Stokes equations in (5.16). The first uses Taylor-Hood (TH) elements [113] and the second uses a hybrid continuous and discontinuous finite element method [77].

Stokes flow can be formulated using a mixed finite element method, where the pressure and velocity are computed directly. If the momentum flux is defined as

$$\boldsymbol{\sigma}_{d|h} = p_h \mathbf{I} - 2\nu \nabla^s \mathbf{u}_h, \quad (5.17)$$

the finite element formulation of equation (5.16) reads: given  $\mathbf{f}(c)$  find  $p_h \in Q_h$  and  $\mathbf{u}_h \in \mathbf{V}_h$  such that

$$\int_D \nabla q_h \cdot \mathbf{u}_h \, dx = 0 \quad \forall q_h \in Q_h, \quad (5.18a)$$

$$-\int_D \nabla \mathbf{v}_h \cdot \boldsymbol{\sigma}_{d|h} \, dx = \int_D \mathbf{v}_h \cdot \mathbf{f} \, dx \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \quad (5.18b)$$

where

$$Q_h := \left\{ p_h \in H^1(D), p_{h|K} \in P_n(K) \, \forall K \in D \right\}, \quad (5.19)$$

$$\mathbf{V}_h := \left\{ \mathbf{u}_h \in [H^1(D)]^d, \mathbf{u}_{h|K} \in [P_{n+1}(K)]^d \, \forall K \in D \right\}, \quad (5.20)$$

and  $n > 0$  denotes the degree of the Lagrange functions spaces of  $K$ . The mixed finite element space  $Q_h \times \mathbf{V}_h$  is known as the Taylor hood element [113].

Alternatively, the hybrid method finite element formulation of the Stokes equations in (5.16) can be used, which was developed by Labeur and Wells [77]. This method assumes that adjacent elements in  $D$  share a common facet,  $\mathcal{F}$ , and  $F = \bigcup \mathcal{F}$  is the union of all facets, including the exterior boundary facets. Following on from  $\boldsymbol{\sigma}_{d|h}$ , defined in equation (5.17), the momentum flux multiplied by the cell normal vector is

$$\boldsymbol{\sigma}_{d,n|h} = \bar{p}_h \mathbf{n} - 2\nu \nabla^s \mathbf{u}_h \mathbf{n} - 2 \frac{\nu \alpha}{h} ((\bar{\mathbf{u}}_h - \mathbf{u}_h) \otimes \mathbf{n}) \mathbf{n}, \quad (5.21)$$

where  $\alpha > 0$  is a penalty parameter and the bar denotes the variable is defined on cell facets only, and the interface mass flux is described by

$$\hat{u}_n = \mathbf{u}_h \cdot \mathbf{n}. \quad (5.22)$$

Therefore, the formulation reads: given the forcing  $\mathbf{f}(c)$  find  $p_h \in Q_h$ ,  $\bar{p}_h \in \bar{Q}_h$ ,  $\mathbf{u}_h \in \mathbf{V}_h$  and  $\bar{\mathbf{u}}_h \in \bar{\mathbf{V}}_h$  such that

$$\int_D \nabla q_h \cdot \mathbf{u}_h \, dx - \int_S ((q_h \hat{u}_n)^+ + (q_h \hat{u}_n)^-) \, dS - \int_\Gamma q_h \hat{u}_n \, d\Gamma = 0 \quad \forall q_h \in Q_h, \quad (5.23a)$$

$$\int_S \bar{q}_h \mathbf{u}_h \cdot \mathbf{n}^+ + \bar{q}_h \mathbf{u}_h \cdot \mathbf{n}^- \, dS + \int_\Gamma \bar{q}_h \mathbf{u}_h \cdot \mathbf{n} \, d\Gamma - \int_\Gamma \bar{q}_h \bar{\mathbf{u}}_h \cdot \mathbf{n} \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_h, \quad (5.23b)$$

$$\begin{aligned} - \int_D \nabla \mathbf{v}_h \cdot \boldsymbol{\sigma}_{d|h} \, dx + \int_S (\mathbf{v}_h \cdot \boldsymbol{\sigma}_{d,n|h})^+ + (\mathbf{v}_h \cdot \boldsymbol{\sigma}_{d,n|h})^- \, dS + \int_\Gamma \mathbf{v}_h \cdot \boldsymbol{\sigma}_{d,n|h} \, d\Gamma \\ + \int_S (2\nu(\bar{\mathbf{u}}_h - \mathbf{u}_h) \cdot \nabla^s \mathbf{v}_h \mathbf{n})^+ + (2\nu(\bar{\mathbf{u}}_h - \mathbf{u}_h) \cdot \nabla^s \mathbf{v}_h \mathbf{n})^- \, dS \\ + \int_\Gamma (2\nu(\bar{\mathbf{u}}_h - \mathbf{u}_h) \cdot \nabla^s \mathbf{v}_h \mathbf{n})^+ \, d\Gamma = \int_D \mathbf{v}_h \cdot \mathbf{f} \, dx \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \end{aligned} \quad (5.23c)$$

$$\int_S (\bar{\mathbf{v}}_h \cdot \boldsymbol{\sigma}_{d,n|h})^+ + (\bar{\mathbf{v}}_h \cdot \boldsymbol{\sigma}_{d,n|h})^- \, dS + \int_\Gamma \bar{\mathbf{v}}_h \cdot \boldsymbol{\sigma}_{d,n|h} \, d\Gamma = 0 \quad \forall \bar{\mathbf{v}}_h \in \bar{\mathbf{V}}_h, \quad (5.23d)$$

where

$$Q_h := \left\{ p_h \in L^2(D), p_{h|K} \in P_n(K) \quad \forall K \in D \right\}, \quad (5.24)$$

$$\bar{Q}_h := \left\{ \bar{p}_h \in H^1(F), \bar{p}_{h|F} \in P_n(F) \quad \forall F \in D \right\}, \quad (5.25)$$

$$\mathbf{V}_h := \left\{ \mathbf{u}_h \in [L^2(D)]^d, \mathbf{u}_{h|K} \in [P_{n+1}(K)]^d \quad \forall K \in D \right\}, \quad (5.26)$$

$$\bar{\mathbf{V}}_h := \left\{ \bar{\mathbf{u}}_h \in [H^1(F)]^d, \bar{\mathbf{u}}_{h|F} \in [P_{n+1}(F)]^d \quad \forall F \in D \right\}, \quad (5.27)$$

assuming that  $P_n(K)$  and  $P_{n+1}(K)$  denote spaces of Lagrange polynomials on  $K$  of degree  $n > 0$  and  $n + 1$  respectively;  $P_n(F)$  and  $P_{n+1}(F)$  denote spaces of Lagrange polynomials on  $F$  of degree  $n > 0$  and  $n + 1$  respectively; and  $dS$  represents an integral over the interior element facets.

## 5.6.2 Numerical examples

To illustrate the importance of compatible function spaces, the Stokes equations in (5.16) have been coupled to the advection driven scalar transport equation in (5.2) for the domain in Figure 5.1. The advection driven scalar transport equation is modelled using the same formulation as before in equation (5.8) with  $\mathcal{D} = 0$ . The boundary conditions are chosen to represent hydrostatic equilibrium, so  $c_{D,\Gamma} = 0$ ,  $c_{D,P} = 0$ ,  $c_{U,\Gamma} = 0$  and  $c_{U,P} = 0$ . The initial concentration profile is the same as for the Darcy flow example in equation (5.13). The parameters  $\Delta t = 0.001$ ,  $\nu = 1.0$  and  $\alpha = 6n^2$ , where  $n$  is the degree of the polynomials used to represent the elements in equation (5.24).

The importance of using compatible function spaces is again shown by considering both polynomial and sinusoidal forcing terms. The TH formulation in equation (5.18) and the hybrid formulation

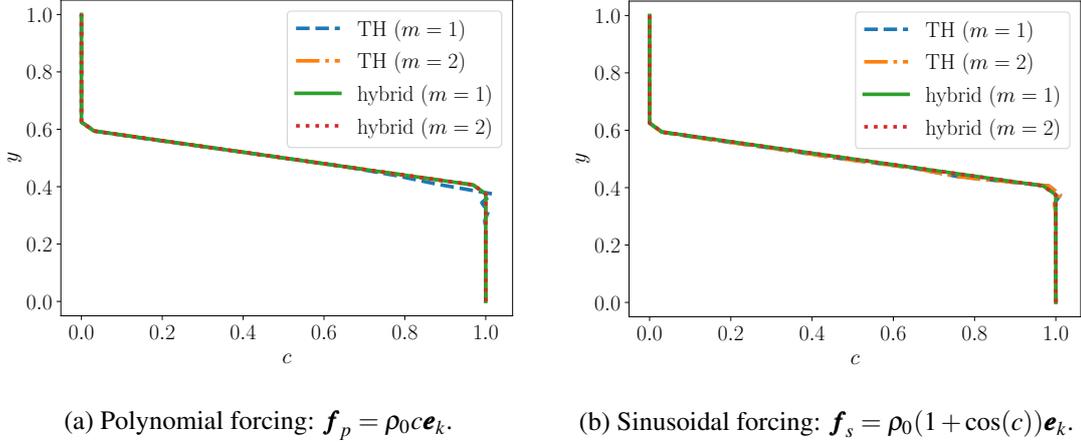


Figure 5.6: Stokes flow concentration profile along  $y$ -axis at  $x = 0.5$  for different forcing terms after  $1500\Delta t$ .

in equation (5.23) are solved for

$$\mathbf{f}_p = \rho_0 c \mathbf{e}_k, \quad (5.28a)$$

$$\mathbf{f}_s = \rho_0(1 + \cos(c)) \mathbf{e}_k, \quad (5.28b)$$

where  $\rho_0 = 1000$ .

From Figure 5.6a, spurious velocities result from first order TH elements with linear forcing,  $\mathbf{f}_p$ , but not for quadratic elements. This is because the TH *a priori* error estimates for  $p_h$  and  $\mathbf{u}_h$  are coupled [15]:

$$\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0 \leq C \left\{ \inf_{\mathbf{v} \in \mathbf{V}_h} \|\mathbf{u} - \mathbf{v}\|_1 + \inf_{q \in Q_h} \|p - q\|_0 \right\}, \quad (5.29)$$

where  $C$  is a constant that is independent of  $h$ ,  $u_h$ ,  $u$ ,  $p_h$  and  $p$ . Therefore, from equation (5.18), second order TH elements are required to match  $\mathbf{f}_p$  point-wise. In contrast, spurious velocities are shown in Figure 5.6b for both linear and quadratic elements because polynomial function spaces cannot contain the sinusoidal forcing term.

The hybrid method does not seem to result in spurious velocities for either  $\mathbf{f}_p$  or  $\mathbf{f}_s$  in Figure 5.6. However, despite the hybrid method being more robust than the TH method, the method only satisfies  $\nabla \cdot \mathbf{u}_h = 0$  point-wise and the pressure still may not be robust to incompatibilities in the function spaces, see John et al. [71]. A new formulation, which is pressure robust, has been proposed by Rhebergen and Wells [99].

## 5.7 Conclusions

In this chapter, the importance of correctly modelling buoyancy driven flows through porous media is illustrated for the Darcy equations coupled to a scalar transport equation. If a primal formulation

is used to solve the Darcy equations, spurious velocities may result if the buoyancy driven forcing cannot match  $\nabla p_h$  point-wise; this always occurs for non-polynomial forcing. In contrast, if a mixed formulation is used, no spurious velocities exist because the error in the pressure and velocity are bound separately. Therefore if  $\mathbf{u} = \mathbf{0}$ ,  $\mathbf{u}_h = \mathbf{0}$  and hydrostatic equilibrium is maintained irrespective of the forcing.

The compatibility of function spaces is also important for the Stokes equations. Despite TH elements representing a mixed function space, the error estimates for the pressure and velocity are coupled. Therefore, it is important to ensure the gradients of the pressure and velocity can match the forcing term point-wise. The hybrid method [77] is an improvement of the TH method since it ensures  $\nabla \cdot \mathbf{u} = 0$  point-wise, but the pressure may not be robust to function space incompatibilities.

In the remaining chapters of this thesis only the mixed Darcy and DG scalar transport equation are considered.

## Chapter 6

# Long term dissolution of CO<sub>2</sub> in background flow<sup>1</sup>

The Darcy and scalar transport models in the previous chapter are used here to investigate the long term behaviour of CO<sub>2</sub> sequestration in the presence of a background hydrological flow. In this example, CO<sub>2</sub> sequestration refers to the long term storage of captured CO<sub>2</sub> in layers of porous rock, which mitigates some of the harmful environmental impacts from generating energy using fossil fuels.

A summary is as follows. The literature about CO<sub>2</sub> sequestration is reviewed in Section 6.1 and the equations that describe the flow of the fluid through the aquifer are non-dimensionalised in Section 6.2. A one dimensional analytical model for the fluid flow through the aquifer is derived in Section 6.3 and the FE model is presented in Section 6.4. The two models are analysed in Section 6.5 and the physical relevance of this modelling is presented in Section 6.6.

### 6.1 CO<sub>2</sub> sequestration

Oceans and geological structures have been identified as potential storage areas for captured CO<sub>2</sub> [9]. Oceans act as a good storage location for CO<sub>2</sub> because they naturally store CO<sub>2</sub> and take up at least 70% of the surface of the earth [84]. However, CO<sub>2</sub> only remains trapped in the oceans for short periods of time and can cause ocean acidification [29].

CO<sub>2</sub> can also be pumped underground and stored in aqueous form in geological aquifers, or layers of porous rock encased in non-porous rock. The aqueous CO<sub>2</sub> migrates along the sedimentary layers of rock, owing to the relative buoyancy of CO<sub>2</sub> and water at depths of 1 – 2km, and ultimately pools in structural highs such as anticlines [69]. The suitable aquifers for CO<sub>2</sub> sequestration are deep to avoid drinking water contamination and to lessen the risk of leakage if cracks are over-pressurised [39]. Depleted oil and gas reservoirs provide potential sites for sequestration because they have proven trapping ability.

---

<sup>1</sup>The chapter is based on Unwin et al. [121].

There has been much research into the processes that govern the long term storage of CO<sub>2</sub>. The integrity of the rock traps, and so the feasibility of gas migrating back to the surface, has already been considered [13, 62, 97, 126]. Therefore, the current focus of research is the buoyancy driven flows that arise when the CO<sub>2</sub> is dissolved in water [64, 82, 94, 100]. CO<sub>2</sub> is soluble in water up to concentrations of a few weight percent, which increases the density of the groundwater when added. Convective dissolution develops beneath the CO<sub>2</sub> trap and the concentration of CO<sub>2</sub> in the fluid increases until the fluid becomes unstable and sinks to the bottom of the aquifer. This unstable fluid is then replaced by unsaturated water. Eventually all the water directly beneath the trap becomes saturated and more dissolution is only possible if this saturated water can be replaced with unsaturated water. This can occur due to a background hydrological flow, lateral convective flows along the aquifer [112, 134], or by vertical exchange flows through fractures, which connect multiple aquifers together [133].

Szulczewski et al. [112] considered the long term evolution of the convective exchange in a horizontal aquifer. They established that following the initial dissolution of CO<sub>2</sub> directly below the trap, a slowly spreading zone of CO<sub>2</sub> enriched groundwater is formed if dissolution continues to occur. This zone is controlled long-term by a balance between the buoyancy driven shear and vertical diffusion of CO<sub>2</sub>. They predicted that the gas may remain trapped in the aquifer for a long time, since the buoyancy driven shear dispersion leads to a slowly waning rate of dissolution. However, they did not consider the impact of a background flow, which may be very important in determining the rate of CO<sub>2</sub> dissolution in aquifers.

## 6.2 Non-dimensional equations

The long term behaviour of CO<sub>2</sub> sequestration is modelled by buoyancy driven Darcy flow, which is dependent on the concentration of the dissolved CO<sub>2</sub>, coupled to a scalar transport equation. The fluid density is defined as

$$\rho = \rho_0 + \beta(c - c_0)\rho_0, \quad (6.1)$$

where  $c \in [c_0, c_D]$  is the concentration of CO<sub>2</sub>,  $\rho_0 \geq 0$  is the initial water density and  $\beta \geq 0$  is an expansion coefficient. The dimensional buoyancy driven forcing term is assumed to be

$$\mathbf{f} = (\rho - \rho_0)g\mathbf{e}_k, \quad (6.2)$$

where  $g$  is the gravitational acceleration and  $\mathbf{e}_k$  is a unit vector in the direction in which gravity acts.

The following non-dimensional scalings are introduced, to recast the Darcy and scalar-transport equations in (5.1) and (5.2) into non-dimensional form, with non-dimensional quantities denoted by a

superscript ‘\*’:

$$\mathbf{u} = \frac{\kappa_0 \beta (c_D - c_0) \rho_0 g}{\mu} \mathbf{u}^*, \quad (6.3)$$

$$p = \beta (c_D - c_0) \rho_0 g H p^*, \quad (6.4)$$

$$t = \frac{H \mu}{\kappa_0 \beta (c_D - c_0) \rho_0 g} t^*, \quad (6.5)$$

$$\kappa = \kappa_0 \kappa^*, \quad (6.6)$$

$$\mathbf{x} = H \mathbf{x}^*, \quad (6.7)$$

where  $\mathbf{u}$  is the velocity,  $H$  is a characteristic height of the domain,  $\kappa$  is the permeability,  $\kappa_0$  is a characteristic permeability,  $\mu$  is the viscosity,  $t$  is time,  $p$  is the pressure and  $\mathbf{x} \in D$ . A scaled concentration  $c^* \in [0, 1]$  is introduced, where

$$c = (c_D - c_0) c^* + c_0. \quad (6.8)$$

The domain of interest for this problem is  $D \in \mathbb{R}^d$ , where  $1 \leq d \leq 3$ , and it is triangulated into non-overlapping cells,  $K$ . The boundary is denoted by  $\Gamma = \partial D$ , where  $\Gamma_c \cup \Gamma_{\text{cap}} \cup \Gamma_B = \partial D$  and  $\Gamma_c \cap \Gamma_{\text{cap}} \cap \Gamma_B = \emptyset$ . The time interval for the model is  $t \in I = [0, t_N)$ . Therefore, in non-dimensional form, the Darcy equations and corresponding boundary conditions for this example read: given  $c^*$ , find  $\mathbf{u}^*$  such that

$$\nabla^* \cdot \mathbf{u}^* = 0 \quad \text{on } D \times I, \quad (6.9a)$$

$$\frac{1}{\kappa^*} \mathbf{u}^* = -\nabla^* p^* + c^* \mathbf{e}_k \quad \text{on } D \times I, \quad (6.9b)$$

$$p = p_D \quad \text{on } \Gamma_c \times I \quad (6.9c)$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_{\text{cap}} \times I \quad (6.9d)$$

$$\mathbf{u} \cdot \mathbf{n} = -u_B \quad \text{on } \Gamma_B \times I, \quad (6.9e)$$

and the scalar transport equations and corresponding boundary conditions in (5.2) read: given  $\mathbf{u}^*$ , find  $c^*$  such that

$$\frac{\partial c^*}{\partial t^*} + \nabla^* c^* \cdot \mathbf{u}^* - \frac{1}{Ra} \nabla^* \cdot \nabla^* c^*(x, \omega) = 0 \quad \text{on } D \times I, \quad (6.10a)$$

$$c^* \mathbf{u}^* \cdot \mathbf{n} = \min(\mathbf{u}^* \cdot \mathbf{n}, 0) c_D \quad \text{on } \Gamma_c \times I, \quad (6.10b)$$

$$\frac{1}{Ra} \nabla^* c^* \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_c \times I, \quad (6.10c)$$

$$\left( -\frac{1}{Ra} \nabla^* c^* + c^* \mathbf{u}^* \right) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_{\text{cap}} \cup \Gamma_B \times I, \quad (6.10d)$$

$$c^*(x, 0) = c_0 \quad \text{on } D, \quad (6.10e)$$

where the constant  $Ra$  is a Rayleigh number,

$$Ra = \frac{\kappa_0 \beta (c_D - c_0) \rho_0 g H}{\mu \mathcal{D}}, \quad (6.11)$$

which describes the ratio between the buoyancy and diffusivity of the fluid. The boundary condition in (6.10b) is chosen so that the advective flow is only prescribed on the inflow boundary via a density based forcing term  $\mathbf{f}(c)$ . Since all equations in the remainder of this chapter are non-dimensional, the ‘ $\star$ ’ notation is dropped.

### 6.3 Analytical model

Consider the two dimensional domain in Figure 6.1a, which represents a high aspect ratio aquifer formed by a fold in the geological strata that extends much further in the direction normal to the page than the width of the fold. This aquifer is assumed to be horizontal and uniform in shape, and a background hydrological flow is assumed to be supplied from the right-hand side boundary, see Figure 6.1b.

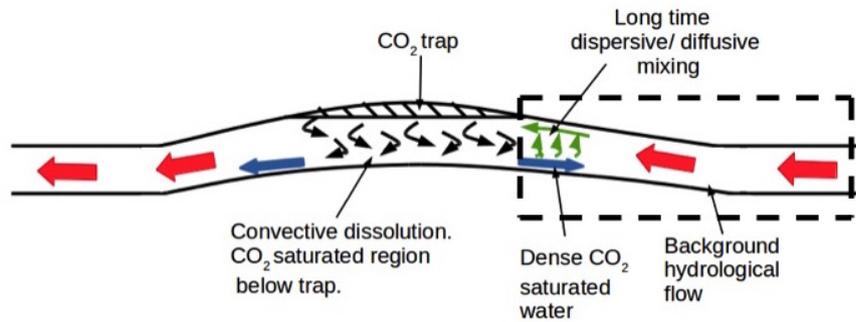
Directly below the CO<sub>2</sub> trap, the groundwater is assumed to be fully saturated with CO<sub>2</sub> due to convective dissolution [112, 133]. This saturated water can be moved downstream due to the background flow, but little is understood about the upstream impact of the dense saturated fluid and groundwater mixing. Therefore,  $\Gamma_c$  is chosen to be upstream of the fully saturated region. It is also assumed that the background hydrological flows in the model are constant in time and that the CO<sub>2</sub> does not react with the layers of rock.

Different flow regimes develop depending on the non-dimensional parameters  $u_B$  and  $Ra$ , in equations (6.9e) and (6.11). The parameter  $u_B$  is the ratio of the background flow speed to the buoyancy driven flow speed, and the parameter  $Ra$  is the ratio of the buoyancy driven flow speed to the diffusion. Three different regimes develop, which are derived below: a gravity intrusion regime, a dispersion dominated regime and a diffusion dominated regime.

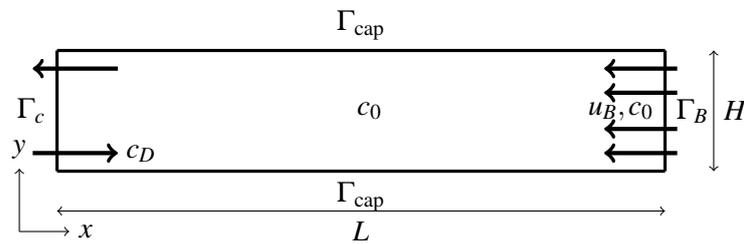
#### 6.3.1 Gravity intrusion regime

For advection dominated flows, where  $1/Ra$  in equation (6.10a) is small, a nearly static intrusion of dense CO<sub>2</sub> saturated fluid extends upstream into the aquifer. This intrusion is modelled as a sharp concentration front at a height  $h(x)$  above the lower boundary, where  $0 \leq h(x) \leq 1$ , with approximately hydrostatic pressure [68, 133]. Therefore in the equilibrium state, the intrusion pressure gradient in the  $x$ -direction  $dh/dx$  balances the pressure gradient associated with the background flow above the intrusion  $-u_B/1-h$ , since  $u_B$  is defined in the opposite direction to the flow:

$$-(1-h) \frac{dh}{dx} = u_B. \quad (6.12)$$



(a) Cartoon of geological problem.



(b) Diagram of model problem.

Figure 6.1: Diagrams of problem of interest. Fig (a) includes a cartoon of the geological problem with region of interest indicated by dashed box. The large red arrows represent the direction of the background hydrological flow, while the smaller black curved arrows represent the convective mixing of the CO<sub>2</sub>. The smaller blue arrow inside the dashed box represents the buoyancy driven flow of dense CO<sub>2</sub> saturated water flowing upstream into the background hydrological flow. Fig (b) depicts the model problem for analysis and simulation [121].

The shape of the intrusion can be found by integrating equation (6.12) with boundary condition  $h(0) = 1$ ,

$$h(x) = 1 - \sqrt{2u_B x}. \quad (6.13)$$

The maximum distance the intrusion,  $X_{\text{int}}$  will extend into the aquifer is the distance at which the height of the aquifer, in equation (6.13), is zero. Therefore,

$$X_{\text{int}} = \frac{1}{2u_B}. \quad (6.14)$$

The dimensionless time taken for the background flow to pass the intrusion is calculated by

$$\tau_{\text{int}} = \int_{X_{\text{int}}}^0 \frac{1}{u} dx, \quad (6.15)$$

where  $u = -u_B/(1-h)$ . Therefore,

$$\tau_{\text{int}} = \int_{X_{\text{int}}}^0 -\frac{(1-h)}{u_B} dx. \quad (6.16)$$

Substituting the height of the intrusion in the  $x$ -direction from equation (6.13) into equation (6.16),

$$\tau_{\text{int}} = \frac{1}{3u_B^2}. \quad (6.17)$$

For the sharp interface to remain between the regions of low and high concentration, which is indicative of this regime,  $\tau_{\text{int}}$  needs to be small relative to the dimensionless diffusion time,  $\tau_{\text{diff}} = Ra$ . Therefore, to enter the gravity intrusion regime  $Ra \gg 1/3u_B^2$ , or  $u_B^2 Ra \gg 1/3$ . This is simplified in the following analysis to

$$u_B^2 Ra \gg 1. \quad (6.18)$$

As discussed further in Unwin et al. [121], the gravity intrusion extends less far into the domain as  $u_B$  increases. This is because  $X_{\text{int}} \leq 1/u_B$ . Therefore, it is expected that the time scale  $1/u_B^2$  is an upper bound for the advection time  $\sim X_{\text{int}}/u_B$ . It is also important to consider diffusion since  $X_{\text{int}} < 1$  for large values of  $u_B$ . Diffusion scales with  $X_{\text{int}}^2 Ra$  and so suggests that  $Ra = 1$  is a good approximation for the upper bound of the transition between the diffusive and intrusive regimes.

### 6.3.2 Buoyancy-driven shear dispersion regimes

When  $u_B^2 Ra \ll 1$ , vertical diffusion is fast, the vertical concentration gradient is small, and the horizontal concentration gradient may be significant. For these values of  $u_B$  and  $Ra$ , two different parameter regimes can be entered: the buoyancy driven shear regime or the diffusion dominated regime. Similar to Unwin et al. [121], the subsequent derivations follow Szulczewski et al. [112] and Woods [133].

Consider the CO<sub>2</sub> concentration, which is decomposed into depth-average and fluctuation parts denoted by the bar and hat notation respectively:

$$c(x, y, t) = \bar{c}(x, t) + \hat{c}(x, y, t), \quad (6.19)$$

where

$$\bar{c} = \int_0^1 c \, dy. \quad (6.20)$$

Assuming that  $\hat{c}$  is small, that the horizontal dimension of the aquifer is much greater than the vertical dimension, and that the non-hydrostatic vertical pressure gradient is small, the pressure is assumed to vary hydrostatically and can be approximated by

$$p = p_0 - y\bar{c}, \quad (6.21)$$

where  $p_0 = p_0(x, t)$  is the pressure at the base of the aquifer. The velocity of the fluid  $\mathbf{u} = (u, v)$ , can be expressed as the sum of the average velocity across the depth of the aquifer,  $\bar{\mathbf{u}} = (\bar{u}, \bar{v})$ , and a fluctuation  $\hat{\mathbf{u}} = (\hat{u}, \hat{v})$

$$\mathbf{u}(x, y, t) = \bar{\mathbf{u}}(x, t) + \hat{\mathbf{u}}(x, y, t), \quad (6.22)$$

where:

$$\bar{\mathbf{u}} = \int_0^1 \mathbf{u} \, dy. \quad (6.23)$$

Assuming the Darcy velocity balances  $-dp/dx$ , the velocity along the aquifer,  $u$ , is the derivative in the  $x$ -direction of the pressure, which is given in equation (6.21),

$$u = -\frac{\partial p_0}{\partial x} + y\frac{\partial \bar{c}}{\partial x}. \quad (6.24)$$

Substituting  $u$ , from equation (6.24), into equation (6.23) and integrating gives the average velocity,

$$\bar{u} = -\frac{\partial p_0}{\partial x} + \frac{1}{2}\frac{\partial \bar{c}}{\partial x}. \quad (6.25)$$

The velocity fluctuation along the aquifer can be found by substituting equations (6.24) and (6.25) into (6.22) and rearranging,

$$\hat{u} = \frac{\partial \bar{c}}{\partial x} \left( y - \frac{1}{2} \right). \quad (6.26)$$

Considering the decomposition of the concentration and velocity given in equations (6.19) and (6.22), and the continuity equation in (6.9a), the scalar transport equation in (6.10a) can be written as

$$\frac{\partial \bar{c}}{\partial t} + \frac{\partial \hat{c}}{\partial t} + \bar{u}\frac{\partial \bar{c}}{\partial x} + \bar{u}\frac{\partial \hat{c}}{\partial x} + \hat{u}\frac{\partial \bar{c}}{\partial x} + \hat{u}\frac{\partial \hat{c}}{\partial x} + \bar{v}\frac{\partial \bar{c}}{\partial y} + \bar{v}\frac{\partial \hat{c}}{\partial y} = \frac{1}{Ra}\frac{\partial^2 \bar{c}}{\partial x^2} + \frac{1}{Ra}\frac{\partial^2 \hat{c}}{\partial x^2} + \frac{1}{Ra}\frac{\partial^2 \hat{c}}{\partial y^2}. \quad (6.27)$$

Taking the average of equation (6.27) over the depth of the aquifer results in an equation for the transport of the mean concentration in the  $x$ -direction

$$\frac{\partial \bar{c}}{\partial t} + \bar{u} \frac{\partial \bar{c}}{\partial x} + \overline{\hat{u} \frac{\partial \bar{c}}{\partial x}} = \frac{1}{Ra} \frac{\partial^2 \bar{c}}{\partial x^2}. \quad (6.28)$$

The equation for the concentration fluctuations in the aquifer can be found by subtracting equation (6.28) from the transport equation in (6.27)

$$\frac{\partial \hat{c}}{\partial t} + \hat{u} \frac{\partial \bar{c}}{\partial x} + \bar{u} \frac{\partial \hat{c}}{\partial x} + \hat{u} \frac{\partial \hat{c}}{\partial x} + \hat{v} \frac{\partial \hat{c}}{\partial y} + \bar{v} \frac{\partial \hat{c}}{\partial y} = \frac{1}{Ra} \left( \frac{\partial^2 \hat{c}}{\partial x^2} + \frac{\partial^2 \hat{c}}{\partial y^2} \right) + \hat{u} \frac{\partial \bar{c}}{\partial x}. \quad (6.29)$$

After long time periods, the dominant balance in equation (6.29) is between the distortion of  $\bar{c}$ , due to the velocity fluctuations, and the cross layer transport mechanism given by  $Ra$  [114]. This gives rise to the following balance:

$$\frac{1}{Ra} \frac{\partial^2 \hat{c}}{\partial y^2} = \hat{u} \frac{\partial \bar{c}}{\partial x}. \quad (6.30)$$

By substituting equation (6.26) into equation (6.30),

$$\frac{\partial^2 \hat{c}}{\partial y^2} = Ra \left( \frac{\partial \bar{c}}{\partial x} \right)^2 \left( y - \frac{1}{2} \right), \quad (6.31)$$

the concentration fluctuation  $\hat{c}$  can be found by integrating equation (6.31) twice, with the boundary conditions  $\partial \hat{c} / \partial y = 0$  at  $y = 0$  and  $\int_0^1 \hat{c} dy = 0$ ,

$$\hat{c} = Ra \left( \frac{\partial \bar{c}}{\partial x} \right)^2 \left( \frac{y^3}{6} - \frac{y^2}{4} + \frac{1}{24} \right). \quad (6.32)$$

Substituting equations (6.26) and (6.32) into equation (6.28), the depth averaged transport equation can be expressed as:

$$\frac{\partial \bar{c}}{\partial t} - u_B \frac{\partial \bar{c}}{\partial x} = \frac{1}{Ra} \frac{\partial^2 \bar{c}}{\partial x^2} + \frac{Ra}{120} \frac{\partial}{\partial x} \left( \frac{\partial \bar{c}}{\partial x} \right)^3, \quad (6.33)$$

where  $u_B$  is the background flow from right-to-left that corresponds to the boundary condition in (6.9e).

After long time periods, the solution becomes steady,  $\partial \bar{c} / \partial t = 0$ , and by rescaling the  $x$  coordinate with

$$x = \left( \frac{Ra}{120u_B} \right)^{\frac{1}{3}} x', \quad (6.34)$$

and rearranging, the transport equation in (6.33) can be re-written as

$$-\frac{\partial \bar{c}}{\partial x'} = \left( \frac{120u_B}{Ra^4} \right)^{\frac{1}{3}} \frac{\partial^2 \bar{c}}{\partial x'^2} + \frac{\partial}{\partial x'} \left( \frac{\partial \bar{c}}{\partial x'} \right)^3. \quad (6.35)$$

This can be simplified to

$$-\frac{\partial \bar{c}}{\partial x'} = \alpha \frac{\partial^2 \bar{c}}{\partial x'^2} + \frac{\partial}{\partial x'} \left( \frac{\partial \bar{c}}{\partial x'} \right)^3, \quad (6.36)$$

where

$$\alpha = \left( \frac{120}{Ra^4 u_B^2} \right)^{\frac{1}{3}}. \quad (6.37)$$

The value of  $\alpha$  determines which of the two regime is entered. Diffusion dominates for large values of  $\alpha$  whereas, dispersion is dominant for small values of  $\alpha$ . No analytical solutions to equation (6.36) have been found, but useful analytical assumptions can be derived by considering the limits of  $\alpha \ll 1$  and  $\alpha \gg 1$ .

### Dispersion dominated regime

In the limit of equation (6.36), where  $\alpha \ll 1$ , the buoyancy driven dispersion balances the advection and the rescaled depth average scalar transport equation becomes

$$-\frac{\partial \bar{c}}{\partial x'} = \frac{\partial}{\partial x'} \left( \frac{\partial \bar{c}}{\partial x'} \right)^3. \quad (6.38)$$

Integrating equation (6.38) once, and assuming the coefficient of integration is zero,

$$-\bar{c} = \left( \frac{\partial \bar{c}}{\partial x'} \right)^3. \quad (6.39)$$

Rearranging equation (6.39),

$$-\frac{1}{\bar{c}^{\frac{1}{3}}} \frac{\partial \bar{c}}{\partial x'} = 1, \quad (6.40)$$

and integrating equation (6.40) with the boundary condition  $\bar{c} = 1$  at  $x' = 0$ ,

$$x' = 1 - \frac{3}{2} \bar{c}^{\frac{2}{3}}. \quad (6.41)$$

Substituting  $x'$  from equation (6.41) into equation (6.34) and rearranging, the buoyancy driven dispersion regime concentration profile is

$$\bar{c} = \left( 1 - \frac{2}{3} \left( \frac{120 u_B}{Ra} \right)^{\frac{1}{3}} x \right)^{\frac{3}{2}}. \quad (6.42)$$

In this regime, the concentrated fluid should not advance further upstream than

$$X_{\text{dis}} = \frac{3}{2} \left( \frac{Ra}{120 u_B} \right)^{\frac{1}{3}}, \quad (6.43)$$

since this is the value of  $x$  for  $c = 0$ .

### Diffusion dominated regime

In the limit of equation (6.36) where  $\alpha \gg 1$ , diffusion balances the advection and the rescaled depth averaged scalar transport equation becomes

$$-\frac{\partial \bar{c}}{\partial x'} = \alpha \frac{\partial^2 \bar{c}}{\partial x'^2}. \quad (6.44)$$

Integrating equation (6.44), and setting the constant of integration to zero,

$$-\bar{c} = \alpha \frac{\partial \bar{c}}{\partial x'}. \quad (6.45)$$

Rearranging equation (6.45),

$$-\frac{1}{\alpha} = \frac{1}{\bar{c}} \frac{\partial \bar{c}}{\partial x'}, \quad (6.46)$$

and integrating equation (6.46) with the boundary condition  $\bar{c} = 1$  at  $x' = 0$ ,

$$-x' = \alpha \ln(\bar{c}). \quad (6.47)$$

Substituting  $x'$  and  $\alpha$  from equations (6.34) and (6.37) respectively and rearranging, the vertically averaged diffusion dominated concentration profile is

$$\bar{c} = e^{-Ra u_B x}. \quad (6.48)$$

For this regime, the concentrated fluid should not advance further upstream than

$$X_{\text{diff}} = \frac{\ln c_{\text{diff}}}{Ra u_B}, \quad (6.49)$$

where  $c_{\text{diff}}$  is the concentration of CO<sub>2</sub> in the groundwater that is considered negligible.

### Regime transition

The value of  $\alpha$  at which the dispersion and diffusion regimes transition, can be found by equating  $X_{\text{dis}}$  from equation (6.43) and  $X_{\text{dif}}$  from equation (6.49)

$$\alpha_e = \frac{3}{2 \ln c_{\text{diff}}}. \quad (6.50)$$

The point at which the concentration becomes negligible is chosen to be  $c_{\text{diff}} = 0.01$ . For this value,  $\alpha_e = 0.326$ .

Figure 6.2 shows how  $X_{\text{diff}}$  and  $X_{\text{dis}}$  vary as function of  $Ra$  for two different values of  $u_B$  to illustrate which regime controls the distance that the CO<sub>2</sub> propagates upstream. The flow belongs to the regime with the longer upstream propagation distance. Therefore, when the Rayleigh number is large the dispersion regime is entered. In contrast a transition to diffusive regimes occurs for small Rayleigh numbers. This is expected since  $Ra \propto 1/\mathcal{D}$ , see equation (6.11).

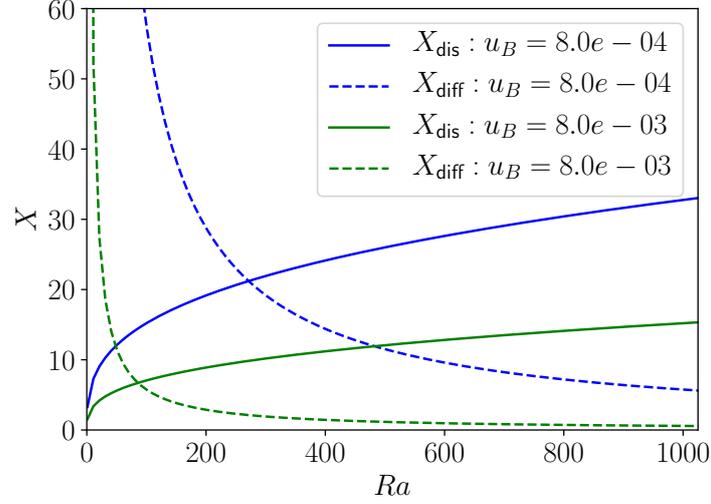


Figure 6.2: Illustration of the distance that CO<sub>2</sub> extends upstream in the dispersion and diffusion limit as a function of  $Ra$  for two different values of  $u_B$  [121].

### 6.3.3 Regime differentiation

From the above derivations, one of three different regimes can occur if  $u_B < 1$ , see Figure 6.3. This figure is important since a background flow has not been considered in previous work, such as Szulczewski et al. [112] and Woods [133].

Gravity intrusions occur when  $u_B^2 Ra \gg 1$  whereas, depending on the value of  $\alpha$  (6.37), either a dispersive or diffusive regime is entered if  $u_B^2 Ra \ll 1$ . When  $u_B$  approaches one, the above analysis suggests that transition between the dispersion and intrusive regimes and the dispersive and diffusive regimes converge. From Figure 6.3, the regime transition occurs at  $Ra = 4.93$ . This is similar to the upper bound of  $Ra = 1$  for the transition between the intrusive and diffusive regimes found in Section 6.3.1.

## 6.4 Numerical model

A mixed method is used to solve the Darcy equations in (6.9), as explained in Section 5.4, to ensure local mass conservation. Therefore, the finite element formulation reads: given  $c_h^{n+1} \in W_h$ , find  $p_h \in Q_h$  and  $\mathbf{u}_h \in \mathbf{V}_h$  such that

$$\int_D q_h \nabla \cdot \mathbf{u}_h \, dx = 0 \quad \forall q_h \in Q_h, \quad (6.51a)$$

$$\int_D \frac{1}{\mathbf{K}} \mathbf{u}_h \cdot \mathbf{v}_h \, dx - \int_D p_h \nabla \cdot \mathbf{v}_h \, dx = \int_D c \mathbf{e}_k \cdot \mathbf{v}_h \, dx - \int_{\Gamma_A} p_D \mathbf{v}_h \cdot \mathbf{n} \, d\Gamma \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \quad (6.51b)$$

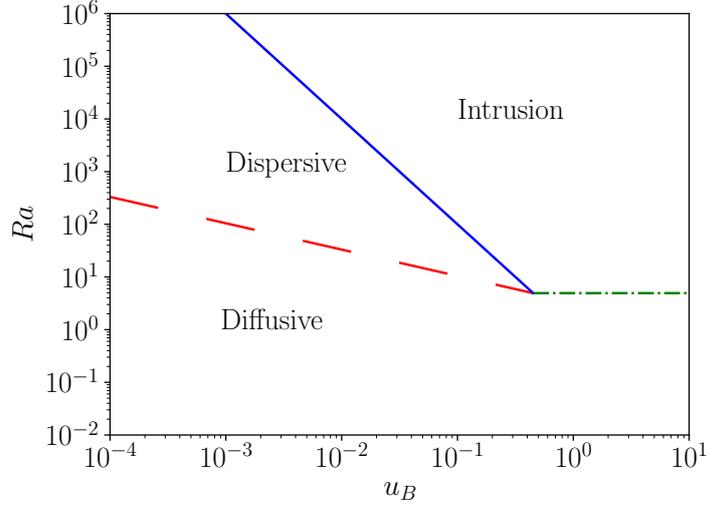


Figure 6.3: Illustration of different regimes. The red dashed line denotes  $\alpha = 1$  and so separates the dispersive and diffusive lines. The blue solid line denotes  $Ra = 1/u_B^2$  and for  $u_B < 1$  separates the intrusive and dispersive regimes. The green dot dashed line denotes  $Ra = 4.93$ . This is an upper bound on the transition between the diffusive and intrusive regimes for  $u_B > 1$ .

where

$$W_h := \{c_h^{n+1} \in L^2(D), c_{h|K}^{n+1} \in P_k(K) \forall K \in \mathcal{D}\}, \quad (6.52)$$

$$Q_h := \left\{ p_h \in L^2(D), p_{h|K} \in P_m(K) \forall K \in \mathcal{D} \right\}, \quad (6.53)$$

$$\mathbf{V}_h := \left\{ \mathbf{u}_h \in H(\text{div}, D), \mathbf{u}_{h|K} \in \text{BDM}_{m+1}(K) \forall K \in \mathcal{D} \right\}, \quad (6.54)$$

and  $P_k(K)$  and  $P_m(K)$  denote spaces of Lagrange polynomials on  $K$  of degree  $m$  and  $k$  respectively and  $\text{BDM}_{m+1}(K)$  denotes a space of BDM elements [16] with degree  $m + 1$ .

A non-dimensional fully up-winded discontinuous Galerkin finite element method is used to discretise the scalar transport equation in (6.10). Assuming a backward Euler time discretisation scheme, the formulation of the scalar transport equation reads: given  $\mathbf{u}_h \in \mathbf{V}_h$ , find  $c_h^{n+1} \in W_h$  such

that

$$\begin{aligned}
& \int_D \sigma_h \frac{c_h^{n+1} - c_h^n}{dt} dx - \int_D c_h^{n+1} \nabla \sigma_h \cdot \mathbf{u}_h dx + \int_D \frac{1}{Ra} \nabla \sigma_h \cdot \nabla c_h^{n+1} dx \\
& \quad + \int_{\Gamma_C} \sigma_h \left( \min(\mathbf{u}_h \cdot \mathbf{n}, 0) c_D + \max(\mathbf{u}_h \cdot \mathbf{n}, 0) c_h^{n+1} \right) d\Gamma \\
& \quad + \int_S \llbracket \sigma_h \rrbracket \left( \max(\mathbf{u}_h \cdot \mathbf{n}, 0)^+ c_h^{n+1,+} - \max(\mathbf{u}_h \cdot \mathbf{n}, 0)^- c_h^{n+1,-} \right) dS \\
& \quad + \int_S \frac{1}{Ra} \frac{\gamma}{h^+} \llbracket \boldsymbol{\sigma}_n \rrbracket \cdot \llbracket \mathbf{c}_n \rrbracket dS - \int_S \frac{1}{Ra} \{ \nabla \sigma_h \} \cdot \llbracket \mathbf{c}_n \rrbracket dS - \int_S \frac{1}{Ra} \{ \nabla c_h^{n+1} \} \cdot \llbracket \boldsymbol{\sigma}_n \rrbracket dS = 0 \quad \forall \sigma_h \in W_h,
\end{aligned} \tag{6.55}$$

where  $\gamma \geq 0$ ,  $W_h$  is the function space defined in equation (6.52), and

$$\llbracket \sigma_h \rrbracket = \sigma_h^+ - \sigma_h^-, \tag{6.56a}$$

$$\llbracket \boldsymbol{\sigma}_n \rrbracket = \sigma_h^+ \mathbf{n}^+ - \sigma_h^- \mathbf{n}^-, \tag{6.56b}$$

$$\llbracket \mathbf{c}_n \rrbracket = c_h^{n+1,+} \mathbf{n}^+ - c_h^{n+1,-} \mathbf{n}^-, \tag{6.56c}$$

$$\{ \nabla \sigma_h \} = \frac{1}{2} (\nabla \sigma_h^+ + \nabla \sigma_h^-), \tag{6.56d}$$

$$\{ \nabla c_h \} = \frac{1}{2} (\nabla c_h^{n+1,+} + \nabla c_h^{n+1,-}) \tag{6.56e}$$

Similar to Section 5.3.2, the superscript ‘+’ and ‘-’ refer to opposite sides of the interior cell edge, denoted by  $S$ , and ‘ $\llbracket \cdot \rrbracket$ ’ refers to a jump and ‘ $\{ \cdot \}$ ’ to an average across the interior cell edges.

For this example, the domain  $D = [0, 100] \times [0, 1]$  to ensure that the aquifer length is much greater than the height. From equations (6.9c) and (6.10b), the boundary conditions  $p_D = -y$  and  $c_D = 1$ , from equation (6.9b)  $\kappa = 1$ , and from equation (6.55)  $\gamma = 5$ . To simulate the different regimes,  $u_B$  is fixed at  $8 \times 10^{-4}$  and  $Ra$  is altered to vary  $\alpha$ , see equation (6.37). The function spaces are such that  $k = 1$  and  $m = 0$  from equations (6.52) and (6.53) respectively. The FEniCS libraries [4] are used to solve equations (6.51) and (6.55) and the code for the simulations can be found as Unwin and Wells [120].

## 6.5 Model comparison

The concentration and velocity profiles for the three different regimes are considered in this section for background flows with varying velocities. First weak background flows are considered with  $u_B < 1$  and then strong background flows with  $u_B \geq 1$ , see Figure 6.3.

### 6.5.1 Weak background flow: $u_B < 1$

Figure 6.4 shows the two dimensional numerical concentration profiles for  $u_B = 0.1$  and three different values of  $Ra$ . This illustrates the three different regimes that can be entered: gravity

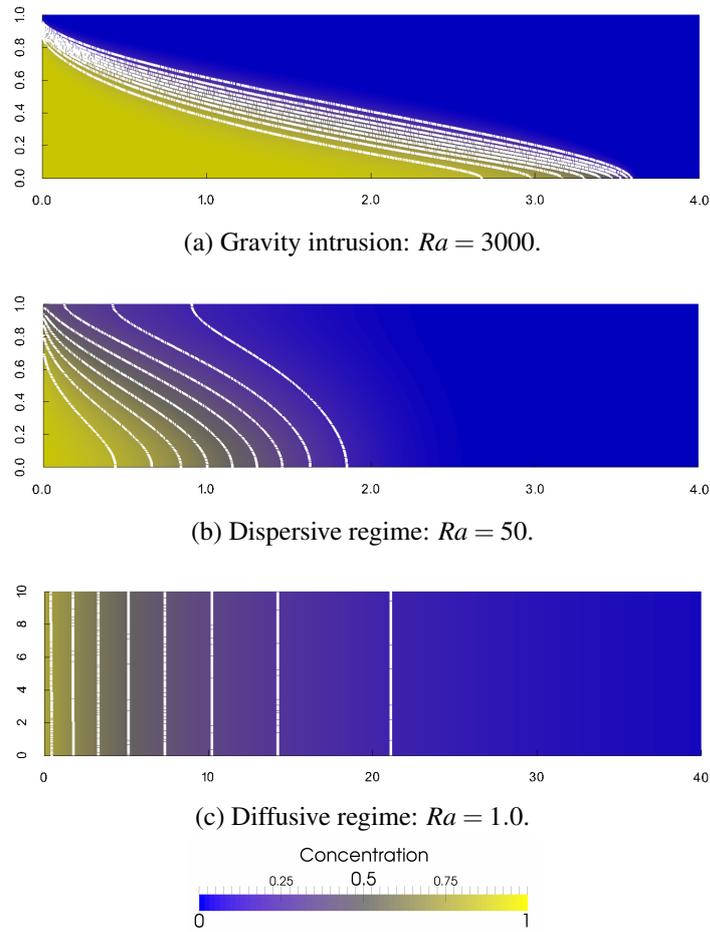


Figure 6.4: Concentration contours for  $u_B = 0.1$  and different values of  $Ra$ . The colours show the concentration locally as defined by the scale in the figure and the contour lines are shown at equal intervals of 0.1 from 0.1 to 0.9. The  $x$ -axis is longer in (c) than (a) or (b) to display the full diffusive regime [121].

intrusion, dispersion dominated and diffusion dominated. First, the gravity intrusion is characterised in Figure 6.4a by the sharp interface between the regions of high and low concentration, which is indicated by the close contour lines. Second, the dispersion regime is characterised in Figure 6.4b by the curved contour lines that are formed from the concentrated fluid slumping to the bottom of the domain and the strong flow recirculation. Third, the diffusion regime is characterised in Figure 6.4c by the nearly straight contour lines that move across the flow linearly. The concentration and velocity profiles for these three different flow regimes are analysed below.

### Concentration profiles

The concentration profiles for the analytical and numerical models can be compared by computing the vertical concentration average,  $\bar{c}$ , for the steady two dimensional finite element model. Since no ana-

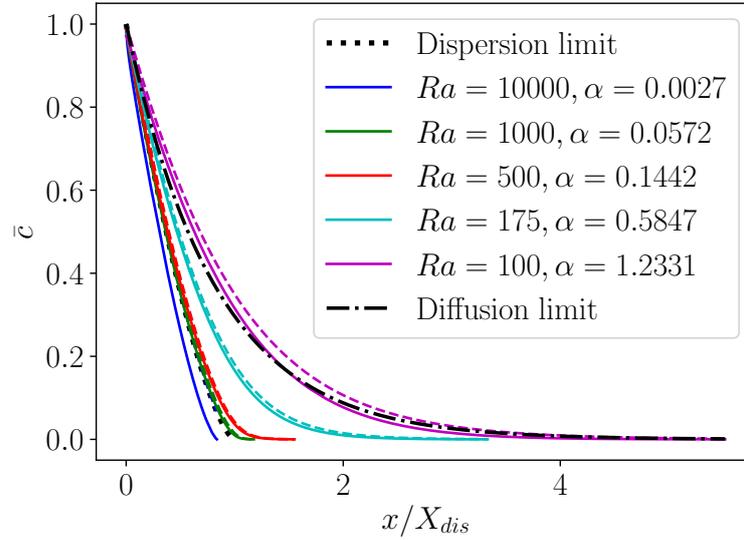


Figure 6.5: Computed and analytical variation of  $\bar{c}$  along the  $x$  dimension in the domain for different values of  $Ra$  when  $u_B = 8 \times 10^{-4}$ . The solid lines represent the vertically averaged two dimensional numerical solution and the dashed lines represent the one dimensional solution [121].

lytical solution can be found for the one dimensional analytical model, the solution to equation (6.36) can be approximated numerically using the FEniCS libraries. The full finite element formulation can be found in Unwin and Wells [120].

Figure 6.5 shows how  $\bar{c}$  varies along the  $x$  dimension of the domain for both numerical and analytical models,  $u_B = 8 \times 10^{-4}$  and values of  $Ra$  between 100 and 10,000. There is good agreement of the one dimensional and vertically averaged two dimensional models between the dispersion limit and the diffusion limit. However, as expected in the limiting cases of small or very large  $Ra$ , the one dimensional and two dimensional numerical models diverge. This is because the analytical modelling assumptions do not capture all the physics of the flow in those areas.

The flow begins to enter the gravity intrusion regime for  $Ra = 10,000$ , so neither the diffusive or dispersive concentration profiles match the vertically averaged concentration profile in Figure 6.5. This is because there are no significant concentration fluctuations in this regime along the height of the aquifer and the model more closely follows the height profile given in equation (6.13).

The gravity intrusion profile is compared to the two dimensional numerical solution in Figure 6.6 for  $Ra = 3000$  and  $u_B = 0.1$ . These parameters result in flow that is definitely in the gravity intrusion regime, see Figure 6.3. There is a reasonable match between the analytical and two dimensional numerical models for  $x < 3.5$ . However, further into the domain the fluid recirculates weakly in a diffusive boundary layer, which is not present in the analytical model. This suggests that the two dimensional model is well formulated and the one dimensional models are a good approximation of the flow, where the analytical assumptions hold.

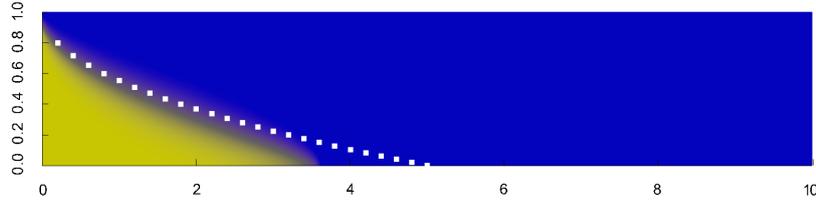


Figure 6.6: Concentration as a function of position in the domain for the gravity intrusion regime, with  $u_B = 0.1$  and  $Ra = 3000$ . The predicted analytical shape of the intrusion, from equation (6.13), is overlain as a dotted white line. The scale is the same as shown in Figure 6.4 [121].

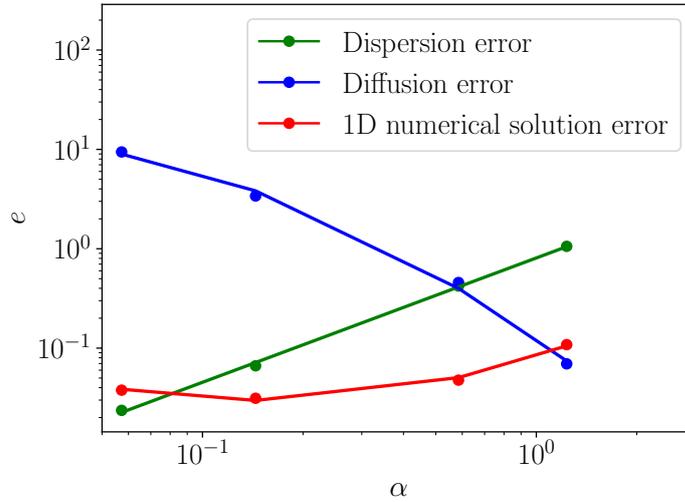


Figure 6.7: Normalised  $L^1$  error norm for the computed two dimensional numeric and analytical values of  $\bar{c}$ .

To investigate further how closely the one dimensional analytical models match the numerical model, the following error norm has been computed

$$e = \frac{\int |\bar{c}_T - \bar{c}_N| dx}{\int \bar{c}_T dx}, \quad (6.57)$$

where  $\bar{c}_N$  is the vertically averaged two dimensional numerical concentration profile and  $\bar{c}_T$  is one of three analytical models: the dispersion profile from equation (6.42), the diffusion profile from equation (6.48) and the full one dimensional numerical model from equation (6.36). The error for the three analytical models has been calculated in Figure 6.7, for four different values of  $\alpha$ . As expected, when  $\alpha \ll 1$ , the error between the dispersion dominated solution and  $\bar{c}_N$  is small since the flow is in the dispersive regime. In contrast, when  $\alpha \gg 1$ , the error between the diffusion dominated solution and  $\bar{c}_N$  is small since the flow is in the diffusive regime. The error between  $\bar{c}_N$  and the full one dimensional solution is always small, since the model represents the physics of the flow well for

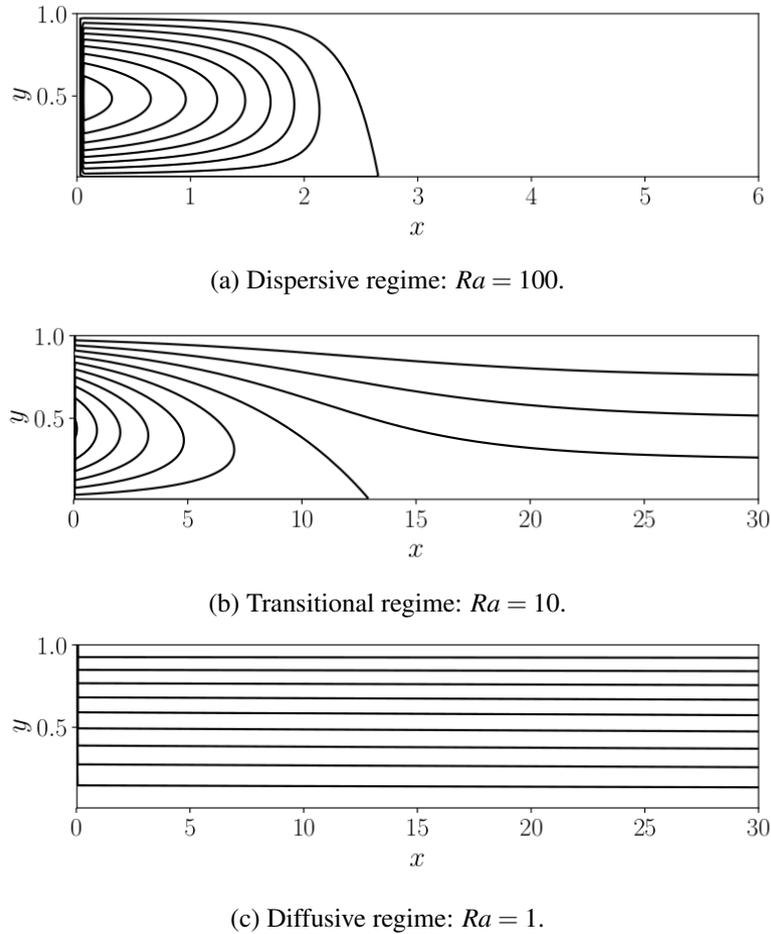


Figure 6.8: Typical streamlines of the flow for  $u_B = 0.01$  and  $Ra = 100, 10$  and  $1$ .

the whole regime. However, the increase in the error for the higher  $\alpha$  values is a result of the one dimensional model better representing the dispersive regime than the diffusive one.

### Velocity profiles

The velocity streamlines are presented in Figure 6.8 for  $u_B = 0.01$  and three different values of  $Ra$  that correspond to a dispersion regime, a transitional regime and a diffusive regime. For large values of  $Ra$ , the groundwater is diverted towards the top of the aquifer and a recirculation region arises upstream, which is characterised by the curved streamlines in Figure 6.8a. In contrast, the streamlines straighten as  $Ra$  decreases and transition to a diffusion driven regime occurs. In the diffusion limit in Figure 6.8c, the streamlines are uniformly horizontal. This shows that the background flow is dominant in the diffusion regime.

In the buoyancy driven shear regime,  $\alpha \leq 1$  and  $u_B^2 Ra \leq 1$ , linear velocity fluctuations are predicted from the analytical model derivation. If the velocity fluctuations  $\hat{u}$  are rescaled with the depth average concentration gradient, equation (6.26) suggests that  $\hat{u} \partial \bar{c} / \partial x = -0.5$  for  $y = 0$  and  $\hat{u} \partial \bar{c} / \partial x = 0.5$

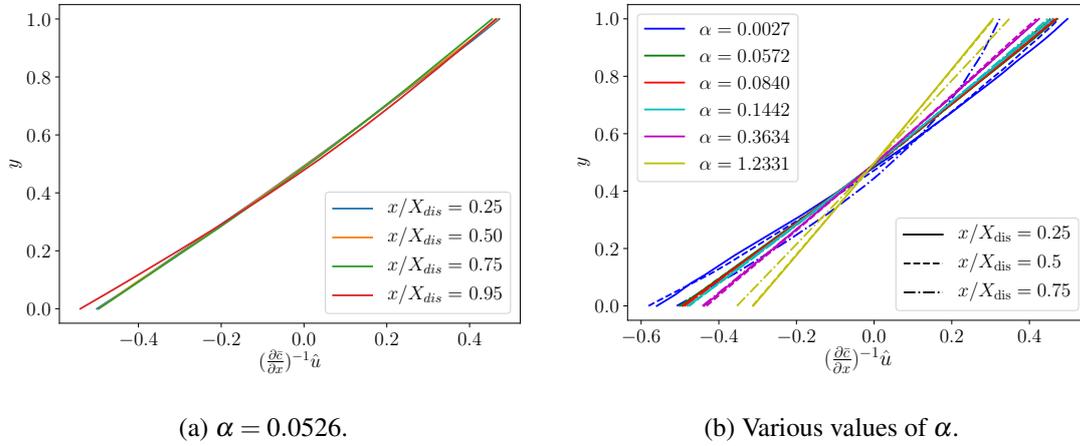


Figure 6.9: Scaled velocity profiles at different points along the domain for different values of  $\alpha$  with  $u_B = 8 \times 10^{-4}$ . Fig (a) shows the velocity fluctuations  $\hat{u}$  scaled by concentration gradient for  $Ra = 1000$  and  $u_B = 8 \times 10^{-4}$ , which corresponds to  $\alpha = 0.0527$  from equation (6.37). Fig (b) shows the velocity fluctuations  $\hat{u}$  for different values of  $\alpha$  at different values of  $x$  [121].

for  $y = 1$ . The scaled velocity profiles are shown in Figure 6.9a for different distances along the  $x$  axis,  $Ra = 1000$  and  $u_B = 8 \times 10^{-4}$ , which correspond to a dispersion dominated flow. While  $x/X_{dis}$  is small, the velocity profiles considered are for distances that are far from the stall point, or the maximum distance the saturated fluid will enter the domain. Here, the velocity fluctuations are linear and have a range of  $[-0.5, 0.5]$ . In contrast, the velocity profiles near the stall points deviate from the simplified analytical theory.

The scaled velocity fluctuations are shown in Figure 6.9b for the three different flow regimes with various values of  $\alpha$ . As the flow transitions from dispersion to diffusion dominated, the model for  $\hat{u}$  in equation (6.26) becomes less applicable. For  $0.0572 \leq x \leq 0.1442$ , the scaled velocity profiles have the predicted linear relationship in equation (6.26). However, as alpha increases the diffusion transport regime becomes more dominant and the fluctuations are small compared to the background flow. For  $\alpha = 0.0027$ , the gravity intrusion regime is approached, so the velocity fluctuations across the domain are no longer small. This means that the assumptions for equation (6.26) do not hold and the velocity fluctuations do not match the predicted profile.

### 6.5.2 Strong background flow: $u_B \gtrsim 1$

The two dimensional concentration profiles for three values of  $Ra$  are shown in Figure 6.10 for  $u_B = 1.0$ . Two different regimes develop, as predicted by the regime diagram in Figure 6.3, because diffusion becomes dominant before the buoyancy driven dispersion has time to develop. Figure 6.10a shows a gravity intrusion, Figure 6.10b shows a transitional regime and Figure 6.10c shows a diffusive regime.

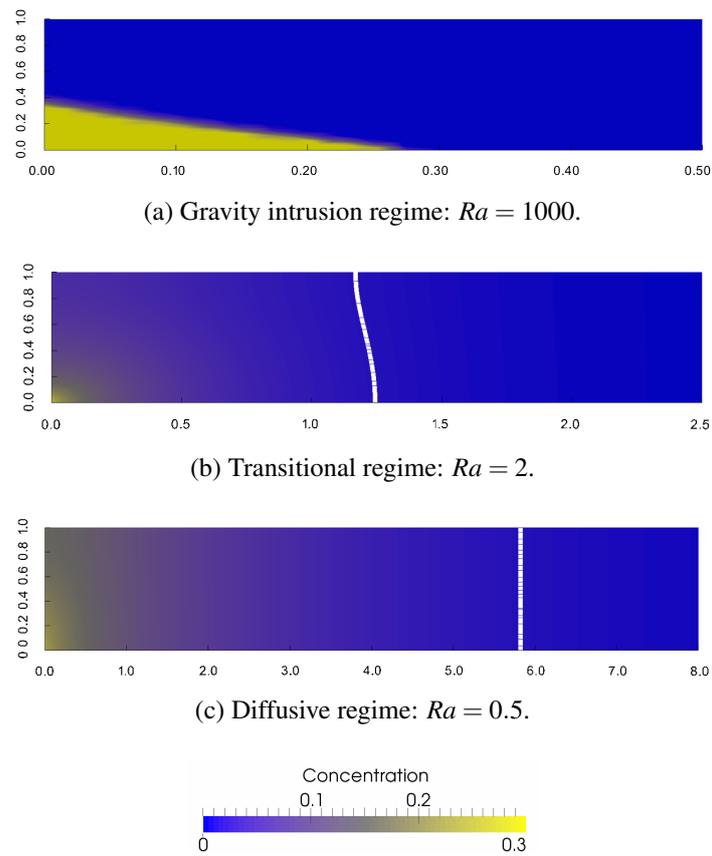


Figure 6.10: Concentration fields for  $u_B = 1.0$  and different values of  $Ra$ . The white contour in (b) and (c) is the  $c = 0.01$  contour. The domains have been truncated at different lengths to best represent each regime. Note that the maximum concentration for the colour bar is set equal to 0.3, which is greater than the highest concentration for panels (b) and (c) [121].

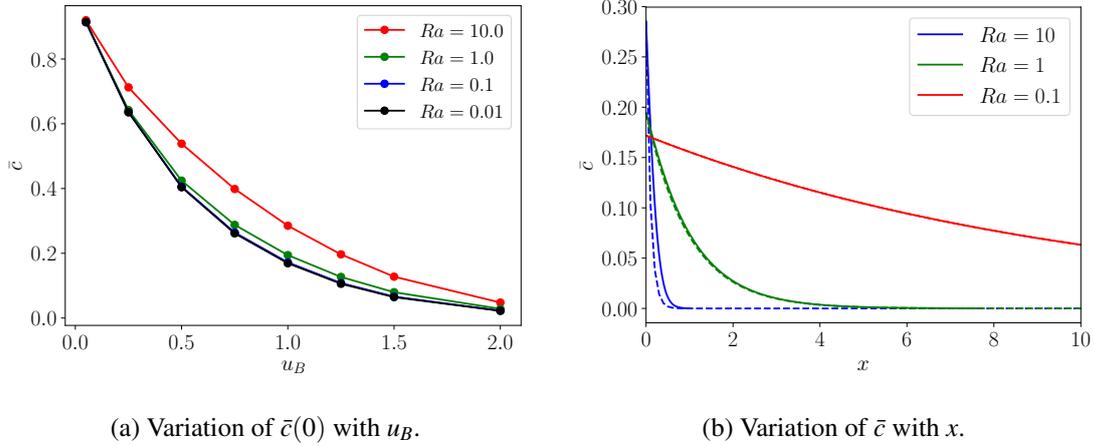


Figure 6.11: Concentration profiles for  $u_B \gtrsim 1$ . Fig (a) shows the vertically averaged concentration computed from the two dimensional numerical model for various values of  $Ra$  and  $u_B$  at  $x = 0$ . Fig (b) shows the computed and analytical variation of  $\bar{c}$  with  $x$  for different values of  $Ra$  when  $u_B = 1.0$ . The solid line represents the two dimensional numerical solution and the dashed line represents the diffusion limit if using the numerically computed  $\bar{c}$  as the boundary condition at  $x = 0$  [121].

The vertically averaged concentration profiles at  $x = 0$  are shown in Figure 6.11a for different values of the background flow,  $u_B$  and  $Ra$ . For all values of  $Ra$ ,  $\bar{c}$  at  $x = 0$  decreases as  $u_B$  increases. This is because there is a progressively stronger background flow, with which the buoyancy driven flow has to compete. This reduces the amount of saturated water that enters the domain.

For small values of  $Ra$  in the presence of a strong background flow, the diffusive  $\bar{c}$  from equation (6.48) can be rescaled to

$$\bar{c}(x) = \bar{c}(0)e^{-Ra u_B x}, \quad (6.58)$$

where  $\bar{c}(0)$  is the vertically averaged concentration at  $x = 0$ . Figure 6.11b shows the variation in the averaged concentration profiles for different values of  $Ra$ . For  $Ra = 0.1$ , the flow is diffusive because the numerical solution matches the theoretical curve from equation (6.58). As the value of  $Ra$  increases, the numerical solution diverges from the theoretical prediction as the gravity intrusion regime is entered.

## 6.6 Conclusions

In this chapter, analytical and numerical models are developed for the long-term buoyancy driven transport of aqueous CO<sub>2</sub> along horizontal aquifers in the presence of a background flow. The dispersion dominated, diffusion dominated and gravity intrusions regimes can be identified in the two dimensional numerical models for the expected values of  $Ra$  and  $u_B$ , which validates both models.

This research identifies that  $\alpha$  is a function of  $u_B$ , so the background hydrological flow has a significant impact on the long term dissolution of CO<sub>2</sub>. Therefore, new estimates can be made for

behaviour of the buoyancy driven mechanisms. If typical parameters for aquifers are considered, it is expected that the flow dynamics are controlled by a balance of buoyancy driven shear dispersion and the background hydrological flow. For a typical aquifer with height 20 – 30m,  $Ra = 10^3 - 10^4$  and a non-dimensional background velocity of 0.01 – 1.0, the length scale for dispersion will be 100 – 400m and will take  $10^5 - 10^6$  years [121].

Real aquifer geometries and parameters could be considered to develop this model further. This would enable scientists to approximate the volumes of CO<sub>2</sub> that could be stored in an actual aquifer so that better informed decisions could be made about the viability of CO<sub>2</sub> sequestration.



## Chapter 7

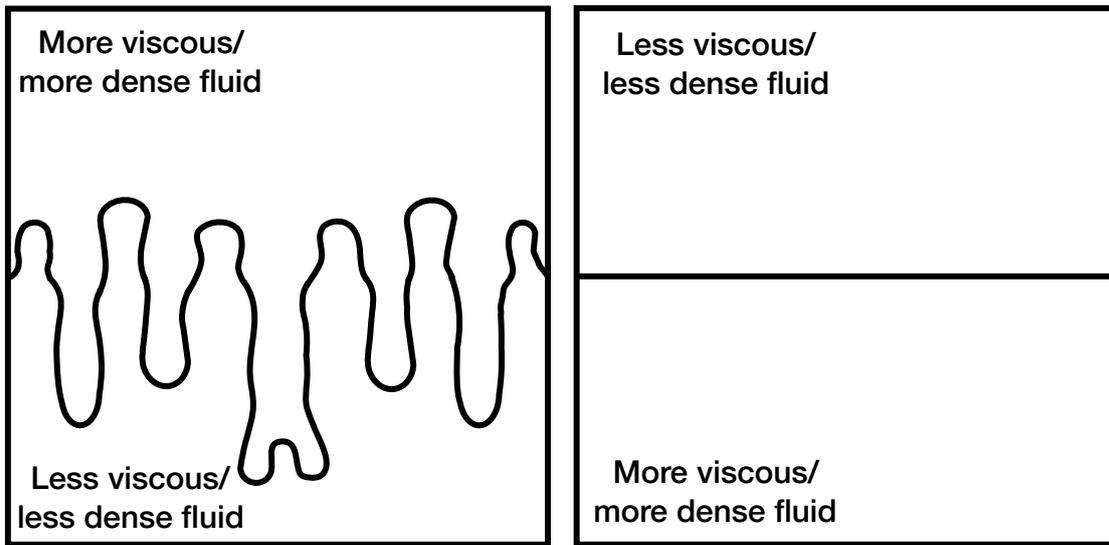
# Quantifying uncertainty in buoyancy driven flows

Uncertainty in buoyancy driven flows through porous media needs to be quantified to understand flow behaviour fully. As discussed in Chapter 1, epistemic uncertainty can be added to hydrogeological models through poor measurement of the rock permeability over the length of an aquifer [133]. Variations in weather conditions can also add aleatoric uncertainty to the model. Different rainfall levels result in varying groundwater velocity and aquifer saturation levels. This means that the real system behaviour can be very different to that described by a deterministic system.

This chapter focuses on one particular type of buoyancy driven flow, where a layer of dense fluid lies on top of a layer of less dense fluid. This unstable scenario results in a phenomena known as fingering. The literature about modelling fingering in buoyancy driven flows is presented in Section 7.1 and the numerical model of interest is described in Section 7.2. Then in Section 7.3, attempts are made to quantify uncertainty in advection dominated buoyancy driven fingering. However, difficulties are found with resolving the fingering phenomena on different fidelity meshes. Therefore, the model is further simplified in Section 7.4 by considering diffusion dominated flows, where fingering is less prominent. Conclusions are then presented in Section 7.5.

### 7.1 Buoyancy driven fingering

From a one dimensional stability analysis and experiments, where liquid sugar was run through water saturated granular bone charcoal, Hill [65] identified that ‘finger-like’ structures can grow in porous media flows under certain conditions. His research established that there are two mechanisms that drive the growth of fingers: the difference between the viscosities of the fluid and the difference in density between the fluids. The flow is unstable if the top fluid is more dense and more viscous than the bottom flow, or if the combination of differences between the viscosity and density of the fluids across the interface is unstable. This instability causes fingers of the more dense fluid to grow vertically into the less dense fluid, see Figure 7.1a. It should be noted that the problem is stable and



(a) More viscous fluid on top.

(b) Less viscous fluid on top.

Figure 7.1: Diagram indicating fingering behaviour in porous media flows.

no fingers grow if these properties are inverted and the more viscous dense fluid is underneath, see Figure 7.1b.

Chuoque et al. [22] and Saffman and Taylor [104] both presented similar in-depth one dimensional linear stability analyses using experiments in Hele-Shaw cells, which gave more detail than Hill [65]. Their new analysis found that the instabilities, and so resulting fingers, are dependent on the density and viscosity of the fluids, as identified by Hill, but also the surface tensions and relative velocities of the fluids and the permeability and length scale of the porous media. In addition to the stability analysis, Saffman and Taylor [104] also considered the shape and evolution of a single dominant finger.

Elder [37, 38] then modelled fingering in thermal flows to form the benchmark for density driven flows in porous media. He considered the effects of heating a small portion of fluid in a rectangular box and found that finger-like structures grow both experimentally, using a Hele Shaw cell, and numerically. Voss and Souza [128] then recast the problem to consider dense fluid mixing into a domain filled with less dense fluid. Again, this unstable scenario resulted in the growth of fingers in the domain.

The suitability of the Elder problem as a benchmark has since been questioned by van Reeuwijk et al. [125] and Diersch and Kolditz [35], who showed that the solution depends on the fidelity of the mesh, the numerical method used, and initial conditions applied. For a coarse representation of the domain, the concentration profile tends to match the profiles in Elder [37] and Voss and Souza [128], where three fingers form in the domain and at the centre there is a downwelling of fluid, see Figure 7.2a. However, a more refined representation results in a central upwelling with four fingers present [1, 35, 93, 125], see Figure 7.2b. Most research also only considered half of the domain, since

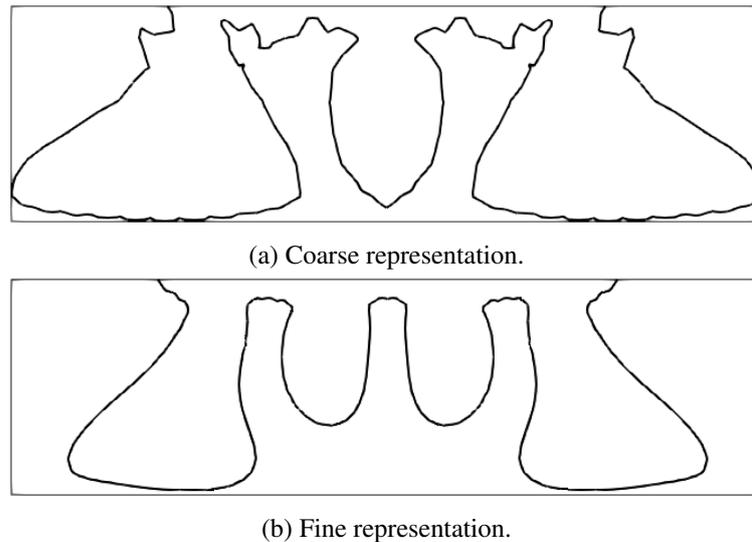


Figure 7.2: Elder benchmark contours indicating 20% of the maximum concentration computed using FEniCS.

symmetrical flow behaviour was assumed. This may be a poor assumption because the solution does not necessarily remain symmetric on non-symmetric meshes.

Elder [37] used his benchmark to propose that the growth of the fingers is accelerated by more less-buoyant fluid, from the dense layer, being drawn into the finger. Wooding [132] developed this work by identifying that if the density gradient at the tips of the fingers exceeds the stabilising mechanisms as the fingers grow, this results in the fingers bifurcating.

The research discussed so far considers homogeneous and deterministic porous media, but Kueper and Frind [75] hypothesised that heterogeneous permeability fields would result in a better model of the rock. They further suggested that this makes the previous stability analysis by Chuoke et al. [22] and Saffman and Taylor [104] invalid because fingering will simply be initiated in the most permeable regions, which exhibit the easiest pathway for the advancing front. Figure 7.3a shows that when the permeability is homogeneous, the fingering is fairly constant across the interface between the two layers of fluid; whereas, Figure 7.3b shows that when the permeability is heterogeneous, the fingering initiates along a region of the interface corresponding to the area of highest permeability.

With the development of efficient high speed computation, Kueper and Frind [75] and Leclerc and Neale [80] advocated modelling the uncertain nature of these flows using the MC method. Both papers simplified the numerical models because it is prohibitively expensive to quantify uncertainty in complex PDE governed systems using the traditional MC method. Uncertainty is now beginning to be quantified in porous media flows using new techniques that are computationally cheaper than the MC method. Cliffe et al. [26] used the MLMC method to assess the safety of geological reservoirs from radioactive waste and Crevillén-García et al. [28] used a Gaussian process emulation method to model CO<sub>2</sub> storage.

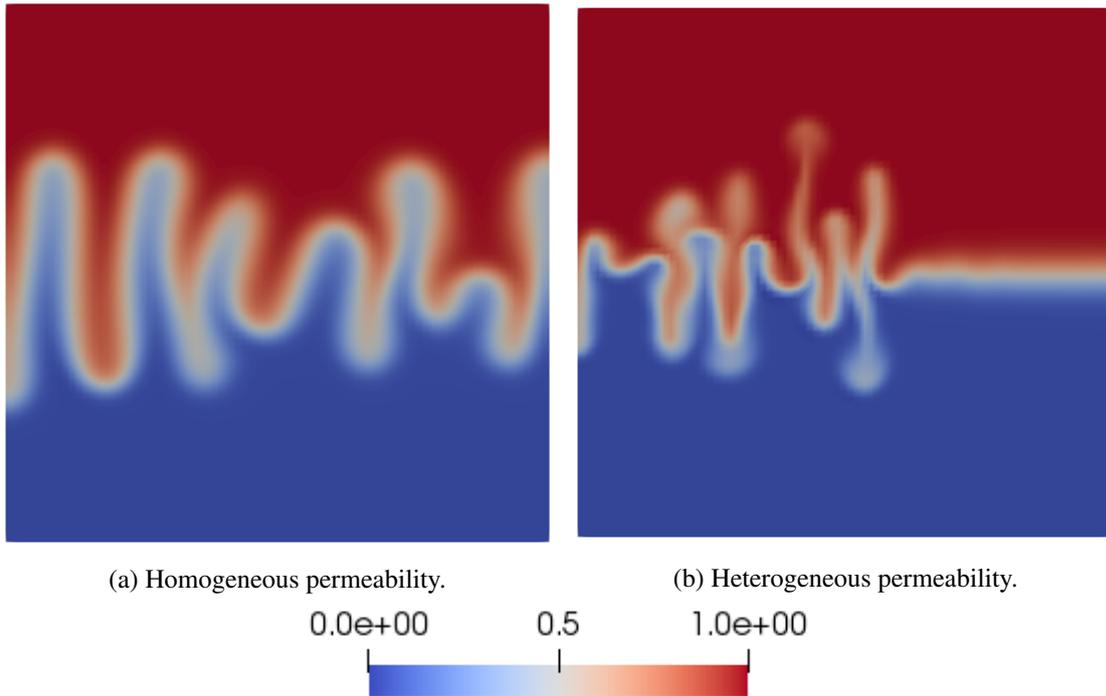


Figure 7.3: Concentration profiles of fingering simulations indicating difference in flow behaviour due to varying permeability fields, computed with FEniCS.

Other than the original work in Kueper and Frind [75] very little, or no, research into the impact of uncertain permeability on the fingering behaviour has been published. Therefore, the impact of heterogeneous permeability fields on buoyancy driven fingering is investigated in this chapter. The MLMC method is applied to simplified buoyancy driven flow examples to indicate the feasibility of using this method to quantify uncertainty. The viscosity of the two fluids is assumed constant in the analysis and only the density of the fluids vary.

## 7.2 Model system

Fingering in buoyancy driven flows is governed by the same equations that govern the long term behaviour of  $\text{CO}_2$  saturated water flows in aquifers. Therefore, the impact of uncertainty in buoyancy driven fingering is investigated by solving the coupled Darcy equations and scalar transport equation using FEniCS [4] and podS [122] for a simplified unit square domain, Figure 7.4. The same equations and FE formulations can be used as Chapter 6. Using notation consistent with Section 6.4, the

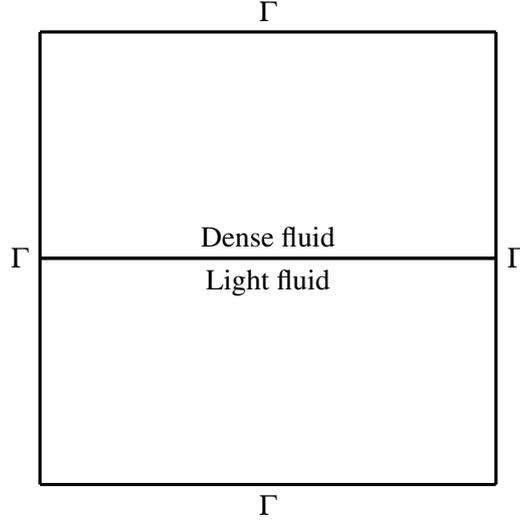


Figure 7.4: Domain for pollution scenario.

boundary and initial conditions are

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma \times I, \quad (7.1a)$$

$$\left(-\frac{1}{Ra} \nabla c + c\mathbf{u}\right) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma \times I, \quad (7.1b)$$

$$c(x, y, 0) = c_0(y) \quad \text{on } D, \quad (7.1c)$$

where

$$c_0(y) = \begin{cases} 1 & \text{if } 0 \leq y < 0.5 \\ 0 & \text{if } 0.5 \leq y \leq 1. \end{cases} \quad (7.2)$$

These conditions represent no fluid or concentration flow over the boundary and an unstable initial condition, from which fingers should grow. Again the equations are coupled by density driven forcing, which is described in equation (6.2).

A separable analytical exponential KL expansion is used to represent the non-dimensional permeability field,  $\kappa$ , as described in Section 2.6. This is computed by taking the exponential of the analytical KL expansion given in equation (2.20) with mean and standard deviation of 0.0 and 1.0 respectively. The expansion is truncated after 100 eigenpairs. The length scale,  $\lambda = 0.25$  ensures that the uncertain permeability field can be represented on all the meshes considered. The size of time step is chosen such that  $dt = 0.005$ . The code for this solver is published as Unwin [118].

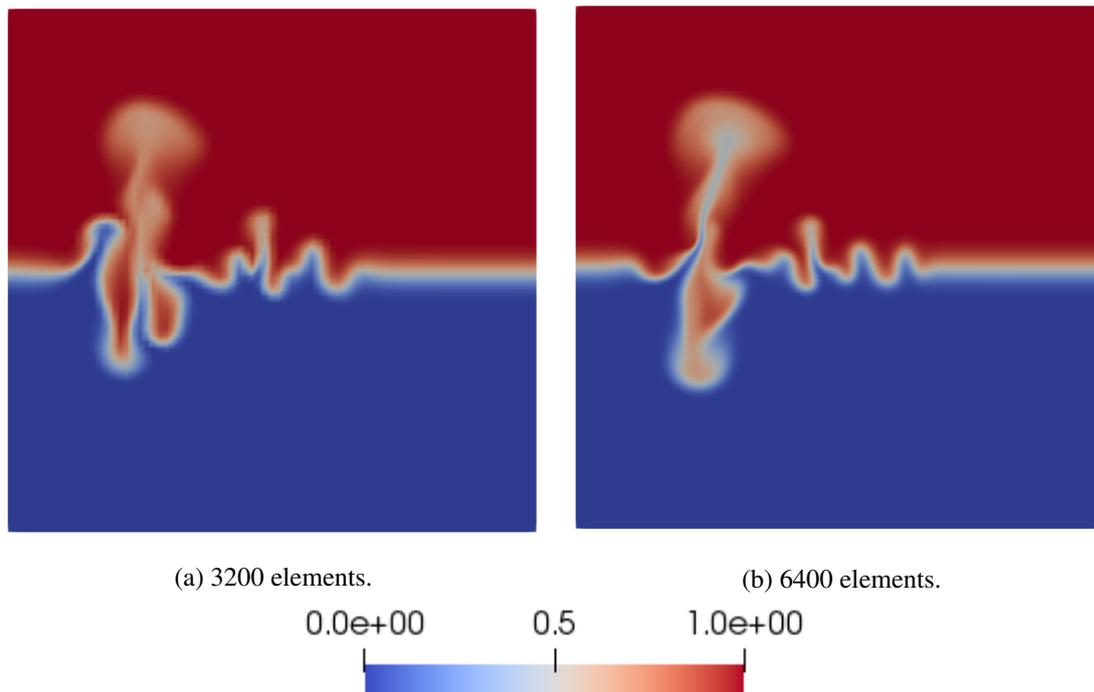


Figure 7.5: Concentration profiles for one realisation of a random field with different fidelity meshes. These snapshots are for time step 50 and  $Ra = 1000$ .

### 7.3 Challenges of quantifying uncertainty in density driven fingering

The integral of the concentration along the  $x$ -axis at  $y = 0.4$  after 50 time steps is chosen as the QoI because it provides a measure of how much dense fluid has been transported into the less dense fluid. An advection dominated flow is chosen initially, with a Rayleigh number of  $Ra = 1000$  from equation (6.11), which is the same value used to compute the flows in Figure 7.3.

The numerical error in the expectation of the QoI must converge as the mesh is refined to be able to use the MLMC method to quantify uncertainty in these flows. However, Figure 7.5 shows that as the mesh is refined, the concentration profiles do not converge to the same solution. This is similar to the reasons why the Elder problem, identified in Section 7.1, is unsuitable as a benchmark for buoyancy driven flows. This lack of convergence is shown again in Figure 7.6, where the finite element error  $\alpha$  and variance  $\beta$  diverge as the mesh is refined, for ten realisations of each element size  $h$ .

Alternative QoIs to that identified above have also been investigated; neither the integral of the concentration for  $y < 0.5$ , nor the time taken for the concentrated fluid to reach the bottom of the domain, converged. Again, this is expected because of the qualitative differences in the solution for each mesh.

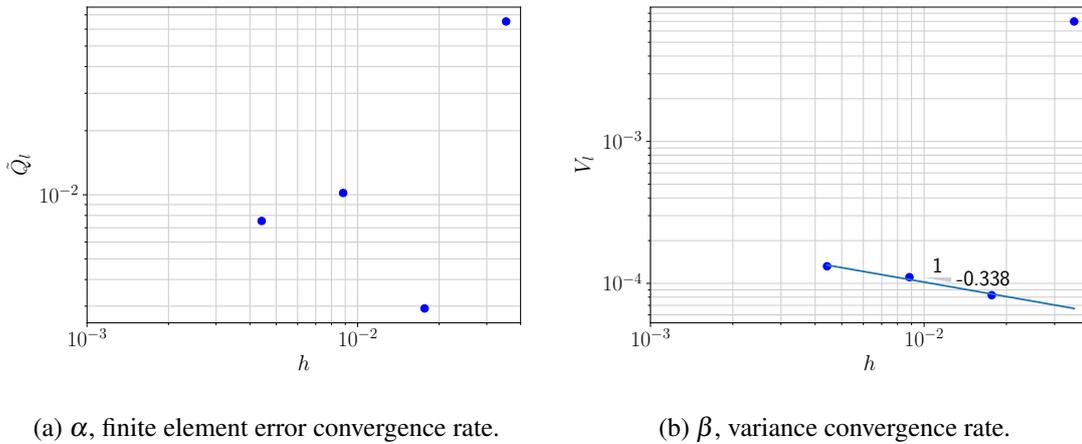


Figure 7.6: Diverging convergence rates for the integral of the concentration at  $y = 0.4$  for  $Ra = 1000$  and time step 50.

## 7.4 Impact of uncertainty on diffusion dominated fingering

If the value of  $Ra$  is reduced, the flow regime transitions from being advection dominated to being diffusion dominated. Diffusive flows are easier to model numerically because diffusion adds natural smoothing to the solution, so sharp fronts do not present. The concentration profiles for models with  $Ra = 100$  do converge as the mesh is refined, so the impact of uncertainty can be quantified using the MLMC method.

As shown in Figure 7.7a, fingers do not grow in these regime. However, this problem is still of interest because the concentration profile is non uniform across the domain, resulting from the uncertain permeability field. This can be seen in Figure 7.7b, which depicts the concentration profile along  $y = 0.4$  at time step 50.

In this example, 3200 elements are required for the coarsest mesh to ensure that the QoI converges for this diffusion dominated flow. Similar to Chapter 4, the number of elements increases two-fold in each dimension, which gives a total of a four times increase between each level for this scenario. Again, the number of elements used for each MC computation is chosen to ensure  $e_h < \epsilon_h$ . The 50th time step is chosen as the time step of interest because the impact of uncertainty can be detected, but each individual simulation, even for the finest resolution, can be solved within three hours.

The probability distributions of the QoI are compared in Figure 7.8 for the MC and MLMC methods. There is good agreement between the expectation of this QoI for the MC and MLMC methods, but the variances are different. This is expected because the tolerance of the MC method is  $4.63 \times 10^{-3}$ , which is only one order of magnitude smaller than the mean. If the tolerance is reduced, convergence between the variance of the two methods is expected. Tighter tolerances could not be investigated for the MC method because these required more RAM than was available on the computer. The expectation and variance of this QoI do converge for the MLMC method as the

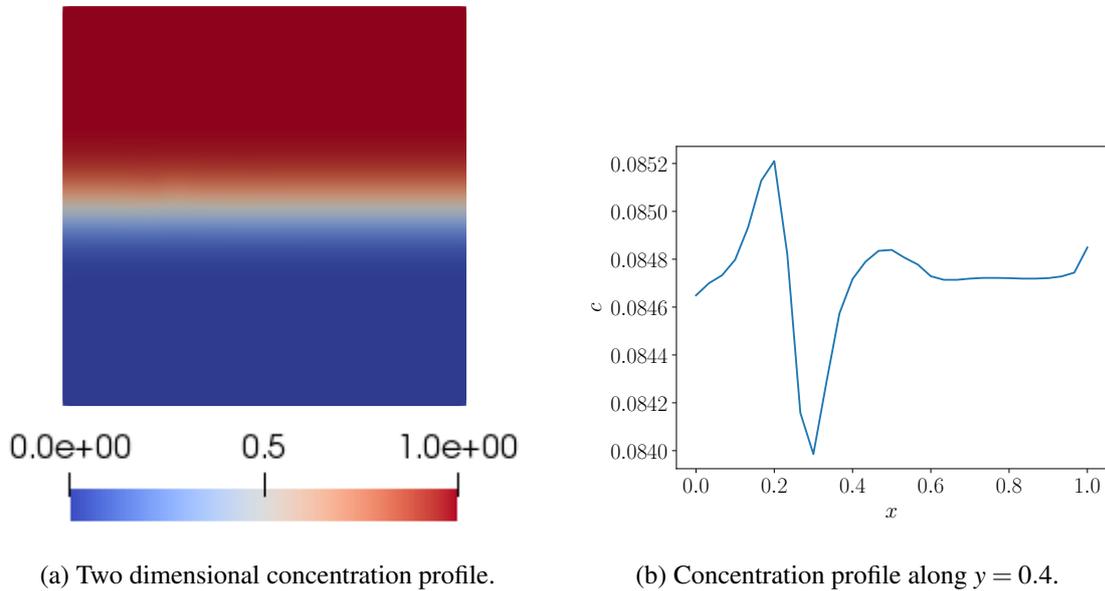


Figure 7.7: Concentration profile for one realisation of a random field for  $Ra = 100$  at time step 50.

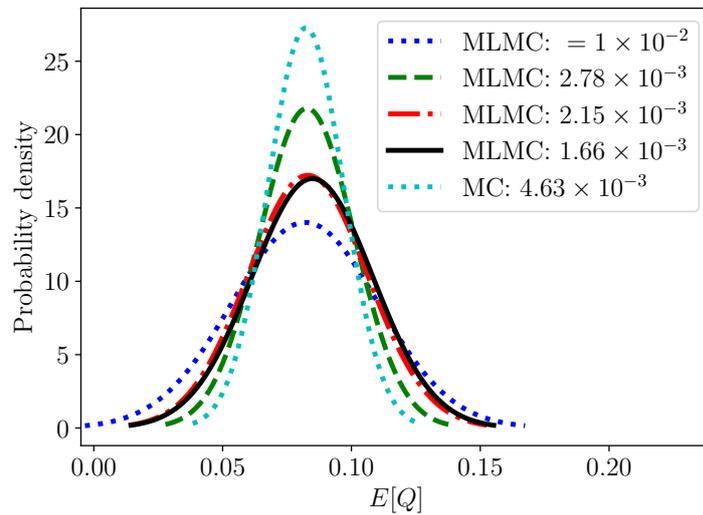


Figure 7.8: Comparison of the MC and MLMC probability distribution for the integral of concentration at  $y = 0.4$  and time step 50 for various tolerances.

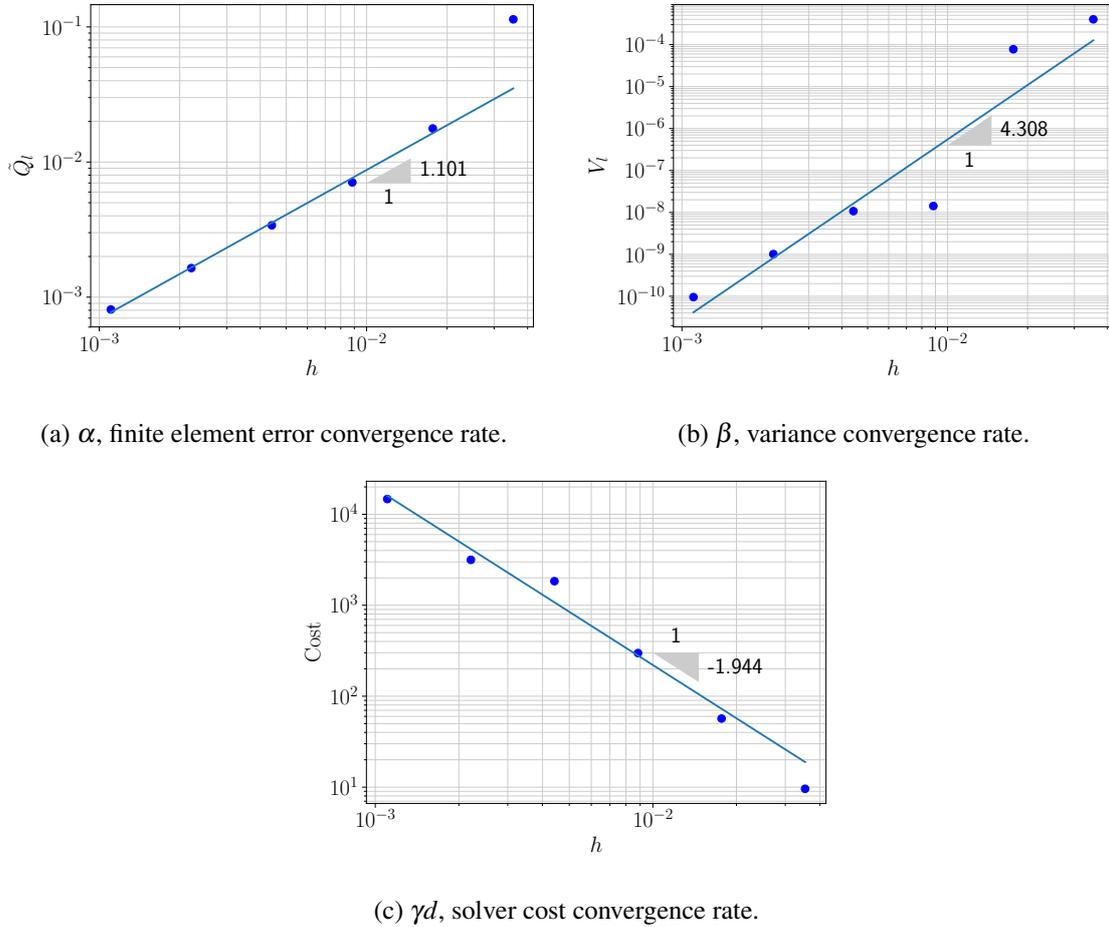


Figure 7.9: MLMC Convergence rates for the integral of concentration at  $y = 0.4$  and time step 50. Fig (a) shows the finite element convergence rate  $\alpha = 1.101$ , Fig (b) shows the convergence rate of the variance  $\beta = 4.308$  and Fig (c) shows the convergence rate of the cost of solving each different level  $\gamma_d = 1.944$  or  $\gamma = 0.972$ , for  $\varepsilon = 1.29 \times 10^{-3}$ .

tolerance of the method is decreased. This is shown by the good agreement between the two tightest MLMC tolerances plotted.

The convergence parameters for this simulation are shown in Figure 7.9 using a sparse LU solver for  $\varepsilon = 1.29 \times 10^{-3}$ . The FE error for the integral of the concentration for this problem converges slowly as  $\alpha = 1.1$ . This suggests that prohibitively fine meshes may be required to satisfy the numerical error for tight tolerances. The value of the solver complexity,  $\gamma \approx 1$ , is also lower for this coupled problem than what is predicted for the sparse LU solver for the individual two dimensional Darcy and scalar transport solves,  $\gamma = 3/2$ . The optimal complexity of the MLMC method,  $O(\varepsilon^{-2})$ , is predicted for this problem since, from Figure 7.9,  $\beta > \gamma_d$ . In contrast, the complexity of the MC method is predicted to be  $O(\varepsilon^{-2-\gamma_d/\alpha})$  or  $O(\varepsilon^{-3.76})$  for this problem.

Again, these simulation complexities have been computed by timing how long the MLMC and MC simulations take to converge to various tolerances on a workstation with two Intel Xeon E5-2670

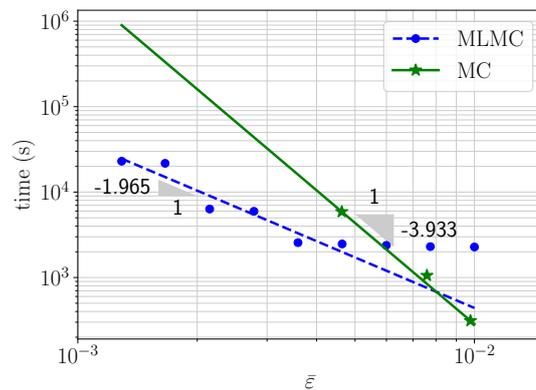


Figure 7.10: Comparison of time taken for the MC and MLMC simulations for the integral of the concentration at  $y = 0.4$  on ten cores.

Table 7.1: Number of samples taken at each level for MLMC simulations of the integral of the concentration at  $y = 0.4$ .

$\epsilon$	0	1	2	3	4	5
$1.00 \times 10^{-2}$	36	12	10	10	-	-
$4.64 \times 10^{-3}$	96	33	10	10	-	-
$3.59 \times 10^{-3}$	134	45	10	10	-	-
$2.78 \times 10^{-3}$	224	74	10	10	1	-
$2.15 \times 10^{-3}$	373	123	10	10	1	-
$1.66 \times 10^{-3}$	626	207	10	10	1	1
$1.29 \times 10^{-3}$	1036	342	10	10	1	1

processors, 12 cores on each processor, a clock speed of 2.60 GHz and 128 GiB RAM. The complexity of the MLMC method is very accurately predicted in Figure 7.10, but the simulation complexity of the MC method is slightly greater than predicted because it was only possible to compute a few data points due to memory restrictions on the workstation used. The MLMC data points are clustered in three linear groups, which can be seen in Table 7.1 to correspond to groups of simulations with the same number of levels. This suggests that adding a new level is far more computationally expensive than sampling the current levels more.

## 7.5 Conclusions

In this chapter, the impact of uncertain permeability fields has been investigated for buoyancy driven flows through porous media using the MLMC method. The MLMC method could not be used to quantify uncertainty in advection dominated flows because the fingering pattern of the fluid differed as the mesh was refined, and the computer memory was insufficient to consider finer meshes. This

means that the QoIs did not converge and so the required number of MLMC levels and samples kept growing.

In contrast, when the flow regime was dominated by diffusion, the natural smoothing of the flow ensured that the QoIs converged as the mesh was refined and so the MLMC method could be used to quantify uncertainty. This method is shown in Section 7.4 to be a good alternative to the traditional MC method for these flows because the expectations of the QoI are similar for the error tolerances that would currently fit into memory. The complexity of the MLMC method is also found to be approximately half that of the MC method. This means that the expectation of the QoI is predicted to be computed at least two orders of magnitude faster than using the MC method for  $\varepsilon < 10^{-3}$ .

However, further research into quantifying uncertainty in unstable buoyancy driven flows is still necessary to usefully apply this technique to pollution management system design. Two areas for more research are therefore discussed below. First, it is only possible to use the MLMC method to investigate diffusion dominated flows. Scheichl [105] suggested that it may be possible to use other multilevel techniques, such as multifidelity modelling, to quantify uncertainty in advection dominated flows by artificially adding diffusion to the coarse models. This will help to ensure that the coarse model represents the flow found in the finer models instead of the fingers growing in a different pattern. Second, analytical KL expansions, which are computed using a separable covariance kernel, are not especially representative of the permeability fields found in rocks and the domains that can be investigated are limited. If Matérn fields [83] are used instead, the cost of realising the random field is comparable, but the domains of interest are no longer limited to lines, rectangles or cuboids.



# Chapter 8

## Conclusions

In this thesis, the impact of uncertainty has been quantified using the MLMC method for two prototypical engineering applications: elastic vibrations, and buoyancy driven flows through porous media. FE models of the vibration and porous media systems have been developed and validated to ensure that the underlying physics of the system have been captured. Uncertainty has been represented by methods familiar to the relevant communities: point masses in the structural vibration models, as in Choi et al. [21]; and KL expansions in the porous media models, as in Teckentrup et al. [116]. The impact of uncertainty has then been quantified for these models using the podS library parallelisation of the Monte Carlo and multilevel Monte Carlo algorithms.

This research advances understanding in two main areas. First, in Chapters 4 and 7 the MLMC method is shown to be a useful alternative to the traditional MC method for quantifying the impact of uncertainty on structural vibration and porous media applications. The expectation and variance of the QoIs, computed using the MLMC method, also agree with the values found using the MC method and the simulated convergence rates agree with the theoretical results. However, the MLMC simulations take orders of magnitude less time to compute because the complexity of the MLMC method is always lower than the MC method.

The second advancement is that the simulation time, for calculating uncertainty using the MLMC method for different tolerances, is significantly accelerated by computing multiple samples in parallel, see Chapter 3. The podS library, which has been developed specifically for this research, provides an open source implementation of the MC and MLMC algorithms. This library schedules multiple parallel samples of the system on different cores and results in approximately linear reductions in simulation time. This means that the simulation time is approximately quartered when four problems are solved in parallel on four cores instead of one after the other.

Specific conclusions, drawn from the two applications, are presented in Sections 8.1 and 8.2, and areas where the research in this thesis can be further developed are identified in Section 8.3.

## 8.1 Structural vibration under uncertainty

In Chapter 4, the impact of uncertain mass density has been effectively quantified using the MLMC method for structural vibration applications in orders of magnitude less time than the MC method. For example, it took under 40 minutes to quantify uncertainty in the wave equation using the MLMC method for a relative tolerance of  $10^{-4}$ , but over one day with the MC method, using eight cores on a workstation. The variances of these structural examples converge faster than the complexities of the solvers, which means that most of the computational effort is used to sample the coarsest meshes instead of the fine ones. This means that quantifying the impact of uncertainty is now tractable for three dimensional domains. In addition to the first eigenvalues, the MLMC method has also been successfully used to quantify the impact of uncertainty on the 25th eigenvalue of the wave equation and the vibrational energy of a Kirchhoff–Love plate.

## 8.2 Buoyancy driven flows under uncertainty

Uncertain permeability fields have also been successfully quantified in Chapter 7 for two dimensional diffusion dominated buoyancy driven flows through porous media. To ensure no spurious flows are added to the system, a mixed FE formulation has been used to solve the Darcy equation, which can represent the solution to the PDE point-wise.

The FE models that were used for this application have been verified against a one dimensional analytical model of CO<sub>2</sub> sequestration as a separate investigation. This research considers the long term behaviour of CO<sub>2</sub> saturated water in the presence of a background flow, which has not been considered in previous research. This enables better estimates to be found for the volume of CO<sub>2</sub> that could be stored in an aquifer. The two dimensional simulations and the one dimensional analytical model agree well in the parameter regimes found in an aquifer, thus validating the formulations whilst considering a novel problem.

Uncertainty has then been introduced to investigate the impact of random permeability fields on buoyancy driven flows. A similar expectation and variance of a QoI have been found using the MC and MLMC methods for the tolerances computed. However, the complexity of the MLMC method was half that of the MC method. This means that the MLMC method is predicted to converge to a tolerance of less than  $10^{-3}$  two orders of magnitude faster than the MC method.

## 8.3 Further research

The research presented in this thesis explores only a few potential avenues for quantifying uncertainty in engineering systems. Areas where this research could be extended include: the MLMC implementation, application specific future research, and the wider applicability of the MLMC method.

### 8.3.1 MLMC implementation

Two methods that could be used to reduce further the time taken to quantify uncertainty with the MLMC method are as follows. First, the computation costs could be reduced further if instead of assuming that the error is split equally between the numerical and the sampling error, the error is weighted towards the error that is easier to reduce. For example, if reducing the numerical error is costly, the same total error could be achieved by further reducing the sampling error and sampling the coarser meshes more. Second, the complexity of the MLMC method could be reduced by using quasi-random sampling, as introduced in Section 1.1.1, to sample the probability space and eliminate the clustering of samples.

In addition to improving the convergence of the MLMC method, further improvements to the podS library MLMC implementation could be made. First, methods for approximating the cost parameter  $\gamma$  could be improved. The MLMC algorithm is formulated assuming that  $\gamma$  is the solver cost convergence parameter, but the current implementation uses the time taken to compute the whole sample to approximate  $\gamma$  and assumes that the solver cost is the dominant contribution. Therefore, a more accurate value could be found by timing individual samples or, if known, using an *a priori* complexity value. Second, the scheduling itself could be improved. If the time for individual solves is measured, different level samples could be solved in parallel. This may relieve some memory restraint problems that occur from solving multiple large problems in parallel. One core could then be used to schedule the samples adaptively and the other cores could communicate with it to request jobs.

### 8.3.2 Application specific developments

Although it is possible to use the MLMC method to quantify uncertainty successfully in structural vibration and porous media problems, the computational cost still constrains the achievable accuracy in the QoI. Tighter tolerances and more interesting domains could be considered with more memory and compute cores, which could be obtained by using different computing facilities such as supercomputers or cloud computing facilities.

The cost of solving the structural vibration problems could also be reduced by improving the efficiency of the eigensolver. Both Krylov-Schur and Jacobi-Davidson methods do not converge quickly for FE formulations with point masses. Therefore, improved FE convergence estimates could be obtained by investigating other eigensolvers or using an alternative method to representing uncertainty in the system, such as Matérn fields [83].

The cost of investigating uncertainty in diffusion dominated buoyancy driven flows could also be reduced by using more efficient solvers such as multigrid, which would interface well with the MLMC method due to the hierarchical nature of the solver. This would enable an exploration of the computations limit of feasible values of the Rayleigh number that could be investigated using the MLMC method.

New multilevel methods, introduced in Section 1.1.3, could be investigated for these two applications. The multi-index Monte Carlo method [55] may improve the accuracy of the expectation of

the energy density of the Kirchhoff–Love plate, since this method enables two directions of refinements. As the mesh fidelity is increased, the number of eigenpairs that could be accurately computed increases, so more eigenpairs could be used in the energy calculation for the finer meshes than the coarser ones. It may also be possible to use multifidelity modelling [95], instead of the MLMC method, to quantify uncertainty in advection driven porous media flows. Scheichl [105] suggested that coarse fidelity models could be developed that contain artificial diffusion, which may stop the fingering patterns significantly changing as the fidelity of the model is refined.

### **8.3.3 Wider applicability of the MLMC method**

Finally, the MLMC method may also be able to be used to quantify uncertainty in wider engineering problems, such as optimisation and control of engineering systems. With the order of magnitude reduction in computational time that have been achieved in this thesis, it may be possible to use multilevel methods to reduce the cost of optimisation and control under uncertainty. Instead of using traditional methods to quantify uncertainty for each set of input parameters, great computational savings can be made if new multilevel methods are used. Research into combining these new multilevel methods with optimisation techniques is just beginning to emerge for simple test cases, such as Peherstorfer et al. [96].

# References

- [1] P. Ackerer and A. Younes. Efficient approximations for the simulation of density driven flow in porous media. *Advances in Water Resources*, 31(1):15–27, 2008. ISSN 0309-1708. doi: <http://dx.doi.org/10.1016/j.advwatres.2007.06.001>.
- [2] S. Adhikari, M. I. Friswell, K. Lonkar, and A. Sarkar. Experimental case studies for uncertainty quantification in structural dynamics. *Probabilistic Engineering Mechanics*, 24(4):473 – 492, 2009. ISSN 0266-8920. doi: <http://dx.doi.org/10.1016/j.probenmech.2009.01.005>.
- [3] X. Albets-Chico and S. Kassinos. A consistent velocity approximation for variable-density flow and transport in porous media. *Journal of Hydrology*, 507:33 – 51, 2013. ISSN 0022-1694. doi: <http://dx.doi.org/10.1016/j.jhydrol.2013.10.009>.
- [4] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015. doi: 10.11588/ans.2015.100.20553.
- [5] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference, AFIPS '67 (Spring)*, pages 483–485, New York, NY, USA, 1967. ACM. doi: 10.1145/1465482.1465560.
- [6] 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. doi: <http://dx.doi.org/10.1137/S0036142902418680>.
- [7] 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. doi: <https://doi.org/10.1137/050645142>.
- [8] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, 2010. doi: 10.1137/100786356.
- [9] S. Bachu and J. Adams. Sequestration of CO<sub>2</sub> in geological media in response to climate change: capacity of deep saline aquifers to sequester CO<sub>2</sub> in solution. *Energy Conversion and Management*, 44(20):3151 – 3175, 2003. ISSN 0196-8904. doi: [http://dx.doi.org/10.1016/S0196-8904\(03\)00101-8](http://dx.doi.org/10.1016/S0196-8904(03)00101-8).
- [10] S. Balay, W. D. Gropp, L. Curfman McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

- [11] A. Barth, C. Schwab, and N. Zollinger. Multi-level Monte Carlo finite element method for elliptic pdes with stochastic coefficients. *Numerische Mathematik*, 119(1):123–161, Sep 2011. ISSN 0945-3245. doi: 10.1007/s00211-011-0377-0.
- [12] R. Bladh, C. Pierre, M. P. Castanier, and M. J. Kruse. Dynamic response predictions for a mistuned industrial turbomachinery rotor using reduced-order modeling. *Journal of Engineering for Gas Turbines and Power*, 124(2):311–324, Mar. 2002. ISSN 0742-4795. doi: <http://dx.doi.org/10.1115/1.1447236>.
- [13] F. C. Boait, N. J. White, M. J. Bickle, R. A. Chadwick, J. A. Neufeld, and H. E. Huppert. Spatial and temporal evolution of injected CO<sub>2</sub> at the Sleipner Field, North Sea. *Journal of Geophysical Research: Solid Earth*, 117(B3), 2012. doi: <http://dx.doi.org/10.1029/2011JB008603>. B03309.
- [14] D. Boffi. Finite element approximation of eigenvalue problems. *Acta Numerica*, 19:1–120, 2010. doi: 10.1017/S0962492910000012.
- [15] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag New York, Inc., New York, NY, USA, 1991. ISBN 0-387-97582-9.
- [16] F. Brezzi, J. J. Douglas, and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985. ISSN 0029-599X. doi: 10.1007/BF01389710.
- [17] F. Brezzi, L. D. Marini, and E. Süli. Discontinuous Galerkin methods for first-order hyperbolic problems. *Mathematical Models and Methods in Applied Sciences*, 14(12):1893–1–903, 2004. doi: 10.1142/S0218202504003866.
- [18] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199 – 259, 1982. ISSN 0045-7825. doi: [http://dx.doi.org/10.1016/0045-7825\(82\)90071-8](http://dx.doi.org/10.1016/0045-7825(82)90071-8).
- [19] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998. doi: 10.1017/S0962492900002804.
- [20] J. Charrier, R. Scheichl, and A. L. Teckentrup. Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. *SIAM Journal on Numerical Analysis*, 51(1):322–352, 2013. doi: 10.1137/110853054.
- [21] W. Choi, J. Woodhouse, and R. S. Langley. Sound radiation from a vibrating plate with uncertainty. *Journal of Sound and Vibration*, 333(17):3966 – 3980, 2014. ISSN 0022-460X. URL <http://dx.doi.org/10.1016/j.jsv.2014.03.018>.
- [22] R. L. Chuoke, P. van Meurs, and C. van der Poel. The instability of slow, immiscible, viscous liquid-liquid displacements in permeable media. *Petroleum Transactions*, 216, Jan. 1959.
- [23] P. G. Ciarlet and P. A. Raviart. *Mathematical Aspects of Finite Elements in Partial Differential Equations: Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin–Madison, April 1 – 3, 1974*, chapter A mixed finite element method for the biharmonic equation. Elsevier Science, 1974.
- [24] A. Ciciello and R. S. Langley. The vibro-acoustic analysis of built-up systems using a hybrid method with parametric and non-parametric uncertainties. *Journal of Sound and Vibration*, 332(9):2165 – 2178, 2013. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2012.05.040>.

- [25] A. Cicirello and R. S. Langley. Efficient parametric uncertainty analysis within the hybrid finite element/statistical energy analysis method. *Journal of Sound and Vibration*, 333(6):1698 – 1717, 2014. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2013.10.040>.
- [26] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3, Aug 2011. ISSN 1433-0369. doi: 10.1007/s00791-011-0160-x.
- [27] V. Cotoni, R. Langley, and M. Kidner. Numerical and experimental validation of variance prediction in the statistical energy analysis of built-up systems. *Journal of Sound and Vibration*, 288(3):701 – 728, 2005. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2005.07.012>. Uncertainty in structural dynamics.
- [28] D. Crevillén-García, R. D. Wilkinson, A. A. Shah, and H. Power. Gaussian process modelling for uncertainty quantification in convectively-enhanced dissolution processes in porous media. *Advances in Water Resources*, 99(Supplement C):1 – 14, 2017. ISSN 0309-1708. doi: <https://doi.org/10.1016/j.advwatres.2016.11.006>.
- [29] G. De Silva, P. Ranjith, and M. Perera. Geochemical aspects of CO<sub>2</sub> sequestration in deep saline aquifers: A review. *Fuel*, 155:128 – 143, 2015. ISSN 0016-2361. doi: <http://dx.doi.org/10.1016/j.fuel.2015.03.045>.
- [30] M. K. Deb, I. M. Babuška, and J. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359 – 6372, 2001. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(01\)00237-7](https://doi.org/10.1016/S0045-7825(01)00237-7).
- [31] B. J. Deusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. L. Maitre. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2):698–719, 2004. doi: <https://doi.org/10.1137/S1064827503427741>.
- [32] A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105 – 112, 2009. ISSN 0167-4730. doi: <https://doi.org/10.1016/j.strusafe.2008.06.020>. Risk Acceptance and Risk Communication.
- [33] G. Detommaso, T. Dodwell, and R. Scheichl. Continuous level Monte Carlo and sample-adaptive model hierarchies. 2018. doi: arXiv:1802.07539.
- [34] S. Dey, T. Mukhopadhyay, H. H. Khodaparast, and S. Adhikari. Stochastic natural frequency of composite conical shells. *Acta Mechanica*, 226(8):2537–2553, 2015. ISSN 1619-6937. URL <http://dx.doi.org/10.1007/s00707-015-1316-4>.
- [35] H. Diersch and O. Kolditz. Variable-density flow and transport in porous media: approaches and challenges. *Advances in Water Resources*, 25(8–12):899 – 944, 2002. ISSN 0309-1708. doi: [http://dx.doi.org/10.1016/S0309-1708\(02\)00063-5](http://dx.doi.org/10.1016/S0309-1708(02)00063-5).
- [36] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Ltd, 2005. ISBN 9780470013823. URL <http://dx.doi.org/10.1002/0470013826.fmatter>.
- [37] J. W. Elder. Steady free convection in a porous medium heated from below. *Journal of Fluid Mechanics*, 27:29–48, 1 1967. ISSN 1469-7645. doi: <http://dx.doi.org/10.1017/S0022112067000023>.
- [38] J. W. Elder. The unstable thermal interface. *Journal of Fluid Mechanics*, 32(1):69–96, 1968. doi: 10.1017/S0022112068000595.

- [39] H. Emami-Meybodi, H. Hassanzadeh, and J. Ennis-King. CO<sub>2</sub> dissolution in the presence of background flow of deep saline aquifers. *Water Resources Research*, 51(4):2595–2615, 2015. ISSN 1944-7973. doi: <http://dx.doi.org/10.1002/2014WR016659>.
- [40] F. J. Fahy. Statistical energy analysis: a critical overview. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 346(1681): 431–447, 1994. ISSN 0962-8428. doi: 10.1098/rsta.1994.0027.
- [41] R. Field and M. Grigoriu. On the accuracy of the polynomial chaos approximation. *Probabilistic Engineering Mechanics*, 19(1):65 – 80, 2004. ISSN 0266-8920. doi: <https://doi.org/10.1016/j.pro bengmech.2003.11.017>.
- [42] H. Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80 (6):615–627, 1973. ISSN 00029890, 19300972. URL <http://www.jstor.org/stable/2319163>.
- [43] P. Frauenfelder, C. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Computer Methods in Applied Mechanics and Engineering*, 194(2):205 – 228, 2005. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2004.04.008>.
- [44] P. Frolkovič. Consistent velocity approximation for density driven flow and transport. In *Advanced Computational Methods in Engineering, Part 2: Contributed papers*, pages 603–611, 1998.
- [45] W. Gao. Natural frequency and mode shape analysis of structures with uncertainty. *Mechanical Systems and Signal Processing*, 21(1):24 – 39, 2007. ISSN 0888-3270. URL <http://dx.doi.org/10.1016/j.ymsp.2006.05.007>.
- [46] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973. doi: 10.1137/0710032.
- [47] R. Ghanem. Ingredients for a general purpose stochastic finite elements implementation. *Computer Methods in Applied Mechanics and Engineering*, 168(1):19 – 34, 1999. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(98\)00106-6](https://doi.org/10.1016/S0045-7825(98)00106-6).
- [48] R. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, 1991. ISBN 9783540974567.
- [49] R. G. Ghanem and P. D. Spanos. Spectral techniques for stochastic finite elements. *Archives of Computational Methods in Engineering*, 4(1):63–100, Mar 1997. ISSN 1886-1784. doi: <https://doi.org/10.1007/BF02818931>.
- [50] M. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo methods 2006*, pages 343–358, 2008.
- [51] M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 5 2015. ISSN 1474-0508. doi: 10.1017/S096249291500001X.
- [52] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.
- [53] B. Gmeiner, D. Drzisga, U. Rűde, R. Scheichl, and B. I. Wohlmuth. Scheduling massively parallel multigrid for multilevel Monte Carlo methods. *CoRR*, abs/1607.03252, 2016. URL <http://arxiv.org/abs/1607.03252>.
- [54] H. Hadfield. Computing uncertainty: scheduling multi-level Monte Carlo methods on a parallel computer. Master’s thesis, University of Cambridge, 2017.

- [55] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, Apr 2016. ISSN 0945-3245. doi: 10.1007/s00211-015-0734-5.
- [56] A. Hall. On an experimental determination of pi. *The Messenger of Mathematics*, 1873.
- [57] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen, 1964.
- [58] J. M. Hammersley and K. W. Morton. Poor man’s Monte Carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, 16(1):23–38, 1954. ISSN 00359246. URL <http://www.jstor.org/stable/2984008>.
- [59] H. Harbrecht, M. Peters, and M. Siebenmorgen. *On Multilevel Quadrature for Elliptic Stochastic Partial Differential Equations*, pages 161–179. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-31703-3. doi: [https://doi.org/10.1007/978-3-642-31703-3\\_8](https://doi.org/10.1007/978-3-642-31703-3_8).
- [60] S. Heinrich. Monte Carlo complexity of global solution of integral equations. *Journal of Complexity*, 14(2):151 – 175, 1998. ISSN 0885-064X. doi: <https://doi.org/10.1006/jcom.1998.0471>.
- [61] A. W. Herbert, C. P. Jackson, and D. A. Lever. Coupled groundwater flow and solute transport with fluid density strongly dependent upon concentration. *Water Resources Research*, 24(10): 1781–1795, 1988. ISSN 1944-7973. doi: <http://dx.doi.org/10.1029/WR024i010p01781>.
- [62] M. A. Hesse, H. A. Tchelepi, B. J. Cantwel, and F. M. Orr. Gravity currents in horizontal porous layers: transition from early to late self-similarity. *Journal of Fluid Mechanics*, 577: 363–383, 4 2007. ISSN 1469-7645. doi: <http://dx.doi.org/10.1017/S0022112007004685>.
- [63] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, Dec. 1952.
- [64] D. R. Hewitt, J. A. Neufeld, and J. R. Lister. High Rayleigh number convection in a three-dimensional porous medium. *Journal of Fluid Mechanics*, 748:879–895, 6 2014. ISSN 1469-7645. doi: 10.1017/jfm.2014.216.
- [65] S. Hill. Channeling in packed columns. *Chemical Engineering Science*, 1(6):247 – 253, 1952. ISSN 0009-2509. doi: [https://doi.org/10.1016/0009-2509\(52\)87017-4](https://doi.org/10.1016/0009-2509(52)87017-4).
- [66] D. E. Hocevar, M. R. Lightner, and T. N. Trick. A study of variance reduction techniques for estimating circuit yields. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2(3):180–192, July 1983. ISSN 0278-0070. doi: 10.1109/TCAD.1983.1270035.
- [67] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering. Dover Publications, 2000. ISBN 9780486411811.
- [68] H. E. Huppert and A. W. Woods. Gravity-driven flows in porous layers. *Journal of Fluid Mechanics*, 292:55–69, 6 1995. ISSN 1469-7645. doi: 10.1017/S0022112095001431.
- [69] IPCC. *Carbon dioxide capture and storage*. IPCC Geneva, Switzerland, 2005. URL [https://www.ipcc.ch/publications\\_and\\_data/\\_reports\\_carbon\\_dioxide.htm](https://www.ipcc.ch/publications_and_data/_reports_carbon_dioxide.htm). [https://www.ipcc.ch/publications\\_and\\_data/\\_reports\\_carbon\\_dioxide.htm](https://www.ipcc.ch/publications_and_data/_reports_carbon_dioxide.htm).
- [70] J. M. Johannes. An example of how the control variate method reduces noise in Monte Carlo experiments. *Communications in Statistics - Simulation and Computation*, 8(4):335–347, 1979. doi: 10.1080/03610917908812123.

- [71] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM Review*, 59(3):492–544, 2017. doi: <http://dx.doi.org/10.1137/15M1047696>.
- [72] C. Johnson. *Numerical solutions of partial differential equations by the finite element method*. Cambridge University Press, 1987.
- [73] A. Kebaier. Statistical Romberg extrapolation: A new variance reduction method and applications to option pricing. *The Annals of Applied Probability*, 15(4):2681–2705, 2005. ISSN 10505164. URL <http://www.jstor.org/stable/30038520>.
- [74] P. Knabner and P. Frolkovič. Consistent velocity approximation for finite volume or element discretizations of density driven flow in porous media. In *Computational methods in water research*, volume 1, 1996.
- [75] B. H. Kueper and E. O. Frind. An overview of immiscible fingering in porous media. *Journal of Contaminant Hydrology*, 2(2):95 – 110, 1988. ISSN 0169-7722. doi: [https://doi.org/10.1016/0169-7722\(88\)90001-0](https://doi.org/10.1016/0169-7722(88)90001-0).
- [76] R. J. Labeur. *Finite element modelling of transport and non-hydrostatic flow in environmental fluid mechanics*. PhD thesis, Delf University of Technology, 2009.
- [77] R. J. Labeur and G. N. Wells. Energy stable and momentum conserving hybrid finite element method for the incompressible Navier–Stokes equations. *SIAM J Sci Comput*, 34(2):A889–A913, 2012. doi: <http://dx.doi.org/10.1137/100818583>.
- [78] R. Langley and A. Brown. The ensemble statistics of the energy of a random system subjected to harmonic excitation. *Journal of Sound and Vibration*, 275(3):823 – 846, 2004. ISSN 0022-460X. doi: [https://doi.org/10.1016/S0022-460X\(03\)00780-6](https://doi.org/10.1016/S0022-460X(03)00780-6).
- [79] LAPACK. LAPACK website. <http://www.netlib.org/lapack/>, 2017.
- [80] D. F. Leclerc and G. H. Neale. Monte Carlo simulations of radial displacement of oil from a wetted porous medium: fractals, viscous fingering and invasion percolation. *Journal of Physics A: Mathematical and General*, 21(13):2979, 1988. URL <http://stacks.iop.org/0305-4470/21/i=13/a=021>.
- [81] G. P. Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192 – 203, 1978. ISSN 0021-9991. doi: [https://doi.org/10.1016/0021-9991\(78\)90004-9](https://doi.org/10.1016/0021-9991(78)90004-9).
- [82] E. Lindeberg and D. Wessel-Berg. Vertical convection in an aquifer column under a gas cap of CO<sub>2</sub>. *Energy Conversion and Management*, 38:S229–S234, 1997. ISSN 0196-8904. doi: [http://dx.doi.org/10.1016/S0196-8904\(96\)00274-9](http://dx.doi.org/10.1016/S0196-8904(96)00274-9). Proceedings of the Third International Conference on Carbon Dioxide Removal.
- [83] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. ISSN 1467-9868. doi: [10.1111/j.1467-9868.2011.00777.x](https://doi.org/10.1111/j.1467-9868.2011.00777.x).
- [84] C. R. Liro, E. E. Adams, and H. J. Herzog. Modeling the release of CO<sub>2</sub> in the deep ocean. Technical report, MIT, 1991. URL [http://dspace.mit.edu/bitstream/handle/1721.1/60614/EL\\_TR\\_1991\\_002.pdf?sequence=1](http://dspace.mit.edu/bitstream/handle/1721.1/60614/EL_TR_1991_002.pdf?sequence=1).

- [85] G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Jun 2014. ISBN 9781139017329. doi: 10.1017/CBO9781139017329.
- [86] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.
- [87] R. H. Lyon. *Statistical energy analysis of dynamical systems*. MIT, 1975.
- [88] R. H. Lyon and R. G. DeJong, editors. *Theory and Application of Statistical Energy Analysis*. Newnes, Boston, second edition edition, 1995. ISBN 978-0-7506-9111-6. doi: <https://doi.org/10.1016/B978-0-7506-9111-6.50001-4>.
- [89] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. ISSN 01621459. URL <http://www.jstor.org/stable/2280232>.
- [90] MPICH. <https://www.mpich.org/>, 2017.
- [91] H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bulletin of the American Mathematical Society*, 84(6):957–1041, 11 1978. URL <https://projecteuclid.org:443/euclid.bams/1183541461>.
- [92] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008. doi: <https://doi.org/10.1137/060663660>.
- [93] C. M. Oldenburg and K. Pruess. Dispersive transport dynamics in a strongly coupled groundwater-brine flow system. *Water Resources Research*, 31(2):289–302, 1995. ISSN 1944-7973. doi: <http://dx.doi.org/10.1029/94WR02272>.
- [94] G. S. H. Pau, J. B. Bell, K. Pruess, A. S. Almgren, M. J. Lijewski, and K. Zhang. Numerical studies of density-driven flow in CO<sub>2</sub> storage in saline aquifers. In *TOUGH Symposium 2009*, September 2009.
- [95] B. Peherstorfer, K. Willcox, and M. Gunzburger. Optimal model management for multifidelity Monte Carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, 2016. doi: 10.1137/15M1046472.
- [96] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference and optimization. Technical report, Aerospace Computational Design Laboratory TR-16-1, MIT, 2016.
- [97] K. Pruess, T. Xu, J. Apps, and J. Garcia. Numerical modeling of aquifer disposal of CO<sub>2</sub>. *Society of Petroleum Engineers*, pages 49–60, 2003. doi: 10.2118/66537-MS. URL <https://www.onepetro.org/conference-paper/SPE-66537-MS>.
- [98] P. A. Raviart and J. M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of Finite Element Methods*, pages 292–315, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg. ISBN 978-3-540-37158-8.
- [99] S. Rhebergen and G. N. Wells. A hybridizable discontinuous Galerkin method for the Navier–Stokes equations with pointwise divergence-free velocity field. *Journal of Scientific Computing*, Feb 2018. ISSN 1573-7691. doi: <http://dx.doi.org/10.1007/s10915-018-0671-4>.

- [100] A. Riaz, M. Hesse, H. A. Tchelepi, and F. M. Orr. Onset of convection in a gravitationally unstable diffusive boundary layer in porous media. *Journal of Fluid Mechanics*, 548:87–111, 2006. ISSN 1469-7645. doi: 10.1017/S0022112005007494.
- [101] E. Rosseel, N. Scheerlinck, and S. Vandewalle. Newton-multigrid for biological reaction-diffusion problems with random coefficients. *Numerical Mathematics: Theory, Methods and Applications*, 5(1):62–84, 2012. doi: 10.1017/S1004897900000222.
- [102] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003. doi: 10.1137/1.9780898718003.
- [103] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3): 856–869, 1986. doi: 10.1137/0907058.
- [104] P. G. Saffman and G. Taylor. The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 245(1242):312–329, 1958. ISSN 0080-4630. doi: 10.1098/rspa.1958.0085.
- [105] R. Scheichl. Multilevel Monte Carlo methods. Talk at "Key UQ methodologies and motivating applications" workshop at the Isaac Newton Institute for mathematical sciences, University of Cambridge, January 2018.
- [106] R. Scott. Finite element convergence for singular data. *Numerische Mathematik*, 21(4):317–327, Aug 1973. ISSN 0945-3245. doi: 10.1007/BF01436386.
- [107] G. Segol, G. F. Pinder, and W. Gray. A Galerkin-finite element technique for calculating the transient position of the saltwater front. *Water Resources Research*, 11(2):343–347, 1975. ISSN 1944-7973. doi: <http://dx.doi.org/10.1029/WR011i002p00343>.
- [108] P. Shorter and R. Langley. Vibro-acoustic analysis of complex systems. *Journal of Sound and Vibration*, 288(3):669 – 699, 2005. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2005.07.010>. Uncertainty in structural dynamics.
- [109] G. L. G. Sleijpen and H. A. van der Vorst. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM Journal of Matrix Analysis and Applications*, 17:401 – 524, 1996.
- [110] J. Slingo and T. Palmer. Uncertainty in weather and climate prediction. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 369(1956):4751–4767, 2011. ISSN 1364-503X. doi: 10.1098/rsta.2011.0161.
- [111] G. W. Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal of Matrix Analysis and Applications*, 23(3):601 – 614, 2001. doi: <http://dx.doi.org/10.1137/S0895479800371529>.
- [112] M. L. Szulczewski, M. A. Hesse, and R. Juanes. Carbon dioxide dissolution in structural and stratigraphic traps. *Journal of Fluid Mechanics*, 736:287–315, 12 2013. ISSN 1469-7645. doi: <http://dx.doi.org/10.1017/jfm.2013.511>.
- [113] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73 – 100, 1973. ISSN 0045-7930. doi: [https://doi.org/10.1016/0045-7930\(73\)90027-3](https://doi.org/10.1016/0045-7930(73)90027-3).

- [114] G. Taylor. Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 219 (1137):186–203, 1953. ISSN 0080-4630. doi: 10.1098/rspa.1953.0139.
- [115] A. Teckentrup, P. Jantsch, C. Webster, and M. Gunzburger. A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1046–1074, 10 2015. ISSN 2166-2525. doi: 10.1137/140969002.
- [116] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann. Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numerische Mathematik*, 125(3):569–600, Nov 2013. ISSN 0945-3245. doi: 10.1007/s00211-013-0546-4.
- [117] S. T. Tokdar and R. E. Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010. ISSN 1939-0068. doi: 10.1002/wics.56.
- [118] H. J. T. Unwin. Porous media examples, 2018. URL <https://doi.org/10.6084/m9.figshare.6463076>.
- [119] H. J. T. Unwin and N. Sime. Structural vibration examples, 2018. URL <https://doi.org/10.6084/m9.figshare.6463031>.
- [120] H. J. T. Unwin and G. N. Wells. Supporting material, 2015. URL <https://www.repository.cam.ac.uk/handle/1810/252804>. <https://bitbucket.org/ettieunwin/co2-dissolution>.
- [121] H. J. T. Unwin, G. N. Wells, and A. W. Woods. CO<sub>2</sub> dissolution in a background hydrological flow. *Journal of Fluid Mechanics*, 789:768–784, 2016. doi: 10.1017/jfm.2015.752.
- [122] H. J. T. Unwin, H. Hadfield, and N. Sime. podS: A scheduled multi-level Monte Carlo library. <https://bitbucket.org/ettieunwin/pods>, 2017.
- [123] H. J. T. Unwin, H. Hadfield, and N. Sime. podS docker container. [quay.io/pods/pods](http://quay.io/pods/pods), 2017.
- [124] H. J. T. Unwin, H. Hadfield, and N. Sime. podS MLMC library, 2018. URL <https://doi.org/10.6084/m9.figshare.6462902>.
- [125] M. van Reeuwijk, S. A. Mathias, C. T. Simmons, and J. D. Ward. Insights from a pseudospectral approach to the Elder problem. *Water Resources Research*, 45(4):n/a–n/a, 2009. ISSN 1944-7973. doi: <http://dx.doi.org/10.1029/2008WR007421>. W04416.
- [126] J. P. Verdon, J.-M. Kendall, A. L. Stork, R. A. Chadwick, D. J. White, and R. C. Bissell. Comparison of geomechanical deformation induced by megatonne-scale CO<sub>2</sub> storage at Sleipner, Weyburn, and In Salah. *Proceedings of the National Academy of Sciences*, 110(30):E2762–E2771, 2013. doi: 10.1073/pnas.1302156110.
- [127] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, 31(3):351–362, 2005.
- [128] C. I. Voss and W. R. Souza. Variable density flow and solute transport simulation of regional aquifers containing a narrow freshwater-saltwater transition zone. *Water Resources Research*, 23:1851–1866, 1987.
- [129] C. Wang, W. Gao, C. Song, and N. Zhang. Stochastic interval analysis of natural frequency and mode shape of structures with uncertainties. *Journal of Sound and Vibration*, 333(9):2483–2503, 2014. ISSN 0022-460X. URL <http://dx.doi.org/10.1016/j.jsv.2013.12.015>.

- [130] P. Wesseling. *An introduction to multigrid methods*. Pure and applied mathematics. John Wiley & Sons Australia, Limited, 1992. ISBN 9780471930839. URL <https://books.google.co.uk/books?id=MzmvAAAAMAAJ>.
- [131] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. ISSN 00029327, 10806377. URL <http://www.jstor.org/stable/2371268>.
- [132] R. A. Wooding. Growth of fingers at an unstable diffusing interface in a porous medium or Hele-Shaw cell. *Journal of Fluid Mechanics*, 39(3):477–495, 1969. doi: 10.1017/S002211206900228X.
- [133] A. W. Woods. *Flow in Porous Rock*. Cambridge University Press, 2015.
- [134] A. W. Woods and T. Espie. Controls on the dissolution of CO<sub>2</sub> plumes in structural traps in deep saline aquifers. *Geophysical Research Letters*, 39(8), 2012. doi: <http://dx.doi.org/10.1029/2012GL051005>. L08401.
- [135] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005. doi: <https://doi.org/10.1137/040615201>.
- [136] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002. doi: <http://dx.doi.org/10.1137/S1064827501387826>.

## Appendix A

# Derivation of analytical eigenpairs for Karhunen Loéve expansion

In this appendix, the analytical eigenpairs of the one dimensional Fredholm integral equation

$$\int_D C(x, x') \phi_j(x') dx' = v_j \phi_j(x) \quad \text{for } j = 1, 2, \dots, \infty, \quad (\text{A.1})$$

are derived for the separable exponential covariance kernel

$$C(x, x') = \sigma^2 e^{-\frac{|x-x'|}{\lambda}} \quad \text{on } D, \quad (\text{A.2})$$

where  $\phi_j$  and  $v_j$  are the  $j^{\text{th}}$  eigenfunctions and eigenvalues respectively,  $\lambda$  is a length scale and  $\sigma$  is the standard deviation. Two domains of interest are considered:  $D_1 = [-a, a]$  and  $D_2 = [0, a]$ . This derivation is based on Cliffe et al. [26], Ghanem and Spanos [48] and Lord et al. [85].

### A.1 Domain one

By substituting equation (A.2) into equation (A.1), the Fredholm integral can be written as

$$\int_D \sigma^2 e^{-\frac{|x-x'|}{\lambda}} \phi_j(x') dx' = v_j \phi_j(x) \quad \text{for } j = 1, 2, \dots, \infty \quad (\text{A.3})$$

for  $D_1 = [-a, a]$ . To aid in integration, by expanding the modulus, equation (A.3) can be written as:

$$v_j \phi_j(x) = \int_{-a}^x \sigma^2 e^{-\frac{-(x-x')}{\lambda}} \phi_j(x') dx' + \int_x^a \sigma^2 e^{-\frac{(x-x')}{\lambda}} \phi_j(x') dx'. \quad (\text{A.4})$$

Differentiating equation (A.4) once, using Leibnitz integration rule [42], results in

$$\begin{aligned} v_j \phi_j'(x) &= \sigma^2 e^{-\frac{(x-x)}{\lambda}} \phi_j(x) - \int_{-a}^x \frac{\sigma^2}{\lambda} e^{-\frac{(x-x')}{\lambda}} \phi_j(x') dx' \\ &\quad - \sigma^2 e^{\frac{(x-x)}{\lambda}} \phi_j(x) + \int_x^a \frac{\sigma^2}{\lambda} e^{\frac{(x-x')}{\lambda}} \phi_j(x') dx' \\ &= - \int_{-a}^x \frac{\sigma^2}{\lambda} e^{-\frac{(x-x')}{\lambda}} \phi_j(x') dx' + \int_x^a \frac{\sigma^2}{\lambda} e^{\frac{(x-x')}{\lambda}} \phi_j(x') dx'. \end{aligned} \quad (\text{A.5})$$

Differentiating equation (A.5) again,

$$\begin{aligned} v_j \phi_j''(x) &= -\frac{\sigma^2}{\lambda} e^{-\frac{(x-x)}{\lambda}} \phi_j(x) + \int_{-a}^x \frac{\sigma^2}{\lambda^2} e^{-\frac{(x-x')}{\lambda}} \phi_j(x') dx' - \frac{\sigma^2}{\lambda} e^{\frac{(x-x)}{\lambda}} \phi_j(x) \\ &\quad + \int_x^a \frac{\sigma^2}{\lambda^2} e^{\frac{(x-x')}{\lambda}} \phi_j(x') dx' \\ &= -\frac{2\sigma^2}{\lambda} \phi_j(x) + \int_{-a}^a \frac{\sigma^2}{\lambda^2} e^{-\frac{|x-x'|}{\lambda}} \phi_j(x') dx' \\ &= -\frac{2\sigma^2}{\lambda} \phi_j(x) + \frac{1}{\lambda^2} v \phi_j(x). \end{aligned} \quad (\text{A.6})$$

By rearranging equation (A.6), an equation for the eigenfunctions,  $\phi_j$ , can be found:

$$\phi_j''(x) + \omega_j^2 \phi_j(x) = 0, \quad (\text{A.7})$$

where

$$\omega_j^2 = \frac{2\sigma^2 \lambda^{-1} - \lambda^{-2} v_j}{v_j}. \quad (\text{A.8})$$

The associated boundary conditions for equation (A.7) can be found by evaluating (A.3) and (A.5) at  $x = \pm a$ :

$$v_j \phi_j(a) = \int_{-a}^a \sigma^2 e^{-\frac{(a-x')}{\lambda}} \phi_j(x') dx', \quad (\text{A.9a})$$

$$v_j \phi_j(-a) = \int_{-a}^a \sigma^2 e^{\frac{(-a-x')}{\lambda}} \phi_j(x') dx', \quad (\text{A.9b})$$

$$v_j \phi_j'(a) = - \int_{-a}^a \frac{\sigma^2}{\lambda} e^{-\frac{(a-x')}{\lambda}} \phi_j(x') dx', \quad (\text{A.9c})$$

$$v_j \phi_j'(-a) = \int_{-a}^a \frac{\sigma^2}{\lambda} e^{\frac{(-a-x')}{\lambda}} \phi_j(x') dx'. \quad (\text{A.9d})$$

These boundary conditions can be written in a more convenient form by equating equations (A.9a) and (A.9c), and equations (A.9b) and (A.9d),

$$\phi_j(a) + \lambda \phi_j'(a) = 0, \quad (\text{A.10a})$$

$$\phi_j(-a) - \lambda \phi_j'(-a) = 0. \quad (\text{A.10b})$$

The eigenvectors can be found by solving equation (A.7) to give

$$\phi_j(x) = A_j \cos(\omega_j x) + B_j \sin(\omega_j x), \quad (\text{A.11})$$

where  $A_j$  and  $B_j$  are constants to be found. Substituting equation (A.11) into the boundary conditions in (A.10a) gives

$$\begin{aligned} \phi_j(a) + \lambda \phi_j'(a) &= 0 \\ A_j \cos(\omega_j a) + B_j \sin(\omega_j a) + \lambda(-A_j \omega_j \sin(\omega_j a) + B_j \omega_j \cos(\omega_j a)) &= 0 \\ A_j(\cos(\omega_j a) - \lambda \omega_j \sin(\omega_j a)) + B_j(\sin(\omega_j a) + \lambda \omega_j \cos(\omega_j a)) &= 0. \end{aligned} \quad (\text{A.12})$$

Dividing equation (A.12) by  $\lambda \cos(\omega_j a)$  results in

$$A_j(\lambda^{-1} - \omega_j \tan(\omega_j a)) + B_j(\omega_j + \lambda^{-1} \tan(\omega_j a)) = 0. \quad (\text{A.13})$$

Then substituting equation (A.11) into the boundary conditions in (A.10b) gives

$$\begin{aligned} \phi_j(-a) - \lambda \phi_j'(-a) &= 0 \\ A_j \cos(-\omega_j a) + B_j \sin(-\omega_j a) - \lambda(-A_j \omega_j \sin(-\omega_j a) + B_j \omega_j \cos(-\omega_j a)) &= 0 \\ A_j \cos(\omega_j a) - B_j \sin(\omega_j a) - \lambda(A_j \omega_j \sin(\omega_j a) + B_j \omega_j \cos(\omega_j a)) &= 0 \\ A_j(\cos(\omega_j a) - \lambda \omega_j \sin(\omega_j a)) - B_j(\sin(\omega_j a) + \lambda \omega_j \cos(\omega_j a)) &= 0. \end{aligned} \quad (\text{A.14})$$

Dividing equation (A.14) by  $\lambda \cos(\omega_j a)$  results in

$$A_j(\lambda^{-1} - \omega_j \tan(\omega_j a)) + B_j(\omega_j - \lambda^{-1} \tan(\omega_j a)) = 0. \quad (\text{A.15})$$

Two transcendental equations can be found by taking the non trivial solutions to solving equations (A.13) and (A.15) simultaneously

$$\lambda^{-1} - \omega_j \tan(\omega_j a) = 0, \quad (\text{A.16a})$$

$$\omega_j + \lambda^{-1} \tan(\omega_j a) = 0. \quad (\text{A.16b})$$

Assuming  $\hat{\omega}_j$  are the positive roots of (A.16a) and  $\tilde{\omega}_j$  of (A.16b),

$$\omega_j := \begin{cases} \hat{\omega}_{\lceil j/2 \rceil}, & j \text{ odd,} \\ \tilde{\omega}_{j/2}, & j \text{ even.} \end{cases} \quad (\text{A.17})$$

Therefore the eigenfunctions

$$\phi_j(x) := \begin{cases} A_j \cos(\omega_j x), & j \text{ odd,} \\ B_j \sin(\omega_j x), & j \text{ even.} \end{cases} \quad (\text{A.18})$$

The constants  $A$  and  $B$  are chosen to ensure the eigenvalues are orthonormal  $\|\phi_j\|_{L^2_{(-a,a)}} = 1$ . Therefore,

$$\begin{aligned} \sqrt{\int_{-a}^a A_j^2 \cos^2(\omega_j x) dx} &= 1 \\ \frac{A_j^2}{2} \int_{-a}^a \cos(2\omega_j x) + 1 dx &= 1 \\ \frac{A_j^2}{2} \left[ \frac{1}{2\omega_j} \sin(2\omega_j x) + x \right]_{-a}^a &= 1 \\ \frac{A_j^2}{2} \left[ \frac{1}{2\omega_j} \sin(2\omega_j a) + a - \frac{1}{2\omega_j} \sin(-2\omega_j a) + a \right] &= 1 \\ A_j \left( \frac{1}{2\omega_j} \sin(2\omega_j a) + a \right) &= 1 \\ A_j &= \frac{1}{\sqrt{a + \sin(2\omega_j a)/2\omega_j}}, \end{aligned} \quad (\text{A.19})$$

and

$$\begin{aligned} \sqrt{\int_{-a}^a B_j^2 \sin^2(\omega_j x) dx} &= 1 \\ \frac{B_j^2}{2} \int_{-a}^a 1 - \cos(2\omega_j x) dx &= 1 \\ \frac{B_j^2}{2} \left[ x - \frac{1}{2\omega_j} \sin(2\omega_j x) \right]_{-a}^a &= 1 \\ \frac{B_j^2}{2} \left[ a - \frac{1}{2\omega_j} \sin(2\omega_j a) + a + \frac{1}{2\omega_j} \sin(-2\omega_j a) \right] &= 1 \\ B_j \left( a - \frac{1}{2\omega_j} \sin(2\omega_j a) \right) &= 1 \\ B_j &= \frac{1}{\sqrt{a - \sin(2\omega_j a)/2\omega_j}}. \end{aligned} \quad (\text{A.20})$$

The eigenvalues can be found by rearranging equation (A.8)

$$\begin{aligned}\omega_j^2 &= \frac{2\sigma^2\lambda^{-1} - \lambda^{-2}v_j}{v_j} \\ v_j\omega_j^2 &= \frac{2\sigma^2\lambda - v_j}{\lambda^2} \\ v_j(\omega_j^2\lambda^2 + 1) &= 2\sigma^2\lambda \\ v_j &= \frac{2\sigma^2\lambda}{\omega_j^2\lambda^2 + 1}.\end{aligned}\tag{A.21}$$

## A.2 Domain two

The derivation of the eigenpairs for the second domain  $D_2 = [0, a]$  begins the same as for the first domain, but the boundary conditions differ. Equations (A.3) and (A.5) are now evaluated for  $x = 0, a$ :

$$v_j\phi_j(a) = \int_0^a \sigma^2 e^{-\frac{(a-x')}{\lambda}} \phi_j(x') dx',\tag{A.22a}$$

$$v_j\phi_j(0) = \int_0^a \sigma^2 e^{\frac{x'}{\lambda}} \phi_j(x') dx',\tag{A.22b}$$

$$v_j\phi_j'(a) = -\int_0^a \frac{\sigma^2}{\lambda} e^{-\frac{(a-x')}{\lambda}} \phi_j(x') dx',\tag{A.22c}$$

$$v_j\phi_j'(0) = \int_0^a \frac{\sigma^2}{\lambda} e^{\frac{x'}{\lambda}} \phi_j(x') dx'.\tag{A.22d}$$

These boundary conditions can be written in a more convenient form by equating equations (A.22a) and (A.22c), and equations (A.22b) and (A.22d),

$$\phi(a) + \lambda\phi'(a) = 0,\tag{A.23a}$$

$$\phi(0) - \lambda\phi'(0) = 0.\tag{A.23b}$$

These boundary conditions can be expanded by substituting the eigenvectors from equation (A.11) into equation (A.23). The first boundary condition remains the same as equation (A.13), but the second boundary condition becomes:

$$A_j \cos(0) + B_j \sin(0) - \lambda(-\omega_j A_j \sin(0) + \omega_j B_j \sin(0)) = 0$$

$$A_j = \lambda\omega_j B_j,\tag{A.24}$$

where  $A_j$  and  $B_j$  are different constants to be found. Substituting (A.24) into (A.13) and ignoring the trivial solution gives the equation for the roots

$$\begin{aligned}\lambda \omega_j B_j (\lambda^{-1} - \omega_j \tan(\omega_j a)) + B_j (\omega_j + \lambda^{-1} \tan(\omega_j a)) &= 0, \\ \omega_j - \omega_j^2 \tan(\omega_j a) + \omega_j + \lambda^{-1} \tan(\omega_j a) &= 0 \\ 2\omega_j + \tan(\omega_j a) (\lambda^{-1} - \omega_j^2 \lambda) &= 0 \\ \tan(\omega_j a) &= \frac{2\omega_j \lambda}{\omega_j^2 \lambda^2 - 1},\end{aligned}$$

where

$$\omega_j = \frac{1}{a} \tan^{-1} \left( \frac{2\omega_j \lambda}{\omega_j^2 \lambda^2 - 1} \right). \quad (\text{A.25})$$

An equation for the eigenvectors can be found by substituting equation (A.24) into equation (A.11)

$$\phi_j = B_j (\lambda \omega_j \cos(\omega_j x) + \sin(\omega_j x)). \quad (\text{A.26})$$

Again  $B_j$  can be found by assuming the eigenfunctions are orthonormal

$$\begin{aligned}\sqrt{\int_0^a \lambda^2 \omega_j^2 B_j^2 \cos^2(\omega_j x) + B_j^2 \sin^2(\omega_j x) dx} &= 1 \\ \int_0^a \frac{\lambda^2 \omega_j^2 B_j^2}{2} (\cos(2\omega_j x) + 1) + \frac{B_j^2}{2} (1 - \cos(2\omega_j x)) dx &= 1 \\ \frac{B_j^2}{2} \left[ \frac{\lambda^2 \omega_j^2}{2\omega_j} \sin(2\omega_j x) + (\lambda^2 \omega_j^2 + 1)x - \frac{1}{2\omega_j} \sin(2\omega_j x) \right]_0^a &= 1 \\ \frac{B_j^2}{2} \left[ \frac{\lambda^2 \omega_j^2}{2\omega_j} \sin(2\omega_j a) + (\lambda^2 \omega_j^2 + 1)a - \frac{1}{2\omega_j} \sin(2\omega_j a) \right] &= 1 \\ \frac{B_j^2}{2} \left[ \frac{\lambda^2 \omega_j^2 - 1}{2\omega_j} \sin(2\omega_j a) + (\lambda^2 \omega_j^2 + 1)a \right] &= 1 \\ B_j^2 &= \frac{4\omega_j}{(\lambda^2 \omega_j^2 - 1) \sin(2\omega_j a) + 2\omega_j a (\lambda^2 \omega_j^2 + 1)}. \quad (\text{A.27})\end{aligned}$$

The eigenvalues can still be found by equation (A.21) but with the roots given by (A.25).

# Appendix B

## PodS module description

This appendix describes the design and distribution of the podS library.

### B.1 Statistics module

The MC and MLMC algorithms, derived in Sections 3.1 and 3.3, are implemented in the statistics module. This module contains two similar classes *MC* and *MLMC* that compute the expectation and variance of the QoI to user given tolerances for each method and a generic solver class.

The generic solver class determines what the user written solver code must adhere to. The generic solver class contains the following public methods that must be implemented by the user:

- *pod\_solve* to return the QoIs from one solve of the stochastic FE model using either the MC or MLMC method. The method must have the following form

```
std::vector<double>  
pod_solve(PodsMesh& pods_mesh, double seed, bool fine_mesh);
```

where *pod\_mesh* contains the coarse and fine mesh for the current level, see Section B.3, *seed* is a process dependent seed that is used to initialise the random field and *fine\_mesh* determines if the current solve should use the coarse or fine mesh.

- *get\_meshes* to build the *PodsMesh* hierarchy, that includes the initial, coarse and fine meshes, for the current level. This method must have the follow form

```
void get_meshes(MPI_Comm comm, PodsMesh& pods_mesh, int level);
```

where *comm* is the MPI communicator associated with the current solve, *pod\_mesh* is an empty *PodsMesh* struct that will hold the meshes and *int* is the current level.

- *get\_num\_qoi* to return the number of QoIs that are to be computed. It must have the following form

```
int get_num_qoi();
```

where there are no input arguments. This is used to size the *QOIholder* struct.

- *get\_solver\_info* to store information about the solver for use by the logger, see Section B.6, and must have the following form

```
solver_info get_solver_info();
```

where there is no input arguments and the return type, *solver\_info*, is a structure containing information about the solver class for use in logging.

- *get\_min\_num\_cores\_per\_level* to return the minimum number of cores that should be assigned to one solve of the stochastic PDE for a given level. It must have the following form

```
std::vector<int> get_min_num_cores_per_level(int num_levels);
```

where *num\_levels* is the current MLMC level. This is necessary to avoid running out of memory when solving each level on different architectures. For example, on one computer six level zero computations may fit in memory but only one level three computation, whereas, on an alternative machine, 24 level three computations may fit easily. Carefully selecting the number of cores is especially important on supercomputers where the memory is not necessarily shared across nodes.

## B.2 Uncertainty module

The uncertainty module contains classes that represent uncertainty in the PDE. This code is divided into two: code that generates a probability space and code that represents a stochastic function.

### B.2.1 Probability space

The *ProbabilitySpace* class is used to determine the type of random number generator that is sampled to generate a stochastic function. This class can be instantiated by

```
ProbabilitySpace ProbabilitySpace(std::string type, double seed);
```

where *type* is the type of random number generator required and *seed* is the process dependent seed used to initialise the random number generator for that sample. This class contains one method, *generate\_random\_number\_generator*, that creates either a pre-seeded uniform or normal random number generator for a given range, or mean and standard deviation, using the *RandomNumberGenerator* class.

### B.2.2 Stochastic function

The uncertainty module also includes classes for representing stochastic functions in PDEs using point masses, numerical KL expansions and analytical KL expansions, see Sections 2.5 and 2.6. However, new classes can be implemented to represent other stochastic functions, using different methods. Any new classes written would need to adhere to the *StochasticFunction* class. This *StochasticFunction* class has one method, *sample*, that realises the stochastic function by sampling the probability space.

The instantiation of the methods for realising the random field included in podS are described below:

- Point masses require a uniformly distributed probability space and can be instantiated by

```
PointSourceGenerator(ProbabilitySpace omega, MPI_Comm comm,
    int num_ps, double mean, double scaling,
    std::vector<double> mesh_dimensions);
```

where *omega* is the probability space, *comm* is the MPI communicator for the current solve, *num\_ps* is the number of point masses required, *mean* and *scaling* are the mean and maximum scaling of the point masses and *mesh\_dimensions* is a vector of the mesh dimensions to ensure point masses are not added outside the domain.

- Numeric KL expansions require a normally distributed probability space and can be instantiated by

```
TruncatedKL(std::shared_ptr<const dolfin::Mesh> mesh,
    ProbabilitySpace& omega, double mu, double sigma,
    std::vector<double> lambda, int truncation_number,
    bool debug);
```

where *mesh* represents the domain with the desired fidelity, *omega* is the probability space, *mu* and *sigma* are the mean and standard deviation of the stochastic field, *lambda* and *truncation\_number* are the length scale and truncation point of the KL expansion, and *debug* is an indicator used to log the eigenpairs.

- Analytic KL expansions require a uniformly distributed probability space and can be initialised similar to numeric KL expansions, but initial guesses of the roots of the transcendental equations are required, see Section 2.6.2,

```
TruncatedKLANalytic(std::shared_ptr<const dolfin::Mesh> mesh,
    ProbabilitySpace& omega, double mu, double sigma,
    int truncation_number, std::vector<NRInfo> nr_info,
    bool debug);
```

The vector of *NRInfo* structs includes information about length scales and roots of the transcendental equations for each dimension of the problem.

It may be more realistic to implement an exponential stochastic field for some applications, such as modelling the permeability in porous media. These stochastic functions can be instantiated using the same arguments as the above distributions. The mean and standard deviation are therefore describing the underlying normal distribution, not the log-normal distribution.

- Numeric exponential truncated KL expansion

```
ExponentialTruncatedKL(std::shared_ptr<const dolfin::Mesh> mesh,
    ProbabilitySpace& omega, double mu, double sigma,
    std::vector<double> lambda, int truncation_number,
    bool debug)
```

- Analytic exponential truncated KL expansion

```
ExponentialTruncatedKLANalytic(
    std::shared_ptr<const dolfin::Mesh> mesh,
    ProbabilitySpace& omega,
    double mu, double sigma, int truncation_number,
    std::vector<NRInfo> nr_info, bool debug)
```

## B.3 Mesh module

The mesh module includes the *PodsMesh* class that is used to generate a mesh hierarchy in the *solver\_class*. The methods in this class set and recover the level zero, coarse and fine meshes for a given level.

Currently, two methods from the *MeshGenerator* class can be used to generate this hierarchy: *create\_mesh* and *read\_mesh*. It is important that the right communicator is given to the mesh so that multiple independent problem solves can be run in parallel.

### B.3.1 Create mesh

The *create\_mesh* method uses refinement to create a *PodsMesh*. Each time the FEniCS [4] refine tool is invoked, each element in the mesh is divided into two. This increases the number of elements in the mesh by  $2^n$ , where  $n$  is the number of dimensions in the mesh. This method can be implemented for a given level as follows:

```
PodsMesh::create_meshes(PodsMesh& pods_mesh,
    std::shared_ptr<Mesh> initial_mesh, MPI_Comm comm, int level);
```

where *pods\_mesh* is the empty *PodsMesh*, *mesh* is the level zero mesh that will be refined, *comm* is the MPI communicator and the *level* determines which *PodsMesh* to generate.

### B.3.2 Read mesh

The `read_mesh` method generates a `PodsMesh` hierarchy by reading meshes from a `.xdmf` file. This is particularly useful if the mesh is large and refining it multiple times is computationally expensive. This method can be implemented as follows

```
PodsMesh::read_meshes(PodsMesh& pods_mesh,
    std::vector<std::string> meshes, MPI_Comm comm, int level);
```

where instead of passing an initial mesh to the constructor, a vector of mesh file paths is required that orders the meshes from coarsest to finest.

## B.4 Scheduling module

The individual samples, from the MC and MLMC routine method, are arranged into batches with other samples of the same level using the scheduling module. These batches are then scheduled to ensure that the MC or MLMC simulations are run in the optimal time for each computer architecture.

The number of samples in each batch depends on the user-defined minimum number of cores for each level. Initially each sample is assigned the minimum number of cores and any extra ones are split equally between the samples. This means that the number of parallel samples may be different for each level. It is important that each sample in a batch takes approximately the same time, since the model is computed from the batch timings not the individual sample timings. If more than one core is used for each sample, the cost parameter is approximated using Amdahl's law [5]. This is a formula that is often used in parallel computing to predict theoretical speed ups when using multiple cores to compute one problem.

## B.5 Pod module

The pod module controls the multiple concurrent samples from within the MC and MLMC *routine* method. This module comprises of two classes: *ParallelJob* and *ParallelController*. The *ParallelJob* class represents an individual sample of the QoI and ensures that the correct communicator is given to the *solver\_class*. The *ParallelController* class controls the concurrent samples and splits the global MPI communicator into individual local ones for each sample within a batch. The estimations of the quantities of interest on each core are broadcast to all the other core so that the same final expectation and variance are obtained on each core.

## B.6 Log module

The log module provides a logging functionality to the podS library and saves the simulation results to a `.csv` file, if a file path is provided to the *MC* or *MLMC* class instantiation. It is based on the FEniCS project logger [4] and provides debug and log messages at different levels of priority.

The *PodsLogManager* controls a singleton *PodsLogger* class, which prints messages to the standard output. A message can be printed to the standard output using:

```
 pods::pods_log(int log_level, std::string msg,  
               const std::vector<MPI_Comm>& comms )
```

where *log\_level* is the priority of the method, *msg* is the message to print and *comms* is a vector of the process numbers to display the message on. If no process numbers are given, the message is printed on all processes.

## B.7 Distribution

There are two methods for obtaining the seven, previously discussed, modules that form the podS library. One option is to clone the source code from the public Bitbucket repository [122]. Installation instructions are provided in the README file or an install script *install\_pods.sh* can be run. The podS library has the following dependencies: FEniCS [4], PETSc [10], SLEPc [127] and MPI, which need to be installed prior to podS. Alternatively, a Docker container including the podS library, PETSc, SLEPc and MPI has been released and is hosted at quay.io [123]. A new build of the library is triggered when the master branch of the source library is updated.

## Appendix C

# Convergence rates for eigenvalue problems

In this appendix, the convergence rates of the FE models presented in Section 4.1 are checked against theoretical predictions, with and without point masses added to the formulations. This is because, as suggested in Section 2.5, the regularity of the FE formulation is reduced by using Dirac delta functions and this can cause sub-optimal convergence rates.

As explained in Section 2.3, if the eigenvalues are computed from a smooth FE model using piecewise polynomials of degree  $p$ , the convergence rate of the error should be [14]

$$|\lambda^k - \lambda_h^k| = O(h^{2p}), \quad (\text{C.1})$$

where  $\lambda^k$  is the exact  $k^{\text{th}}$  eigenvalue and  $\lambda_h^k$  is the approximate  $k^{\text{th}}$  eigenvalue. Since there is no known exact solution of  $\lambda^k$  for each example, it is approximated using a fine representation of the mesh.

To ensure a deterministic system is used to investigate the convergence rates of the FE formulations with point masses, the same 30 point masses are added for all the different fidelities considered. The maximum magnitude of a point mass used in this investigation is 10% of the total mass of the unperturbed structure.

### C.1 Wave equation

The convergence rates of the error in the first non zero eigenvalue of the wave equation in (4.2) are shown in Figure C.1a for linear and quadratic elements. As expected, when no point masses are added, the convergence rate of the  $p = 1$  elements is two and the  $p = 2$  elements is four. However, when point masses are added, the convergence rate become sub-optimal for both order elements and tends towards one.

The reduction in convergence rate in Figure C.1b is similar to the reduction in convergence rates for the Poisson equation, with a singular forcing term, that is presented in Section 2.5. When point

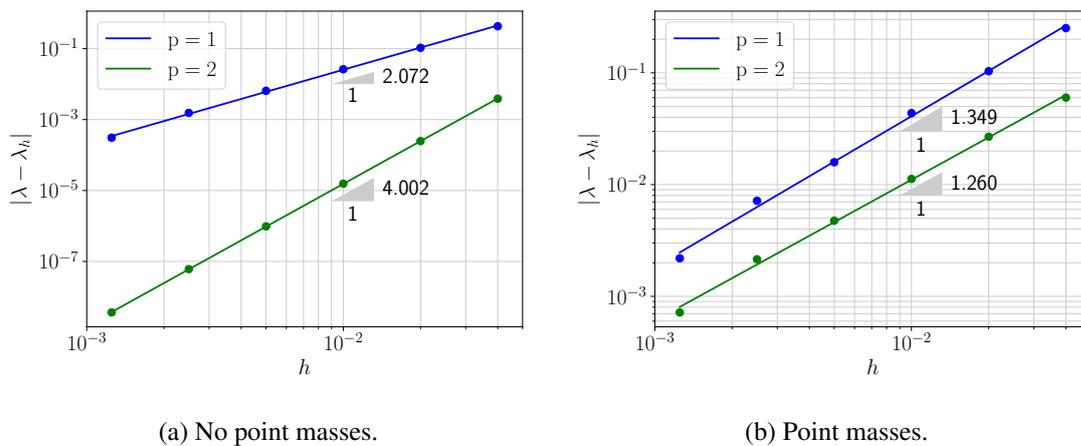


Figure C.1: Convergence rates of the error in the first non zero eigenvalue of the wave equation for  $p = 1, 2$ . In Fig (a) no point masses have been added, but in Fig (b) 30 point masses have been added.

masses are used to add uncertainty to the problem, the mass matrices includes singularities. This means the regularity of the formulation is reduced and the convergence rates are sub-optimal.

## C.2 Kirchhoff–Love plate

The convergence rate of the error in the first eigenvalue of the Kirchhoff–Love plate equation in (4.7) are shown in Figure C.2. Again as expected, when no point masses are added, the convergence rate of the error in the first eigenvalue of the Kirchhoff–Love plate with  $p = 1$  elements is two and with  $p = 2$  elements is four. However, when point masses are added, the convergence rates of the error drop to approximately two for both order elements. This is different to the change in order of convergence for the wave equation because the Kirchhoff–Love plate equation is a fourth order equation, so  $u \in H^2(D)$ . In contrast, the wave equation is a second order equation so  $u \in H^1(D)$ .

## C.3 Three dimensional elastodynamics

The convergence rate of the error in the first eigenvalue of the three dimensional elastodynamics problem in equation (4.10) is shown in Figure C.3a. Again as expected, the convergence rate of the  $p = 1$  element is two. However, the convergence rate is only 3.4 for the  $p = 2$  element. This is most likely because the mesh cell size was too large to observe optimal convergence. Finer meshes could not be used due to computer memory limitations. In addition, the exact solution  $\lambda^k$  used is not exact but only an approximation, which may worsen the convergence rate.

As observed in Figure C.3b, the convergence rates fall when point masses are added. Since the elastodynamics equation is second order, the convergence rates should tend to one when point masses are added.

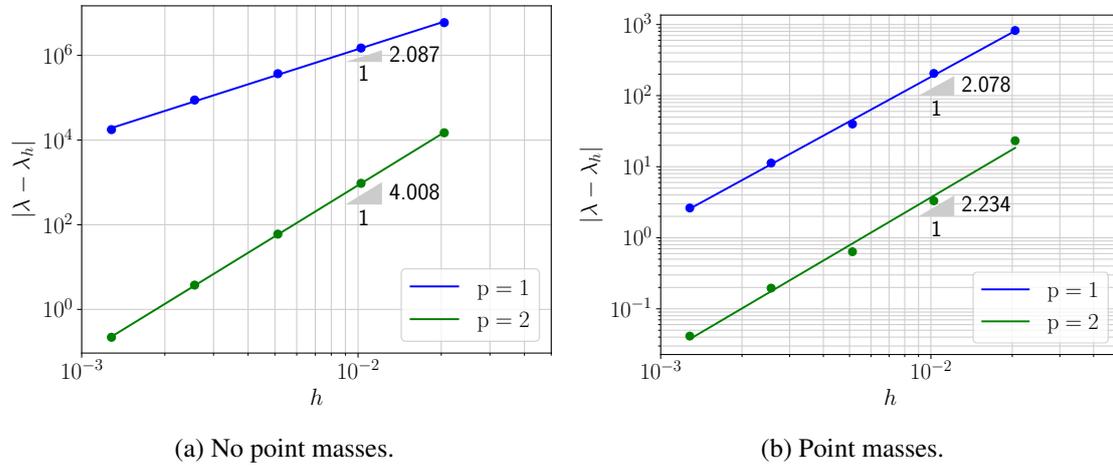


Figure C.2: Convergence rates of the error in the first non zero eigenvalue of the Kirchhoff-Love plate equation for  $p = 1, 2$ . In Fig (a) no point masses have been added but in Fig (b) 30 point masses have been added.

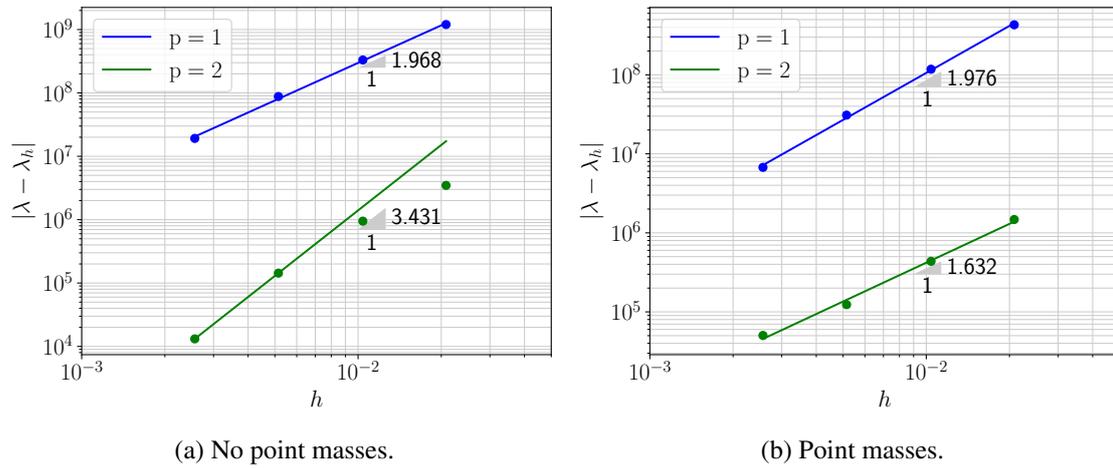


Figure C.3: Convergence rates of the error in the first non zero eigenvalue of the elastodynamics equation for  $p = 1, 2$ . In Fig (a) no point masses have been added but in Fig (b) 30 point masses have been added.



## Appendix D

# Derivation of kinetic energy density of the Kirchhoff–Love plate

The damped Kirchhoff–Love plate equation can be written as

$$\frac{\partial^2 \tilde{u}}{\partial t^2} + \beta \frac{\partial \tilde{u}}{\partial t} + \frac{K}{2a\rho} \nabla^4 \tilde{u} = \frac{f}{2a\rho} \quad \text{in } D, \quad (\text{D.1})$$

where  $2a$  is the plate thickness,  $\rho$  is the density,  $K$  is the constant bending stiffness,  $f$  is a body force and  $\beta \geq 0$  is a constant coefficient.

The generalised eigenvalue problem can be found by the method of separation of variables applied to the homogeneous system ( $f = 0$ ). If  $u(\mathbf{x}, t) = T(t)X(\mathbf{x})$ , equation (D.1) can be re-written as

$$X \left( \frac{\partial^2 T}{\partial t^2} + \beta \frac{\partial T}{\partial t} \right) + \frac{K}{2a\rho} T \nabla^4 X = 0. \quad (\text{D.2})$$

Rearranging equation (D.2) yields

$$\frac{1}{T} \left( \frac{\partial^2 T}{\partial t^2} + \beta \frac{\partial T}{\partial t} \right) = -\frac{1}{X} \frac{K}{2a\rho} \nabla^4 X. \quad (\text{D.3})$$

Since the left-hand side of equation (D.3) is a function of  $t$  only, and the right-hand side is a function of  $x$  only, then for some  $\lambda$

$$\frac{\partial^2 T}{\partial t^2} + \beta \frac{\partial T}{\partial t} = -\lambda T, \quad (\text{D.4a})$$

$$\nabla^4 X = \frac{2a\rho\lambda}{K} X. \quad (\text{D.4b})$$

Equation (D.4b) is an eigenvalue problem with solutions  $(\lambda_n, \phi_n)$  for  $n = 1, 2, \dots, \infty$  where  $\lambda_n$  is the  $n^{\text{th}}$  eigenvalue and  $\phi_n$  is the  $n^{\text{th}}$  eigenfunction. These eigenfunctions are orthonormal with respect

to the density

$$\int_D \frac{2a\rho}{K} \phi_m \phi_n \, dx = \delta_{m,n}. \quad (\text{D.5})$$

and form the basis of a Hilbert space, so can be used for Fourier expansions. The displacement of the plate,  $\tilde{u}$ , can therefore be expanded in the Fourier basis for  $n = 1, 2, \dots, \infty$

$$\tilde{u}(\mathbf{x}, t) = \sum_{n=1}^{\infty} u_n = \sum_{n=1}^{\infty} \alpha_n(t) \phi_n(\mathbf{x}), \quad (\text{D.6})$$

where  $\alpha_n$  is the  $n^{\text{th}}$  unknown time dependent coefficient and  $\phi_n$  is the  $n^{\text{th}}$  eigenfunction. Letting  $F = f/2a\rho$ , the Fourier expansion of the forcing for  $n = 1, 2, \dots, \infty$  is

$$F(\mathbf{x}, t) = \sum_{n=1}^{\infty} F_n(t) \phi_n(\mathbf{x}). \quad (\text{D.7})$$

where  $F_n$  is the  $n^{\text{th}}$  time dependent coefficient. Using the expansions in (D.6) and (D.7), the Kirchhoff–Love plate equation (D.1) can be written as

$$\sum_{n=1}^{\infty} \left( \phi_n \frac{\partial^2 \alpha_n}{\partial t^2} + \beta \phi_n \frac{\partial \alpha_n}{\partial t} + \frac{K}{2\rho a} \alpha_n \nabla^4 \phi_n \right) = \sum_{n=1}^{\infty} F_n \phi_n. \quad (\text{D.8})$$

Substituting in equation (D.4b)

$$\sum_{n=1}^{\infty} \phi_n \frac{\partial^2 \alpha_n}{\partial t^2} + \beta \phi_n \frac{\partial \alpha_n}{\partial t} + \alpha_n \lambda_n \phi_n = \sum_{n=1}^{\infty} F_n \phi_n. \quad (\text{D.9})$$

Equation (D.9) can be further simplified by multiplying the equation by  $2a\rho\phi_m/K$  and integrating over the domain

$$\int_D \sum_{n=1}^{\infty} \frac{2a\rho}{K} \left( \phi_n \frac{\partial^2 \alpha_n}{\partial t^2} + \beta \phi_n \frac{\partial \alpha_n}{\partial t} + \alpha_n \lambda_n \phi_n \right) \phi_m = \int_D \sum_{n=1}^{\infty} \frac{2a\rho}{K} F_n \phi_n \phi_m \quad m = 1, 2, \dots, \infty. \quad (\text{D.10})$$

Using the orthogonality property in equation (D.5) yields an independent system of equations

$$\frac{\partial^2 \alpha_n}{\partial t^2} + \beta \frac{\partial \alpha_n}{\partial t} + \alpha_n \lambda_n = F_n. \quad (\text{D.11})$$

Now the inhomogeneous system is considered with time harmonic forcing of frequency,  $\zeta$ , at a point,  $x_0$ ,

$$f = \delta(\mathbf{x} - \mathbf{x}_0) e^{i\zeta t}. \quad (\text{D.12})$$

where  $i = \sqrt{-1}$  and  $\delta$  is a Dirac delta function. Therefore,

$$F = \frac{1}{2a\rho} \delta(\mathbf{x} - \mathbf{x}_0) e^{i\zeta t}. \quad (\text{D.13})$$

The projection of  $F$  in equation (D.13) onto the space of eigenfunctions is

$$F_n := \int_D \frac{1}{2a\rho} \frac{2a\rho}{K} \delta(\mathbf{x} - \mathbf{x}_0) e^{i\zeta t} \phi_n \, dD = \frac{1}{K} e^{i\zeta t} \phi_n(\mathbf{x}_0). \quad (\text{D.14})$$

Assuming the solution to equation (D.11) has the form

$$\alpha_n = \sigma_n e^{i\zeta t}; \quad \frac{\partial \alpha_n}{\partial t} = i\zeta \sigma_n e^{i\zeta t}; \quad \frac{\partial^2 \alpha_n}{\partial t^2} = -\zeta^2 \sigma_n e^{i\zeta t}, \quad (\text{D.15})$$

where  $\sigma_n$  is the  $n^{\text{th}}$  transfer function, the equation (D.11) can be re-written as

$$-\zeta^2 \sigma_n e^{i\zeta t} + \beta i \zeta \sigma_n e^{i\zeta t} + \lambda_n \sigma_n e^{i\zeta t} = \frac{1}{K} e^{i\zeta t} \phi_n(\mathbf{x}_0). \quad (\text{D.16})$$

Simplifying equation (D.16),

$$\sigma_n \left( -\zeta^2 + \beta i \zeta + \lambda_n \right) = \frac{1}{K} \phi_n(\mathbf{x}_0), \quad (\text{D.17})$$

and rearranging equation (D.17),

$$\sigma_n = \frac{\phi_n(\mathbf{x}_0)}{K (\lambda_n - \zeta^2 + \beta i \zeta)}. \quad (\text{D.18})$$

Equation (D.18) is more commonly written as

$$\sigma_n = \frac{\phi_n(\mathbf{x}_0)}{K (\zeta_n^2 - \zeta^2 + \eta i \zeta \zeta_n)}, \quad (\text{D.19})$$

where  $\lambda_n = \zeta_n^2$  and  $\eta = \beta / \zeta_n$ .

The time averaged kinetic energy density of a two dimensional system is given by

$$E = \frac{\zeta^2}{4R} \int_D \rho |\sigma_n|^2 \, dx, \quad (\text{D.20})$$

where  $R$  is the area of the plate [78]. Substituting equation (D.19) into equation (D.20),

$$E = \frac{\zeta^2}{4R} \int_D \rho \frac{\phi_n^2(\mathbf{x}_0)}{K^2 ((\zeta_n^2 - \zeta^2)^2 + (\eta \zeta \zeta_n)^2)} \, dx, \quad (\text{D.21})$$

and using the orthogonality property from equation (D.5),

$$E = \frac{1}{4R(2aK)} \frac{\phi_n^2(\mathbf{x}_0)}{((\zeta_n^2 - \zeta^2)^2 + (\eta \zeta \zeta_n)^2)}. \quad (\text{D.22})$$