University of Strathclyde Glasgow

Efficient multi-fidelity methods for global sensitivity analysis of pollutant dispersion models

Department of Mathematics and Statistics

University of Strathclyde,

Glasgow, Scotland

A thesis submitted for the degree of

Doctor of Philosophy

Stephen Smith

October 2025

Declaration of Authenticity

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: *S Smith*

Date: October 22, 2025

Acknowledgements

I am grateful to my supervisors, Professor John Mackenzie and Dr Alison Ramage for their guidance throughout this thesis. I would also like to thank NPL, and in particular Professor Alistair Forbes, for their support.

I am sincerely grateful to my brother Paul, for going above and beyond to keep me going, especially in the most challenging moments. Without his efforts completing this thesis would not have been possible, and I cannot thank him enough. I am also grateful to Natalie Smith, without whom Paul would never have been able to help me. I would like to thank all those who supported me over the years to make sure I had the time and freedom to work, particularly Michael and Alana Smith who were always available when I needed them.

Finally and most importantly, I want to thank my parents, Fiona and Brian Smith for their incredible support throughout the last five years. Without their unbelievable patience, encouragement, love and emotional support, I would never have been able to begin this project or see it through to the end. I am more grateful than I can ever express.

Abstract

Air pollution models play a vital role in understanding environmental impact, but ensuring model accuracy comes with significant computational expense. To balance this trade off, we investigate multi-fidelity approaches that mix costly high-fidelity simulations with inexpensive approximations to achieve efficient and reliable predictions. This thesis develops methods for uncertainty quantification of advection–diffusion models, focusing on variance reduction. We study how uncertainty in advection, diffusion, and source strength affects pollutant transport, and design efficient strategies to estimate means, confidence intervals, and sensitivity indices.

A two-dimensional advection–diffusion equation is discretised using finite element methods. Using Monte Carlo estimation, we confirm that computed mean-squared errors agree with theory, and extend the method with bootstrapped confidence intervals, providing reliable uncertainty estimates at lower cost.

Variance reduction is achieved through new control variate methods. Building on control variates methods which allow control variates with unknown means, we introduce the control variates with estimated means and budget control (CVB) method which incorporates an explicit budget. Both methods are analysed, showing significant accuracy gains over standard Monte Carlo. We also propose a multi-fidelity bootstrap estimator, combining CVB with confidence intervals to produce consistently smaller intervals. We test numerical and data-informed surrogate models and find that neural network approximations are the most suitable as a control variate.

For sensitivity analysis, we compare standard Sobol index estimators and introduce a CVB based alternative that lowers variance while preserving confidence

interval coverage. We extend the CVB multi-fidelity bootstrap estimator to find confidence intervals for Sobol indices. Applied to the advection–diffusion model, this method enables parameter ranking with smaller budgets.

Finally, we extend the model to dimensional form with realistic input distributions and more uncertain parameters. Wind speed, diffusion, source strength, and wind direction are represented by Weibull, log-normal, gamma, and von Mises distributions respectively, with surrogates tested as low-fidelity models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Pollutant dispersal

There is documented evidence of air pollution being a going concern as far back as 400 B.C., when Hippocrates published "Airs, waters and place" [43]. Throughout history the problem of air pollution has been referenced in written accounts, fictional work and laws throughout the world. Air pollution affecting much of the world in the 1800's, driven by the industrial revolution, motivated a large expansion in measurement and study of air pollution. Measurements are useful as they can capture air pollution at a moment in time, but the need to predict levels of pollution, and mitigate any associated health effects, necessitated the development of mathematical models that could give a more complete view of air quality problems [28].

Sources of pollution are often not difficult to identify, with smoke stacks rising over most cities since the 1700's continuously releasing pollutants. These clouds of smoke, referred to as turbulent plumes, also appear in nature, for example when volcanos erupt. The pollutants are released into the atmosphere, with a different density than the surrounding air, causing them to rise before being carried by the wind. The physics of pollutant release was understood in the late 19th century [66], but it took several decades to develop models which could describe the physical processes within the plumes, such as the rate at which the pollutant is transported

(advection), the rate at which the pollutant spreads throughout the atmosphere (diffusion), the reactivity of the particles within the pollutant, or the amount of pollution emitted over time. In the second half of the 20th century, a number of mathematical models emerged, the first of which was the Gaussian dispersion model in the 1950's, which is still one of the most widely used air pollution dispersion models today [135].

The Gaussian dispersion model can model a single instantaneous release of a pollutant (puff), or a turbulent plume. It provides a mathematical framework for estimating the concentration of pollutants as they disperse from a source into the atmosphere [135]. The Gaussian plume model makes several assumptions, including that pollutant dispersion follows a normal (Gaussian) distribution in both horizontal and vertical directions; the pollutant dispersal, wind direction and wind speed are all constant; and that there are no chemical reactions within the puff or plume [132]. Despite the limitations of the Gaussian dispersion model, it is a simple and effective way of predicting pollutant concentration levels which is used by governments and industries throughout the world for many applications, such as predicting the impact of industry, modelling the impact of natural disasters, and assessing air pollution strategies.

While the Gaussian plume model remains widely used for air pollution dispersion modelling, its limitations have necessitated the development of alternative models for more complex scenarios. These models are better suited to handling varying weather conditions, the impact of ground geography, and chemical reactions of pollutants. Alternatives developed over the years include Eulerian grid models, Lagrangian models, and data-informed models. Eulerian grid models divide the atmosphere into a grid of cells and solve equations governing pollutant transport, dispersion, and transformation in each cell [69]. These models are useful for regional and global air quality assessments as they can account for chemical reactions, multiple pollutant sources, and meteorological influences. Eulerian models allow for a more comprehensive understanding of pollution behaviour at larger scales. However, they require large

amounts of data [75], such as continuous input fields describing wind direction and velocity, diffusion, boundary conditions and any other parameters, unlike Gaussian models which can use single values for many of these parameters. These fields can be found using interpolated measurements in a real domain, or can be described using a model, which would likely be derived from real data. Eulerian models are also computationally expensive, which makes them impractical for fine scale applications.

Lagrangian models track individual air parcels or particles as they move through the atmosphere, simulating their dispersion based on wind and turbulence [132]. Unlike the Gaussian plume model, which is typically derived under steady-state assumptions, Lagrangian models are naturally capable of capturing changes in wind direction and atmospheric conditions over time. They are suitable for studying pollutant dispersion over any distance, and analysing the impact of the terrain on pollutant transfer. However, they are computationally expensive in comparison to Eulerian models when applied over large spatial scales, and can be difficult to compare directly with real-world measurement data due to differences in how concentrations are represented [69].

In modern research, data-informed models can provide an alternative approach by using historical air quality data and machine learning to predict future pollution levels. These models typically do not rely explicitly on physical dispersion equations but instead use pattern recognition and predictive algorithms to estimate quantities of interest [120]. They are particularly useful for real-time air quality forecasting and situations where emissions data are incomplete or unreliable. However, their reliance on past trends makes them less effective for modelling new pollution sources or rare dispersion events, and they do not provide insight into the physical processes governing pollution transport.

Each of these models offers distinct advantages and is better suited to specific applications. While the Gaussian plume model remains a practical choice for regulatory and localised pollution modelling, more sophisticated approaches are necessary for capturing complex dispersion scenarios, secondary pollutant releases, and large-scale

atmospheric interactions. The best choice of model depends on the specific needs of the study, the available computational resources, available data, and the level of detail required in the analysis. In recent years, a hybrid approach combining one or more of these models is often used [132], as in [40,45,49], where a hybrid Lagrangian-Eulerian approach is used. Recent advances in machine learning (ML) have led to a lot of research in modelling using data-informed ML (e.g. [107, 117, 129, 140, 143]). However ML still faces problems with limited model insight, data quality, and data scarcity. These data-informed models can also be used alongside physical models to improve performance, such as in [74, 125, 147]. In this thesis, we use the Gaussian plume model as the main approach for analysing pollutant dispersal, as it is practical for localised pollution modelling.

The National Physical Laboratory (NPL) has interests in monitoring air pollution throughout the UK. Monitoring could be done by measuring, with measurements taken using large numbers of air quality monitors, but this approach could be prohibitively costly and would require a large amount of energy [93], likely contributing to the pollution problems concerning NPL such as greenhouse gas emissions. Instead, a mixture of monitoring and modelling is the preferred approach for predicting air pollution, so NPL has an interest in advancing the quality of models predicting pollutant levels. In this thesis we work to develop techniques to model and analyse the properties of a pollutant dispersal problem of interest to NPL.

1.1.1 Solving the advection-diffusion equation

The Gaussian plume model is derived from an advection-diffusion equation which describes the transport and dispersion of a quantity such as a pollutant, temperature, or chemical concentration within a moving fluid [135]. The general form of the advection-diffusion equation considered in this thesis is

$$\frac{\partial u}{\partial t} - D\nabla^2 u + a\boldsymbol{w} \cdot \nabla u = f, \tag{1.1}$$

where u is the pollutant concentration, D is the diffusion coefficient, $a\boldsymbol{w}$ is the advection velocity and direction, and f is the source term. This equation can be solved together with appropriate boundary conditions to help us understand a wide range of physical and environmental processes in the atmosphere. We can solve the advection-diffusion equation using a number of analytical or numerical methods, which will each have different levels of complexity, computational cost, and suitability, depending on the type of flow, shape of the domain, boundary conditions, and the desired level of detail or realism in the solution.

Analytical methods can be used when the governing equations and boundary conditions are relatively simple [113]. These methods give exact solutions and offer physical insight into the problem, although their use is limited to simplistic scenarios. Analytical solutions can be obtained using separation of variables or integral transform techniques, such as the Fourier or Laplace transform [96]. Analytical techniques cannot typically be used when there are irregular domains, variable parameters, or when the model is non-linear.

For more realistic scenarios, we can solve the advection-diffusion equation using numerical methods, such as those based on finite differences, finite volumes, and finite elements, each of which discretise the domain and approximate the solution. The finite difference method is simple and well suited to square or rectangular domains [68]. It approximates the derivatives of the solution using Taylor series expansions and finds concentration values at discrete grid points over time. The finite difference method may struggle with more complex boundary shapes, and the solution can become numerically unstable and unreliable, especially for advection-dominated flows.

The finite volume method is a method which is widely used in fluid dynamics [34]. It divides the domain into cells and integrates over each cell, ensuring that quantities, such as mass or concentration, entering and leaving each cell are balanced, making it well suited for modelling pollutant transport. However, the finite volume method may not be suitable when working with highly irregular domains, as it can be more

difficult to generate suitable meshes [136]. Also, it may not be ideal when more accurate solutions are required, as it typically provides less accurate, lower order approximations compared to other numerical methods [25].

The finite element method approximates the solution using weighted residuals and basis functions defined over a mesh of finite elements [38]. It is particularly effective for applications with complex domains and unstructured meshes, although it is usually more computationally expensive than finite difference and finite volume methods at comparable grid resolutions.

Our choice of method for solving the advection-diffusion equation is a trade off between realism and efficiency. For our advection-diffusion equation, we use the finite element method due to its ability to handle variable parameters and non-uniform domain discretization. Its flexibility when dealing with local source terms makes it well suited for simulating pollutant dispersion.

1.2 Uncertainty quantification

When modelling air pollution, the parameters such as advection velocity and diffusion coefficients are not likely to be constant, and may not be known exactly, although they often have a known range of historical values, or empirically estimated distributions. This uncertainty in the partial differential equation (PDE) can be modelled using uncertainty quantification techniques. Uncertainty quantification (UQ) can be used when modelling physical systems where inputs are not known, or where the model itself is an approximation of reality, to assess how uncertainty in model inputs, parameters, or assumptions affect the model outputs.

1.2.1 Sources of error

In the advection-diffusion equation (1.1), we are modelling the continuous release of a Gaussian plume in a domain with uncertain inputs. This model is going to have several sources of error if used to represent a real world problem. Firstly, the

model itself will have an error relative to reality, regardless of how many parameters we include, because it is an ideal mathematical representation of many complex physical processes, which can never fully capture the full variability, interactions, and uncertainties present in the real world [44]. This is sometimes referred to as structural or model formulation error, and it is present even if all parameters are known exactly.

Secondly, we can describe our uncertain parameters using probabilities or ranges, but these are approximations to any real world parameters affecting any air pollution model. These input uncertainties will introduce an additional source of error to our concentration solution [59]. They can generally be grouped as either aleatoric or epistemic [77]. Aleatoric uncertainty comes from randomness that is built into the system itself, such as fluctuations in wind speed or small-scale turbulence in the air, and will always be present, no matter how much we measure or how accurate our sensors become. Epistemic uncertainty arises from things we do not know yet, such as a parameter that has not been measured, an assumption we have made, or a simplification in the model. While aleatoric uncertainty is considered irreducible, epistemic uncertainty can theoretically be reduced with more data, improved experiments, or a better understanding of the system. In real world air pollution modelling, both types of uncertainty are usually present, and distinguishing between them helps clarify whether a model could be improved by gathering more information or whether the uncertainty is simply a natural variability. The distributions or ranges we assign to uncertain inputs are themselves approximations, and any assumptions made about their shape, spread, or independence can introduce a further source of uncertainty [91].

As our advection-diffusion problem is a simplified illustrative example, there is no real world equivalent input or output measurements, so none of the previously discussed errors will be present. The solution of the PDE will introduce a numerical error due to the discretisation of the domain, as the finite element method (FEM) solution is a discrete approximation to the true continuous solution [97]. The accu-

racy of this solution depends on how fine the mesh is, the basis functions used, how smooth the true solution is, how boundary conditions are applied, and how integrals are computed numerically [73]. We can use more sophisticated basis functions, or use a finer mesh, to reduce the numerical error, but it can become very computationally expensive to do so and a numerical error will still be present even if it is small. Numerical inaccuracies can also be introduced through implementation details such as solver tolerances.

The final source of error comes from UQ which introduces more approximations [44]. Most common UQ techniques, such as Monte Carlo sampling, polynomial chaos expansions, or surrogate models like Gaussian processes, rely on mathematical techniques to estimate how uncertainty affects the system. All of these techniques require simplifications, such as simplifying series expansions, using a limited number of samples, or building approximations of the full model, therefore they cannot perfectly capture the full range of possible outcomes which introduces another source of error. In this thesis the two sources of error present will be numerical error and UQ error.

1.2.2 Methods of uncertainty quantification

When using UQ to analyse our advection-diffusion problem, we can use either intrusive or non-intrusive methods. Intrusive UQ methods modify the governing equations to directly incorporate uncertainty, as in, for example, the stochastic Galerkin finite element method, where the solution is represented using a polynomial chaos expansion in terms of random variables [152]. The solution is then expressed as a function of both space and uncertainty. When the uncertainty is low-dimensional and the response is smooth, this can be a highly accurate approach. However, if there are multiple uncertain parameters, the number of terms in the expansion grows quickly. When the uncertain inputs are random fields, Karhunen–Loève expansions may be used to represent them in a finite-dimensional form before constructing a polynomial chaos expansion. These methods differ mainly in how uncertainty is represented,

but they face similar challenges. As the number of uncertain parameters increases, the resulting system grows rapidly in size, leading to high computational cost. All intrusive approaches require reformulating the governing PDEs to incorporate uncertainty, which makes them less suited to our purposes, as we wish to apply existing numerical techniques without modifying the underlying model.

To avoid modifying the underlying problem, we can use non-intrusive methods that treat the problem solution method as a black box, requiring only the ability to provide inputs and observe the resulting outputs. The most commonly used UQ method is the Monte Carlo method [17], which represents uncertain parameters by drawing random samples from a probability distribution, and evaluating the model 'black box' for each random sample. The samples of the model output can then be used to estimate statistics of interest, such as the mean or variance. The Monte Carlo method is simple to implement, but error reduction is slow, and the need to evaluate the model a large number of times can be computationally expensive. Monte Carlo methods form the baseline non-intrusive approach against which sampling and surrogate methods are often compared.

An alternative to Monte Carlo simulation is stochastic collocation [20]. Like Monte Carlo, stochastic collocation is a non-intrusive method, as it requires only repeated solutions of the deterministic PDE for prescribed values of the uncertain parameters. The method constructs a polynomial approximation of the model output using solutions of the PDE evaluated at a set of collocation points in the input parameter space. These points are typically chosen according to a tensor-product or sparse-grid quadrature rule, allowing statistics such as the mean and variance to be computed efficiently when the model output depends smoothly on the uncertain inputs. Because the governing equations are not modified, stochastic collocation treats the solver as a black box, in contrast to intrusive approaches such as stochastic Galerkin methods. As with other polynomial-based methods, the computational cost increases rapidly as the number of uncertain parameters grows, which limits their practical use to problems with relatively few uncertain inputs.

More generally, surrogate models are often used within non-intrusive UQ methods to reduce the computational cost of repeated model evaluations, where a cheaper approximation to the original model is constructed. There are several types of surrogate model we could use for our advection-diffusion problem, including physics-based or data-informed models. Physics-based surrogate models reduce the complexity of the model [78], using methods such as coarser meshes or fewer dimensions, so that the model is cheaper to evaluate. These approximations are derived from the same underlying physical model, either through simplifications of the governing equations or by using coarser discretisations, and are expected to capture similar behaviour, but they still require solving the underlying PDE so may not be as cheap as data-informed models.

Data-informed surrogates are trained using a training data set with a number of input and output samples [89]. The key difference between physics-based and data-informed surrogates is whether the surrogate is derived from the governing equations or learned purely from data. These models can be simple linear regression [100], or more complicated machine learning models such as neural networks or Gaussian process regression models [102]. Once they have been trained, they can then typically give an approximate output value very cheaply for any given input, allowing for cheap estimation of statistics of interest. Machine learning–based models can achieve high accuracy with a relatively small number of model evaluations for some problems. However, their performance depends critically on the quality and representativeness of the training data, and insufficient or biased data can result in poor approximations. Data-informed surrogates may also require extensive adjusting of parameters which govern the learning process but are not learned from the data, known as hyper-parameter tuning [2].

Recent work in UQ has focused on improving surrogate models using ML [8,131], for problems involving high-dimensional uncertainty or expensive model evaluations such as in [7, 64, 111, 130]. Bayesian methods, which express the uncertainty as a probability distribution over model parameters and predictions, have also been

applied in a wide range of fields, such as in [53, 61, 82, 86, 144]. In [4, 11, 39, 47], adaptive learning and adaptive sampling methods are used to select sample points where the uncertainty is highest, allowing for more efficient improvement of surrogate model accuracy for better UQ.

For UQ of our advection-diffusion problem, we will use the Monte Carlo method as it is non-intrusive, simple to implement and can be used to estimate a wide range of statistics.

1.3 Variance reduction

When using the Monte Carlo method to estimate a quantity of interest (QoI), there will be a sampling error, which will be dependent on the bias, number of samples, and the variance of our problem parameters. If we wish to minimise this error for a given budget, we can reduce the variance associated with the estimate of the QoI output using variance reduction techniques. Variance reduction techniques can be very effective and reduce the variance of Monte Carlo estimate many thousands-fold, or they can reduce the variance by only a small percentage [98]. There is no set rule on how large the improvement in efficiency must be in order for variance reduction to be worthwhile, for example a 10% reduction in time to calculate an approximation that keeps thousands of CPUs busy for months would certainly be worthwhile. However using variance reduction with a 60-fold improvement that turns a one minute wait into a one second wait would likely not be worth the time spent programming. When using variance reduction we must decide for our problem whether we wish to reduce cost and what the cost is that we wish to save, e.g., time to run, sample size etc, or if reducing the sampling error of our solution is more important.

There are many variance reduction techniques available when using Monte Carlo [35, 98]. One of the simplest is control variates, which uses a correlated variable with a known expected value (known as a control variate) to reduce the variance of an estimator. Control variates can be very effective when the control variate and original variable are highly correlated, and the method is highly flexible and can be

combined with other variance reduction techniques. However, the method requires us to know the expected value of the control variate, and if the control variate is not correlated enough then it can have limited or no effect.

Another common example is antithetic sampling, where a negatively correlated value is taken. For example when sampling inputs x from a normal distribution for each function evaluation $f(x)$, we also evaluate $f(-x)$, and the average of these two function values often has lower variance than a single evaluation [16]. This method is easy to implement as it does not require changes to our underlying model, and it is unbiased. However, it is not simple to implement if the distribution of inputs is multi-variate or complicated, and it may not reduce variance if the function is non-monotonic.

Stratified sampling can be used by dividing the input range into a number of strata, then sampling number of inputs from each stratum and combining the results for each to estimate the expected value $\mathbb{E}(f(x))$ [16]. If the samples are distributed with more important strata over-sampled and less important strata under-sampled, then the variance can be reduced for a given sample size. This ensures that each part of the domain is proportionally represented, however, it becomes exponentially harder to stratify input domains in higher dimensions. Stratified sampling also requires additional knowledge of the underlying function, such as which strata have higher variance.

Conditioning is a variance reduction technique used in Monte Carlo simulation, particularly when a function $f(x, y)$ depends on two or more sources of randomness [16]. Instead of sampling both x and y, we compute the conditional expectation $\mathbb{E}[f(x, y) \mid y]$, and then average over y. This can significantly reduce variance when the conditional expectation can be computed analytically or efficiently. However, calculating the conditional expectation may be computationally expensive, and in some cases may offer little benefit if the variance structure is not favourable.

1.3.1 Multi-fidelity methods

A common approach to variance reduction is the use of multi-fidelity methods, which uses models with different accuracies and levels of detail [105]. High-fidelity models $f_H(x)$, the most accurate models associated with a given problem, are often computationally expensive, particularly when repeated evaluations are required. Low-fidelity models, $f_L(x)$, are less costly to evaluate but typically less accurate. Multi-fidelity methods aim to reduce computational cost by combining models of varying fidelity, while preserving the accuracy of high-fidelity predictions.

Multi-fidelity model management strategies define how samples are allocated between high and low-fidelity models, and how outputs from models of differing fidelity are combined. The three primary approaches to model management are adaptation, fusion, and filtering [105].

Adaptation strategies aim to improve the low-fidelity model using outputs from the high-fidelity model, so the low-fidelity model can then be used as a control variate or a surrogate model. This can be done using an additive correction model, where a discrepancy function $\delta(x) \approx f_H(x) - f_L(x)$ is found, and UQ can be performed using the corrected model $f_L(x) + \delta(x)$ [94]. The correction model can be trained using data-informed models such as Gaussian processes regression or neural networks, or derived analytically using Taylor series expansions. Similarly, multiplicative correction models can be constructed, where $\delta(x) \approx \frac{f_H(x)}{f_L(x)}$, and UQ is performed on $\delta(x) f_L(x)$ [105].

Other adaptation strategies include parameter tuning, where the low-fidelity model contains adjustable parameters that are optimised to minimise the difference with the high-fidelity outputs [90]. Another technique is hybrid model replacement [121], in which specific physical terms of the low-fidelity model are replaced with surrogates trained on high-fidelity data, resulting in a partially corrected model. Another method is the use of gradient-enhanced correction models, which incorporate not only function evaluations but also derivative information from both models to construct more accurate approximations of the difference [15].

Fusion strategies take evaluations from both low and high-fidelity models, and combine their outputs to produce more accurate predictions than either model alone [105]. Rather than modifying the low-fidelity model directly, fusion methods integrate outputs from multiple fidelity levels to construct a single predictive model. One common approach is the use of cokriging [51], a multi-fidelity extension of Gaussian process regression, where the high-fidelity output is modelled as $f_H(x) = \rho f_L(x) + \delta(x)$, where ρ is a scalar or learned function, and $\delta(x)$ is the correction modelled as a Gaussian process. This allows the surrogate to be trained on low-fidelity data while being corrected by sparse but accurate high-fidelity evaluations. Other fusion strategies include the Bayesian model fusion [124], where both models are treated as noisy observations of an underlying process, and a posterior distribution is inferred from all available data. By integrating sources of information, fusion methods reduce the variance of predictive estimates and enable efficient uncertainty quantification.

Filtering strategies use the low-fidelity model to guide when and how often the high-fidelity model is used. Unlike adaptation and fusion, which focus on modifying or combining models, filtering methods prioritise sample allocation to minimise variance within a fixed computational budget [105]. A commonly used example is the multi-fidelity Monte Carlo (MFMC), introduced in [104], which can use any number of correlated approximations to accelerate Monte Carlo sampling, and has been widely applied in a range of fields [3, 5, 46, 110]. Additionally, adaptive sampling strategies fall under filtering, where the low-fidelity model is used to identify regions of high variance or importance in the input space, and high-fidelity evaluations are concentrated in these areas [22]. By using the low-fidelity model to control the use of expensive simulations, filtering methods achieve significant variance reduction without sacrificing estimator accuracy.

Multilevel methods are a specific example of multi-fidelity methods which are relevant to our advection-diffusion model. Multilevel approaches use multiple evaluations of the same underlying model or PDE, with differing levels of discretisations,

rather than combining models of different fidelities [50]. Examples of multilevel methods include the multilevel Monte Carlo (MLMC) method, which approximates the expected value by summing differences between model outputs computed on a hierarchy of discretisations with increasing resolution, allocating more samples to cheaper, coarse models and fewer to expensive, fine ones. The effectiveness can be improved further using methods such as multilevel control variates and multilevel quasi-Monte Carlo. Although multilevel methods are not strictly multi-fidelity as they do not mix different model types, they share similar objectives and are well suited to PDE-based solvers, particularly when mesh refinement plays a role in model accuracy and cost.

In modern research, there are a number of variance reduction techniques being widely used. Multi-fidelity methods are used to improve UQ, such as in [5, 54, 87, 142]. Control variates with machine learning approximations are popular [83,85,108]. Quasi-Monte Carlo methods replace random sampling with samples determined using a sequence, such as a Sobol, Halton or Faure sequence [24]. These sequences can be used on their own [62, 88, 99], or combined with Monte Carlo [65, 101, 106, 145] to improve error estimation (referred to as randomised quasi-Monte Carlo). Adaptive importance sampling (AIS) improves the distribution used in importance sampling by learning from previously drawn samples [119]. By concentrating sampling effort in regions where the integrand contributes most to the estimate, AIS increases efficiency and accuracy [55, 63, 151].

For our advection-diffusion problem, we choose to focus on control variates and the MFMC method, as they are unbiased, simple to implement, do not require altering sampling distributions, and are very generally applicable.

1.4 Sensitivity analysis

While UQ describes how uncertain model outputs such as the concentration in our advection–diffusion model are overall, sensitivity analysis allows us to examine how much individual inputs contribute to that uncertainty. Sensitivity analysis examines

how uncertainty in the output of a model with multiple inputs, $y = f(x_1, x_2, \ldots, x_m)$ say, is influenced by the uncertainty in each of the model inputs. This is particularly relevant to our advection-diffusion problem, as identifying the factors that contribute most to uncertainties surrounding pollutant dispersion is vital in real-world environmental modelling. Sensitivity analysis is also useful for detecting interactions between inputs, where a variable may have little effect on its own, but significantly amplifies the influence of other variables in combination.

There are several methods of sensitivity analysis we can use. The most basic of these is one-at-a-time (OAT) analysis [139]. This method works by choosing a single value for each x_i, such as a mean or base case, and evaluating f. Next, one x_i is changed and f is evaluated again, and the difference between the evaluations is taken. This method is simple to understand and implement, and can be useful for initial exploration of a model, but it gives no insight into model interactions, and it only determines the sensitivity at that single point in the input domain, not across the entire domain.

One of the most popular methods of sensitivity analysis is the use of Sobol indices. First introduced in the English language in [133], Sobol indices express the variance of the inputs of a model with multiple inputs (factors) as a proportion of the overall variance. They are a global sensitivity measure, as they evaluate a variable's influence over the entire input space, not just at a single point. Sobol indices are also applicable to any model with an input and output, and they can capture interactions between inputs. However, they are computationally expensive, and can become excessively so when there are too many input terms.

Another common method of sensitivity analysis is the Morris Method, which samples a single random value for each input [122]. One input value is then changed at a time, and the model is evaluated after each variable change, to examine each input's effect on the output. The process is repeated a number of times, and for each input it returns a mean absolute effect value which determines how influential that variable is, and standard deviation of effects which shows how non-linear or

interactive the variable is. This method is computationally cheaper than Sobol indices while also being a global measure which detects interaction. It is also easy to implement, however it does not return a precise measure of factor importance, only a rough importance ranking, and it does not quantify interactions well.

Fourier Amplitude Sensitivity Testing (FAST) is a global, variance-based sensitivity analysis method [122]. It maps each input variable onto a unique frequency using a sine function, samples the model along a trajectory through the input space, and then applies a Fourier transform to the output to estimate how much variance each input contributes. FAST typically requires fewer model evaluations than Sobol indices to compute single variable sensitivities, but it is more challenging to implement and its results can be less intuitive. While standard FAST does not capture input interactions, these can be quantified using the extended FAST (eFAST) method [141].

Moment-independent methods assess how input variables influence the distribution of the model output [27], unlike moment-based methods such as Sobol indices which only examine the influence of an output's moments, such as the mean or variance. There are several widely used moment-independent methods, such as Borgonovo's sensitivity indices [80, 81, 138], and probabilistic analysis of variance [21, 134, 150]. Variogram Analysis of Response Surfaces (VARS) is a more recently developed moment-independent sensitivity analysis method that examines how the output reacts to changes from structured input variations, for each input independently [116]. VARS enables the analysis of how the sensitivity changes depending on scale and direction, and has been applied in a number of problems [1, 14, 58, 123]. Modern research on sensitivity analysis uses a number of methods. Sobol indices are still widely applied due to their simplicity and flexibility [10, 18, 79, 146], and Sobol indices are frequently used with machine learning based surrogates for efficiency [13, 95, 148, 149]. We choose to focus on Sobol indices for our advection-diffusion problem as they give global sensitivity and capture interactions between inputs. They are also flexible and simple to compute for any black-box problem we

may wish, and do not require changing the underlying model.

1.5 Thesis outline

In Chapter 2 we introduce a pollutant dispersal model problem on a 2D domain, representing a two-dimensional horizontal plane at a fixed height above the ground. We examine the pollutant dispersal with fixed parameters and discuss discretising the domain using the Galerkin finite element method. We examine oscillations in the solution that result when using the Galerkin method for advection-dominated cases, and introduce the Streamline Upwind Petrov-Galerkin method to ameliorate this.

In Chapter 3 we discuss the Monte Carlo method and introduce the mean-squared error as a measure of error. We use the well known Ishigami function [67] as a simple problem to test the Monte Carlo method, then apply the Monte Carlo method to find the mean of our advection-diffusion problem with uncertain advection velocity and diffusion coefficients. We compare the mean of the computed concentration when using Galerkin and Streamline Upwind Petrov-Galerkin discretisations, when the model is advection dominated, and when it is not.

We introduce confidence intervals using bootstrapping to estimate the mean of the Ishigami function with some uncertainty, and examine the coverage of the confidence intervals using known means. We then use bootstrapping to find the mean concentration of the advection-diffusion problem within confidence intervals.

In Chapter 4 we examine multi-fidelity methods of variance reduction. We consider control variates with known and estimated means, and we derive their mean-squared errors for the Ishigami function. We propose a new method of adding a budget control to the control variates with estimated means. We compare the theoretical mean-squared errors of the control variates with estimated means and budget control, and a two-level multi-fidelity Monte Carlo method and find the two methods to be equivalent. We estimate confidence intervals using bootstrapping and control variates with estimated mean and budget control, then test the confidence interval

coverage of this method using the Ishigami function. We use low-fidelity approximations to our advection-diffusion problem based on coarse-grid finite element solutions, and data-informed approximations such as linear regression, Gaussian process regression, and neural networks for regression. We compare the approximation outputs to the advection-diffusion model output, then use the approximations to find narrower confidence intervals for the mean of our advection-diffusion solution.

In Chapter 5 we introduce Sobol indices, and examine Monte Carlo methods of estimating the Sobol indices, which we again test on the Ishigami function. We propose a multi-fidelity approach for estimating Sobol indices using control variates with estimated means and budget control, and confirm its accuracy using the Ishigami function. We then discuss methods for calculating confidence intervals for Sobol indices using bootstrapping with Monte Carlo, and control variates with estimated means and budget control. We confirm the robustness of our control variates approach using the Ishigami function, then apply these methods to find the Sobol indices of parameters in our advection-diffusion problem using the approximations from Chapter 4.

In Chapter 6 we formulate a dimensional eddy-diffusion model with three sources, introduce additional uncertainties in wind speed, diffusion, source strengths, and wind direction, and assign them Weibull, truncated log-normal, gamma, and von Mises distributions respectively. We find reasonable distribution parameters using real world data. Then, we vary wind direction via a rotation of coordinates to avoid repeated matrix assembly, and assess both coarse-grid and data-informed surrogates (linear regression, Gaussian process regression, and a neural network). We then apply the CVB approach with bootstrapping to estimate first- and total-order Sobol indices for the new model, finding reduced surrogate correlations relative to Chapter 4 but small confidence intervals when using the neural network.

In Chapter 7 we summarise the key findings of this thesis and suggest potential avenues for future research.

The novel contributions of this thesis are as follows. First, a new control vari-

ate with budget control (CVB) framework is proposed and analysed for estimating means and confidence intervals using bootstrapping. Second, methods for estimating means and confidence intervals are compared for advection–diffusion equations with uncertain parameters. Third, the CVB framework is extended to Sobol index sensitivity analysis, including the construction of confidence intervals. Fourth, a novel air pollutant dispersion model with multiple sources and uncertain parameters is formulated. Finally, a comprehensive sensitivity analysis of this air pollutant model is performed using the proposed CVB-based methodology.

Chapter 2

Model problem

2.1 Advection-diffusion model in 2D

The non-dimensional physical model we will study in this thesis represents a pollutant released at a single point $(x, y) = (x_s, y_s)$ in a 2D domain $\Omega = [0, 8] \times [0, 1]$, representing a two-dimensional horizontal plane at a fixed height above the ground. We assume the flow characterised by $\boldsymbol{w}$ is incompressible, that is $\nabla \cdot \boldsymbol{w} = 0$, and $\partial\Omega$ is the boundary of the domain.

Homogeneous Dirichlet boundary conditions, $u = 0$, are imposed on $\partial\Omega$. For the inflow boundary and the boundaries at $y = 0$ and $y = 1$, this is physically reasonable, as the pollutant concentration is negligible outside the plume. At the outflow boundary ($x = 8$), however, enforcing $u = 0$ can artificially force the solution to decay to zero near the boundary when the problem is advection-dominated. Alternative outflow boundary conditions, such as zero diffusive flux $\partial u/\partial x = 0$ or zero total flux $au - D\,\partial u/\partial x = 0$, could be used to avoid this behaviour. However, for simplicity and consistency across all numerical experiments, homogeneous Dirichlet boundary conditions are used throughout. In all cases, the outflow boundary is located sufficiently far downstream of the plume that this choice does not affect the solution. The pollutant concentration, $u(x, y)$, is modelled by the steady-state advection-diffusion equation

$$-D\nabla^2 u + a\boldsymbol{w}\cdot\nabla u = f \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega, \tag{2.1}$$

where the diffusion constant D and the wind speed a are uncertain positive parameters, $\boldsymbol{w} = [w_x, w_y] \in \mathbb{R}^2$ represents the direction of the advective wind, and $|\boldsymbol{w}| = 1$. The source of the pollutant will be represented by $f(x, y)$ where we set f as the Gaussian pulse

$$f(x,y) = \frac{1}{2\pi S^2}\exp\left(-\frac{(x-x_s)^2+(y-y_s)^2}{2S^2}\right), \tag{2.2}$$

where S is the spread of the pollutant at release. As the pollutant is released at a single point, in theory it would ideally be modelled as a function that has a non-zero value at (x_s, y_s) and 0 elsewhere. In (2.2) as $|S| \to 0$ the Gaussian pulse will represent a Dirac delta function. We fix $S = 0.1$ throughout this thesis.

2.2 Galerkin finite element method

In this section, we describe the numerical model which will be the focus of our study. Equation (2.1) can be discretised by the Galerkin finite element method, however before that we must define some vector spaces. The $L_2(\Omega)$ vector space is defined as the space of functions that are square integrable in the sense of Lebesgue [38], that is

$$L_2(\Omega) := \left\{ u : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega u^2 < \infty \right\}.$$

The Sobolev space $H^1(\Omega)$ is given by

$$H^1(\Omega) := \left\{ v : \Omega \to \mathbb{R} \;\middle|\; v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L_2(\Omega) \right\}.$$

We will also use the restriction of this space to functions which are zero on the boundary [38]. This space is defined by

$$H_0^1(\Omega) := \{v \in H^1(\Omega) \mid v = 0 \text{ on } \partial\Omega\}.$$

2.2.1 Constructing the weak form

To construct the weak form, we multiply equation (2.1) by a test function $v \in H_0^1(\Omega)$ and integrate over the domain to obtain

$$-D\int_\Omega (\nabla^2 u)v \, d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)v \, d\Omega = \int_\Omega fv \, d\Omega. \tag{2.3}$$

Using the identity

$$(\nabla^2 u)v = -\nabla u\cdot\nabla v + \nabla\cdot(v\nabla u),$$

we can write (2.3) as

$$D\left[\int_\Omega \nabla u\cdot\nabla v \, d\Omega - \int_\Omega \nabla\cdot(v\nabla u)\, d\Omega\right] + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)v \, d\Omega = \int_\Omega fv \, d\Omega.$$

From the divergence theorem [38], as $v \in H_0^1$, we have

$$\int_\Omega \nabla\cdot(v\nabla u)\, d\Omega = \int_{\partial\Omega} v(\nabla u\cdot\boldsymbol{n})\, dl = 0.$$

The final weak formulation is: find $u \in H_0^1$ such that

$$D\int_\Omega \nabla u\cdot\nabla v \, d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)v \, d\Omega = \int_\Omega fv \, d\Omega \quad \forall\, v \in H_0^1.$$

A discrete weak formulation can now be defined by choosing a basis for a finite-dimensional test space $S_0^h \subset H_0^1$ consisting of continuous, piecewise linear functions. We then look for $u_h \in S_0^h$ such that

$$D\int_\Omega \nabla u_h\cdot\nabla v_h \, d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u_h)v_h \, d\Omega = \int_\Omega fv_h \, d\Omega \quad \forall v_h \in S_0^h. \tag{2.4}$$

Now assume that $S_0^h \subset H_0^1$ is a finite n-dimensional vector space of test functions for which $\{\phi_1, \phi_2, ..., \phi_n\}$ is a convenient basis. The finite element approximation $u_h \in S_0^h$ is then uniquely associated with the vector $\mathbf{u} = (u_1, u_2, ..., u_n)^T$ of real coefficients in the expansion

$$u_h = \sum_{j=1}^{n} u_j \phi_j(x, y).$$

For a Galerkin approximation, we use this to substitute for u_h in (2.4) and use the n basis functions as test functions. This leads to the problem: find $u_j, j = 1, \ldots, n$ such that

$$\sum_{j=1}^{n} u_j \left[D \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \, d\Omega + \int_\Omega (a\boldsymbol{w} \cdot \nabla\phi_j)\phi_i \, d\Omega \right] = \int_\Omega f\phi_i \, d\Omega$$

for $i = 1, \ldots, n$. This can be written in matrix form as the linear system of equations

$$[DK + aC]\mathbf{u} = \mathbf{f} \tag{2.5}$$

where K, C and $\mathbf{f}$ have entries

$$K_{i,j} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \, d\Omega, \qquad C_{i,j} = \int_\Omega (\boldsymbol{w} \cdot \nabla\phi_j)\phi_i \, d\Omega,$$

and

$$f_i = \int_\Omega f\phi_i \, d\Omega$$

for $i, j = 1, \ldots, n$. The matrix K, usually referred to as the stiffness matrix, is symmetric and positive definite and the matrix C is referred to as the advection matrix.

2.2.2 Discretising the domain

We now need to discretise the domain. As Ω is polygonal, it can be exactly covered by a mesh of triangular elements. To construct the grid we require the (x, y) coordinates of each node, a global numbering $k = 1, \ldots, N^*$ of all nodes on the grid, a numbering $E = 1, \ldots, n$ of all the elements, and a local numbering $i = 1, 2, 3$ of the nodes on each triangular element.

We partition the grid with $M + 1$ nodes on the x-axis and $N + 1$ nodes on the y-axis so we will have a total of N^* nodes where $N^* = (M + 1)(N + 1)$. We then create n uniformly sized triangular elements where $n = 2MN$. The stiffness matrix K and advection matrix C will have dimension $N^* \times N^*$ and the source vector $\mathbf{f}$ will have dimension $N^* \times 1$. Our chosen domain is a rectangular $[0, 8] \times [0, 1]$ domain, and thus if we divide the domain into uniform sized elements then $M = 8N$ and the grid will have $(8N + 1)(N + 1)$ nodes and $16N^2$ elements.

Note that increasing the number of elements in the grid will increase the computational cost of solving the finite element problem significantly but will lead to a more accurate solution.

Below we will show how the local stiffness matrix, advection matrix and source vector, K_E, C_E and $\mathbf{f}_E$, associated with each element can be calculated. These arrays are calculated on each element and can then be assembled to populate their global equivalents, K, C and $\mathbf{f}$. Focusing on the stiffness matrix, each entry of K_E can be mapped to a matrix K_E^* which has the same dimension as K, where $K^*_{E_{k,l}} = K_{E_{i,j}}$ for i, j (the local node numbering of the element) and k, l (the corresponding global node numbering of the element), and all other entries of K_E^* are zero. The global stiffness matrix can then be calculated as the sum of all K_E^* such that

$$K = \sum_{E=1}^{n} K_E^*.$$

In the same manner, the entries of the global advection matrix C and the global source vector $\mathbf{f}$ can be calculated by a sum of their local counterparts constructed

on each individual element.

We now describe the computation of the element stiffness matrix. Suppose we have an element E with local nodes 1, 2 and 3 with (xy) co-ordinates $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and a reference triangle T with local nodes 1, 2 and 3 with (ξ, η) co-ordinates $(0,0), (1,0), (1,1)$ making T a right-angled triangle. We define linear element basis functions

$$\phi_1(\xi, \eta) = 1 - \xi,$$
$$\phi_2(\xi, \eta) = \xi - \eta,$$
$$\phi_3(\xi, \eta) = \eta$$

on the reference element T then map to any general triangle with vertex coordinates (x_v, y_v), $v = 1, 2, 3$ by the change of variables

$$x(\xi, \eta) = \sum_{v=1}^{3} x_v \phi_v(\xi, \eta), \quad y(\xi, \eta) = \sum_{v=1}^{3} y_v \phi_v(\xi, \eta).$$

An example of E and T can be seen in Figure 2.1.

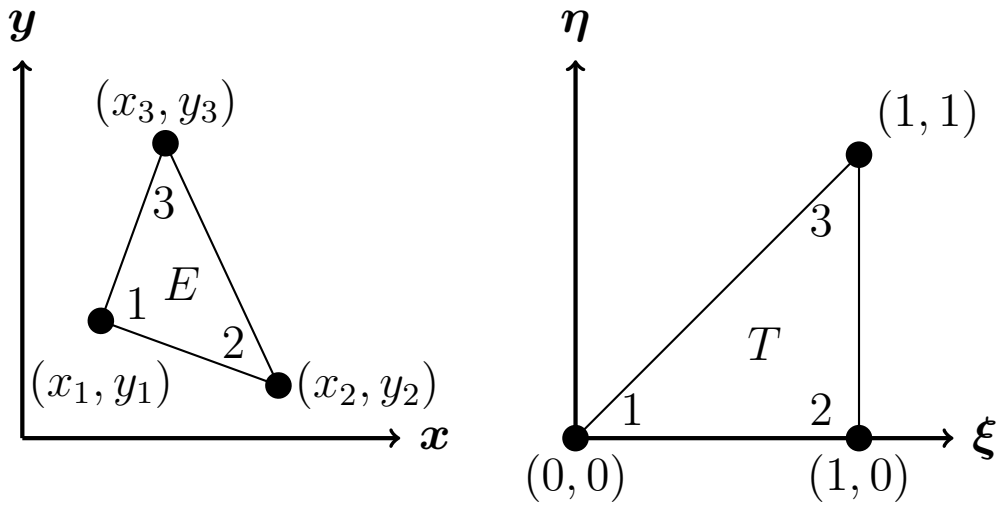


Figure 2.1: A triangular element, E, and the reference element, T.

The local stiffness matrix on an element can be calculated as

$$K_{E_{ij}} = \int_E \nabla\phi_j \cdot \nabla\phi_i \, dE = \int_E \left[\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + \frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y} \right] \, dxdy, \tag{2.6}$$

where $i, j = 1, 2, 3$. To transform this into an integral over T, we need the determinant of the Jacobian of the transformation, which is

$$J = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = (x_2 - x_1)(y_3 - y_2) - (x_3 - x_2)(y_2 - y_1).$$

Now consider the terms in the integrand in (2.6). Using the Chain Rule we have

$$\frac{\partial\phi_i}{\partial x} = \frac{\partial\phi_i}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\phi_i}{\partial\eta}\frac{\partial\eta}{\partial x}, \qquad \frac{\partial\phi_i}{\partial y} = \frac{\partial\phi_i}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\phi_i}{\partial\eta}\frac{\partial\eta}{\partial y}.$$

To find the derivatives of ξ, η with respect to x, y, we use the relationship

$$\begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta} \\ \frac{\partial y}{\partial\xi} & \frac{\partial y}{\partial\eta} \end{bmatrix}^{-1} = \frac{1}{J}\begin{bmatrix} \frac{\partial y}{\partial\eta} & -\frac{\partial x}{\partial\eta} \\ -\frac{\partial y}{\partial\xi} & \frac{\partial x}{\partial\xi} \end{bmatrix}$$

to obtain

$$\frac{\partial\xi}{\partial x} = \frac{1}{J}\frac{\partial y}{\partial\eta}, \qquad \frac{\partial\xi}{\partial y} = -\frac{1}{J}\frac{\partial x}{\partial\eta}, \qquad \frac{\partial\eta}{\partial x} = -\frac{1}{J}\frac{\partial y}{\partial\xi}, \qquad \frac{\partial\eta}{\partial y} = \frac{1}{J}\frac{\partial x}{\partial\xi}.$$

These can be used to write each integral in (2.6) in terms of ξ, η and we can obtain an entry of the element stiffness matrix using the calculation

$$\begin{aligned} K_{E_{i,j}} = \frac{1}{J}\int_0^1\int_0^\xi & \left(\frac{\partial\phi_i}{\partial\xi}\frac{\partial y}{\partial\eta} - \frac{\partial\phi_i}{\partial\eta}\frac{\partial y}{\partial\xi} \right)\left(\frac{\partial\phi_j}{\partial\xi}\frac{\partial y}{\partial\eta} - \frac{\partial\phi_j}{\partial\eta}\frac{\partial y}{\partial\xi} \right) \\ & + \left(-\frac{\partial\phi_i}{\partial\xi}\frac{\partial x}{\partial\eta} + \frac{\partial\phi_i}{\partial\eta}\frac{\partial x}{\partial\xi} \right)\left(-\frac{\partial\phi_j}{\partial\xi}\frac{\partial x}{\partial\eta} + \frac{\partial\phi_j}{\partial\eta}\frac{\partial x}{\partial\xi} \right) \, d\xi d\eta. \end{aligned}$$

For the advection matrix C, the process is similar. We have

$$C_{E_{i,j}} = \int_E a\mathbf{w} \cdot (\nabla\phi_j)\phi_i \, dE = a\int_E \left[w_x\phi_i\frac{\partial\phi_j}{\partial x} + w_y\phi_i\frac{\partial\phi_j}{\partial y} \right] \, dxdy \tag{2.7}$$

where $i, j = 1, 2, 3$. Transforming to T, this becomes

$$\begin{aligned} C_{E_{i,j}} &= a \int_T \left[w_x \phi_i \left(\frac{\partial \phi_j}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \phi_j}{\partial \eta} \frac{\partial \eta}{\partial x} \right) + w_y \phi_i \left(\frac{\partial \phi_j}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \phi_j}{\partial \eta} \frac{\partial \eta}{\partial y} \right) \right] J d\xi d\eta \\ &= a \int_T \left[w_x \phi_i \frac{1}{J} \left(\frac{\partial \phi_j}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial \phi_j}{\partial \eta} \frac{\partial y}{\partial \xi} \right) + w_y \phi_i \frac{1}{J} \left(-\frac{\partial \phi_j}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial \phi_j}{\partial \eta} \frac{\partial x}{\partial \xi} \right) \right] J \, d\xi d\eta \\ &= \int_0^1 \int_0^\xi \left[a w_x \phi_i \left(\frac{\partial \phi_j}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial \phi_j}{\partial \eta} \frac{\partial y}{\partial \xi} \right) + a w_y \phi_i \left(-\frac{\partial \phi_j}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial \phi_j}{\partial \eta} \frac{\partial x}{\partial \xi} \right) \right] d\eta d\xi. \end{aligned}$$

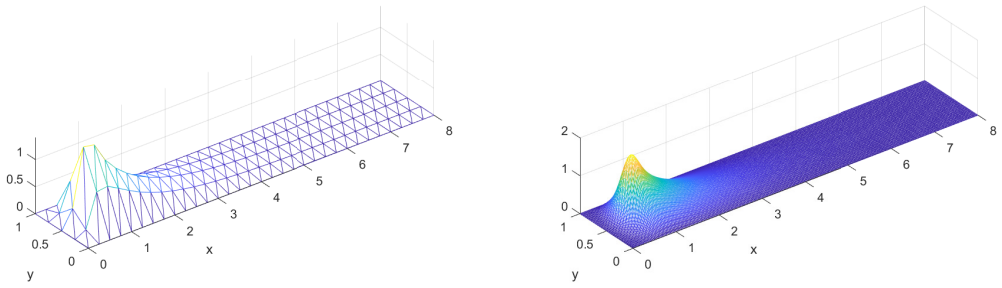
To construct the entries of the element source vectors we need to evaluate the integrals

$$f_{E_i} = \int_E f \phi_i \, dE$$

where $i = 1, 2, 3$. This can be rewritten using the same transformation as before to get

$$\int_E f \phi_i dE = J \int_0^1 \int_0^\xi f \phi_i \, d\eta d\xi \tag{2.8}$$

for $i = 1, 2, 3$. In Figure 2.2 we fix the wind direction as $\boldsymbol{w} = [1, 0]$, the source size S is set to 0.1, the advection strength is set to $a = 1$, and $D = 0.1$. We can see two sample Galerkin approximations to the solution of PDE (2.1) with high and low N. The higher value of N results in a finer mesh while a lower value of N results in a



(a) Plot of the concentration with $N = 4$, $a = 1$, and $D = 0.1$.

(b) Plot of the concentration with $N = 32$, $a = 1$, and $D = 0.1$.

Figure 2.2: Comparison of the pollutant concentration with low and high values of N with consistent values of a and D.

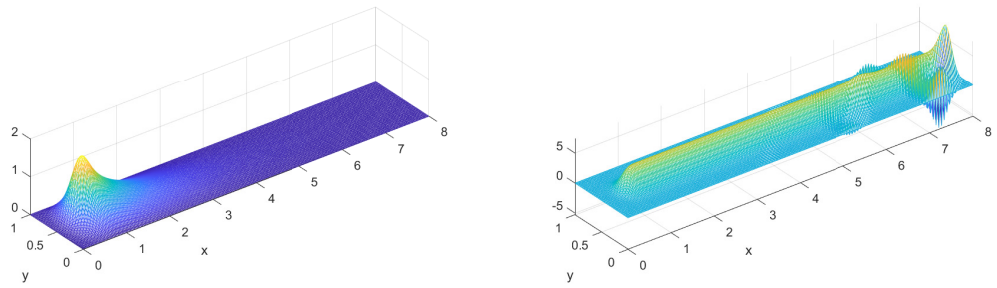
coarser mesh.

2.3 The Streamline Upwind Petrov-Galerkin method

Galerkin finite element solutions of (2.1) are known to suffer from spurious oscillations if the mesh-Péclet number, as defined by

$$P_h = \frac{h|a\boldsymbol{w}|}{2D}$$

where $h = \frac{1}{N}$ is greater than one [38]. An example of oscillations in the solution can be seen in Figure 2.3. Here, the diffusivity D is reduced purely as a numerical device to increase the mesh–Péclet number while keeping the mesh fixed. The resulting value is therefore not intended to be physically realistic, but is chosen to deliberately induce an advection-dominated regime and highlight numerical instabilities. With a higher value of $D = 0.1$ the mesh-Péclet number is less than one and the plot of the solution is smooth, whereas decreasing D to 0.0001 results in a mesh-Péclet number much higher than one, and oscillations in the pollutant concentration are clearly visible as the boundary is approached. To alleviate this, it is usual to add some



(a) Plot of the concentration with $N = 32$, $a = 1$, $D = 0.1$, and $P_h = 0.156$.

(b) Plot of the concentration with $N = 32$, $a = 1$, $D = 0.0001$ and $P_h = 156$.

Figure 2.3: Comparison of the pollutant concentration with low and high mesh-Péclet numbers. These numerical approximations are obtained using a Galerkin approximation.

diffusion in the direction of the wind. Here we use a Streamline Upwind Petrov-Galerkin (SUPG) method, where a test space is defined which is different from the trial space. This contrasts with the Galerkin method, where the test and trial spaces

are the same. For the convection-diffusion problem, we take the test space to be the space spanned by functions of the form

$$v + \alpha a \boldsymbol{w} \cdot \nabla v$$

where $v \in H_0^1$ and $\alpha > 0$ is a constant stabilisation parameter. Multiplying (2.1) by these new test functions and integrating over the domain gives

$$-D\int_\Omega (\nabla^2 u)(v+\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)(v+\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega = \int_\Omega f(v+\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega$$

so

$$\begin{aligned} -D\int_\Omega (\nabla^2 u)v\,d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)v\,d\Omega - D\int_\Omega (\nabla^2 u)(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega \\ + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega = \int_\Omega fv + \int_\Omega f(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega. \end{aligned}$$

The first two LHS terms and the first RHS term here are exactly the same as in the Galerkin case so can be treated in the same way to obtain

$$\begin{aligned} D\int_\Omega \nabla u\cdot\nabla v\,d\Omega + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)v\,d\Omega - D\int_\Omega (\nabla^2 u)(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega \\ + \int_\Omega (a\boldsymbol{w}\cdot\nabla u)(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega = \int_\Omega fv\,d\Omega + \int_\Omega f(\alpha a\boldsymbol{w}\cdot\nabla v)\,d\Omega. \end{aligned} \tag{2.9}$$

We now need a discrete version of this, so we replace u and v by u_h and $v_h \in S_0^h$. The Galerkin-like terms are the same as before, but we now need to consider the extra terms in (2.9), namely,

$$-D\int_\Omega (\nabla^2 u_h)(\alpha a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega, \quad \int_\Omega (a\boldsymbol{w}\cdot\nabla u_h)(\alpha a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega \text{ and } \int_\Omega f(\alpha a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega.$$

In the first term, u_h might not be smooth enough to have a second derivative. However, it can be shown (see e.g [38]) that this integral can be replaced by a sum over

all of the elements, so we can use

$$-D\int_{\Omega}(\nabla^2 u_h)(\alpha a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega \equiv -\alpha D\sum_{E=1}^{n}\int_{E}(\nabla^2 u_h)(a\boldsymbol{w}\cdot\nabla v_h)\,dE.$$

As we are using piecewise linear finite elements, this will be zero due to the second derivative. The full SUPG formulation is therefore

$$D\int_{\Omega}\nabla u_h\cdot\nabla v_h\,d\Omega+\int_{\Omega}(a\boldsymbol{w}\cdot\nabla u_h)v_h\,d\Omega+\alpha\int_{\Omega}(a\boldsymbol{w}\cdot\nabla u_h)(a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega$$
$$=\int_{\Omega}fv_h\,d\Omega+\alpha\int_{\Omega}f(a\boldsymbol{w}\cdot\nabla v_h)\,d\Omega.$$

Approximating the solution by

$$u_h=\sum_{j=1}^{n}u_j\phi_j,$$

and using SUPG test functions of the form

$$v_h=\phi_i+(a\boldsymbol{w}\cdot\nabla\phi_i),$$

the weak formulation can be written as the following system of linear equations:

$$\sum_{j=1}^{n}u_j D\int_{\Omega}\nabla\phi_j\cdot\nabla\phi_i\,d\Omega+\int_{\Omega}(a\boldsymbol{w}\cdot\nabla\phi_j)\phi_i\,d\Omega+\alpha\int_{\Omega}(a\boldsymbol{w}\cdot\nabla\phi_j)(a\boldsymbol{w}\cdot\nabla\phi_i)\,d\Omega$$
$$=\int_{\Omega}\phi_i f d\Omega+\alpha\int_{\Omega}f(a\boldsymbol{w}\cdot\nabla\phi_i)\,d\Omega,$$

for $i=1,...,n$. This can be written in matrix form as the linear system of the form

$$[DK+aC+\alpha U]\mathbf{u}=\mathbf{f}+\alpha\mathbf{f}^{\mathbf{U}}. \tag{2.10}$$

The matrices K and C and the vector $\mathbf{f}$ are exactly as in (2.5). The entries of the additional upwinding matrix U and right-hand side component $\mathbf{f^U}$ are given by

$$U_{i,j} = \int_\Omega (\boldsymbol{w} \cdot \nabla\phi_j)(a\boldsymbol{w} \cdot \nabla\phi_i)\, d\Omega, \qquad f_i^U = \int_\Omega f(a\boldsymbol{w} \cdot \nabla\phi_i)\, d\Omega, \tag{2.11}$$

for $i, j = 1 \ldots, n$. Note that the matrix U is symmetric and positive definite.

There has been much discussion of what the value of the stabilisation parameter α in (2.10) should be. It is possible to show that the SUPG solution satisfies the 'best possible' error estimate (for any degree of polynomial approximation) under the assumption that α is of the form

$$\alpha = \frac{\delta h}{|a\boldsymbol{w}|} = \frac{\delta h}{a}$$

where $\delta > 0$ is a parameter. One choice for δ (which is shown to be optimal in some contexts in [42]) is

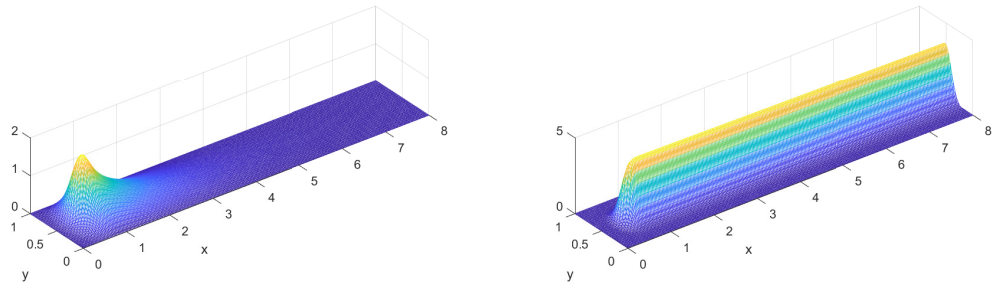
$$\delta = \max\left(0, \frac{1}{2}\left(1 - \frac{1}{P_h}\right)\right) = \max\left(0, \frac{1}{2} - \frac{D}{h|a\boldsymbol{w}|}\right).$$

For a uniform mesh with $|\boldsymbol{w}| = 1$, as we will use in all of the experiments, this will lead to one fixed value of

$$\delta = \max\left(0, \frac{1}{2} - \frac{D}{ha}\right).$$

Note that when $\delta = 0$, the SUPG and Galerkin discretisations are equivalent.

Using the same parameters as in Figure 2.3, we can see the effect of the SUPG stabilisation. In Figure 2.4 we can see that plot (a) with $P_h < 1$ is unchanged by the SUPG method as would be expected with $\delta = 0$. Plot (b) with $P_h \geq 1$ is much smoother than in Figure 2.3 as the solution approaches the boundary, showing the effect of the SUPG method.



(a) Plot of the concentration with $N = 32$, $a = 1$, $D = 0.1$, and $P_h = 0.156$.

(b) Plot of the concentration with $N = 32$, $a = 1$, $D = 0.0001$, and $P_h = 156$.

Figure 2.4: Comparison of the pollutant concentration with low and high mesh-Péclet numbers. These numerical approximations are obtained using an SUPG approximation.

2.4 Conclusions

In this chapter we introduced a physical model describing the release of a pollutant in a 2D domain, with the initial distribution represented by a Gaussian pulse. In §2.2 we presented the Galerkin finite element method, which was used to discretise the domain and obtain an approximate solution of the model with fixed parameter values, using a fine and coarse grid. In §2.3 we discussed the mesh-Péclet number and in Figure 2.3 we presented sample solutions with low and high mesh-Péclet numbers, demonstrating how solutions with higher mesh-Péclet numbers had oscillations. We presented the Streamline-Upwind Petrov Galerkin method, and found the solution to the model with the same parameter values as Figure 2.3 (b), showing in Figure 2.4 that the SUPG method results in solutions with fewer oscillations.

Chapter 3

Monte Carlo methods

3.1 Monte Carlo method for estimating the mean

Monte Carlo simulation is a process that relies on repeated random sampling and statistical analysis to estimate the value of a quantity of interest. When we have a mathematical model that is dependent upon a number of uncertain parameters, such as the advection-diffusion problem, we can identify probability distributions for each of the input parameters. We then draw random samples from the distributions of the input parameters to obtain a set of input parameters. The set of input parameters is used to obtain a corresponding set of outputs. For example, suppose X is a random variable and we want to estimate the mean $\theta^{(1)} = \mathbb{E}[f^{(1)}(X)]$ for some function $f^{(1)}$. We can construct a Monte Carlo approximation to $\theta^{(1)}$ by generating k independent samples $\{X_i\}_{i=1}^{k}$ of X and forming

$$\hat{\theta}^{(1)} = \frac{1}{k} \sum_{i=1}^{k} f_{X_i}^{(1)} \tag{3.1}$$

where $f_{X_i}^{(1)} \equiv f^{(1)}(X_i)$. Since X_i is a random variable, the evaluation $f_{X_i}^{(1)}$ is a random variable [57]. Algorithm 1 in Appendix A outlines an algorithm for estimating the mean using the Monte Carlo method.

3.1.1 Mean-squared error

To measure the accuracy of the estimate (3.1), we examine its mean-squared error (MSE) which is the mean-squared difference between the estimated value and the actual value. We will calculate the mean-squared error for (3.1) using k samples with variance, $\mathbb{V}(f^{(1)})$, of $f^{(1)}(X)$. Assuming that the values $f^{(1)}_{X_i}$ are independent and identically distributed, then $\hat{\theta}^{(1)}$ is unbiased, as

$$\begin{aligned}\mathbb{E}(\hat{\theta}^{(1)}) &= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} f^{(1)}_{X_i}\right)\\ &= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}(f^{(1)}_{X_i})\\ &= \frac{1}{k}(k\,\mathbb{E}(f^{(1)}_{X})) = \mathbb{E}(f^{(1)}_{X}) = \theta^{(1)}.\end{aligned}$$

The mean-squared error is

$$\begin{aligned}\mathrm{MSE}(\hat{\theta}^{(1)}) &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \theta^{(1)})^2\right]\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}) + \mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\right]\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))^2 + 2(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))(\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)}) + (\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\right]\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))^2\right] + \mathbb{E}\left[2(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))(\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})\right] + \mathbb{E}\left[(\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\right]\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))^2\right] + 2(\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})\,\mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))\right] + (\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))^2\right] + 2(\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})(\mathbb{E}(\hat{\theta}^{(1)}) - \mathbb{E}(\hat{\theta}^{(1)})) + (\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\\ &= \mathbb{E}\left[(\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))^2\right] + (\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)})^2\\ &= \mathbb{V}(\hat{\theta}^{(1)}) + \mathrm{Bias}(\hat{\theta}^{(1)}, \theta^{(1)})^2.\end{aligned}$$

As $\hat{\theta}^{(1)}$ is an unbiased estimate of $\theta^{(1)}$, to find the MSE with respect to k we only need to consider the variance:

$$\mathbb{V}(\hat{\theta}^{(1)}) = \mathbb{V}\left[\frac{1}{k}\sum_{i=1}^{k} f^{(1)}_{X_i}\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{k} f_{X_i}^{(1)} - \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} f_{X_i}^{(1)}\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{k} f_{X_i}^{(1)} - \frac{1}{k}k\theta^{(1)}\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{k} f_{X_i}^{(1)} - \theta^{(1)}\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{k^2}\left(\sum_{i=1}^{k} f_{X_i}^{(1)}\right)^2 - \frac{2\hat{\theta}^{(1)}}{k}\sum_{i=1}^{k} f_{X_i}^{(1)} + (\theta^{(1)})^2\right] \\
&= \frac{1}{k^2}\,\mathbb{E}\left[\left(\sum_{i=1}^{k} f_{X_i}^{(1)}\right)^2\right] - \frac{2\theta^{(1)}}{k}\,\mathbb{E}\left[\sum_{i=1}^{k} f_{X_i}^{(1)}\right] + (\theta^{(1)})^2 \\
&= \frac{1}{k^2}\,\mathbb{E}\left[\left(\sum_{i=1}^{k} f_{X_i}^{(1)}\right)^2\right] - (\theta^{(1)})^2.
\end{aligned}
$$

Given that the variance of $f_{X_i}^{(1)}$ is defined as

$$
\mathbb{V}(f_{X_i}^{(1)}) = \mathbb{E}((f_{X_i}^{(1)})^2) - (\mathbb{E}(f_{X_i}^{(1)}))^2 = \mathbb{E}((f_{X_i}^{(1)})^2) - (\theta^{(1)})^2,
$$

and that

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_{i=1}^{k} f_{X_i}^{(1)}\right)^2\right] &= \mathbb{E}\left[\sum_{i=1}^{k}\sum_{j=1}^{k} f_{X_i}^{(1)} f_{X_j}^{(1)}\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{k}(f_{X_i}^{(1)})^2 + \sum_{i=1}^{k}\sum_{j=1(j\neq i)}^{k} f_{X_i}^{(1)} f_{X_j}^{(1)}\right] \\
&= k\,\mathbb{E}((f_{X_i}^{(1)})^2) + \sum_{i=1}^{k}\sum_{j=1(j\neq i)}^{k} \mathbb{E}(f_{X_i}^{(1)})\,\mathbb{E}(f_{X_j}^{(1)}) \\
&= k\,\mathbb{E}((f_{X_i}^{(1)})^2) + (k^2-k)(\hat{\theta}^{(1)})^2,
\end{aligned}
$$

we can see that

$$\mathbb{V}(\hat{\theta}^{(1)}) = \frac{1}{k^2}(k\,\mathbb{E}((f_{X_i}^{(1)})^2) + (k^2 - k)(\theta^{(1)})^2) - (\theta^{(1)})^2 \tag{3.2}$$

$$= \frac{\mathbb{E}((f_{X_i}^{(1)})^2) - (\theta^{(1)})^2}{k} = \frac{\mathbb{V}(f_{X_i}^{(1)})}{k}. \tag{3.3}$$

Therefore we have established that the MSE of $\hat{\theta}^{(1)}$ is

$$\text{MSE} = \frac{\mathbb{V}(f^{(1)})}{k}. \tag{3.4}$$

The rate of convergence of the MSE of $\hat{\theta}^{(1)}$ is $O(k^{-1})$. The cost of estimating the mean using the Monte Carlo method is k evaluations of $f^{(1)}$.

3.1.2 The Ishigami function

Before experimenting with the Monte Carlo method and the advection-diffusion problem, we would first like to check the accuracy of the estimated MSE and the convergence of the Monte Carlo estimate of the mean. To do this, we will use a simple function with uncertain inputs and known mean and variance. First introduced in [67], the Ishigami function has been used frequently to test methods for sensitivity analysis and uncertainty quantification [114]. The function is given by

$$f^{(1)}(X_1, X_2, X_3) = \sin(X_1) + a\sin^2(X_2) + bX_3^4\sin(X_1), \qquad X_i \sim \text{Uniform}(-\pi, \pi) \tag{3.5}$$

for $i = 1, 2, 3$. The parameters a and b are fixed at 5 and 0.1, respectively, and the associated mean and variance are $\theta^{(1)} = 2.5$ and $\mathbb{V}(f^{(1)}) = 10.845$.

We can now examine the convergence of the MSE of $\hat{\theta}^{(1)}$ estimated using repeated applications of the Monte Carlo method. We select five values of $k = 100, 200, 400, 800, 1600$ and using the algorithm in Algorithm 1 (see Appendix A), we can calculate a single value of $\hat{\theta}^{(1)}$ for each given value of k. Having calculated $R = 100$ values of $\hat{\theta}^{(1)}$ (which we will call replicates) for each value of k, we plot the 100 replicate estimated means of the Ishigami function in a boxplot for each k. We

know the true mean to be 2.5, so we centre the boxplots around zero. In Figure 3.1

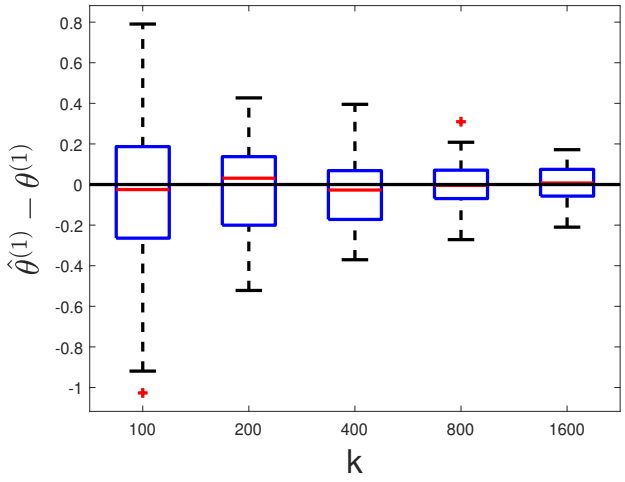


Figure 3.1: Boxplot of $R = 100$ replicates of $\hat{\theta}^{(1)}$.

we can see range of calculated values of $\hat{\theta}^{(1)}$, as well as the quantiles and mean of the estimates. The mean of the estimates is near the true value at each k, and the range of estimates of the mean converge to the true mean as k increases. Next we wish to examine whether the computed MSE of $\hat{\theta}^{(1)}$ matches the theoretical MSE for the Monte Carlo method. We plot the computed MSE $= \frac{\mathbb{V}(\hat{\theta}^{(1)})}{k}$, found using the variance of the replicates of $\hat{\theta}^{(1)}$ in Figure 3.1, and the theoretical MSE found using the true $\mathbb{V}(f^{(1)})$, for each value of k. In Figure 3.2 we can see the computed MSE of $\hat{\theta}^{(1)}$ approximately matches the theoretical MSE for each k, and the computed MSE decreases at the same rate as the theoretical MSE, so we can be confident in the accuracy of the theoretical MSE and the convergence of the Monte Carlo method.

3.1.3 The advection-diffusion equation

The advection-diffusion equation (2.1) models pollution concentration throughout a two-dimensional domain. There are many potential quantities of interest (QoI) we could examine, for example the pollution concentration at an individual point or the mean concentration in a localised region. The pollution concentration throughout the domain is also dependent upon the advection a, and the diffusion D, both of

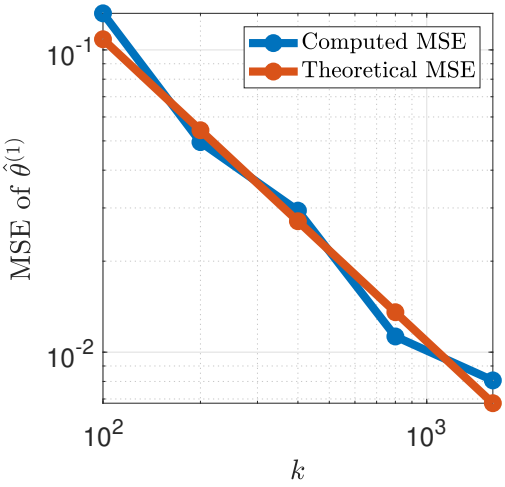


Figure 3.2: Comparison of computed and theoretical MSE of $\hat{\theta}^{(1)}$.

which are uncertain. This means we can use equation (2.1), with a chosen QoI as the output and a and D as the inputs. We fix the mesh size as $N = 32$, the pollutant source spread $S = 0.1$, and the point of release as $(x_s, y_s) = (0.5, 0.5)$. We choose to look at the average values of the pollution in the domain at $x = 1$, which we call ω. We measure the average pollution by looking at the L^2-norm of the concentration at $x = 1$, that is

$$f^{(1)}(a, D) \simeq \left\{ \int_{\omega} (u_h)^2 dl \right\}^{\frac{1}{2}}. \tag{3.6}$$

This choice provides a representative scalar measure of the downstream concentration, while remaining close enough to the source that the effects of advection and diffusion are clearly reflected in the output. Other quantities of interest could be defined (for example, the solution at a single point, a different norm of u, or measurements taken at a different downstream location), but fixing both the form and location of the measurement enables consistent comparison across parameter variations.

We discretise the domain as in §2.2.2 such that the element size is $h = \frac{1}{N}$, and we always have nodes on $x = 1$, which we call y_i. We approximate the solution of the differential equation using the SUPG method and compute the corresponding L^2-norm along $x = 1$ using the midpoint rule. This defines the function $f^{(1)}(a, D)$,

where

$$f^{(1)}(a, D) = \frac{1}{\sqrt{2N}} \left\{ \sum_{i=1}^{N} \left(u_h(1, y_i) + u_h(1, y_{i+1}) \right) \right\}^{\frac{1}{2}}. \tag{3.7}$$

The function $f^{(1)}$ has mean $\theta^{(1)}$ which can be approximated by $\hat{\theta}^{(1)}$, generated by using k samples in the Monte Carlo method as described above, that is,

$$\hat{\theta}^{(1)} = \frac{1}{k} \sum_{i=1}^{k} f_{X_i}^{(1)}, \tag{3.8}$$

where X_i are random samples of the vector $\mathbf{X} = (a, D)$. To calculate the MSE we need to know the true mean value of the concentration for the chosen distributions of the advection and diffusion terms. Here we will take the 'true' mean to be a Monte Carlo estimate generated using a large value of $k = 10^7$ samples.

We now consider the effect of these parameter distributions for the mesh-Péclet number. As previously discussed, the mesh-Péclet number is

$$P_h(a, D) = \frac{ah}{2D},$$

where a is the strength of the advection, D is the diffusion coefficient and h is the mesh size. In the model, h is fixed but a and D are random variables, therefore P_h is also a random variable and P_h will have a distribution that is a ratio of the distributions of a and D. It is theoretically possible to estimate the full probability distribution function for P_h, however we are only interested in the expectation of P_h. If μ_a and μ_D are the means of a and D respectively, and σ_a^2 and σ_D^2 are their variances, then we can approximate the mean of P_h as follows.

Letting $g(a, D) = \frac{a}{D}$, we can find the Taylor series expansion of $g(a, D)$ around μ_a and μ_D. We start by taking the first and second partial derivatives:

$$\frac{\partial g}{\partial a} = \frac{1}{D}, \qquad \frac{\partial g}{\partial D} = \frac{-a}{D^2},$$
$$\frac{\partial^2 g}{\partial a^2} = 0, \qquad \frac{\partial^2 g}{\partial D^2} = \frac{2a}{D^3},$$

$$\frac{\partial^2 g}{\partial a \partial D} = \frac{-1}{D^2}.$$

The Taylor series expansion of $\frac{a}{D}$ at (μ_a, μ_D) is therefore

$$\frac{a}{D} \simeq \frac{\mu_a}{\mu_D} + \frac{(a-\mu_a)}{\mu_D} + (D-\mu_D)\left(\frac{-\mu_a}{\mu_D^2}\right) + (D-\mu_D)^2\left(\frac{\mu_a}{\mu_D^3}\right) + (a-\mu_a)(D-\mu_D)\left(\frac{-1}{\mu_D^2}\right).$$

Next we take the expectation of $\frac{a}{D}$, so

$$\mathbb{E}\left(\frac{a}{D}\right) \simeq \frac{\mu_a}{\mu_D} + \frac{(\mathbb{E}(a)-\mu_a)}{\mu_D} + (\mathbb{E}(D)-\mu_D)\left(\frac{-\mu_a}{\mu_D^2}\right) + \mathbb{E}(D-\mu_D)^2\left(\frac{\mu_a}{\mu_D^3}\right) + \mathbb{E}((a-\mu_a)(D-\mu_D))\left(\frac{-1}{\mu_D^2}\right).$$

We know $\mathbb{E}(D-\mu_D)^2 = \mathbb{V}(D)$ and $\mathbb{E}((a-\mu_a)(D-\mu_D)) = \mathrm{Cov}(a, D)$, however, a and D are independent random variables so $\mathrm{Cov}(a, D) = 0$. The expected value of $g(a, D) = \frac{a}{D}$ is now

$$\mathbb{E}\left(\frac{a}{D}\right) \simeq \frac{\mu_a}{\mu_D} + \frac{\sigma_D^2 \mu_a}{\mu_D^3}$$

therefore we can approximate the mean mesh-Péclet number using

$$\mathbb{E}(P_h) \simeq \frac{h}{2}\left(\frac{\mu_a}{\mu_D} + \frac{\sigma_D^2 \mu_a}{\mu_D^3}\right).$$

We need to select reasonable distributions for a and D. As a represents wind strength, we choose a uniform distribution as we can set the wind within a range, but we do not have any information of the distribution within the range of α_a and β_a. For the diffusion, we know diffusion must be positive, so we choose a log-normal distribution with mean α_D and variance β_D. Using the distributions

$$a \sim U(\alpha_a, \beta_a) \qquad D \sim \text{log-normal}(\alpha_D, \beta_D)$$

then

$$\mu_a = \frac{1}{2}(\alpha_a + \beta_a), \qquad \mu_D = e^{(\alpha_D + \frac{\beta_D}{2})},$$

and

$$\sigma_D^2 = (e^{\beta_D} - 1)(e^{(2\alpha_D + \beta_D)}).$$

These can be used to estimate the mean mesh-Péclet number for a given set of input distribution parameters.

By varying the parameters of the distributions of a and D, we can intentionally increase or decrease the mean mesh-Péclet number. A higher mean mesh-Péclet number would result in the Péclet number being greater than 1 more frequently and would potentially mean Galerkin solutions being affected by oscillations. Our principal model has $h = \frac{1}{32}$, and we use the mean Péclet number calculation to select two sets of parameters. Distribution $\mathcal{I}_{\text{high}}$ is sampled from

$$a \sim U(0.5, 1.5) \qquad \text{and} \qquad D \sim \text{log-normal}(-7, 0.9), \tag{3.9}$$

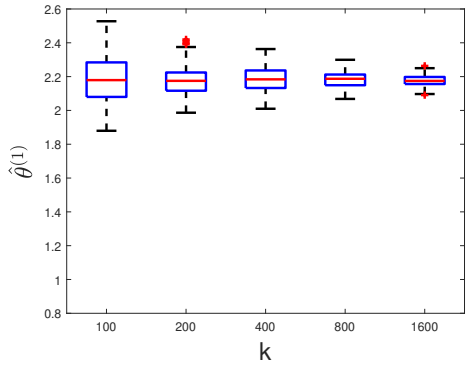
and has $\mathbb{E}(a) = 1$, $\mathbb{E}(D) = 0.001$, and $\mathbb{E}(P_h) \simeq 26.873$. This distribution has samples with a low mean diffusion, so we expect the solution will be advection dominated. Distribution $\mathcal{I}_{\text{low}}$ is sampled from

$$a \sim U(0.5, 1.5) \qquad \text{and} \qquad D \sim \text{log-normal}(-3, 0.5), \tag{3.10}$$

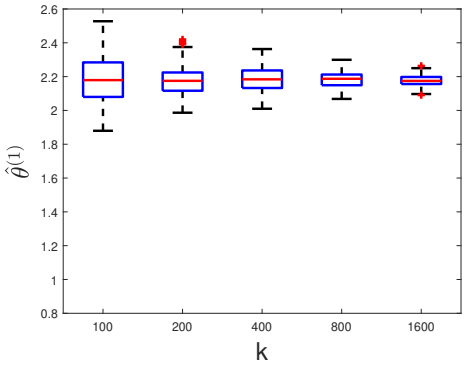
and has $\mathbb{E}(a) = 1$, $\mathbb{E}(D) = 0.056$, and $\mathbb{E}(P_h) \simeq 0.403$. The mean and range of the sampled diffusion values for this input distribution will be higher. These two distributions will allow us to compare the advection-diffusion model problem when the mean mesh-Péclet number is high and low.

We now examine how these $\mathcal{I}_{\text{low}}$ and $\mathcal{I}_{\text{high}}$ effect the performance of different numerical discretisations. In this section we will compare the solutions of the model problem found using the SUPG and Galerkin methods. Previously in §3.1.3 we decided on the L^2-norm of the concentration of the pollutant at $x = 1$ (see (3.7)) as the QoI. As with the Ishigami function in §3.1.2, we use the Monte Carlo method in (3.8) to get an estimate of the mean of the QoI, $\hat{\theta}^{(1)}$, for a given set of input parameters. We then repeat the process for $R = 100$ replications for the five values

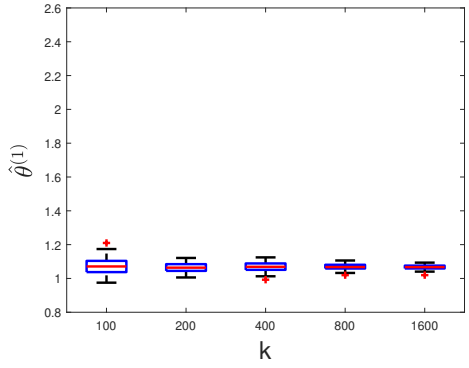
of $k = 100, 200, 400, 800, 1600$, using both the Galerkin and SUPG method for inputs chosen from distributions $\mathcal{I}_{\text{low}}$ and $\mathcal{I}_{\text{high}}$. We can then plot boxplots of the chosen output for five values of k, using both methods of solving the PDE, for both input distributions. In Figure 3.3, the mean and variance of $\hat{\theta}^{(1)}$ is lower in distribution



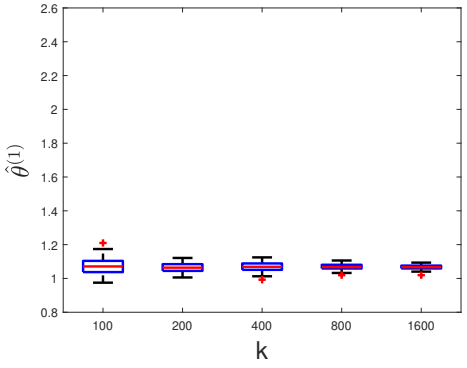
(a) Boxplot of $R = 100$ replicates of $\hat{\theta}^{(1)}$ using SUPG for input distribution $\mathcal{I}_{\text{high}}$.



(b) Boxplot of $R = 100$ replicates of $\hat{\theta}^{(1)}$ using Galerkin for input distribution $\mathcal{I}_{\text{high}}$.



(c) Boxplot of $R = 100$ replicates of $\hat{\theta}^{(1)}$ using SUPG for input distribution $\mathcal{I}_{\text{low}}$.



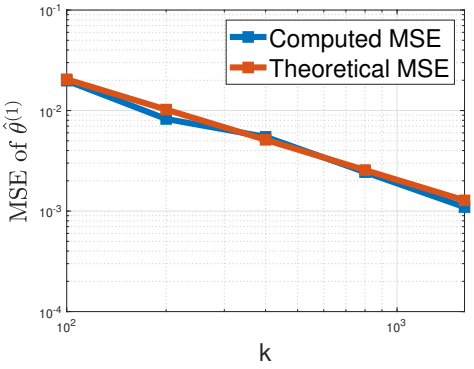
(d) Boxplot of $R = 100$ replicates of $\hat{\theta}^{(1)}$ using Galerkin for input distribution $\mathcal{I}_{\text{low}}$.

Figure 3.3: Box-plots of 100 Monte Carlo replications of $\hat{\theta}^{(1)}$.

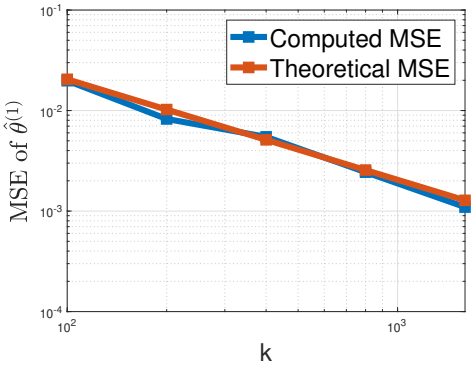
$\mathcal{I}_{\text{low}}$ than distribution $\mathcal{I}_{\text{high}}$, suggesting the model problem output is sensitive to the diffusion. The range of mean values appear to converge as we would expect for both distributions and methods. Given the very similar results for both distributions whether SUPG or Galerkin is used, it would seem there is minimal difference between the methods for the model. Since the oscillations at the outflow boundary occur sufficiently downstream so as not to affect the QoI, and the wind is fixed at $\mathbf{w} = [1, 0]$,

oscillations at the y-boundary are minimal.

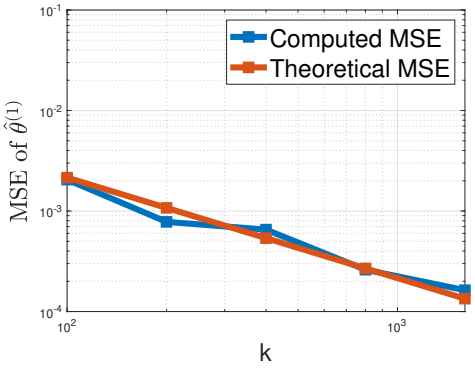
For the next experiment we examine the effect of using SUPG or Galerkin discretisation on the MSE of $\hat{\theta}^{(1)}$. Using the $R = 100$ replicates of $\hat{\theta}^{(1)}$ from Figure 3.3, we can compute the MSE of $\hat{\theta}^{(1)}$ for each k. We can compare to the theoretical MSE using the 'true' values of $\mathbb{V}(f^{(1)})$ from (3.9) and (3.10). Figure 3.4 shows, for



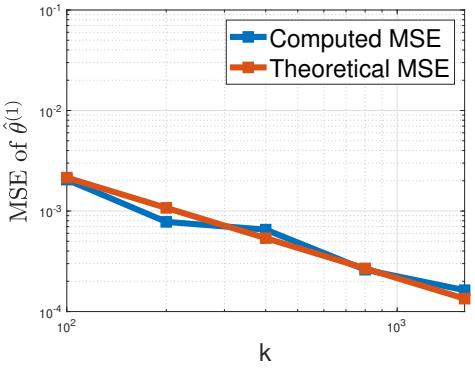
(a) Estimated and theoretical MSE of SUPG estimates of $\theta^{(1)}$ using distribution $\mathcal{I}_{\text{high}}$.



(b) Estimated and theoretical MSE of Galerkin estimates of $\theta^{(1)}$ using distribution $\mathcal{I}_{\text{high}}$.



(c) Estimated and theoretical MSE of SUPG estimates of $\theta^{(1)}$ using distribution $\mathcal{I}_{\text{low}}$.



(d) Estimated and theoretical MSE of Galerkin estimates of $\theta^{(1)}$ using distribution $\mathcal{I}_{\text{low}}$.

Figure 3.4: Comparison of the theoretical MSE to the Monte Carlo estimate of the MSE for SUPG and Galerkin method.

both input distributions and methods of solving, the MSE of $\hat{\theta}^{(1)}$ is similar to the theoretical MSE and that the MSE is converging as expected. The methods using input parameters from $\mathcal{I}_{\text{low}}$ have a lower MSE than those using input parameters from $\mathcal{I}_{\text{high}}$ due to the lower variance of $f^{(1)}$ when the diffusion is lower. We will use

the SUPG method exclusively for $f^{(1)}(a, D)$ going forward as it is a more robust method if we wish to vary other parameters.

3.2 Bootstrapping and confidence intervals

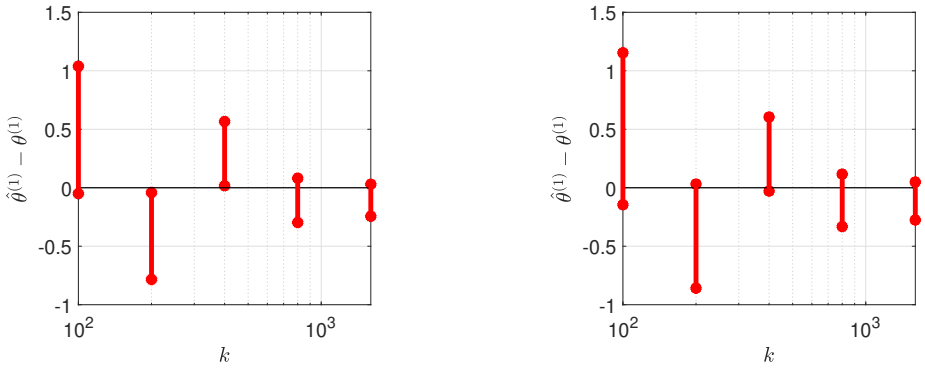
So far, we have focused on estimating the mean of a random variable using the Monte Carlo method. In order to assess the accuracy of any estimate, it is helpful to also estimate the associated confidence intervals. That is, we want to calculate an interval such that we are confident that this interval contains the true value of the mean in a fixed percentage of cases. We will consider the 90% and 95% confidence intervals, that is, we set $\gamma = 0.05$ and $\gamma = 0.1$ and consider an interval bounded by the $\gamma/2$ and $1-\gamma/2$ quantiles of the sampling distribution. Confidence intervals can be estimated in different ways [26, 32]. For example, to estimate confidence intervals using Monte Carlo estimates based on k independent samples, we could calculate replicates of the Monte Carlo estimate of the mean as we did previously, then we could take the $\gamma/2$ and $1-\gamma/2$ quantiles of these estimates as the lower and upper bounds, respectively, of the confidence interval. If R is large enough, this should give useful results but it is clearly very computationally expensive, as it involves evaluating the model with k additional independent samples to generate each of these R estimates of the mean required for confidence intervals. In this section, we will describe a bootstrapping approach which we use to obtain estimates of confidence intervals at a significantly lower cost.

Bootstrapping is a statistical resampling technique first proposed by Efron [36]. A bootstrapping method, instead of requiring the model to be evaluated with additional samples, resamples the existing function evaluations (with replacement) repeatedly to create many additional simulated data points. The process involves drawing random samples from the original dataset with equal probability, the "with replacement" aspect meaning that the procedure can select a data point more than once for the resampled dataset. Hence the resampled dataset will have many different combinations of the values that already exist in the original dataset. Suppose

we resample the dataset R_b times to get R_b new estimates of the mean. The key assumption is that the sampled distribution is unbiased with independent samples. It can be shown that this approach yields reasonable estimates of the means and standard deviations of, e.g., normal and log-normal distributions [37].

Algorithm 2 in Appendix A outlines an algorithm for bootstrapping the Monte Carlo estimation of confidence intervals for the mean of a function $f^{(1)}$. Note that the choice of the value of R_b is important: it has to be large enough to ensure that the confidence intervals cover the true mean $100(1-\gamma)\%$ of the time. The cost of re-sampling R_b values and calculating the quantiles can be considered negligible, so we consider the only cost of using this method to be evaluating $f^{(1)}$, k times.

We can test this bootstrapping method using the Ishigami function introduced in Section 3.1.2. We can calculate the 90% and 95% confidence intervals using the method outlined in Algorithm 2 and, as we know the true mean, centre them around zero see whether the confidence intervals cover the true mean. In Figure 3.5 the 90%



(a) 90% confidence intervals of the mean of the Ishigami function.

(b) 95% confidence intervals of the mean of the Ishigami function.

Figure 3.5: Comparison of 90% and 95% confidence intervals for $\theta^{(1)}$ centered around 0.

confidence intervals are slightly narrower at each level of k, and do not cover the true mean for all k whereas the 95% confidence intervals do cover the true mean for all values of k. These are only a single realisation of the confidence intervals, and it is known the confidence intervals do not cover the true mean every time. We want

to be certain they will cover the true mean $100(1-\gamma)\%$ of the time. To check the coverage we calculate the confidence intervals 1000 times, allowing us to see what percentage of the time the confidence intervals cover the true mean. In Table 3.1

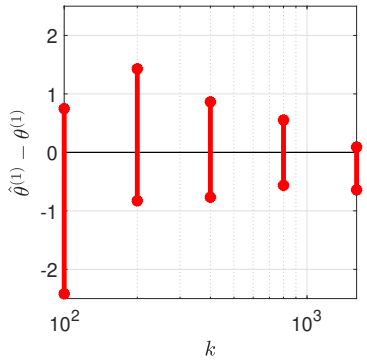
Sample Size	**k = 100**	**k = 200**	**k = 400**	**k = 800**	**k = 1600**
90% Coverage %	89.7	89.2	90.7	91	89.7
95% Coverage %	94.2	93.9	94.6	96	94.8

Table 3.1: Observed coverage for confidence intervals of $\theta^{(1)}$ with varying sample size. Figures based on 1000 replications of the confidence intervals.

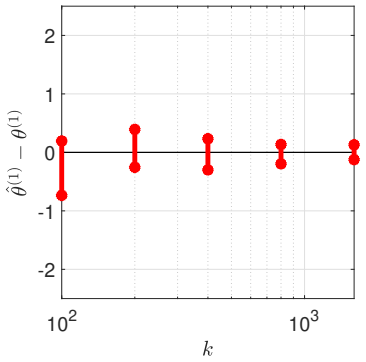
we see that the confidence intervals cover the true mean approximately $100(1-\gamma\%)$ of the time as we would hope. This allows us to confidently use bootstrapping to estimate the confidence intervals much more cheaply than by calculating estimates of the mean using Monte Carlo replications. We focus on the 95% confidence intervals going forward.

3.2.1 Bootstrapping for the advection-diffusion problem

In this section we will use bootstrapping to find confidence intervals for the mean $\theta^{(1)}$ of $f^{(1)}(a, D)$, defined in (3.6), for the advection-diffusion model. We calculate the confidence intervals then subtract the true mean, centering the confidence intervals around zero. In Figure 3.6 we can see the size of the confidence intervals is reduced as k increases. We can also see the size of the confidence interval is affected by the chosen input distribution, with distribution $\mathcal{I}_{\text{low}}$ having a lower variance in $f^{(1)}$ which results in smaller confidence intervals. Although Figure 3.6 is promising, it is only a single realisation of the confidence intervals. To trust that the confidence intervals are robust for the advection-diffusion problem, we need to verify they cover the true mean $100(1-\gamma)\%$ of the time. We calculate the confidence intervals 1000 times, then we can see what percentage of the 1000 realisations of the confidence intervals cover the true means. In Table 3.2 we can see that the bootstrapped confidence intervals cover the true mean approximately 95% of the time as would be expected for 95% confidence intervals. The observed coverage does not depend on sample size in the



(a) Bootstrap estimates of the 95% confidence interval of $\theta^{(1)}$ sampled from distribution $\mathcal{I}_{\text{high}}$



(b) Bootstrap estimates of the 95% confidence interval of $\theta^{(1)}$ sampled from distribution $\mathcal{I}_{\text{low}}$

Figure 3.6: Estimates of the 95% confidence interval of the output calculated using 1000 bootstrap estimates of $\theta^{(1)}$.

Input Distribution	**Sample Size**				
	k = 100	**k = 200**	**k = 400**	**k = 800**	**k = 1600**
$\mathcal{I}_{\text{low}}$	93.9	95.7	94	94.2	95.6
$\mathcal{I}_{\text{high}}$	94.4	95.1	94.9	94.8	96.5

Table 3.2: Observed coverage for 95% confidence intervals of $\theta^{(1)}$ with increasing sample size. Figures based on 1000 replications of the confidence intervals.

advection-diffusion problem. As the coverage is as expected we can be confident in using confidence intervals calculated using bootstrapping for the advection-diffusion problem going forward.

3.3 Conclusions

In §3.1 we presented the Monte Carlo method to estimate an uncertain quantity of interest. We chose the mean-squared error as a measure of accuracy of the estimate and derived an expression for the theoretical MSE of the Monte Carlo method as a function of the output variance and the number of samples evaluated. We presented the Ishigami function with known mean and variance and examined how the mean estimated using the Monte Carlo method converged as the number of samples increased in Figure 3.1. In Figure 3.2 we showed that the calculated estimated MSE of the mean is the same as the theoretical MSE found using true variance values.

In §3.1.3 we introduced an uncertain advection strength, and diffusion, to the model, and chose the L^2-norm of the concentration at $x = 1$ on the advection-diffusion problem domain as a QoI, which we estimated using the MC method. We selected a uniform and log-normal distribution for the uncertain advection and diffusion terms respectively, then we derived a term to calculate the expected mesh-Péclet number of the domain for a chosen uniform lower bound, upper bound, log-mean, and log-standard deviation. We used the calculated expected mesh-Péclet to find two sets of input parameters, with high and low expected mesh-Péclet numbers. In Figure 3.3 we showed box-plots of repeated estimates of the mean of the QoI evaluated using the Monte Carlo method. We found these mean estimates when the model output is evaluated using both the Galerkin and SUPG methods, showing solutions for both methods converged as the number of samples is increased, whether the mesh-Péclet number is high or low. In Figure 3.4 we found, for both methods of evaluating the model, the calculated MSE is the same as the theoretical MSE, whether the expected mesh-Péclet number is high or low. We then chose the SUPG method of evaluating the model as the method of solving when evaluating the principal model $f^{(1)}$.

In §3.2 we discussed confidence intervals as a method of estimating the mean and visualising uncertainty more cheaply than using Monte Carlo replicates. We presented a method for finding confidence intervals using bootstrapping, and we tested this method for the Ishigami equation, showing in Figure 3.5 that the confidence intervals became narrower as the number of evaluations (k) of the equation increases. We tested the coverage of the confidence intervals using the Ishigami equation in Table 3.1, confirming the confidence intervals did cover the true mean the expected percentage of the time for all values of k, at two different coverage levels. In Figure 3.6 we found the 95% confidence intervals of the mean of the advection-diffusion problem converged as k increases, and in Table 3.2 we confirmed the 95% confidence intervals covered the true mean approximately as expected.

Chapter 4

Variance reduction

In this chapter we develop and analyse multi-fidelity variance reduction methods for Monte Carlo estimation under a fixed computational budget. We begin by reviewing standard control variates and methods that allow control variates with unknown means. We then introduce the novel CVB method, derive its theoretical properties, and compare its performance with existing multi-fidelity approaches using numerical experiments based on the Ishigami function and advection diffusion model. The results in this chapter motivate the application of CVB to more complex models in later chapters.

4.1 Standard control variates with known means

The control variates method is an estimation technique that requires two or more models. Suppose X is a random variable and we want to evaluate the mean $\theta^{(1)} = \mathbb{E}[f^{(1)}(X)]$ for some function $f^{(1)}$. We can construct a Monte Carlo approximation to $\theta^{(1)}$ in the standard way by generating independent samples, X_i for $i = 1, \ldots, k$ and forming

$$\hat{\theta}^{(1)} = \frac{1}{k} \sum_{i=1}^{k} f_{X_i}^{(1)} \tag{4.1}$$

where $f_{X_i}^{(1)} \equiv f^{(1)}(X_i)$. Now, suppose that we have an additional model, $f^{(2)}(X_i)$, where we know the true mean value $\theta^{(2)} = \mathbb{E}[f^{(2)}]$, and we can construct the Monte Carlo estimate

$$\hat{\theta}^{(2)} = \frac{1}{k} \sum_{i=1}^{k} f_{X_i}^{(2)} \tag{4.2}$$

where $f_{X_i}^{(2)} \equiv f^{(2)}(X_i)$. Here, the random variable $f_{X_i}^{(2)}$ is known as a control variate. To use the control variates method we must use the same values of X_i to obtain both $f_{X_i}^{(1)}$ and $f_{X_i}^{(2)}$. We refer to this method of estimating the mean as the control variates with known means (CVKM) method. Introducing a constant $\alpha \in \mathbb{R}$, we can construct a new estimator of $\theta^{(1)}$ as

$$\hat{\theta}_{\text{CVKM}}^{(1)} = \hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \theta^{(2)}). \tag{4.3}$$

The motivation for the control variate estimator arises from the correlation between the quantity of interest $f^{(1)}$ and the auxiliary model $f^{(2)}$. If these two quantities are positively correlated, then an overestimate of the mean of $f^{(2)}$ is likely to coincide with an overestimate of the mean of $f^{(1)}$, and similarly for underestimates. The control variate estimator exploits this behaviour by subtracting a scaled version of the estimation error in $f^{(2)}$ from the Monte Carlo estimate of the mean of $f^{(1)}$. By choosing an appropriate scaling parameter α, this subtraction can partially cancel the sampling error in the Monte Carlo estimator, leading to a reduced variance compared to standard Monte Carlo.

The coefficient α determines the strength of the control variate correction and arises from approximating the high-fidelity output using a correlated lower-cost approximation. When the control variate has a mean close to that of the quantity of interest and is strongly correlated, this approximation provides an estimate of the mean while significantly reducing variance. The optimal value of α depends on the covariance between the two variables and their variances, with larger values typically arising when the control variate is strongly correlated with the quantity of

interest [52]. Note that

$$\mathbb{E}(\hat{\theta}^{(1)}_{\text{CVKM}}) = \mathbb{E}(\hat{\theta}^{(1)}) - \alpha(\mathbb{E}(\hat{\theta}^{(2)}) - \theta^{(2)}) = \mathbb{E}(\hat{\theta}^{(1)}) = \theta^{(1)},$$

so this estimator is unbiased. As $\hat{\theta}^{(1)}_{\text{CVKM}}$ is an unbiased estimator, the MSE (MSE_{CVKM}) of (4.3) is $\mathbb{V}(\hat{\theta}^{(1)}_{\text{CVKM}})$ as in the Monte Carlo method. Given

$$\mathbb{V}(\hat{\theta}^{(1)}) = \mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})^2), \qquad \mathbb{V}(\hat{\theta}^{(2)}) = \mathbb{E}((\hat{\theta}^{(2)} - \theta^{(2)})^2)$$

and

$$\text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) = \mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})(\hat{\theta}^{(2)} - \theta^{(2)})),$$

we have

$$\begin{aligned}
\text{MSE}_{\text{CVKM}} &= \mathbb{E}((\hat{\theta}^{(1)}_{\text{CVKM}} - \mathbb{E}(\hat{\theta}^{(1)}_{\text{CVKM}}))^2) = \mathbb{E}((\hat{\theta}^{(1)}_{\text{CVKM}} - \theta^{(1)})^2) \\
&= \mathbb{E}(((\hat{\theta}^{(1)} - \theta^{(1)}) - \alpha(\hat{\theta}^{(2)} - \theta^{(2)}))^2) \\
&= \mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})^2 - 2\alpha(\hat{\theta}^{(1)} - \theta^{(1)})(\hat{\theta}^{(2)} - \theta^{(2)}) + \alpha^2(\hat{\theta}^{(2)} - \theta^{(2)})^2) \\
&= \mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})^2) - 2\alpha\,\mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})(\hat{\theta}^{(2)} - \theta^{(2)})) + \alpha^2\,\mathbb{E}((\hat{\theta}^{(2)} - \theta^{(2)})^2) \\
&= \mathbb{V}(\hat{\theta}^{(1)}) + \alpha^2\mathbb{V}(\hat{\theta}^{(2)}) - 2\alpha\text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}). \qquad (4.4)
\end{aligned}$$

Suppose now we wish to find the optimal value of α that minimises MSE_{CVKM}. To do this, we differentiate MSE_{CVKM} with respect to α:

$$\frac{d\text{MSE}_{\text{CVKM}}}{d\alpha} = 2\alpha\mathbb{V}(\hat{\theta}^{(2)}) - 2\text{Cov}(f^{(1)}, f^{(2)}).$$

Setting this equal to 0 and solving for α gives

$$\alpha = \alpha_* = \frac{\text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)})}{\mathbb{V}(\hat{\theta}^{(2)})}. \qquad (4.5)$$

The exact value of α_* cannot be computed exactly, so must be estimated using prior

samples. Differentiating $\frac{d\text{MSE}_{\text{CVKM}}}{d\alpha}$ again gives

$$\frac{d^2\text{MSE}_{\text{CVKM}}}{d\alpha^2} = 2\mathbb{V}(\hat{\theta}^{(2)}) > 0,$$

so this is a local minimum MSE.

We are interested in the dependence of MSE_{CVKM} on k. To find this, we only need to find the relationship between the covariance of the mean estimators (4.1) and (4.2) and the MSE of the functions $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$. As both $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$ are unbiased estimators of $\theta^{(1)}$ and $\theta^{(2)}$, respectively, the covariance of $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$ is

$$\begin{aligned}
\text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) &= \mathbb{E}((\hat{\theta}^{(1)} - \mathbb{E}(\hat{\theta}^{(1)}))(\hat{\theta}^{(2)} - \mathbb{E}(\hat{\theta}^{(2)}))) \\
&= \mathbb{E}((\hat{\theta}^{(1)} - \theta^{(1)})(\hat{\theta}^{(2)} - \theta^{(2)})) \\
&= \mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)} - \theta^{(2)}\hat{\theta}^{(1)} - \theta^{(1)}\hat{\theta}^{(2)} + \theta^{(1)}\theta^{(2)}) \\
&= \mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) - \theta^{(2)}\,\mathbb{E}(\hat{\theta}^{(1)}) - \theta^{(1)}\,\mathbb{E}(\hat{\theta}^{(2)}) + \theta^{(1)}\theta^{(2)} \\
&= \mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) - 2\theta^{(1)}\theta^{(2)} + \theta^{(1)}\theta^{(2)} \\
&= \mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) - \theta^{(1)}\theta^{(2)}. \qquad (4.6)
\end{aligned}$$

Expanding $\mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)})$ gives

$$\begin{aligned}
\mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) &= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} f_{X_i}^{(1)} \frac{1}{k}\sum_{j=1}^{k} f_{X_j}^{(2)}\right) \\
&= \frac{1}{k^2}\,\mathbb{E}\left(\sum_{i=1}^{k}\sum_{j=1}^{k} f_{X_i}^{(1)} f_{X_j}^{(2)}\right) \\
&= \frac{1}{k^2}\,\mathbb{E}\left(\sum_{i=1}^{k} f_{X_i}^{(1)} f_{X_i}^{(2)} + \sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k} f_{X_i}^{(1)} f_{X_j}^{(2)}\right).
\end{aligned}$$

As $f_{X_i}^{(1)}$ and $f_{X_j}^{(1)}$ are independent, we get the product of the expectations

$$\begin{aligned}
\mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) &= \frac{1}{k^2}\left(k\,\mathbb{E}(f^{(1)}f^{(2)}) + (k^2-k)\,\mathbb{E}(f^{(1)})\,\mathbb{E}(f^{(2)})\right) \\
&= \frac{1}{k^2}\left(k\,\mathbb{E}(f^{(1)}f^{(2)}) + (k^2-k)\theta^{(1)}\theta^{(2)}\right). \qquad (4.7)
\end{aligned}$$

Substituting (4.7) back in to (4.6) we get

$$\begin{aligned}
\mathrm{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) &= \mathbb{E}(\hat{\theta}^{(1)}\hat{\theta}^{(2)}) - \theta^{(1)}\theta^{(2)} \\
&= \frac{1}{k}\,\mathbb{E}(f^{(1)}f^{(2)}) + \theta^{(1)}\theta^{(2)} - \frac{1}{k}\theta^{(1)}\theta^{(2)} - \theta^{(1)}\theta^{(2)} \\
&= \frac{1}{k}\left(\mathbb{E}(f^{(1)}f^{(2)}) - \theta^{(1)}\theta^{(2)}\right) \\
&= \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k}. \qquad (4.8)
\end{aligned}$$

Recall that for $i = 1, 2$, the variance $\mathbb{V}(\hat{\theta}^{(i)}) = \frac{\mathbb{V}(f^{(i)})}{k}$. Therefore

$$\mathrm{MSE}_{\mathrm{CVKM}} = \frac{\mathbb{V}(f^{(1)})}{k} + \alpha^2\frac{\mathbb{V}(f^{(2)})}{k} - 2\alpha\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k}.$$

Now, if $\alpha = \alpha_*$, then

$$\begin{aligned}
\mathrm{MSE}_{\mathrm{CVKM}} &= \frac{\mathbb{V}(f^{(1)})}{k} + \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)^2\left(\frac{\mathbb{V}(f^{(2)})}{k}\right) \\
&\quad - 2\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k}\right) \\
&= \frac{\mathbb{V}(f^{(1)})}{k} + \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{k\mathbb{V}(f^{(2)})} - 2\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{k\mathbb{V}(f^{(2)})} \\
&= \frac{\mathbb{V}(f^{(1)})}{k} - \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{k\mathbb{V}(f^{(2)})} \\
&= \frac{\mathbb{V}(f^{(1)})}{k}\left(1 - \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})}\right).
\end{aligned}$$

Defining the correlation coefficient

$$\rho = \mathrm{Corr}(f^{(1)}, f^{(2)}) = \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\sqrt{\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})}},$$

the value of $\mathrm{MSE}_{\mathrm{CVKM}}$ with optimal α is

$$\mathrm{MSE}_{\mathrm{CVKM}} = \frac{\mathbb{V}(f^{(1)})}{k}(1 - \rho^2). \qquad (4.9)$$

The accuracy of the control variates method therefore depends on the correlation

between $f^{(2)}$ and the principal model $f^{(1)}$. The key steps of the CVKM method are shown in Algorithm 3 in Appendix A.

In the remainder of this section, we derive the theoretical MSE of the CVKM method, and perform numerical experiments to compare the computed and theoretical MSEs. We will also compare the MSE of the CVKM method to that of the Monte Carlo method for fixed computational budgets. To estimate the mean using Monte Carlo we require k evaluations of $f^{(1)}$, so the budget required is $P = k$ and the MSE of the Monte Carlo method (MSE_{MC}) method is (from (3.4))

$$\text{MSE}_{\text{MC}} = \frac{\mathbb{V}(f^{(1)})}{P}. \tag{4.10}$$

The CVKM method requires k evaluations of $f^{(1)}$ and $f^{(2)}$. If we assume the cost of calculating $f^{(1)}$ is 1, and the cost of evaluating $f^{(2)}$ is c, the budget required to use the CVKM method is

$$P = k(1 + c).$$

Therefore, from (4.9), we have

$$\text{MSE}_{\text{CVKM}} = \frac{\mathbb{V}(f^{(1)})}{P}(1 + c)(1 - \rho^2). \tag{4.11}$$

Equation (4.11) expresses the mean-squared error of the CVKM estimator explicitly in terms of the total computational budget P. This makes clear how the efficiency of CVKM depends on both the cost c and the correlation ρ between the high- and low-fidelity models. For a fixed budget, the MSE reduces at the same rate as standard Monte Carlo, but is smaller by the multiplicative factor $(1 + c)(1 - \rho^2)$. This highlights that the variance is reduced significantly when the low-fidelity model is strongly correlated with the high-fidelity output and inexpensive to evaluate. Conversely, when the correlation is weak or the low-fidelity cost is large, the benefit of CVKM diminishes. Equation (4.11) provides a clear criterion for assessing when the CVKM method is advantageous under budget constraints.

We now use the Ishigami function to compare the Monte Carlo and CVKM

methods in practice. Two approximations to the Ishigami function along with their mean values are given in [114]. These functions, $f^{(2)}$ and $f^{(3)}$, are defined as

$$f^{(2)}(X_1, X_2, X_3) = \sin(X_1) + 0.95a\sin^2(X_2) + bX_3^4\sin(X_1),$$

and

$$f^{(3)}(X_1, X_2, X_3) = \sin(X_1) + 0.6a\sin^2(X_2) + 9bX_3^2\sin(X_1), \tag{4.12}$$

with known means $\theta^{(2)} = 2.375$, $\theta^{(3)} = 1.5$, and known variances $\mathbb{V}(f^{(2)}) = 10.56$, $\mathbb{V}(f^{(3)}) = 12.46$. The correlations between $f^{(1)}$ and these two functions are $\rho_{12} = 0.9997$ and $\rho_{13} = 0.9442$, respectively, and the costs are fixed as $c_2 = 0.05$ and $c_3 = 0.001$. The theoretical MSE_{CVKM} can be calculated for a given P using the known values and (4.11). To examine the MSE of each method, we calculate 100 replicates of $\hat{\theta}^{(1)}$, obtained using standard Monte Carlo, then use control variates methods with $f^{(2)}$ and $f^{(3)}$ to calculate two sets of $\hat{\theta}^{(1)}_{\text{CVKM}}$. We then compare the calculated MSE, and theoretical MSE for 5 different values of P using both methods. In Figure 4.1 we can see the MSE of the mean estimate is reduced using both control

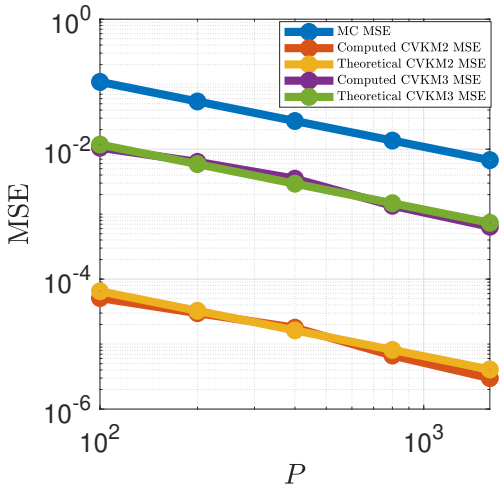


Figure 4.1: Comparison of the computed and theoretical MSE of $\hat{\theta}^{(1)}_{\text{CVKM}}$ for the Ishigami function.

variates, with the more highly correlated $f^{(2)}$ reducing the MSE more than $f^{(3)}$ when used as a control variate. The computed and theoretical values of MSE_{CVKM} using

both control variates are similar. For the Ishigami function we know the theoretical correlation between the function and its approximations, but we would normally estimate these correlations as described in Algorithm 3 in Appendix A, which may introduce a source of sampling error to any control variates results.

4.2 Control variates with estimated means

To use control variates you must know the expected value of any approximation to $f^{(1)}$. In [103], the authors introduce a method using control variates with estimated means. This involves replacing the known mean $\theta^{(2)}$ in (4.3) with an estimated mean, $\tilde{\theta}^{(2)}$, calculated via a pilot experiment. Here, the pilot experiment refers to an initial, Monte Carlo simulation performed independently of the main simulation, using the same approximation model but a separate set of k' independent samples, Y_i. The purpose of this pilot experiment is to estimate the expected value of the approximation before applying the control variate method in the main experiment. The estimated mean is therefore

$$\tilde{\theta}^{(2)} = \frac{1}{k'} \sum_{i=1}^{k'} f_{Y_i}^{(2)}, \tag{4.13}$$

where $f_{Y_i}^{(2)} = f^{(2)}(Y_i)$. Note that $\tilde{\theta}^{(2)}$ is a random variable with mean $\theta^{(2)}$ and variance $\mathbb{V}[f^{(2)}]/k'$. The control variates with estimated means (CVEM) method replaces (4.3) with the estimate

$$\hat{\theta}_{\text{CVEM}}^{(1)} = \hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \tilde{\theta}^{(2)}), \tag{4.14}$$

where $\alpha \in \mathbb{R}$ is a constant. Given that

$$\mathbb{E}(\hat{\theta}_{\text{CVEM}}^{(1)}) = \mathbb{E}(\hat{\theta}^{(1)}) - \alpha(\mathbb{E}(\hat{\theta}^{(2)}) - \mathbb{E}(\tilde{\theta}^{(2)})) = \mathbb{E}(\hat{\theta}^{(1)}) - \alpha(\theta^{(2)} - \theta^{(2)}) = \theta^{(1)},$$

the estimator (4.14) is an unbiased estimator of $\theta^{(1)}$, therefore, as with the CVKM method, the MSE of the CVEM method (MSE_{CVEM}) is $\text{MSE}_{\text{CVEM}} = \mathbb{V}(\hat{\theta}^{(1)}_{\text{CVEM}})$. When using the CVEM method we select a value of k for estimating $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$, then calculate the k' and α values based on k. In order to find the optimal values of k' and α which minimise the MSE, we must express MSE_{CVEM} in terms of k, k', and α. Given that

$$\text{MSE}_{\text{CVEM}} = \mathbb{V}(\hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \tilde{\theta}^{(2)}))$$

and the results [109]

$$\mathbb{V}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \mathbb{V}(X_i) + 2\sum_{i<j} \text{Cov}(X_i, X_j) \quad \text{and} \quad \mathbb{V}(aX) = a^2\mathbb{V}(X)$$

we have

$$\begin{aligned}\text{MSE}_{\text{CVEM}} = {} & \mathbb{V}(\hat{\theta}^{(1)}) + \alpha^2\mathbb{V}(\hat{\theta}^{(2)}) + \alpha^2\mathbb{V}(\tilde{\theta}^{(2)}) - 2\alpha\text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) \\ & + 2\alpha\text{Cov}(\hat{\theta}^{(1)}, \tilde{\theta}^{(2)}) - 2\alpha^2\text{Cov}(\hat{\theta}^{(2)}, \tilde{\theta}^{(2)}). \end{aligned} \tag{4.15}$$

From (3.3) and (4.8) we know

$$\mathbb{V}(\hat{\theta}^{(1)}) = \frac{\mathbb{V}(f^{(1)})}{k} \qquad \mathbb{V}(\hat{\theta}^{(2)}) = \frac{\mathbb{V}(f^{(2)})}{k}, \quad \text{and} \quad \text{Cov}(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) = \frac{\text{Cov}(f^{(1)}, f^{(2)})}{k}.$$

We next examine $\text{Cov}(\hat{\theta}^{(n)}, \tilde{\theta}^{(2)})$ where $n = 1, 2$,

$$\text{Cov}(\hat{\theta}^{(n)}, \tilde{\theta}^{(2)}) = \mathbb{E}((\hat{\theta}^{(n)} - \mathbb{E}(\hat{\theta}^{(n)}))(\tilde{\theta}^{(2)} - \mathbb{E}(\tilde{\theta}^{(2)}))) \tag{4.16}$$

$$= \mathbb{E}(\hat{\theta}^{(n)}\tilde{\theta}^{(2)}) - \theta^{(n)}\theta^{(2)}. \tag{4.17}$$

Focusing on $\mathbb{E}(\hat{\theta}^{(n)}\tilde{\theta}^{(2)})$,

$$\mathbb{E}(\hat{\theta}^{(n)}\tilde{\theta}^{(2)}) = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} f^{(n)}_{X_i}\frac{1}{k'}\sum_{j=1}^{k'} f^{(2)}_{Y_j}\right)$$

$$= \frac{1}{kk'} \sum_{i=1}^{k} \sum_{j=1}^{k'} \mathbb{E}\left(f_{X_i}^{(n)} f_{Y_j}^{(2)}\right).$$

The samples of $f^{(2)}$ are sampled independently of $f^{(n)}$, therefore

$$\begin{aligned} \mathbb{E}(\hat{\theta}^{(n)}\tilde{\theta}^{(2)}) &= \frac{1}{kk'} kk' \, \mathbb{E}(f_{X_i}^{(n)}) \, \mathbb{E}(f_{Y_j}^{(2)}) \\ &= \theta^{(n)}\theta^{(2)}. \end{aligned} \tag{4.18}$$

Substituting (4.18) into (4.17) gives us $\mathrm{Cov}(\hat{\theta}^{(n)}, \tilde{\theta}^{(2)}) = 0$. Therefore we have

$$\mathrm{MSE}_{\mathrm{CVEM}} = \frac{\mathbb{V}(f^{(1)})}{k} + \alpha^2 \frac{\mathbb{V}(f^{(2)})}{k} - 2\alpha \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} + \alpha^2 \frac{\mathbb{V}(f^{(2)})}{k'}. \tag{4.19}$$

To find the optimal α_* that minimises $\mathrm{MSE}_{\mathrm{CVEM}}$ we take the first partial derivative of $\mathrm{MSE}_{\mathrm{CVEM}}$ w.r.t. α,

$$\begin{aligned} \frac{\partial \mathrm{MSE}_{\mathrm{CVEM}}}{\partial \alpha} &= 2\alpha \frac{\mathbb{V}(f^{(2)})}{k} - 2\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} + 2\alpha \frac{\mathbb{V}(f^{(2)})}{k'} \\ &= \frac{2\alpha}{k}\left(\mathbb{V}(f^{(2)}) + \frac{\mathbb{V}(f^{(2)})k}{k'}\right) - \frac{2}{k}\mathrm{Cov}(f^{(1)}, f^{(2)}). \end{aligned}$$

Setting $\frac{\partial \mathrm{MSE}_{\mathrm{CVEM}}}{\partial \alpha}$ equal to zero and rearranging gives the optimal value of α that minimises $\mathrm{MSE}_{\mathrm{CVEM}}$:

$$\alpha = \alpha_* = \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)\left(\frac{1}{1 + k/k'}\right). \tag{4.20}$$

We can now find $\mathrm{MSE}_{\mathrm{CVEM}}$ when $\alpha = \alpha_*$ using (4.19):

$$\begin{aligned} \mathrm{MSE}_{\mathrm{CVEM}} = \frac{\mathbb{V}(f^{(1)})}{k} &+ \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)^2 \left(\frac{1}{1 + k/k'}\right)^2 \frac{\mathbb{V}(f^{(2)})}{k} \\ &- 2\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)\left(\frac{1}{1 + k/k'}\right)\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} \\ &+ \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)^2 \left(\frac{1}{1 + k/k'}\right)^2 \frac{\mathbb{V}(f^{(2)})}{k'} \end{aligned}$$

$$
\begin{aligned}
&= \frac{\mathbb{V}(f^{(1)})}{k} + \left(\frac{1}{1+k/k'}\right)^2 \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)^2 \left(\frac{\mathbb{V}(f^{(2)})}{k} + \frac{\mathbb{V}(f^{(2)})}{k'}\right) \\
&\quad - 2\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)\left(\frac{1}{1+k/k'}\right)\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} \\
&= \frac{\mathbb{V}(f^{(1)})}{k} + \left(\frac{1}{1+k/k'}\right)\left(\frac{k'}{k'+k}\right)\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(2)})^2}\right)\left(\frac{k'\mathbb{V}(f^{(2)})}{kk'} + \frac{k\mathbb{V}(f^{(2)})}{kk'}\right) \\
&\quad - 2\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\right)\left(\frac{1}{1+k/k'}\right)\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} \\
&= \frac{\mathbb{V}(f^{(1)})}{k} + \left(\frac{1}{1+k/k'}\right)\left(\frac{1}{k}\right)\left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2 - 2\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(2)})}\right) \\
&= \frac{\mathbb{V}(f^{(1)})}{k}\left(1 - \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})}\left(\frac{1}{1+k/k'}\right)\right).
\end{aligned}
$$

Therefore $\mathrm{MSE}_{\mathrm{CVEM}}$ when $\alpha = \alpha_*$ is

$$
\mathrm{MSE}_{\mathrm{CVEM}} = \frac{\mathbb{V}(f^{(1)})}{k}\left(1 - \rho^2\left(\frac{1}{1+k/k'}\right)\right) \tag{4.21}
$$

where ρ is the correlation coefficient between $f^{(1)}$ and $f^{(2)}$. In practice ρ is unknown so must be estimated using prior experiments.

To select k' for a given k, the generalised mean-squared error is introduced. Measures such as GMSE are defined, as in [56], as the product of computing effort and MSE. As the MSE is normally inversely proportional to the budget, the GMSE can be considered a fundamental measure of simulation efficiency [126]. For the CVEM method, the GMSE is the product of the MSE and the budget required (P) for the CVEM method. The cost of the CVEM method is the cost of k evaluations of $f^{(1)}$ and $f^{(2)}$ and k' evaluations of $f^{(2)}$. If we assume the cost of a single evaluation of $f^{(1)}$ is 1 and the cost of $f^{(2)}$ is c, the total cost of the CVEM method is

$$
P = ck' + (1+c)k. \tag{4.22}
$$

The GMSE is therefore

$$
g(\alpha, k, k') = \mathrm{MSE}_{\mathrm{CVEM}}(\alpha, k, k')(ck' + (1+c)k),
$$

and we wish to minimise g with respect to α and k'. As P is not a function of α,

$$\frac{\partial g}{\partial \alpha} = \frac{\partial \text{MSE}_{\text{CVEM}}}{\partial \alpha}(ck' + (1+c)k)$$

so the optimal α_* in (4.20) also produces the optimal GMSE. Next we differentiate g with respect to k' to get

$$\begin{aligned}
\frac{\partial g}{\partial k'} &= \frac{\partial \text{MSE}_{\text{CVEM}}}{\partial k'} P + c\text{MSE}_{\text{CVEM}} \\
&= -\frac{\text{Cov}(f^{(1)}, f^{(2)})^2 (ck' + (1+c)k)}{\mathbb{V}(f^{(2)})(k+k')^2} + \frac{c\left[\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})(k+k') - \text{Cov}(f^{(1)}, f^{(2)})^2 k'\right]}{\mathbb{V}(f^{(2)})(k+k')k} \\
&= \frac{-\text{Cov}(f^{(1)}, f^{(2)})^2\, k(ck' + (1+c)k)}{\mathbb{V}(f^{(2)})(k+k')^2 k} \\
&\quad + \frac{c\left[\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})(k+k')^2 - \text{Cov}(f^{(1)}, f^{(2)})^2 k'(k+k')\right]}{\mathbb{V}(f^{(2)})(k+k')^2 k}.
\end{aligned}$$

Substituting $\rho^2 = \frac{\text{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(1)})\mathbb{V}(f^{(2)})}$ into $\frac{\partial g}{\partial k'}$ and setting the result equal to zero gives the quadratic function

$$[c(\rho^2 - 1)](k')^2 + [2kc(\rho^2 - 1)]k' + [k^2(\rho^2 - 1)c + \rho^2 k^2] = 0.$$

The positive root gives

$$\begin{aligned}
k' &= \frac{-2kc(\rho^2-1) + \sqrt{(2kc(\rho^2-1))^2 - 4c(\rho^2-1)(k^2(c(\rho^2-1)+\rho^2))}}{2c(\rho^2-1)} \\
&= \frac{-2kc(\rho^2-1) + \sqrt{4k^2c^2(\rho^2-1)^2 - 4k^2c(\rho^2-1)(c(\rho^2-1)+\rho^2)}}{2c(\rho^2-1)} \\
&= \frac{-2kc(\rho^2-1) + \sqrt{-4k^2c(\rho^2-1)\rho^2}}{2c(\rho^2-1)} \\
&= \frac{-2kc(\rho^2-1) + 2k\sqrt{c\rho^2(1-\rho^2)}}{2c(\rho^2-1)} \\
&= k\left(-1 + \sqrt{\frac{\rho^2}{1-\rho^2}\frac{1}{c}}\right).
\end{aligned}$$

As k' must be greater than zero, the optimal value of k' that minimises the GMSE

is

$$k'_* = k \times \max\{0, \beta - 1\} \tag{4.23}$$

where

$$\beta = \left[\left(\frac{\rho^2}{1-\rho^2} \right) \left(\frac{1}{c} \right) \right]^{\frac{1}{2}}. \tag{4.24}$$

Finally, we use the optimal value of k' to express MSE_{CVEM} as a function of k only. Defining $\phi = \max\{0, \beta - 1\}$, we have $k' = k\phi$, which can be substituted into (4.21) to give the MSE_{CVEM} when $\alpha = \alpha_*$ and $k' = k'_*$ as

$$\text{MSE}_{\text{CVEM}} = \frac{\mathbb{V}(f^{(1)})}{k} \left(1 - \rho^2 \left(\frac{\phi}{\phi + 1} \right) \right). \tag{4.25}$$

Note that when $\phi = 0$, the chosen approximation to the high-fidelity model is not an appropriate control variate, so the value of MSE_{CVEM} is equivalent to MSE_{MC}. This indicates that no sufficiently correlated or inexpensive control variate is available for the CVEM method to provide a variance reduction. We can further rewrite MSE_{CVEM} w.r.t. P by rearranging (4.22) and substituting into (4.25) giving

$$\text{MSE}_{\text{CVEM}} = \frac{\mathbb{V}(f^{(1)})}{P}(1 + c(1 + \phi)) \left(1 - \rho^2 \left(\frac{\phi}{\phi + 1} \right) \right). \tag{4.26}$$

The steps of the CVEM method are outlined in Algorithm 4 in Appendix A.

We now estimate the mean of the Ishigami function using CVEM, and compare the computed MSE_{CVEM} to the theoretical MSE_{CVEM}. In [67] we are given costs for the approximations to the Ishigami function (from §4.1) relative to $f^{(1)}$ with $c = 0.05$ for $f^{(2)}$ and $c = 0.001$ for $f^{(3)}$. To compare the computed MSE_{CVEM} to the theoretical MSE_{CVEM} we first calculate the theoretical MSE_{CVEM} for five values of k using (4.25) and the known values given in [67]. We then calculate $R = 100$ replications of estimates of the mean using CVEM Algorithm 4 and compare to MSE_{MC} of the standard Monte Carlo estimate of the mean. In Figure 4.2 we can see the theoretical and calculated values of MSE_{CVEM} are similar using both control variates, and both are an improvement over MSE_{MC}. The improvement in MSE_{CVEM}

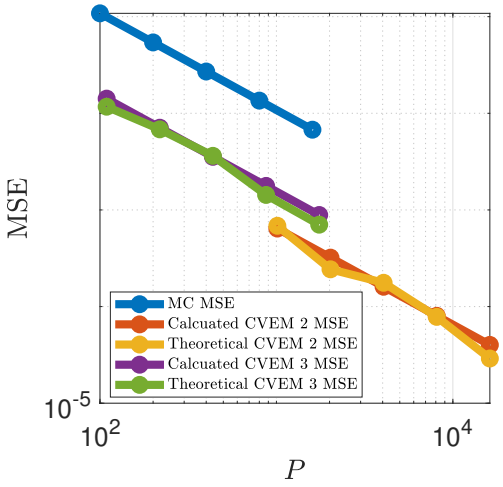


Figure 4.2: Comparison of the theoretical and calculated MSE_{CVEM} of the mean of the Ishigami function.

using the highly correlated approximation $f^{(2)}$ control variate is better, but comes at significantly increased cost.

Next we compare the MSE_{CVEM} of the Ishigami mean to the MSE_{CVKM} by calculating 100 replicates of the estimate of the mean using the known values for $f^{(2)}$ and $f^{(3)}$ and the method in Algorithm 3, and compare to MSE_{CVEM} of $R = 100$ CVEM estimates. In Figure 4.3 we can see using the less highly correlated control

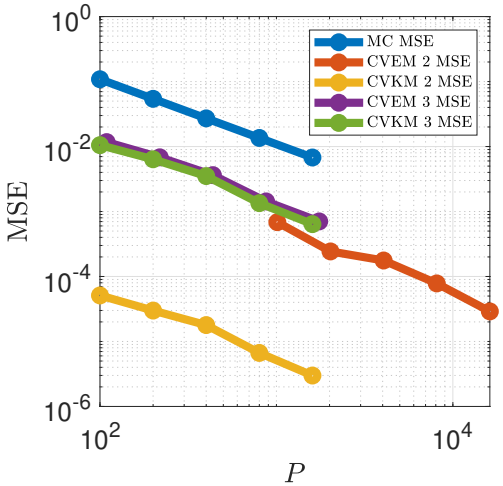


Figure 4.3: Comparison of MSE_{MC}, MSE_{CVKM}, and MSE_{CVEM} for the estimated mean of the Ishigami function using control variates $f^{(2)}$ and $f^{(3)}$.

variate $f^{(3)}$, the difference in cost and MSE of the CVKM and CVEM estimates of the mean is minimal, however when the approximation is very highly correlated in $f^{(2)}$, the cost of the CVEM method increases significantly and the improvement in MSE is reduced significantly by not knowing the true mean. This is due to the low value of $1-\rho^2$ which leads to a large value of β (see equation (4.24)), and a very large value of k' (see equation (4.23)). As the correlation ρ approaches one, $1-\rho^2$ becomes small, leading to a large value of β (see equation (4.24)). A larger β requires a large number of samples k' (see equation (4.23)). The additional sampling cost from estimating the control variate mean offsets much of the variance reduction gained from the high correlation, leading to only marginal improvements in MSE at substantially increased cost.

4.3 Adding a budget control to CVEM

One of the issues with the CVEM method is that given a value for k, the number of samples needed to estimate the control variate mean can become very large, increasing the overall budget required to improve the mean estimate. This problem can be seen in Figure 4.3 where the CVEM estimate requires a lot of evaluations of $f^{(2)}$, increasing P significantly. Typically, this will conflict with the practical constraint of having a fixed computational budget available. In this section we look at methods to bound the cost of the CVEM method by a budget constraint, where the budget P is fixed and the optimal values of k and k' which minimise the MSE can be calculated. To begin, we recall that the budget required to use the CVEM method is

$$P = ck' + (1+c)k \tag{4.27}$$

where, as before, c is the cost of $f^{(2)}$ relative to $f^{(1)}$. We define $\phi = \max\{0, \beta - 1\}$ with β as in (4.24). We can then define $k' = k\phi$ as in (4.23) and rearrange (4.27) to

get

$$k = \frac{P}{1 + c + c\phi}. \tag{4.28}$$

This gives a value of k for a fixed P value, which can then be used as before to calculate the optimal values of k' and α using (4.23) and (4.20). As in the standard control variate setting described in Section 4.1, the parameter α controls the strength of the variance-reducing correction, and is chosen to minimise the variance of the resulting estimator. By substituting k into (4.25) we can find the MSE of the control variates with constrained budget and estimated means (CVB) method to be

$$\text{MSE}_{\text{CVB}} = \frac{\mathbb{V}(f^{(1)})}{P}(1 + c(1 + \phi)) \left(1 - \rho^2 \left(\frac{\phi}{1 + \phi}\right)\right). \tag{4.29}$$

Note that (4.29) and (4.26) are the same, so $\text{MSE}_{\text{CVB}} \equiv \text{MSE}_{\text{CVEM}}$.

An alternative way to constrain the cost of the CVEM method is using Lagrange multipliers. Suppose we now want to minimise (4.25) subject to

$$k'c + k(1 + c) = P. \tag{4.30}$$

We can do this by introducing the constraint

$$g(k, k') = k'c + k(1 + c) - P = 0$$

to obtain the new objective function from (4.19),

$$\begin{aligned}\hat{E}(k, k', \alpha, \lambda) &= \text{MSE}_{\text{CVEM}} + \lambda g(k, k') \\ &= \frac{\mathbb{V}(f^{(1)})}{k} + \alpha^2 \frac{\mathbb{V}(f^{(2)})}{k} - 2\alpha \frac{\text{Cov}(f^{(1)}, f^{(2)})}{k} + \alpha^2 \frac{\mathbb{V}(f^{(2)})}{k'} + \lambda(k'c + k(1 + c) - P).\end{aligned}$$

We seek optimal values for each of the unknowns by finding the partial derivatives

of $\hat{E}$. The partial derivatives are

$$\begin{aligned}
\frac{\partial \hat{E}}{\partial \alpha} &= 2\alpha\frac{\mathbb{V}(f^{(2)})}{k} + 2\alpha\frac{\mathbb{V}(f^{(2)})}{k'} - 2\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} \\
\frac{\partial \hat{E}}{\partial k} &= -\frac{\mathbb{V}(f^{(1)})}{k^2} - \alpha^2\frac{\mathbb{V}(f^{(2)})}{k^2} + \frac{2\alpha\mathrm{Cov}(f^{(1)}, f^{(2)})}{k^2} + \lambda(1+c), \\
\frac{\partial \hat{E}}{\partial k'} &= -\frac{\alpha^2\mathbb{V}(f^{(2)})}{(k')^2} + \lambda c, \\
\frac{\partial \hat{E}}{\partial \lambda} &= k'c + k(1+c) - P.
\end{aligned}$$

As $\frac{\partial \hat{E}}{\partial \alpha} = \frac{\partial \mathrm{MSE}_{\mathrm{CVEM}}}{\partial \alpha}$ we obtain $\alpha = \alpha_*$ as in (4.20). This leaves us with three unknowns, λ, k and k', and four equations which we set equal to zero:

$$\alpha\frac{\mathbb{V}(f^{(2)})}{k} + \alpha\frac{\mathbb{V}(f^{(2)})}{k'} - \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{k} = 0, \tag{4.31}$$

$$-\frac{\mathbb{V}(f^{(1)})}{k^2} - \alpha^2\frac{\mathbb{V}(f^{(2)})}{k^2} + \frac{2\alpha\mathrm{Cov}(f^{(1)}, f^{(2)})}{k^2} + \lambda(1+c) = 0, \tag{4.32}$$

$$-\frac{\alpha^2\mathbb{V}(f^{(2)})}{(k')^2} + \lambda c = 0, \tag{4.33}$$

$$k'c + k(1+c) - P = 0. \tag{4.34}$$

By rearranging (4.34) we find that

$$k' = \frac{P - k(1+c)}{c}. \tag{4.35}$$

Substituting k' into (4.31) gives

$$\begin{aligned}
\alpha &= \frac{\mathrm{Cov}(f^{(1)}f^{(2)})}{\mathbb{V}(f^{(2)})}\left(\frac{1}{1 + \frac{k}{\frac{P-k(1+c)}{c}}}\right) \\
&= \frac{\mathrm{Cov}(f^{(1)}f^{(2)})}{\mathbb{V}(f^{(2)})}\left(\frac{P - k(1+c)}{P-k}\right).
\end{aligned}$$

From (4.33) we have

$$\lambda = \frac{\alpha^2\mathbb{V}(f^{(2)})}{(k')^2 c},$$

and we can substitute k' and α to find

$$\begin{aligned}\lambda &= \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(2)})^2}\left(\frac{(P-k(1+c))^2}{(P-k)^2})\mathbb{V}(f^{(2)})\right) \Big/ \frac{(P-k(1+c))^2}{c^2}c \\ &= \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2 c}{\mathbb{V}(f^{(2)})(P-k)^2}.\end{aligned}$$

We now have three unknowns as a function of k so we only need to calculate k as a function of P. By substituting k', α, and λ into (4.32), we get

$$\begin{aligned}0 &= -\mathbb{V}(f^{(1)}) - \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(2)})^2}\left(\frac{(P-k(1+c))^2}{(P-k)^2}\right)\mathbb{V}(f^{(2)}) \\ &\quad + 2\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}\left(\frac{(P-k(1+c))}{(P-k)}\right)\mathrm{Cov}(f^{(1)}, f^{(2)})) \\ &\quad + \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})^2 c}{\mathbb{V}(f^{(2)})(P-k)^2}(1+c)k^2 \\ \implies 0 &= -1 - \rho^2\left(\frac{(P-k(1+c))^2}{(P-k)^2}\right) + 2\rho^2\left(\frac{(P-k(1+c))}{(P-k)}\right) + \rho^2\frac{c(1+c)k^2}{(P-k)^2} \\ \implies 0 &= -\frac{1}{\rho^2}(P-k)^2 - (P-k(1+c))^2 + 2(P-k(1+c))(P-k) + c(1+c)k^2 \\ &= c(1+c)k^2 - \frac{1}{\rho^2}(P^2 - 2Pk + k^2) - (P^2 - 2kP(1+c) + k^2(1+c)^2) \\ &\quad + 2(P^2 - kP - k(1+c)P + k^2(1+c)) = 0\end{aligned}$$

which can be rearranged into the quadratic equation

$$\begin{aligned}&k^2(c(1+c) - \frac{1}{\rho^2} - (1+c)^2 + 2(1+c)) \\ &+ k(2\frac{P}{\rho^2} + 2P(1+c) - 2(P + (1+c)P) + (\frac{-P^2}{\rho^2} - P^2 + 2P^2) = 0.\end{aligned}$$

Discarding the negative root, we can get a value of k for a given P from

$$k = P\left(\frac{(1-\frac{1}{\rho^2}) \pm \sqrt{(1-\frac{1}{\rho^2})(-c)}}{1-\frac{1}{\rho^2}+c}\right)$$

$$= P\left(\frac{1}{1 + c\sqrt{\frac{\rho^2}{\rho^2-1}\frac{1}{c}}}\right).$$

This gives the final expression

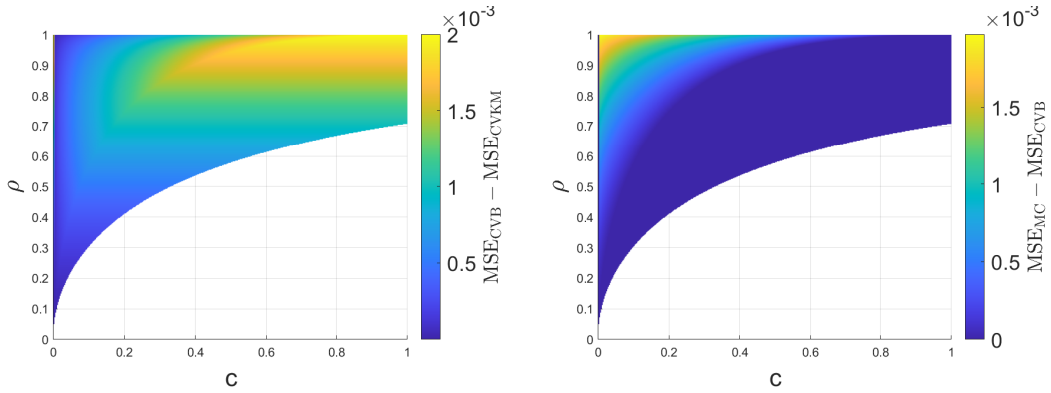
$$k = \frac{P}{1 + c\sqrt{\frac{\rho^2}{\rho^2-1}\frac{1}{c}}} = \frac{P}{1 + c(1+\phi)}, \tag{4.36}$$

which is the same as (4.28). Note that $\rho^2 \neq 1$, therefore any perfectly correlated approximation is not suitable for this method. After evaluating k we can then use k to calculate k' and α as in the CVEM method. The key steps of the CVB method to estimate the mean are presented in Algorithm 5 in Appendix A.

We wish to examine what is lost when we use the control variates method but do not know the mean of the approximation to $f^{(1)}$, as well as when the CVB method is not suitable due to $\phi = 0$. To do this, we compare the theoretical expressions for the MSE of the CVB and CVKM methods, derived in (4.29) and (4.9), for a range of cost and correlation values.

We fix $P = 1000$ and set $\mathbb{V}(f^{(1)}) = 2$, then define a grid of 1000 equally spaced values for c and ρ in $[0, 1]$. For each (c, ρ) pair, we compute the theoretical MSE values MSE_{CVKM} and MSE_{CVB} using the analytic expressions, and plot the difference $\text{MSE}_{\text{CVB}} - \text{MSE}_{\text{CVKM}}$. We also plot the difference between MSE_{CVB} and the standard Monte Carlo error MSE_{MC} for comparison.

In Figure 4.4(a) we can see that the CVB method has higher MSE than CVKM, with the largest difference occurring when the correlation and the cost of the approximation are high. The MSE of the CVB method approaches that of CVKM as both cost and correlation decrease. Over a large region of the (c, ρ) parameter space, the CVB method is not used in practice, as the optimal allocation gives $k' = 0$, corresponding to $\beta < 1$. The boundary of this region is given by the curve $\beta = 1$ in the (c, ρ) plane. In this case, the CVB estimator reduces to standard Monte Carlo and therefore provides no variance reduction. As a result, compared to standard Monte



(a) Plot of MSE_{CVEM} - MSE_{CVB} for all values of c and ρ.

(b) Plot of MSE_{MC} - MSE_{CVB} for all values of c and ρ.

Figure 4.4: Theoretical difference in mean estimate MSE between the CVB method and the CVKM and Monte Carlo methods.

Carlo, the CVB method yields a lower MSE only when the correlation is sufficiently high and the cost of the approximation is sufficiently low, so that $\beta \geq 1$. Increasing the total budget reduces the variance, and hence the MSE, of all estimators, but does not remove the relative disadvantage of the CVB method compared to CVKM, as the penalty from estimating the control variate mean persists at all budget levels.

We now use the Ishigami function and its control variates to test the CVB method. We first find $R = 100$ replicates of the mean using the CVB algorithm in Algorithm 5, and find the MSE of the replicates. We examine the theoretical MSE from (4.29), and the theoretical Monte Carlo MSE, at each P vs computed MSE_{CVB} using 5 values of $P = 100, 200, 400, 800, 1600$. In Figure 4.5 we can see the MSE_{CVB} of the estimate of the mean of the Ishigami function is similar to the theoretical MSE_{CVB} using both control variates. The theoretical MSE_{CVB} accurately predicts the experimental performance of the CVB method, and MSE_{CVB} improves upon MSE_{MC}.

Next we compare MSE_{CVB} against MSE_{CVEM} and MSE_{CVKM}. We find $R = 100$ estimates of the mean, for five values of $P = 100, 200, 400, 800, 1600$, using the CVKM, CVEM, and CVB methods and find the MSE of the mean estimates. We plot the computed MSE_{CVB}, MSE_{CVEM}, MSE_{CVKM}, and the theoretical MSE_{MC} for each P. In Figure 4.6 we can see plots of MSE_{MC}, MSE_{CVKM}, MSE_{CVEM} and MSE_{CVB}.

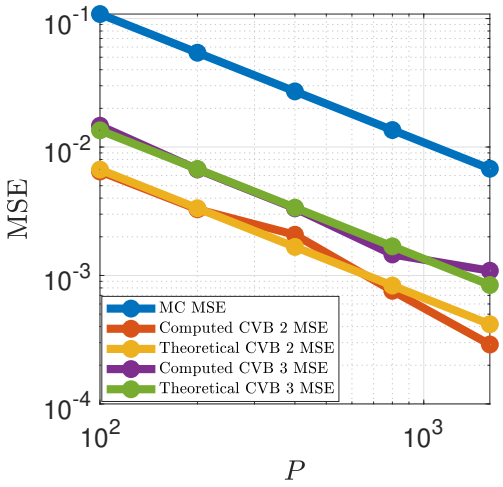
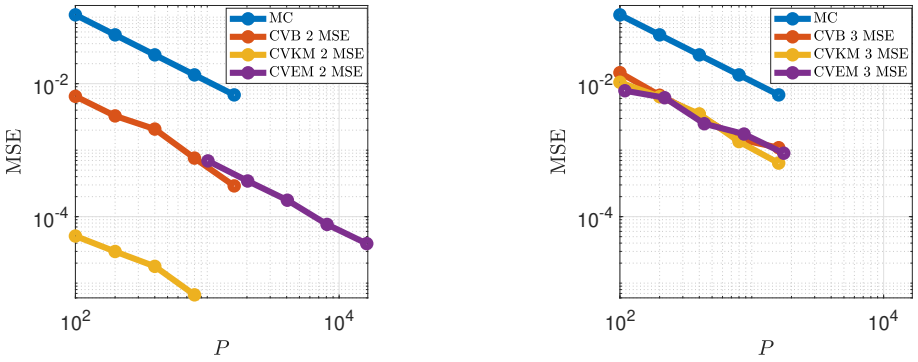


Figure 4.5: Comparison of the theoretical and computed MSE_{CVB} and the theoretical MSE_{MC} for the mean of the Ishigami function.



(a) MSE of estimates of the Ishigami function mean with $f^{(2)}$ as a control variate.

(b) MSE of estimates of the Ishigami function mean with $f^{(3)}$ as a control variate.

Figure 4.6: Comparison of the MSE_{MC}, MSE_{CVKM}, MSE_{CVEM} and MSE_{CVB} of the estimate of the mean of the Ishigami function.

For the more highly correlated, more expensive $f^{(2)}$, the increase in error caused by not knowing the mean of the control variate is large, however for the cheaper, less correlated $f^{(3)}$, almost nothing is lost by estimating the mean. The effect of not constraining the overall budget results in a much higher budget needed for the more correlated, more expensive $f^{(2)}$, whereas $f^{(3)}$ as a control variate does not need much additional budget when using the unbounded CVEM method.

4.4 Multi-fidelity Monte Carlo estimator

We now consider another estimator of the mean using multiple models, which was introduced in [104]. The multi-fidelity Monte Carlo (MFMC) method considers the use of approximations to accelerate Monte Carlo sampling. As with the CVEM method, this method does not require us to know the mean of the approximations to the model, but it does require us to know the correlations between the model $f^{(1)}$ and its approximations, similar to the CVEM and CVB methods, however these can be estimated. Suppose we have a model, $f^{(1)}$, and an approximation to the model, $f^{(2)}$. The MFMC method can use any number of approximation models but here we focus on the two-level method for comparison with the other methods of estimating the mean we have discussed. Let $\mathbf{m} = (m_1, m_2)^T$ be a vector with integer components $0 < m_1 < m_2$ where m_i is the number of evaluations of the model $f^{(i)}$. We assume that the cost of evaluating the models is $\mathbf{c} = (1, c)^T$. The reuse of the first m_1 input samples for both fidelity levels is not required for unbiasedness, but is necessary to exploit the correlation between models that underpins the variance reduction achieved by the method.

We require the variances of both models and their correlation coefficient ρ. Now let $\mathbf{X}$ be a vector of m_2 i.i.d. realisations of the random input variable $\mathbf{X} = (x_1, \ldots, x_{m_2})$. The first m_1 samples of $\mathbf{X}$ are used to obtain m_1 evaluations of the model $f^{(i)}$ for $i = 1, 2$, which we then use to compute the Monte Carlo estimates of the mean $\hat{\theta}^{(1)}_{m_1}$ and $\hat{\theta}^{(2)}_{m_1}$. We then use all m_2 samples of $\mathbf{X}$ to estimate the mean of the approximation model, $\hat{\theta}^{(2)}_{m_2}$. As $\hat{\theta}^{(2)}_{m_2}$ is calculated using m_1 of the same input samples

as $\hat{\theta}^{(2)}_{m_1}$, the two approximation mean estimates are correlated. The two-level MFMC estimate of the mean $\hat{\theta}^{(1)}_{\text{MFMC}}$ is then evaluated as

$$\hat{\theta}^{(1)}_{\text{MFMC}} = \hat{\theta}^{(1)}_{m_1} + \alpha_2(\hat{\theta}^{(2)}_{m_2} - \hat{\theta}^{(2)}_{m_1}), \tag{4.37}$$

where α_2 is a weighting coefficient. The MFMC method of estimating the mean is an unbiased estimator of $\theta^{(1)}$ as

$$\begin{aligned}\mathbb{E}(\hat{\theta}^{(1)}_{\text{MFMC}}) &= \mathbb{E}(\hat{\theta}^{(1)}_{m_1}) + \mathbb{E}(\alpha_2(\hat{\theta}^{(2)}_{m_2} - \hat{\theta}^{(2)}_{m_1})) \\ &= \mathbb{E}(\hat{\theta}^{(1)}_{m_1}) + \alpha_2(\mathbb{E}(\hat{\theta}^{(2)}_{m_2}) - \mathbb{E}(\hat{\theta}^{(2)}_{m_1})) = \theta^{(1)}.\end{aligned}$$

As the estimator is unbiased, the MSE of $\hat{\theta}^{(1)}_{\text{MFMC}}$ (MSE_{MFMC}) is $\mathbb{V}(\hat{\theta}^{(1)}_{\text{MFMC}})$, therefore we can find MSE_{MFMC} by finding the variance of (4.37):

$$\begin{aligned}\text{MSE}_{\text{MFMC}} &= \mathbb{V}(\hat{\theta}^{(1)}_{m_1}) + \mathbb{V}(\alpha_2(\hat{\theta}^{(2)}_{m_2} - \hat{\theta}^{(2)}_{m_1})) \\ &= \mathbb{V}(\hat{\theta}^{(1)}_{m_1}) + \alpha_2^2\mathbb{V}(\hat{\theta}^{(2)}_{m_2}) + \alpha_2^2\mathbb{V}(\hat{\theta}^{(2)}_{m_1}) \\ &+ 2\alpha_2\text{Cov}(\hat{\theta}^{(1)}_{m_1}, \hat{\theta}^{(2)}_{m_2}) - 2\alpha_2\text{Cov}(\hat{\theta}^{(1)}_{m_1}, \hat{\theta}^{(2)}_{m_1}) - 2\alpha_2^2\text{Cov}(\hat{\theta}^{(2)}_{m_2}, \hat{\theta}^{(2)}_{m_1}).\end{aligned} \tag{4.38}$$

Focusing on the covariance term $\text{Cov}(\hat{\theta}^{(1)}_{m_1}, \hat{\theta}^{(2)}_{m_2})$ from (4.38), we consider the general form $\text{Cov}(\hat{\theta}^{(l)}_{m_1}, \hat{\theta}^{(t)}_{m_2})$, where $1 \leq l, t \leq 2$, We know $m_2 > m_1$ by definition, and

$$\text{Cov}(\hat{\theta}^{(l)}_{m_1}, \hat{\theta}^{(t)}_{m_2}) = \frac{1}{m_1 m_2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_2}\text{Cov}(f^{(l)}(X_i), f^{(t)}(X_j)),$$

where X_i and X_j are independent and identically distributed random variables. When $i \neq j$ the covariance is 0, and as $m_2 > m_1$,

$$\begin{aligned}\text{Cov}(\hat{\theta}^{(l)}_{m_1}, \hat{\theta}^{(t)}_{m_2}) &= \frac{1}{m_1 m_2} m_1 \text{Cov}(f^{(l)}, f^{(t)}) \\ &= \frac{\text{Cov}(f^{(l)}, f^{(t)})}{m_2}.\end{aligned} \tag{4.39}$$

Using (4.39) with $l = 1$, $t = 2$ and $m_2 = m_1$, we have

$$\text{Cov}(\hat{\theta}_{m_1}^{(1)}, \hat{\theta}_{m_1}^{(2)}) = \frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_1} = \frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_1}. \tag{4.40}$$

Here we use $\text{Cov}(\hat{\theta}_{m_1}^{(1)}, \hat{\theta}_{m_1}^{(2)}) = \text{Cov}(f^{(1)}, f^{(2)})/m_1$, as both estimators are formed from the same m_1 paired samples and observations are independent across samples. We can substitute the results in (4.39) and (4.40) into (4.38) to find MSE_{MFMC} as

$$\begin{aligned}
\text{MSE}_{\text{MFMC}} &= \frac{\mathbb{V}(f^{(1)})}{m_1} + \alpha_2^2\frac{\mathbb{V}(f^{(2)})}{m_2} + \alpha_2^2\frac{\mathbb{V}(f^{(2)})}{m_1} \\
&\quad + 2\alpha_2\frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_2} - 2\alpha_2\frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_1} - 2\alpha_2^2\frac{\mathbb{V}(f^{(2)})}{m_2} \\
&= \frac{\mathbb{V}(f^{(1)})}{m_1} + \alpha_2^2\frac{\mathbb{V}(f^{(2)})}{m_1} - 2\alpha_2\frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_1} \\
&\quad - \alpha_2^2\frac{\mathbb{V}(f^{(2)})}{m_2} + 2\alpha_2\frac{\text{Cov}(f^{(1)}, f^{(2)})}{m_2} \\
&= \frac{\mathbb{V}(f^{(1)})}{m_1} + \left(\frac{1}{m_1} - \frac{1}{m_2}\right)(\alpha_2^2\mathbb{V}(f^{(2)}) - 2\alpha_2\text{Cov}(f^{(1)}, f^{(2)})).
\end{aligned} \tag{4.41}$$

The two-level MFMC requires m_1 evaluations of $f^{(1)}$ and m_2 evaluations of $f^{(2)}$, so the cost is

$$\mathbf{c}^T\mathbf{m} = m_1 + c(m_2). \tag{4.42}$$

As the MFMC method has a capped budget, P say, we can now find the optimal values of α_2, and number of evaluations ($\mathbf{m}$) that minimises MSE_{MFMC} subject to this budget constraint. This means we have an optimisation problem where we wish to minimise MSE_{MFMC} subject to

$$0 < m_1 < m_2, \qquad \text{and} \qquad m_1 + c(m_2) = P.$$

We consider the Lagrangian

$$\hat{E}(m_1, m_2, \alpha_2, \lambda) = \frac{\mathbb{V}(f^{(1)})}{m_1} + \left(\frac{1}{m_1} - \frac{1}{m_2}\right)(\alpha_2^2\mathbb{V}(f^{(2)}) - 2\alpha_2\text{Cov}(f^{(1)}, f^{(2)})) + \lambda(m_1 + cm_2 - P)).$$

We find the partial derivatives w.r.t. m_1, m_2, α_2, and λ and set each equal to 0 which gives

$$\frac{\partial \hat{E}}{\partial m_1} = -\frac{\mathbb{V}(f^{(1)})}{m_1^2} - \frac{1}{m_1^2}(\alpha_2^2 \mathbb{V}(f^{(2)}) - 2\alpha_2 \text{Cov}(f^{(1)}, f^{(2)})) + \lambda = 0, \tag{4.43}$$

$$\frac{\partial \hat{E}}{\partial m_2} = \frac{1}{m_2^2}(\alpha_2^2 \mathbb{V}(f^{(2)}) - 2\alpha_2 \text{Cov}(f^{(1)}, f^{(2)})) + \lambda c = 0, \tag{4.44}$$

$$\frac{\partial \hat{E}}{\partial \alpha_2} = \left(\frac{1}{m_1} - \frac{1}{m_2}\right)(2\alpha_2 \mathbb{V}(f^{(2)}) - 2\text{Cov}(f^{(1)}, f^{(2)})) = 0, \tag{4.45}$$

$$\frac{\partial \hat{E}}{\partial \lambda} = m_1 + cm_2 - P = 0. \tag{4.46}$$

By rearranging (4.45) we obtain

$$\alpha_2 = \frac{\text{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})},$$

which can then be substituted into (4.43) to get

$$\begin{aligned}\lambda &= \frac{\mathbb{V}(f^{(1)})}{m_1^2} - \frac{1}{m_1^2}\left(\frac{\text{Cov}(f^{(1)}, f^{(2)})^2}{\mathbb{V}(f^{(2)})}\right) \\ &= \frac{\mathbb{V}(f^{(1)})(1-\rho^2)}{m_1^2}.\end{aligned}$$

Substituting α_2 and λ into (4.44) then gives

$$\begin{aligned} 0 &= -\frac{\mathbb{V}(f^{(1)})\rho^2}{m_2^2} + \frac{\mathbb{V}(f^{(1)})(1-\rho^2)}{m_1^2}c \\ \implies 0 &= -m_1^2\rho^2 + m_2^2(1-\rho^2)c) \\ \implies m_2 &= \pm m_1 \sqrt{\frac{\rho^2}{1-\rho^2}\frac{1}{c}}.\end{aligned}$$

We discard the negative root as the constraints define $m_2 > m_1 > 0$, and introduce the constraint $\rho^2 < 1$. If we define $r_2 = \sqrt{\frac{\rho^2}{1-\rho^2}\frac{1}{c}}$, then substitute m_2 into (4.46) we

get

$$\begin{aligned} m_1 + cm_1 r_2 - P &= 0 \\ \implies m_1 &= \frac{P}{1 + cr_2}. \end{aligned}$$

This gives the final MSE_{MFMC} optimal values as

$$m_1 = \frac{P}{1 + cr_2}, \tag{4.47}$$

$$m_2 = r_2 m_1, \tag{4.48}$$

$$\alpha_2 = \frac{\text{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}, \tag{4.49}$$

where

$$r_2 = \sqrt{\frac{\rho^2}{1 - \rho^2} \frac{1}{c}}. \tag{4.50}$$

The key steps of the two-level MFMC method are outlined in Algorithm 6 in Appendix A. It can be noted from (4.24) and (4.50) that r_2 here is equal to β in the CVB method. Despite the MFMC method being derived differently from the control variates with estimated means method, it can be shown that the MSE of the two methods are in fact equal. By expressing the low-fidelity mean $\hat{\theta}_{m_2}^{(2)}$ in terms of the shared low-fidelity samples and the additional low-fidelity samples, the two-level MFMC estimator can be written in a form analogous to a control variate estimator with an estimated mean. In this interpretation, the additional low-fidelity samples provide an estimate of the control variate mean. Under a fixed computational budget, the optimal sample allocation leads to $r_2 = \beta$, and substituting this relation into the corresponding mean-squared error expressions shows that the two methods attain the same mean-squared error in the two-level setting.

Examining MSE_{MFMC}, we start by substituting (4.47), (4.48), and (4.49) into

(4.41) to find

$$\begin{aligned}\mathrm{MSE}_{\mathrm{MFMC}} &= \frac{\mathbb{V}(f^{(1)})(1+cr_2)}{P} + \left(\frac{1+cr_2}{P} - \frac{1+cr_2}{r_2 P}\right)(-\mathbb{V}(f^{(1)})\rho^2) \\ &= \frac{\mathbb{V}(f^{(1)})}{P}(1+cr_2)\left(1-\rho^2\left(1-\frac{1}{r_2}\right)\right).\end{aligned}$$

As $r_2 = \beta$ in (4.24), it follows that $r_2 = \phi + 1$ when $r_2 > 1$, so from (4.36) we can see $m_1 = k$. Therefore $\mathrm{MSE}_{\mathrm{MFMC}}$ is

$$\mathrm{MSE}_{\mathrm{MFMC}} = \frac{\mathbb{V}(f^{(1)})}{P}(1+c(1+\phi))\left(1-\rho^2\left(\frac{\phi}{1+\phi}\right)\right) \tag{4.51}$$

which is equal to (4.29), so for a given budget P, the CVB and the two-level MFMC method have the same theoretical MSE. The MFMC framework naturally extends to more than two fidelity levels. Additional levels can provide further variance reduction if the corresponding models are both substantially cheaper and sufficiently correlated with the high-fidelity model. However, the benefits of introducing additional levels are problem dependent and must be balanced against increased implementation complexity and the need to estimate multiple correlations reliably. In this thesis, the two-level formulation is used to enable direct comparison with the control variate methods considered earlier. As noted earlier, the two-level MFMC estimator is equivalent to the CVB estimator under the present formulation. The theoretical MSE and numerical results obtained for CVB therefore apply directly to MFMC, so separate experiments are not included.

4.5 Bootstrapping and confidence intervals with control variates

Previously we examined the method of bootstrapping as a method of quantifying the error in the mean estimation. We now look to calculate 95% confidence intervals while using the CVB method. To do this, we begin by using (4.28) and (4.23) to

choose values of k and k'. We then sample k values of X_i and k' samples of Y_i to get samples of $f_{X_i}^{(1)}, f_{X_i}^{(2)}$, and $f_{Y_i}^{(2)}$, allowing us to calculate $\hat{\theta}^{(1)}, \hat{\theta}^{(2)}$ and $\tilde{\theta}^{(2)}$. Next we calculate the optimal values of β and α_* from (4.24) and (4.20) and calculate a single realisation of $\hat{\theta}_{\text{CVB}}^{(1)}$. We then resample $\hat{\theta}^{(1)}, \hat{\theta}^{(2)}$ and $\tilde{\theta}^{(2)}$ from the original samples as before (with replacement), making sure that the sample of $f_{X_i}^{(1)}$ being drawn is the result of the same X_i as $f_{X_i}^{(2)}$ to ensure proper correlation. It is important to note that the pilot experiment estimate of the mean in the CVB method ($\tilde{\theta}^{(2)}$) is a random variable, so it must be resampled with replacement for every estimate of $\hat{\theta}_{\text{CVB}}^{(1)}$, for otherwise the static $\tilde{\theta}^{(2)}$ will result in a sampling error and the $1-\gamma$ confidence intervals will not cover the true mean $100(1-\gamma)\%$ of the time. We resample each component of the CVB method R_b times, calculating $\hat{\theta}_{\text{CVB}}^{(1)}$ each time using (4.14) and obtain a list of R_b estimates of the mean, from which we can take the $\gamma/2$ and $1-\gamma/2$ quantiles to find the confidence intervals. As we consider the cost of resampling to be negligible, the budget required to calculate the confidence intervals using the CVB method is $P = ck' + (1+c)k$. The key steps of the method are presented in Algorithm 7 (see Appendix A).

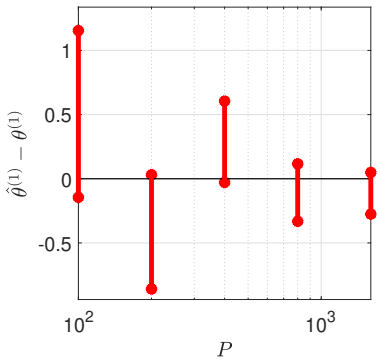
We begin by checking that the CVB confidence intervals do cover the true mean $100(1-\gamma)\%$ of the time. By finding 1000 realisations of the CVB confidence intervals of the Ishigami function mean, we can check that they cover true mean as expected for the two different control variates. In Table 4.1 we can see the 95% confidence

Sample Size	**k = 100**	**k = 200**	**k = 400**	**k = 800**	**k = 1600**
$f^{(2)}$ **95% Coverage %**	96	93.9	95.1	94.4	95.4
$f^{(3)}$ **95% Coverage %**	94.3	94.7	95.3	94.7	94.1

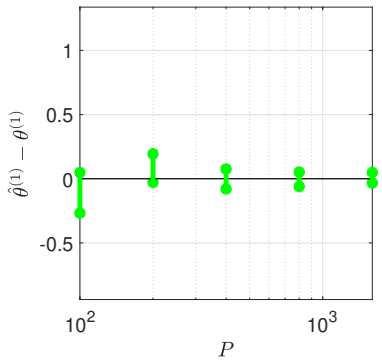
Table 4.1: Observed coverage for confidence intervals of $\theta^{(1)}$ estimated using the CVB method with increasing sample size. Figures based on 1000 replications of the confidence intervals.

intervals cover the true mean approximately 95% of the time as expected for both approximations to the Ishigami function as control variables. We can therefore be confident the CVB confidence intervals are robust and can be used going forward. Now we compare the confidence intervals estimated using standard Monte Carlo

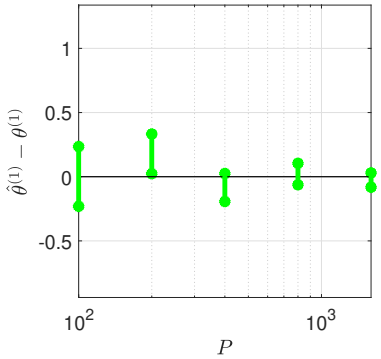
and resampling to those estimated using CVB and resampling. In Figure 4.7 the



(a) 95% confidence intervals of the mean of the Ishigami functions calculated using Monte Carlo.



(b) 95% confidence intervals of the mean of the Ishigami functions calculated using CVB with control variate $f^{(2)}$.



(c) 95% confidence intervals of the mean of the Ishigami functions calculated using CVB with control variate $f^{(3)}$.

Figure 4.7: Comparison of the confidence intervals of the Ishigami function with two different control variates.

confidence intervals are narrower using CVB for each given budget P so the CVB method of calculating confidence intervals can introduce a significant improvement without any additional cost. The more highly correlated $f^{(2)}$ control variate improves the confidence intervals more than the the less correlated $f^{(3)}$ but both result in significant improvement.

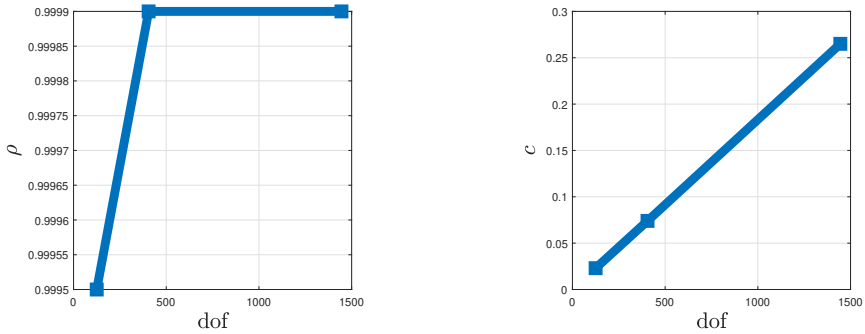
4.6 Multi-fidelity approach for estimating the advection-diffusion mean using CVs

We wish to use the methods discussed in previous sections to improve the estimation of the mean in the advection-diffusion problem, which requires approximations to the principal model. We recall the principal model and quantity of interest considered in this chapter. The high-fidelity model $f^{(1)}$ corresponds to the numerical solution of the two-dimensional steady advection-diffusion equation with uncertain parameters. The inputs are the advection strength a and the diffusion coefficient D, which are treated as independent random variables. The quantity of interest is defined as the L^2-norm of the concentration at $x = 1$, as introduced in Section 3.1.3. The objective in this section is to estimate the mean of this quantity of interest efficiently using variance reduction techniques, with $f^{(1)}$ serving as the reference model against which lower-fidelity approximations are compared.

4.6.1 Coarse-grid approximation

We first examine the use of an SUPG coarse-grid approximation as a control variate. The principal model $f^{(1)}$ has 5445 degrees of freedom (dof). We select three values of $N = 16, 8, 4$, which lead to grids with 1445, 405, and 125 dof, respectively. We will then use these as three coarse-grid approximations to the principal advection-diffusion model, which we will call $f^{(2)}$. We plot the relative cost and correlation between $f^{(1)}$, and each coarse grid $f^{(2)}$ in Figure 4.8. This allows us to examine how much correlation is lost, and how much cheaper $f^{(2)}$ is, as the number of degrees of freedom is reduced. We find the correlation between $f^{(1)}$ and $f^{(2)}$ by sampling 10^7 inputs a and D, and evaluating 10^7 values of $f^{(1)}$ and $f^{(2)}$, then calculating the correlation, using a sufficiently large number of samples that the sampling error in the estimated correlation is negligible. We estimate c by discounting the cost of assembly and assuming the linear system arising from the advection–diffusion problem is solved efficiently using, for example, a multigrid method, resulting in linear scaling with the

number of degrees of freedom. We therefore divide the degrees of freedom of each coarse-grid approximation by 5445 to estimate the cost of $f^{(2)}$ relative to $f^{(1)}$.



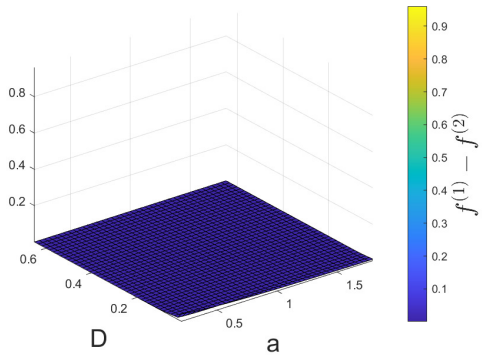
(a) Correlation between different coarse-grid approximations $f^{(2)}$ and $f^{(1)}$ for given dof.

(b) Cost of different coarse-grid approximations $f^{(2)}$ relative to $f^{(1)}$ for given dof.

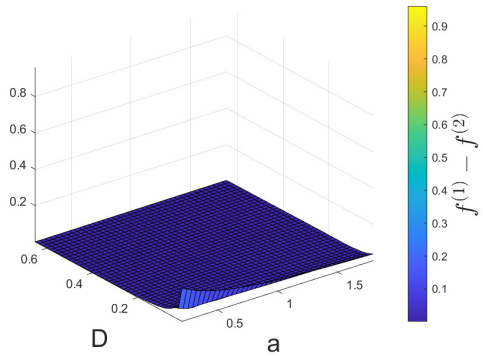
Figure 4.8: Plots of correlation and cost of coarse-grid $f^{(2)}$ for increasing dof

Figure 4.8 shows that using coarser grids reduces the computational cost while only slightly reducing the correlation with $f^{(1)}$. As discussed in Section 4.3, the CVB mean-squared error reflects a trade-off between cost and correlation, so a reduction in cost may compensate for a loss of correlation. This suggests that coarser grids may be preferable to finer grids for this problem. Even with just 125 dof the coarse-grid approximation is very highly correlated. Although coarse meshes introduce error in the solution, this does not necessarily reduce correlation. If the error is consistent across parameter samples, such as through a shift or rescaling—then the low-fidelity model may still preserve the overall trends, resulting in strong linear association [19].

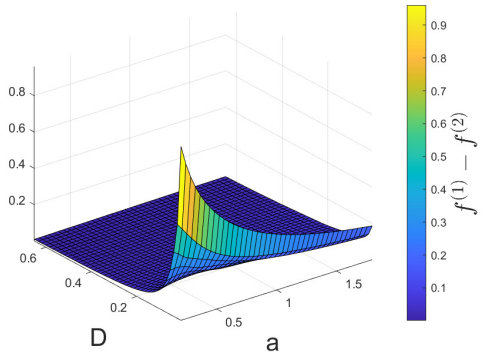
Next we examine the output of each $f^{(2)}$ for a range of a and D input values. We sample 40 evenly distributed values of a and D, then evaluate $f^{(2)}$, for each of the coarse-grid approximations, at every pair of a and D. We then evaluate $f^{(1)}$ at each point on the same grid of inputs, then plot $f^{(1)} - f^{(2)}$ at each pair of a and D. This allows us to see where in the parameter input domain $f^{(2)}$ is most different from $f^{(1)}$, for each coarse-grid approximation. In Figure 4.9 we can see the output is very similar across all values of a and D for the first two coarse-grid approximations, except when both advection and diffusion are small. With 125 dof,



(a) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is a coarse-grid approximation with 1445 dof.



(b) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is a coarse-grid approximation with 405 dof.



(c) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is a coarse-grid approximation with 125 dof.

Figure 4.9: Surface plots of difference in output of approximations for a, D.

the approximation is very accurate for most low a values. This suggests when the grid is too coarse, the chosen output (3.7) approximation is less accurate, particularly when advection is small.

4.6.2 Data-informed approximations

We now consider some alternative low-fidelity approximations. We use data-informed approximations to $f^{(1)}$, examining three in particular: linear regression, Gaussian process regression and a neural network. All three chosen data-informed approximations are trained using the Regression Learner app in MATLAB and were trained using the MSE loss function. The first is linear regression model (LM), which uses the ordinary least squares method to fit a plane to the training dataset [30], which can then be used to find $\hat{\theta}^{(2)}$ for a given a and D.

The second data-informed model is Gaussian process regression (GPR), which is a Bayesian regression technique. When training our GPR model, we choose one of four kernel functions which is then used to create the covariance matrices used in GPR. The choices of kernel covariance functions are rational quadratic, exponential, squared exponential and Matérn 5/2. Details of the kernel functions, as well as the process of training a GPR model can be found in [115]. There are three hyperparameters we must consider, the length scale, signal variance and noise variance. However, one of the advantages of GPR is the relative simplicity of optimising the hyperparameters, so these are all chosen by MATLAB. GPR models are expensive to train as they require inverting the $k^* \times k^*$ kernel matrix (where k^* is the number of samples in the training set) which scales like $\mathcal{O}(k^{*3})$, so can become prohibitively expensive as k^* becomes large. Depending on the length scale (or correlation length) relative to the size of the domain, certain approximations can sometimes be used to reduce computational cost. However, the methods of variance reduction do not consider the cost to train, only the cost to evaluate $f^{(2)}_{X_i}$ given X_i.

The third data-informed model considered is a feedforward, fully connected neural network used for regression. The details of training a neural network and using it

for regression can be found in [31], but the key hyperparameters we adjust are the number of hidden layers and the number of nodes in each layer. We choose from five neural networks with preset hyperparameters provided by the Regression Learner app which implements multilayer perceptrons with ReLU activation functions in the hidden layers and a linear output layer. Three have a single hidden layer with 10, 25, and 100 nodes respectively, while two have 10 nodes per hidden layer, with two and three hidden layers of nodes.

We now examine the minimum training set size k^* required to achieve an acceptable correlation to $f^{(1)}$ for each of the data-informed models. We sample $k^* = 10, 10^2, 10^3, 10^4$ independent random input samples of a and D from input distribution $\mathcal{I}_{\text{low}}$ from (3.10), as distribution $\mathcal{I}_{\text{low}}$ has a larger range of input D values, and solve the PDE to give a training set with k^* values of a, D, and $f^{(1)}_{X_i}$. All simulations were carried out in MATLAB R2024a on a standard desktop workstation equipped with a 13th Gen Intel Core i7-1360P processor and 16 GB RAM. The dominant computational cost arises from solving the high-fidelity PDE to generate the training data, whereas evaluation of the trained surrogate models is computationally negligible by comparison. Since the focus of this thesis is on statistical accuracy rather than absolute runtime performance, detailed wall-clock timings were not recorded.

We also sample $k = 10^7$ independent values of a and D, and evaluate the corresponding $f^{(1)}_{X_i}$ values to use as a test set, again choosing a large number of test samples to ensure the sampling error of our correlation estimate is negligible. Simple linear regression has no hyperparameters so requires only a single realisation for each k^*. For each k^*, we use MATLAB to train four realisations of the GPR model, one for each of the four kernel functions discussed above, then we use each GPR model and the test set inputs to find 10^7 values of $f^{(2)}_{X_i}$. We find the correlation of $f^{(1)}_{X_i}$ and $f^{(2)}_{X_i}$ for each of the four GPR models, and select the model hyperparameters with the highest correlation to be $f^{(2)}$. Likewise, with the neural network, we train five realisations of the model, assess the correlation and select the neural network with hyperparameters resulting in the highest correlation as $f^{(2)}$. Note that the chosen

set of hyperparameters of the neural network and GPR models may not be the same for different values of k^*. For example, the highest correlated GPR model when $k^* = 10^3$ uses an exponential kernel function, whereas when $k^* = 10^4$ the Matérn 5/2 covariance function is used. We plot the correlation of $f_{X_i}^{(1)}$ and $f_{X_i}^{(2)}$, with the hyperparameters which result in highest correlation, for each of the data-informed low-fidelity approximations $f^{(2)}$, against the increasing values of k^*.

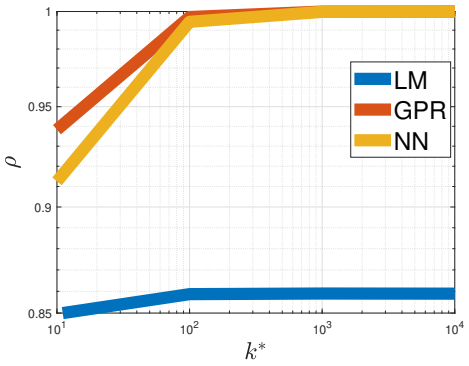


Figure 4.10: Correlation between $f^{(1)}$ and data-informed approximations trained using k^* samples of a, D and $f_{X_i}^{(1)}$.

Figure 4.10 shows that the correlation of the linear model increases slightly with k^*, but quickly converges to a value well below one. This relatively poor performance is likely due to the limited number of degrees of freedom in the linear regression model, which includes only a constant and two coefficients corresponding to the input parameters a and D. A model this simple cannot capture non-linear interactions in the advection-diffusion model, whereas the Gaussian process and neural network models have far greater flexibility. The correlation of the linear model could be improved by using higher-order basis functions, such as Chebyshev polynomials [84]. The correlation of the data-informed approximations $f^{(2)}$ improves as k^* is increased. All three have very high correlation when trained using 10 samples suggesting the advection-diffusion problem may be simple with only two random variables. The neural network and GPR both have near perfect correlation when $k^* = 10000$.

The input samples in the training set are drawn independently at random from the input distributions specified in §3.1.3, therefore we do not know if the parameter

domain of the training set is well covered. We can attempt to improve the sampling using Latin hypercube sampling (LHS), a method of sampling that partitions the input space into strata, then draws random samples within each strata, ensuring the entire domain is (approximately) equally represented. In order to use LHS with the uniform and log-normal distributions of distribution $\mathcal{I}_{\text{high}}$, we use inverse transform sampling. We draw k^* LHS samples between 0 and 1, and find the corresponding values on the CDF of the input distributions. We repeat the experiment from Figure 4.10 with models trained using LHS, and compare the result. In Figure 4.11, LHS

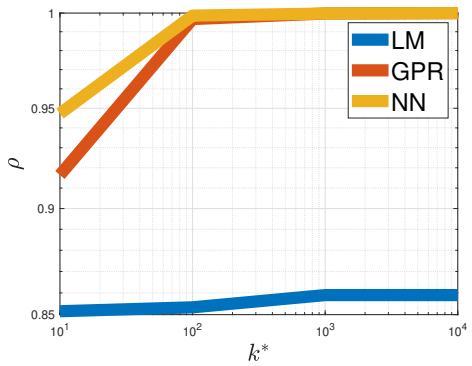
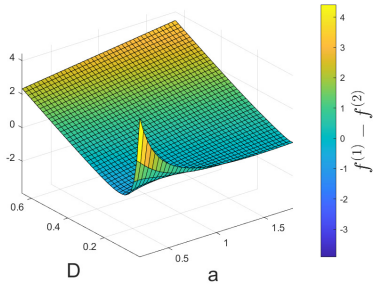


Figure 4.11: Correlation between $f^{(1)}$ and data-informed approximations trained using k^* samples of a, D and $f^{(1)}_{X_i}$, sampled using Latin hypercube sampling.

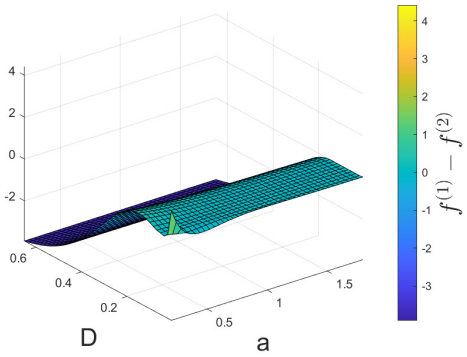
results in all three low-fidelity models $f^{(2)}$ being more highly correlated to $f^{(1)}$, at each $k^* > 10$. The neural network is more highly correlated than the GPR with only 10 samples, which is in contrast to the randomly sampled training sets. In order to train $f^{(2)}$ we will use inverse transform sampling with more than $k^* = 10$ training samples to obtain better low-fidelity models for a lower training cost.

Next we use a grid of values of a and D, as in Figure 4.9, and find $f^{(2)}_{X_i}$ using each of the data-informed models $f^{(2)}$. We use the models trained with 100 inverse transform LHS samples to minimise training cost. We can see in Figure 4.11 that all chosen models have near perfect correlation with $k^* = 100$. We then subtract the values of $f^{(2)}_{X_i}$ from $f^{(1)}_{X_i}$, for all pairs a and D, for each of the data-informed models. This allows us to see where the results of the approximations differ the most from the true $f^{(1)}$ values, and to compare with the results using coarse-grid approximations

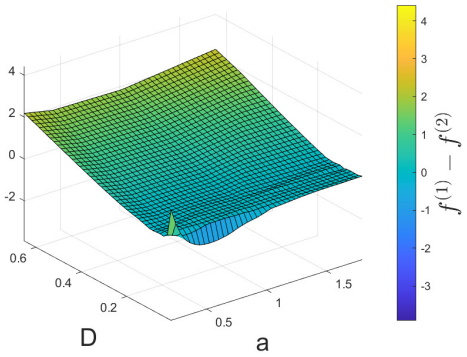
shown in Figure 4.9. In Figure 4.12 the LM approximations gives the least accurate



(a) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is a linear regression model approximation.



(b) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is Gaussian process regression approximation.



(c) Value of $f^{(1)} - f^{(2)}$ when $f^{(2)}$ is a neural network approximation.

Figure 4.12: Surface plots of difference in output of data-informed approximations for a, D.

results at the high and low a and D values, as a linear model uses a plane so would struggle to approximate $f^{(1)}$ where $f^{(1)}$ is not linear. The GPR approximation gives consistently lower results than $f^{(1)}$ except when D is small. The neural network gets less accurate as diffusion increases. Despite the inaccuracies, all three are highly correlated to $f^{(1)}$. This behaviour is consistent with the sampling strategy described in Section 3.1.3, in which the diffusion coefficient is sampled from a distribution concentrated near low values. As a result, the majority of input samples correspond to low diffusion, where all three approximations are most accurate.

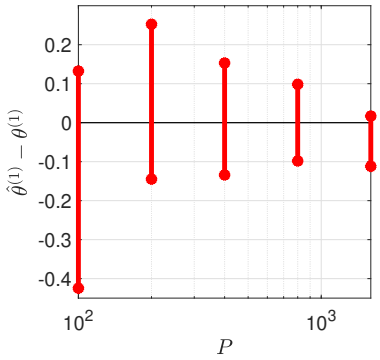
To assess the cost of the data-informed approximations, we use the number of observations per second as recorded by MATLAB, that is, the number of evaluations of $f^{(2)}_{X_i}$ that MATLAB can perform in one second for a given pair of input parameters a

and D. For a set of a and D values, the neural network and linear model both evaluate 15000 observations of $f_{X_i}^{(2)}$ per second, while the Gaussian process regression evaluates approximately 1000 observations per second. The principal $f^{(1)}$ can be evaluated 70 times per second, so the relative cost can be found. This means the neural network and linear model both have cost $c = 0.005$, and the Gaussian processes regression has cost $c = 0.07$.

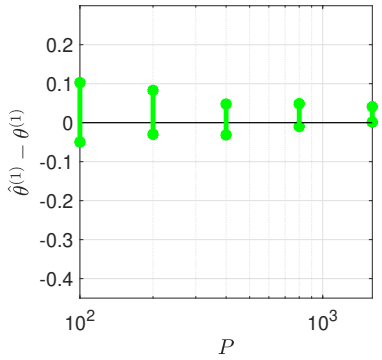
We can also use the CVB method to find 95% confidence intervals for the advection-diffusion model using five approximations: coarse-grid approximations with 405 and 125 degrees of freedom respectively, linear regression model approximation, Gaussian processes regression approximation, and a neural network approximation. We train the data-informed approximations using 100 LHS inputs from distribution $\mathcal{I}_{\text{high}}$. The GPR with the highest correlation to $f^{(1)}$ uses the exponential kernel, while the NN regression with the highest correlation has two hidden layers of ten nodes. We then sample inputs from distribution $\mathcal{I}_{\text{high}}$ and find the confidence intervals first by using standard Monte Carlo for comparison, then using the method in Algorithm 5, giving six realisations of confidence intervals which we can then compare. We do this for five values of P and centre the results around zero in each case.

In Figure 4.13 we can see all of the confidence intervals are smaller when calculated using CVB than when calculated using only Monte Carlo. The neural network control variate performs the best of all, with the coarse-grid control variates performing better than the GPR and LM control variates. The $N = 4$ coarse-grid performs better than the $N = 8$ grid, suggesting the large savings on cost mean the slight drop in correlation is worth it. This is because the substantially lower cost of the coarser grid reduces allows a larger number of high-fidelity evaluations of $f^{(1)}$ to be performed within the same computational budget. This increase in the number of high-fidelity samples can compensate for a modest reduction in correlation, leading to a smaller overall mean-squared error.

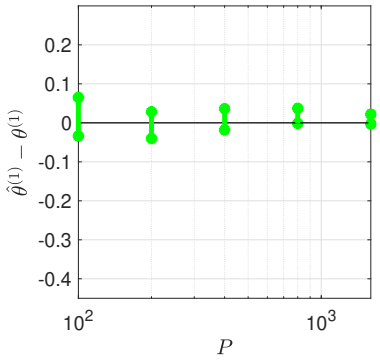
We can repeat the experiment from Figure 4.13 with input distribution (3.10),



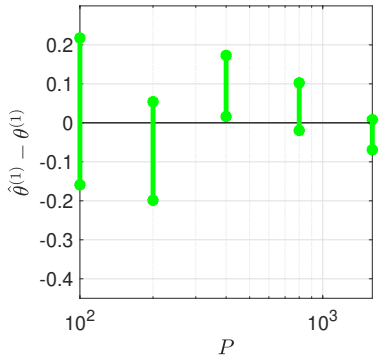
(a) 95% confidence intervals of $f^{(1)}$, calculated using Monte Carlo.



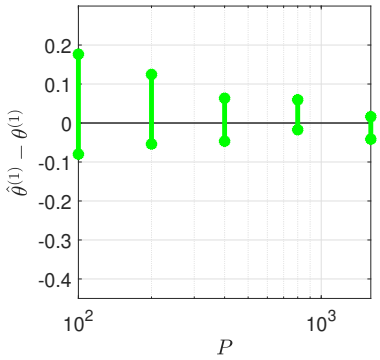
(b) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a coarse-grid approximation with 405 dof.



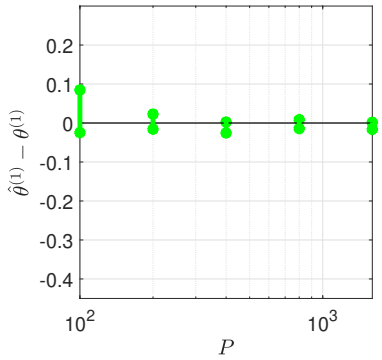
(c) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a linear regression approximation.



(e) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a Gaussian processes regression approximation.



(f) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a neural network approximation.

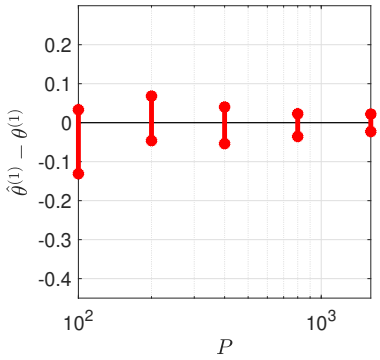
Figure 4.13: 95% confidence intervals calculated with increasing budget P and inputs from distribution $\mathcal{I}_{\text{high}}$.

to examine how sampling inputs with a diffusion coefficient with larger mean and smaller variance affects the confidence intervals and the effect of the CVB method. We retrain the data-informed models with 100 LHS samples from distribution $\mathcal{I}_{\text{low}}$, and 100 evaluations of $f^{(1)}$. The GPR with the highest correlation to $f^{(1)}$ now uses the Matérn 5/2 kernel, while the NN regression with the highest correlation once again has two hidden layers of ten nodes. In Figure 4.14 we can see the Monte Carlo confidence intervals are smaller using distribution $\mathcal{I}_{\text{low}}$, and the CVB method reduces the size of the confidence intervals in all cases. As with the previous results, the neural network approximation and the coarse-grid with 125 dof improve the results the most.

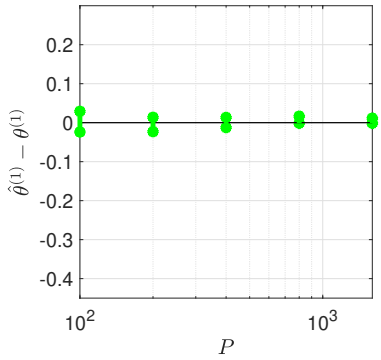
4.7 Conclusions

In this chapter we examined methods of variance reduction for estimating the mean of an advection-diffusion model problem. In §4.1 we examined the standard control variates method and derived its theoretical MSE as a function of variance. We then confirmed that the theoretical MSE matches the estimated MSE using the Ishigami function and approximations with known means, as shown in Figure 4.1. In §4.2 we presented a method for estimating the mean when the control variate mean is unknown. We introduced a budget P which can be used optimally to estimate the control variate mean and to estimate the mean of the QoI with minimal variance. We derived the theoretical MSE of the CVEM method, and in Figure 4.2 we tested the CVEM method by estimating the mean of the Ishigami function and showed that the computed MSE matches the theoretical MSE. In Figure 4.3 we compared the MSE of the mean estimate of the Ishigami function calculated using control variates and the CVEM method, showing that there is minimal difference in the budget required and accuracy unless the control variate used is very highly correlated.

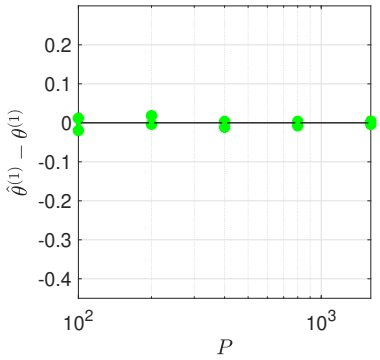
In §4.3 we introduced a method of adding a budget control to the CVEM method, allowing for a capped budget to be used. We presented two methods of adding a budget control to the CVEM method, and we derive the theoretical MSE for the



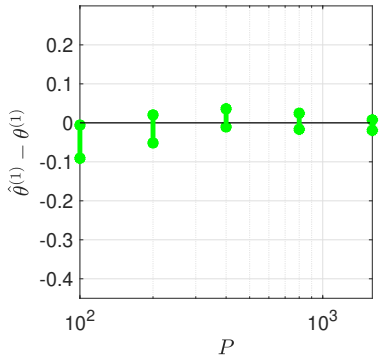
(a) 95% confidence intervals of $f^{(1)}$, calculated using Monte Carlo.



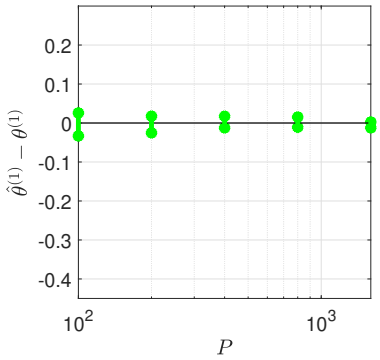
(b) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a coarse-grid approximation with 405 dof.



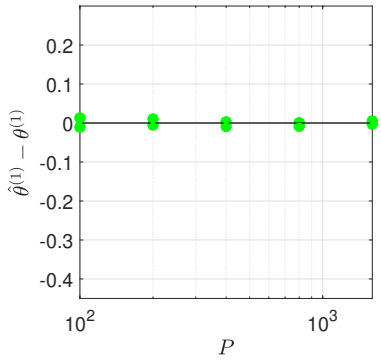
(c) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a linear regression approximation.



(e) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a Gaussian processes regression approximation.



(f) 95% confidence intervals of $f^{(1)}$, calculated using CVB with a neural network approximation.

Figure 4.14: 95% confidence intervals calculated with increasing budget P and inputs from distribution $\mathcal{I}_{\text{low}}$.

new method, called the CVB method. In Figure 4.4 we compared the MSE of the CVB method to the MC and CVKM methods for a theoretical approximation with a correlation and relative cost, showing that the CVB method has a higher MSE than the CVKM method, but that the difference is small with most correlations and costs. We confirmed the theoretical MSE of the CVB method by estimating the mean of the Ishigami equation in Figure 4.5, and we compared the MSE of the mean estimated using CVB to CVEM and MC, showing there is minimal MSE difference between knowing the mean and estimating it, for a given budget, unless the control variate is extremely highly correlated.

In §4.4 we presented the two-level multi-fidelity Monte Carlo method and derived the theoretical MSE of this approach. We compared the MFMC MSE to the CVB MSE and found they are equivalent for a given P. In §4.5, we found a multi-fidelity method for finding confidence intervals using bootstrapping with the CVB method, and we tested the coverage using the Ishigami function and its approximations. In Table 4.1 we confirmed the coverage of CIs calculated using the CVB method is as accurate, and in Figure 4.7 we see the CVB method gives smaller confidence intervals than the MC method when estimating the mean of the Ishigami equation.

In §4.6 we examined possible low-fidelity approximations to the advection-diffusion problem. We looked at coarse-grid and data-informed approximations, and we examined which values of a and D cause the outputs of the approximations to differ most from the principal model output. We then used these approximations to find confidence intervals using the CVB method for both the $\mathcal{I}_{\text{high}}$ and $\mathcal{I}_{\text{low}}$ distributions, finding that all of the approximations resulted in smaller confidence intervals using the CVB method. We also found that the combination of low cost and high correlation results in the coarse-grid with 125 dof, and the neural network approximation giving the smallest CIs for both input distributions.

Taken together, the results in this chapter suggest a clear set of practical choices for the advection-diffusion problem considered. SUPG stabilisation is preferred to standard Galerkin discretisation in order to obtain stable and physically meaningful

quantities of interest. Control variate methods provide substantial variance reduction relative to standard Monte Carlo, and when the control variate mean is unknown, the CVEM and CVB formulations achieve near-optimal performance when the computational budget is allocated appropriately. In particular, inexpensive approximations with moderate correlation are shown to be more effective control variates than more costly, highly resolved models. Finally, while Latin hypercube sampling offers modest improvements over independent sampling, it does not alter the overall conclusions. For this problem, the most effective strategy combines SUPG discretisation with a two-level control variate or multifidelity estimator using a low-cost, moderately correlated approximation.

Chapter 5

Global sensitivity analysis

The aim of this chapter is to apply variance reduction techniques to the estimation of Sobol sensitivity indices, with a focus on reducing sampling error in both point estimates and confidence intervals. We begin by reviewing standard Monte Carlo estimators for first- and total-order Sobol indices and comparing their statistical efficiency. We then introduce a CVB-based approach for estimating Sobol indices and derive the corresponding MSE reduction. Numerical experiments are presented to assess estimator performance and to demonstrate the benefits of the proposed methods when compared with standard Monte Carlo approaches.

5.1 Sobol indices

In this thesis we will use Sobol sensitivity indices which are commonly used in variance-based sensitivity analysis. We use Sobol indices to quantify the contribution of each uncertain parameter to the variance of the output of our advection–diffusion model. Consider a model with a scalar output $Y = f^{(1)}(\mathbf{X})$, where $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ is a vector of input parameters (factors). Here each X_i is a random variable, uncorrelated with any other X_i, with a known probability distribution (although the probability distribution function does not need to be the same for each X_i). We wish to know the importance of each X_i in influencing the output

value Y, that is, we wish to know how each X_i contributes to the overall variance $\mathbb{V}(Y)$.

We begin by considering the law of total expectation [6] which states that

$$\mathbb{E}(Y) = \mathbb{E}(\mathbb{E}(Y|X_i)), \tag{5.1}$$

where $\mathbb{E}(Y|X_i)$ is the conditional expectation of Y when X_i is fixed at a specific value. Given

$$\mathbb{V}(Y) = \mathbb{E}(Y^2) - \mathbb{E}(Y)^2, \tag{5.2}$$

we use (5.2) and (5.1) to find

$$\begin{aligned}
\mathbb{E}(Y^2) &= \mathbb{E}(Y^2) - \mathbb{E}(\mathbb{E}(Y|X_i)^2) + \mathbb{E}(\mathbb{E}(Y|X_i)^2) \\
&= \mathbb{E}(\mathbb{E}(Y^2|X_i)) - \mathbb{E}(\mathbb{E}(Y|X_i)^2) + \mathbb{E}(\mathbb{E}(Y|X_i)^2) \\
&= \mathbb{E}(\mathbb{V}(Y|X_i) + \mathbb{E}(Y|X_i)^2),
\end{aligned}$$

and subtract $\mathbb{E}(Y)^2$ from both sides to get

$$\begin{aligned}
\mathbb{V}(Y) &= \mathbb{E}(\mathbb{V}(Y|X_i) + \mathbb{E}(Y|X_i)^2) - \mathbb{E}(Y)^2 \\
&= \mathbb{E}(\mathbb{V}(Y|X_i)) + \mathbb{E}(\mathbb{E}(Y|X_i)^2) - \mathbb{E}(\mathbb{E}(Y|X_i))^2 \\
&= \mathbb{E}(\mathbb{V}(Y|X_i)) + \mathbb{V}(\mathbb{E}(Y|X_i)).
\end{aligned} \tag{5.3}$$

This is known as the law of total variance. If we were to fix one input at a single value, $X_i = x_i^*$, we could estimate the variance of Y caused by all inputs except X_i. Let $\mathbf{X}_{\sim i}$ be $\mathbf{X}$ where the i^{th} input is fixed at $X_i = x_i^*$, then the conditional variance $\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i = x_i^*)$ is the variance of Y over $\mathbf{X}_{\sim i}$. This could be a measure of the importance of X_i to the variance of Y, however the chosen value of x_i^* will influence the value $\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i = x_i^*)$, and there exist functions for which the sum of $\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i = x_i^*)$ for all i will be greater than $\mathbb{V}(Y)$ [122]. Instead, we can take the mean of $\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i = x_i^*)$ over all values of x_i^*, which we write as $\mathbb{E}_{X_i}(\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i))$.

So from (5.3) we have

$$\mathbb{E}_{X_i}(\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i)) + \mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i)) = \mathbb{V}(Y). \tag{5.4}$$

From (5.4) we can see that a small value of $\mathbb{E}_{X_i}(\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i))$ would result in the mean conditional variance without X_i being small, and that $\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i))$ would be close to $\mathbb{V}(Y)$, which would suggest that X_i is an important input. We can use $\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i))$ as a measure of sensitivity where a higher value means X_i is more important, and it is clear $0 \leq \mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i)) \leq \mathbb{V}(Y)$. We can therefore divide $\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i))$ by $\mathbb{V}(Y)$ to get a measure of the importance of X_i which takes a value between 0 and 1:

$$S_i = \frac{\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i))}{\mathbb{V}(Y)}. \tag{5.5}$$

This is a first-order sensitivity index, known as the first-order Sobol index. As the inputs are uncorrelated, the sum of all first-order indices will be less than one [122].

The first-order Sobol index can give a measure of the contribution of each input in isolation, but cannot account for interaction effects, so cannot show the importance of each input when it interacts with other inputs. To account for interactions, we consider $\mathbf{X}_i$, in which all inputs are fixed at single random values except X_i. The model $Y = f^{(1)}(\mathbf{X}_i)$ will have a true expected value which can be found by considering all possible X_i, given by $\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i})$. Note this is different from $\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i)$ as it considers all possible values of X_i, as opposed to considering all values of all inputs except X_i. As $\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i})$ is a function of random variables, it is itself a random variable, so the true expected value will have a variance over all $\mathbf{X}_{\sim i}$, $\mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i}))$, and from (5.3)

$$\mathbb{V}(Y) = \mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i})) + \mathbb{E}_{\mathbf{X}_{\sim i}}(\mathbb{V}_{X_i}(Y|\mathbf{X}_{\sim i})). \tag{5.6}$$

The measure $\mathbb{E}_{\mathbf{X}_{\sim i}}(\mathbb{V}_{X_i}(Y|\mathbf{X}_{\sim i})) = \mathbb{V}(Y) - \mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i}))$ is the mean variance of Y remaining, if we knew the variance of the true values of Y when all inputs except X_i are fixed. This could be considered a measure of all variance that can be

explained by varying X_i. As before, we can divide by $\mathbb{V}(Y)$ to get a measure between zero and one:

$$S_{T_i} = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}}(\mathbb{V}_{X_i}(Y|\mathbf{X}_{\sim i}))}{\mathbb{V}(Y)} = 1 - \frac{\mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i}))}{\mathbb{V}(Y)}. \tag{5.7}$$

The measure S_{T_i} is known as the total-order Sobol index and it includes all variance contributed by X_i, including all variance caused by its interactions with other inputs. As variance between interacting inputs can be counted multiple times, the sum of all S_{T_i} will not necessarily be equal to one [122].

We cannot always know the true values of $\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y|X_i))$ or $\mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}(Y|\mathbf{X}_{\sim i}))$, but we can estimate the Sobol indices (5.5) and (5.7). If we have two sample sets of inputs $A = (x_1, \ldots, x_m)$ and $C = (x_1, \ldots, x_m)$, where x_i is the same value in both inputs, while all other x are independently sampled. We can evaluate two realisations of $f^{(1)}$, say $Y^{(A)} = f^{(1)}(A)$ and $Y^{(C)} = f^{(1)}(C)$. The covariance of $Y^{(A)}$ and $Y^{(C)}$ is then

$$\begin{aligned}\mathrm{Cov}(Y^{(A)}, Y^{(C)}) &= \mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}Y^{(C)}) - \mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)})\,\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(C)}) \\ &= \mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}Y^{(C)}) - \mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)})^2, \end{aligned} \tag{5.8}$$

and, as $Y^{(A)}$ and $Y^{(C)}$ are independent,

$$\begin{aligned}\mathbb{E}_{\mathbf{X}_{\sim i}}((Y^{(A)}Y^{(C)})) &= \mathbb{E}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}((Y^{(A)}Y^{(C)})|X_i)) \\ &= \mathbb{E}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}|X_i)\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(C)}|X_i)) = \mathbb{E}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}|X_i)^2). \end{aligned} \tag{5.9}$$

It therefore follows from (5.8), (5.9) and (5.1) that

$$\mathrm{Cov}(Y^{(A)}, Y^{(C)}) = \mathbb{E}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}|X_i)^2) - \mathbb{E}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}|X_i))^2 = \mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}(Y^{(A)}|X_i)).$$

The first-order Sobol index for X_i can therefore be estimated as

$$S_i = \frac{\text{Cov}(Y^{(A)}, Y^{(C)})}{\mathbb{V}(Y^{(A)})}, \tag{5.10}$$

with $Y^{(A)} = f^{(1)}(A)$ and $Y^{(C)} = f^{(1)}(C)$. If we sample another set of inputs, say $B = (x_1, \ldots, x_m)$, where all x have the same value as C, except x_i, then we can find $Y^{(B)} = f^{(1)}(B)$. Then from (5.7) the total-order index can be estimated as

$$S_{T_i} = 1 - \frac{\text{Cov}(Y^{(B)}, Y^{(C)})}{\mathbb{V}(Y^{(B)})}. \tag{5.11}$$

The error of the Sobol indices estimates, and the number of evaluations of $f^{(1)}$ required, will depend on the estimators used to find the covariance and variance, so we will now look at estimators of (5.10).

5.2 Monte Carlo estimators for Sobol indices

There are many possible methods we could use to estimate the Sobol indices. One of the most commonly used approaches for estimating first-order and total-order Sobol indices is a Monte Carlo approach known as the Saltelli method [9].

To estimate first-order indices using the Saltelli method, we construct two independently sampled $k \times m$ design matrices of inputs, where k is the sample size and m is the number of components in the input,

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}. \tag{5.12}$$

We define $\mathbf{A}_{i,:}$ and $\mathbf{B}_{i,:}$ as the i^{th} row of $\mathbf{A}$ and $\mathbf{B}$ respectively, then evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{A})} = \begin{bmatrix} \tilde{Y}_{1,:}^{(\mathbf{A})} \\ \vdots \\ \tilde{Y}_{k,:}^{(\mathbf{A})} \end{bmatrix} = \begin{bmatrix} f^{(1)}(\mathbf{A}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{A}_{k,:}) \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{Y}}^{(\mathbf{B})} = \begin{bmatrix} \tilde{Y}_{1}^{(\mathbf{B})} \\ \vdots \\ \tilde{Y}_{k}^{(\mathbf{B})} \end{bmatrix} = \begin{bmatrix} f^{(1)}(\mathbf{B}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{B}_{k,:}) \end{bmatrix}. \tag{5.13}$$

To estimate the first-order Sobol index for X_i, we create the matrices $\mathbf{C}^{(i)}$ where the i^{th} column is from $\mathbf{A}$, and all other columns are from $\mathbf{B}$, and evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})} = \begin{bmatrix} \tilde{Y}_{1}^{(\mathbf{C}^{(i)})} \\ \vdots \\ \tilde{Y}_{k}^{(\mathbf{C}^{(i)})} \end{bmatrix} = \begin{bmatrix} f^{(1)}((\mathbf{C}^{(i)})_{1,:}) \\ \vdots \\ f^{(1)}((\mathbf{C}^{(i)})_{k,:}) \end{bmatrix}. \tag{5.14}$$

We can then estimate the first-order and total-order Sobol indices, (5.10) and (5.11), for X_i as

$$\tilde{S}_i^{\mathrm{S}} = \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - (\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})})(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{C}^{(i)})})}{\frac{1}{k}\sum_{j=1}^{k} (\tilde{Y}_j^{(\mathbf{A})})^2 - (\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})})^2} \tag{5.15}$$

and

$$\tilde{S}_{T_i}^{\mathrm{S}} = 1 - \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - (\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})})(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{C}^{(i)})})}{\frac{1}{k}\sum_{j=1}^{k} (\tilde{Y}_j^{(\mathbf{B})})^2 - (\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})})^2}. \tag{5.16}$$

An algorithm for implementing Saltelli's method for estimating the first-order and total-order indices is shown in Algorithm 8 in Appendix A.

We now test the Saltelli method using the Ishigami function (3.5) with uncertain inputs $x_i \in u[-\pi, \pi]$, $i = 1, 2, 3$. The Ishigami function has known values of Sobol indices [114] given in Table 5.1. We set the cost for estimating Sobol indices ($P = 1$) to be the computational cost of finding one value of $\tilde{S}_i^{\mathrm{S}}$ and $\tilde{S}_{T_i}^{\mathrm{S}}$ i.e. $m+2$ evaluations of $f^{(1)}$ (with $k = 1$). When estimating Sobol indices using only high-fidelity $f^{(1)}$, the total budget $P = k$. We select total budgets of $P = 100, 200, 400, 800, 1600$, and calculate 100 replicates of $\tilde{S}_i^{\mathrm{S}}$ and $\tilde{S}_{T_i}^{\mathrm{S}}$ for the Ishigami function, using the method

i	1	2	3
S_i	0.401	0.288	0
S_{T_i}	0.712	0.288	0.311

Table 5.1: Sobol first-order and total-order values for the Ishigami function with input $\mathbf{X} = (X_1, X_2, X_3)$.

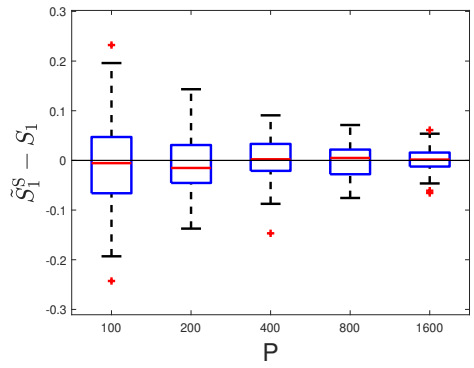
in Algorithm 8, for each value of P. We then subtract the true value of the Sobol indices from the estimate, to centre around zero, and plot the results in a boxplot to examine whether the estimates $\tilde{S}_i^{\mathrm{S}}$ and $\tilde{S}_{T_i}^{\mathrm{S}}$ cover the true values as given in Table 5.1, and whether they converge to the true S_i and S_{T_i}, as P is increased. In Figure 5.1 we can see the boxplots of $\tilde{S}_i^{\mathrm{S}}$ and $\tilde{S}_{T_i}^{\mathrm{S}}$ cover the true values of S_i and S_{T_i}, and converge to the values given in Table 5.1 as P is increased. This suggests that this method of estimating Sobol indices is effective.

The Saltelli estimate of the first-order and total-order indices are dependent upon $\mathbb{E}(\tilde{Y}^{(\mathbf{A})})$ and $\mathbb{E}(\tilde{Y}^{(\mathbf{B})})$, which are estimates of the mean of $f^{(1)}$. By observing that $\mathbb{E}(\tilde{Y}^{(\mathbf{C}^{(i)})})$ is also an estimate of the mean of $f^{(1)}$, the variance of the estimate of the Sobol indices can be reduced using an alternate method of estimating the first-order indices known as the Sobol-Janon (SJ) method [70]. The SJ method samples inputs to find $\tilde{\mathbf{Y}}^{(\mathbf{A})}$, $\tilde{\mathbf{Y}}^{(\mathbf{B})}$ and $\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})}$ as in (5.13) and (5.14), then the first-order indices for X_i are estimated as

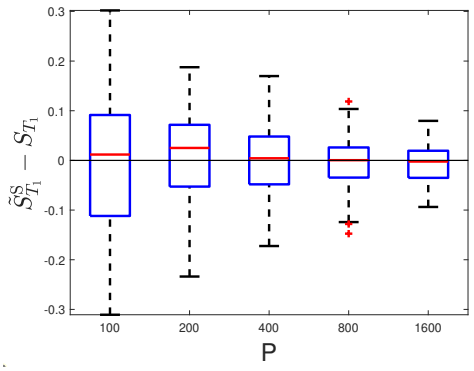
$$\tilde{S}_i^{\mathrm{SJ}} = \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k}\left[\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)^2}{\frac{1}{k}\sum_{j=1}^{k}\left[\frac{(\tilde{Y}_j^{(\mathbf{A})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2}\right] - \left(\frac{1}{k}\sum_{j=1}^{k}\left[\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)^2}. \tag{5.17}$$

We can also reduce the variance of the total-order index estimator using the Janon-Monod (JM) method [112], which estimates the total-order indices using

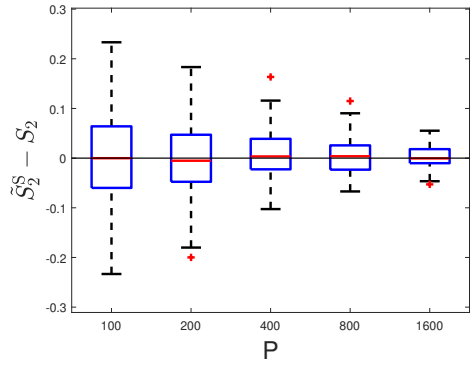
$$\tilde{S}_{T_i}^{\mathrm{JM}} = 1 - \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k}\left[\frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)^2}{\frac{1}{k}\sum_{j=1}^{k}\left[\frac{(\tilde{Y}_j^{(\mathbf{B})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2}\right] - \left(\frac{1}{k}\sum_{j=1}^{k}\left[\frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)^2}. \tag{5.18}$$



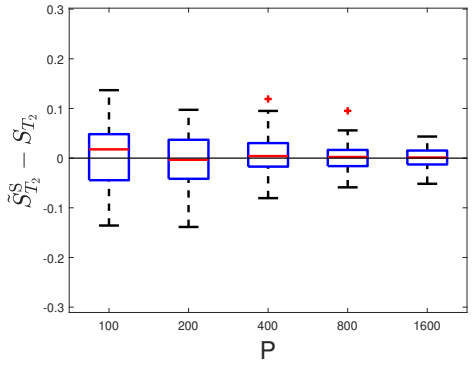
(a) Boxplots of 100 replicates of $\tilde{S}_1^S$.



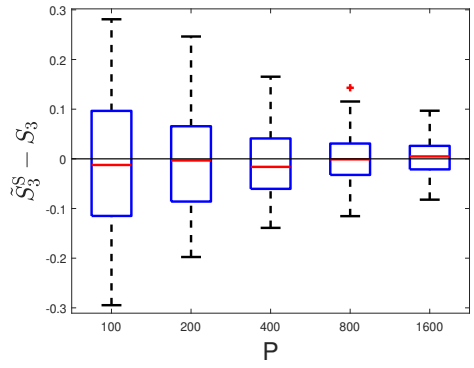
(b) Boxplots of 100 replicates of $\tilde{S}_{T_1}^S$.



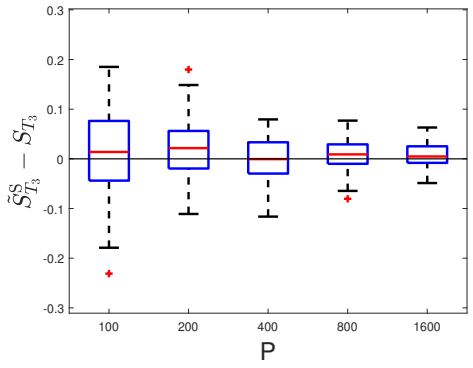
(c) Boxplots of 100 replicates of $\tilde{S}_2^S$.



(d) Boxplots of 100 replicates of $\tilde{S}_{T_2}^S$.



(e) Boxplots of 100 replicates of $\tilde{S}_3^S$.



(f) Boxplots of 100 replicates of $\tilde{S}_{T_3}^S$.

Figure 5.1: Boxplots of replications of estimates of Sobol first-order and total-order indices for the Ishigami function using the Saltelli method.

An algorithm for estimating the first-order and total-order indices using the Sobol-Janon and Janon-Monod methods can be found in Algorithm 9 in Appendix A.

Previously, when comparing estimators of the mean, we used the MSE as an error. If the first-order Sobol index estimator is unbiased, then the MSE of the estimator is $\frac{\mathbb{V}(\tilde{S}_i)}{P}$. From [70] we know the empirical variances of the first-order estimators to be

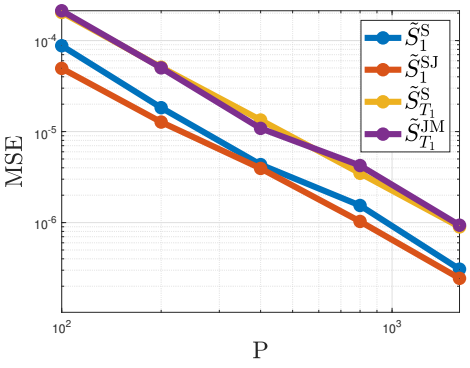
$$\mathbb{V}(\tilde{S}_i^{\mathrm{S}}) = \frac{\frac{1}{k}\sum_{j=1}^{k}\left(\tilde{Y}_j^{(\mathbf{A})} - \frac{1}{k}\sum_{j=1}^{k}\tilde{Y}_j^{(\mathbf{A})}\right)\left(\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \frac{1}{k}\sum_{j=1}^{k}\tilde{Y}_j^{(\mathbf{C}^{(i)})}\right)}{\frac{1}{k}\sum_{j=1}^{k}\left(\tilde{Y}_j^{(\mathbf{A})} - \frac{1}{k}\sum_{j=1}^{k}\tilde{Y}_j^{(\mathbf{A})}\right)^2}, \tag{5.19}$$

and

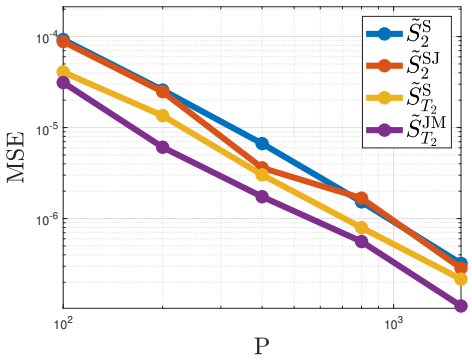
$$\mathbb{V}(\tilde{S}_i^{\mathrm{SJ}}) = \frac{\sum_{j=1}^{k}\left(\tilde{Y}_j^{(\mathbf{A})} - \left[\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)\left(\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left[\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]\right)}{\sum_{j=1}^{k}\left(\left[\frac{(\tilde{Y}_j^{(\mathbf{A})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2}\right] - \left[\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right]^2\right)}. \tag{5.20}$$

We do not have empirical values of the variance of the total-order estimates, but we can use the Ishigami function to calculate the MSE of each of the estimators of the first-order and total-order indices. We now compare the performance of the two estimators when used to find the Sobol indices of the Ishigami function. We calculate 100 replicates of each of $\tilde{S}_i^{\mathrm{S}}$, $\tilde{S}_{T_i}^{\mathrm{S}}$, $\tilde{S}_i^{\mathrm{SJ}}$, and $\tilde{S}_{T_i}^{\mathrm{JM}}$, using the methods in Algorithms 8 and 9, for five values of $P = 100, 200, 400, 800, 1600$. In Figure 5.2 we plot the MSE using the replicates against P, comparing the methods of estimating each Sobol index.

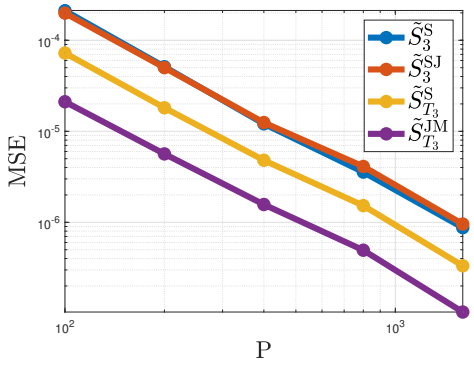
In Figure 5.2 we can see the Sobol-Janon and Janon-Modol methods have equivalent or lower MSE than the Saltelli method for all Sobol indices, suggesting the Janon methods are better for the problem. Next we wish to examine the Monte Carlo estimates of S_i and S_{T_i} using the Sobol-Janon and Janon-Modol methods of estimation. We calculate 100 replications of $\tilde{S}_i^{\mathrm{SJ}}$ and $\tilde{S}_{T_i}^{\mathrm{JM}}$, for the Ishigami function. As in Figure 5.1 we do this for $P = 100, 200, 400, 800, 1600$ and plot boxplots of the replicates in Figure 5.3. The Monte Carlo estimates of $\tilde{S}_i^{\mathrm{S}}$, $\tilde{S}_i^{\mathrm{SJ}}$, $\tilde{S}_{T_i}^{\mathrm{S}}$, and $\tilde{S}_{T_i}^{\mathrm{JM}}$ cover



(a) MSE of $\tilde{S}_1^{\mathrm{S}}$, $\tilde{S}_1^{\mathrm{SJ}}$, $\tilde{S}_{T_1}^{\mathrm{S}}$, and $\tilde{S}_{T_1}^{\mathrm{JM}}$

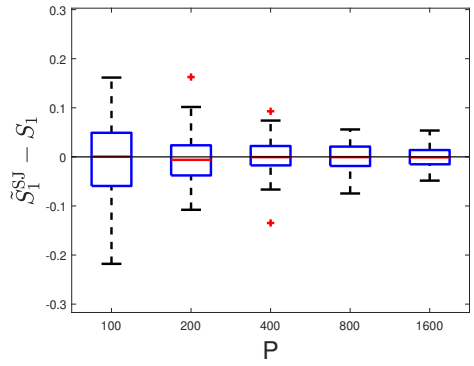


(b) MSE of $\tilde{S}_2^{\mathrm{S}}$, $\tilde{S}_2^{\mathrm{SJ}}$, $\tilde{S}_{T_2}^{\mathrm{S}}$, and $\tilde{S}_{T_2}^{\mathrm{JM}}$

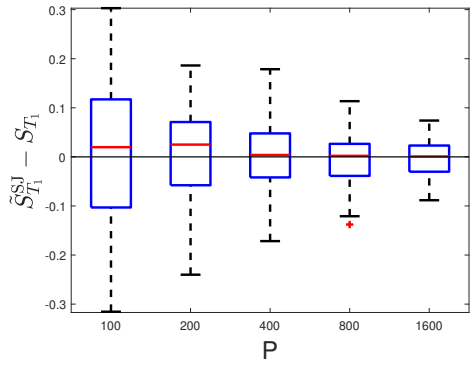


(c) MSE of $\tilde{S}_3^{\mathrm{S}}$, $\tilde{S}_3^{\mathrm{SJ}}$, $\tilde{S}_{T_3}^{\mathrm{S}}$, and $\tilde{S}_{T_3}^{\mathrm{JM}}$

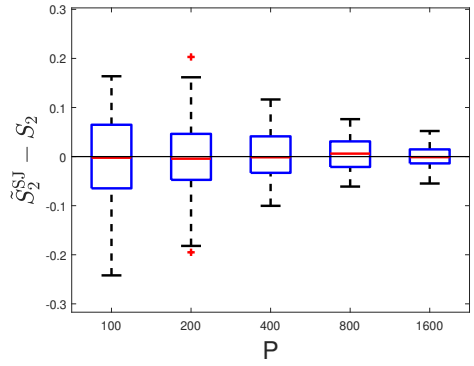
Figure 5.2: MSE of replicates of estimates of S_i and S_{T_i} of the Ishigami function, using the Saltelli, Sobol-Janon and Janon-Modol methods.



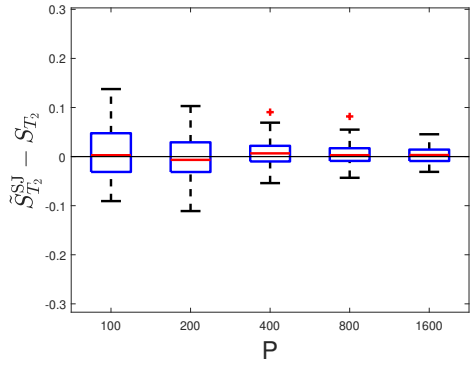
(a) Boxplots of 100 replicates of $\tilde{S}_1^{\mathrm{SJ}}$.



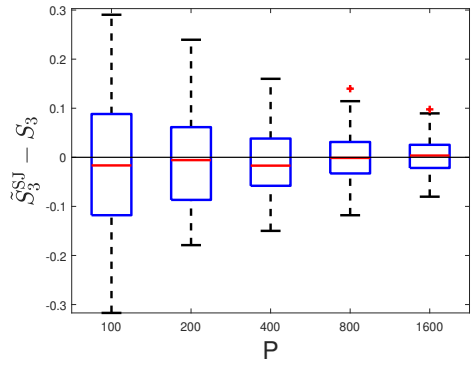
(b) Boxplots of 100 replicates of $\tilde{S}_{T_1}^{\mathrm{JM}}$.



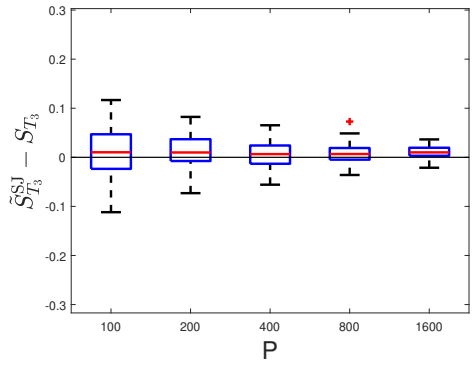
(c) Boxplots of 100 replicates of $\tilde{S}_2^{\mathrm{SJ}}$.



(d) Boxplots of 100 replicates of $\tilde{S}_{T_2}^{\mathrm{JM}}$.



(e) Boxplots of 100 replicates of $\tilde{S}_3^{\mathrm{SJ}}$.



(f) Boxplots of 100 replicates of $\tilde{S}_{T_3}^{\mathrm{JM}}$.

Figure 5.3: Boxplots of replications of estimates of Sobol first-order and total-order indices of the Ishigami function using the Sobol-Janon and Janon-Modol methods, with increasing P.

the true values of S_i and S_{T_i}, and converge as expected when the budget is increased. As they have lower variance, going forward we will only use the Sobol-Janon and Janon-Modol methods of estimation when calculating the approximations $\tilde{S}_i$ and $\tilde{S}_{T_i}$.

5.3 Multi-fidelity Sobol index estimation

Estimating Sobol indices using a Monte Carlo method can become computationally expensive, requiring $k(m+2)$ evaluations of $f^{(1)}$ to estimate the first-order and total-order indices for each input. If we have an approximation to $f^{(1)}$, with known relative cost c and correlation ρ, then the control variates with budget control method introduced in §4.3 can be used to reduce the variance of the Sobol estimates, for a given P. As when estimating the mean in §4.3, we find β, ϕ, k, k', and α as

$$\beta = \left[\left(\frac{\rho^2}{1-\rho^2}\right)\left(\frac{1}{c}\right)\right]^{\frac{1}{2}}, \quad \phi = \max\{0, \beta - 1\}, \quad k = \frac{P}{1 + c(1+\phi)}, \quad \text{and} \quad k' = k \times \phi. \tag{5.21}$$

In order to get approximations to $\tilde{S}_i$ and $\tilde{S}_{T_i}$, which we will call $\tilde{S}_i^{(2)}$ and $\tilde{S}_{T_i}^{(2)}$ respectively, we estimate the Sobol indices as normal using the method in Algorithm 9, only with $f^{(2)}$ instead of $f^{(1)}$. Looking first at the first-order indices, we consider the only cost of estimating the Sobol indices to be evaluating $f^{(1)}$ and $f^{(2)}$. We require k input samples, as in (5.12), to construct $\mathbf{A}$ and $\mathbf{B}$, in order to evaluate $\tilde{\mathbf{Y}}^{(\mathbf{A})}$ from (5.13). We use the same $\mathbf{A}$ to calculate k realisations of $f^{(2)}$, stored in $\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}$. We can then use $\tilde{\mathbf{Y}}^{(\mathbf{A})}$ and $\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}$ to find α via

$$\alpha = \left(\frac{\mathbb{C}\text{ov}(\tilde{\mathbf{Y}}^{(\mathbf{A})}, \tilde{\mathbf{Y}}^{(\mathbf{A}_2)})}{\mathbb{V}\text{ar}\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}}\right)\left(\frac{\phi}{\phi+1}\right). \tag{5.22}$$

We assemble $\mathbf{C}^{(i)}$ as in (5.14) and find $\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})}$ and $\tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$. We can use $\tilde{\mathbf{Y}}^{(\mathbf{A})}$ and $\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})}$ to find $\tilde{S}_i$ using (5.17), and use the same equation with $\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}$ and $\tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$ to find $\tilde{S}_i^{(2)}$.

Next we sample $m \times k'$ input samples to find two additional independent design

matrices, which we call $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$. We assemble $\overline{\mathbf{C}^{(i)}}$ from these design matrices as before, then we use the approximation model to find

$$\overline{\mathbf{Y}}^{(\mathbf{A}_2)} = \begin{bmatrix} \overline{Y}_1^{(\mathbf{A}_2)} \\ \vdots \\ \overline{Y}_{k'}^{(\mathbf{A}_2)} \end{bmatrix} = \begin{bmatrix} f^{(2)}(\overline{\mathbf{A}}_{1,:}) \\ \vdots \\ f^{(2)}((\overline{\mathbf{A}}_{k',:}) \end{bmatrix}, \tag{5.23}$$

and

$$\overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})} = \begin{bmatrix} \overline{Y}_1^{(\mathbf{C}_2^{(i)})} \\ \vdots \\ \overline{Y}_{k'}^{\mathbf{C}_2^{(i)}} \end{bmatrix} = \begin{bmatrix} f^{(2)}(\overline{\mathbf{C}^{(i)}}_{1,:}) \\ \vdots \\ f^{(2)}((\overline{\mathbf{C}^{(i)}}_{k',:}) \end{bmatrix}. \tag{5.24}$$

Using $\overline{\mathbf{Y}}^{(\mathbf{A}_2)}$ and $\overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$ in (5.17) gives $\overline{S}_i^{(2)}$. Finally we find $\tilde{S}_i^{\text{CVB}}$ as

$$\tilde{S}_i^{\text{CVB}} = \tilde{S}_i - \alpha(\tilde{S}_i^{(2)} - \overline{S}_i^{(2)}). \tag{5.25}$$

Likewise we can estimate the total-order indices by following the same process, substituting $\mathbf{A}$, $\mathbf{A}_2$, and $\overline{\mathbf{A}}_2$ for $\mathbf{B}$, $\mathbf{B}_2$ and $\overline{\mathbf{B}}_2$ to find $\tilde{S}_{T_i}$, $\tilde{S}_{T_i}^{(2)}$, and $\overline{S}_{T_i}^{(2)}$. We can then find $\tilde{S}_{T_i}^{\text{CVB}}$ as

$$\tilde{S}_{T_i}^{\text{CVB}} = \tilde{S}_{T_i} - \alpha(\tilde{S}_{T_i}^{(2)} - \overline{S}_{T_i}^{(2)}). \tag{5.26}$$

An algorithm for estimating the Sobol first-order and total-order indices using control variates with estimated means and budget control can be found in Algorithm 10 in Appendix A.

We now test $\tilde{S}_i^{\text{CVB}}$ and $\tilde{S}_{T_i}^{\text{CVB}}$ using the $f^{(2)}$ approximation to the Ishigami function

$$f^{(2)}(X_1, X_2, X_3) = \sin(X_1) + 0.95a\sin^2(X_2) + bX_3^4\sin(X_1).$$

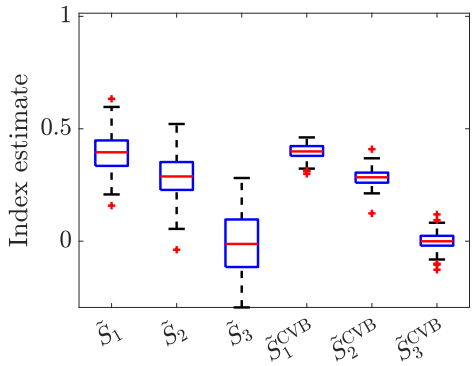
We select three values of $P = 100, 400, 1600$ and calculate 100 replicate values of $\tilde{S}_i$ and $\tilde{S}_{T_i}$, using a Monte Carlo method, as in Algorithm 9. We then use Algorithm 10 to get 100 replicates of estimates of all Sobol indices with each P for each X_i, using the CVB method. This will allow us to determine if the CVB method is more

accurate in estimating the Sobol indices, and if it results in lower variance than the Monte Carlo method alone. In Figure 5.4 we show boxplots for each value of P as before. We can see that the CVB method reduces the variance of all estimates significantly, while having the same mean estimates of S_i and S_{T_i} as the Monte Carlo method. This shows the CVB method used to estimate Sobol indices is an improvement over standard Monte Carlo.

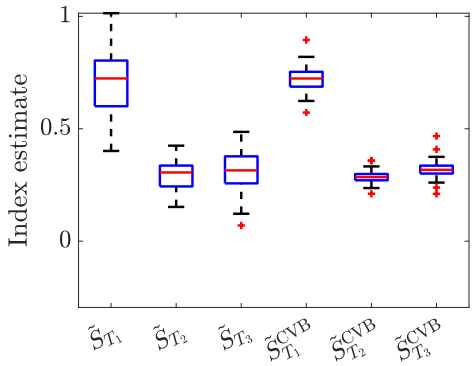
5.4 Confidence intervals for estimated Sobol indices

Often the primary interest in Sobol indices is not their precise values, but ranking the parameters in order of importance in terms of their contribution to the output variance. In examples such as the Ishigami equation we can find the ranking of the inputs using repeated Monte Carlo estimates, but calculating replicates of the estimates of Sobol indices can become very computationally expensive for larger problems, and we do not know how accurate the estimates are without replicates. To save computational costs, and find the accuracy of the estimates,we want to find confidence intervals for Sobol indices using bootstrapping as we did when estimating the mean.

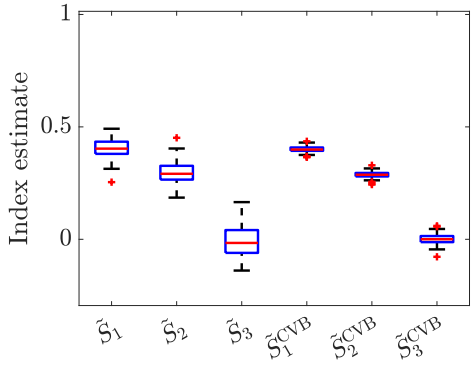
When finding the confidence intervals for the mean using bootstrapping in §3.2, we resampled evaluations of $f^{(1)}$, with replacement, and found a mean value for each resampling. In contrast, when finding confidence intervals of Sobol indices using Monte Carlo, we must find $3 \times P$ realisations of $f^{(1)}$, to evaluate $\mathbf{Y}^{(\mathbf{A})}$, $\mathbf{Y}^{(\mathbf{B})}$ and $\mathbf{Y}^{(\mathbf{C}^{(i)})}$ from (5.13). We can then resample $\mathbf{Y}^{(\mathbf{A})}$, $\mathbf{Y}^{(\mathbf{B})}$, and $\mathbf{Y}^{(\mathbf{C}^{(i)})}$ R_b times to calculate R_b estimates of $\tilde{S}_i$ and $\tilde{S}_{T_i}$ for each X_i. We then find their $\gamma/2$ and $1-(\gamma/2)$ quantiles to get a single realisation of the confidence intervals. We consider the only cost of bootstrapping confidence intervals to be evaluating $f^{(1)}$, as the cost of assembling $\tilde{S}_i$, $\tilde{S}_{T_i}$, and resampling is negligible. An algorithm for finding the $1-\gamma\%$ confidence intervals for the Sobol first-order and total-order indices can be



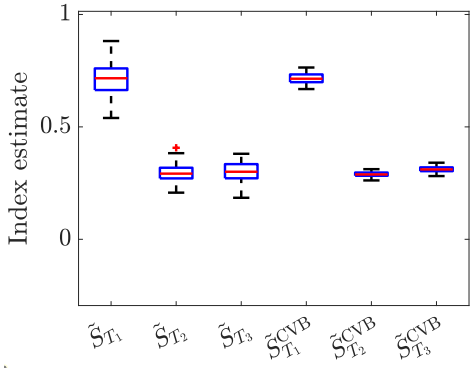
(a) Ishigami $\tilde{S}_i$ and $\tilde{S}_i^{\text{CVB}}$ with $P = 100$.



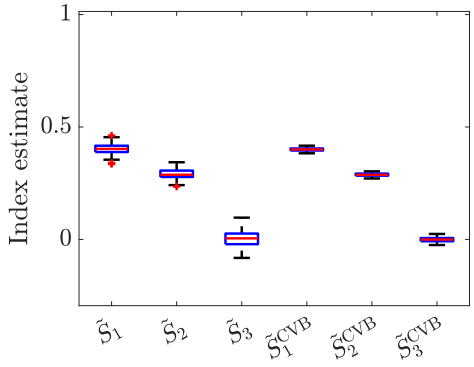
(b) Ishigami $\tilde{S}_{T_i}$ and $\tilde{S}_{T_i}^{\text{CVB}}$ with $P = 100$.



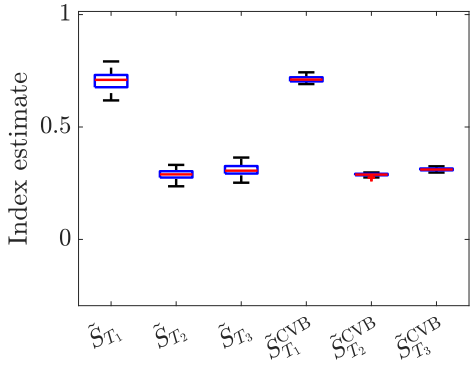
(c) Ishigami $\tilde{S}_i$ and $\tilde{S}_i^{\text{CVB}}$ with $P = 400$.



(d) Ishigami $\tilde{S}_{T_i}$ and $\tilde{S}_{T_i}^{\text{CVB}}$ with $P = 400$.



(e) Ishigami $\tilde{S}_i$ and $\tilde{S}_i^{\text{CVB}}$ with $P = 1600$.



(f) Ishigami $\tilde{S}_{T_i}$ and $\tilde{S}_{T_i}^{\text{CVB}}$ with $P = 1600$.

Figure 5.4: Boxplots of 100 replicates of $\tilde{S}_i$ and $\tilde{S}_{T_i}$ estimated using Monte Carlo and CVB, with increasing P.

found in Algorithm 11 in Appendix A.

We test the method of finding the confidence intervals of the Sobol indices using Monte Carlo with the Ishigami function. We select $P = 100, 200, 400, 800, 1600$, and use the method in Algorithm 11 to find five realisations of 95% confidence intervals for the Ishigami function. We do this to examine whether the confidence intervals get smaller as P increases, as would be expected, and whether the confidence intervals cover the true Sobol indices values from Table 5.1. In Figure 5.5 we can see the confidence intervals of both the first-order and total-order indices cover the true value in most cases as we would hope, and all converge as P is increased. This suggests the method of bootstrapping Sobol indices works as expected. Next we check the coverage of the estimates. We repeat the method from Figure 5.5 to find 10^4 realisations of the confidence intervals, then check how many of these cover the true Sobol indices. We do this for all X_i and five values of P. In Table 5.2 we

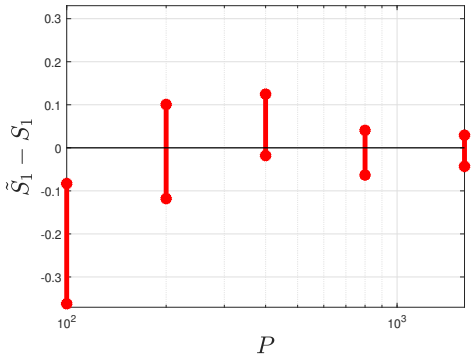
	$\tilde{S}_1$	$\tilde{S}_2$	$\tilde{S}_3$	$\tilde{S}_{T_1}$	$\tilde{S}_{T_2}$	$\tilde{S}_{T_3}$
$P = 100$	94.53%	94.00%	93.13%	93.98%	94.22%	94.07%
$P = 200$	94.35%	94.52%	94.49%	94.19%	93.73%	94.31%
$P = 400$	94.11%	94.82%	95.65%	94.23%	94.53%	94.76%
$P = 800$	94.94%	94.97%	94.71%	95.16%	94.82%	95.26%
$P = 1600$	95.86%	94.63%	94.95%	94.46%	95.20%	95.95%

Table 5.2: Sobol Indices coverage for 10^4 realisations of confidence intervals of Ishigami function, for given P values.

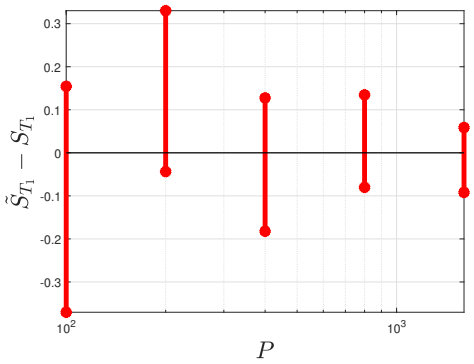
can see the 95% intervals do cover the true values approximately 95% of the time as expected. Bootstrapping therefore is a robust procedure when calculating confidence intervals for estimating Sobol indices.

5.4.1 Multi-fidelity bootstrapping for Sobol indices

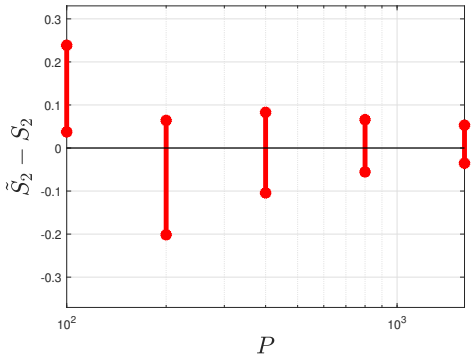
We previously used the CVB method to improve the Monte Carlo estimates of the Sobol indices, and to improve the confidence intervals of the mean. We can now use the same methods to improve the confidence intervals of the Sobol indices. Firstly for the first-order indices, for a budget P, we find β, ϕ, k, and k' as in (5.21). We



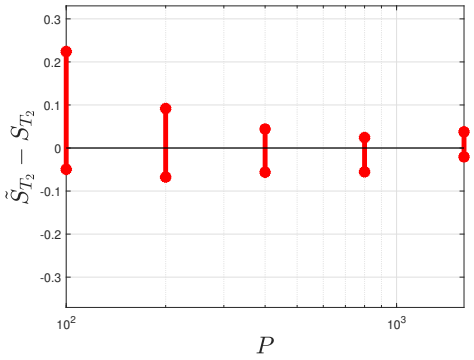
(a) 95% confidence intervals of $\tilde{S}_1$.



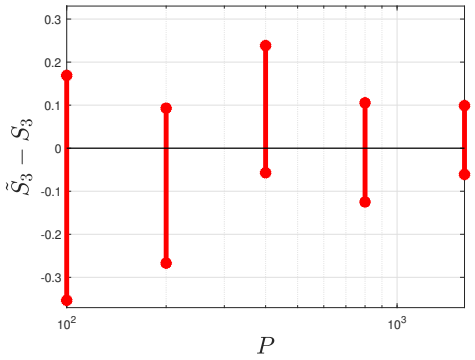
(b) 95% confidence intervals of $\tilde{S}_{T_1}$.



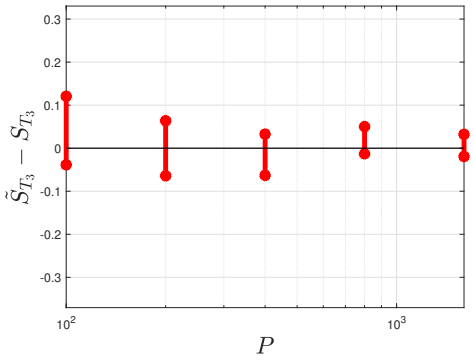
(c) 95% confidence intervals of $\tilde{S}_2$.



(d) 95% confidence intervals of $\tilde{S}_{T_2}$.



(e) 95% confidence intervals of $\tilde{S}_3$.



(f) 95% confidence intervals of $\tilde{S}_{T_3}$.

Figure 5.5: Typical realisations of confidence intervals of the first-order and total-orders of the Ishigami function.

then find $\tilde{\mathbf{Y}}^{(\mathbf{A})}$, $\tilde{\mathbf{Y}}^{(\mathbf{B})}$, $\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})}$, $\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}$, $\tilde{\mathbf{Y}}^{(\mathbf{B}_2)}$, $\tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$, $\overline{\mathbf{Y}}^{(\mathbf{A}_2)}$, $\overline{\mathbf{Y}}^{(\mathbf{B}_2)}$, and $\overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$ as in §5.3. We resample, with replacement, to generate a list containing k values from $\tilde{\mathbf{Y}}^{(\mathbf{A})}$, $\tilde{\mathbf{Y}}^{(\mathbf{B})}$, $\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})}$, $\tilde{\mathbf{Y}}^{(\mathbf{A}_2)}$, $\tilde{\mathbf{Y}}^{(\mathbf{B}_2)}$, and $\tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$, ensuring the same values in the same positions of all six lists are used. This allows us to estimate a new value of $\tilde{S}_i$ and $\tilde{S}_{T_i}$ with no additional evaluations of $f^{(1)}$ or $f^{(2)}$. We then independently resample a list of length k' containing values of $\overline{\mathbf{Y}}^{(\mathbf{A}_2)}$, $\overline{\mathbf{Y}}^{(\mathbf{B}_2)}$, and $\overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})}$, to find new $\overline{S}_i^{(2)}$ and $\overline{S}_{T_i}^{(2)}$, then assemble new $\tilde{S}_i^{\text{CVB}}$ and $\tilde{S}_{T_i}^{\text{CVB}}$ using (5.25). We repeat this process R_b times, to get a list of R_b values of S_i, then take the $\gamma/2$ and $1-\gamma/2$ quantiles to find a realisation of the confidence intervals. An algorithm of the key steps for estimating the $1-\gamma\%$ confidence intervals of the Sobol first-order and total-order indices, using control variates with estimated means and budget control, can be found in Algorithm 12 in Appendix A.

We will test the CVB confidence intervals method for Sobol indices using the Ishigami function. First we use $P = 100, 200, 400, 800, 1600$ and calculate 10^4 realisations of the 95% confidence intervals of the first-order and total-order indices of the Ishigami function for each P. In Table 5.3, we can see the confidence intervals

	$\tilde{S}_1^{\text{CVB}}$	$\tilde{S}_2^{\text{CVB}}$	$\tilde{S}_3^{\text{CVB}}$	$\tilde{S}_{T_1}^{\text{CVB}}$	$\tilde{S}_{T_2}^{\text{CVB}}$	$\tilde{S}_{T_3}^{\text{CVB}}$
$P = 100$	9504	9478	9459	9488	9462	9525
$P = 200$	9499	9482	9497	9497	9501	9461
$P = 400$	9515	9506	9480	9503	9477	9471
$P = 800$	9495	9465	9476	9509	9485	9498
$P = 1600$	9434	9483	9500	9526	9502	9487

Table 5.3: Sobol Indices coverage for 10^4 realisations of confidence intervals of Ishigami function estimated using CVB method, for given P values.

cover the true mean approximately 95% of the time as we would hope, suggesting the method of using bootstrapping to find the confidence intervals of the CVB method to estimate Sobol indices is robust.

Next we want to see if the Sobol first-order and total-order confidence intervals are improved by the CVB method. We do this by selecting $P = 100, 200, 400, 800, 1600$, and finding a single realisation of the confidence intervals of both intervals, using

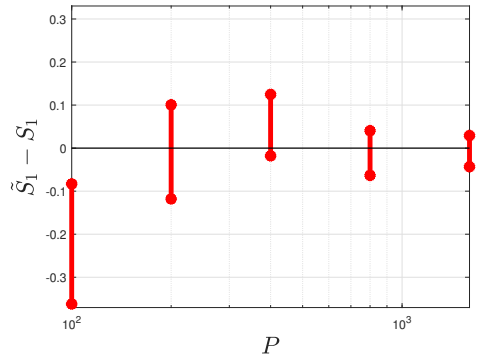
the Monte Carlo method in Algorithm 11. We then find a single realisation of the same confidence intervals using the CVB method in Algorithm 12. We centre all confidence intervals around zero by subtracting the true values of S_i and S_{T_i}, and plot the results in Figures 5.6 and 5.7.

The CVB method results in smaller 95% confidence intervals for all budgets and all indices. The bootstrapping method covers the true value of the CVB method most of the time as expected, and converges to the true value as P is increased. This suggests the method is robust and can be used for the advection-diffusion problem.

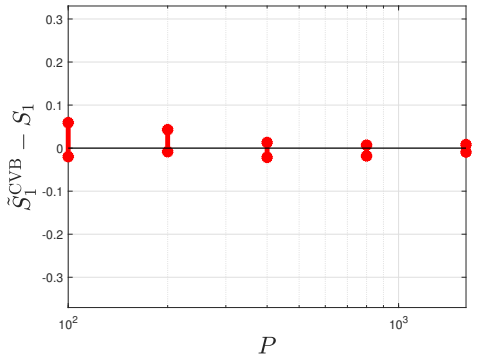
5.5 Estimating Sobol indices for the advection-diffusion problem

Our objective is to estimate Sobol sensitivity indices for the advection–diffusion model introduced in Chapter 3, with uncertain advection strength and diffusion, and a scalar quantity of interest. The high-fidelity model corresponds to the numerical solution of the two-dimensional steady advection–diffusion equation with independent random inputs a (advection strength) and D (diffusion coefficient), and the quantity of interest is the L^2-norm of the concentration at $x = 1$, as defined in §3.1.3. In §5.1-5.4, these indices were estimated using Monte Carlo sampling of the high-fidelity model, and variance reduction was achieved using the CVB framework introduced in Chapter 4. In this section, we combine the high-fidelity model and the correlated low-fidelity approximations described in §4.5 to construct confidence intervals for Sobol indices under a fixed computational budget. In §4.6 we examined several multi-fidelity approaches to finding confidence intervals for the mean of the advection-diffusion problem introduced in §3.1.3. We can now use the low-fidelity approximations and the CVB method to find confidence intervals for the Sobol indices of the inputs of the advection-diffusion model.

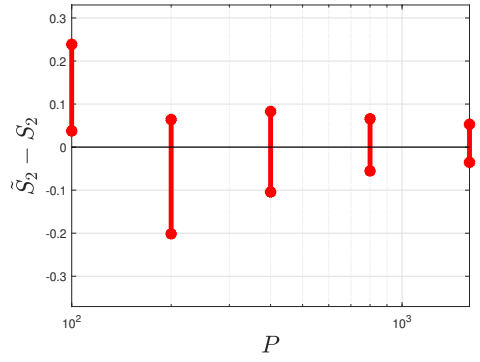
We have five approximations to the advection-diffusion problem: two coarse-grid approximations with 405 and 125 dof, respectively, a linear regression model, a



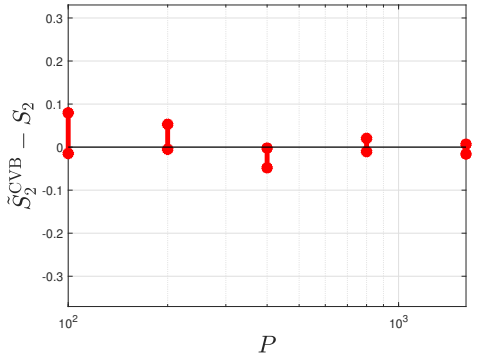
(a) 95% confidence intervals of $\tilde{S}_1$.



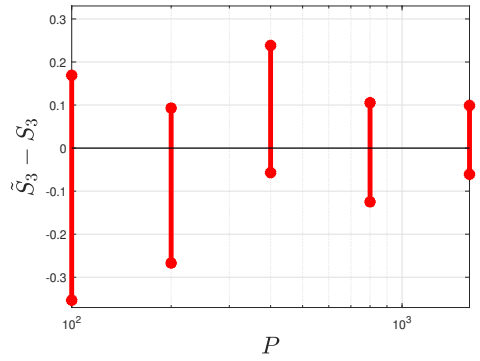
(b) 95% confidence intervals of $\tilde{S}_1^{\mathrm{CVB}}$.



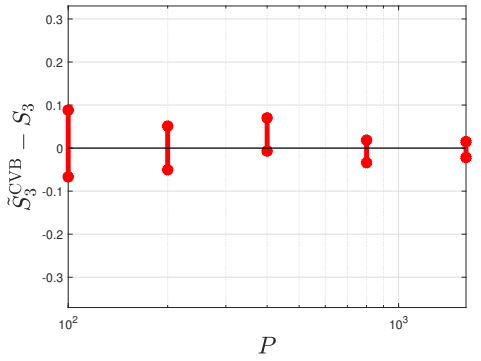
(c) 95% confidence intervals of $\tilde{S}_2$.



(d) 95% confidence intervals of $\tilde{S}_2^{\mathrm{CVB}}$.

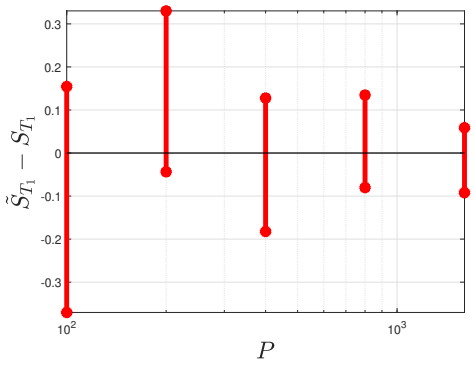


(e) 95% confidence intervals of $\tilde{S}_3$.

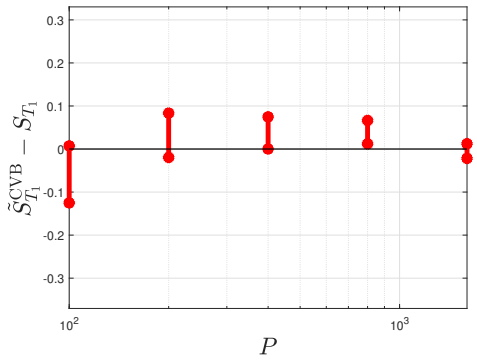


(f) 95% confidence intervals of $\tilde{S}_3^{\mathrm{CVB}}$.

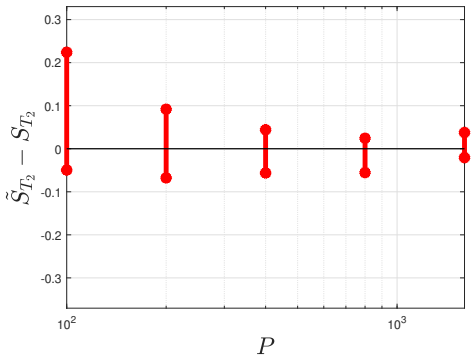
Figure 5.6: Confidence intervals of the first-order indices of the Ishigami function, calculated using bootstrapping with Monte Carlo and CVB.



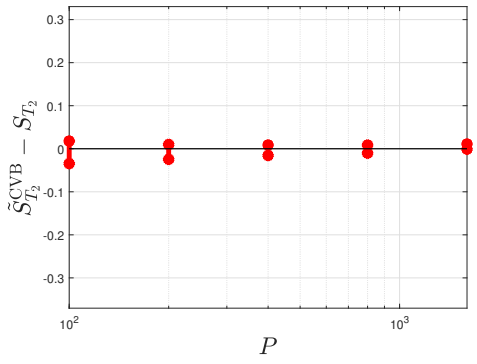
(a) 95% confidence intervals of $\tilde{S}_{T_1}$.



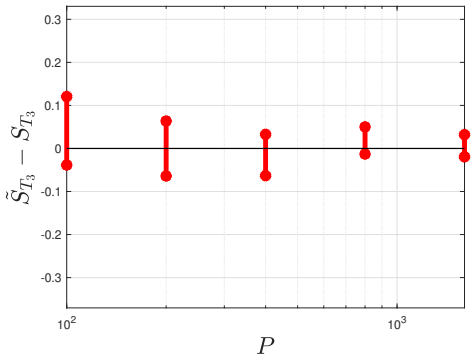
(b) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_1}$.



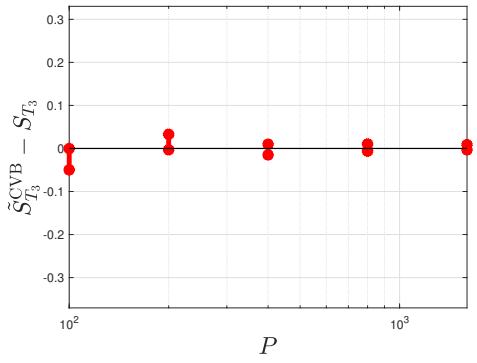
(c) 95% confidence intervals of $\tilde{S}_{T_2}$.



(d) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_2}$.



(e) 95% confidence intervals of $\tilde{S}_{T_3}$.



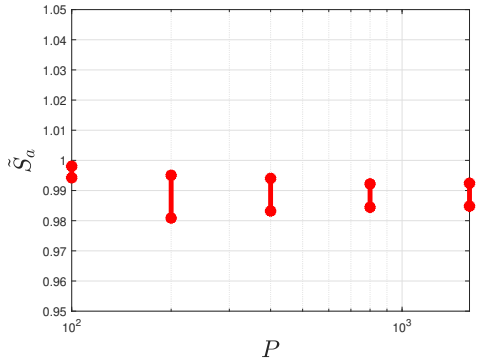
(f) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_3}$.

Figure 5.7: Confidence intervals of the first total-orders of the Ishigami function, calculated using the Monte Carlo and CVB methods.

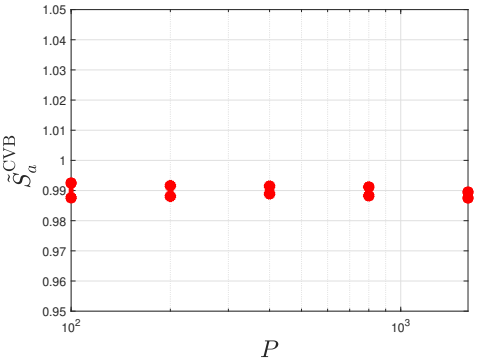
Gaussian process regression approximation and a neural network approximation. We use input parameters for distribution $\mathcal{I}_{\text{high}}$ from (3.9), and five values of the budget, $P = 100, 200, 400, 800, 1600$. Beginning with the first-order indices we find a single realisation of the 95% confidence intervals of S_i, for each P using the bootstrapping method in Algorithm 11. We compare this to five realisations of the confidence intervals using the CVB method in Algorithm 12, using each of the five low-fidelity approximations. This allows us to examine the convergence of the Sobol indices as P is increased, the improvement over the Monte Carlo method, and the importance of the uncertain parameters to the variance of the advection-diffusion model.

Using the parameter distributions in $\mathcal{I}_{\text{high}}$, the estimated first-order indices for a and D are shown in Figures 5.8 and 5.9, respectively. As $\mathcal{I}_{\text{high}}$ corresponds to an advection-dominated regime, Figures 5.8 and 5.9 have been plotted using different y-axis scales for clarity. The advection-diffusion problem is advection dominated. The bootstrapping method gives confidence intervals for first-order indices for the advection parameter with size near zero. The size of the confidence intervals for the first-order indices of a are made narrower by using CVB with coarse-grid approximations, but are not reliably narrower using data-informed approximations. However, each confidence interval is only a single realisation, and the confidence intervals do converge towards the same value as the budget increases, with all data-informed approximations resulting in smaller confidence intervals at the largest budget. The confidence intervals of the first-order indices for the diffusion parameter are made narrower using CVB with all five low-fidelity approximations, with the neural network approximation resulting in the smallest intervals.

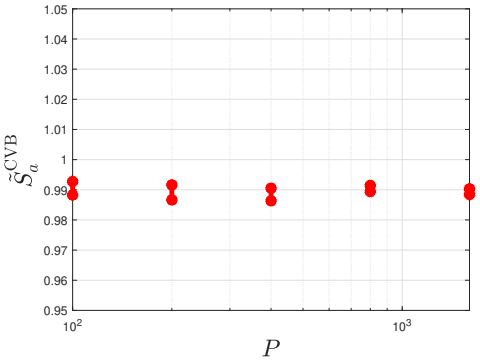
Next we examine the total-order indices in Figures 5.10 and 5.11, which again use different y-axis scales for clarity. The total-order indices also show the advection is responsible for almost all of the variance, but the CVB method reduces the size of the confidence intervals using all approximations, with the neural network approximation resulting in the narrowest confidence intervals. The confidence intervals for $\tilde{S}_{T_D}$ are narrower than $\tilde{S}_{T_a}$, but the data-informed approximations do not consistently result



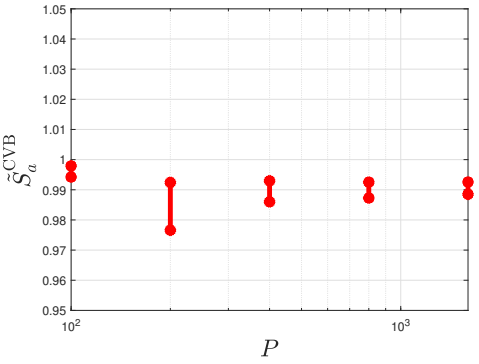
(a) 95% confidence intervals $\tilde{S}_a$.



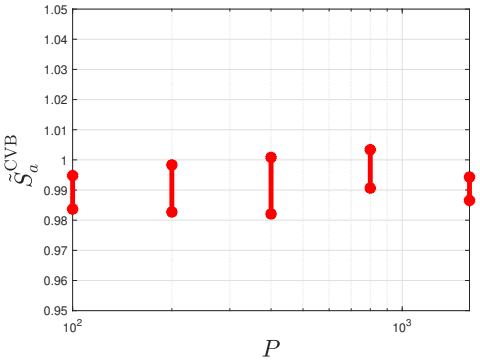
(b) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated with a coarse-grid approximation with 405 dof.



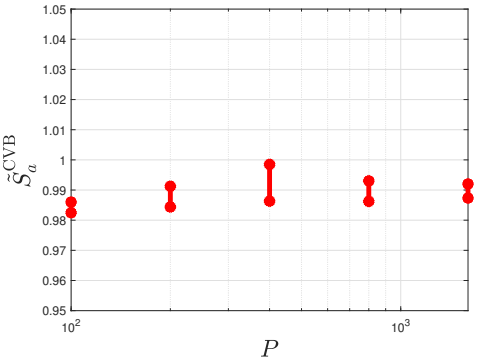
(c) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated with a linear regression model approximation.

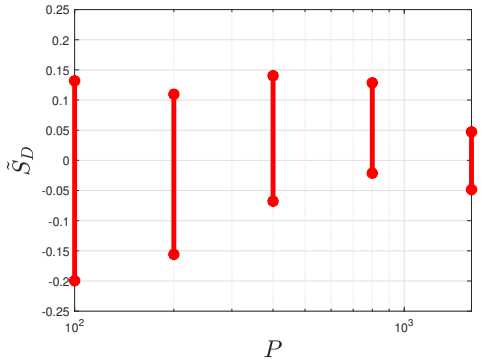


(e) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated with a Gaussian process regression approximation.

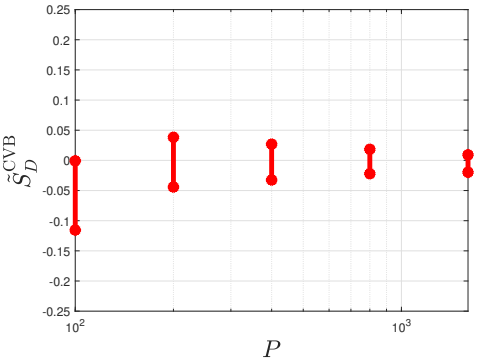


(f) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated with a neural network approximation.

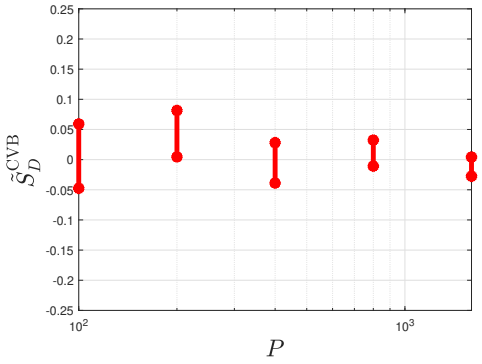
Figure 5.8: 95% confidence intervals of Sobol first-order indices of a with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{high}}$.



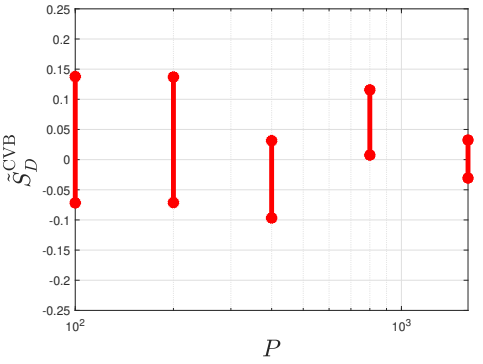
(a) 95% confidence intervals $\tilde{S}_D$.



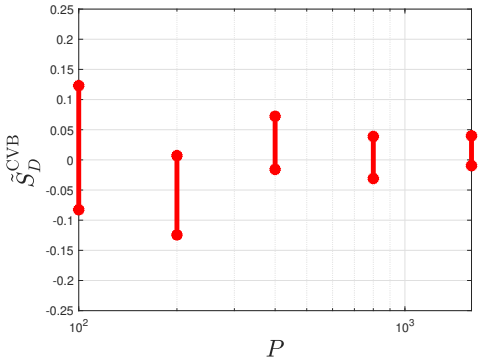
(b) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated with a coarse-grid approximation with 405 dof.



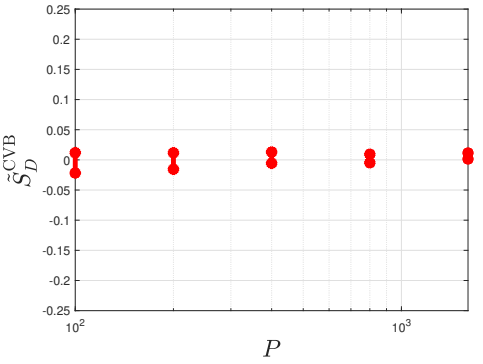
(c) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated with a linear regression model approximation.

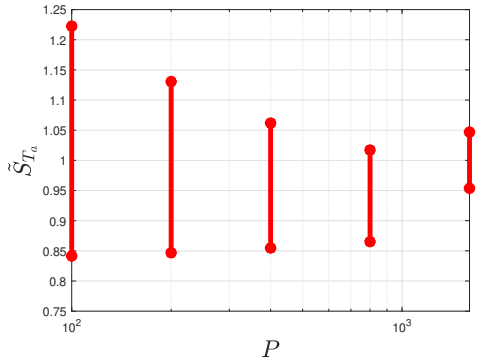


(e) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated with a Gaussian process regression approximation.

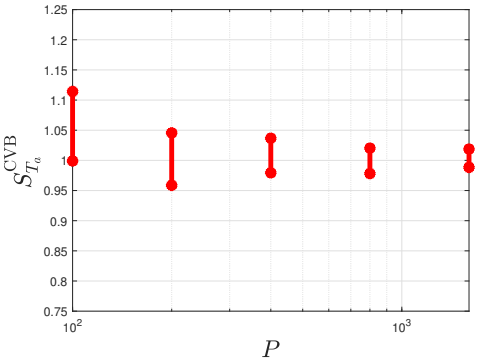


(f) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated with a neural network approximation.

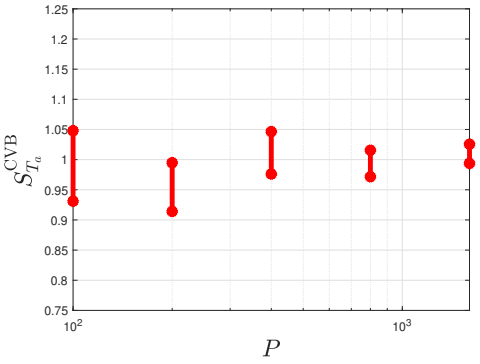
Figure 5.9: 95% confidence intervals of Sobol first-order indices of D with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{high}}$.



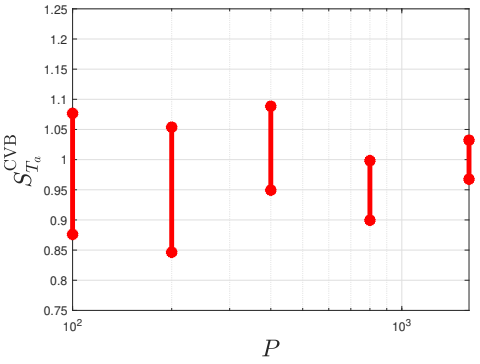
(a) 95% confidence interval of $\tilde{S}_{T_a}$.



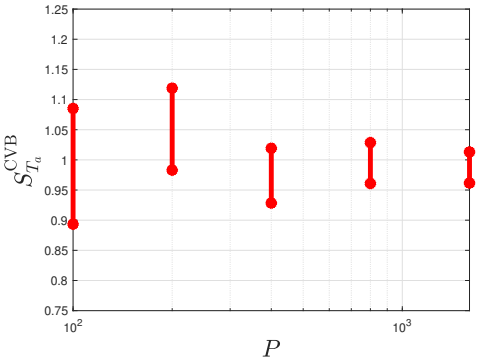
(b) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_a}$, calculated using CVB with a coarse-grid approximation with 405 dof.



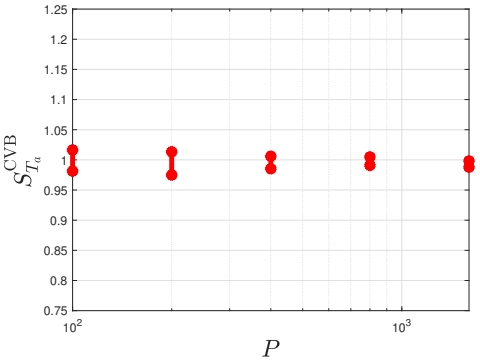
(c) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_a}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_a}$, calculated using CVB with a linear model approximation.

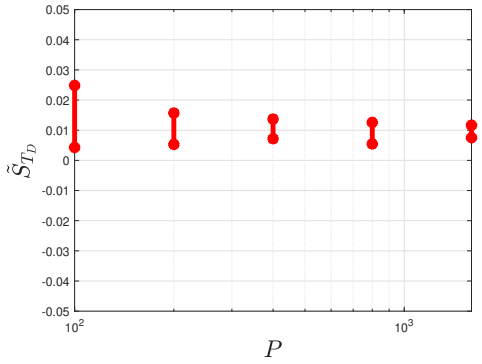


(e) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_a}$, calculated using CVB with a Gaussian process regression approximation.

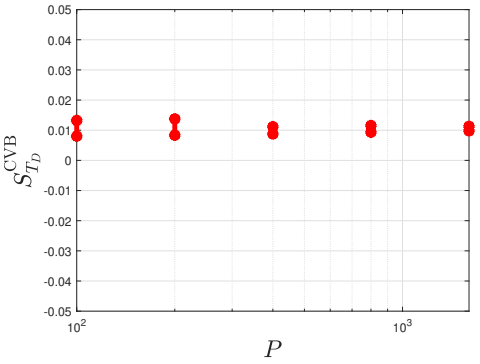


(f) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_a}$, calculated using CVB with a neural network approximation.

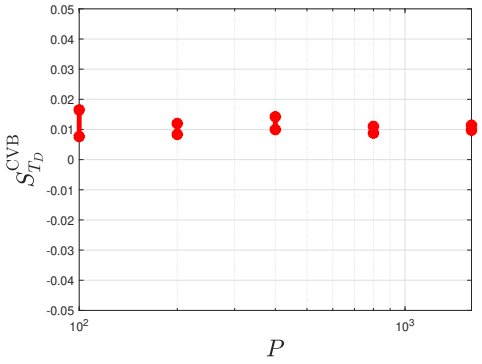
Figure 5.10: 95% confidence intervals of Sobol total-order indices of a with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{high}}$.



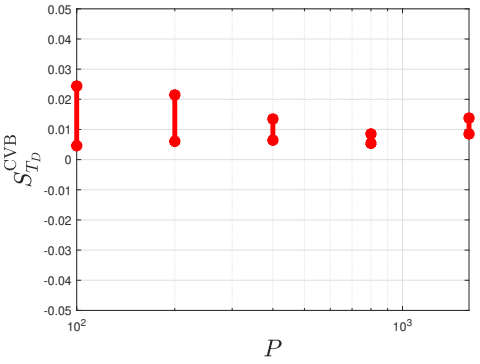
(a) 95% confidence interval of $\tilde{S}_{T_D}$.



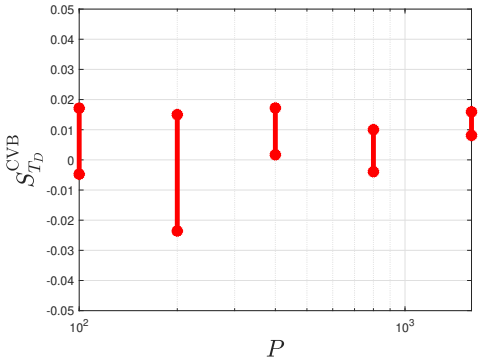
(b) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_D}$, calculated using CVB with a coarse-grid approximation with 405 dof.



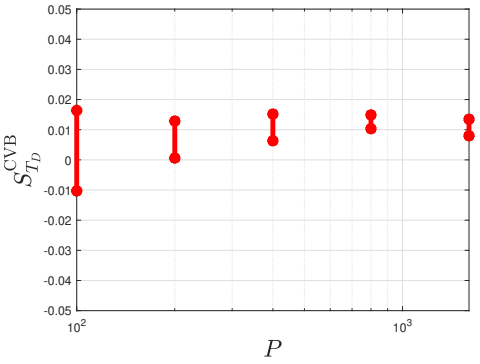
(c) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_D}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_D}$, calculated using CVB with a linear model approximation.



(e) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_D}$, calculated using CVB with a Gaussian process regression approximation.



(f) 95% confidence interval of $\tilde{S}^{\mathrm{CVB}}_{T_D}$, calculated using CVB with a neural network approximation.

Figure 5.11: 95% confidence intervals of Sobol total-order indices of D with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{high}}$.

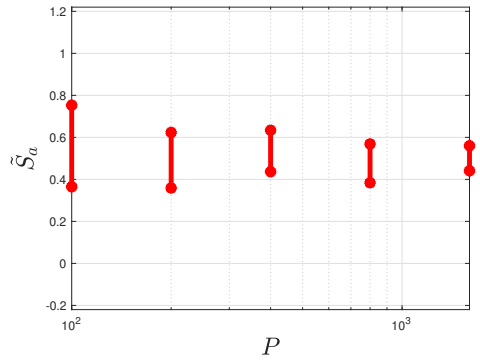
in narrower confidence intervals. Notably, the first-order index for D is close to zero, while its total-order index is approximately 0.01. This indicates that D contributes very little to the variance of the quantity of interest on its own, but that a small interaction exists between D and the wind speed a. The total-order index for a is approximately 1, comprising its dominant first-order effect (around 0.99) and this small interaction with D. However these are only a single realisation of the confidence intervals, and all confidence intervals cover the same values when $P = 1600$.

We now repeat the above experiment with inputs sampled from distribution $\mathcal{I}_{\text{low}}$ from (3.10). This allows us to examine the importance of the parameters when the advection-diffusion model is not advection dominated. In Figures 5.12 and 5.13, we can see the advection is slightly more important than the diffusion, but the problem is not as advection dominated as it was with distribution $\mathcal{I}_{\text{high}}$. Interestingly, the confidence intervals calculated using Monte Carlo, Figures (5.12)(a) and (5.13)(a), overlap until $P = 1600$, so we could not say definitively which parameter is more important with a smaller budget. However, using the CVB method, we can say for sure advection is more important with just $P = 100$ in some cases, showing the usefulness of the CVB method. The confidence intervals for the first-order indices do not appear to sum to one, suggesting an interaction effect between advection and diffusion, which we can examine by looking at the total-order indices.

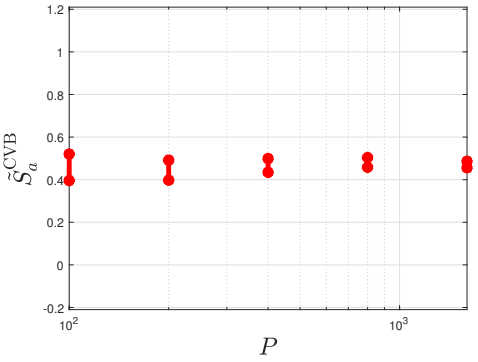
In Figures 5.14 and 5.15, we can see once again that advection causes more of the variance in the model than diffusion. We can also see the CVB method gives non-overlapping confidence intervals for a lower budget, allowing for cheaper ranking of parameters.

5.6 Conclusions

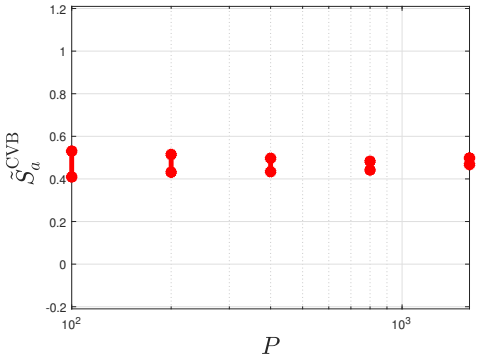
In this chapter we examined global sensitivity analysis using Sobol indices. In §5.1 we motivated methods for estimating Sobol indices. In §5.2 we compared high-fidelity methods of estimating the Sobol first- and total-order indices of the Ishigami function. We compared methods of estimating Sobol indices by plotting the MSE of



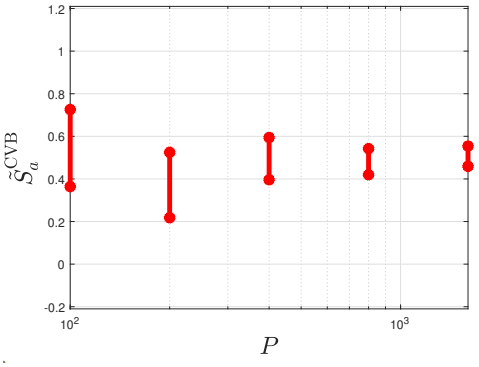
(a) 95% confidence interval of $\tilde{S}_a$.



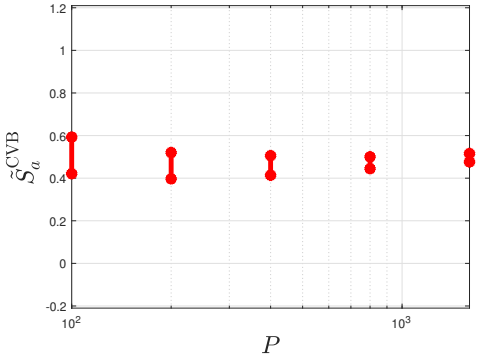
(b) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 405 dof.



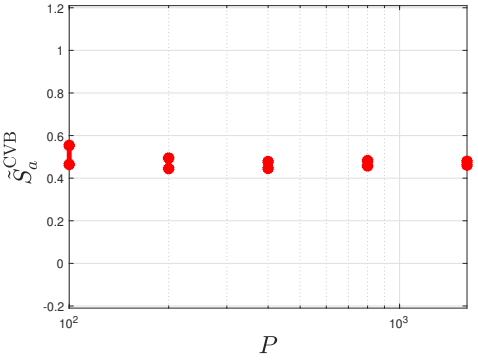
(c) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated using CVB with a linear model approximation.

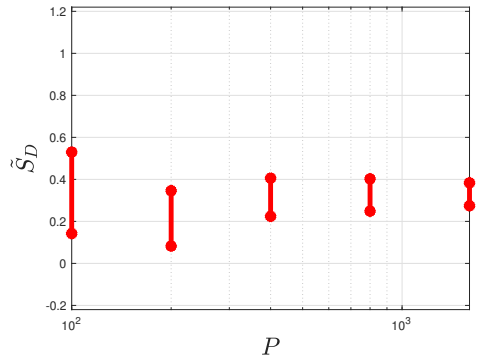


(e) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated using CVB with a Gaussian process regression approximation.

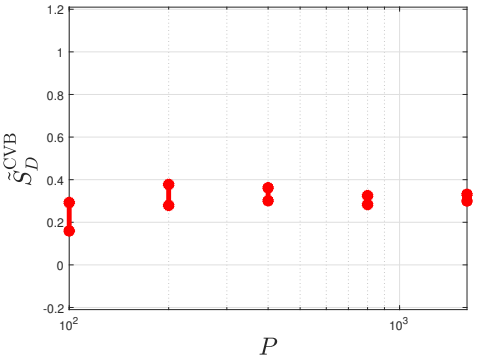


(f) 95% confidence interval of $\tilde{S}_a^{\mathrm{CVB}}$, calculated using CVB with a neural network approximation.

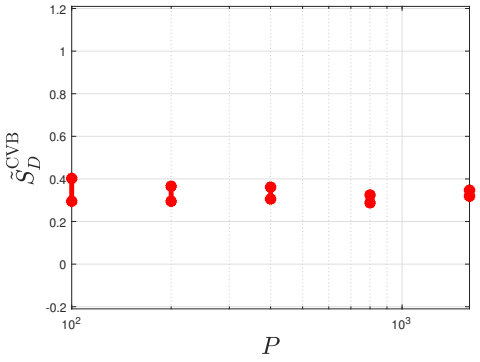
Figure 5.12: 95% confidence intervals of Sobol first-order indices of a with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{low}}$.



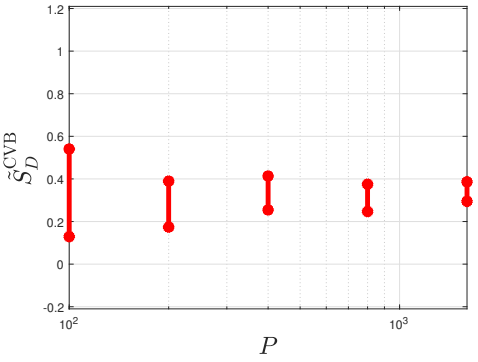
(a) 95% confidence interval of $\tilde{S}_D$.



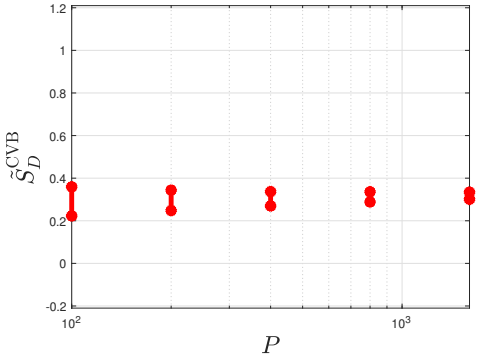
(b) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 405 dof.



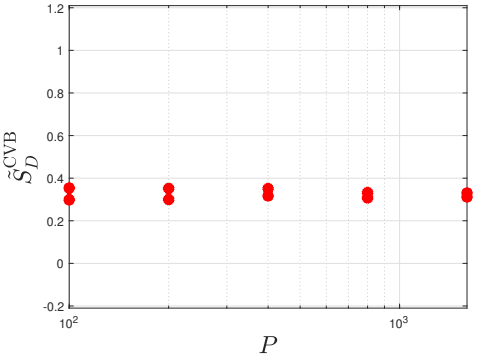
(c) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated using CVB with a linear model approximation.

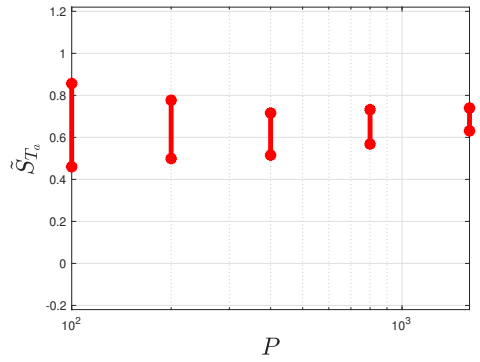


(e) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated using CVB with a Gaussian process regression approximation.

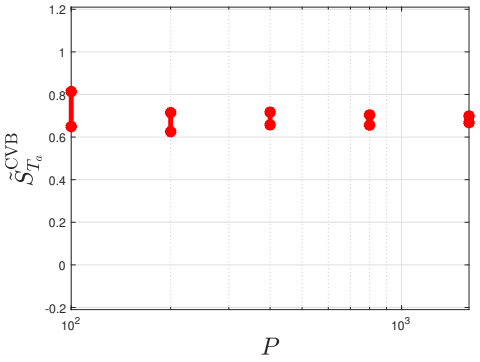


(f) 95% confidence interval of $\tilde{S}_D^{\mathrm{CVB}}$, calculated using CVB with a neural network approximation.

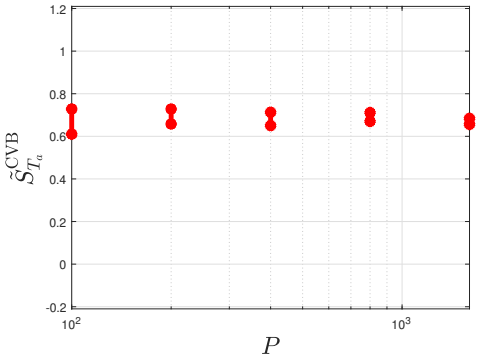
Figure 5.13: 95% confidence intervals of Sobol first-order indices of D with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{low}}$.



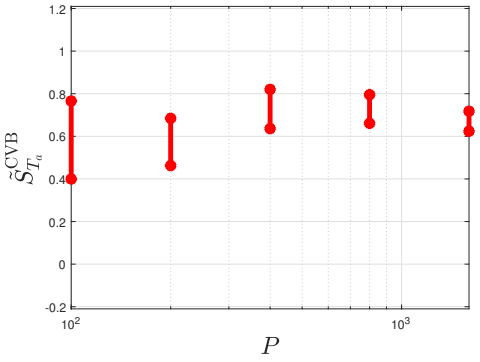
(a) 95% confidence interval of $\tilde{S}_{T_a}$.



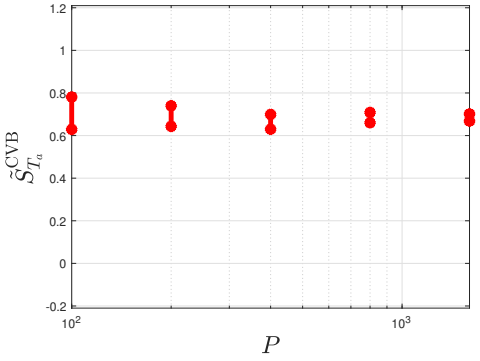
(b) 95% confidence interval of $\tilde{S}_{T_a}^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 405 dof.



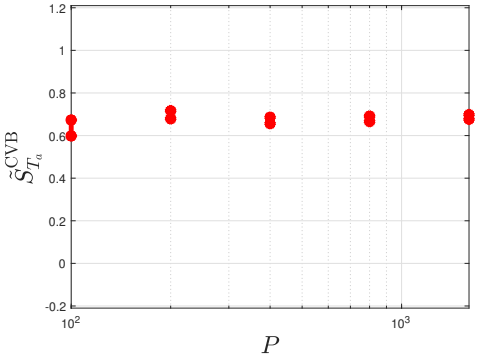
(c) 95% confidence interval of $\tilde{S}_{T_a}^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_{T_a}^{\mathrm{CVB}}$, calculated using CVB with a linear model approximation.

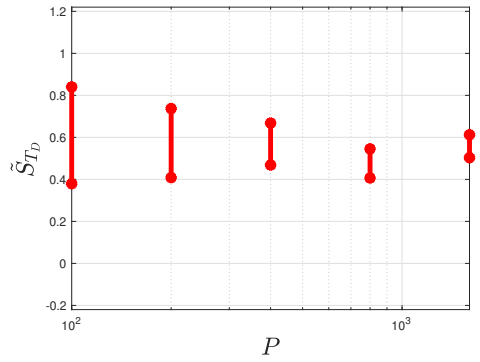


(e) 95% confidence interval of $\tilde{S}_{T_a}^{\mathrm{CVB}}$, calculated using CVB with a Gaussian process regression approximation.

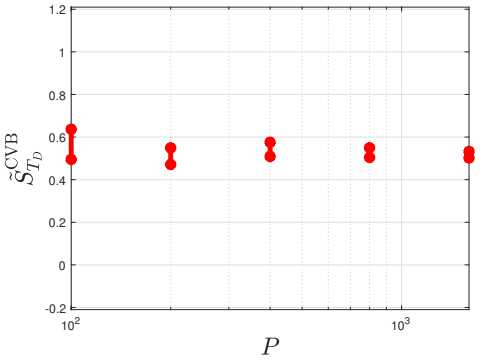


(f) 95% confidence interval of $\tilde{S}_{T_a}^{\mathrm{CVB}}$, calculated using CVB with a neural network approximation.

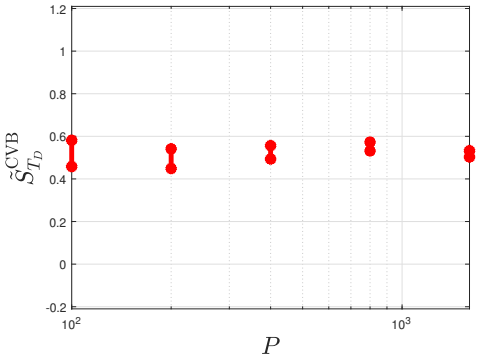
Figure 5.14: 95% confidence intervals of Sobol total-order indices of a with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{low}}$.



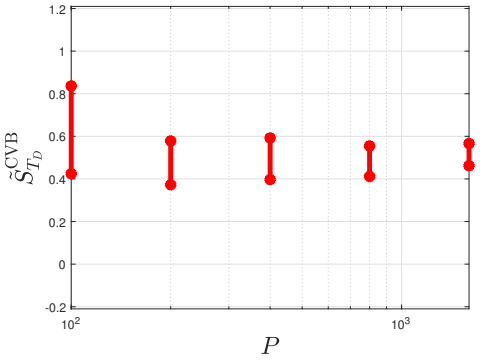
(a) 95% confidence interval of $\tilde{S}_{T_D}$.



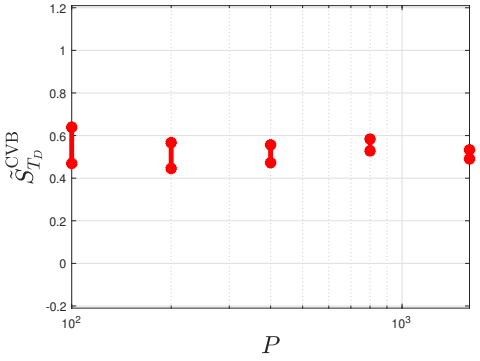
(b) 95% confidence interval of $\tilde{S}_{T_D}^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 405 dof.



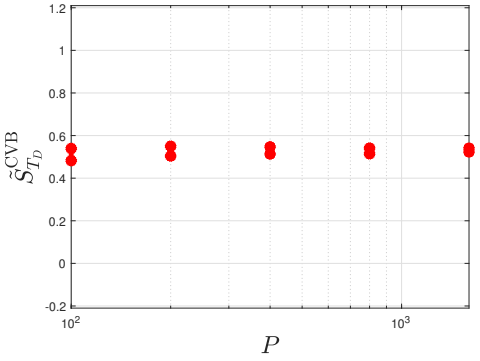
(c) 95% confidence interval of $\tilde{S}_{T_D}^{\mathrm{CVB}}$, calculated using CVB with a coarse-grid approximation with 125 dof.



(d) 95% confidence interval of $\tilde{S}_{T_D}^{\mathrm{CVB}}$, calculated using CVB with a linear model approximation.



(e) 95% confidence interval of $\tilde{S}_{T_D}^{\mathrm{CVB}}$, calculated using CVB with a Gaussian process regression approximation.



(f) 95% confidence interval of $\tilde{S}_{T_D}^{\mathrm{CVB}}$, calculated using CVB with a neural network approximation.

Figure 5.15: 95% confidence intervals of Sobol total-order indices of D with an increasing P, and inputs sampled from distribution $\mathcal{I}_{\mathrm{low}}$.

the Sobol indices of the Ishigami function using multiple methods and found that the estimated MSE was lowest using the Sobol-Janon and Janon-Modol methods. We plotted boxplots of repeated estimates of the Sobol indices of the Ishigami function in Figure 5.3, finding the variance of the Sobol estimates converged as the budget used to estimate them, P, increased.

In §5.3, we presented a novel method for estimating Sobol indices using the control variates with estimated means and budget control method. We tested this method using the Ishigami function and plotted repeated Sobol indices estimates for several budgets in Figure 5.4, and found that the Sobol estimates have a lower variance when using the CVB method, and that as P increased the variance of the CVB method was reduced.

In §5.4, we presented a method of finding confidence intervals of the Sobol indices using bootstrapping. In Figure 5.5 we showed the confidence intervals of the Sobol indices of the Ishigami function converged as P was increased, and in Table 5.1 we showed the confidence intervals cover the true Sobol indices value approximately 95% of the time as expected. We then introduced a novel method of estimating Sobol indices using the CVB method and bootstrapping. We plotted the Sobol indices of the Ishigami function using the CVB method, confirming the CVB method results in smaller confidence intervals than the MC method, and in Table 5.3 we confirmed the confidence intervals covered the true Sobol indices values 95% of the time as expected.

In §5.5, we used the MC method to find the Sobol indices of the advection-diffusion model with input distribution $\mathcal{I}_{\text{high}}$. We then used the low-fidelity approximations to the advection-diffusion model introduced in §4.6 to estimate the Sobol indices using the CVB method. We found that with input distribution $\mathcal{I}_{\text{high}}$, the problem was entirely advection dominated, and the CVB method reduced the width of the confidence intervals significantly with each approximation. We repeated the experiment using the input distribution $\mathcal{I}_{\text{low}}$, and found that while advection was more important, it was more balanced than with $\mathcal{I}_{\text{high}}$. In the examples with in-

put distribution $\mathcal{I}_{\text{low}}$, the CVB method allowed us to definitively rank the inputs in terms of importance with a smaller budget than the MC method, as the confidence intervals overlapped when using MC with a small budget but did not when using the same budget with CVB.

Chapter 6

Global sensitivity analysis of an eddy diffusion model of pollutant dispersion

The aim of this chapter is to extend the advection–diffusion model to a more realistic setting by introducing additional uncertain parameters informed by real-world data, and to assess the performance of the proposed variance reduction and sensitivity analysis methods in this context. We introduce uncertainty in wind speed, diffusion, source strength, and wind direction, and define appropriate probability distributions for each. The CVB method is then applied to estimate Sobol sensitivity indices and their confidence intervals using a range of low-fidelity approximations. Numerical results are presented to examine estimator performance and to identify the dominant contributors to output variance in the extended model.

6.1 Eddy diffusion model

In previous chapters, the advection-diffusion model was not grounded in any specific physical problem, and as such it was presented in dimensionless form. In this chapter, we present an advection-diffusion equation in dimensional form, introduce additional

sources of uncertainty to make the model more physically realistic, and increase the complexity of the model. We then examine the effectiveness of surrogate models and CVB methods for estimating Sobol indices of the new model. We now consider the equation

$$-D\nabla^2 u + a\boldsymbol{w}\cdot\nabla u = \sum_{i=1}^{3} q_i f_i \quad \forall x \in \Omega, \quad u = 0 \text{ on } \partial\Omega. \tag{6.1}$$

where q_i is the source strength, measured in kg s^{-1}. We model three points of pollutant release with

$$f_i(x,y) = \frac{1}{2\pi S^2}\ \exp\left(-\frac{(x-x_{s_i})^2+(y-y_{s_i})^2}{2S^2}\right) \tag{6.2}$$

representing a localised source centred at release points (x_{s_i}, y_{s_i}) modelled here as a Gaussian function. Here $S = 0.1$ is the spatial spread (standard deviation) of each Gaussian source, measured in kilometres, which is the same for each pollutant release point.

We express the domain size in kilometres, and set the domain to be $\Omega = [0,16] \times [0,2]$. The pollution u is expressed in kg km^{-2}, and we set three points of pollutant release ($P_{s_i} = (x_{s_i}, y_{s_i})$) at $P_{s_1} = (2,1)$ km, $P_{s_2} = (4,1)$ km, and $P_{s_3} = (6,1)$ km. This could be the radius of a large stack, the plume after it has risen from a chimney or stack into the domain, or a distributed source such as an industrial park with several pollutant sources.

6.2 Advection and diffusion terms

In previous chapters, the wind speed was sampled from a uniform distribution, which can be suitable when limited information is available, such as the range of possible values. If wind speed data is available, then the Weibull distribution has commonly been used to fit this data [33, 71, 72, 76], as it has inherently positive support and is

right-skewed. The pdf of a Weibull distribution is given by

$$f_{\text{Weibull}}(x;\lambda_W,\kappa_W) = \begin{cases} \dfrac{\kappa_W}{\lambda_W}\left(\dfrac{x}{\lambda_W}\right)^{\kappa_W-1} e^{-\left(\frac{x}{\lambda_W}\right)^{\kappa_W}}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

where κ_W is a shape parameter, and λ_W is a scale parameter. The shape parameter determines the shape of the distribution, with $\kappa_W > 1$ resulting in a right-skewed bell curve which narrows as $\kappa_W \to \infty$. The scale parameter shifts the distribution left or right on the x-axis, with a larger λ_W value moving the centre of the towards larger values on the x-axis. The mean of the Weibull distribution is

$$\mathbb{E}[f_{\text{Weibull}}] = \lambda_W \, \Gamma\left(1 + \frac{1}{\kappa_W}\right), \tag{6.3}$$

where Γ is the gamma function.

In order to fit realistic shape and scale parameters for the Weibull distribution, we use real world data. The Centre for Environmental Data Analysis (CEDA) publishes data from the Met Office with UK mean wind information, including mean wind speed and direction, as well as the direction, speed, and time of the maximum gust, recorded over one or more hours. The data is collected from observation stations across the UK with records available from 1949 to the present [92]. We can use this data to fit a realistic distribution. In this chapter the data used corresponds to mean daily wind speeds in the Arrochar region of Stirlingshire, from January 1$^{\text{st}}$ to December 27$^{\text{th}}$ 1991. We use the `wblfit` function in MATLAB to estimate the shape and scale parameters of the Weibull distribution using maximum likelihood estimation. The resulting distribution has $\lambda_W = 0.0024$, and $\kappa_W = 1.57$, resulting in a mean wind speed of 2.16×10^{-3} km s^{-1}, and the central 95% quantile interval, between the 2.5$^{\text{th}}$ and 97.5$^{\text{th}}$ percentiles, of $[2.308\times10^{-4}, 5.118\times10^{-3}]$ km s^{-1}. Figure 6.1 shows a histogram of the wind speed data used and the pdf of the fitted Weibull distribution. Also indicated are the mean and central 95% quantile interval.

Next, we select a distribution for the diffusion coefficient. Previously, the physics

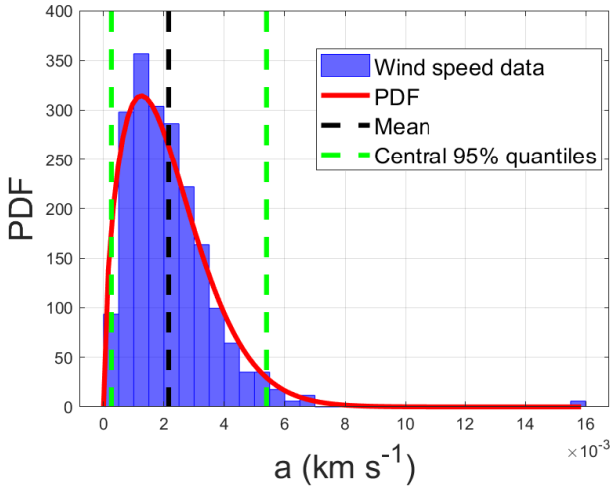


Figure 6.1: Histogram of wind speed data and fitted Weibull distribution.

of the diffusive process represented by the diffusion coefficient D, was left unspecified. Here, diffusion refers specifically to turbulent horizontal eddy diffusion, and not to molecular or atomic diffusion, the latter of which is typically on the order of 10^{-6} to 10^{-5} m^2 s^{-1} [23], and is negligible over a scale of kilometres. Horizontal eddy diffusion captures the mixing caused by atmospheric turbulence and unresolved eddies, particularly in complex urban environments. Horizontal diffusivity is commonly expressed as

$$D = \frac{v z_i^{3/4}}{10(-\kappa_D L)^{1/3}}, \tag{6.4}$$

where v is the friction velocity, z_i the mixing layer height, $\kappa_D = 0.4$ the von Kármán constant, and L the Monin-Obukhov length [48]. However, equation (6.4), as presented in [48], contains a dimensional inconsistency. Horizontal diffusivity should have units of length^2 time^{-1}, whereas the right-hand side of (6.4) has units of length$^{17/12}$ time^{-1}. This expression is transcribed from [127], where the same form appears. Inspection of multiple editions of this text confirms that the dimensional inconsistency is present in the second and subsequent editions. In [127], an alternative expression is also given:

$$D = \frac{w_* z_i}{10}, \tag{6.5}$$

with w_* the convective velocity scale [29], defined as

$$w_* = \left(\frac{g}{T_0} \, z_i \, (\overline{wT})_0 \right)^{1/3}, \tag{6.6}$$

where g/T_0 is the buoyancy parameter and $(\overline{wT})_0$ the kinematic heat flux. The Monin-Obukhov length is given by

$$-L = \frac{v^3}{\kappa_D \, (g/T_0) \, (\overline{wT})_0}. \tag{6.7}$$

Substituting (6.6) and (6.7) into (6.5) gives the corrected form

$$D = \frac{v z_i^{4/3}}{10(-\kappa_D L)^{1/3}}, \tag{6.8}$$

which is dimensionally consistent.

To estimate a realistic distribution for D, we generate synthetic data by sampling each parameter in (6.8) from uniform distributions over plausible physical ranges. For the friction velocity v, a plausible physical range of $[5 \times 10^{-5}, 10^{-3}]$ km s^{-1} is given in [137]. From [128], we have ranges of values for the mixing layer height of $[0.1, 3]$ km, and the Monin-Obukhov length of $[-0.5, -10^3]$ km. Using uniform distributions within these ranges, we generate 10^3 diffusion values and figure 6.2 shows a histogram of the synthetic diffusion data on a linear scale. The histogram of D shows a strong right-skew, with the majority of values concentrated near zero and a long positive tail, suggesting that a log-normal distribution may be an appropriate fit. The pdf of the log-normal distribution [76] is given by

$$f_{\text{log-normal}}(x; \mu_D, \sigma_D) = \frac{1}{x \, \sigma_D \sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu_D)^2}{2\sigma_D^2} \right), \quad x > 0.$$

We fit a log-normal distribution to the synthetic data using the `fitdist` function in MATLAB and the maximum likelihood estimation method, which gives log-mean $\mu_D = -8.90$ and log-standard deviation $\sigma_D = 1.28$. In Figure 6.3 we can see a

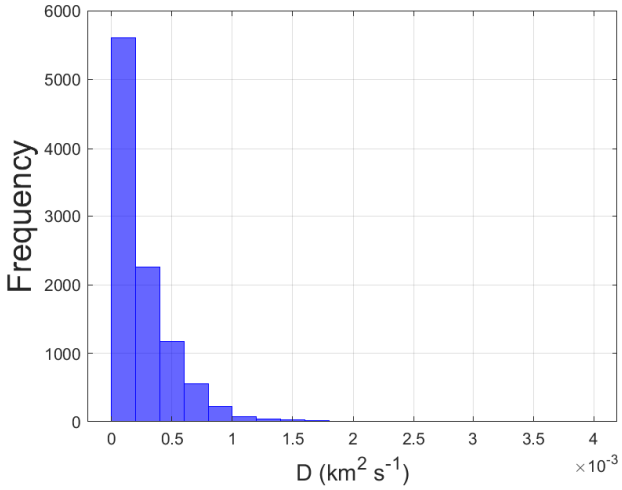


Figure 6.2: Histogram of synthetic diffusion data on a linear scale.

histogram of the synthetic data on a log scale, the fitted pdf, the mean, and the central 95% quantile interval. The diffusion distribution has mean 1.36×10^{-4} km^2

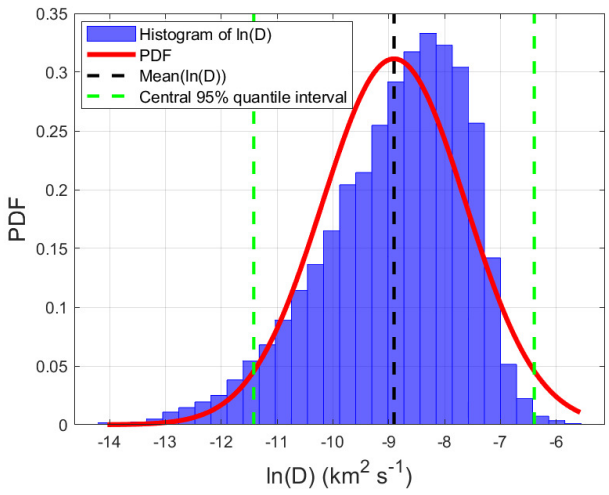
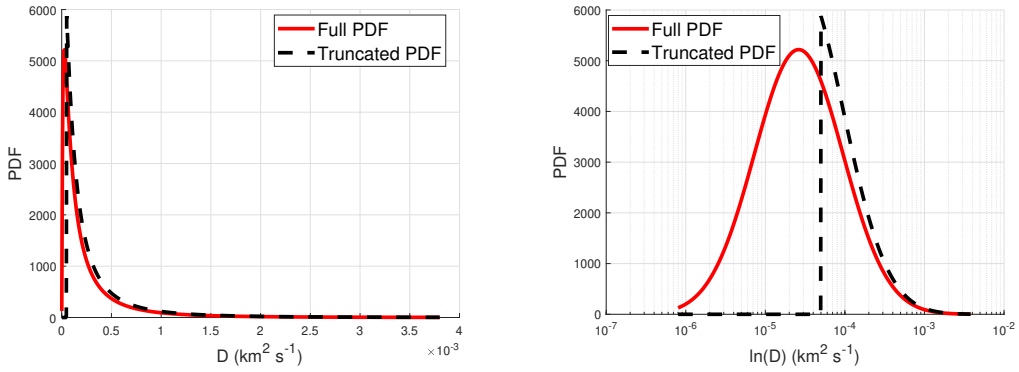


Figure 6.3: Histogram of synthetic diffusion data and fitted log-normal distribution.

s^{-1} and central 95% quantile interval $[1.11 \times 10^{-5}, 1.17 \times 10^{-3}]$ km^2 s^{-1}.

When D is small relative to a, the problem becomes advection-dominated, which can potentially lead to steep gradients or spurious oscillations in the numerical solution [118]. Although SUPG mitigates some of these issues, extremely small values of D can still result in unstable or non-physical solutions. To prevent this, we impose

a lower bound on the diffusion coefficient and truncate the distribution from below at $D = 5 \times 10^{-5}$. Approximately 20% of sampled values lie below this threshold, so when $D \leq 5 \times 10^{-5}$, the value is resampled until $D > 5 \times 10^{-5}$. This ensures that the PDE remains well-posed and numerically stable for all sampled pairs (a, D). Truncating the diffusion reduces its variance, which will likely reduce its effect on the scalar output and result in lower Sobol indices. Alternative approaches could include adopting more aggressive numerical stabilisation, allowing unstable solves and treating numerical failure as part of the output, or reformulating the problem using a measure of the relative strength of advection and diffusion, such as the Péclet number. However, these options would either substantially increase computational cost or introduce sensitivity caused by numerical behaviour rather than the underlying model. The truncation used here therefore represents a compromise that enables robust sensitivity analysis within a well-posed numerical setting. Figure 6.4 shows



(a) Full and truncated pdfs of D on a linear scale.

(b) Full and truncated pdfs of D on a log scale.

Figure 6.4: Full and truncated pdfs of D.

the full and truncated pdfs of D on a linear and log scale. The final truncated D distribution has mean 3.86×10^{-4} km^2 s^{-1} and central 95% quantile interval $[5.4324 \times 10^{-5}, 1.90 \times 10^{-3}]$ km^2 s^{-1}.

6.3 Modelling the source strength

We now wish to make the source strengths, q_i, $i = 1, 2, 3$, more physically realistic. An uncertain source strength can be represented using a gamma distribution, as its support is non-negative and right skewed, reflecting the fact that typical emissions are moderate while larger releases occur less frequently [128]. The gamma distribution has pdf

$$f_{\text{gamma}}(x;\, \alpha_{q_i}, \beta_{q_i}) \;=\; \frac{\beta_{q_i}^{\alpha_{q_i}}}{\Gamma(\alpha_{q_i})}\, x^{\alpha_{q_i}-1} e^{-\beta_{q_i} x}, \qquad x > 0,$$

where Γ is the gamma function, α_{q_i} is a shape parameter, and β_{q_i} is a rate parameter. To find appropriate values of α_{q_i} and β_{q_i}, we use data from [60], which reports engineering estimates of annual airborne particulate emissions from a zinc smelter. We select three of the four pollutant sources provided with values of

$$q_1^{\text{Eng}} = 80 \text{ t yr}^{-1}, \quad q_2^{\text{Eng}} = 35 \text{ t yr}^{-1}, \quad q_3^{\text{Eng}} = 5 \text{ t yr}^{-1}.$$

Converting the emission values into kg s^{-1}, gives engineering estimates of

$$q_1^{\text{Eng}} = 2.54 \times 10^{-3} \text{ kg s}^{-1}, \quad q_2^{\text{Eng}} = 1.11 \times 10^{-3} \text{ kg s}^{-1}, \quad q_3^{\text{Eng}} = 1.59 \times 10^{-4} \text{ kg s}^{-1}.$$

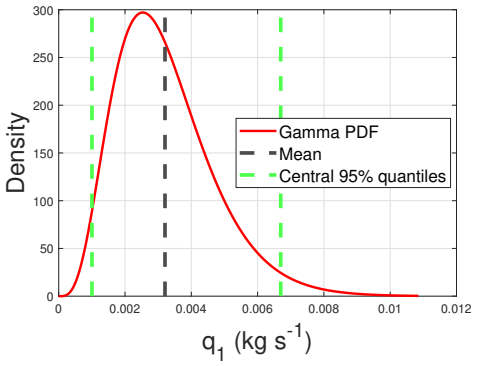
Following the approach in [48], we assume that the mode of the gamma distribution is equal to the engineering estimate, and that 99% of values lie below three times this amount. This implies the constraint

$$\text{ICDF}_{\text{gamma}}(0.99;\, \alpha_{q_i}, \beta_{q_i}) = 3q_i^{\text{Eng}},$$

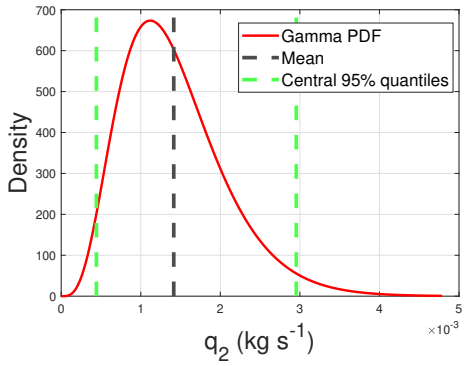
where ICDF denotes the inverse cumulative distribution function. Combining this with the analytical expression for the mode of the gamma distribution,

$$\frac{\alpha_{q_i} - 1}{\beta_{q_i}} = q_i^{\text{Eng}},$$

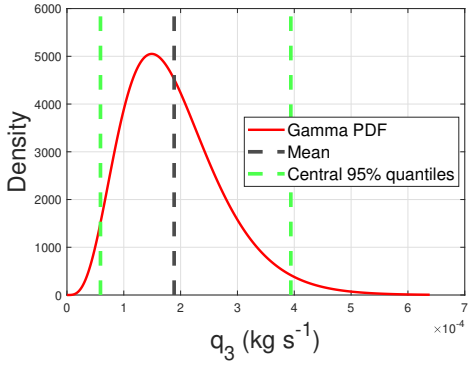
we can solve for α_{q_i} and β_{q_i} for each source. This gives values of $\alpha_{q_i} = 4.73$ for $i = 1, 2, 3$, and $\beta_{q_1} = 6.81 \times 10^{-4}$, $\beta_{q_2} = 2.98 \times 10^{-4}$, $\beta_{q_3} = 4.26 \times 10^{-5}$. In Figure 6.5 we plot the pdfs of the distributions, and the means and central 95% quantile intervals.



(a) Mean, pdf and central 95% quantile range of q_1.



(b) Mean, pdf and central 95% quantile range of q_2.



(c) Mean, pdf and central 95% quantile range of q_3.

Figure 6.5: Probability density functions of the random source strengths q_1, q_2, and q_3.

The mean values and central 95% quantile intervals are shown in Table 6.1.

6.4 Modelling the wind direction

In previous chapters, we assumed a constant wind direction, which is not physically realistic in a pollutant dispersal problem. We now introduce a variable wind

q_i	$\mathbb{E}(q_i)$ ($\mathrm{kg\,s^{-1}}$)	Central 95% quantile range ($\mathrm{kg\,s^{-1}}$)
q_1	3.20×10^{-3}	$[1.00 \times 10^{-3},\ 6.67 \times 10^{-3}]$
q_2	1.40×10^{-3}	$[4.42 \times 10^{-4},\ 2.98 \times 10^{-3}]$
q_3	1.89×10^{-4}	$[5.90 \times 10^{-5},\ 3.94 \times 10^{-4}]$

Table 6.1: Expected values and central 95% quantile ranges for q_i based on fitted gamma distributions.

direction. The most obvious approach would be to sample the wind direction vector $\mathbf{w}$ from a wind direction distribution, while keeping the grid fixed. However, this would require recomputing the advection matrix (see (2.7) in §2.2.2) and the SUPG/upwinding matrix (see (2.11) in §2.3) for each new wind direction sample. Assembling the advection and SUPG/upwinding matrices for each sample would account for the majority of computational cost when solving the advection-diffusion model problem repeatedly.

Alternatively, we could rotate the entire computational grid by rotating all spatial coordinates (x, y) to $(\hat{x}, \hat{y})$ to align the coordinate system with the wind direction. However, this would also require recalculating the matrices for each wind direction sample.

Instead, we introduce a coordinate transformation by rotating all relevant point coordinates, such as pollutant release points and quantities of interest, while keeping the computational grid, wind vector, and finite element matrices fixed. This preserves the mesh structure, finite element connectivity, and zero Dirichlet boundary conditions, and allows the wind direction to vary without requiring matrix reassembly.

We rotate about the coordinates of the third pollutant source P_{s_3}, as it is centrally located on the x-axis. Rotating about this point maximises the range of angles over which the other point sources and the quantity of interest remain within the domain. The transformation rotates by an angle $-\varphi$ about P_{s_3}, mapping physical coordinates $\boldsymbol{x}$ onto transformed coordinates $\hat{\boldsymbol{x}}$ via

$$\hat{\boldsymbol{x}} = P_{s_3} + R(-\varphi)(\boldsymbol{x} - P_{s_3}),$$

where

$$R(-\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$$

is the standard 2D rotation matrix. The computational domain is defined as $\Omega = [0,\, 16] \times [0,\, 2]$. We introduce a rotated coordinate frame $\hat{\Omega} = [0,\, 16] \times [0,\, 2]$. We apply the coordinate transformation to the source locations P_{s_1} and P_{s_2}, and to the quantity of interest location $P_{\mathcal{Q}}$, where the quantity of interest is defined as the pollutant concentration evaluated at $P_{\mathcal{Q}}$; that is,

$$\mathcal{Q} = u(P_{\mathcal{Q}}). \tag{6.9}$$

The wind direction vector is given by $\mathbf{w} = (\cos\varphi, \sin\varphi)$. In the rotated reference frame, the wind direction is $\hat{\mathbf{w}} = (1, 0)$ for all realisations of φ. Figure 6.6 shows both the physical and rotated frames. While the source vector must still be reassembled for each realisation (see (2.8) in §2.2.2), this is outweighed by the computational savings from avoiding repeated assembly of the advection, and SUPG/upwinding matrices.

For the range of mild rotations considered here, the outflow boundary remains sufficiently downstream of the quantity of interest to avoid spurious oscillations arising from the imposition of Dirichlet boundary conditions on an advective outflow. When the coordinate frame is rotated, the quantity of interest may lie closer to the $\hat{y}$-boundaries, however, in the rotated frame the wind direction is $(1, 0)$, so no advective transport occurs across the $\hat{y}$-boundaries and their influence is limited to diffusion, which is small but non-zero for the parameter values considered. Consequently, the use of homogeneous Dirichlet boundary conditions remains appropriate and does not introduce non-physical behaviour near the boundaries.

To model uncertainty in wind direction, we treat φ as a random variable sampled from a circular distribution. There are several circular distributions, and we choose to use the von Mises distribution [12] as it is mathematically simple and easy to

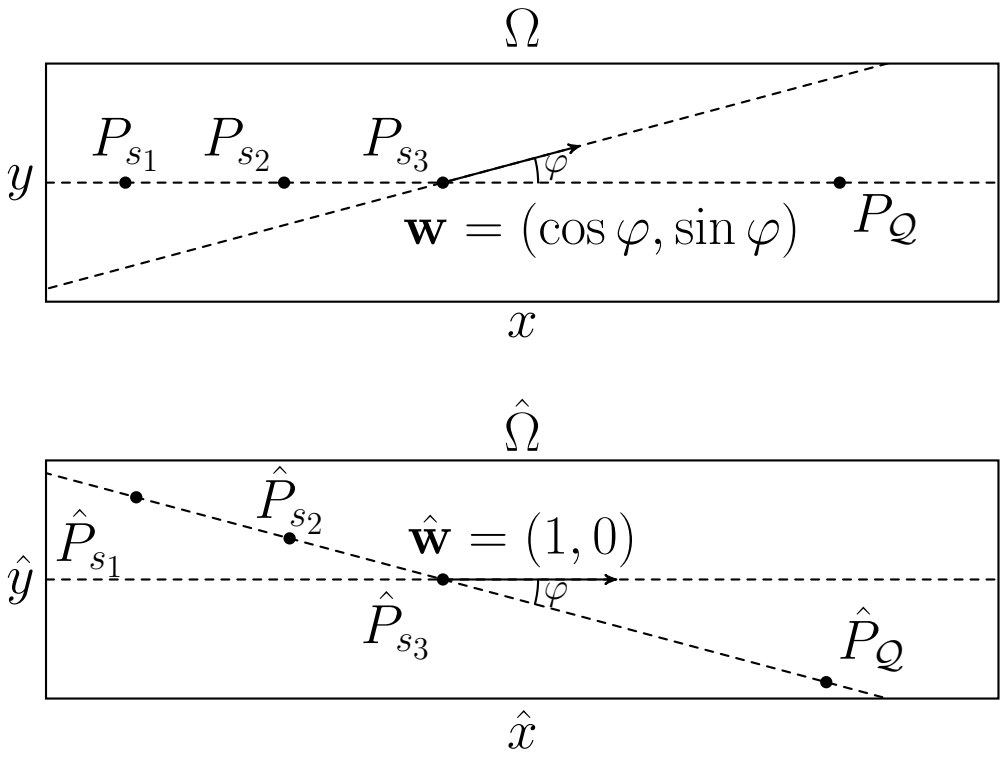


Figure 6.6: Wind-direction rotation: physical frame Ω (top) and rotated frame $\hat{\Omega}$ (bottom) with sources and QoI.

work with. It has probability density function

$$f_{\text{vM}}(\varphi; \mu_{\text{vM}}, \kappa_{\text{vM}}) = \frac{\exp(\kappa_{\text{vM}} \cos(\varphi - \mu_{\text{vM}}))}{2\pi I_0(\kappa_{\text{vM}})},$$

where μ_{vM} is the mean direction, $\kappa_{\text{vM}} \geq 0$ is a concentration parameter, and $I_0(\kappa_{\text{vM}})$ is the modified Bessel function of the first kind of order zero,

$$I_0(\kappa_{\text{vM}}) = \frac{1}{\pi} \int_0^{\pi} \exp(\kappa_{\text{vM}} \cos t)\, \mathrm{d}t.$$

We set $P_{s_3} = (6, 1)$ and $P_{\mathcal{Q}} = (10, 1)$, so $\|P_{\mathcal{Q}} - P_{s_3}\| = 4$. The rotated QoI location is then

$$\hat{P}_{\mathcal{Q}}(\varphi) = P_{s_3} + 4(\cos\varphi,\ -\sin\varphi) = (6 + 4\cos\varphi,\ 1 - 4\sin\varphi).$$

Requiring $\hat{P}_{\mathcal{Q}}(\varphi) \in \hat{\Omega} = [0, 16] \times [0, 2]$ gives the vertical constraint

$$0 \leq 1 - 4\sin\varphi \leq 2 \quad \Longrightarrow \quad -\frac{1}{4} \leq \sin\varphi \leq \frac{1}{4},$$

and the horizontal constraint

$$0 \leq 6 + 4\cos\varphi \leq 16 \implies -\frac{3}{2} \leq \cos\varphi \leq \frac{5}{2},$$

Focusing on the narrower vertical bound,

$$|\varphi| \leq \arcsin(\frac{1}{4}) \approx 0.25268 \text{ rad} = 14.48°.$$

To ensure $\hat{P}_{\mathcal{Q}}(\varphi)$ remains within $\hat{\Omega}$ in at least 99% of realisations while maximising the possible variability, we choose $\mu_{\text{vM}} = 0$ and $\kappa_{\text{vM}} = 110$, which gives a central 99% interval of approximately $\varphi = \pm 14.15°$. As $\hat{P}_{\mathcal{Q}}(\varphi)$ rarely sits on a mesh node, we evaluate $\mathcal{Q}$ using interpolation at $\hat{P}_{\mathcal{Q}}$.

In Figure 6.7 we can see the pdf of the von Mises distribution is contained in a narrow range, with central 95% quantile interval $\varphi = \pm 0.1877$ radians $= \pm 10.74°$. In a real-world setting, this could correspond to something like a narrow valley or a

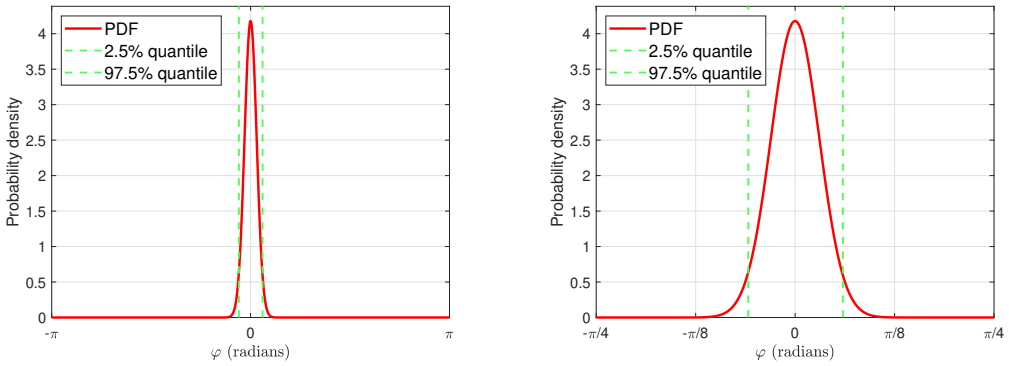


Figure 6.7: The pdf of φ, sampled from a von Mises distribution.

coastal inlet, where the surrounding terrain or shoreline naturally keeps the plume confined even when the wind direction changes.

6.5 Estimation of Sobol indices

We now have an advection-diffusion model with uncertain parameters which are listed in Table 6.2. Figure 6.8 shows two solutions of the PDE with the mean

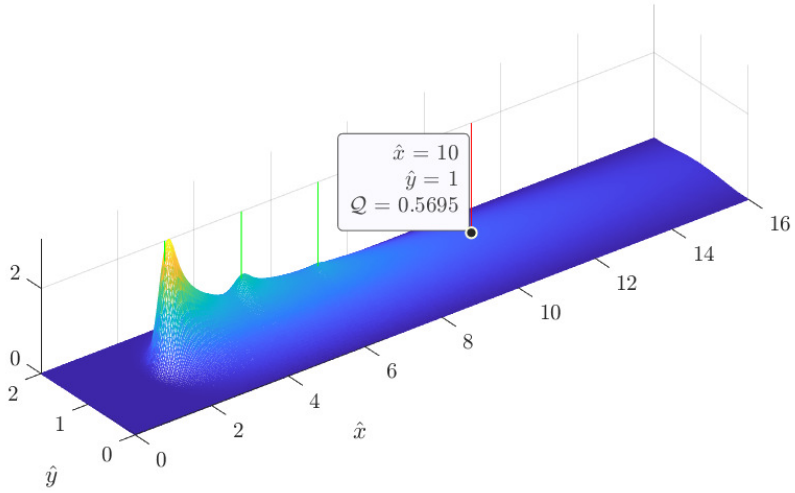
Parameter	Description	Distribution
a	Wind speed	$f_{\text{Weibull}}(x;\, 2.4 \times 10^{-3},\, 1.57)$
D	Diffusion coefficient	$f_{\text{log-normal}}(x;\, -8.90,\, 1.28)$[a]
q_1	Source strength 1	$f_{\text{gamma}}(x;\, 4.73,\, 6.81 \times 10^{-4})$
q_2	Source strength 2	$f_{\text{gamma}}(x;\, 4.73,\, 2.98 \times 10^{-4})$
q_3	Source strength 3	$f_{\text{gamma}}(x;\, 4.73,\, 4.26 \times 10^{-5})$
φ	Wind direction (angle)	$f_{\text{vM}}(\varphi;\, 0,\, 110)$

[a] Truncated below at $x = 5 \times 10^{-5}$.

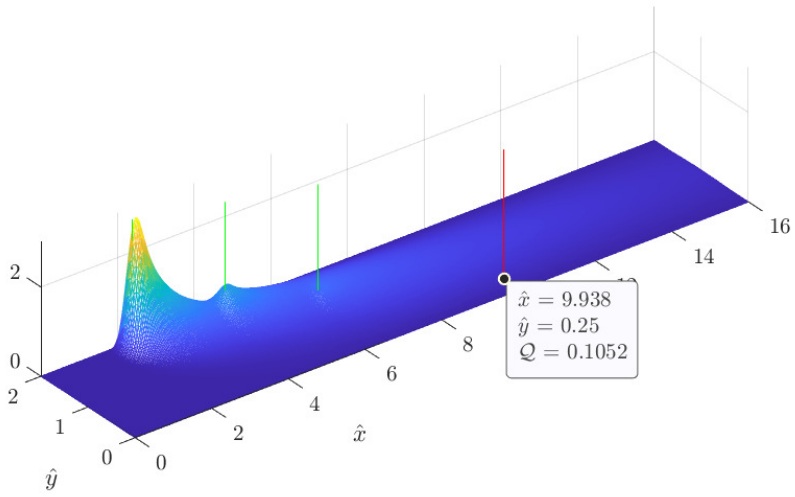
Table 6.2: Distributions used for each uncertain model parameter.

of the sample distributions of the parameters, found using the Streamline Upwind Petrov–Galerkin (SUPG) finite element method, as described in §2.3 with 33345 dof. These solutions have $\varphi = 0$ rad and 0.1877 rad respectively, to demonstrate the effect of the variable wind direction.

In Figure 6.8 we can see the new pollutant dispersal problem with three sources of pollutant release. When $\varphi = 0$, the quantity of interest location $P_{\mathcal{Q}} = (10, 1)$



(a) Plot of the pollutant concentration with $\varphi = 0$ rad.



(b) Plot of the pollutant concentration with $\varphi = 0.1877$ rad.

Figure 6.8: Plot of the pollutant concentration with $a = \mathbb{E}(a)$, $D = \mathbb{E}(D)$, $q_1 = \mathbb{E}(q_1)$, $q_2 = \mathbb{E}(q_2)$, $q_3 = \mathbb{E}(q_3)$, and (a) $\varphi = 0$, and (b) $\varphi = 0.1877$ rad.

lies on a node, however, in Figure 6.8(b), the concentration at the point of interest is found using interpolation. Figure 6.8 therefore illustrates the effect of rotating the coordinate frame on the relative position of the quantity of interest within the domain. In addition, rotating the wind direction alters whether the quantity of interest lies directly downwind of the pollutant sources. As a result, the value of the quantity of interest is larger for $\varphi = 0$ than for $\varphi = 0.1877$, which is consistent with physical intuition, as the concentration is expected to be lower when the point of interest is not directly downstream of the sources.

We now consider the estimation of the Sobol indices for this new problem using the CVB method. We define the principal model as

$$f^{(1)}(\mathbf{X}) = f(a, D, \boldsymbol{q}, \varphi),$$

where $f^{(1)}$ is found by solving the PDE using the SUPG method with 33345 dof, and the output is the QoI introduced in (6.9). We consider six surrogate models $f^{(2)}(\mathbf{X})$ as before: three coarse grid approximations with 8481, 2193, and 585 degrees of freedom, respectively, a linear regression model (LM), a Gaussian process regression model (GPR), and a neural network (NN). As the surrogate model has a physical requirement that the pollution cannot be less than 0, we introduce the constraint that $f^{(2)} \geq 0$ for the data-informed approximations by setting the output to be the greater of $f^{(2)}$ and 0. We train the data-informed approximations using 10^4 Latin-hypercube samples as before, then use the hyperparameters which result in the highest correlation from the Regression Learner app in MATLAB. This results in a GPR model with a Matérn 5/2 kernel covariance function, and a neural network with two hidden layers of ten nodes each. These are the same hyperparameters that produced the highest correlation in §4.6 when the mesh-Péclet number was high. We then take 10^4 independent random test sample inputs and outputs of $f^{(1)}$, and estimate the correlation between $f^{(1)}$ and each $f^{(2)}$.

In Table 6.3, we can see all of the surrogate models are less correlated than for the problem considered in Chapter 4. The coarsest mesh approximation is uncorrelated,

$f^{(2)}$	ρ
8481 dof Coarse Grid	0.7828
2193 dof Coarse Grid	0.3564
585 dof Coarse Grid	-0.0489
LM	0.4940
GPR	0.9321
NN	0.9921

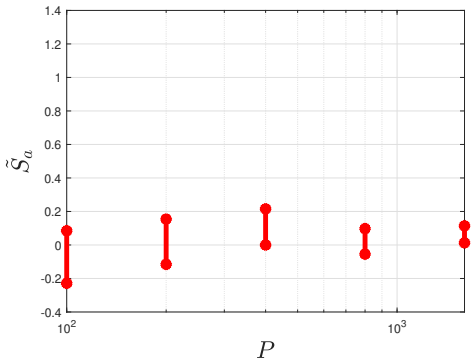
Table 6.3: Correlation values ρ between $f^{(1)}$ and different $f^{(2)}$.

while the finest mesh approximation only has $\rho = 0.7828$. The linear regression model is also only weakly correlated, while the neural network remains more highly correlated than the Gaussian processes regression model. To estimate Sobol indices, we will focus on the coarse grid with 8481 degrees of freedom, and the neural network as surrogate models. The relative cost of evaluating $f^{(2)}$ compared with $f^{(1)}$ is $c = 0.2543$ for the coarse-grid approximation (see §4.6.1), whereas the neural network approximation has a relative cost of $c = 0.005$ (see §4.6.2).

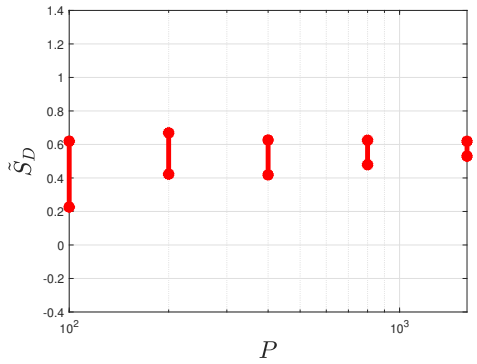
We first examine the Sobol first-order indices. In Figure 6.9 we use the Monte Carlo bootstrapping method presented in Algorithm 2 (see Appendix A) to find confidence intervals, and in Figures 6.10 and 6.11 we find confidence intervals using the method presented in Algorithm 12 (see Appendix A).

In Figures 6.9 – 6.11, the first-order intervals suggest that diffusion is the most influential parameter, with only the confidence intervals for a, D, and φ not covering zero when the budget is $P = 1600$. In Figure 6.10, the confidence intervals are slightly narrower but do not differ noticeably from those high-fidelity model evaluations obtained previously. Figure 6.11 shows the narrowest confidence intervals, even with just $P = 100$, and suggests a non-zero influence from a, q_2, q_3, and φ, although the output remains diffusion dominated. The first-order indices do not sum to one in all cases, indicating that a substantial portion of the variance arises from interactions between inputs. These interactions can be examined using the total-order indices.

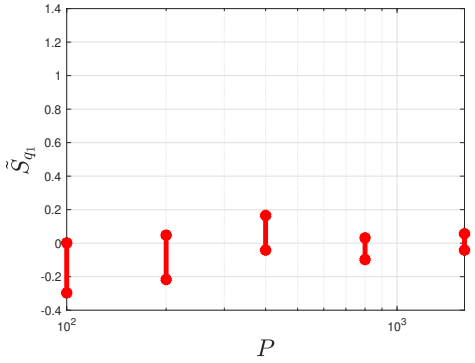
In Figures 6.12 – 6.14, the total-order indices suggest that diffusion remains the most influential parameter, even when accounting for interaction effects. The CVB method with a neural network approximation produces the narrowest confidence



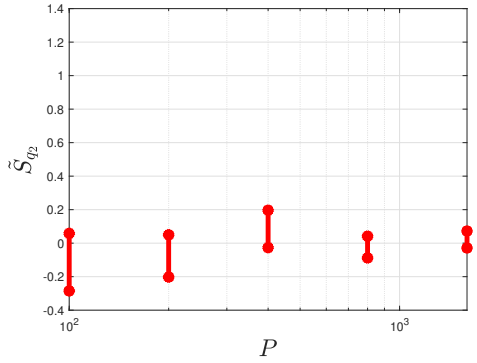
(a) 95% confidence intervals of $\tilde{S}_a$.



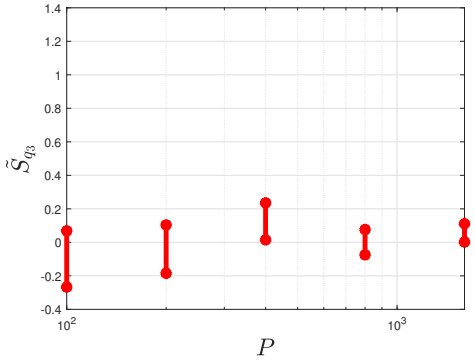
(b) 95% confidence intervals of $\tilde{S}_D$.



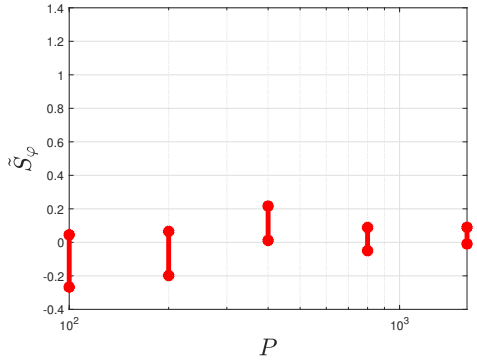
(c) 95% confidence intervals of $\tilde{S}_{q_1}$.



(d) 95% confidence intervals of $\tilde{S}_{q_2}$.

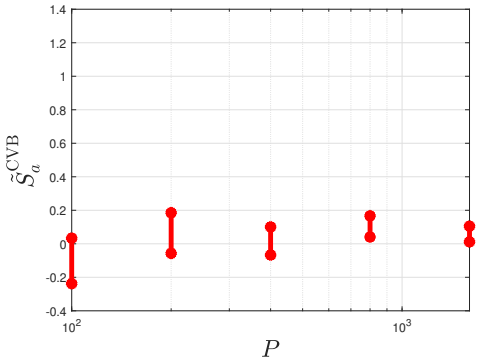


(e) 95% confidence intervals of $\tilde{S}_{q_3}$.

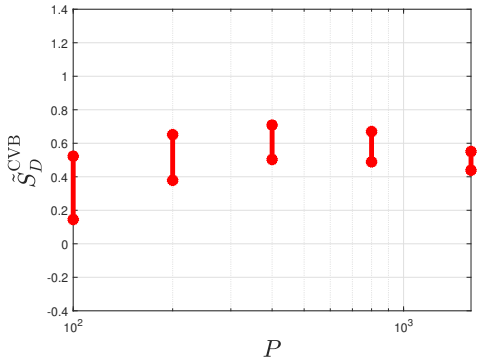


(f) 95% confidence intervals of $\tilde{S}_\varphi$.

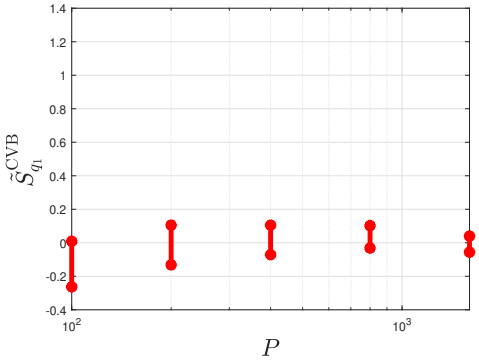
Figure 6.9: Confidence intervals of the first-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using the Monte Carlo method.



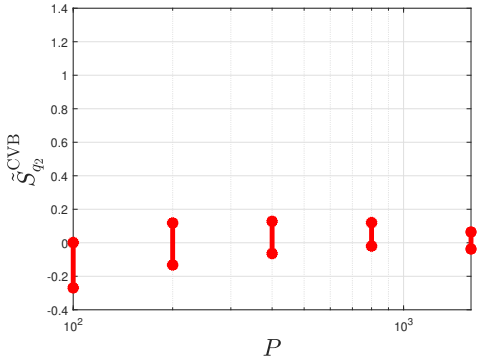
(a) 95% confidence intervals of $\tilde{S}_a^{\mathrm{CVB}}$.



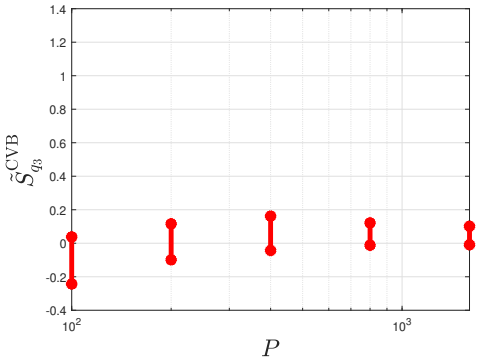
(b) 95% confidence intervals of $\tilde{S}_D^{\mathrm{CVB}}$.



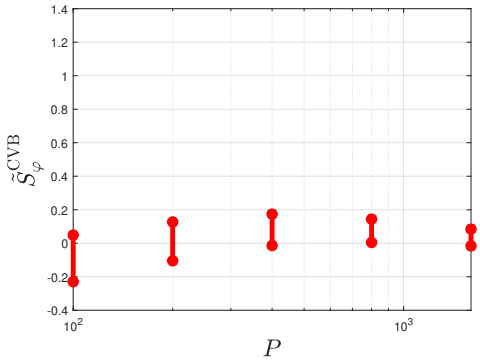
(c) 95% confidence intervals of $\tilde{S}_{q_1}^{\mathrm{CVB}}$.



(d) 95% confidence intervals of $\tilde{S}_{q_2}^{\mathrm{CVB}}$.

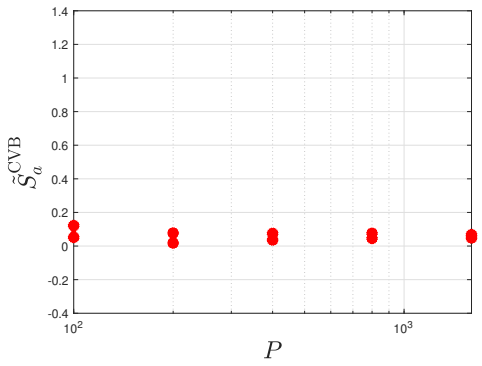


(e) 95% confidence intervals of $\tilde{S}_{q_3}^{\mathrm{CVB}}$.

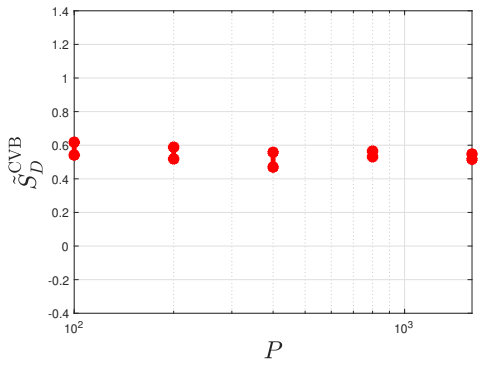


(f) 95% confidence intervals of $\tilde{S}_{\varphi}^{\mathrm{CVB}}$.

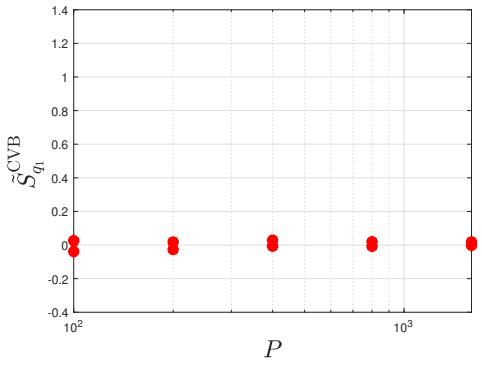
Figure 6.10: Confidence intervals for the first-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using the CVB method with a coarse grid approximation of 8481 degrees of freedom.



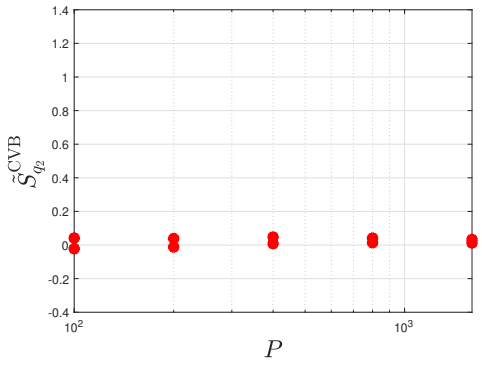
(a) 95% confidence intervals of $\tilde{S}_a^{\mathrm{CVB}}$.



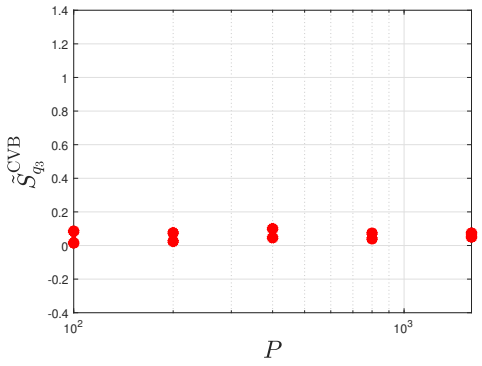
(b) 95% confidence intervals of $\tilde{S}_D^{\mathrm{CVB}}$.



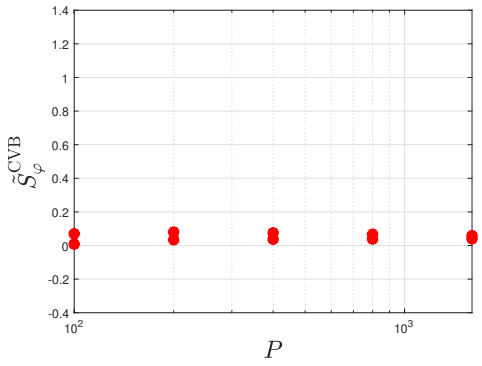
(c) 95% confidence intervals of $\tilde{S}_{q_1}^{\mathrm{CVB}}$.



(d) 95% confidence intervals of $\tilde{S}_{q_2}^{\mathrm{CVB}}$.

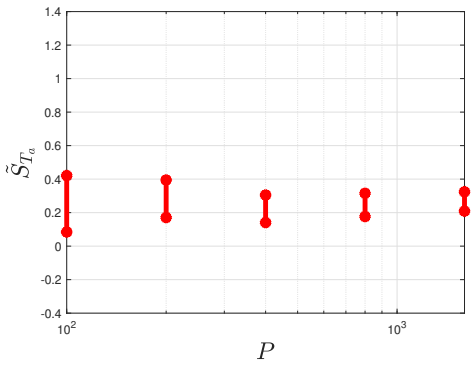


(e) 95% confidence intervals of $\tilde{S}_{q_3}^{\mathrm{CVB}}$.

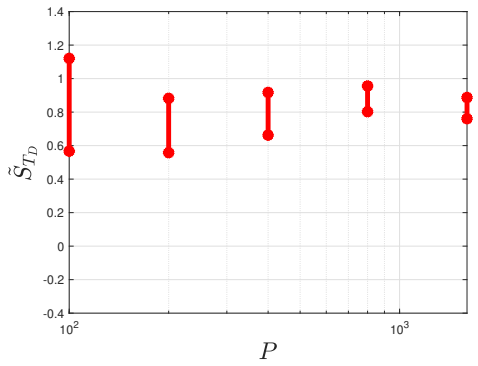


(f) 95% confidence intervals of $\tilde{S}_{\varphi}^{\mathrm{CVB}}$.

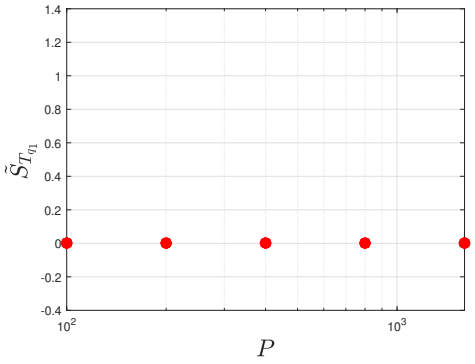
Figure 6.11: Confidence intervals for the first-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using the CVB method with a neural network approximation.



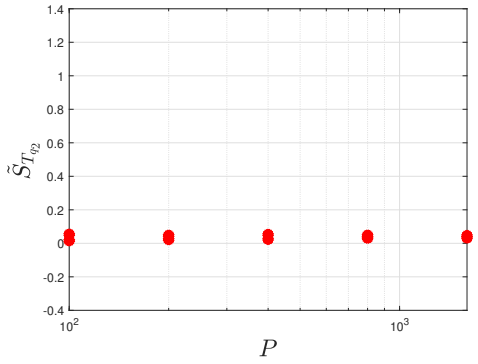
(a) 95% confidence intervals of $\tilde{S}_{T_a}$.



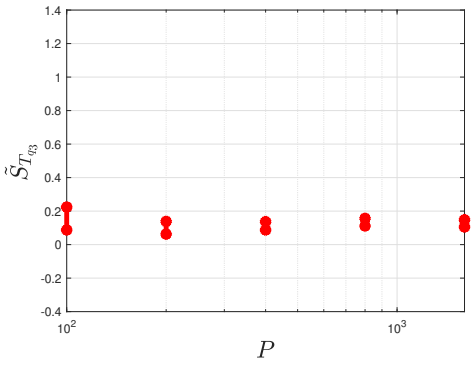
(b) 95% confidence intervals of $\tilde{S}_{T_D}$.



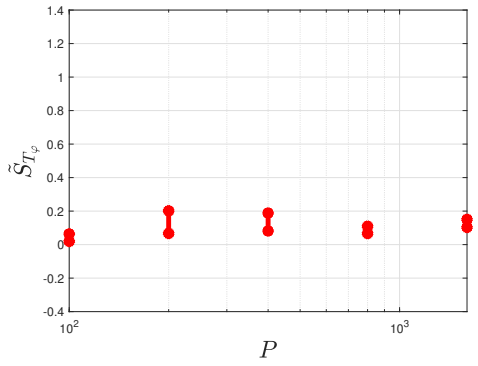
(c) 95% confidence intervals of $\tilde{S}_{T_{q_1}}$.



(d) 95% confidence intervals of $\tilde{S}_{T_{q_2}}$.

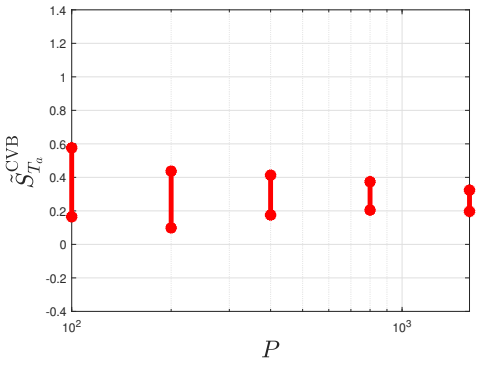


(e) 95% confidence intervals of $\tilde{S}_{T_{q_3}}$.

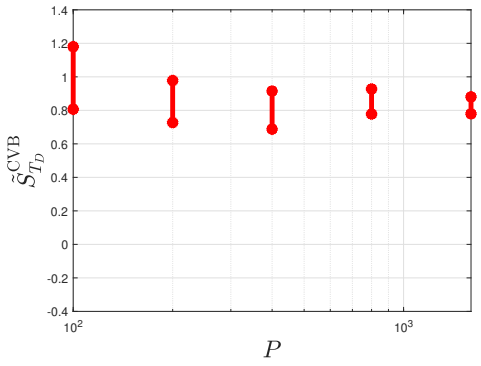


(f) 95% confidence intervals of $\tilde{S}_{T_\varphi}$.

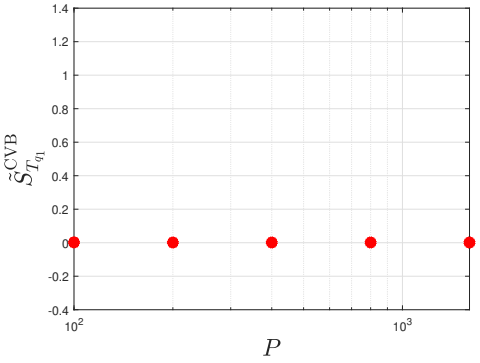
Figure 6.12: Confidence intervals of the total-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using Monte Carlo.



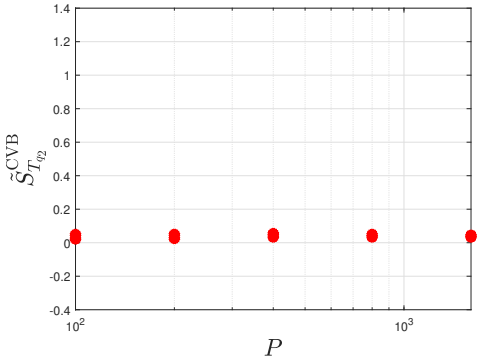
(a) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_a}$.



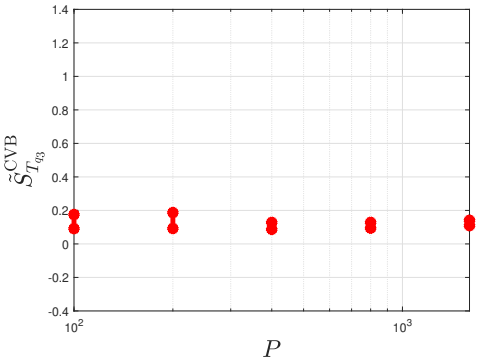
(b) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_D}$.



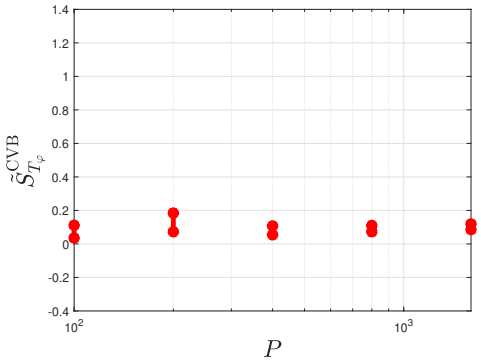
(c) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{Tq_1}$.



(d) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_{q_2}}$.

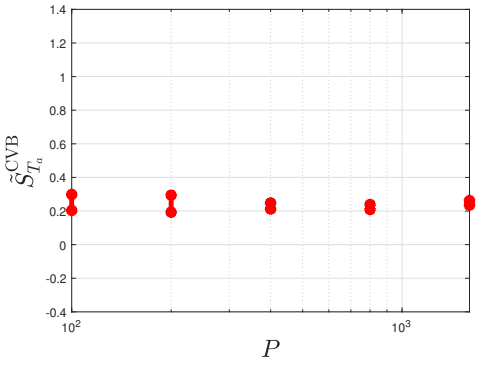


(e) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{Tq_3}$.

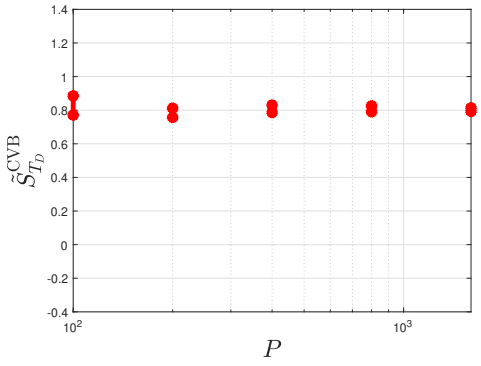


(f) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_\varphi}$.

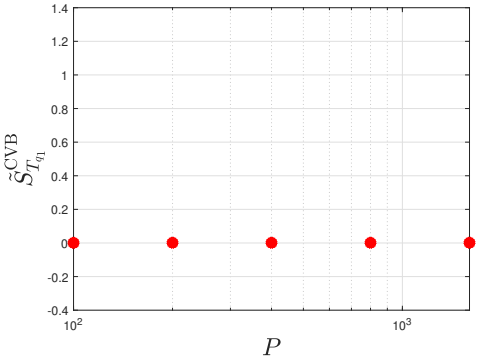
Figure 6.13: Confidence intervals for the total-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using the CVB method with a coarse grid approximation of 8481 degrees of freedom.



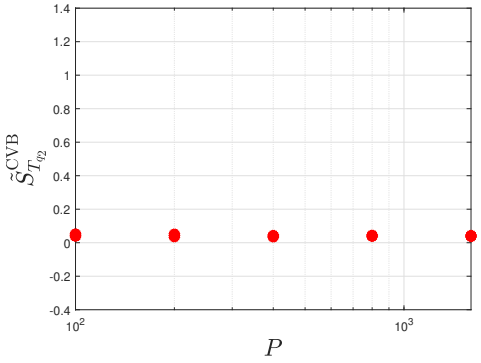
(a) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_a}$.



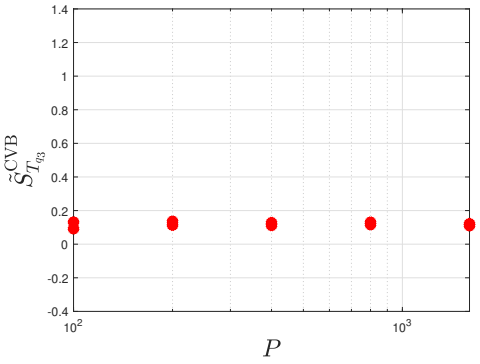
(b) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_D}$.



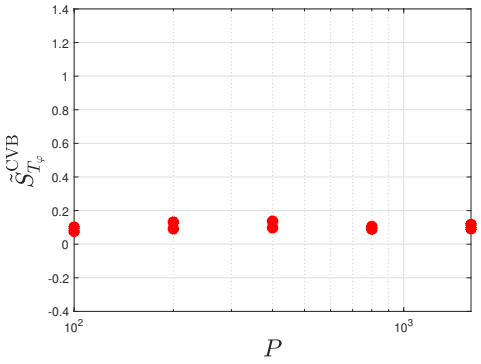
(c) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_{q_1}}$.



(d) 95% confidence intervals of $\tilde{S}T_{q_2}{}^{\mathrm{CVB}}$.



(e) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_{q_3}}$.



(f) 95% confidence intervals of $\tilde{S}^{\mathrm{CVB}}_{T_\varphi}$.

Figure 6.14: Confidence intervals for the total-order Sobol indices of the advection-diffusion equation with six uncertain parameters, calculated using the CVB method with a neural network approximation.

intervals and indicates that q_1 still has little influence under interactions. The total-order indices for a, q_3, and φ are slightly larger than their corresponding main-effect indices, but remain very small. These results indicate that uncertainty in q_1 and q_2 (associated with the release points P_{s_1} and P_{s_2}) has negligible influence on the output, as reflected by their near-zero Sobol indices, whereas q_3, associated with the source located at P_{s_3}, remains influential. This is physically consistent as P_{s_3} lies closest to the quantity of interest $P_{\mathcal{Q}}$, and the rotated-frame rotates about P_{s_3}, so that increasing φ increases the distance in the y-direction between $P_{\mathcal{Q}}$ and the sources at P_{s_1} and P_{s_2}. With the wind direction fixed at $(1, 0)$ in the rotated frame, pollutant released from P_{s_1} and P_{s_2} is advected downstream and is less likely to reach $P_{\mathcal{Q}}$. In addition, any contribution that does reach $P_{\mathcal{Q}}$ has travelled further, giving more opportunity for spreading and dilution.

Overall, the results in this chapter are intended primarily as a proof-of-concept, demonstrating how uncertainty-quantification and variance-reduction techniques behave in advection-dominated dispersion problems. While diffusivity is shown to be the most influential parameter, the primary aim here is methodological. Nevertheless, the approach could be incorporated into a more detailed parametric study to provide information that may be useful for environmental dispersion modelling in practice.

6.6 Conclusions

In this chapter, we introduced a dimensional form of the advection-diffusion model. In Section 6.1, we incorporated physical units into the model, increased the domain size, redefined the quantity of interest, and introduced eddy diffusion. We also modified the source term to represent three distinct points of pollutant release. These changes were made to better reflect realistic conditions while maintaining a simplified model framework.

In Section 6.2, we introduced distributions to represent uncertainty in the wind speed and diffusion in the model. Real-world wind speed data was used to fit a

Weibull distribution. For the diffusion term, we used an expression based on friction velocity, mixing layer height, and the Monin-Obukhov length. Synthetic data was generated using uniform distributions with physically feasible bounds, from which a truncated log-normal distribution was fitted.

In Section 6.3, we introduced uncertainty in the source strength of the pollutant releases. A gamma distribution was chosen, with parameters fitted using real-world estimates of particulate emissions from a zinc smelter, as reported in [60]. In Section 6.4, we introduced variable wind direction by rotating the coordinate system on a fixed mesh to preserve computational efficiency. The wind direction angle, φ, was sampled from a von Mises distribution. A narrow range of values for φ was identified to ensure the point where the quantity of interest was measured remained within the domain, and parameters were selected accordingly.

In Section 6.5, we tested approximations of the new model and found that both the coarse-grid and data-informed approaches have lower correlation with the more realistic model in this chapter compared to the model in Chapter 4, while the neural network approximation remains strongly correlated. We applied the CVB method, introduced in Section 5.5, to compute confidence intervals for the Sobol indices. The neural network resulted in the narrowest confidence intervals due to its low cost and high correlation. Our results showed that D and its interaction effects are the dominant contributors to output variance, with the total-order index of D being close to one, while the remaining indices are small.

Chapter 7

Conclusions and further work

In this thesis we analysed a range of methods for uncertainty quantification and sensitivity analysis of advection–diffusion models. In Chapter 2 we first modelled pollution dispersion in 2D using an advection-diffusion partial differential equation, which we used as the basis for uncertainty quantification methods. We began by applying the Galerkin finite element method, where numerical oscillations were observed in advection-dominated problems. This motivated the use of the streamline upwind Petrov-Galerkin (SUPG) method to stabilise the solution, and we confirmed that SUPG provided more stable solutions in advection-dominated problems. This motivated our choice of SUPG as the discretisation method throughout the remainder of the thesis.

In Chapter 3 we used Monte Carlo to estimate the mean of the Ishigami function, ensuring the theoretical and empirical MSEs were equivalent. We then selected a quantity of interest in the advection–diffusion model from Chapter 2. Treating wind speed and diffusion as random variables, we used Monte Carlo to quantify output uncertainty and verified that estimates of the mean and MSE were equivalent to theory. We derived a method to compute the expected mesh-Péclet number for these random inputs, and introduced two distributions for wind speed and diffusion resulting in high and low Péclet numbers respectively. By introducing uncertainty in the advection and diffusion terms, and choosing appropriate probability distributions,

we demonstrated how Monte Carlo sample size influences the estimate of the QoI.

Confidence intervals were then estimated using bootstrapping as a computationally efficient means of quantifying uncertainty. Our results showed that confidence intervals computed using bootstrapping shrink as the number of evaluations increases, while maintaining appropriate coverage for both the Ishigami function and our advection–diffusion model. This established bootstrapping as a practical alternative to the use of replications to determine the uncertainty in an estimated mean.

In Chapter 4 we considered multi-fidelity variance reduction methods in order to reduce sampling error. Standard control variates were analysed, before examining the CVEM and MFMC methods, which allowed for control variates with unknown means. We then introduced the novel CVB method, which introduces a budget constraint to CVEM. In each case we derived theoretical error bounds and confirmed them with numerical experiments using the Ishigami function with known mean. These methods showed that estimation error can be reduced without increasing computational cost, even when only approximate information about control variates is available. We compared the theoretical MSE of our CVB method to the two-level MFMC method and found that when using low-fidelity models with the same cost and correlation, their theoretical mean-squared errors are equivalent. We also introduced a multi-fidelity bootstrap method, combining CVB with confidence intervals, which consistently produced narrower intervals than Monte Carlo when estimating the mean of the Ishigami function.

We introduced coarse-grid and data-informed low-fidelity models to approximate the output of our advection–diffusion problem from Chapter 3. We found that coarse-grid models remained highly correlated with the advection–diffusion model, and that, after hyperparameter tuning, the neural network produced the most highly correlated low-fidelity models at the lowest computational cost. We obtained confidence intervals for the mean that were consistently narrower than those from standard Monte Carlo, with the neural network approximation resulting in the narrowest intervals.

In Chapter 5 we carried out some sensitivity analysis based on Sobol indices. First

we compared existing high-fidelity estimators by estimating the Sobol indices of the Ishigami function and comparing the MSEs. We then developed a novel CVB based approach for estimating Sobol indices. This method reduced estimator variance and, when combined with bootstrapping, provided confidence intervals of the Sobol indices of the Ishigami function, with correct coverage and reduced width compared to Monte Carlo. Applying this to our advection–diffusion model revealed the difference in Sobol indices when wind and diffusion are modelled using the different input distributions from Chapter 3. When the expected mesh-Péclet number is low, the CVB approach enabled clear parameter ranking at lower computational budgets, while Monte Carlo intervals remained overlapping. When the model had a high expected mesh-Péclet number, the problem was advection dominated, but the CVB method still gave smaller intervals for a lower budget than the Monte Carlo method.

In Chapter 6, we extended the advection-diffusion model to include multiple pollutant sources and uncertain parameters derived from real-world data. Wind speed was represented using a Weibull distribution fit to real world measurements, horizontal eddy diffusion was expressed in terms of turbulence parameters and fit to a lognormal distribution. Source strengths were modelled with gamma distributions based on emission estimates from industrial activity, and wind direction was represented using a von Mises distribution. We then applied our variance reduction and sensitivity analysis techniques to this more realistic setting, testing low-fidelity approximations including coarse grids and neural network surrogates. The neural network approximation gave the narrowest confidence intervals for Sobol indices. The sensitivity measures showed diffusion and its interaction effects were the principal factor in output variance, with advection and its interactions being second most influential, and all other parameters having indices of zero or near zero.

There are a number of methodological and applied directions in which would be of interest in future work. Looking first at methodology, the CVB method could be extended to use multiple control variates. In Chapters 4 and 6, we trained data-informed low-fidelity models using mean-squared error (MSE) as the loss function.

However, since control variates rely on maximising correlation with the high-fidelity model, an interesting direction for future research would be to explore loss functions that directly promote correlation during training. An important problem for future work to examine would be to investigate why coarse mesh surrogate models remain so highly correlated with the fine-mesh principal model, even as the discretisation becomes extremely coarse. A natural extension would be to to design model management strategies that require less prior information, since both CVB and MFMC currently rely on accurate estimates of correlation between models. This problem has been partially addressed in [41], where for MFMC, optimal trade-offs between training and sampling are identified to minimise upper bounds on the mean-squared error under fixed computational budgets. The work in this thesis has focused exclusively on sampling error, while assuming the underlying numerical error is insignificant. An important direction for future research would be to investigate how multi-fidelity methods can balance both sampling error and numerical error, particularly when low-fidelity surrogates are biased.

Future applications of CVB could consider three dimensional and time dependent advection–diffusion models, where computational demands are substantially higher and the potential benefits of variance reduction are much greater. The methodology could also be applied to more complex transport models, such as advection–diffusion–reaction systems or coupled atmospheric chemistry models. Further work could focus on improving the accuracy and correlation of data-informed low-fidelity surrogates, for example by incorporating physical insights or more targeted training strategies. Additionally, alternative low-fidelity approximations, such as those based on Bayesian surrogates, reduced-order models, or physics-informed machine learning, could be explored beyond the coarse-grid and data-informed approaches considered here.

In conclusion, this thesis demonstrates how variance reduction and multi-fidelity approaches can substantially improve on Monte Carlo based sensitivity analysis for PDE models. By combining stabilised discretisations, control variate strategies,

and bootstrapping methods, we have developed algorithms that are theoretically sound and practically effective, providing a foundation for more reliable and efficient sensitivity analysis in complex models of environmental transport and beyond.

Appendix A

Algorithms

Algorithm 1 Monte Carlo estimation of the mean.

Require: Function $f^{(1)}$; sample size k

1: Draw k samples X_i, $i = 1, \ldots, k$, from the associated distributions.

2: Evaluate $f^{(1)}_{X_i}$ for each sample.

3: Compute $\hat{\theta}^{(1)}$ as

$$\hat{\theta}^{(1)} = \frac{1}{k} \sum_{i=1}^{k} f^{(1)}_{X_i}.$$

Ensure: Return $\hat{\theta}^{(1)}$.

Algorithm 2 Bootstrap γ-confidence interval for the mean.

Require: Function $f^{(1)}$; sample size k; number of bootstrap replications R_b; confidence level γ

1: Draw k samples X_i, $i = 1, \ldots, k$ from the associated distribution.
2: Evaluate $f^{(1)}_{X_i}$ for each sample.
3: Compute the sample mean

$$\hat{\theta}^{(1)} = \frac{1}{k} \sum_{i=1}^{k} f^{(1)}_{X_i}.$$

4: **for** $r = 1, \ldots, R_b$ **do**
5: Draw at random a list L of length k (with replacement) from $\{1, 2, \ldots, k\}$.
6: Compute the mean estimate

$$\hat{\theta}^{(1)}_r = \frac{1}{k} \sum_{j \in L} f^{(1)}_{X_j}.$$

7: **end for**
8: Return the $\gamma/2$ and $1 - \gamma/2$ quantiles of $\left\{\hat{\theta}^{(1)}_1, \hat{\theta}^{(1)}_2, \ldots, \hat{\theta}^{(1)}_{R_b}\right\}$.
9: **End**

Algorithm 3 Control variates with known means (CVKM).

Require: Function $f^{(1)}$ and approximation $f^{(2)}$; known $\mathbb{E}(f^{(2)}) = \theta^{(2)}$, $\mathrm{Cov}(f^{(1)}, f^{(2)})$, and $\mathbb{V}(f^{(2)})$; sample size k

1: Draw k samples X_i, $i = 1, \ldots, k$, from the associated distributions.
2: Evaluate $f^{(1)}_{X_i}$ and $f^{(2)}_{X_i}$ for $i = 1, \ldots, k$ using the same samples X_i.
3: Calculate the mean estimates $\hat{\theta}^{(n)}$, $n = 1, 2$:

$$\hat{\theta}^{(n)} = \frac{1}{k} \sum_{i=1}^{k} f^{(n)}_{X_i}.$$

4: Evaluate α_* as

$$\alpha_* = \frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})}.$$

5: Construct the control variate estimator $\hat{\theta}^{(1)}_{\mathrm{CVKM}}$ as

$$\hat{\theta}^{(1)}_{\mathrm{CVKM}} = \hat{\theta}^{(1)} - \alpha_*(\hat{\theta}^{(2)} - \theta^{(2)}).$$

Ensure: Return $\hat{\theta}^{(1)}_{\mathrm{CVKM}}$.

Algorithm 4 Control variates with estimated means (CVEM).

Require: Function $f^{(1)}$ and approximation $f^{(2)}$; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation coefficient ρ between $f^{(1)}$ and $f^{(2)}$; sample size k

1: Draw k samples X_i, $i = 1, \ldots, k$, from the associated distributions.
2: Evaluate $f^{(1)}_{X_i}$ and $f^{(2)}_{X_i}$ for $i = 1, \ldots, k$ using the same samples X_i.
3: Compute the mean estimates $\hat{\theta}^{(n)}$, $n = 1, 2$, using

$$\hat{\theta}^{(n)} = \frac{1}{k} \sum_{i=1}^{k} f^{(n)}(X_i).$$

4: Compute k' as

$$k' = k \times \max\{0, \beta - 1\}, \qquad \beta = \left[\left(\frac{\rho^2}{1 - \rho^2} \right) \left(\frac{1}{c} \right) \right]^{\frac{1}{2}}.$$

5: Draw k' samples Y_i, $i = 1, \ldots, k'$, from the associated distribution.
6: Compute the mean estimate $\tilde{\theta}^{(2)}$ as

$$\tilde{\theta}^{(2)} = \frac{1}{k'} \sum_{i=1}^{k'} f^{(2)}_{Y_i}.$$

7: Evaluate α as

$$\alpha = \left(\frac{\mathrm{Cov}(f^{(1)}, f^{(2)})}{\mathbb{V}(f^{(2)})} \right) \left(\frac{1}{1 + k/k'} \right).$$

8: Construct the estimator $\hat{\theta}^{(1)}_{\mathrm{CVEM}}$ as

$$\hat{\theta}^{(1)}_{\mathrm{CVEM}} = \hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \tilde{\theta}^{(2)}).$$

Ensure: Return $\hat{\theta}^{(1)}_{\mathrm{CVEM}}$.

Algorithm 5 Control variates with estimated means and budget control (CVB).

Require: Function $f^{(1)}$ and approximation $f^{(2)}$; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation coefficient ρ between $f^{(1)}$ and $f^{(2)}$; total budget P

1: Compute

$$\phi = \max\{0, \beta - 1\}, \qquad \beta = \left[\left(\frac{\rho^2}{1-\rho^2}\right)\left(\frac{1}{c}\right)\right]^{\frac{1}{2}}.$$

2: **if** $\phi \leq 0$ **then**
3: **End**
4: **end if**
5: Evaluate k and k' as

$$k = \frac{P}{1 + c(1+\phi)} \qquad \text{and} \qquad k' = k\,\phi.$$

6: Draw k samples X_i, $i = 1, \ldots, k$, from the associated distributions and evaluate $f^{(1)}_{X_i}$ and $f^{(2)}_{X_i}$ using the same X_i.
7: Calculate $\hat{\theta}^{(n)}$, $n = 1, 2$, using

$$\hat{\theta}^{(n)} = \frac{1}{k}\sum_{i=1}^{k} f^{(n)}(X_i).$$

8: Draw k' samples Y_i, $i = 1, \ldots, k'$, from the associated distributions.
9: Estimate the mean

$$\tilde{\theta}^{(2)} = \frac{1}{k'}\sum_{i=1}^{k'} f^{(2)}_{Y_i}.$$

10: Evaluate α as

$$\alpha = \left(\frac{\mathrm{Cov}(f^{(1)}_{X_i}, f^{(2)}_{X_i})}{\mathbb{V}(f^{(2)}_{X_i})}\right)\left(\frac{\phi}{\phi+1}\right).$$

11: Construct the estimator $\hat{\theta}^{(1)}_{\mathrm{CVB}}$ as

$$\hat{\theta}^{(1)}_{\mathrm{CVB}} = \hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \tilde{\theta}^{(2)}).$$

Ensure: Return $\hat{\theta}^{(1)}_{\mathrm{CVB}}$.

Algorithm 6 Two-level multi-fidelity Monte Carlo (MFMC) for mean estimation.

Require: Function $f^{(1)}$ and approximation $f^{(2)}$; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation coefficient ρ between $f^{(1)}$ and $f^{(2)}$; total budget P

1: Evaluate
$$r_2 = \sqrt{\frac{\rho^2}{1-\rho^2}\,\frac{1}{c}}.$$

2: Evaluate m_1 and m_2 as
$$m_1 = \frac{P}{1+cr_2} \qquad \text{and} \qquad m_2 = r_2\, m_1.$$

3: Draw m_2 samples X_i from the associated distributions and evaluate $f^{(1)}_{X_i}$ for $i = 1, \ldots, m_1$ and $f^{(2)}_{X_j}$ for $j = 1, \ldots, m_2$.

4: Compute $\hat{\theta}^{(1)}_{m_1}$, $\hat{\theta}^{(2)}_{m_1}$, and $\hat{\theta}^{(2)}_{m_2}$ as
$$\hat{\theta}^{(1)}_{m_1} = \frac{1}{m_1}\sum_{i=1}^{m_1} f^{(1)}(X_i), \qquad \hat{\theta}^{(2)}_{m_1} = \frac{1}{m_1}\sum_{i=1}^{m_1} f^{(2)}(X_i), \qquad \hat{\theta}^{(2)}_{m_2} = \frac{1}{m_2}\sum_{i=1}^{m_2} f^{(2)}(X_i).$$

5: Evaluate
$$\alpha_2 = \frac{\mathrm{Cov}(f^{(1)}_{X_i}, f^{(2)}_{X_i})}{\mathbb{V}(f^{(2)}_{X_i})}.$$

6: Construct the estimator $\hat{\theta}^{(1)}_{\mathrm{MFMC}}$ as
$$\hat{\theta}^{(1)}_{\mathrm{MFMC}} = \hat{\theta}^{(1)}_{m_1} + \alpha_2(\hat{\theta}^{(2)}_{m_2} - \hat{\theta}^{(2)}_{m_1}).$$

Ensure: Return $\hat{\theta}^{(1)}_{\mathrm{MFMC}}$.

Algorithm 7 γ-confidence intervals for control variates with estimated means and budget control (CVB).

Require: Function $f^{(1)}$ and approximation $f^{(2)}$; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation coefficient ρ between $f^{(1)}$ and $f^{(2)}$; total budget P; number of bootstrap replications R_b; confidence level γ

1: Compute

$$\phi = \max\{0, \beta - 1\}, \qquad \beta = \left[\left(\frac{\rho^2}{1-\rho^2}\right)\left(\frac{1}{c}\right)\right]^{\frac{1}{2}}.$$

2: **if** $\phi \leq 0$ **then**

3: **End**

4: **end if**

5: Evaluate k and k' as

$$k = \frac{P}{1 + c(1+\phi)}, \qquad k' = k\,\phi.$$

6: Draw samples X_i, $i = 1, \ldots, k$, and Y_j, $j = 1, \ldots, k'$, from the associated distributions and evaluate $f^{(1)}_{X_i}$, $f^{(2)}_{X_i}$, and $f^{(2)}_{Y_j}$.

7: Calculate $\hat{\theta}^{(j)}$, $j = 1, 2$, and $\tilde{\theta}^{(2)}$ using

$$\hat{\theta}^{(j)} = \frac{1}{k}\sum_{i=1}^{k} f^{(j)}(X_i), \qquad \tilde{\theta}^{(2)} = \frac{1}{k'}\sum_{i=1}^{k'} f^{(2)}_{Y_i}.$$

8: Calculate α as

$$\alpha = \left(\frac{\mathrm{Cov}(f^{(1)}_{X_i}, f^{(2)}_{X_i})}{\mathbb{V}(f^{(2)}_{X_i})}\right)\left(\frac{\phi}{\phi+1}\right).$$

9: Construct $\hat{\theta}^{(1)}_{\mathrm{CVB}}$ as

$$\hat{\theta}^{(1)}_{\mathrm{CVB}} = \hat{\theta}^{(1)} - \alpha(\hat{\theta}^{(2)} - \tilde{\theta}^{(2)}).$$

10: **for** $r = 1, \ldots, R_b$ **do**

11: Draw at random a list L_X of length k (with replacement) from $\{1, 2, \ldots, k\}$.

12: Compute the means

$$\overline{\theta}_r^{(1)} = \frac{1}{k} \sum_{j \in L_X} f_{X_j}^{(1)}, \qquad \overline{\theta}_r^{(2)} = \frac{1}{k} \sum_{j \in L_X} f_{X_j}^{(2)}.$$

13: Draw at random a list L_Y of length k' (with replacement) from $\{1, 2, \ldots, k'\}$.

14: Compute the mean

$$\overline{\overline{\theta}}_r^{(2)} = \frac{1}{k'} \sum_{j \in L_Y} f_{Y_j}^{(2)}.$$

15: Construct the control variate

$$\overline{\theta}_{\mathrm{CVB}_r}^{(1)} = \overline{\theta}_r^{(1)} + \alpha(\overline{\theta}_r^{(2)} - \overline{\overline{\theta}}_r^{(2)}).$$

16: **end for**

Ensure: Return the $\gamma/2$ and $1 - \gamma/2$ quantiles of $\left\{\overline{\theta}_{\mathrm{CVB}_1}^{(1)}, \overline{\theta}_{\mathrm{CVB}_2}^{(1)}, \ldots, \overline{\theta}_{\mathrm{CVB}_{R_b}}^{(1)}\right\}$.

Algorithm 8 Saltelli method for estimating first-order and total-order Sobol indices

Require: Function $f^{(1)}$. Inputs $(X_1, \ldots, X_m)$ with known distributions; sample size k.

1: Draw two independent $k \times m$ design matrices

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}.$$

2: Evaluate the model on the rows of $\mathbf{A}$ and $\mathbf{B}$:

$$\tilde{\mathbf{Y}}^{(\mathbf{A})} = \begin{bmatrix} f^{(1)}(\mathbf{A}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{A}_{k,:}) \end{bmatrix}, \qquad \tilde{\mathbf{Y}}^{(\mathbf{B})} = \begin{bmatrix} f^{(1)}(\mathbf{B}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{B}_{k,:}) \end{bmatrix}.$$

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

4: Form $\mathbf{C}^{(i)}$ by copying $\mathbf{B}$ and replacing its i-th column with the i-th column of $\mathbf{A}$.

5: Evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})} = \begin{bmatrix} f^{(1)}((\mathbf{C}^{(i)})_{1,:}) \\ \vdots \\ f^{(1)}((\mathbf{C}^{(i)})_{k,:}) \end{bmatrix}.$$

6: Compute the first-order index estimate

$$\tilde{S}_i^{\mathrm{S}} = \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})}\right)\left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{C}^{(i)})}\right)}{\frac{1}{k}\sum_{j=1}^{k} (\tilde{Y}_j^{(\mathbf{A})})^2 - \left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})}\right)^2}.$$

7: Compute the total-order index estimate

$$\tilde{S}_{T_i}^{\mathrm{S}} = 1 - \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})}\right)\left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{C}^{(i)})}\right)}{\frac{1}{k}\sum_{j=1}^{k} (\tilde{Y}_j^{(\mathbf{B})})^2 - \left(\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})}\right)^2}.$$

8: **end for**

Ensure: Return $\{\tilde{S}_i^{\mathrm{S}}, \tilde{S}_{T_i}^{\mathrm{S}}\}_{i=1}^{m}$.

Algorithm 9 Sobol–Janon (SJ) first-order and Janon–Monod (JM) total-order estimators

Require: Function $f^{(1)}$; inputs $(X_1, \dots, X_m)$ with known distributions; sample size k

1: Draw two independent $k \times m$ design matrices

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \dots & X_{k,m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \dots & X_{k,m} \end{bmatrix}.$$

2: Evaluate the model on the rows of $\mathbf{A}$ and $\mathbf{B}$:

$$\tilde{\mathbf{Y}}^{(\mathbf{A})} = \begin{bmatrix} f^{(1)}(\mathbf{A}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{A}_{k,:}) \end{bmatrix}, \qquad \tilde{\mathbf{Y}}^{(\mathbf{B})} = \begin{bmatrix} f^{(1)}(\mathbf{B}_{1,:}) \\ \vdots \\ f^{(1)}(\mathbf{B}_{k,:}) \end{bmatrix}.$$

3: **for** $i = 1, \dots, m$ **do**

4: Form $\mathbf{C}^{(i)}$ by copying $\mathbf{B}$ and replacing its i-th column with the i-th column of $\mathbf{A}$.

5: Evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})} = \begin{bmatrix} f^{(1)}((\mathbf{C}^{(i)})_{1,:}) \\ \vdots \\ f^{(1)}((\mathbf{C}^{(i)})_{k,:}) \end{bmatrix}.$$

6: Compute the SJ first-order index estimate

$$\tilde{S}_i^{\mathrm{SJ}} = \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{A})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k} \frac{\tilde{Y}_j^{(\mathbf{A})} + \tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2} \right)^2}{\frac{1}{k}\sum_{j=1}^{k} \frac{(\tilde{Y}_j^{(\mathbf{A})})^2 + (\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^{k} \frac{\tilde{Y}_j^{(\mathbf{A})} + \tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2} \right)^2}.$$

7: Compute the JM total-order index estimate

$$\tilde{S}_{T_i}^{\mathrm{JM}} = 1 - \frac{\frac{1}{k}\sum_{j=1}^{k} \tilde{Y}_j^{(\mathbf{B})} \tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^{k} \frac{\tilde{Y}_j^{(\mathbf{B})} + \tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2} \right)^2}{\frac{1}{k}\sum_{j=1}^{k} \frac{(\tilde{Y}_j^{(\mathbf{B})})^2 + (\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^{k} \frac{\tilde{Y}_j^{(\mathbf{B})} + \tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2} \right)^2}.$$

8: **end for**

Ensure: Return $\{\tilde{S}_i^{\mathrm{SJ}}, \tilde{S}_{T_i}^{\mathrm{JM}}\}_{i=1}^{m}$.

Algorithm 10 Estimating Sobol first-order and total-order indices using CVB

Require: Budget P; function $f^{(1)}$ and approximation $f^{(2)}$; inputs $(X_1, \ldots, X_m)$ with known distributions; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation ρ between $f^{(1)}$ and $f^{(2)}$

1: Compute

$$\beta = \left[\left(\frac{\rho^2}{1-\rho^2}\right)\left(\frac{1}{c}\right)\right]^{1/2}, \qquad \phi = \max\{0, \beta - 1\},$$
$$k = \frac{P}{1 + c(1+\phi)}, \qquad k' = k\,\phi.$$

2: Draw two independent $k \times m$ design matrices

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}.$$

3: For $j = 1, \ldots, k$, evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{A})} = (f^{(1)}(\mathbf{A}_{1,:}), \ldots, f^{(1)}(\mathbf{A}_{k,:}))^\top, \quad \tilde{\mathbf{Y}}^{(\mathbf{B})} = (f^{(1)}(\mathbf{B}_{1,:}), \ldots, f^{(1)}(\mathbf{B}_{k,:}))^\top,$$

and, using the same $\mathbf{A}, \mathbf{B}$,

$$\tilde{\mathbf{Y}}^{(\mathbf{A}_2)} = (f^{(2)}(\mathbf{A}_{1,:}), \ldots, f^{(2)}(\mathbf{A}_{k,:}))^\top, \quad \tilde{\mathbf{Y}}^{(\mathbf{B}_2)} = (f^{(2)}(\mathbf{B}_{1,:}), \ldots, f^{(2)}(\mathbf{B}_{k,:}))^\top.$$

4: Compute

$$\alpha = \left(\frac{\mathrm{Cov}(\tilde{\mathbf{Y}}^{(\mathbf{A})}, \tilde{\mathbf{Y}}^{(\mathbf{A}_2)})}{\mathrm{Var}(\tilde{\mathbf{Y}}^{(\mathbf{A}_2)})}\right)\left(\frac{\phi}{\phi+1}\right).$$

5: Draw two independent $k' \times m$ design matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$

$$\overline{\mathbf{A}} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k',1} & \ldots & X_{k',m} \end{bmatrix}, \qquad \overline{\mathbf{B}} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k',1} & \ldots & X_{k',m} \end{bmatrix}.$$

6: **for** $i = 1, \ldots, m$ **do**

7: Form $\mathbf{C}_i$ by copying $\mathbf{B}$ and replacing its i-th column with the i-th column of $\mathbf{A}$.

8: Evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})} = (f^{(1)}((\mathbf{C}_i)_{1,:}), \ldots, f^{(1)}((\mathbf{C}_i)_{k,:}))^\top, \quad \tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})} = (f^{(2)}((\mathbf{C}_i)_{1,:}), \ldots, f^{(2)}((\mathbf{C}_i)_{k,:}))^\top.$$

9: Compute the first-order estimates

$$\tilde{S}_i = \frac{\frac{1}{k}\sum_{j=1}^k \tilde{Y}_j^{(\mathbf{A})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j=1}^k \frac{(\tilde{Y}_j^{(\mathbf{A})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2},$$

$$\tilde{S}_i^{(2)} = \frac{\frac{1}{k}\sum_{j=1}^k \tilde{Y}_j^{(\mathbf{A}_2)}\tilde{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{A}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j=1}^k \frac{(\tilde{Y}_j^{(\mathbf{A}_2)})^2+(\tilde{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{A}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}.$$

10: Compute the total-order estimates

$$\tilde{S}_{T_i} = 1 - \frac{\frac{1}{k}\sum_{j=1}^k \tilde{Y}_j^{(\mathbf{B})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j=1}^k \frac{(\tilde{Y}_j^{(\mathbf{B})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2},$$

$$\tilde{S}_{T_i}^{(2)} = 1 - \frac{\frac{1}{k}\sum_{j=1}^k \tilde{Y}_j^{(\mathbf{B}_2)}\tilde{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{B}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j=1}^k \frac{(\tilde{Y}_j^{(\mathbf{B}_2)})^2+(\tilde{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j=1}^k \frac{\tilde{Y}_j^{(\mathbf{B}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}.$$

11: Form $\overline{\mathbf{C}}_i$ by copying $\overline{\mathbf{B}}$ and replacing its i-th column with the i-th column of $\overline{\mathbf{A}}$.

12: Evaluate

$$\begin{aligned}\overline{\mathbf{Y}}^{(\mathbf{A}_2)} &= (f^{(2)}(\overline{\mathbf{A}}_{1,:}), \ldots, f^{(2)}(\overline{\mathbf{A}}_{k',:}))^\top,\\ \overline{\mathbf{Y}}^{(\mathbf{B}_2)} &= (f^{(2)}(\overline{\mathbf{B}}_{1,:}), \ldots, f^{(2)}(\overline{\mathbf{B}}_{k',:}))^\top,\\ \overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})} &= (f^{(2)}((\overline{\mathbf{C}}_i)_{1,:}), \ldots, f^{(2)}((\overline{\mathbf{C}}_i)_{k',:}))^\top.\end{aligned}$$

13: Compute the low-fidelity estimates

$$\overline{S}_i^{(2)} = \frac{\frac{1}{k'}\sum_{j=1}^{k'} \overline{Y}_j^{(\mathbf{A}_2)}\overline{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k'}\sum_{j=1}^{k'} \frac{\overline{Y}_j^{(\mathbf{A}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k'}\sum_{j=1}^{k'} \frac{(\overline{Y}_j^{(\mathbf{A}_2)})^2+(\overline{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k'}\sum_{j=1}^{k'} \frac{\overline{Y}_j^{(\mathbf{A}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2},$$

$$\overline{S}_{T_i}^{(2)} = 1 - \frac{\frac{1}{k'}\sum_{j=1}^{k'} \overline{Y}_j^{(\mathbf{B}_2)}\overline{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k'}\sum_{j=1}^{k'} \frac{\overline{Y}_j^{(\mathbf{B}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k'}\sum_{j=1}^{k'} \frac{(\overline{Y}_j^{(\mathbf{B}_2)})^2+(\overline{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k'}\sum_{j=1}^{k'} \frac{\overline{Y}_j^{(\mathbf{B}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}.$$

14: Form the CVB estimators

$$\tilde{S}_i^{\mathrm{CVB}} = \tilde{S}_i - \alpha(\tilde{S}_i^{(2)} - \overline{S}_i^{(2)}), \qquad \tilde{S}_{T_i}^{\mathrm{CVB}} = \tilde{S}_{T_i} - \alpha(\tilde{S}_{T_i}^{(2)} - \overline{S}_{T_i}^{(2)}).$$

15: **end for**

Ensure: Return $\{\, \tilde{S}_i^{\mathrm{CVB}},\ \tilde{S}_{T_i}^{\mathrm{CVB}} \,\}_{i=1}^m$.

16: **End**

Algorithm 11 Bootstrap confidence intervals for Sobol first-order and total-order indices

Require: Budget P; function $f^{(1)}$; inputs $(X_1, \ldots, X_m)$ with known distributions; bootstrap replicates R_b

Set $k = P$ and draw two independent $k \times m$ design matrices

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \ldots & X_{k,m} \end{bmatrix}.$$

For $j = 1, \ldots, k$, evaluate

$$\mathbf{Y}^{(\mathbf{A})} = (f^{(1)}(\mathbf{A}_{1,:}), \ldots, f^{(1)}(\mathbf{A}_{k,:}))^\top, \qquad \mathbf{Y}^{(\mathbf{B})} = (f^{(1)}(\mathbf{B}_{1,:}), \ldots, f^{(1)}(\mathbf{B}_{k,:}))^\top.$$

for $i = 1, \ldots, m$ **do**

Form the hybrid matrix $\mathbf{C}_i$ by copying $\mathbf{B}$ and replacing its i-th column with the i-th column of $\mathbf{A}$.

Evaluate

$$\mathbf{Y}^{(\mathbf{C}^{(i)})} = (f^{(1)}((\mathbf{C}_i)_{1,:}), \ldots, f^{(1)}((\mathbf{C}_i)_{k,:}))^\top.$$

for $r = 1, \ldots, R_b$ **do**

Draw at random a list L of length k (with replacement) from $\{1, 2, \ldots, k\}$.

Compute the first-order Sobol replicate

$$S_i^{(r)} = \frac{\frac{1}{k}\sum_{j\in L} Y_j^{(\mathbf{A})} Y_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j\in L} \frac{Y_j^{(\mathbf{A})} + Y_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L} \frac{\left(Y_j^{(\mathbf{A})}\right)^2 + \left(Y_j^{(\mathbf{C}^{(i)})}\right)^2}{2} - \left(\frac{1}{k}\sum_{j\in L} \frac{Y_j^{(\mathbf{A})} + Y_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}.$$

Compute the total-order Sobol replicate

$$S_{T_i}^{(r)} = 1 - \frac{\frac{1}{k}\sum_{j\in L} Y_j^{(\mathbf{B})} Y_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j\in L} \frac{Y_j^{(\mathbf{B})} + Y_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L} \frac{\left(Y_j^{(\mathbf{B})}\right)^2 + \left(Y_j^{(\mathbf{C}^{(i)})}\right)^2}{2} - \left(\frac{1}{k}\sum_{j\in L} \frac{Y_j^{(\mathbf{B})} + Y_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}.$$

end for

Let $q_{S_i}^{\mathrm{lo}}$ and $q_{S_i}^{\mathrm{hi}}$ be the $\gamma/2$ and $1-\gamma/2$ quantiles of $\{S_i^{(r)}\}_{r=1}^{R_b}$.

Let $q_{S_{T_i}}^{\mathrm{lo}}$ and $q_{S_{T_i}}^{\mathrm{hi}}$ be the $\gamma/2$ and $1-\gamma/2$ quantiles of $\{S_{T_i}^{(r)}\}_{r=1}^{R_b}$.

Ensure: Return the level-γ confidence intervals

$$[\, q_{S_i}^{\mathrm{lo}},\ q_{S_i}^{\mathrm{hi}}\,] \quad \text{and} \quad [\, q_{S_{T_i}}^{\mathrm{lo}},\ q_{S_{T_i}}^{\mathrm{hi}}\,].$$

end for

Algorithm 12 Estimating Sobol first-order and total-order indices using CVB with bootstrapping

Require: Budget P; function $f^{(1)}$ and approximation $f^{(2)}$; inputs $(X_1, \dots, X_m)$ with known distributions; relative cost c of $f^{(2)}$ to $f^{(1)}$; correlation ρ between $f^{(1)}$ and $f^{(2)}$; bootstrap replicates R_b

1: Compute

$$\beta = \left[\left(\frac{\rho^2}{1-\rho^2}\right)\left(\frac{1}{c}\right)\right]^{1/2}, \qquad \phi = \max\{0, \beta - 1\},$$
$$k = \frac{P}{1 + c(1+\phi)}, \qquad k' = k\,\phi.$$

2: Draw two independent $k \times m$ design matrices

$$\mathbf{A} = \begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \dots & X_{k,m} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & & \vdots \\ X_{k,1} & \dots & X_{k,m} \end{bmatrix}.$$

3: For $j = 1, \dots, k$, evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{A})} = (f^{(1)}(\mathbf{A}_{1,:}), \dots, f^{(1)}(\mathbf{A}_{k,:}))^\top, \quad \tilde{\mathbf{Y}}^{(\mathbf{B})} = (f^{(1)}(\mathbf{B}_{1,:}), \dots, f^{(1)}(\mathbf{B}_{k,:}))^\top,$$

and (using the same $\mathbf{A}, \mathbf{B}$)

$$\tilde{\mathbf{Y}}^{(\mathbf{A}_2)} = (f^{(2)}(\mathbf{A}_{1,:}), \dots, f^{(2)}(\mathbf{A}_{k,:}))^\top, \quad \tilde{\mathbf{Y}}^{(\mathbf{B}_2)} = (f^{(2)}(\mathbf{B}_{1,:}), \dots, f^{(2)}(\mathbf{B}_{k,:}))^\top.$$

4: Form, for each $i = 1, \dots, m$, the hybrid matrix $\mathbf{C}_i$ by copying $\mathbf{B}$ and replacing its i-th column with the i-th column of $\mathbf{A}$, then evaluate

$$\tilde{\mathbf{Y}}^{(\mathbf{C}^{(i)})} = (f^{(1)}((\mathbf{C}_i)_{1,:}), \dots, f^{(1)}((\mathbf{C}_i)_{k,:}))^\top,$$

and

$$\tilde{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})} = (f^{(2)}((\mathbf{C}_i)_{1,:}), \dots, f^{(2)}((\mathbf{C}_i)_{k,:}))^\top.$$

5: Compute

$$\alpha = \left(\frac{\mathrm{Cov}(\tilde{\mathbf{Y}}^{(\mathbf{A})}, \tilde{\mathbf{Y}}^{(\mathbf{A}_2)})}{\mathrm{Var}(\tilde{\mathbf{Y}}^{(\mathbf{A}_2)})}\right)\left(\frac{\phi}{\phi+1}\right).$$

6: Draw two independent $k' \times m$ design matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$; for each i, form $\overline{\mathbf{C}}_i$ by replacing column i of $\overline{\mathbf{B}}$ with the i-th column of $\overline{\mathbf{A}}$.

7: Evaluate the low-fidelity vectors

$$\overline{\mathbf{Y}}^{(\mathbf{A}_2)} = (f^{(2)}(\overline{\mathbf{A}}_{1,:}), \ldots, f^{(2)}(\overline{\mathbf{A}}_{k',:}))^\top,$$
$$\overline{\mathbf{Y}}^{(\mathbf{B}_2)} = (f^{(2)}(\overline{\mathbf{B}}_{1,:}), \ldots, f^{(2)}(\overline{\mathbf{B}}_{k',:}))^\top,$$
$$\overline{\mathbf{Y}}^{(\mathbf{C}_2^{(i)})} = (f^{(2)}((\overline{\mathbf{C}}_i)_{1,:}), \ldots, f^{(2)}((\overline{\mathbf{C}}_i)_{k',:}))^\top.$$

8: **for** $i = 1, \ldots, m$ **do**

9: **for** $r = 1, \ldots, R_b$ **do**

10: Draw (with replacement) a list L of length k from $\{1, 2, \ldots, k\}$.

11: Compute

$$\tilde{S}_i(r) = \frac{\frac{1}{k}\sum_{j\in L} \tilde{Y}_j^{(\mathbf{A})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L}\frac{(\tilde{Y}_j^{(\mathbf{A})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{A})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2},$$

$$\tilde{S}_{T_i}(r) = 1 - \frac{\frac{1}{k}\sum_{j\in L} \tilde{Y}_j^{(\mathbf{B})}\tilde{Y}_j^{(\mathbf{C}^{(i)})} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L}\frac{(\tilde{Y}_j^{(\mathbf{B})})^2+(\tilde{Y}_j^{(\mathbf{C}^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{B})}+\tilde{Y}_j^{(\mathbf{C}^{(i)})}}{2}\right)^2}.$$

12: Also using the same L on the low-fidelity counterparts, compute

$$\tilde{S}_i^{(2)}(r) = \frac{\frac{1}{k}\sum_{j\in L} \tilde{Y}_j^{(\mathbf{A}_2)}\tilde{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{A}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L}\frac{(\tilde{Y}_j^{(\mathbf{A}_2)})^2+(\tilde{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{A}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2},$$

$$\tilde{S}_{T_i}^{(2)}(r) = 1 - \frac{\frac{1}{k}\sum_{j\in L} \tilde{Y}_j^{(\mathbf{B}_2)}\tilde{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{B}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k}\sum_{j\in L}\frac{(\tilde{Y}_j^{(\mathbf{B}_2)})^2+(\tilde{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k}\sum_{j\in L}\frac{\tilde{Y}_j^{(\mathbf{B}_2)}+\tilde{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}.$$

13: Draw (with replacement) a list L_2 of length k' from $\{1, 2, \ldots, k'\}$.

14: Compute

$$\overline{S}_i^{(2)}(r) = \frac{\frac{1}{k'}\sum_{j\in L_2} \overline{Y}_j^{(\mathbf{A}_2)}\overline{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k'}\sum_{j\in L_2}\frac{\overline{Y}_j^{(\mathbf{A}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k'}\sum_{j\in L_2}\frac{(\overline{Y}_j^{(\mathbf{A}_2)})^2+(\overline{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k'}\sum_{j\in L_2}\frac{\overline{Y}_j^{(\mathbf{A}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2},$$

$$\overline{S}_{T_i}^{(2)}(r) = 1 - \frac{\frac{1}{k'}\sum_{j\in L_2} \overline{Y}_j^{(\mathbf{B}_2)}\overline{Y}_j^{(\mathbf{C}_2^{(i)})} - \left(\frac{1}{k'}\sum_{j\in L_2}\frac{\overline{Y}_j^{(\mathbf{B}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}{\frac{1}{k'}\sum_{j\in L_2}\frac{(\overline{Y}_j^{(\mathbf{B}_2)})^2+(\overline{Y}_j^{(\mathbf{C}_2^{(i)})})^2}{2} - \left(\frac{1}{k'}\sum_{j\in L_2}\frac{\overline{Y}_j^{(\mathbf{B}_2)}+\overline{Y}_j^{(\mathbf{C}_2^{(i)})}}{2}\right)^2}.$$

15: Form the CVB bootstrap replicates

$$\tilde{S}_i^{\mathrm{CVB}}(r) = \tilde{S}_i(r) - \alpha(\tilde{S}_i^{(2)}(r) - \overline{S}_i^{(2)}(r)), \quad \tilde{S}_{T_i}^{\mathrm{CVB}}(r) = \tilde{S}_{T_i}(r) - \alpha(\tilde{S}_{T_i}^{(2)}(r) - \overline{S}_{T_i}^{(2)}(r)).$$

16: **end for**

17: Let $q_{S_i}^{\mathrm{lo}}$, $q_{S_i}^{\mathrm{hi}}$ be the empirical $\gamma/2$ and $1-\gamma/2$ quantiles of $\{\tilde{S}_i^{\mathrm{CVB}}(r)\}_{r=1}^{R_b}$.

18: Let $q_{S_{T_i}}^{\mathrm{lo}}$, $q_{S_{T_i}}^{\mathrm{hi}}$ be the empirical $\gamma/2$ and $1-\gamma/2$ quantiles of $\{\tilde{S}_{T_i}^{\mathrm{CVB}}(r)\}_{r=1}^{R_b}$.

Ensure: Return the level-γ confidence intervals

$$[\, q_{S_i}^{\mathrm{lo}},\ q_{S_i}^{\mathrm{hi}}\,] \quad \text{and} \quad [\, q_{S_{T_i}}^{\mathrm{lo}},\ q_{S_{T_i}}^{\mathrm{hi}}\,].$$

19: **end for**

Bibliography

[1] J.S. Acero Triana, H. Ajami, and S. Razavi. The dilemma of objective function selection for sensitivity and uncertainty analyses of semi-distributed hydrologic models across spatial and temporal scales. *Journal of Hydrology*, 650:132482, 2025.

[2] T. Agrawal. *Hyperparameter Optimization in Machine Learning*. Apress Berkely, CA, 2020.

[3] A. Albà, A. Adelmann, and D. Rochman. Uncertainty quantification of spent nuclear fuel with multifidelity Monte Carlo. *Annals of Nuclear Energy*, 211:110892, 2025.

[4] A. Alsheghri, Y. Ladini, G. Hosseinimanesh, I. Chafi, J. Keren, F. Cheriet, and F. Guibault. Adaptive point learning with uncertainty quantification to generate margin lines on prepared teeth. *Applied Sciences*, 14(20):9486, 2024.

[5] N. Aretz, M. Gunzburger, M. Morlighem, and K. Willcox. Multifidelity uncertainty quantification for ice sheet simulations. *Computational Geosciences*, 29(1):5, 2025.

[6] P.M. Aronow and B.T. Miller. *Foundations of Agnostic Statistics*. Cambridge University Press, 2019.

[7] A. Ayoub, H.M. Wainwright, L. Wang, and G. Sansavini. An enhanced Fourier neural operator surrogate for radioactive plume transport forecasting. *Stochastic Environmental Research and Risk Assessment*, 38(8):3165–3175, 2024.

[8] Z. Azarhoosh and M. Ilchi Ghazaan. A review of recent advances in surrogate models for uncertainty quantification of high-dimensional engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 433:117508, 2025.

[9] I. Azzini, T.A. Mara, and R. Rosati. Comparison of two sets of Monte Carlo estimators of Sobol indices. *Enviromental Modelling and Software*, 144:105167, 2021.

[10] R. Ballester-Ripoll and M. Leonelli. Global sensitivity analysis of uncertain parameters in Bayesian networks. *International Journal of Approximate Reasoning*, 180:109368, 2025.

[11] Y. Bao and J.M. Velni. Adaptive uncertainty quantification for scenario-based control using meta-learning of Bayesian neural networks. *IFAC-PapersOnLine*, 58(28):486–491, 2024.

[12] S. Benlakhdar, M. Rziza, and O. Rachid. Statistical modeling of directional data using a robust hierarchical Von Mises distribution model: perspectives for wind energy. *Computational Statistics*, 37:1–21, 09 2022.

[13] M. Birrell, Y. Li, and R. Astroza. A Gaussian process surrogate approach for analyzing parameter uncertainty in mechanics-based structural finite element models. *Engineering Structures*, 336:120435, 2025.

[14] M. Blanchette, É. Foulon, and A.N. Rousseau. Spatio-temporal sensitivity analysis of the wetland modules of a semi-distributed hydrological model. *Journal of Hydrology*, 623:129783, 2023.

[15] V. Bone, C. van der Heide, K. Mackle, I. Jahn, P.M. Dower, and C. Manzie. Gradient-enhanced deep Gaussian processes for multifidelity modeling. *Journal of Computational Physics*, 520:113474, 2025.

[16] P. Brandimarte. *Handbook in Monte Carlo Simulation: Applications in Financial Engineering, Risk Management, and Economics*. John Wiley & Sons, Hoboken, NJ, 2014.

[17] R.E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

[18] G. Cao, Z. Fei, and C. Vogel. Uncertainty quantification analysis in the blade element momentum method. *Renewable Energy*, 250:123181, 2025.

[19] G. Casella and R.L. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, 2nd edition, 2002.

[20] H. Chang and D. Zhang. A comparative study of stochastic collocation methods for flow in spatially correlated random fields. *Communications in Computational Physics*, 6:509–535, 2009.

[21] S. Chaturvedi and E. Rajasekar. Application of a probabilistic LHS-PAWN approach to assess building cooling energy demand uncertainties. *Building Simulation*, 15:373–387, 2022.

[22] Y. Chen, L. Wang, Y. Comlek, and W. Chen. A latent variable approach for non-hierarchical multi-fidelity adaptive sampling. *Computer Methods in Applied Mechanics and Engineering*, 421:116773, 2024.

[23] COMSOL. Diffusion coefficient. `https://www.comsol.com/multiphysics/diffusion-coefficient?` Accessed: 2025-07-20.

[24] R. Cools and D. Nuyens, editors. *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC, Leuven, Belgium, April 2014*, volume 163 of *Springer Proceedings in Mathematics & Statistics*, Cham, Switzerland, 2016. Springer International Publishing.

[25] R. Costa, S. Clain, R. Loubère, and G.J. Machado. Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state

convection–diffusion equation with Dirichlet condition. *Applied Mathematical Modelling*, 54:752–767, 2018.

[26] D. Cousineau. Varieties of confidence intervals. *Advances in Cognitive Psychology*, 13(2):140–155, 2017.

[27] S. Cucurachi, C.F. Blanco, B. Steubing, and R. Heijungs. Implementation of uncertainty analysis and moment-independent global sensitivity analysis for full-scale life cycle assessment models. *Journal of Industrial Ecology*, 26(2):374–391, 2022.

[28] A. Daly and P. Zannetti. *Ambient Air Pollution*. The EnviroComp Institute, 2007.

[29] J.W. Deardorff. Convective velocity and temperature scales for the unstable planetary boundary layer and for Rayleigh convection. *Journal of Atmospheric Sciences*, 27(8):1211 – 1213, 1970.

[30] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaä, and L.E. Meester. *A Modern Introduction to Probability and Statistics Understanding Why and How*. London: Springer, 2005.

[31] A. Derry, M. Krzywinski, and N. Altman. Neural networks primer. *Nature Methods*, 20:165–167, 2023.

[32] J. Di Stefano. A confidence interval approach to data analysis. *Forest Ecology and Management*, 187(2-3):173–183, 2003.

[33] P. Drobinski, C. Coulais, and B. Jourdier. Surface wind-speed statistics modelling: Alternatives to the Weibull distribution and performance evaluation. *Boundary-Layer Meteorology*, 157:97–123, 2015.

[34] J. Droniou. Finite volume schemes for diffusion equations: Introduction to and review of modern methods. *Mathematical Models and Methods in Applied Sciences*, 24(08):1575–1619, 2014.

[35] W.L. Dunn and J.K. Shultis. *Exploring Monte Carlo Methods (Second Edition)*. Elsevier, 2023.

[36] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 10(1):1–26, 1979.

[37] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994.

[38] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite elements and fast iterative solvers: With applications in incompressible fluid dynamics*. Oxford University Press, 2014.

[39] S. Fan, X. Hong, Z. Liao, C. Ren, Y. Yang, J. Wang, and Y. Yang. Adaptive sampling Bayesian algorithm for constrained black-box optimization problems. *AIChE Journal*, 71(4):e18715, 2025.

[40] X. Fan, J. Zhao, C. Qi, X. Wang, and X. Lyu. On the water entry impact characteristics of high-speed vehicle with an arbitrary Lagrangian-Eulerian method. *Applied Ocean Research*, 150:104118, 2024.

[41] I.G. Farcaș, B. Peherstorfer, T. Neckel, F. Jenko, and H. Bungartz. Context-aware learning of hierarchies of low-fidelity models for multi-fidelity uncertainty quantification. *Computer Methods in Applied Mechanics and Engineering*, 406:115908, 2023.

[42] B. Fischer, A. Ramage, D.J. Silvester, and A.J. Wathen. On parameter choice and iterative convergence for stabilised discretisations of advection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 179(1):179–195, 1999.

[43] D. Fowler, P. Brimblecombe, J. Burrows, M.R. Heal, P. Grennfelt, D.S. Stevenson A. Jowett, E. Nemitz, M. Coyle, X. Liu, Y. Chang, G.W. Fuller, M.A. Sutton, Z. Klimont, M.H. Unsworth, and M. Vieno. A chronology of global

air quality. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2183):20190314, 2020.

[44] D.G. Fox. Uncertainty in air quality modeling: A summary of the AMS workshop on quantifying and communicating model uncertainty, Woods Hole, Mass., September 1982. *Bulletin of the American Meteorological Society*, 65(1):27–36, 1984.

[45] D. Gao and X. Yuan. A hybrid Lagrangian-Eulerian model for vector-borne diseases. *Journal of Mathematical Biology*, 89(16), 2024.

[46] X. Gao, C. Peng, C. Chen, and T. Guo. Seismic fragility estimation through real-time hybrid simulation and surrogate-based multi-fidelity Monte Carlo predictor. *Engineering Structures*, 318:118763, 2024.

[47] A. Garbo, J. Parekh, T. Rischmann, and P. Bekemeyer. Multi-fidelity adaptive sampling for surrogate-based optimization and uncertainty quantification. *Aerospace*, 11(6):448, 2024.

[48] J. G. García, B. Hosseini, and J. M. Stockie. Simultaneous model calibration and source inversion in atmospheric dispersion models. *Pure and Applied Geophysics*, 178:757–776, 2021.

[49] H. Garcke, R. Nürnberg, and Q. Zhao. Arbitrary Lagrangian-Eulerian finite element approximations for axisymmetric two-phase flow. *Computers and Mathematics with Applications*, 155:209–223, 2024.

[50] M.B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[51] R. Giraldo, V. Leiva, and C. Castro. An overview of kriging and cokriging predictors for functional random fields. *Mathematics*, 11(15), 2023.

[52] P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer, New York, 2004.

[53] T. Gorda, O. Komoltsev, A. Kurkela, and A. Mazeliauskas. Bayesian uncertainty quantification of perturbative QCD input to the neutron-star equation of state. *Journal of High Energy Physics*, 2023(6):2, 2023.

[54] A. Gruber, M. Gunzburger, L. Ju, R. Lan, and Z. Wang. Multifidelity Monte Carlo estimation for efficient uncertainty quantification in climate-related modeling. *Geoscientific Model Development*, 16(4):1213–1229, 2023.

[55] T. Guilmeau, N. Branchini, E. Chouzenoux, and V. Elvira. Adaptive importance sampling for heavy-tailed distributions via α-divergence minimization. In S. Dasgupta, S. Mandt, and Y. Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3871–3879. PMLR, 02–04 May 2024.

[56] J.M. Hammersly and D.C. Handscomb. *Monte Carlo Methods*. Methuen and Co LTD, 1964.

[57] B. Harding, C. Tremblay, and D. Cousineau. Standard errors: A review and evaluation of standard error estimators using Monte Carlo simulations. *The Quantitative Methods for Psychology*, 10(2):107–123, 2014.

[58] M. Hashemi and N. Mahjouri. Global sensitivity analysis-based design of low impact development practices for urban runoff management under uncertainty. *Water Resources Management*, 36:2953–2972, 2022.

[59] P. Holnicki and Z. Nahorski. Emission data uncertainty in urban air quality modeling—case study. *Environmental Modeling & Assessment*, 20:583–597, 2015.

[60] B. Hosseini and J.Stockie. Estimating airborne particulate emissions using a finite-volume forward solver coupled with a Bayesian inversion approach. *Computers and Fluids*, 154:27–43, 2017.

[61] B. Hosseini and J.M. Stockie. Bayesian estimation of airborne fugitive emissions using a Gaussian plume model. *Atmospheric Environment*, 141:122–138, 2016.

[62] B. Hou, M. Yuan, K. Diao, and X. Du. Probabilistic characteristic analysis of seismic performance of water distribution system based on quasi-Monte Carlo simulation. *Earthquake Engineering and Engineering Vibration*, 24(3):595–611, 2025.

[63] S.E. Houng and L. Ceferino. Fast probabilistic seismic hazard analysis through adaptive importance sampling. *Bulletin of the Seismological Society of America*, 115(2):646–663, 02 2025.

[64] X. Huang. *Data-Driven Approach for Turbulence Modeling in Rotating Flows and Stratified Flows*. Phd dissertation, The Pennsylvania State University, University Park, PA, 2023.

[65] Z. Huang, Y.K. Kwok, and Z. Xu. Efficient algorithms for calculating risk measures and risk contributions in copula credit risk models. *Insurance: Mathematics and Economics*, 115:132–150, 2024.

[66] G.R. Hunt and T.S. Van Den Bremer. Classical plume theory: 1937-2010 and beyond. *IMA Journal of Applied Mathematics*, 76:424–448, 2010.

[67] T. Ishigami and T. Homma. An importance quantification technique in uncertainty analysis for computer models. *Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, pages 398–403, 1990.

[68] A. Itkin. *Pricing Derivatives Under Lévy Models*. Springer New York, 2017.

[69] R.K. Jain, Z. Cui, and J.K. Domen. *Enviromental Impact of Mining and Mineral Processing*. Butterworth-Heinemann, 2016.

[70] A. Janon, T. Klein, A. Lagnoux, and M. Nodet C.Prieur. Asymptotic normality and efficiency of two Sobol index estimators. *ESAIM: Probability and Statistics*, 18:342–364, 2014.

[71] H. Jiang, J. Wang, J. Wu, and W. Geng. Comparison of numerical methods and metaheuristic optimization algorithms for estimating parameters for wind energy potential assessment in low wind regions. *Renewable and Sustainable Energy Reviews*, 69:1199–1217, 2017.

[72] L. Jiang, X. Ji, S. Yang, and Z. Zhang. Assessment of wind resource potential at different heights in urban areas: A case study of Beijing. *Renewable Energy*, 248:123175, 2025.

[73] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, 2009.

[74] D.P. Johnson, N. Ravi, G. Filippelli, and A. Heintzelman. A Novel Hybrid Approach: Integrating Bayesian SPDE and Deep Learning for Enhanced Spatiotemporal Modeling of PM2.5 Concentrations in Urban Airsheds for Sustainable Climate Action and Public Health. *Sustainability*, 16(23), 2024.

[75] M. Karl, S.E. Walker, G. Sousa-Santos, M. Vogt, D. Vo-Thanh, S. Lopez-Aparicio, P. Schneider, M.O.P. Ramacher, and P.D. Hamer. The urban dispersion model episode v10.0 – part 1: An Eulerian and sub-grid-scale air quality model and its application in Nordic winter conditions. *Geoscientific Model Development*, 13(9):4323–4353, 2020.

[76] N. Khlaifat, A. Altaee, J. Zhou, and Y. Huang. Evaluation of wind resource potential using statistical analysis of probability density functions in New South Wales, Australia. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 47(1):194–211, 2025.

[77] A.D. Kiureghian and O. Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112, 2009.

[78] J. Kocijan, N. Hvala, M. Perne, P. Mlakar, B. Grašič, and M.Z. Božnar. Surrogate modelling for the forecast of Seveso-type atmospheric pollutant dispersion. *Stochastic Environmental Research and Risk Assessment*, 37:275–290, 2023.

[79] A. Kumar and A. Singh. Entropy-based groundwater quality evaluation with multivariate analysis and Sobol sensitivity for non-carcinogenic health risks in mid-Gangetic plains, India. *Environmental Geochemistry and Health*, 47:186, 2025.

[80] A. Kumar and G. Tiwari. A methodology for imprecise moment-independent global sensitivity analysis with limited data of copula-dependent inputs: Application for slopes. *Journal of Engineering Mechanics*, 150(4):04024004, 2024.

[81] A. Kumar and G. Tiwari. A re-sampling statistics based imprecise moment independent global sensitivity analysis methodology with limited data of uncorrelated and correlated geotechnical properties. *Structures*, 70:107686, 2024.

[82] S. Lan, S. Li, and B. Shahbaba. Scaling up Bayesian uncertainty quantification for inverse problems using deep neural networks. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4):1684–1713, 2022.

[83] J. Lelong, Z. El Filali Ech-Chafiq, and A. Reghai. Automatic control variates for option pricing using neural networks. *Monte Carlo Methods and Applications*, 27(2):91–104, 2021.

[84] Z. Li, G. Sun, C. He, X. Liu, R. Zhang, Y. Li, D. Zhao, H. Liu, and F. Zhang. Multi-variable regression methods using modified Chebyshev polynomials of class 2. *Journal of Computational and Applied Mathematics*, 346:609–619, 2019.

[85] Z. Li, G. Yang, Q. Zhao, X. Deng, L. Guibas, B. Hariharan, and G. Wetzstein. Neural control variates with automatic integration. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, page 1–9. ACM, July 2024.

[86] T.I. Liaudat, M. Mars, M.A Price, M. Pereyra, M.M. Betcke, and J.D. McEwen. Scalable Bayesian uncertainty quantification with data-driven priors for radio interferometric imaging. *RAS Techniques and Instruments*, 3(1):505–534, 08 2024.

[87] A. Litvinenko, D. Logashenko, R. Tempone, E. Vasilyeva, and G. Wittum. Uncertainty quantification in coastal aquifers using the multilevel Monte Carlo method. *Proceedings in Applied Mathematics and Mechanics*, 23(1):e202300005, 2023.

[88] J. Liu, H. Liu, C. Zhang, J. Cao, A. Xu, and J. Hu. Derivative-variance hybrid global sensitivity measure with optimal sampling method selection. *Mathematics*, 12(3), 2024.

[89] E. Lumet, M.C. Rochoux, T. Jaravel, and S. Lacroix. Uncertainty-aware surrogate modeling for urban air pollutant dispersion prediction. *Building and Environment*, 267:112287, 2025.

[90] A. March and K. Willcox. Constrained multifidelity optimization using model calibration. *Structural and Multidisciplinary Optimization*, 46(1):93–109, 2012.

[91] L.S Matott, J.E. Babendreier, and S.T. Purucker. Evaluating uncertainty in integrated environmental models: A review of concepts and tools. *Water Resources Research*, 45(6), 2009.

[92] Met Office. MIDAS: UK Mean Wind Data. NCAS British Atmospheric Data Centre, 2006. Accessed: 23 July 2025.

[93] D. Mooney. A guide for local authorities purchasing air quality monitoring equipment. Technical Report AEAT/ENV/R/2088 Issue 2, Department for Environment, Food and Rural Affairs (Defra), London, UK, 2006.

[94] H. Nam, H. Yoo, H. Kim, and K.H. Kim. Corrected multi-fidelity surrogate model based on deep neural networks for predicting aerodynamic loads of missile. *International Journal of Aeronautical and Space Sciences*, 2025.

[95] M.A. Noureddine, F. De Martin, R. El Meouche, M.A. Sammuneh, F. Ababsa, and M. Beaufils. Neural network-based metamodel of synthetic seismograms: Application for uncertainty quantification. *Engineering Applications of Artificial Intelligence*, 151:110613, 2025.

[96] A. Novruzi. *A Short Introduction to Partial Differential Equations*. Springer Cham, 2023.

[97] W.L. Oberkampf, S.M. DeLand, B.M. Rutherford, K.V. Diegert, and K.F. Alvin. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety*, 75(3):333–357, 2002.

[98] A.B. Owen. *Monte Carlo theory, methods and examples*, chapter 8, pages 3–38. https://artowen.su.domains/mc/, 2013.

[99] A.B. Owen and Z. Pan. Gain coefficients for scrambled Halton points. *SIAM Journal on Numerical Analysis*, 62(3):1021–1038, 2024.

[100] S.G.S. Pai and I.F.C. Smith. *Using Regularized Linear-Regression Surrogate Models for Accurate Probabilistic Structural Identification*, chapter 47, pages 367–373. American Society of Civil Engineers (ASCE), 2019.

[101] Z. Pan and A.B. Owen. Skewness of a randomized quasi-Monte Carlo estimate. *Journal of Complexity*, 90:101956, 2025.

[102] Q. Parsons, D. Nowak, M. Bortz, T. Johnson, A. Mark, and F. Edelvik. Machine learning surrogates for the optimization of curing ovens. *Engineering Applications of Artificial Intelligence*, 133:108086, 2024.

[103] R. Pasupathy, B. W. Schmeiser, M. R. Taaffe, and J. Wang. Control-variate estimation using estimated control means. *IIE Transactions*, 44(5):381–385, 2012.

[104] 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.

[105] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[106] Y. Peng, M.C. Fu, J. Hu, P. L'Ecuyer, and B. Tuffin. Generalized likelihood ratio method for stochastic models with uniform random numbers as inputs. *European Journal of Operational Research*, 321(2):493–502, 2025.

[107] Z. Peng, B. Zhang, D. Wang, X. Niu, J. Sun, H. Xu, J. Cao, and Z. Shen. Application of machine learning in atmospheric pollution research: A state-of-the-art review. *Science of the Total Enviroment*, 910:168588, 2024.

[108] T. Pham and A.A. Gorodetsky. Ensemble approximate control variate estimators: Applications to multifidelity importance sampling. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1250–1292, 2022.

[109] H. Pishro-Nik. *Introduction to Probability, Statistics, and Random Processes*. Kappa Research, LLC, 2014.

[110] J.D. Pribe, P.E. Leser, S.R. Yeratapally, and E.H. Glaessgen. Multi-model Monte Carlo estimation for crystal plasticity structure–property simulations of additively manufactured metals. *Computational Materials Science*, 247:113481, 2025.

[111] A.M. Propp and D.M. Tartakovsky. Transfer learning on multi-dimensional data: A novel approach to neural network-based surrogate modeling. *Journal of Machine Learning for Modeling and Computing*, 6(2):13–27, 2025.

[112] A. Puy, W. Becker, S. Lo Piano, and A. Saltelli. A comprehensive comparison of total-order estimators for global sensitivity analysis. *International Journal for Uncertainty Quantification*, 12(2):1–18, 2022.

[113] J.S. Pérez Guerrero, L.C.G. Pimentel, T.H. Skaggs, and M.Th. van Genuchten. Analytical solution of the advection–diffusion transport equation using a change-of-variable and integral transform technique. *International Journal of Heat and Mass Transfer*, 52(13):3297–3304, 2009.

[114] E. Qian, B. Peherstorfer, D. O'Malley, V. V. Vesselinov, and K. Willcox. Multi-fidelity Monte Carlo estimation of variance and sensitivity indices. *SIAM/ASA J. Uncertainty Quantification*, 6(2):683–706, 2018.

[115] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[116] S. Razavi and H.V. Gupta. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 1. Theory. *Water Resources Research*, 52(1):423–439, 2016.

[117] Y. Ren, G. Wang, Q. Zhang, C. Tao, S. Ji, Q. Wang, and W. Wang. The impact of Chinese new year on air quality in north China based on machine learning. *Atmospheric Environment*, 342:120874, 2025.

[118] A. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, volume 24 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 2008.

[119] R. Y. Rubinstein and D. P. Kroese. *Simulation and The Monte Carlo Method*. John Wiley and Sons, Inc, 2017.

[120] J. Rubio-Herrero, C. Ortiz Marrero, and W.T. Fan. Modeling atmospheric data and identifying dynamics temporal data-driven modeling of air pollutants. *Journal of Cleaner Production*, 333:129863, 2022.

[121] S.S. Sajjadinia, B. Carpentieri, D. Shriram, and G.A. Holzapfel. Multi-fidelity surrogate modeling through hybrid machine learning for biomechanical and finite element analysis of soft tissues. *Computers in Biology and Medicine*, 148:105699, 2022.

[122] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis The Primer*. John Wiley & Sons LTD, 2008.

[123] J. Samper, C. López-Vázquez, B. Pisani, A. Mon, A.C. Samper-Pilar, and F.J. Samper-Pilar. Global sensitivity analysis of reactive transport modelling for the geochemical evolution of a high-level radioactive waste repository. *Applied Geochemistry*, 180:106286, 2025.

[124] J. Sander and J. Beyerer. Bayesian fusion: Modeling and application. In *2013 Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, 2013.

[125] A. Sayeed, Y. Choi, E. Eslami, J. Jung, Y. Lops, and A. K. Salman. A novel CMAQ-CNN hybrid model to forecast hourly surface-ozone concentrations 14 days in advance. *Scientific Reports*, 11(1):10891, 2021.

[126] B.W. Schmeiser, M.R. Taaffe, and J. Wang. Biased control-variate estimation. *IIE Transactions*, 33:219–228, 2001.

[127] J.H. Seinfeld. *Atmospheric Chemistry and Physics of Air Pollution*. John Wiley & Sons, New York, 1 edition, 1986.

[128] J.H. Seinfeld and S.N Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. John Wiley & Sons, New York, 1 edition, 1998.

[129] X. Shao, S. Zhang, X. Liu, Z. Liu, and J. Huang. Rapid prediction for the transient dispersion of leaked airborne pollutant in urban environment based on graph neural networks. *Journal of Hazardous Materials*, 478:135517, 2024.

[130] H. Sharma, L. Novák, and M. Shields. Physics-constrained polynomial chaos expansion for scientific machine learning and uncertainty quantification. *Computer Methods in Applied Mechanics and Engineering*, 431:117314, 2024.

[131] Y. Shi, P. Wei, K. Feng, D.C. Feng, and M. Beer. A survey on machine learning approaches for uncertainty quantification of engineering systems. *Machine Learning for Computational Science and Engineering*, 1(11):1–39, 2025.

[132] H. Snoun, M. Krichen, and H. Chérif. A comprehensive review of Gaussian atmospheric dispersion models : current usage and future perspectives. *Euro-Mediterranean Journal for Enviromental Integration*, 8:219–242, 2023.

[133] I.M. Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4):407–414, 1993.

[134] I. Sosoma, E. Mureithi, and N. Shaban Mbare. Mathematical modeling of brucellosis transmission dynamics and its age-specific applications. *Abstract and Applied Analysis*, 2025(1):6552487, 2025.

[135] J. Stockie. The mathematics of atmospheric dispersion modeling. *SIAM Review*, 53(2):349–372, 2011.

[136] J.M. Stockie, K. Promislow, and B.R. Wetton. A finite volume method for multicomponent gas transport in a porous fuel cell electrode. *International Journal for Numerical Methods in Fluids*, 41(6):577–599, 2003.

[137] Roland B. Stull. *Practical Meteorology: An Algebra-based Survey of Atmospheric Science*. University of British Columbia, 2017.

[138] S. Tiwari, A. Kumar, G. Tiwari, and P. Sharma. An efficient uncertainty analysis of performance of hydrogen storage systems. *Renewable and Sustainable Energy Reviews*, 209:115107, 2025.

[139] O. Tsvetkova and T.B.M.J. Ouarda. A review of sensitivity analysis practices in wind resource assessment. *Energy Conversion and Management*, 238:114112, 2021.

[140] J. Vachon, J. Kerckhoffs, S. Buteau, and A. Smargiassi. Do machine learning methods improve prediction of ambient air pollutants with high spatial contrast? A systematic review. *Environmental Research*, 262:119751, 2024.

[141] M.A. Vazquez-Cruz, R. Guzman-Cruz, I.L. Lopez-Cruz, O. Cornejo-Perez, I. Torres-Pacheco, and R.G. Guevara-Gonzalez. Global sensitivity analysis by means of EFAST and Sobol' methods and calibration of reduced state-variable TOMGRO model using genetic algorithms. *Computers and Electronics in Agriculture*, 100:1–12, 2014.

[142] P. Vitullo, N. R. Franco, and P. Zunino. Deep learning enhanced cost-aware multi-fidelity uncertainty quantification of a computational model for radiotherapy. *Foundations of Data Science*, 7(1):386–417, 2025.

[143] S. Wang, M. Zhang, H. Zhao, P. Wang, S. H. Kota, Q. Fu, and H. Zhang. Extracting regional and temporal features to improve machine learning for hourly air pollutants in urban India. *Atmospheric Environment*, 338:120834, 2024.

[144] A. Whata, K. Dibeco, K. Madzima, and I. Obagbuwa. Uncertainty quantification in multi-class image classification using chest X-ray images of COVID-19 and pneumonia. *Frontiers in Artificial Intelligence*, 7:1410841, 2024.

[145] J. Xiao, F. Fu, and X. Wang. Deep learning based on randomized quasi-Monte Carlo method for solving linear Kolmogorov partial differential equation. *Journal of Computational and Applied Mathematics*, 451:116088, 2024.

[146] Y. Xue, Y. Feng, and X. Zheng. Uncertainty quantification of compressibility corrections on SA turbulence model for high Mach number flows. *Aerospace Science and Technology*, 162:110184, 2025.

[147] H. Yang, Y. Zhang, and G. Li. Air quality index prediction using a new hybrid model considering multiple influencing factors: A case study in China. *Atmospheric Pollution Research*, 14(3):101677, 2023.

[148] Q. Yao, S. Tian, W. Pan, W. Jin, J. Li, and L. Yuan. Prediction and sensitivity analysis of pressure pulsation in a LDI combustor based on a fully connected neural network. *Case Studies in Thermal Engineering*, 71:106151, 2025.

[149] M. Yaseen, A. Sadek, W. Osman, M. Altahhan, X. Wu, M. Avramova, and K. Ivanov. Sensitivity and uncertainty analysis in pebble-bed reactors: A study using the High-Temperature Code Package (HCP). *Annals of Nuclear Energy*, 219:111428, 2025.

[150] M. Zhang, C. Liu, R. Hu, H. Han, and Q. Wu. Comparing global sensitivity analysis methods for the joint thermal design parameters of a space manipulator based on the Sobol' and PAWN. *Thermal Science and Engineering Progress*, 47:102338, 2024.

[151] Z. Zhang, J. Li, and B. Liu. Annealed adaptive importance sampling method in pinns for solving high dimensional partial differential equations. *Journal of Computational Physics*, 521:113561, 2025.

[152] M. Zhong, T. Xiao, M.J. Krause, M. Frank, and S. Simonis. A stochastic Galerkin lattice Boltzmann method for incompressible fluid flows with uncertainties. *Journal of Computational Physics*, 517:113344, 2024.