Surrogate Groundwater Models

Michael J. Asher

A thesis submitted for the degree of

Doctor of Philosophy

The Australian National University

March 2021

Except where otherwise indicated, this thesis is my own original work.

Michael J. Asher

1 March 2021

To the one who did the most important work of all, while I did this.

Acknowledgments

My sincere thanks to my family and advisors for their encouragement and help. Thanks also to John Jakeman at Sandia, the CSIRO and the Australian Government for their generous support.

Abstract

The spatially and temporally variable parameters and inputs to complex groundwater models typically result in long runtimes which hinder comprehensive analysis. These analyses typically involving calibration, sensitivity analysis and uncertainty propagation. Surrogate modelling aims to provide a simpler, and hence faster, model which emulates the specified output of a more complex model as a function of its inputs and parameters. A faster model enables more model runs, which are critical for understanding models through methods such as sensitivity and uncertainty analysis. Three broad categories of approaches to surrogate modelling are data-driven, projection, and hierarchical-based. Data-driven surrogates approximate a groundwater model through an empirical model that captures the input-output mapping of the original model. Projection-based models reduce the dimensionality of the parameter space by projecting the governing equations onto a basis of orthonormal vectors. In hierarchical or multi-fidelity methods the surrogate is created by simplifying the representation of the physical system, such as by ignoring certain processes, or reducing the numerical resolution. A surrogate method can only be of practical value if it significantly reduces model runtimes in common contexts, robustly emulates the output of the complex model and can be implemented simply and flexibly. While a gamut of surrogate techniques have been applied to groundwater and similar partial differential equation (PDE) based simulators, the practicability of all approaches is not clear.

Among the promising approaches to emulation are Polynomial Chaos Expansions (PCE), Multi-fidelity Stochastic Collocation (MFSC) and modern Deep Learning (DL) Neural Networks (NN). These are investigated in the thesis. They represent the three categories above as projection-based (depending on implementation), multi-fidelity and data-driven methods respectively. However all three methods are black box in that they do not require re-implementation of the complex model, making them relevant to practitioners.

PCEs are a versatile, highly efficient and statistically rigorous surrogate approach with a number of well developed methods for their calibration. In the framework we present, they are suited to accelerating sensitivity analysis and uncertainty propagation of generic models with a moderate number of parameters. MFSC overcomes many shortcomings of other surrogate methods by employing a lower resolution model as the surrogate. The approach is shown to faithfully emulate spatially and temporally distributed parameters. Complex model runs can be easily parallelized and reused. While traditional NN are not the most promising surrogate technique, there is enormous potential in the DL software frameworks associated with the recent boom in their popularity. This promise extends not just to efficient uncertainty analysis and data assimilation for groundwater modelling, but numerical modelling in general.

From a wide survey of the literature we advance three promising techniques. Model emulation using either PCE, MFSC or DL as demonstrated in this thesis will add value to practical groundwater modelling by not only reducing model runtimes but deepening understanding of the underlying model. The PCE approach iteratively selects the underlying groundwater model samples, training a surrogate with less than 1% error in under 200 model runs. The MFSC method achieves similar accuracy with less than 30 full resolution model runs. The DL approaches are less efficient, requiring 500 model runs. However they emulate the full spatially distributed output of the underlying model, and can be applied in situations with 100s of uncertain parameters. Further contributions of this work include two improvements to the MFSC algorithm, reducing surrogate error by two orders of magnitude. We identify a gap between existing research in applied DL, theory-rich applied mathematics and the increasing quantity of spatially distributed data. We create a new surrogate form which combines PCE theory with a DL implementation, and develop another which captures physical aquifer properties during the training of a state of the art DL architecture.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Groundwater is a key natural resource, making up about 50% of drinking water and 43% of irrigation water globally [Connor, 2015]. Understanding its behavior is also key to mining, construction and natural resource management. Even in the age of the "Internet of Things", we cannot collect data dense and frequent enough to capture the variability present in the rivers, landscapes and climates that make up groundwater systems. Hence computer simulations are the principal tool available to policy makers and engineers wishing to understand, regulate or use groundwater. Indeed, since at least the early 1970s, numerical models based on partial differential equations have been used to simulate groundwater levels and inform policy [Ungemach, 1975]. Natural groundwater systems cover large areas and are highly heterogenous in their properties. The measurements we do have are typically subject to large uncertainties (think of what a flood does to a stream gauge, or how well a 10cm core sample represents an aquifer's geology 10km away). Thus there is a need for numerical implementations with a fine spatial and temporal scale in the governing equations, but also to analyze the uncertainty inherent in these models - something typically achieved by running them repeatedly in various configurations. Unfortunately both of these requirements increase the time and cost of computation.

The briefing for this thesis was to investigate computationally efficient methods for facilitating the uncertainty analysis of groundwater models. A secondary research question was how to assimilate monitoring data (heads, flow rates, remote sensing, flux measurements) into these models. To assimilate more data into models typically

requires model inversion. Heads, flow rates, flux and some remote sensing variables are model outputs. So to include them we must optimize the inputs to the model so its outputs match these data. Again, this is a computationally expensive process. Hence the two goals boil down to the same bottleneck. To make better policy decisions we need to run complex models many times, and our ability to do this is limited by the computational cost of doing so. Surrogate modelling, that is emulating a complex model with a simpler one, is a promising technique [Razavi et al., 2012b] for reducing the computational cost of running a model. Many surrogate approaches are simply statistical emulators of input-output data produced by the complex model [e.g. Tang et al., 2020]. More involved methods include multi-fidelity [e.g. Menberg et al., 2020] approaches where the numerical resolution of the underlying model must be changed and projection based surrogates [e.g. Xia et al., 2020] where the underlying equations are altered. There are a many approaches available in the literature, but few have seen widespread adoption in practical settings. Understanding the benefits of surrogate methods is made challenging by the wide range of reported efficiency gains and the breadth of modelling contexts (e.g. data assimilation, sensitivity analysis, uncertainty propagation, Bayesian inversion) in which they are applied.

An additional benefit of employing surrogate modelling is that it may provide further insight into the model or data, in addition to a more efficient emulator. An example is the sensitivity indices which can be derived [Sudret, 2008; Jakeman et al., 2019a] from a polynomial chaos surrogate (discussed in Chapter 3). This thesis focuses on how surrogate modelling techniques may add value in a groundwater modelling context by increasing computational efficiency and understanding of the underlying model.

1.2 Outline

In Chapter 2 we introduce surrogate modelling and review the application of surrogates to groundwater models. In the following chapters we further progress three cutting edge surrogate techniques for application to groundwater models.

Polynomial Chaos Expansions (PCE), addressed in Chapter 3, are a highly efficient

and statistically rigorous surrogate approach with a number of well developed methods for their calibration [Xiu and Karniadakis, 2003; Najm, 2009]. Once calibrated, the statistical moments and sensitivity indices of the outputs can be computed analytically. We apply an efficient adaptive method to calibrate a PCE on a variety of inputs and outputs of a groundwater model and demonstrate how the surrogate can be used for uncertainty propagation and sensitivity analysis in a computationally optimal manner. Further, we develop a new method for the calibration of a PCE using nested polynomials implemented in a modern Deep Learning software package discussed later in Chapter 5 .

In Chapter 4 we look at Multi-fidelity Stochastic Collocation, another versatile emulator, which employs two or more resolutions of the same model. Surrogate runs involve a solution to the lowest fidelity model, which practitioners may prefer to a data-driven emulator whose structure bears no resemblance to the original model. The approach is well suited to spatially and temporally distributed parameters. We identify two improvements to the algorithm in practical groundwater modelling settings.

Finally, in Chapter 5 we look at an exploding area of research in Neural Networks (NNs). This area has enjoyed several booms in popularity with only limited benefit to the groundwater modelling community. NNs are now faster and more complex, incrementally improving this work. However, more importantly, the recent rise of Deep Learning (DL) software frameworks represent an unprecedented opportunity for numerical modelling in general, and groundwater modelling specifically. They provide simple and open access to cutting edge computational resources, algorithms and software quality. We show that they are as relevant to researchers of applied mathematics as to the machine learning practitioners who developed them, and there is great promise in greater cross pollination between the two communities. We develop two state of the art DL models to be best in class surrogates of groundwater models in a wide range of scenarios, particularly where emulating large numbers of inputs and outputs. We review recent work applying DL methods to groundwater simulation and speculate that these methods might be useful beyond emulation. We move the field forward by providing the first case studies (to our knowledge) of

how DL methods function fundamentally differently to other "black-box" surrogate methods. Architectures are possible which mimic physical systems in ways the internals of other statistical methods cannot.

Chapter 2

Review

2.1 Introduction

The physical properties and processes that determine groundwater flow are highly heterogeneous. To adequately capture such heterogeneity, many groundwater management problems require complex, fully distributed models that can accommodate fields for the hydraulic properties and boundary conditions that vary in time and space. There has been a tendency towards including more physical processes, increasing numerical resolution, and expanding the model domain in fully distributed groundwater models [Leube et al., 2012; Doherty and Simmons, 2013], typically solved using a finite difference approximation such as implemented in MODFLOW [Harbaugh, 2005] or a finite element method, as used for example by FEFLOW [Diersch, 2005]. Greater conceptual model complexity, however, translates to a larger number of parameters and increased model runtimes.

Long runtimes inhibit the use of models in applications which require many model runs, such as integrated modelling (where groundwater flow models are coupled with models of different processes), uncertainty analysis, sensitivity analysis and inverse modelling. Slow runtimes also prevent models being used in real time, necessary for applications such as decision support. Furthermore, the "curse of dimensionality" is encountered as the number of samples required to cover the parameter space in uncertainty analysis, sensitivity analysis or calibration increases exponentially with the number of model parameters. An increase in model runtime means that numerical resolution needs to be reduced or physical processes ignored to decrease runtime and make many model runs computationally tractable. Surrogate

models have the potential to speed up complex models without sacrificing accuracy or detail.

Also known as metamodels [e.g. Blanning, 1975], reduced models [e.g. Willcox and Peraire, 2002], model emulators [e.g. O'Hagan, 2006], proxy models [e.g. Bieker et al., 2007], lower-fidelity models [e.g. Robinson et al., 2008] and response surfaces [e.g. Regis and Shoemaker, 2005], surrogate models are computationally cheaper models designed to approximate the dominant features of a complex model. While the main motivation for applying a surrogate model is achieving computational efficiency [Razavi et al., 2012b], other reasons do exist.

Simple surrogate models can reduce numerical instability which facilitates calibration and uncertainty analysis [Doherty and Christensen, 2011]. The process of building an emulator can reveal insensitive outputs and irrelevant parameters of a complex model [Young and Ratto, 2011]. Surrogates may serve as didactic tools for analyzing model simplification and the ways in which models simplify reality [Watson et al., 2013]. They can also be used to smooth an objective function surface, allowing the use of gradient based, nonlinear programming methods for optimization problems [Hemker et al., 2008; Kavetski and Kuczera, 2007], or to reduce ill-conditioning of a conjugate gradient optimizer by using eigenvector approximations [Vuik et al., 1999]. The gain in computational efficiency opens the door for exploration of structural model uncertainty by simultaneous simulation and calibration of alternative model structures [Matott and Rabideau, 2008] or inclusion of data and physical processes at multiple scales [Weinan and Engquist, 2003]. Also, surrogates with sufficiently short runtimes have been used in interactive decision support environments [Roach and Tidwell, 2009]. Lastly, surrogates have been used for "complementary" modelling, where a simple model is fitted to the residual of a complex model to improve accuracy [Demissie et al., 2009; Xu et al., 2012].

This review is structured around a taxonomy of surrogate models, based on their mathematical structure. We follow Robinson et al. [2008] by classifying surrogate models into three categories as outlined in Table 2.1: data-driven surrogates involving empirical approximations of the complex model output calibrated on a set of inputs and outputs of the complex model (snapshots); projection-based methods, where the

governing equations are projected onto a reduced dimensional subspace characterized by a basis of orthonormal vectors; and multi-fidelity methods, built by simplifying the underlying physics or reducing numerical resolution. The three categories are addressed in Sections 2.2, 2.3 and 2.4. Rather than describing each technique in detail, we give an illustrative example. We then discuss the other techniques in each category, how they relate to the example, and their historical and potential application to groundwater modelling. Since reducing runtime is a major motivation for using surrogate models, we discuss some alternative techniques for doing so in Section 2.5.

Several recent publications compare a subset of surrogate modelling approaches [Forrester et al., 2008; Forrester and Keane, 2009; Castelletti et al., 2012; Frangos et al., 2010; Antoulas, 2005; Gugercin and Antoulas, 2000]. Razavi et al. [2012b] published a review of surrogate models in the water resources literature. However, there appears a general imbalance in the literature towards data-driven methods. Razavi et al. [2012b] for example, devote 19 pages to data-driven methods, and only four to the section covering both multi-fidelity and projection-based approaches. In particular, we note the sparse treatment of surrogates developed for spatially distributed models [Pasetto et al., 2011].

Two important motivations exist to correct this imbalance. Firstly, Razavi et al. [2012a] have noted that despite the optimism about data-driven surrogate modelling in the literature, it can be an inefficient and unreliable approach to optimizing complex numerical models. Secondly, Razavi et al. [2012b] observe that "lower-fidelity surrogates" (encompassing both projection-based and multi-fidelity methods reviewed here) overcome many of the limitations of data-driven approaches; namely they can be applied to larger numbers of parameters, and perform better further from snapshots used in calibration of the surrogate. These advantages are particularly relevant to spatially distributed models which are ubiquitous in groundwater studies. Despite this, many sophisticated "lower-fidelity" methods have not yet been applied to groundwater problems.

As well as summarizing the current state of the art in surrogate models, we aim to give a more comprehensive coverage of surrogate modelling approaches and discuss their historical, as well as their potential, application to distributed groundwater

models. We aim to glean from the wider research community surrogate methods which show promise for groundwater applications.

Few well-accepted criteria exist for assessing surrogate modelling approaches. Razavi et al. [2012b] note the need to validate the whole surrogate-enabled analysis framework rather than testing the surrogate in isolation. They also note the need for more developed metrics of computational efficiency gains, and recommend approaches which make use of estimates of surrogate-introduced uncertainty. Several standards of good modelling practice [Jakeman et al., 2006] are particularly relevant to surrogate modelling. Approaches should be as simple as possible to avoid coding errors. Justification should be given for the choice of surrogate technique. Insofar as possible, surrogate performance should be thoroughly analyzed. Finally, results and methods should be reported in sufficient detail as to allow informed criticism.

We analyze the surrogate approaches below with reference to the following criteria.

1. If purported to do so, the approach should significantly increase computational efficiency. Ideally this would be assessed based on:

   (a) Average runtime of a single surrogate versus complex model run.

   (b) The number of complex model runs used to calibrate the surrogate, justified with reference to surrogate-introduced uncertainty.

   (c) Runtime for combining surrogate and complex models, typically this involves (iteratively) calibrating the surrogate on the output of complex model runs.

   (d) An indication of effort required to apply the method. Does the complex model code typically require modification?

2. The surrogate should allow more thorough analysis and testing of the original model.

3. An indication of surrogate-introduced uncertainty should be given.

Table 2.1: Taxonomy of surrogate models

Category	Also known as	Examples
Data-driven surrogates involving empirical approximations of the complex model output calibrated on a set of inputs and outputs of the complex model (snapshots).	Response surface, statistical and black box methods.	Polynomials [Hussain et al., 2002], neural networks [Kourakos and Mantoglou, 2009; Yan and Minsker, 2006], Gaussian processes [Stone, 2011; Kennedy and O'Hagan, 2001], kriging [Baú and Mayer, 2006; Garcet et al., 2006] , radial basis functions [Regis and Shoemaker, 2005], support vector machines [Yoon et al., 2011], dynamic mode analysis [Young and Ratto, 2011], (generalized) polynomial chaos expansions [Laloy et al., 2013] , genetic programming [Fallah-Mehdipour et al., 2013], Bayesian networks [Fienen et al., 2013].
Projection-based methods, where the governing equations are projected onto a reduced dimension subspace characterized by a basis of orthonormal vectors. Typically divided into SVD and Krylov based methods.	Reduced order, reduced basis and model reduction methods.	Proper orthogonal decomposition (POD) [McPhee and Yeh, 2008; Siade et al., 2012; Galbally et al., 2010], Karhunen-Loève expansion [Laloy et al., 2013], proper generalized decomposition [Chinesta et al., 2011], Krylov subspace methods [Dunbar and Woodbury, 1989; Woodbury et al., 1990], dynamic mode decomposition [Ghommem et al., 2013], Fourier mode reduction [Willcox and Megretski, 2005] and (certified) reduced basis [Lieberman et al., 2010; Chen et al., 2010; Knezevic and Peterson, 2011; Efendiev et al., 2012].
Multi-fidelity based surrogates, built by simplifying the underlying physics or reducing numerical resolution.	Multiscale, hierarchical and physically based methods.	Multigrid method [Ashby and Falgout, 1996; Saied and Mahinthakumar, 1998; Thum et al., 2011], multiscale finite element method [Shi et al., 2012; Hou and Wu, 1997], heterogeneous multiscale method [Weinan et al., 2002], residual free bubbles [Sangalli, 2003], conservative subgrid [Arbogast, 2002] and variational multiscale method [Ganapathysubramanian and Zabaras, 2007; Hughes et al., 1998].

2.2 Data-driven methods

In this section, we introduce data-driven methods by describing a particular example, Gaussian processes. We then discuss other techniques and their application to groundwater modelling.

Let us denote the output of a complex model F as $\mathbf{h} = F(\boldsymbol{\theta})$. Consider a vector, $\boldsymbol{\theta}$, whose values represent a groundwater model's inputs and parameters: the hydraulic properties, sources, sinks, and initial conditions. Data-driven surrogates attempt to emulate the mapping from $\boldsymbol{\theta}$ to $\mathbf{h}$ without considering the inner workings of F. The complex model is run on a set $\{\boldsymbol{\theta}^{(i)}\}$, known as the design of experiment, to produce

a set of snapshots, $\{\mathbf{h}^{(i)})\}$. Depending on the data-driven method, a function

$$\widehat{F}(\boldsymbol{\phi}, \cdot) \tag{2.1}$$

is chosen, and the snapshots are used to fit the hyperparameters $\boldsymbol{\phi}$ such that

$$\widehat{F}(\boldsymbol{\phi}, \boldsymbol{\theta}^{(i)}) \approx F(\boldsymbol{\theta}^{(i)}) \qquad \text{for all } i. \tag{2.2}$$

Data-driven methods differ in how they select the snapshots and the functional form of $\widehat{F}$, chosen to emulate the $\boldsymbol{\theta}$ to $\mathbf{h}$ relationship.

2.2.1 Gaussian processes - an example

Gaussian processes [e.g. Kennedy and O'Hagan, 2001; Stone, 2011] assume the relationship can be captured by a surrogate

$$\widehat{F}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbf{f}(\boldsymbol{\theta})^T \boldsymbol{\beta} + e(\boldsymbol{\theta}), \tag{2.3}$$

where $\mathbf{f}(\boldsymbol{\theta})$ are chosen regression functions, and the unknown hyperparameters, $\boldsymbol{\phi}$, include the regression coefficients, $\boldsymbol{\beta}$, and the parameters of the stochastic process, e. The stochastic process, e, has zero mean and a specified positive semi-definite covariance function. A common choice is a linear regression function

$$\mathbf{f}(\boldsymbol{\theta}) = (1, \boldsymbol{\theta}) \tag{2.4}$$

and the covariance function

$$Cov(e(\boldsymbol{\theta}^{(j)}), e(\boldsymbol{\theta}^{(k)})) = \sigma^2 \exp\left(-\sum_i \omega_i |\theta_i^{(j)} - \theta_i^{(k)}|^{\alpha_i} \right). \tag{2.5}$$

Two of the hyperparameters, $\boldsymbol{\beta}$ and σ, can be estimated by analytic expressions using a set of snapshots [Sacks et al., 1989], but numerical optimization of ω_i and α_i is required.

2.2.2 Other data-driven methods

Razavi et al. [2012b] thoroughly cover data-driven surrogates with only a few exceptions, which we mention briefly below.

Bayesian Networks Bayesian Networks are annotated acyclic graphs which give the joint probability distributions of a number of variables. Nodes are random variables $X_i, \ldots, X_n$ and edges (links) are probabilities $P(x_i|\pi_i)$ where x_i is a realization of X_i, and π_i its parents. Fienen et al. [2013] create a Bayesian Network surrogate of a groundwater model. In order to reduce the number of parameters, surrogate parameters are taken as the maximum or mean of a property along a cross section of the complex model domain. As is a requirement for Bayesian Networks, parameters are discretized into bins. The approach makes causal relationships easy to see and can be integrated simply with larger Bayesian Networks. Being largely a static method, their main restriction is the difficulty in emulating the temporally varying outputs.

Transfer functions Transfer function models consist of a function

$$h_k = F(h_{k-1}, \cdots, h_{k-n}, u_k, \cdots, u_{k-m}), \tag{2.6}$$

which predicts an output time series h_k in terms of its history and possibly a driving input series u_k. Young and Ratto [2011] present a transfer function based method which, as a dynamic model, purportedly exhibits greater predictive capability for a wider range of scenarios than static approaches. Immediate drawbacks are the limitation to linear models, and difficulty in creating a map from the transfer function to complex model parameters.

Response Matrices Cheng et al. [2011] construct a response matrix surrogate using "one at a time" sensitivity analysis of a finite difference groundwater model's output heads to changes in pumping and recharge. The sensitivity analysis is used to compute influence coefficients, $\frac{\partial h_{i,T}}{\partial Q_{p,t}}$, which represent the change in head at location i and time T caused by a change in pumping rate of well p at time t. Using these

derivatives, head can then be approximated by a truncated Taylor series around $Q_{p,0}$.

Genetic programming Genetic programming [e.g. Fallah-Mehdipour et al., 2013] seeks to find a functional form that approximates the input-output relationship of a model. An evolutionary algorithm searches through relationships randomly created using inputs, outputs, random variables, and operators (functions, arithmetic operators, boolean operators, or logical expressions). Disadvantages of genetic programming surrogates include computationally demanding calibration and over-fitting.

Polynomial chaos The popular polynomial chaos method involves expanding a random variable (parameter or output) in an orthogonal polynomial basis. Any stochastic process with finite variance can be expanded as a polynomial of random variables so long as the polynomials are orthogonal with respect to distribution of that random variable. Xiu and Karniadakis [2002] describes the method and the Askey scheme, which details which polynomial basis is to be used with which distribution. A common example is the use of Hermite polynomials of Gaussian processes (normally distributed variables). The approach has recently been incorporated into a two step MCMC (Markov Chain Monte Carlo) Bayesian inversion of a groundwater flow model [Laloy et al., 2013]. As is common in the wider literature [e.g. Ghanem and Dham, 1998] the polynomial chaos surrogate is coupled with a Karhunen-Loève parameterization of conductivity. The most common approaches of finding the coefficients of the polynomials are stochastic collocation and pseudo-spectral methods, which are non-intrusive as they do not involve editing the complex model code. However, it is possible to use an intrusive Galerkin method which would make the method a projection based approach [Herzog et al., 2008].

2.2.3 Comparisons and Recommendations

There are notable similarities between a number of the data-driven methods listed in Table 2.1. Forrester et al. [2008] and Forrester and Keane [2009] note that certain forms of kriging, Guassian process models, radial basis functions, support vector machines, and single-layer neural networks with radial coordinate neurons are identical.

While many methods do share mathematical structures, such as Guassian function of distance, as in (2.5), it would be a mistake to consider all methods labeled as "kriging" or "radial basis functions" to be identical. Each of the terms listed in Table 2.1 represents a sizable volume of literature devoted to developing a variety of methods known by that name.

Furthering the confusion, recent efforts have combined several data-driven surrogate methods. Rather than selecting a data-driven surrogate method a priori, Viana et al. [2009] suggest a framework for using multiple surrogates simultaneously. Matott and Rabideau [2008] propose a method for the simultaneous calibration of multiple models. In their test case employing multiple analytic surrogates, they improve the optimized objective function and reduce runtime. Schöbi et al. [2015] combine polynomial chaos expansions with kriging.

Comparisons of data-driven techniques in the literature are numerous [e.g. Garcet et al., 2006; Forrester and Keane, 2009; Razavi et al., 2012a; Villa-Vialaneix et al., 2012; Espinet and Shoemaker, 2013]. However, as Forrester and Keane [2009] note, no method performs best universally. Results will depend on the application and factors such as the size of training set [Breiman, 2001]. There is some consensus [Forrester and Keane, 2009; Villa-Vialaneix et al., 2012; Razavi et al., 2012b] about the strength of kriging [Jones et al., 1998] and radial basis function [Regis and Shoemaker, 2007] based optimization frameworks. Subsequent literature [Espinet and Shoemaker, 2013] indicates the viability of these methods for groundwater modelling.

For calibration and uncertainty quantification, we would, in addition to kriging and radial basis functions, advocate for the use of polynomial chaos expansions. The polynomial chaos method of Marzouk and Najm [2009] and related stochastic partial differential equation approaches form a very active area of research. These methods are worthy of consideration since they have only recently been applied to groundwater models and few comparisons with other data-driven approaches have been published other than the theoretical work of O'Hagan [2013]. Advantages of polynomial chaos methods include the ability to calculate sensitivity indices of parameters and their interaction analytically from the expansions [Sudret, 2008] and the depth of literature devoted to selecting samples at which to calibrate the surrogate [e.g. Xiu

and Hesthaven, 2005]. While it has been noted [Forrester and Keane, 2009; Razavi et al., 2012b] that it is better to select snapshots iteratively, taking into account surrogate error, many surrogates do not have established methods for selecting snapshots of the complex model.

Data-driven surrogates have a number of limitations. Razavi et al. [2012a] warn they may be subject to computationally demanding calibration, subjective structure, and over-fitting. They are of the opinion that the optimism in the literature around data-driven surrogates is in some areas ill-founded, pointing out that in some cases optimization methods without surrogates were more effective than those employing surrogates. It is worth noting that there are well established methods for addressing issues such as over-fitting [Hastie et al., 2009]. Other disadvantages include the possibility of getting trapped in local minima [Demissie et al., 2009] and the limitation of only being able to handle a relatively small number of parameters. This latter limitation often results in aquifer parameters being assumed homogeneous [Mugunthan et al., 2005] or known a priori [Yan and Minsker, 2006]. However, novel methods of calibrating polynomial chaos surrogates using adaptive sparse grids [Jakeman and Roberts, 2013] have been applied to increasingly large numbers of parameters.

Data-driven surrogates are not expected to perform well away from design sites. A known limitation of all global surrogates, where a single surrogate is used for the full parameter range of interest, is their inability to adequately capture heterogeneity [Najm, 2009]. Solutions for polynomial chaos surrogates have been proposed involving piecewise polynomial bases [Wan and Karniadakis, 2005], multivariate wavelet bases [Le Maitre et al., 2004], or sparse grid collocation with local interpolants [Matthies and Keese, 2005; Xiu and Hesthaven, 2005]. Marzouk and Najm [2009] proposes adaptive polynomial degree, sparse truncation of the basis, or partitioning the prior support as other possible improvements.

Despite their drawbacks, well used data-driven approaches remain a valuable tool in applications such as decision support and integrated modelling, where it may be necessary to limit both the number of parameters and the ranges which they take. Quick runtimes once calibrated and their non-intrusive nature make data-driven methods particularly useful for these applications.

2.3 Projection-based methods

The essence of projection-based surrogates is to replace a vector space, $\mathbf{h}$, by a linear combination of (orthogonal) basis vectors

$$\mathbf{h} \approx \mathbf{\Phi}\mathbf{h}_r. \tag{2.7}$$

Complexity is reduced because the number of basis vectors (columns of $\mathbf{\Phi}$), and therefore elements in $\mathbf{h}_r$, needed to approximate $\mathbf{h}$ is often much smaller than the dimension of $\mathbf{h}$. Projection-based methods vary according to which vector space is approximated and how the basis vectors are found. The process invariably involves substituting $\mathbf{\Phi}\mathbf{h}_r$ for $\mathbf{h}$ in the governing equations, and using an orthogonality condition to simplify the result.

2.3.1 Proper orthogonal decomposition - an example

The proper orthogonal decomposition (POD) method (also referred to as singular value decomposition (SVD), principal component analysis and empirical orthogonal functions) is a common projection-based surrogate. It proceeds as follows [McPhee and Yeh, 2008]. For a complex model F, we obtain a set of snapshots $\{\mathbf{y}^{(i)}\}$ from a set of samples of the input space $\{\boldsymbol{\theta}^{(i)}\}$ where $\mathbf{y}^{(i)} = F(\boldsymbol{\theta}^{(i)})$. A set of normalized snapshots is then combined in a matrix

$$\mathbf{Y} = \begin{bmatrix} \frac{\mathbf{y}^{(1)}}{\|\mathbf{y}^{(1)}\|} & \frac{\mathbf{y}^{(2)}}{\|\mathbf{y}^{(2)}\|} & \cdots & \frac{\mathbf{y}^{(n)}}{\|\mathbf{y}^{(n)}\|} \end{bmatrix}. \tag{2.8}$$

The method requires the eigenvalues and eigenvectors of the covariance matrix $\mathbf{C} = \mathbf{Y}\mathbf{Y}^T$. However, in practice the smaller $\mathbf{C}_s = \mathbf{Y}^T\mathbf{Y}$ is used to solve the eigenproblem

$$\mathbf{C}_s\mathbf{g}^{(i)} = \mathbf{g}^{(i)}\lambda^{(i)} \qquad i \in \{1, \cdots, n\}. \tag{2.9}$$

The eigenvectors $\mathbf{\Phi}$ of $\mathbf{C}$ are then computed as

$$\mathbf{\Phi} = \mathbf{Y}\begin{bmatrix} \mathbf{g}^{(1)} & \mathbf{g}^{(2)} & \cdots & \mathbf{g}^{(n)} \end{bmatrix}\mathbf{\Lambda}^{-\frac{1}{2}} \tag{2.10}$$

where

$$\mathbf{\Lambda}^{-\frac{1}{2}} = \begin{bmatrix} \sqrt{\lambda^{(1)}} & & & \\ & \sqrt{\lambda^{(2)}} & & \\ & & \ddots & \\ & & & \sqrt{\lambda^{(n)}} \end{bmatrix}^{-1} \tag{2.11}$$

The dimensions of $\mathbf{\Phi}$ are the same as $\mathbf{Y}$. However, the proportion of the variance in $\mathbf{Y}$ explained by each eigenvector (column of $\mathbf{\Phi}$) is given by the normalized value of its corresponding eigenvalue

$$\psi^{(i)} = \frac{\lambda^{(i)}}{\sum_{j=1}^{n} \lambda^{(j)}}. \tag{2.12}$$

In practice, a minimum acceptable explained variance $\theta_{\min}$ can be chosen, and m identified such that $\sum_{i=1}^{m} \psi^{(i)} \geq \theta_{\min}$, where $\psi^{(i)}$ are arranged in descending order. A useful surrogate is created when m is orders of magnitude smaller than n.

In the case of confined aquifers (linear flow) where transmissivity does not vary in time, the discretized groundwater equation may be written [Harbaugh, 2005]

$$\mathbf{M}\frac{d\mathbf{h}}{dt} + \mathbf{A}\mathbf{h} = \mathbf{q} \tag{2.13}$$

where $\mathbf{h}$ is head, $\mathbf{q}$ a source-sink term, and $\mathbf{M}$ and $\mathbf{A}$ are derived from hydraulic conductivity, storativity, and head-dependent boundary conditions.

Substituting $\mathbf{\Phi}\mathbf{h}_r$ for $\mathbf{h}$ in (2.13), left multiplying with $\mathbf{\Phi}^T$, and noting $\mathbf{h}_r$ is time dependent but not $\mathbf{\Phi}$, we obtain

$$\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}\frac{d\mathbf{h}_r}{dt} + \mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}\mathbf{h}_r = \mathbf{\Phi}^T\mathbf{q}. \tag{2.14}$$

Although this reduced order equation can be solved using similar methods as those used for (2.13), this typically involves editing the complex model solvers.

2.3.2 The Lanczos method - another example

As with the POD method described above, the Lanczos [Dunbar and Woodbury, 1989] approach is based on the reduced order equation (2.14). However, the basis vectors making up $\mathbf{\Phi}$ are computed differently. Rather than using the eigenvectors of the snapshot covariance as basis vectors, the Lanczos method uses the solution to the generalized eigenvalue problem

$$\mathbf{A\Phi} = \mathbf{M\Phi\Lambda}, \tag{2.15}$$

where $\mathbf{A}$ and $\mathbf{M}$ are from (2.13), $\mathbf{\Phi}$ are the eigenvectors to be found, and $\mathbf{\Lambda}$ is the diagonal matrix of corresponding eigenvalues. Again, we only need find the m smallest eigenvalues and corresponding vectors. Using the orthogonality properties of the eigenvectors, $\mathbf{\Phi}^T\mathbf{M\Phi} = \mathbf{I}$ and $\mathbf{\Phi}^T\mathbf{A\Phi} = \mathbf{\Lambda}$, (2.14) can be reduced to

$$\frac{d\mathbf{h}_r}{dt} + \mathbf{\Lambda h}_r = \mathbf{\Phi}^T\mathbf{q}. \tag{2.16}$$

If $g = \mathbf{\Phi}^T\mathbf{q}$ is time independent, this has the analytic solution

$$(h_r)_i(t) = (h_r)_i^0 e^{-\lambda_i t} + \frac{g_i}{\lambda_i}(1 - e^{-\lambda_i t}). \tag{2.17}$$

A time stepping algorithm can be applied for calculating time dependent g as done by Sahuquillo [1983]. However, finding the m smallest eigenvalues of $\mathbf{A\Phi} = \mathbf{M\Phi\Lambda}$ is a computationally challenging problem.

Note that we can rewrite (2.15) as

$$\mathbf{\Phi\Lambda}^{-1} = \mathbf{A}^{-1}\mathbf{M\Phi}. \tag{2.18}$$

The Lanczos algorithm seeks the solution to the tridiagonal system

$$\mathbf{LT} = \mathbf{A}^{-1}\mathbf{ML} \tag{2.19}$$

by finding the tridiagonal matrix

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{bmatrix} \tag{2.20}$$

and the so called Lanczos vectors $\mathbf{L} = (L_1, L_2, \cdots, L_m)$.

The eigenvalues of $\mathbf{T}$ give good approximations to the smaller eigenvalues of $\mathbf{A\Phi} = \mathbf{M\Phi\Lambda}$. The strength of the approach lies in the efficient algorithm for computing L_i, α_i and β_i using a standard tridiagonal solution algorithm. Similarly to above, substituting $\mathbf{Lh}_r$ for $\mathbf{h}$ in (2.13) and left multiplying with $\mathbf{L}^T\mathbf{MA}^{-1}$, gives

$$\mathbf{L}^T\mathbf{MA}^{-1}\mathbf{ML}\frac{d\mathbf{h}_r}{dt} + \mathbf{L}^T\mathbf{MLh}_r = \mathbf{L}^T\mathbf{MA}^{-1}\mathbf{q}. \tag{2.21}$$

Which can be reduced to

$$\mathbf{T}\frac{d\mathbf{h}_r}{dt} + \mathbf{h}_r = \mathbf{L}^T\mathbf{MA}^{-1}\mathbf{q} \tag{2.22}$$

using the orthogonality properties $\mathbf{L}^T\mathbf{MA}^{-1}\mathbf{ML} = \mathbf{T}$ and $\mathbf{L}^T\mathbf{ML} = \mathbf{I}$. A solution for $\mathbf{h}_r$ can then be found using standard time integration techniques.

2.3.3 Other projection-based methods

Other reduced basis methods employ different techniques to find bases which reduce the dimensionality of the parameter or output space. Table 2.2 lists a number of different approaches.

We note here that projection-based surrogates, despite employing many of the same techniques, differ from parameterization methods. In a discussion of parameterization methods for calibrating reservoir models, Oliver and Chen [2011] mention zonation, pilot points, splines, Karhunen-Loève approximations, wavelets, and singular vectors of the model sensitivities. While these might be considered surrogate

Table 2.2: Methods for computing a reduced basis

Method	Basis	Reference
POD	SVD of snapshot covariance	McPhee and Yeh [2008]
Proper generalized decomposition	Separation of variables	Chinesta et al. [2011]
Dynamic mode decomposition	SVD of forward operator	Ghommem et al. [2013]
Lanczos and Arnoldi	Krylov subspace	Dunbar and Woodbury [1989], Woodbury et al. [1990]
Fourier model reduction	Fourier expansion of the discrete-frequency transfer function	Willcox and Megretski [2005] Gugercin and Willcox [2008]
(Certified) reduced basis	Greedy algorithm	Lieberman et al. [2010], Chen et al. [2010], Knezevic and Peterson [2011]

parameterizations, they do not emulate the model. The distinguishing feature of the work listed in Table 2.2 is that each uses an intrusive (model driven) method to derive an orthogonal basis.

2.3.4 Comparisons and Recommendations

Antoulas [2005] and Gugercin and Antoulas [2000] divide "reduced models" into two categories: SVD based (e.g. POD) and Krylov subspace based. While SVD based methods have an error bound they cannot be applied to highly complex systems. Antoulas [2005] argue that Krylov methods can be implemented iteratively and so are more appropriate for systems of high complexity. He describes a method combining the two categories designed to overcome the limitations of both. However Frangos et al. [2010] claim that Krylov methods are limited to linear cases. There seems to be very limited application of Krylov methods to groundwater, some being the Lanczos algorithm by Dunbar and Woodbury [1989] and the Arnoldi method by Woodbury et al. [1990].

There has been more attention paid to the POD method which, for a given dimension, minimizes the least squares error of the surrogate on the snapshots [Frangos et al., 2010]. Hay et al. [2012] note that variation in parameters reduces the effectiveness of a POD surrogate. They propose replacing each mode with a first-order Taylor expansion. A finite difference method is used to compute the sensitivity of the mode to variations in each parameter. Ghommem et al. [2013] compare POD and dynamic mode decomposition methods and find the latter basis makes more accurate predictions for a variety of parameter and boundary condition values. Chinesta et al. [2011] advocate for proper generalized decomposition based on empirical studies, but note the lack of rigorous mathematical foundations. Nouy [2007] note that the method allows greater online computational savings compared to POD.

For a linear system, Willcox and Megretski [2005] compare their approach to POD and Arnoldi methods, and demonstrate superior computational efficiency and error bounds over a wide range of frequencies. Chen et al. [2010] points out that POD based methods have only the largest ignored singular value as an error approximation, and Krylov methods have no estimate at all. He advocates the certified reduced basis method employing the"greedy" algorithm [e.g. Knezevic and Peterson, 2011]) - so called because it iteratively adds basis vectors in the direction of the residual. Buffa et al. [2012] provide the proof underlying the greedy algorithm's error estimate. Lieberman et al. [2010] employ the reduced basis method on the parameter and state spaces of a groundwater inverse problem.

Rewieński Michałand White [2006] propose a method which combines a reduced basis and a quasi piecewise linear approximation of the state function for a non-linear differential equation. It is claimed the approach outperforms POD on their case studies.

The projection based approach of Sahuquillo [1983] has been applied by Andreu et al. [1996] to provide real-time spatially distributed groundwater flow modelling for decision support.

Projection based methods have two principal drawbacks: the basis vectors depend on the snapshots used to compute them, making inverse modelling and uncertainty analysis difficult, and solving the reduced model typically involves editing the model

code. To address the former issue, recent work aims at selecting optimal snapshots which cover both the parameter space and the time domain. Lieberman et al. [2010] develop an approach for the reduced basis method, Siade et al. [2010]; Baú [2012] for POD and Pasetto et al. [2013, 2014]; Boyce and Yeh [2014] for syntheses of the two methods. For confined aquifers, reduced models have been demonstrated to run orders of magnitude faster than the full order equivalent. However, the approach does not yield similar results in nonlinear problems as many more basis vectors need to be included [Cardoso et al., 2009].

2.4 Multi-fidelity methods

Multi-fidelity surrogates refer to those constructed from the complex model by reducing numerical resolution, increasing tolerances, or removing processes. Perhaps the simplest surrogate possible is created by reducing the numerical discretization of the complicated model. Previous research has focused on how to upscale properties from the scale of measurements or a fine grid to a coarse grid for rapid computation. However, Farthing et al. [2012] note that under-resolved models can produce inaccurate objective function values and may produce "false" solutions.

The disadvantages of simply reducing resolution and the requirement for finer detail has led to multi-fidelity methods that attempt to combine models at multiple levels of complexity to attain the detail of the complex, at the speed of the simple. This typically involves solving the global problem on a coarse grid, along with multiple local problems on a finer grid. Methods often differ in how they relate the results; in particular, in how they set the boundary conditions for the local problems.

Weinan and Engquist [2003] differentiate between homogeneous multiscale methods, which use identical models at different scales and heterogeneous methods, which allow for different processes at each scale; for example, Darcy's Law at one scale, and lattice Boltzmann pore-scale effects at another. In this review we consider only homogeneous multiscale methods, which we refer to as multi-fidelity methods as they can be more readily classified as surrogates. While they do not have the advantage of incorporating multiple physical processes, they do still simplify the inclusion of data

at multiple scales, possibly negating the need for upscaling.

2.4.1 The multiscale finite volume method - an example

Consider the groundwater flow equation (the continuous form of (2.13))

$$\nabla \cdot (K \nabla h) = S_s \frac{\partial h}{\partial t} + q. \tag{2.23}$$

For the multiscale finite volume method (MsFV) developed by Jenny et al. [2003] (see Hajibeygi et al. [2008] for a detailed explanation), the model domain Ω is divided into a coarse grid with M control volumes (grid cells) $\bar{\Omega}_m$ and a dual coarse grid, formed by joining the midpoints of the control volumes of the coarse grid. The dual grid has N control volumes $\widetilde{\Omega}_n$. Head on the fine grid is computed as

$$h(\mathbf{x}) \approx \sum_{n=1}^{N} \left(\sum_{m=1}^{M} \phi_{m,n}(\mathbf{x}) \bar{h}_m + \phi_n^*(\mathbf{x}) \right) \tag{2.24}$$

where basis functions $\phi_{m,n}$ and correction functions ϕ_n^* are computed by solving the following systems of equations for each dual grid cell [Lunati and Jenny, 2008]:

$$\begin{aligned} \nabla \cdot (K \nabla \phi_{m,n}) &= 0 \qquad && x \in \widetilde{\Omega}_n \\ (\mathbf{n} \cdot \nabla)((K \nabla \phi_{m,n}) \cdot \mathbf{n}) &= 0 \qquad && x \in \partial \widetilde{\Omega}_n \\ \phi_{m,n}(x_i) &= \delta_{ni} \qquad && \text{at dual grid point } x_i \end{aligned} \tag{2.25}$$

$$\begin{aligned} \nabla \cdot (K \nabla \phi_n^*) &= R_n \qquad && x \in \widetilde{\Omega}_n \\ (\mathbf{n} \cdot \nabla)((K \nabla \phi_n^*) \cdot \mathbf{n}) &= R_n^* \qquad && x \in \partial \widetilde{\Omega}_n \\ \phi_n^*(x_i) &= 0 \qquad && \text{at dual grid point } x_i \end{aligned} \tag{2.26}$$

where R_n and R_n^* are some specified functions. In practice, good results are achieved with $R_n = 0$, but more advanced techniques exist [e.g. Hajibeygi et al., 2008]. In a typical finite volume fashion, (2.24) can then be substituted into (2.23), the resulting equation integrated over $\bar{\Omega}_m$ and Gauss' theorem applied to find the coarse grid

heads $\bar{h}_m$. The MsFV method has potential for wide-scale application were it to be included in the recently released finite volume package MODFLOW-USG [Panday et al., 2013].

2.4.2 Other multi-fidelity methods

In this section we discuss the application of several multi-fidelity methods to groundwater modelling. The multiscale finite element method (MsFEM) [Efendiev and Hou, 2007] proceeds similarly to the above MsFV approach, but without the correction functions. The approach has recently been applied to groundwater flow by He et al. [2013]. Recent work [Efendiev et al., 2012, 2013] has combined MsFEM with projection based approaches to reduce online runtime. Sun [2008] reviews a number of multiscale methods for groundwater modelling including the multiscale finite element method (MsFEM), the mixed MsFEM, subgrid upscaling, mixed mimitec multiscale methods for corner-point grids, the stochastic variation multiscale method, the MsFV method and ghost node local grid refinement. All but the latter two are mixed finite element methods. Although MsFEM and MsFV methods are not typically more computationally efficient than the corresponding fine grid solution, they allow greater parallelization [Sun, 2008].

Multigrid [e.g. Bastian and Reichenberger, 2000] and adaptive mesh refinement methods [e.g. Mansour and Spink, 2013] allow the problem to be solved on a hierarchy of resolutions, providing methods for interpolating between multiple scales on the same computational domain. Similar to traditional multigrid methods, the ghost node local grid refinement method involves an iterative interpolation of head from the coarse-grid solution to local fine-grid boundaries, and flux solutions in the opposite direction. The approach is widely available as the MODFLOW-LGR package [Mehl and Hill, 2005]. Vilhelmsen et al. [2012] conclude that the method is more efficient than a uniform fine grid only when the area of refinement covers less than 10-15% of the total model.

Weinan et al. [2007] refer to a number of other classical and recent multiscale methods. Domain decomposition aims to allow the problem to be solved independently on

a number of sub-domains to facilitate parallel computation. Wavelet methods decompose a model into its components of different frequencies, allowing each component to be computed using the appropriate resolution.

In certain cases a simpler model which has a different structure or ignores certain physical processes may be available. For example, Keating et al. [2010] build an ad hoc analytic surrogate to predict influences on groundwater head of nuclear tests. Although potentially useful, the approach is not generic enough to be classified and contrasted against more general methods.

2.4.3 Comparisons and Recommendations

The multi-fidelity surrogates discussed above aim to solve the forward model, and can therefore be substituted for the complex model in any application. While they are intrusive, the MsFEM, MsFV, and multigrid methods maintain the detail and accuracy of the complex model. This makes them ideal for those implementing groundwater modelling codes, but less relevant to practitioners.

Related to such forward model surrogates, several approaches have been developed to use multi-fidelity models in optimization and uncertainty analysis. Robinson et al. [2008] aim to develop a general framework for optimization using multiple models with different sets of parameters. A space mapping, linking low to high fidelity parameters is created by varying low fidelity parameters to match high fidelity output. A common uncertainty analysis approach is to use a coarse model to increase acceptance rates of MCMC during Bayesian inversion of a complex model [Efendiev et al., 2005]. Samples are evaluated by the complex model only if accepted by the surrogate. Cliffe et al. [2011] apply the Multilevel Monte Carlo method to groundwater flow. Narayan et al. [2014] develop an uncertainty quantification method based on multi-fidelity stochastic collocation which uses low-fidelity results to inform sampling locations for the high-fidelity model.

Doherty and Christensen [2011] advocate the use of both simple and complex models. Complex models allow the use of expert information in prior distributions, since their parameters correspond more readily to physical quantities. Simple models

allow numerical stability and efficient calibration. They develop a method to compare the results of simple and complex models to infer whether errors are due to dependency on the complex model null space, measurement noise in the calibration data, or surrogate parameters compensating for the structural simplification. Results allow the correction of predictions of a simple model calibrated on field data. A case study is presented using two finite difference models of different grid resolutions.

If different fidelity models (e.g. a complex model and a data-driven surrogate or the same model at different resolutions) exist, these multi-fidelity inverse modelling methods are particularly attractive since they can be applied non-intrusively. As noted by Robinson et al. [2008], such approaches require a mapping from surrogate to high-fidelity parameters. For groundwater models, even upscaling parameters from a fine to a coarse grid is no simple matter [Wen and Gómez-Hernández, 1996; Vermeulen et al., 2006; Mehl and Hill, 2010]. To what extent multi-fidelity models can be used in the aforementioned inverse modelling frameworks, using simple and practical mappings, remains an open question.

2.5 Decreasing runtime

Decreasing model runtime can be a major motivation for employing one of the surrogates mentioned in Sections 2.2, 2.3 and 2.4. In this section we note some of the techniques available for reducing model runtime other than employing a surrogate.

A Simpler model Determining appropriate model complexity is a complex issue. At one extreme, proponents of analytic models [Matott et al., 2006; Craig and Read, 2010; Estabragh et al., 2013] espouse not just their fast runtimes, but other advantages such as numerical stability. Others, such as Doherty and Christensen [2011]; Miller et al. [2013], argue for the many advantages of increasing model complexity. In any case, computational expense should certainly feature as a consideration when determining model complexity [Hill, 2006].

Parameterization Rather than simplifying the model itself, a common approach is to reduce the number of parameters, thus reducing the number of samples necessary for uncertainty analysis or calibration. A number of parameterization methods are common in practice; for example zones, pilot points, splines, Karhunen-Loève, wavelets, and singular vectors of the model sensitivities [Oliver and Chen, 2011]. Such approaches are often coupled with surrogate modelling techniques. Laloy et al. [2013], for example, combine a polynomial chaos surrogate with a Karhunen-Loève parameterization of conductivity.

Uncertainty analysis algorithm Surrogates are commonly used to accelerate uncertainty quantification and the choice of uncertainty analysis algorithm may also have a large effect on total runtime by reducing the number of model runs required. A large number of uncertainty analysis methods exist. Matott et al. [2009] for example, evaluate 65 software packages for evaluating model uncertainty. Mariethoz et al. [2010] propose a method, iterative spatial re-sampling, that requires fewer model runs than MCMC but yields "reasonably" similar posterior distributions. Franssen and Kinzelbach [2009] compare ensemble Kalman filtering to a Monte Carlo based inversion algorithm and note a significant speed up. The field of stochastic partial differential equations provides many alternatives to sampling. For example, moment differential equations [Winter et al., 2003] involve solving the flow equation for as many probability moments as one is interested in. Perturbation based solutions exist for the first and second moments of head and flux [Guadagnini and Neuman, 1999]. The wide-scale application of the approach is limited by its ability to account for highly heterogeneous media. Vrugt et al. [2003] combine the Metropolis algorithm with an evolutionary approach to reduce the number of model runs required for Bayesian inversion compared to traditional MCMC. Shafii et al. [2014] suggest that simply relaxing the convergence criterion of a MCMC sampler may yield sufficiently accurate uncertainty estimates at a fraction of the computational cost.

Optimization algorithm As is the case with uncertainty analysis, an apt choice of algorithm may reduce the number of forward runs necessary to calibrate a model.

The SVD-assist [Tonkin and Doherty, 2005], adjoint sensitivity [LaVenue and Pickens, 1992], and Principal Component Geostatistical [Kitanidis and Lee, 2014] approaches all claim to reduce the computational burden of the inverse problem. Franssen et al. [2009]; Oliver and Chen [2011]; Zhou et al. [2014] review some of the many optimization algorithms available to modelers.

Computational techniques If the aim of a surrogate model is simply to reduce runtime, improving computational techniques may be a viable alternative. Relevant topics, including choice of solvers, programming language, and parallelization techniques are covered by Miller et al. [2013]. Seemingly banal choices, such as that of compiler, may have significant implementations. For example, Intel compiled MODFLOW can be almost eight times faster than the gfortran version according to Dong and Li [2009].

Of particular note, parallelizing the forward model can reduce runtimes for a variety of applications. HYDROLAB [Erhel et al., 2009], ParFlow [Ashby and Falgout, 1996] and PFLOTRAN [Mills et al., 2009] are examples of softwares developed explicitly for highly parallelized groundwater modelling. Linear reduction of parallel walltime versus the number of processors, up to 27580 cores, has been achieved using PFLOTRAN. By comparison, the most significant result in the data-driven surrogate approaches reviewed by Razavi et al. [2012b] is a 97% reduction in complex model evaluations reported by Regis and Shoemaker [2012]. Ignoring time to develop and run the surrogate, this time saving could be achieved by parallelization of the original model on a 30 core machine. Parallel implementations of the algebraic multigrid solver [Thum et al., 2011] exist for both MODFLOW and FEFLOW. Fienen and Hunt [2015] outline approaches for further parallelizing high throughput applications, where it is unnecessary for computations to interact when running, such as uncertainty analysis and calibration.

2.6 Further research

Based on our review of the methods above, we consider that a number of areas warrant further research. The limitations of global surrogates have been identified. In particular there is a need to deal with complex fields with discontinuities. Localization techniques attempt to overcome these difficulties by dividing parameter space into a number of sub-domains, and employing a different surrogate on each. Such methods are still in their infancy, but are receiving increasing attention (see polynomial chaos references in Section 2.2.3). While localized parameterizations have been developed for groundwater models [Nan and Wu, 2011, e.g.], localized surrogates, to our knowledge, have not.

Quantification of the uncertainty introduced by the surrogate model is another nascent field. Many approaches have not yielded rigorous bounds on surrogate computed posteriors. Application of, and comparison between, approaches is made difficult by the lack of an established measure of surrogate-induced uncertainty [Chen et al., 2010], and surrogate-enabled runtime reduction [Razavi et al., 2012a].

Multiscale methods which preserve the accuracy and detail of the complex forward model hold promise for implementation in industry groundwater modelling codes. For example, as mentioned in Section 2.4, there is the possibility of a MsFV or MsFEM version of the numerous finite volume and finite element codes in widespread use. Further investigation is required into the possible increases in computational efficiency of each multiscale method.

Many data-driven methods rely on ad hoc approaches to select snapshots on which to calibrate the surrogate. Further research is warranted into the application to groundwater model surrogates of the innovative snapshot selection methods from the multi-fidelity [e.g. Narayan et al., 2014] and projection based [e.g. Pasetto et al., 2014] literature. Although these approaches were developed in an uncetainty analysis framework, they could as easily be applied to snapshot selection for integrated modelling or decision support.

In addition to the above areas which require further research, a number of promising techniques appear as active areas of research in the literature. For a small number

of parameters, radial basis function and kriging based optimizers [Regis and Shoemaker, 2012] have been shown to compare favorably to industry standard methods such as gradient-based PEST [Espinet and Shoemaker, 2013]. Tensor based sparse grid collocation methods for polynomial chaos based uncertainty analysis [Espig et al., 2013; Jakeman and Roberts, 2013] deserve attention, as it is a rapidly improving field of research with the ability to handle increasing numbers of parameters. Parameter independent projection based methods [Boyce and Yeh, 2014; Pasetto et al., 2014] have been shown capable of significant runtime reductions while maintaining spatially distributed parameters and outputs, making them applicable to inverse modelling along with other applications.

2.7 Conclusion

The purpose of this review is to summarize approaches to surrogate modelling which are applicable to groundwater modelling. As has been already indicated, no surrogate method is universally superior. We conclude by summarizing our findings on surrogate models appropriate to different use cases.

Multiscale and parameter independent projection based methods have potential to replace groundwater models in any context, since they can emulate the full output of a complex model. However, their likely application is by developers of groundwater model codes, rather than everyday users, because they are intrusive. If the aim of a surrogate model is simply to reduce runtime, we add that computational techniques may be a viable alternative in any context.

For decision support, where very short runtimes are required, data-driven approaches are the obvious choice. Since decision support typically involves a small number of parameters, many of the drawbacks of data-driven methods are irrelevant. The only alternatives are certain projection based approaches which only require the once off computation of basis vectors.

In the case of inverse modelling, we echo previous concerns that despite the ease of use and popularity of data-driven methods, they have well-established limitations and should be used with care. Preference should be given to approaches which have

been validated in the context in which they are to be used. For example the kriging and radial basis functions enabled optimization, or polynomial chaos expansion uncertainty analysis methods mentioned in Section 2.6. Projection based approaches have only recently been applied to the inverse modelling of groundwater models, with promising results for runtime reduction. Doubtless, the benefits and disadvantages of these methods will be exposed more clearly as they are further explored. Multi-fidelity inverse modelling methods (discussed in Section 2.4.3) offer potential for both direct application and future development. Again, they have only recently been applied to groundwater models, but hold great promise. Developing maps from low to high fidelity models remains the biggest hurdle to their widespread application.

Chapter 3

Polynomial Chaos Expansions

3.1 Introduction

Thorough analysis of mathematical models is a prerequisite to their application to system or policy design. Models of hydro-geological systems are typically PDE-based and characterized by a high degree of complexity and uncertainty in the parameterization of hydraulic properties and boundary conditions. Fine grids are required to capture the spatial heterogeneity, making even single model runs computationally expensive. Large samples of model runs are needed to how understand uncertainty in the inputs effects confidence in the predictions. Hence thorough analysis in the form of uncertainty propagation, sensitivity analysis, convergence studies and inversion can be prohibitively expensive.

In this chapter we propose a computationally efficient method to perform uncertainty propagation and sensitivity analysis on a groundwater model using polynomial chaos expansions. We demonstrate the effectiveness of the method given a range of typical input-output relationships. The first output of the study is a computationally efficient estimate of sensitivities and output distributions.

A major limitation of polynomial surrogate methods is the growth in the number of coefficients with an increase in random variables and polynomial order (see Equation (3.6)). The second output of this study is a method of mitigating this effect by fitting nested lower order polynomials, rather than a single higher order polynomial.

3.1.1 Balancing stochastic and spatial resolution

Sensitivity and uncertainty analysis runtimes of groundwater models are affected by model complexity and the number of samples necessary. Models can be systematically simplified [Pachepsky et al., 2006] or emulated [Razavi et al., 2012b; Asher et al., 2015] in many ways. However, the choice of spatial and temporal resolution resolution is often the most pertinent aspect of physically based model complexity. The most obvious way to simplify a model based on a set of discretized differential equations is to reduce the resolution of the discretization (the physical resolution). Likewise, while many uncertainty analysis methods exist [Refsgaard et al., 2007; Matott et al., 2009], their computational cost boils down to the number of model runs they require (or the stochastic resolution). Hence, in most cases there is a direct trade off in accuracy between physical resolution (spatial grid size), and stochastic resolution (the number of samples used to compute sensitivity and uncertainty metrics).

In practice, discretized inputs for groundwater flow models are often produced from continuous conceptual data using graphical user interfaces such as Groundwater Vistas [Rumbaugh and Rumbaugh, 2007], Visual MODFLOW FLEX [Fitzpatrick, 2012], Aquaveo GMS [Aquaveo, 2011] or ModelMuse [Winston, 2009]. Recent work has developed automated tools to extract parameters from geospatially enabled databases and discretize them to the model mesh [e.g. Bhatt et al., 2014]. Where the original conceptual data is available, such methods allow the same conceptual model to be implemented using a range of spatial resolutions.

While this chapter suggests that spatial resolution should be adjusted to minimize associated error given computational constraints, a noteworthy aside is that adjusting the spatial resolution of groundwater models is a non-trivial task. This is particularly true with respect to aquifer properties. Appropriate grid resolution depends on modelled physics, parameterization, and quantities of interest [Ababou et al., 1989; Bower et al., 2005]. Numerous studies [Vermeulen et al., 2006; Neuman and Di Federico, 2003; Wen and Gómez-Hernández, 1996; Mehl and Hill, 2010] have documented the limitations in upscaling groundwater models. Upscaling parameters refers to reducing their spatial resolution, computing their values on a coarse grid

from the corresponding values on a fine grid. Two approaches are used to upscale parameters to a coarser grid: local and non-local techniques. In local techniques, coarse scale values are an explicit function of the fine scale values contained in that cell. Non-local techniques solve the groundwater flow equation for the coarse grid with boundary conditions given by the fine grid. It should be noted though that coarse scale conductivities are not a "material property", but depend on sub-grid level flow conditions [Wen and Gómez-Hernández, 1996]. Hence conductivity parameter values depend on the grid size. Mehl and Hill [2010] find that stream bed conductance parameters (Cauchy boundary conditions) also depend on grid discretization. Although a number of optimal approaches for upscaling transmissivity and boundary conditions exist, none significantly reduce upscaling error in practice [Vermeulen et al., 2006]. Pogson and Smith [2015] provide a method to quantify the spatial resolution's contribution to uncertainty with reference to a fine grid.

Despite these nuances, implicit in the use of process based models is a belief that finer grids are more accurate. Therefore it is desirable to compute the "error" associated with spatial resolution by conducting convergence studies as conducted in this chapter. Previous work has been conducted into analyzing groundwater model convergence as spatial grids are refined [Brooks and Tobias, 1996; Bower et al., 2005; Zyvoloski and Vesselinov, 2006].

It has been noted that while a significant body of literature has developed concerning uncertainty in groundwater models (i.e. stochastic hydrogeology), there remains a significant gap between research and industry [Renard, 2007; Neuman, 2004; Hunt and Doherty, 2011]. Therefore, this chapter focuses on three classes of methods with both established support in the literature and readily available software implementations: uncertainty propagation, sensitivity analysis and sampling based inverse modelling. Polynomial chaos expansions have been used to implement each of these uncertainty quantification methods for groundwater models. The method's flexibility is a key strength: it develops a surrogate of the forward model that can be applied in many problem settings. Ghanem [1998] used polynomial chaos expansions to propagate uncertainty in a groundwater model. Ciriello et al. [2013] employed polynomial chaos expansions for global sensitivity analysis of a groundwater model. Laloy et al.

[2013] apply a spectral projection based polynomial chaos approach to Bayesian inference of a groundwater model. This chapter aims to serve as a more general guide to the application of polynomial chaos surrogates to groundwater modelling. Hence we explore different configurations of flow models, as well as the convergence with respect to spatial and stochastic resolution.

Similarly to the case with physical resolution, the error in sensitivity indices or output distributions due to the number of samples can be computed using convergence studies. Ballio and Guadagnini [2004]; Ernst et al. [2012]; Xiu and Karniadakis [2003] have studied the stochastic convergence of groundwater models. Leube et al. [2013] recently found considerable efficiencies could be made by optimally allocating resources by selecting the number of Monte Carlo samples and spatial resolution for the estimation of statistical moments of groundwater level. However, their brute force approach involved computing the budget and error at each point on a dense grid of spatial and stochastic resolutions, making it difficult to apply in practice. Moslehi et al. [2015] instead used a non-linear programming method, approximating the sampling error by the sample variance and spatial discretization error by a polynomial.

3.1.2 Structure

In this chapter we conduct a convergence study on a practical groundwater model - and then conduct sensitivity analysis using Polynomial Chaos methods, which proves significantly more efficient than Monte Carlo sampling.

3.2 Polynomial Chaos Expansions

Consider the generic model

$$h = F(\boldsymbol{\theta}) \qquad \text{or} \qquad L(\boldsymbol{\theta}, h) = 0 \tag{3.1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_r) \in \Gamma \subset \mathbb{R}^r$ are the random parameters and h is the output of interest. F is the forward operator, which typically cannot be defined analytically. Instead, it is given as the solution to a system of equations L.

Groundwater models are typically based on solving the flow equation (i.e. L)

$$\nabla \cdot (K \nabla h) = S \frac{\partial h}{\partial t} + q \tag{3.2}$$

where $h(x,t)$ is pressure, $K(x)$ hydraulic conductivity, $S(x)$ is specific storage, $q(x,t)$ is a source sink term, $x = (x_1, \cdots, x_d) \in \mathbb{R}^d$ are the spatial coordinates and t is time. The models in this chapter are 3 dimensional so $d = 3$. The inputs h (for previous t), K, S and q will be determined from available data and the random parameters $\boldsymbol{\theta}$.

Polynomial chaos expansions allow efficient approximations of relationships such as (3.1). Cameron and Martin [1947] proved that any process $F(\boldsymbol{\theta})$ of finite variance can be represented as

$$\begin{aligned} F(\boldsymbol{\theta}) = & a_0 \psi_0 \\ & + \sum_{i_1=1}^{\infty} a_{i_1} \psi_{i_1}(\theta_{i_1}) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} \psi_{i_1}(\theta_{i_1}) \psi_{i_2}(\theta_{i_2}) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} \psi_{i_1}(\theta_{i_1}) \psi_{i_2}(\theta_{i_2}) \psi_{i_3}(\theta_{i_3}) \\ & + \cdots, \end{aligned} \tag{3.3}$$

where ψ_i is the order-i Hermite polynomial and θ_i are independent Gaussian random variables. The key property of the multivariate Hermite polynomials in (3.5), is that they satisfy the orthonormality condition with respect to Gaussian random variables

$$\int_{\Gamma} \psi_i(\theta) \psi_j(\theta) \rho(\theta) d\theta = \delta_{i,j} \tag{3.4}$$

where ρ is the Gaussian probability density function and $\delta_{i,j}$ is the Kronecker delta function.

In multiple dimensions, typically tensor product polynomials are used and we

can employ a multi-index to rewrite (3.3) as

$$F(\boldsymbol{\theta}) = \sum_{i=0}^{\infty} \alpha_i \boldsymbol{\Psi}_i(\boldsymbol{\theta}) \tag{3.5}$$

where there is a one-to-one correspondence between the coefficients and functions in (3.3) and (3.5). $\boldsymbol{\Psi}$ is a tensor product of the univariate polynomials.

In practice, for n random variables and maximum polynomial order p, (3.5) is truncated

$$F(\boldsymbol{\theta}) \approx \sum_{i=0}^{P} \alpha_i \boldsymbol{\Psi}_i(\theta_1, \cdots, \theta_n) \tag{3.6}$$

where $P = \binom{n+p}{n} = \frac{(n+p)!}{n!p!}$. Convergence of the Hermite-chaos, (3.6), is optimal if $\boldsymbol{\theta}$ are Gaussian random variables.

The Wiener-Askey scheme of Xiu and Karniadakis [2002] gives the corresponding one dimensional orthogonal polynomial which has a weighting function identical to that distribution's probability density function. Functions with random inputs of normal, uniform, beta, exponential and gamma distributions can be optimally approximated by expansions in Hermite, Legendre, Jacobi, Laguerre and Generalized Laguerre polynomials respectively. Algorithms do exist for computing orthonormal one dimensional polynomials for general distributions [Liu and Narayan, 2021].

Of practical note too are algorithms [e.g. Jakeman et al., 2019b] which allow construction of PCE where the variables are dependent.

Once calibrated, a polynomial chaos expansion (3.6) can be used as a surrogate of a function (3.1). The expansion is a simple analytic expression so can be evaluated quickly. Furthermore, statistical moments and sensitivity indices can be computed directly from the coefficients.

A number of approaches exist for calibrating a polynomial chaos expansion, ie. finding α_i of (3.6), for a given model. In the following sections we explain several of the most common.

3.2.1 Stochastic Galerkin

In a similar fashion to other projection based surrogates [McPhee and Yeh, 2008], the stochastic Galerkin method solves for the coefficients in an intrusive manner. The output h is written as a truncated polynomial chaos expansion, such as (3.6) to give

$$h \approx \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta}) \tag{3.7}$$

and substituted into the governing equations (3.1)

$$L(\boldsymbol{\theta}, \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta})) = 0. \tag{3.8}$$

Similar to the weak form in the finite element method, Galerkin projection onto the basis polynomials $\boldsymbol{\Psi}_i$, using the same inner product as the orthogonality condition (3.4), is then used to form a set of $P+1$ equations

$$\int_{\Gamma} L(\boldsymbol{\theta}, \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta})) \boldsymbol{\Psi}_j(\boldsymbol{\theta}) \rho(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0, \qquad \forall j = 0, \cdots, P. \tag{3.9}$$

These can be solved for the coefficients, $\widehat{h}_i$, using typical numerical techniques, similar but not identical to the solver used for the original model defined by (3.1).

Following Xiu [2009], consider the example of the stochastic Galerkin solution to groundwater flow (from (3.2))

$$\frac{\partial h(x,t,\boldsymbol{\theta})}{\partial t} = \nabla_x \cdot (k(x,\boldsymbol{\theta}) \nabla_x h(x,t,\boldsymbol{\theta})) + f(x,t,\boldsymbol{\theta}), \qquad x \in \mathbb{R}^d, t \in (0,T] \tag{3.10}$$

$$h(x,0,\boldsymbol{\theta}) = h_0(x,\boldsymbol{\theta}), \qquad h(\cdot,t,\boldsymbol{\theta})|_{\partial D} = 0 \tag{3.11}$$

Assume the conductivity field k can be parameterized using the Karhunen-Loève expansion with n random terms

$$k(x,y) = \sum_{i=0}^{n} \widehat{k}_i(x) \theta_i \tag{3.12}$$

where $\theta_0 = 1$ and $k(x,\boldsymbol{\theta}) \geq k_{\min} > 0$, $\forall x, \boldsymbol{\theta}$. Further assume the head will have finite

variance, so can be approximated by

$$h(x,t,\boldsymbol{\theta}) \approx \sum_{i=0}^{P} \widehat{h}_i(x,t)\boldsymbol{\Psi}_i(\boldsymbol{\theta}). \tag{3.13}$$

Substituting (3.12) and (3.13) into the governing equations (3.10), then taking the inner product of the equation with each basis to project them onto the random space spanned by the basis polynomials $\boldsymbol{\Psi}_i$ yields

$$\frac{\partial \widehat{h}_k}{\partial t}(x,t) = \sum_{i=0}^{n}\sum_{j=0}^{P} \nabla_x \cdot \left(\widehat{k}_i(x)\nabla_x \widehat{h}_j(x,t)\right) e_{ijk} + f(x,t), \qquad \forall k = 0,\cdots,P \tag{3.14}$$

where

$$e_{ijk} = \mathbb{E}(\theta_i \boldsymbol{\Psi}_j \boldsymbol{\Psi}_k) = \int_\Gamma \theta_i \boldsymbol{\Psi}_j(\boldsymbol{\theta})\boldsymbol{\Psi}_k(\boldsymbol{\theta})\rho(\boldsymbol{\theta})d\boldsymbol{\theta}, \qquad i \in (0,n), 0 \le j,k \le P, \tag{3.15}$$

Simplifying (3.14),

$$\frac{\partial \widehat{h}_k}{\partial t}(x,t) = \sum_{j=0}^{P} \nabla_x \cdot \left((\sum_{i=0}^{n} \widehat{k}_i(x) e_{ijk}) \nabla_x \widehat{h}_j(x,t) \right) + f(x,t), \qquad \forall k = 0,\cdots,P \tag{3.16}$$

which can be solved as a system of $P+1$ diffusion equations.

3.2.2 Pseudo-Spectral Projection

Pseudo-spectral methods are a type of collocation approach with a long history of approximating deterministic functions with orthonormal polynomials [e.g. Gheorghiu, 2007]. More recently, they have been applied to stochastic equations.

Consider the expansion from (3.6)

$$F(\boldsymbol{\theta}) \approx \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta}). \tag{3.17}$$

As with Fourier series, the optimal coefficients can be found using the projection

$$\widehat{h}_i = \int_\Gamma F(\boldsymbol{\theta})\Psi_i(\boldsymbol{\theta})\rho(\boldsymbol{\theta})d\boldsymbol{\theta}, \qquad i = 1,\cdots,P$$

(3.18)

which ensures the error of the projection is orthogonal to the span of the polynomial basis [Le Maitre et al., 2002]. The stochastic integral in (3.2.2) is typically approximated using quadrature rules which select Q nodes and weights $\{(\boldsymbol{\theta}^{(j)}, w^{(j)})\}_{j=1}^{Q}$, obtain the corresponding solutions $\{h^{(j)} = F(\boldsymbol{\theta}^{(j)})\}_{j=1}^{Q}$ and compute the coefficients [e.g. Eldred and Burkardt, 2009]

$$\widehat{h}_i = \sum_{j=1}^{Q} h^{(j)} \Psi_i(\boldsymbol{\theta}^{(j)}) w^{(j)}, \qquad i = 1, \cdots, P \tag{3.19}$$

3.2.3 Regression

A polynomial chaos expansion can be also calibrated in a non-intrusive manner using regression [Isukapalli, 1999]. While the pseudo-spectral projection approach approximates the integral which would minimize the L^2 expansion error for a given set of Q sample points $\{\boldsymbol{\theta}^{(j)}\}$, least squares regression minimizes the discrete error

$$\|F(\boldsymbol{\theta}) - \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta})\|^2 = \frac{1}{Q} \sum_{j=1}^{Q} \left(F(\boldsymbol{\theta}^{(j)}) - \sum_{i=0}^{P} \widehat{h}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta}^{(j)}) \right)^2. \tag{3.20}$$

At set of Q sample points, a polynomial chaos expansion (3.6) can be written in matrix form

$$\mathbf{h} = \mathbf{P}\mathbf{c} \tag{3.21}$$

where $\mathbf{P}$ is the $Q \times (P+1)$ "Vandermonde" matrix with elements defined

$$\mathbf{P}_{ij} = \boldsymbol{\Psi}_j(\boldsymbol{\theta}^{(i)}), \tag{3.22}$$

$\mathbf{h}$ is made up of solutions of the original model

$$\mathbf{h} = \left(F(\boldsymbol{\theta}^{(1)}), \cdots, F(\boldsymbol{\theta}^{(Q)}) \right)^T \tag{3.23}$$

and the coefficient array

$$\mathbf{c} = \left(\widehat{h}_0, \cdots, \widehat{h}_P\right)^T \tag{3.24}$$

is computed as the least squares solution:

$$\mathbf{c} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{h}. \tag{3.25}$$

The sample points can be chosen in any way that leads to an invertible matrix $(\mathbf{P}^T\mathbf{P})$. Winokur [2015] review other methods, such as hybrid least-angle regression or compressed sensing which may be used to solve (3.24) while taking advantage of the sparsity of the expansions.

3.2.4 Sparse Grids

Another method of fitting a PCE is by first using sparse grid [e.g. Barthelmann et al., 2000; Hegland, 2003] interpolation and then transforming these into a PCE [Jakeman et al., 2019a]. Sparse grids are a method of stochastic approximation of multivariate functions at a fraction of the computational cost of a full tensor product interpolation. Consider the function $F_\alpha(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the inputs of interest, and α parameterizes the coarseness of the physical discretization. A full tensor product interpolation, at $n_\beta = d$ points, is a weighted sum

$$F_{\alpha,\beta}(\boldsymbol{\theta}) = \sum_{j \leq \beta} F_\alpha(\theta^{(j)}) \prod_{i=1}^{d} l_{i,j}(\theta_i) \tag{3.26}$$

where the weights are tensor products of univariate Lagrange polynomials

$$l_{i,j}(\theta_i) = \prod_{k=1, k\neq j}^{m_{\beta_i}} \frac{\theta_i - \theta_i^{(k)}}{\theta_i^{(j)} - \theta_i^{(k)}}, \quad i \in \{1, \ldots, d\} \tag{3.27}$$

for the set of points $\theta_i^{(j)}, j \in \{1, \ldots, m_{\beta_i}\}$;

To fit the interpolant, F_α must be evaluated on $M_\beta = \prod_{i=1}^{d} m_{\beta_i}$ points, written

$$\bigotimes_{i=1}^{d} {}^{i}_{\beta_i} = \left[\boldsymbol{\theta}^{(1)} \quad \dots \quad \boldsymbol{\theta}^{(M_\beta)} \right] \in \mathbb{R}^{d \times M_\beta}. \tag{3.28}$$

Nested Clenshaw-Curtis points are often used for uniform variables to define the univariate Lagrange polynomials

$$\theta_i^{(j)} = \cos\left(\frac{(j-1)\pi}{m_{\beta_i} - 1} \right). \tag{3.29}$$

In tensor product interpolation the number of points grows exponentially with the number of random variables d. Sparse grids on the other hand use a weighted sum of low-resolution tensor-products [Barthelmann et al., 2000]

$$F_{\alpha,\mathcal{I}}(\boldsymbol{\theta}) = \sum_{\beta \in \mathcal{I}} c_\beta F_{\alpha,\beta}(\boldsymbol{\theta}) \tag{3.30}$$

with weights c_β given

$$c_\beta = \sum_{j \in \otimes_{i=1}^{d} \{0,1\}} (-1)^{|j|_1} \chi_{\mathcal{I}}(\beta + j), \qquad \chi_{\mathcal{I}}(k) = \begin{cases} 1 & k \in \mathcal{I} \\ 0 & \text{otherwise.} \end{cases} \tag{3.31}$$

The points of a sparse grid are made up of union of tensor products $\mathcal{Z}_{\mathcal{I}} = \bigcup_{\beta \in \mathcal{I}} \mathcal{Z}_\beta$. A common implemenation is the Clenshaw-Curtis isotropic sparse grid where the total number of points in the grid is given

$$M_{(l)} \approx \frac{2^l}{l!} d^l,$$

versus the larger number of points in the full tensor product

$$M_l = m_l^d = \left(2^l + 1\right)^d \qquad\qquad l = (l, \dots, l).$$

The sparse grid approximation (based on Lagrange polynomials l) given in Equa-

tion 3.30 can now be transformed to a PCE [Jakeman et al., 2019a] (based on orthonormal polynomials ϕ). For each tensor-product interpolant (Equation 3.26) that makes up the sparse grid, we can write the Lagrange basis function $l_{i,j}$ as a PCE

$$l_{i,j}(\theta_i) = \sum_{k=1}^{m_{\beta_i}} \alpha_k \psi_k(\theta_i) \qquad j = 1, \cdots, m_{\beta_i}, i = 1, \cdots, d. \tag{3.32}$$

The coefficients α_k for the orthonormal polynomial can be derived using Gaussian quadrature.

3.2.5 Statistical Information

Once we have a calibrated polynomial chaos expansion (3.6) from one of the above methods, various important statistical metrics can be computed analytically using the surrogate. Typically this is several orders of magnitude faster than the alternative, which involves solving the underlying differential equations on a random sample of inputs.

A common method for characterizing the relative importance of a number of inputs and parameters to model outputs is variance based global sensitivity analysis. Two established methods are the Sobol [Sobol, 2001] and the Fourier Amplitude Sensitivity Test [Cukier et al., 1973]. Recently, Sudret [2008] observed that the same sensitivity indices can be computed efficiently from polynomial chaos expansions.

These metrics can be computed as follows.

Mean

$$\mathbb{E}(F) = \int \left(\sum_{i=0}^{P} \alpha_i \mathbf{\Psi}_i(\theta_1, \cdots, \theta_n) \right) \rho(\theta) d\theta = \alpha_0$$

Variance

$$\begin{aligned}\mathrm{Var}(F) &= \mathbb{E}\left((F-\mathbb{E}(F))^2\right)\\ &= \sum_{i=1}^{P}\alpha_i^2\mathbb{E}\left(\mathbf{\Psi}_i^2(\theta_1,\cdots,\theta_n)\right)\\ &= \sum_{i=1}^{P}\alpha_i^2\end{aligned}$$

Sobol indices

$$S_{(i_1,\cdots,i_s)} = \frac{\sum_{i\in\mathcal{J}}\alpha_i^2\mathbb{E}\left(\mathbf{\Psi}_i^2(\theta_1,\cdots,\theta_N)\right)}{\mathrm{Var}(F)}$$

where $\mathcal{J}$ is the set of multi-indices with only non-zero entries in the positions $(i_1,\cdots,i_s)$.

Main effect

The contribution to variance from a single variable.

$$S_{(i_1)} = \frac{\alpha_{i_1}^2\mathbb{E}\left(\mathbf{\Psi}_{i_1}^2(\theta_1,\cdots,\theta_N)\right)}{\mathrm{Var}(F)} \tag{3.33}$$

Total effect

The sensitivity of the variance due to the single variable and its interactions.

$$S_{(i_1)}^T = \sum_{(j_1,\cdots,j_s)\in\mathcal{T}} S_{(j_1,\cdots,j_s)} \tag{3.34}$$

where $\mathcal{T} = \{(j_1,\cdots,j_s): \quad i_1\in(j_1,\cdots,j_s)\}$.

3.3 Application of PCE to a Groundwater flow model

In this section we demonstrate the effectiveness of a PCE surrogate for the uncertainty analysis of a groundwater flow model. The purpose of the model is to be used as a component in an integrated model like that of Iwanaga et al. [2018]. They

combine a similar groundwater model with surface water, ecology, economic and policy components. It is noted that the groundwater model's computational expense is a bottleneck in iterative and participatory modeling. Surrogate methods, such as PCE, allow such models to be run quickly without a significant reduction in model output.

As an example of the application of PCE to a groundwater model, a small (20x40km) part of the Peel River catchment (Northern NSW, Australia) is used, with the PCE surrogate was trained to emulated the response of water pressure (head) at four locations (see Figure 3.1) to 12 uniformly distributed parameters, including the four uncertain parameters in Table 3.1.

Name	Description	Range
`hk KLE mode i`	The ith modes of Karhunen-Loève parameterization of the conductivity.	$[0, 4]$ meters
`stage height i`	The river stage height during time period i	$[0, 4]$ meters
`well rate i`	The flow out of the ith well	$[20, 50]$ gigaliters per year
`rch intensity`	Effective rainfall (amount of rainfall that reaches the water table)	$[10, 50]$ mm per year

Table 3.1: The uncertain inputs for the groundwater flow model are shown. All were given uniform distributions.

To test the flexibility of the approach we created a model with a variety of common inputs and outputs. These include examples of two types of boundary conditions in groundwater flow models, namely specified flux boundary (Neumann) in `well rate i` and head-dependent flux boundaries (Robin or mixed) in `stage height i`. Specified head boundaries (Dirichlet) were not included in this particular analysis, but behaved similarly to head-dependent flux boundaries in early analysis. This range was selected to test the application of the approach in a variety of modelling contexts.

Using a common method [e.g. Ghanem and Dham, 1998] we create Karhunen-Loève parameterization of conductivity. By assuming several statistical properties of the conductivity (mean, standard deviation and correlation length of the covariance

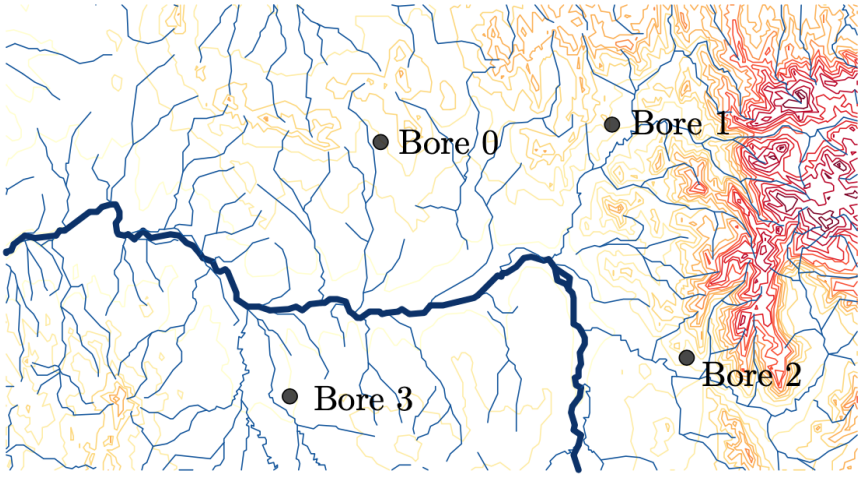
kernel), we can write a given conductivity field (2D image) as a weighted sum of basis fields. We can now parameterize uncertainty in the conductivity field by treating the weights (`hk KLE mode i`) as random variables.

The data for the study site is from publicly available, continental scale datasets [Carrara et al., 2017]. We develop an automated conceptualization workflow from continential scale data sets to MODFLOW [Mehl and Hill, 2005] input files as shown in Figure 3.1, so as to allow multiple models of different spatial resolutions. The worflow was built using utility libraries contributors [2021] and Bakker et al. [2016] to perform geospatial operations and model input file formatting respectively.

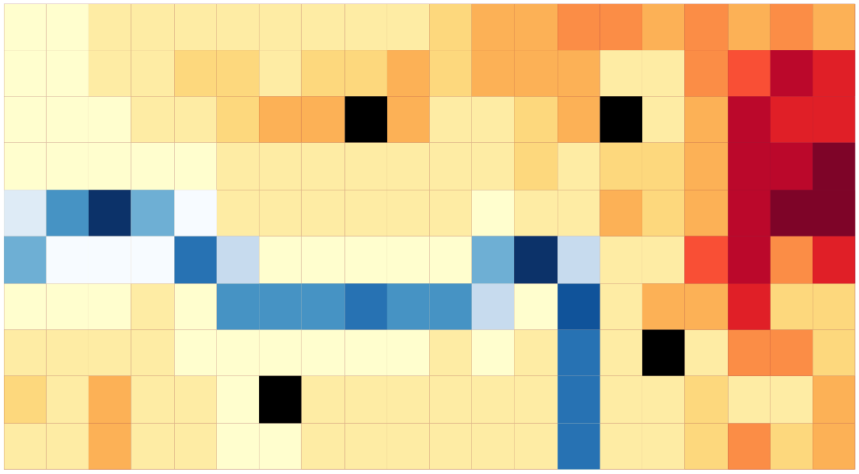
3.3.1 Surrogate error and runtime reduction

The aim was to develop an efficient, yet comprehensive understanding of the input-output relationships of the model for a range of inputs at the four selected bore sites. Following the adaptive algorithm of Narayan and Jakeman [2014]; Jakeman et al. [2019a] we considered error introduced by spatial discretization alongside the stochastic error. Figure 3.2 shows the convergence in surrogate model error with the number of complex model runs used to train the surrogate (i.e. stochastic resolution). Results are plotted for three spatial resolutions. The error is the RMSE with respect to the complex model at the highest spatial resolution. It is of note that the adaptive sparse grid surrogate here converges to less than 1% error in less than 200 runs of a moderately complex groundwater model with a range of common inputs and outputs. Note the error on this plot is RMSE in units of meters, the percentage error is an order of magnitude lower. As seen in Figure 3.3, this convergence holds for the statistical outputs of the model. Here the error reported is averaged across the four locations. With a validation set of 1000 samples of the uncertain parameters (see Table 3.1), we compare the PCE estimated mean with the error of the Monte Carlo (MC) estimate using the same number of complex model runs. This shows the PCE estimate of the mean to be three orders of magnitude better than the MC estimate using 100 complex model runs.

Surrogate and complex model predictions are plotted against one another in



(a) Vector



(b) Raster

Figure 3.1: Examples of the geospatial data available for the Peel river catchment in its raw vector and processed raster form. Bores (black), rivers (blue) and elevation (reds) ar shown.

Figure 3.4 using the surrogate trained on 100 complex model runs. Note that the depth of bore 3 is much less (6m) compared to the other bores (25-60 m), which explains the different RMSE seen in Figure 3.2. A major finding is that a surrogate of this accuracy has a one time offline cost of 100 complex model runs, after which predictions can be made at almost negligible cost as seen by the online surrogate runtime ratios given in Table 3.2. Note that the complex model runtime includes not only expensive computation, but a significant amount of I/O (reading and writing

Model Size (number of finite difference cells)	Ratio of Complex/Surrogate Online Runtime
10x20	115,158
20x40	470,293
40x80	2,219,158

Table 3.2: Computational requirements of the adaptive sparse grid calibrated PCE surrogate. These ratios are very high because the surrogate execution takes a small fraction of a second.

files). The offline time to calibrate a PCE is dominated by the necessary complex runs. Fitting the coefficients of the PCE once the data is generated is computationally negligble.

As described in Section 3.2.5, this PCE surrogate can be used to compute sensitivity indices [Jakeman et al., 2019a]. These are plotted in Figures 3.5(a) and 3.5(b). These plots show the values of the Sobol' main effect for different stochastic and spatial resolutions respectively. The main effect indicates the contribution to variance of that input, excluding interactions. If all the main effects sum to one then the model is additive (i.e. there are few interactions). The magnitude is not important here, but the relative height of the curves shows the relative importance of that variable according to the surrogate trained at that point. The horizontal axis shows the change as more full model runs are added to the surrogate training set. Note that low variations of the curves in Figure 3.5(a) indicate the PCE surrogate is able to accurately estimate these indices based on a few (< 200) model runs; i.e. by the first point on the plot. For two of the locations in Figure 3.5(b), the relative importance of `well rate i` and the `hk KLE i` parameters shift as spatial resolution is increased. As resolution increases boundary conditions change from covering one to multiple cells, and from occurring in the same cell as an input to one nearby. As discussed above, there are few techniques for thoroughly treating these kinds of grid scale changes. We conclude here that given the rapid convergence of the surrogate with stochastic resolution, many modellers would benefit from an increased focus (and computational investment) in quantifying uncertainty due to physical resolution.

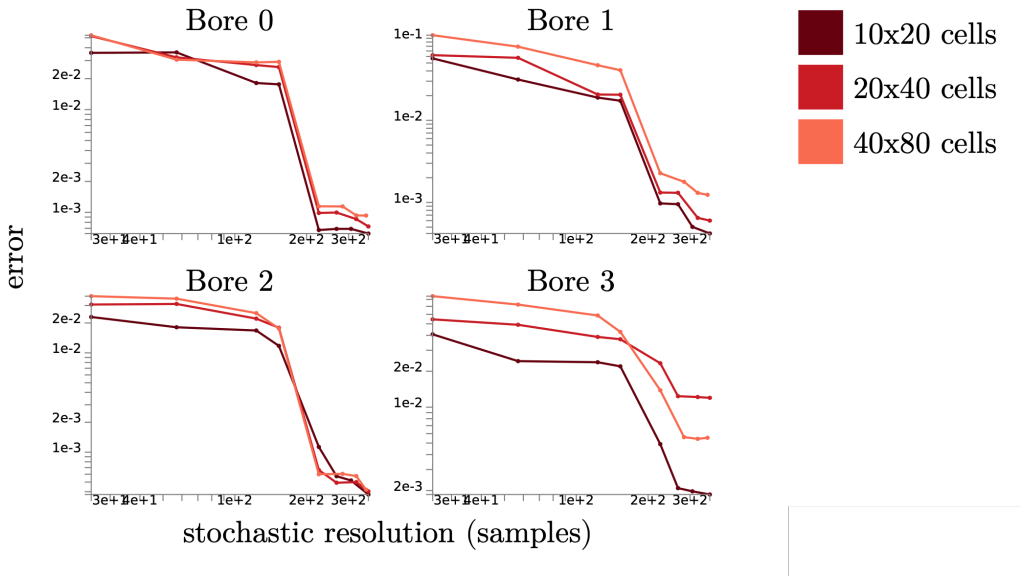


Figure 3.2: Convergence of the RMSE error (vertical axis) with number of complex model runs (horizontal axis) at four locations. Results are shown for three spatial resolutions.

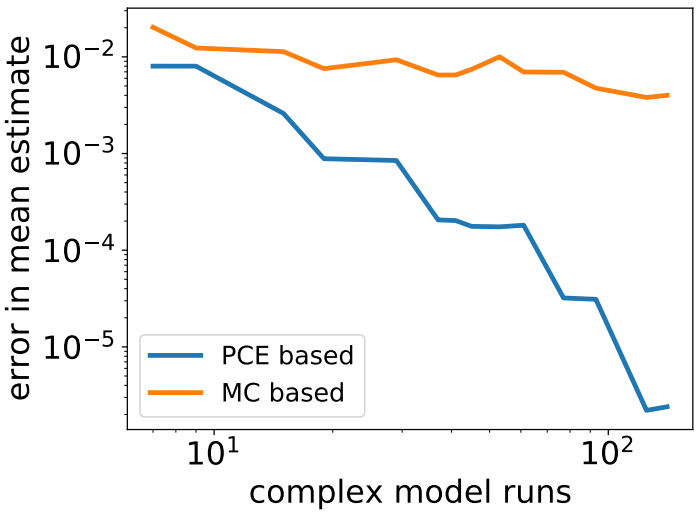


Figure 3.3: Error in the mean estimate as computed by the PCE surrogate for a given number of complex model runs and the highest physical resolution. This is compared with the error of the Monte Carlo (MC) estimate using the same number of complex model runs. Error is the relative RMSE averaged across the four locations.

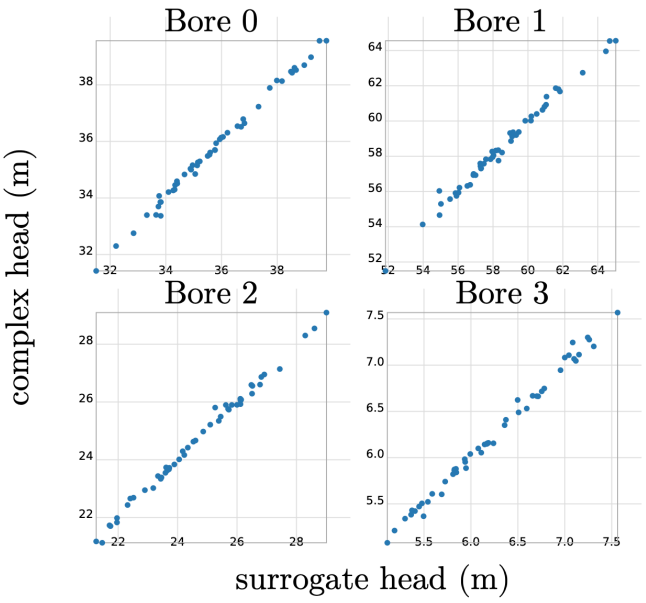
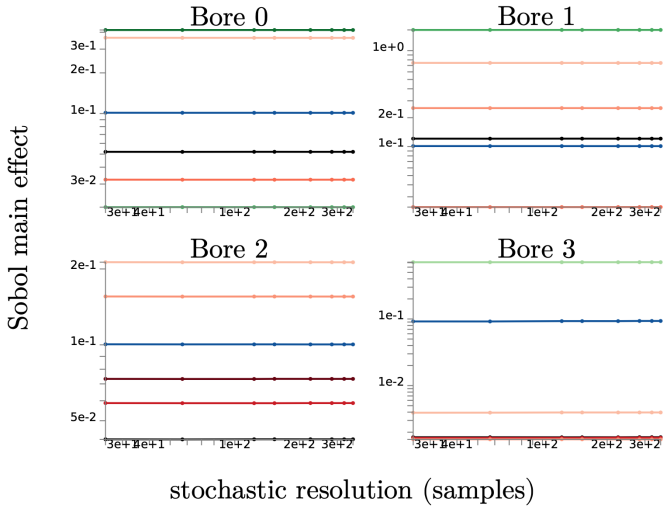
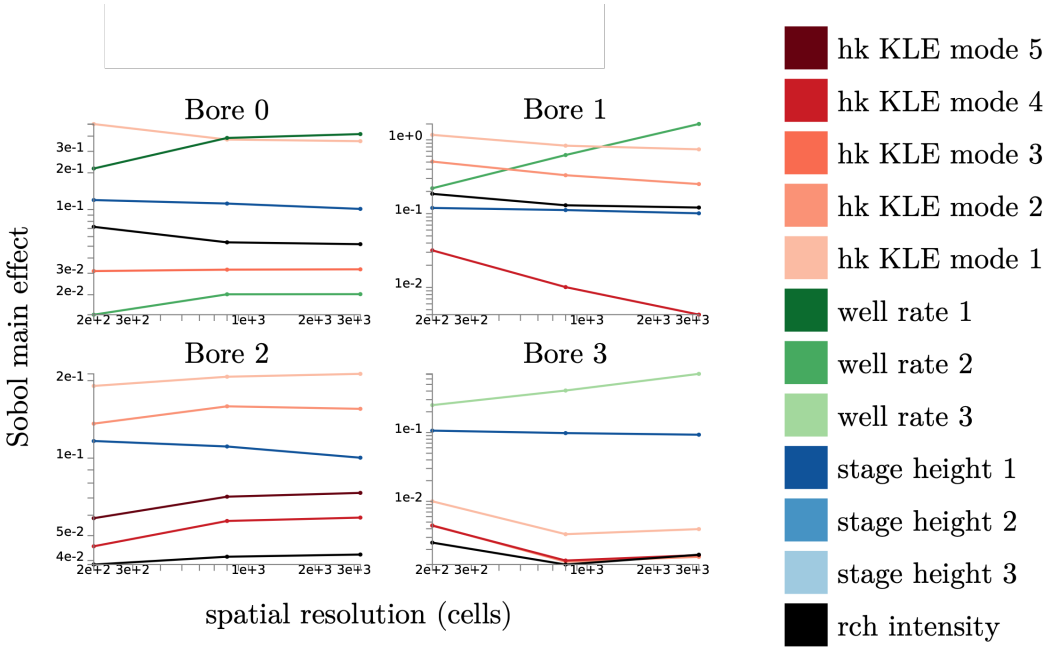


Figure 3.4: Scatter plots of complex v surrogate computed head at four locations.



(a) Convergence of the Sobol' main effect sensitivity index with number of complex model runs. The "boring" flat curves indicate the PCE surrogate is able to accurately estimate these indices based on a few (< 100) model runs; i.e. by the first point on the plot. The colours of the curves match the legend in the below sub figure.



(b) Convergence of the Sobol' main effect sensitivity index with the number of points in the spatial discretization.

Figure 3.5: Sensitivity indices.

3.4 Nested Polynomials

As noted above, a PCE surrogate of n random variables and maximum polynomial order p will have $\frac{(n+p)!}{n!p!}$ unknown coefficients. The principal limitation of the PCE method, as with many surrogates, is the curse of dimensionality. The rapid growth in polynomial order (and hence computational requirements) for the large n necessary to parameterize uncertainty in many practical problems. The sparse grid method used in Section 3.3 are able to delay growth in computational requirement compared to full tensor product interpolation, and have been demonstrated on larger numbers of random variables (e.g. 40 in Narayan and Jakeman [2014]).

However, there is still a need for algorithms which deal with greater numbers of random parameters. In this section we explore the potential of nested polynomials to reduce the dimensionality of the same groundwater problem discussed in Section 3.3. While nested lower order polynomials cannot emulate the full flexibility of a single polynomial with the same total order, they have important advantages. First, they have fewer terms and hence require less training data and computational resources. Second, the nested (i.e. "deep") structure is uniquely suited to implementation in the deep learning software packages discussed in Chapter 5. While the underlying reasons are not yet fully understood, there is significant empirical evidence [Lin et al., 2017] that nested structures outperform flat functions in many modelling contexts.

Setup

In this section we deal with employ two types of polynomials. The first few (probabilists') Hermite polynomials are given below along with the recursion formula used

to compute higher orders [Chihara, 1978]

$$\begin{aligned} H^0(x) &= 1 \\ H^1(x) &= x \\ H^2(x) &= x^2 - 1 \\ H^3(x) &= x^3 - 3x \\ H^4(x) &= x^4 - 6x^2 + 3 \\ H^5(x) &= x^5 - 10x^3 + 15x \\ &\cdots \\ H^{n+1}(x) &= xH^n(x) - nH^{n-1}(x). \end{aligned} \tag{3.35}$$

Monomials are simply written

$$\begin{aligned} M^0(x) &= 1 \\ M^1(x) &= x \\ M^2(x) &= x^2 \\ M^3(x) &= x^3 \\ &\cdots \\ M^n(x) &= x^n. \end{aligned} \tag{3.36}$$

To implement our nested model, we define a "layer" function f^p as the expansion

$$f^p(x) = \sum_{i \in 1,\ldots,p} \alpha_i P^i(x) \tag{3.37}$$

where P^i are either the Hermite polynomials (H) or monomials (M) described above, p is the order of the layer, θ the input and α_i the coefficients which need to be calibrated. We will denote 3.37 as either H^p or M^p depending on the polynomial used. Layers can be composed, e.g. $\mathbf{M}^2 \cdot \mathbf{M}^2$, where the output of the first 3.37 becomes the input x for the second.

Multiple random variables n are introduced by defining e.g.

$$\mathbf{H}_n^p = f^p(x) = \sum_{i \in 1,\ldots,p} \alpha_i \left(\prod_{j \in 1,\ldots,n} H^i(x_j) \right) \tag{3.38}$$

Multiple outputs are implemented in parallel with a layer 3.38 for each.

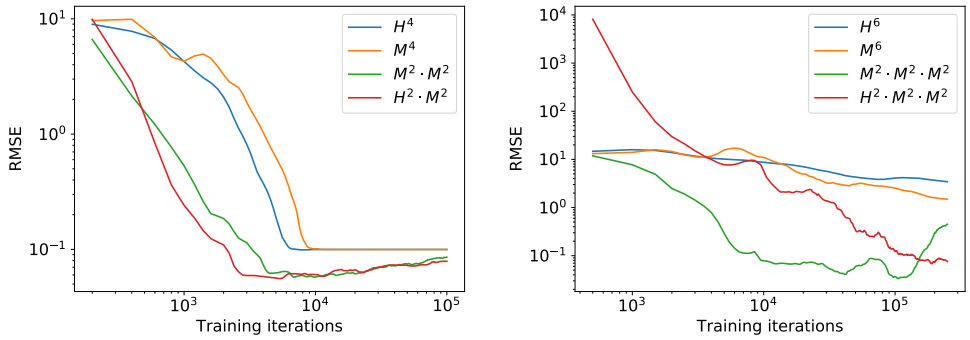
Demonstration

As noted above, nested polynomials have fewer terms than a single polynomial of the same order. For example, a polynomial $\mathbf{M}_5^6$ of order $p = 6$ and $n = 5$ random variables has 462 terms. A composition of lower order polynomials $\mathbf{M}_5^2 \cdot \mathbf{M}_5^2 \cdot \mathbf{M}_1^2$ has 231. While the latter will have the same total order, it evidently cannot emulate the full range of functions spanned by the larger expression. The question is whether nested polynomials are more efficient in practice, where the coefficients of surrogate polynomials are typically sparse.

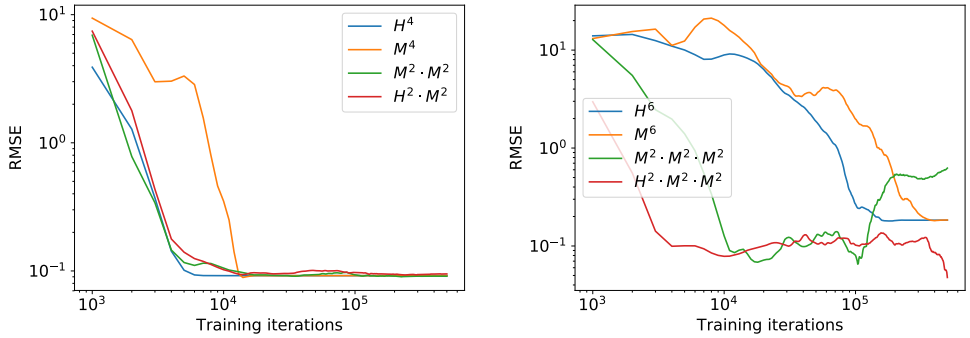
Since libraries developed for deep learning (see Chapter 5) applications are based on similar nested structures, we implemented a nested PCE framework using the deep learning library Pytorch [PyTorch]. We built a number of surrogates using layers like Equation 3.38 of different depths, and calibrated them on data from the same groundwater model as the previous section. To simplify this analysis a single output was considered, the water pressure at a particular location and time. The inputs were re-sampled using normal distributions, with the same center.

As shown in Figure 3.6, the nested structure results in a more accurate surrogate in each of the scenarios we tested. The plots show the convergence of the surrogate error with optimizer iterations (as additional training data is used in the calibration), for a mix of polynomial structures and training data sizes. In most cases the nested structure also converges more quickly. Optimization was done via limited memory BFGS (acronym for original author's names) algorithm of Byrd et al. [1995]. Performance was similar with the more popular Adams optimization routine of Kingma and Ba [2014].

Note that each training iteration in Figure 3.6 introduces new data (constrained by the total number of samples in the training set), so the horizontal axis is partly



(a) 200 model runs in training set



(b) 1000 model runs in training set

Figure 3.6: PCE surrogates of `mid head mean`, implemented as deep learning models. H are Hermite Polynomials and M are standard Monomials. The exponent indicates the order of the polynomial.

a proxy for how many complex model runs are used. This is an artefact of the optimization implementation in Pytorch [PyTorch], which is designed to handle parallel training of large datasets (too big to fit in memory) on high performance computers. A more thorough future investigation into precise convergence over multiple optimization runs for a given number of complex model runs is beyond the scope of this work. The architectural choices made here: the optimization routine, how to combine multiple inputs and outputs, the number of layers and polynomial choice are somewhat arbitrary and warrant further investigation. From this early demonstration, there are two promising findings. First, it shows nested polynomials, implemented in a state of the art deep learning framework, can outperform flat structures. Second, it shows the potential of the PCE theory [Xiu and Karniadakis,

2002] in this practical setting. As expected, the Hermite polynomials converged significantly faster than monomials for this problem where the inputs were normally distributed. Further research is required to systematically understand the impact of each architectural consideration on surrogate accuracy and efficiency.

3.5 Discussion and Conclusions

There are two main results from this chapter. The first relates to the application of existing PCE approaches to groundwater modelling. The second explores a novel direction PCE research could take.

3.5.1 PCE of a Groundwater Model

In this paper we demonstrate the application of a PCE surrogate to the analysis of a groundwater model. The method allows very efficient uncertainty propagation and sensitivity analysis. The adaptive sparse grid algorithm [Jakeman et al., 2019a] trains a PCE in several hundred model runs which captures crucial parameter interactions and emulates the model to a fidelity which is sufficient for many applications. While accuracy requirements will vary by application, this approach finds the point at which the surrogate induced error will be outweighed by spatial resolution errors. We propose that the method can be used in a thorough analysis framework by studying convergence in the error of PCE and sensitivity indices with an increase in stochastic and spatial resolution. Such analysis gives an understanding of model complexity, interactions and uncertainties. While highly non-linear relationships would make the method impracticable by requiring high order polynomials and therefore a large number of training runs, such relationships are not common in many groundwater modeling applications.

PCE software implementations are widely available [Eldred et al., 2013; Feinberg and Langtangen, 2015; Jakeman, 2020b] and do not require significant expertise to train. Therefore we recommend the technique as among the most valuable surrogate based tools for the efficient analysis of groundwater models.

PCE can provide an easy to implement initial analysis of a model. It could aid

in selection of model features and structure during early stages of the modelling process [Jakeman et al., 2006]. It might also be incorporated into a number of the well established frameworks which exist for inverse modelling. For example, the method of Doherty [2009]; Doherty and Hunt [2010], which is based on Truncated Singular Value Decomposition, Tikhonov regularization and Gauss-Marquardt-Levenberg parameter-estimation. A PCE could be used to compute the sensitivities necessary for the the Null-Space Monte Carlo [Tonkin and Doherty, 2009] method employed. Similarly, the ad-hoc surrogate used by Keating et al. [2010] could be replaced by a more generalizable PCE. PCE surrogates have also been used within another well developed Bayesian setting, Markov Chain Monte Carlo methods [Laloy et al., 2013], to accelerate uncertainty quantification.

The principal drawback of the adaptive approach employed in this paper is difficulty dealing with fragile full models. Future work could focus on robust algorithms able to continue past iterations which fail when the underlying model cannot process the sampled inputs. A key difficulty remains in performing computationally efficient convergence studies, necessary to understanding how physical and stochastic resolution impact error.

3.5.2 New Directions

In Section 3.4 we demonstrated how a composition of lower order polynomial chaos expansions can outperform a single, higher order expansion. Specifically they converged faster (with fewer complex model runs in the training data) and did not converge with significantly greater errors. This is of note because these nested structures have fewer parameters than flat structures of the same order. The experiment was implemented in a deep learning framework because highly nested (deep) models are common in this domain. Herein lies another significant finding; to our knowledge this is the first practical step (albeit a minor one) in combining PCE and deep learning methods. Shahane et al. [2019] have contrasted the two approaches, but we posit that a more fruitful area of research will be their combination. The simple benefit observed in this chapter (in Section 3.4) is that matching the polynomial with

the input distribution (in line with PCE theory) leads to a surrogate which can be calibrated on fewer data points.

As discussed in Chapter 5, deep learning is a rapidly expanding field of research and practice, and there are many opportunities for cross pollination between mathematical techniques with solid theoretical underpinnings, such as PCE, and state of the art software methods developed by this new community. On the other hand, deep learning research is highly empirical. Model structures, objective functions and optimization algorithms are developed based on performance on real world data. A deeper analytical understanding would doubtless yield practical outcomes.

Recently, Cheng et al. [2018] claimed that the neural networks typically used in deep learning are "essentially" polynomials. However, there has been little knowledge transfer from applied mathematics to deep learning experiments. For example, to our knowledge, PCE theory that informs the choice of polynomials based on input distributions has not been applied to that area of research. In Chapter 5 we discuss further potential collaborations between the two fields.

Chapter 4

Multi-fidelity Stochastic Collocation

4.1 Introduction

Surrogate methods are a valuable tool for reducing complexity and increasing the efficiency of numerical models. Consider the generic model

$$\mathbf{u} = u(\mathbf{z}), \tag{4.1}$$

which takes a number of inputs $\mathbf{z} = (z_1, z_2, \cdots z_D)$ and produces a number of outputs $\mathbf{u} = (u_1, u_2, \cdots u_Q)$. Models may be run in many settings, for prediction, uncertainty quantification, sensitivity analysis, or optimization. Here the inputs are the quantities to be varied in a particular setting. Similarly, the outputs are only the quantities of interest for a particular application. All other model coefficients, parameters, data, inputs and outputs which are not to be varied for the specific application at hand are not included in $\mathbf{u}$ or $\mathbf{z}$. A complex implementation of u, such as a partial differential equation solver with fine resolution, allows a variety of data and processes to be included at a many temporal and spatial scales and detailed predictions to be made. However, long runtimes limit the use of such high-fidelity models in applications which require real-time results, such as decision support, or many model runs, such as optimization, uncertainty analysis or integrated modelling.

Surrogate models aim to approximate the input-output relationship of u at a fraction of the computational cost. Three broad categories of surrogates exist. Data-driven

methods calibrate an empirical approximation on at set of inputs, $\{\mathbf{z}_n\}_{n\in(1,2,\cdots,N)}$ and corresponding outputs $\{\mathbf{u}_n\}_{n\in(1,2,\cdots,N)}$. Commonly used examples are Polynomial Chaos Expansions [Xiu and Karniadakis, 2002] and Radial Basis Functions [Powell, 1987]. In projection based methods the governing equations of the complex model u are projected onto a basis of orthogonal vectors. Examples include Proper Orthogonal Decomposition [Sirovich, 1987] and Reduced Basis Methods [Fox and Miura, 1971]. Multi-fidelity or hierarchical surrogates are constructed by simplifying the underlying physics of a model or reducing numerical resolution to create a low-fidelity surrogate [e.g. Doherty and Christensen, 2011]. Many multi-fidelity approaches, for example Multi-scale Finite Element methods [Efendiev and Hou, 2007], Multilevel Monte Carlo Giles [2008] or Sparse Grid Combination Techniques Harding and Hegland [2014] prescribe iterative procedures for combining models at various resolutions.

Which surrogate methods are most promising depends, of course, on the context. [Asher et al., 2015] review techniques from all three categories. This chapter focuses on methods appropriate to increasing the computational efficiency of distributed groundwater flow models for applications such as sensitivity analysis, uncertainty analysis, and calibration. Many surrogate methods are unsuitable for this application as they are unable to handle large numbers of inputs and outputs (specifically spatial fields and time series). Other approaches are intrusive (require complex model code to be edited) or have subjective structures. The time and expertise required by practitioners to use them is thus prohibitive. Data-driven surrogates in particular exhibit poor performance at new values of $\mathbf{z}$ far away from the runs used to calibrate the surrogate.

Surrogate approaches are typically developed for a particular setting. For example, Le Gratiet et al. [2014] present a multi-fidelity method specific to the estimation of Sobol' sensitivity indices. Giles [2008] presents another multi-fidelity method for uncertainty propagation (estimating statistical moments of model outputs based on varying inputs). An oft overlooked consideration for practitioners is the flexibility of a surrogate method. Practical approaches should be relevant in a number of settings and allow the reuse of model runs.

Bi-fidelity stochastic collocation, the approach employed in this chapter, combines

techniques from both Hierarchical and Projection based methods. As in the reduced basis method [e.g. Lieberman et al., 2010], the greedy algorithm is used to select a basis which spans the output space of the complex model. High-fidelity outputs can then be approximated in terms of this basis. A defining feature of the algorithm is the use of a low-fidelity model in both the basis selection and the projection required to approximate new outputs. Runtimes are hence reduced, as the vast majority of necessary high-fidelity runs can be approximated at the cost of a low fidelity simulation.

The method outperforms many surrogates in the aforementioned criteria. It is non-intrusive and uses computational resources efficiently by allowing simple parallelization, reuse of low and high fidelity model runs, and iterative addition of more high-fidelity runs to improve accuracy. A further advantage is that the low-fidelity output need not be a direct approximation of the high-fidelity output. For example a low-fidelity spatial field $\mathbf{u}^L = (u_1^L, \cdots u_{Q^L}^L)$ may be used with a high-fidelity timeseries $\mathbf{u}^H = (u_1^H, \cdots u_{Q^H}^H)$, provided the uniqueness of the $\mathbf{z}$ to $\mathbf{u}^H$ relationship is emulated by u^L. Although not necessary, $\mathbf{u}^L$ and $\mathbf{u}^H$ will often relate to similar quantities - such as different spatial resolutions of the same output. In this case, an important practical advantage of this method over statistical approaches is the ability to compare the surrogate approximation with a low-fidelity output. Our work provides anecdotal evidence that the algorithm deals with correlated inputs, a known shortcoming of methods such as polynomial chaos expansions. This study investigates the ability of bi-fidelity stochastic collocation for building a surrogate model. We have not investigated the effect of using more than two fidelities. (Zhu et al. [2014] suggest three may be optimal) and its performance in the case of "tensor stratification" [Narayan et al., 2014], where the inputs of the low-fidelity model are a subset of the high-fidelity inputs.

One purpose of this work is to determine whether bi-fidelity stochastic collocation, applied to multiple resolutions of the same groundwater flow model, is a practical surrogate approach. The process of creating discretized numerical models from various data sources is increasingly automated, in common graphical user interfaces (eg. Groundwater Vistas Rumbaugh and Rumbaugh [2007], Visual MODFLOW FLEX

Fitzpatrick [2012], Aquaveo GMS Aquaveo [2011] or ModelMuse Winston [2009]) or purpose built tools [e.g. Vermeulen, 2013; Bhatt et al., 2014]. This has created an opportunity for modellers to easily produce multiple fidelities of the same model, making the multi-fidelity methods of real practical interest. Despite the well documented problems with changing the resolution of hydrological parameters [Vermeulen et al., 2006; Neuman and Di Federico, 2003; Wen and Gómez-Hernández, 1996; Mehl and Hill, 2010], multi-fidelity groundwater surrogates have been demonstrated to be effective [Mehl and Hill, 2002; Burrows and Doherty, 2014]. We aim here to investigate whether different resolutions of the same groundwater model can be combined effectively using a multi-fidelity surrogate.

This chapter is structured as follows. Section 4.2 details the bi-fidelity algorithm. Section 4.3 gives several illustrative examples of its application to trivial problems, introducing our analysis methods. Section 4.4 demonstrates our development of the method and the application to a groundwater flow model. Conclusions follow in Section 4.5.

4.2 Multi-fidelity Stochastic Collocation

Consider again the generic model (4.1)

$$\mathbf{u} = u(\mathbf{z}) \tag{4.2}$$

with uncertain or unknown inputs $\mathbf{z} = (z_1, \cdots z_D) \in I_z \subseteq \mathbb{R}^D$ and outputs $\mathbf{u} = (u_1, \cdots u_Q)$. Narayan et al. [2014] developed a bi-fidelity stochastic collocation surrogate which combines a high-fidelity implementation of (4.2),

$$u^H : I_z \to V^H, \tag{4.3}$$

and a low-fidelity implementation

$$u^L : I_z \to V^L \tag{4.4}$$

to produce an approximator of u^H at the cost of u^L.

The bi-fidelity surrogate is created as follows.

1. Evaluate the low-fidelity model u^L on a candidate set $\Gamma = \{\mathbf{z}_m\}_{m\in\{1,2,\cdots,M\}} \subset I_Z$, ie. compute
$$u^L(\Gamma) = \{u^L(\mathbf{z}_m)\}_{m\in(1,2,\cdots,M)}. \tag{4.5}$$

2. Choose an ordered subset $\gamma = \{\mathbf{z}_n\}_{n\in(1,2,\cdots,N)} \subset \Gamma$ which best spans Γ using the Greedy algorithm. See Section 4.2.1 for details.

3. Evaluate the high-fidelity model u^H on γ, ie. compute
$$u^H(\gamma) = \{u^H(\mathbf{z}_n)\}_{n\in(1,2,\cdots,N)}. \tag{4.6}$$

4. For any new input $\mathbf{z} \in I_z$, $\mathbf{z} \notin \gamma$, the bi-fidelity surrogate is found by projecting $u^L(\mathbf{z})$ onto $u^L(\gamma)$, then use these coordinates to estimate $u^H(\mathbf{z})$ in terms of $u^H(\gamma)$. See Section 4.2.2 for details.

An example of the above method implemented in Python is given in Figure 4.19, applied to two resolutions of a one dimensional diffusion equation 4.20. Together with the below algorithms (Figures 4.8 and 4.1), these scripts form a complete example which readers can run on any computer with Python and a few common libraries (Numpy and Scipy) installed. We are of the opinion that very little efficiency of expression or clarity is lost by presenting Equation 4.8 in Python as opposed to the more common "pseudo-code". Using a real language has the advantage of providing a ready to use example for readers to try, and perhaps more importantly, allowing the authors to test the algorithm exactly as it is presented. It is our experience that "pseudo-code" in the literature contains many errors, likely because authors cannot test it in this fashion.

As detailed in Zhu et al. [2014], the simulation cost has two important components. One is the N low-fidelity runs making up $u^L(\Gamma)$, and the other the M high-fidelity runs $u^H(\gamma)$. Each new value $\mathbf{z} \in I_z$, $\mathbf{z} \notin \gamma$ requires the evaluation of $u^L(\mathbf{x})$. The expense of the linear algebra required for the node selection and projection will be

insignificant compared to these model evaluations. A key advantage of this algorithm is that if either M or N is increased, the previously computed low and high-fidelity outputs can be reused. Furthermore, as the samples are independent, both $u^L(\Gamma)$ and $u^H(\gamma)$ can be evaluated in parallel and can be reused. For example, $u^L(\Gamma)$ might be used to select the nodes, and also to provide uncertainty estimates of the outputs of u^H using the multi-fidelity surrogate.

4.2.1 Selection of the collocation nodes

We aim to choose a set of N basis vectors, $u^H(\gamma)$, which minimises the distance between $\text{span}(u^H(\gamma))$ and $\text{span}(u^H(\Gamma))$. Implicit is the assumption that $\text{span}(u^H(\Gamma)) \approx V^H$. To reduce computational cost, we use u^L instead of u^H and treat Γ as a finite set rather than a continuum.

The Greedy Algorithm achieves this as follows. Define a distance function between an output $v \in V^L$ and a set of outputs $W \subset V^L$

$$\begin{aligned} d^L(v, W) &= \inf_{w \in W} \|v - w\|^L \\ &= \|(I - P_W)v\|^L \\ &= \|v - \sum_{w \in W} \frac{\langle w, v \rangle}{\langle w, w \rangle} w\|^L \end{aligned} \tag{4.7}$$

where P_W is the orthogonal projection operator onto W. Start with $\gamma_0 = \{\}$ and iteratively expand the set

$$\begin{aligned} \mathbf{z}_n &= \arg\max_{\mathbf{z} \in \Gamma} d^L \left(u^L(\mathbf{z}), u^L(\gamma_{n-1}) \right) \\ \gamma_n &= \gamma_{n-1} \cup \{\mathbf{z}_n\} \end{aligned} \tag{4.8}$$

where $u^L(\gamma_{n-1}) = \{u^L(\mathbf{z}_i)\}_{i \in (1,2,\cdots,n-1)}$. The algorithm is implemented as in Figure 4.1 using the pivoted Cholesky decomposition, also known as an incomplete Cholesky decomposition or partial Gram-Schmidt orthogonalization [Hardoon et al., 2004]. As in Zhu et al. [2014], we assume that the continuous inner product in (4.7) can be computed by a discrete dot product.

```
from numpy import dot, array, arange, zeros, empty, argmax, finfo, sqrt
def select_nodes(V, N):
    # The columns of $V$ are the $M$ candidate outputs of the low-fidelity model
    # $V = u^L(\Gamma)=[ u^L(\mathbf{z}_1), u^L(\mathbf{z}_2) \cdots u^L(\mathbf
        {z}_M)  ]$.
    # $N$ is number of interpolation nodes/ high-fidelity runs
    M = V.shape[1]; assert N <= M;
    # Elements of $w$ are the norms for corresponding parameter $\mathbf{z}_m$
    w = array([dot(V[:, m], V[:, m]) for m in range(M)])
    # $P$ is the permutation vector
    P = arange(M, dtype=int)
    # $L$ is the Cholesky factor
    L = zeros((M, N))
    r = empty((M))
    for n in range(N):
        # Choose largest norm for the next pivot/ interpolation point
        p = argmax(w[n:M]) + n
        # Avoid ill-conditioning if norm is less than machine precision
        if w[p] < 2*finfo( float ).eps:
            print 'Grammian␣is␣numerically␣singular...The␣grammian␣has␣rank␣%s␣
                and␣size␣%s'%(n,M)
            n -= 1
            break
        # Update indices in $P$ and swap columns $n$ and $p$ of $V$, $L$, and
            $w$
        P[[n, p]] = P[[p, n]]
        V[:, [n, p]] = V[:, [p, n]]
        L[[n, p], :] = L[[p, n], :]
        w[[n, p]] = w[[p, n]]
        # Update $L$
        for t in range(n+1, M):
            r[t] = dot(V[:, t], V[:, n]) - sum([ L[t, j]*L[n, j] for j in range(
                n)])
        L[n, n] = sqrt(w[n])
        for t in range(n+1, M):
            L[t, n] = r[t]/L[n, n]
            w[t] = w[t] - L[t, n]**2
    # Truncate the Cholesky factor and permutation vector
    L = L[:n+1, :]
    P = P[:n+1]
    return P, L
    # Note that $L L^T = V^T V$ where $V$ is the input $V$ with columns permuted
         according to $P$
```

Figure 4.1: Algorithm as Python code to select the interpolation nodes γ from the candidates Γ using a pivoted Cholesky decomposition based Greedy algorithm.

4.2.2 Bi-fidelity surrogate

For any input $\mathbf{z} \in I_z$, $\mathbf{z} \notin \gamma$, the bi-fidelity surrogate is constructed as the projection of $u^H(\mathbf{z})$ onto $u^H(\gamma)$

$$u^H(\mathbf{z}) \approx u^B(\mathbf{z}) = \sum_{n=1}^{N} c_n u^H(\mathbf{z}_n) \tag{4.9}$$

where c_n solve

$$\langle \sum_{n=1}^{N} c_n u^L(\mathbf{z}_n), \phi \rangle^L = \langle u^L(\mathbf{z}), \phi \rangle^L, \qquad \forall \quad \phi \in u^L(\gamma). \tag{4.10}$$

Since only $u^L(\mathbf{z})$ is required, the computational cost is reduced by the ratio of u^H to u^L runtime. The discrete form of this equation algorithm is implemented in Figure 4.2. Again the continuous inner product in Equation (4.10) can be computed by a discrete dot product.

```
from numpy import dot
from numpy.linalg import inv, cond
def synthesis_operator(LF_selected_values, HF_selected_values, L, LF_new):
        # LF_selected_values is $u^L(\gamma)$
        # HF_selected_values is $u^H(\gamma)$
        # L is the Cholesky product from the node selection algorithm
        # LF_new is $u^L(\mathbf{z})$
        G = dot(L,L.T)
        G_inv = inv( G )
        g = dot( LF_selected_values.T, LF_new )
        c = dot( G_inv, g )
        # MF_new approximates $u^H(\mathbf{z})$
        MF_new = dot( HF_selected_values, c ).squeeze()
        return MF_new, cond(G)
```

Figure 4.2: Algorithm as Python code to compute multi-fidelity surrogate for new input.

4.2.3 Configurations

Given a high-fidelity model $u^H : I_z \to V^H$, a number of choices must be made when building a bi-fidelity surrogate.

1. A low-fidelity model $u^L : I_z \to V^L$ must be used that replicates the uniqueness of the $I_z \to V^H$ relationship. Note that V^L need not approximate V^H.

2. Choose the size M and structure (sampling scheme) of Γ such that $u^L(\Gamma)$ spans

V^L. Without prior knowledge of which samples will span the space it is necessary for M to be a large number. M is the number of low-fidelity runs.

3. Select the size N of the subset γ such that $u^L(\gamma)$ may be selected from the larget set, so that it too spans V^L. N is the number of high-fidelity runs, and will be significantly smaller than M. The assumption is that $u^H(\gamma)$ will span V^H.

4. Determine the discrete form of the inner product to be used in (4.7) and (4.10). Narayan et al. [2014] suggest that it may be possible to improve the accuracy of certain quantities of interest by weighting the inner product. We could not identify examples of this in our investigation.

In this work, we consider a customization to the algorithm whereby the columns of $u^L(\Gamma)$ are normalized for the selection of the collocation nodes. Hence the Grammian $\mathbf{G}$ cannot be computed from the Cholesky decomposition as LL^T, as in Narayan et al. [2014]. Instead we compute $\mathbf{G}$ less efficiently, but equivalently, as $u^L(\gamma)^T u^L(\gamma)$. Models based on spatially and temporally varying partial differential equations typically result in a large number of outputs; one or more for each point in space and time. Depending on the purpose, the quantities of interest $\mathbf{u} = (u_1, \cdots u_Q)$ are chosen or computed from these. Here, we experiment (in Section 4.4) with a number of low-fidelity outputs $\mathbf{u}^L$ based on the same underlying model, in an attempt to approximate a given high-fidelity output $\mathbf{u}^H$.

4.3 Simple numerical examples

To introduce our analysis of the methods, we give a number of simple examples from the literature. We use several error metrics below to compare a set of P bi-fidelity outputs $\{u^B(\mathbf{z}_i)\}_{i=(1,\cdots,P)}$ to a test set of high-fidelity outputs $\{u^H(\mathbf{z}_i)\}_{i=(1,\cdots,P)}$. Where both outputs consist of Q quantities of interest, the mean L^2 error based on P

samples is calculated

$$\mathbb{E}\left(\|u^H - u^B\|_{L^2}\right) = \frac{1}{P}\sum_{i=1}^{P}\|u^H(\mathbf{z}_i) - u^B(\mathbf{z}_i)\|_{L^2} = \frac{1}{P}\sum_{i=1}^{P}\sqrt{\frac{\sum_{q=1}^{Q}(u_{i,q}^H - u_{i,q}^B)^2}{Q}} \tag{4.11}$$

where $u_{i,q}^H$ is the q^{th} quantity of interest of the bi-fidelity model evaluated at $\mathbf{z}_i$. The relative mean L^2, which attempts to better account for sampled outputs which vary by orders of magnitude, is defined

$$\mathbb{E}\left(\|u^H - u^B\|_{L^2}\right) = \frac{1}{P}\sum_{i=1}^{P}\|u^H(\mathbf{z}_i) - u^B(\mathbf{z}_i)\|_{L^2} = \frac{1}{P}\sum_{i=1}^{P}\sqrt{\frac{\sum_{q=1}^{Q}(u_{i,q}^H - u_{i,q}^B)^2}{\sum_{q=1}^{Q}(u_{i,q}^H)^2}}. \tag{4.12}$$

The L^∞ error is

$$\|u^H - u^B\|_{L^\infty} = \max_{i\in(1,\cdots,P),q\in(1,\cdots,Q)}|u_{i,q}^H - u_{i,q}^B|. \tag{4.13}$$

4.3.1 Trigonometric example

Narayan et al. [2014] illustrate the technique with approximations of the function

$$u(x,z) = g(x, z+\epsilon z^2) \triangleq \cos\left(x(z+\epsilon z^2)+1\right), \qquad (x,z)\in[-1,1]\times[0,10\pi].$$

The low-fidelity model is

$$u^L(x,z) = \sum_{k=0}^{Q^L-1}\widehat{g}_k(z)\widetilde{L}_k(x)$$

where

$$\widehat{g}_k(z) = c_k\sqrt{\frac{\pi(2k+1)}{|z|}}J_{k+1/2}(|z|), \qquad z\neq 0$$

$$c_k = \Re\left[e^{i\frac{z}{|z|}}i^k\right]$$

J_k is the order-k Bessel function of the first kind and $\widetilde{L_k}$ is the degree-k Legendre polynomial orthonormal under the unit weight function on $[-1, 1]$. Here Q^L and Q^H are the number of terms in the low and high-fidelity expansions respectively. The high-fidelity model has more terms ($Q^L < Q^H$) and includes a fidelity parameter ϵ.

$$u^H(x, z) = \sum_{k=0}^{Q^H-1} \widehat{g_k}(z + \epsilon z^2)\widetilde{L_k}(x).$$

Convergence of the bi-fidelity surrogate is shown in Figure 4.3. Here we reveal an aspect of the algorithm not obvious in previous work. While the performance of the multi-fidelity surrogate improves with the size, N, of γ to a point, it degrades rapidly thereafter. We include three more plots which explain this phenomenon. The error at the interpolation nodes γ increases almost identically to the condition number of the grammian $u^L(\gamma)^T u^L(\gamma)$. These in turn correspond to the decay of the eigenvalues of the candidates $u^L(\Gamma)$. A key limitation of the algorithm is that if too many interpolation nodes are used, redundancy begins to affect the uniqueness of the solution to the projection, as we see around $N = 22$ in Figure 4.3. This is mitigated as in the algorithm given in Figure 4.1 by stopping when the Grammian is singular.

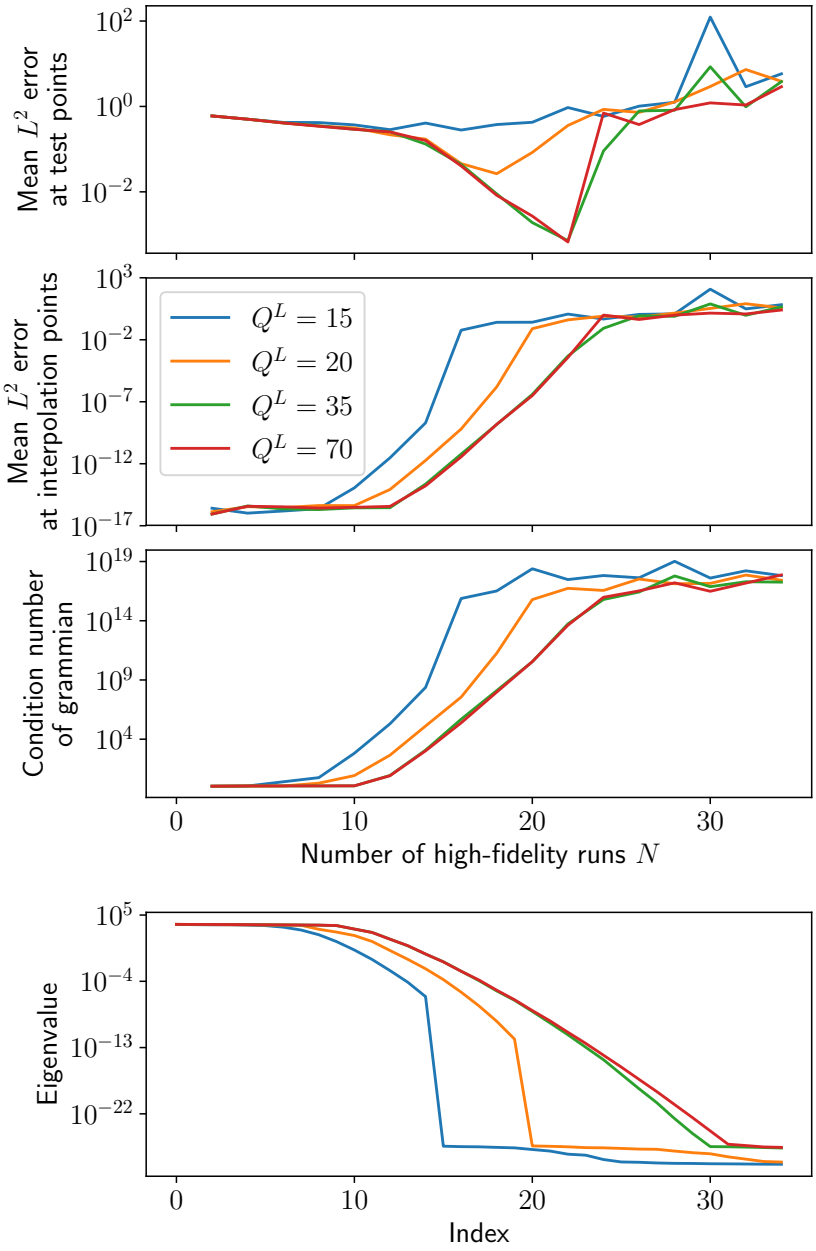


Figure 4.3: Bi-fidelity example from Narayan et al. [2014]. Mean L^2 error at test points, collocation points, the condition number of Grammian and the eigenvalues of the $M = 1000$ candidate points. Errors calculated using 1000 MC tests. High and low-fidelity are evaluated at $Q = 100$ points in space. The high-fidelity model has $Q^H = 100$ terms. $\epsilon = 5 \times 10^{-3}$. The top plot is the same as Figure 1a or Narayan et al. [2014].

4.3.2 1D Diffusion with KLE using Chebyshev Collocation

Another example from the preceding literature [Zhu et al., 2014], involves the steady state 1D diffusion equation

$$\begin{aligned} \frac{\partial}{\partial x}\left(c(x,\mathbf{z})\frac{\partial u}{\partial x}\right) &= -f(x,\mathbf{z}) \\ u(0) = u(1) &= 0 \qquad 0 < x < 1 \end{aligned} \tag{4.14}$$

where the quantity of interest is the density, $u(x)$, and the unknowns are the conductivity, $c(x,\mathbf{z})$, and forcing term, $f(x,\mathbf{z})$.

As in Zhu et al. [2014], we parameterize conductivity using the Karhunen-Loeve expansion (KLE)

$$c(x,\mathbf{z}) = 1 + \sigma \sum_{d=1}^{D-1} \frac{1}{d\pi}\cos(2\pi d x) z_d \tag{4.15}$$

with $\sigma = 0.5$ and $(z_1, \cdots, z_{D-1}) \in [-1,1]^{D-1}$ is uniformly distributed. The forcing term is considered constant

$$f(x,\mathbf{z}) = 1. \tag{4.16}$$

Chebyshev collocation is used to solve (4.14). The low and high-fidelity models differ only in the number of collocation nodes used. Lagrange interpolation is used to interpolate low-fidelity outputs to the locations of the high-fidelity outputs. Convergence of the bi-fidelity surrogate is shown in Figure 4.4. We plot the error at interpolation points and the condition number along with the error because these are the quantities available in a real setting (where the test point error is not). In these settings, those data points must be used to determine the number of high fidelity runs to be used.

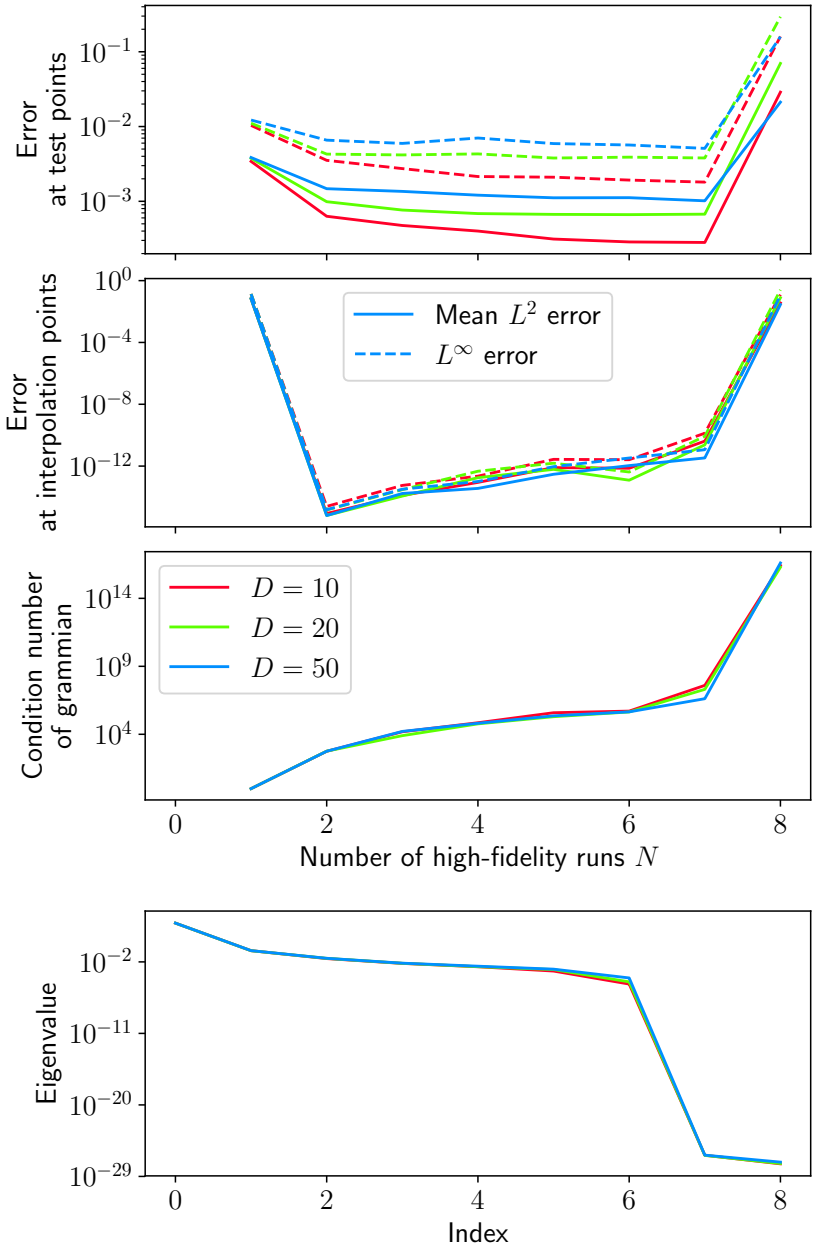


Figure 4.4: 1D diffusion equation with diffusivity represented by a KLE of $D-1$ terms. LF has 2^4 Chebyshev collocation nodes, HF has 2^7. The top figure replicates Figure 2 of Zhu et al. [2014].

4.3.3 1D Diffusion with KLE using FEM

Another example uses the finite element method to solve (4.14). The forcing term is given

$$f(x, \mathbf{z}) = 50 z_D^2 \tag{4.17}$$

where z_D normally distributed on $N(0, 1)$. The same KLE expansions as above (4.15) is used for conductivity, with coefficients all uniformly distributed on $U(-1, 1)$. Using a 2^9 node solution as the high-fidelity model, Figure 4.5 shows the convergence of the multi-fidelity method. The use of a random forcing term may be causing the high error when using a high resolution u^L. If there was serial correlation in the driving force, then the lower resolution u^L may be more effective.

To introduce diagnostics used later, Figure 4.6 plots the low $u^L(\mathbf{z})$, high $u^H(\mathbf{z})$, and multi-fidelity output $u^B(\mathbf{z})$ for a few random test inputs $\mathbf{z}$.The residuals, $u^H(\mathbf{z}) - u^B(\mathbf{z})$ and $u^H(\mathbf{z}) - u^L(\mathbf{z})$ are also plotted. We include these for each of the examples below to visualize the model output and basis functions and explain some of the artefacts of the algorithm. For instance in 4.6 we note the periodic nature of the residuals. These are the shapes of the truncated basis functions (beyond the 6 basis functions included in this MF surrogate).

Figure 4.7 shows the candidates $u^L(\Gamma)$ along with the chosen $u^L(\gamma)$ and $u^H(\gamma)$. In common with the other examples in this section, and many such "toy" problems from the literature, u^L is itself a close approximation to u^H. This motivates the use of a more "complex" example in Section 4.4.

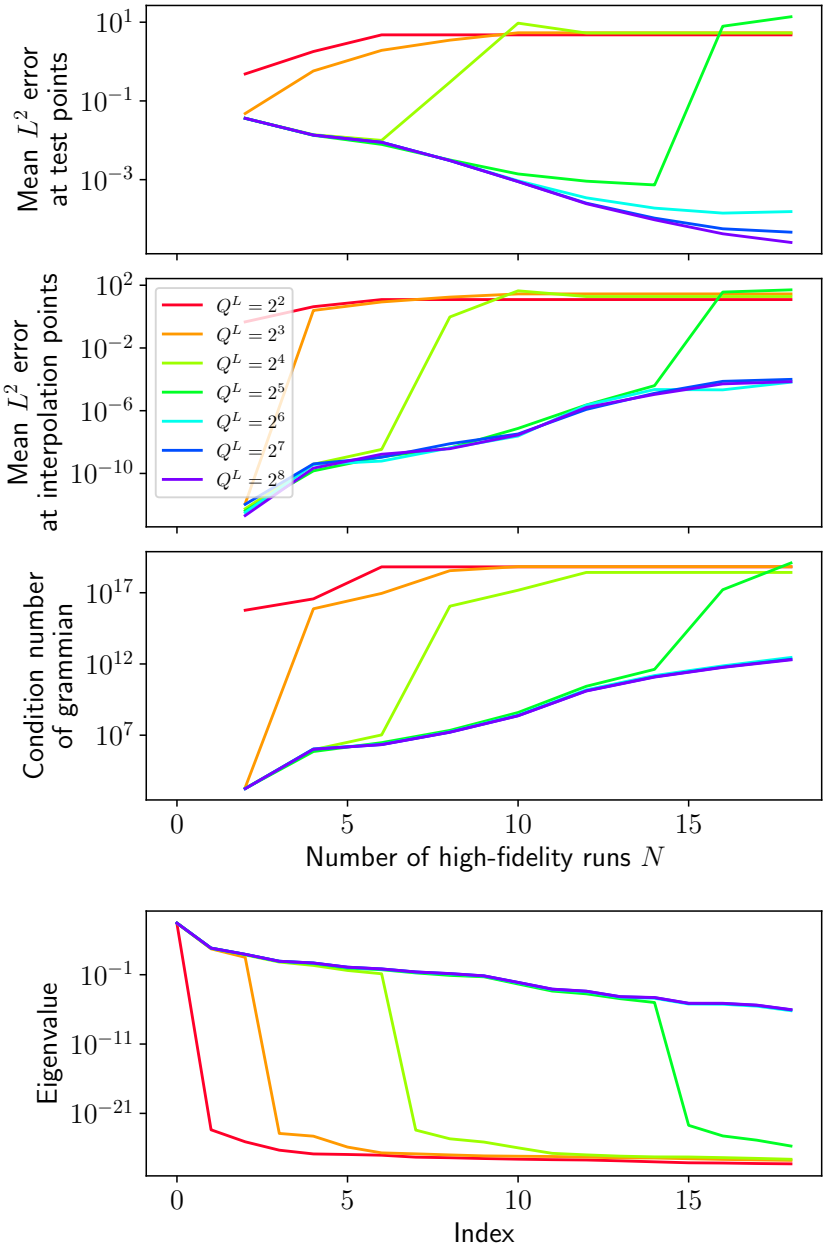


Figure 4.5: 1D diffusion equation, with diffusivity represented by a KLE of 10 terms and random forcing term. The solution is by the Finite Element method. High-fidelity resolution $Q^H = 2^9$.

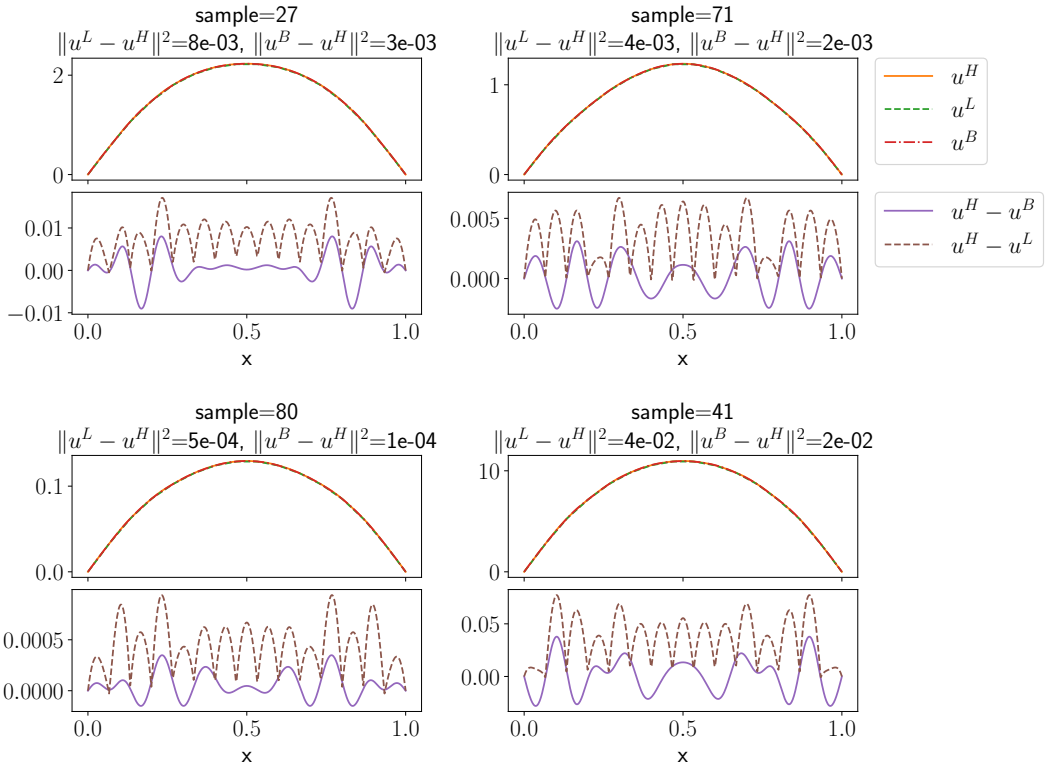


Figure 4.6: LF, MF and HF for four random inputs. LF resolution=2^4, HF runs=6, mean L^2 for LF=2.32e-02, mean L^2 for MF=7.54e-03, condition number for plots=1.19e+07 HF resolution is 2^9.

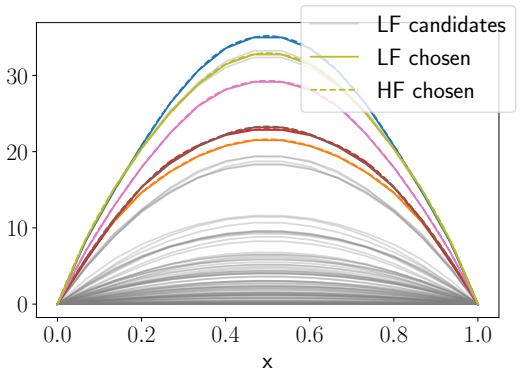


Figure 4.7: LF candidate outputs, and chosen LF and HF outputs.

4.4 Practical example

In this section we apply the bi-fidelity stochastic collocation algorithm to two fidelities of a groundwater flow model. We describe the model in Section 4.4.1, and then present results based on various configurations of the algorithm in Sections 4.4.2 through 4.4.4.

4.4.1 Bi-fidelity groundwater model

The purpose of the model used in this section is to find the maximum drawdown (decrease in pressure) in surface aquifers due to coal seam gas development. It is implemented in the 2.5D finite difference groundwater flow model code, MODFLOW Harbaugh [2005], based on the equation

$$s(\mathbf{x},\mathbf{z})\frac{\partial h(\mathbf{x},t,\mathbf{z})}{\partial t} = \nabla_{\mathbf{x}}\cdot(k(\mathbf{x},\mathbf{z})\nabla_{\mathbf{x}}h(\mathbf{x},t,\mathbf{z})) + f(\mathbf{x},t,\mathbf{z}), \qquad \mathbf{x}\in D, \quad t\in(0,T] \tag{4.18}$$

where h is pressure, k is conductivity, s is specific storage and f is a source sink term. Discretization is as follows. A 30 year development period with pumping to keep production wells in lower layers dry is divided into 20 timesteps, a further 60 years without pumping is divided into 30 timesteps. Vertically, the model is 1200m thick, composed of 34 layers including coal seams and confining units. A 10×40km area is divided into horizontal cells of 400×400m for high-fidelity and 2400×2400m for low-fidelity model. This results in a HF runtime which is 50 times the LF on average. Our quantity of interest, and therefore $\mathbf{u}^H$, is a drawdown timeseries for a particular location $\mathbf{x}_a$ in the top layer

$$u^H(\mathbf{z}) = (h(\mathbf{x}_a,0,\mathbf{z}),\cdots,h(\mathbf{x}_a,T,\mathbf{z}))\,. \tag{4.19}$$

As noted above, $\mathbf{u}^L$ need not be the equivalent output of the low fidelity model. We explore a number of possibilities for $\mathbf{u}^L$ in Sections 4.4.2 through 4.4.4

Much of the uncertainty in such models concerns the conductivity values of aquifer and faults. A 2D linear parameterization is employed for the aquifer properties, with the random variables $\mathbf{z} = (z_1, z_2, \cdots z_{10})$ made up of

1. anisotropy of conductivity (ratio of conductivity North-South to East-West)
2. conductance of faults
3. slope of conductivity in confining unit
4. intercept of conductivity in confining unit
5. slope of conductivity in coal seem
6. intercept of conductivity in coal seem
7. slope of storage in confining unit
8. intercept of storage in confining unit
9. slope of storage in coal seem
10. intercept of storage in coal seem.

4.4.2 Timeseries LF

Perhaps the most straightforward configuration of the bi-fidelity algorithm is to take u^L identical to u^H, but based on the coarser model:

$$u^L(\mathbf{z}) = (h(\mathbf{x}_a, 0, \mathbf{z}), \cdots, h(\mathbf{x}_a, T, \mathbf{z}))\,. \tag{4.20}$$

The convergence of this approach is given in Figure 4.8. We see that for this naive application of the algorithm, as the number of high-fidelity runs increases, the error of the surrogate increases. This undesired behaviour is due to the system becoming ill-conditioned, and steps are taken below to rectify the algorithm. Candidates $u^L(\Gamma)$ and chosen nodes $u^L(\gamma)$ and $u^H(\gamma)$ are plotted in Figure 4.9. Note that most of the basis functions chosen (in colour) are of a large magnitude, while the density of functions is higher in general for lower magnitudes. This is a flaw of the algorithm as it favours basis functions of large magnitude, but the issue is rectified below. We can see in Figure 4.10, where we plot low, $u^L(\mathbf{z})$, high, $u^H(\mathbf{z})$, and multi-fidelity output, $u^M(\mathbf{z})$, for 8 random test inputs $\mathbf{z}$. The residuals, $u^H(\mathbf{z}) - u^M(\mathbf{z})$ are also plotted.

These 8 samples were selected as the quantiles of high-fidelity output from 1000 test samples. As this approach reveals, the multi-fidelity approximator performs much better where $u^H(\mathbf{z})$ is large. While the mean L^2 errors do not vary drastically, the mean relative errors do. This is an artifact of the greedy algorithm, which favours the selection of nodes larger in magnitude, something we address in the following sections.

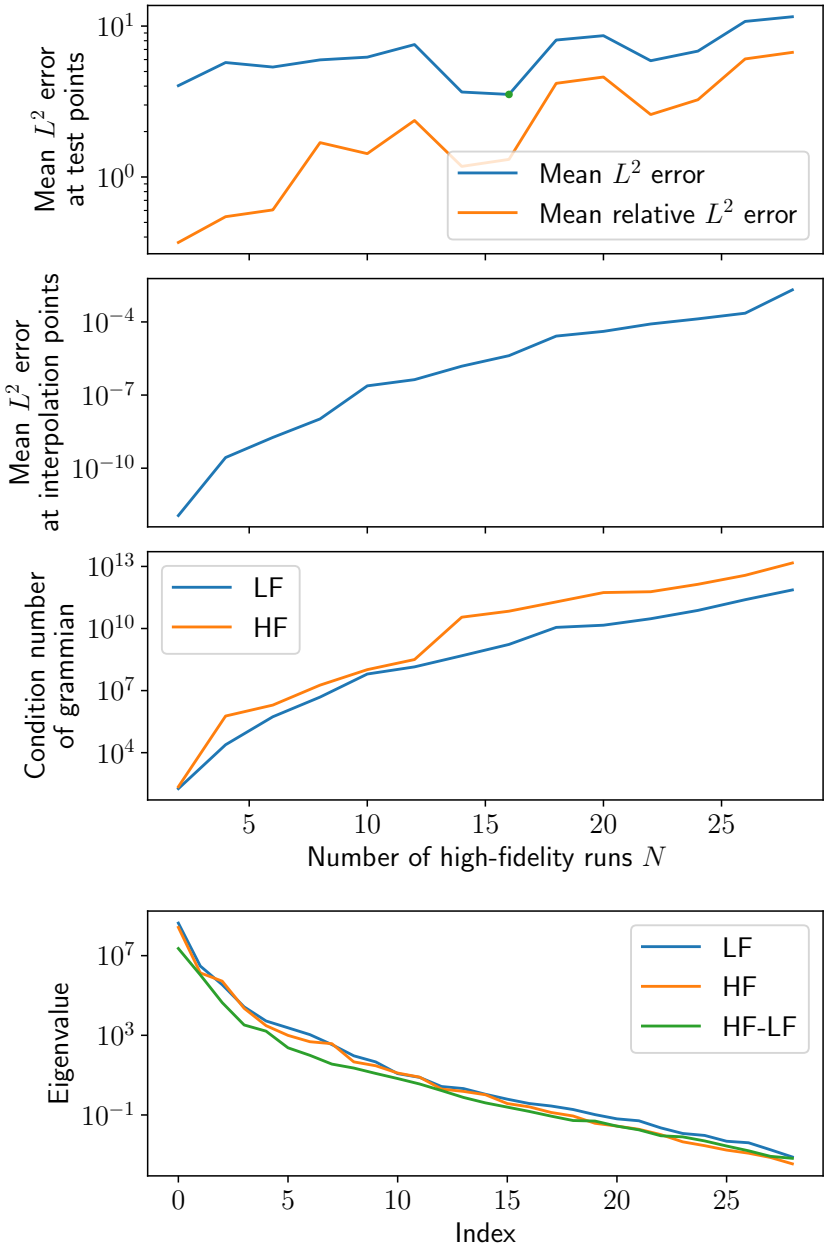


Figure 4.8: Error at 1000 test samples, error at collocation nodes, condition number of Grammian and eigenvalues of $u^L(\Gamma)$. High-fidelity cell size is $Q^H = 400$ meters. $Q^L = 2400$ meters. Candidate runs $M = 1000$.

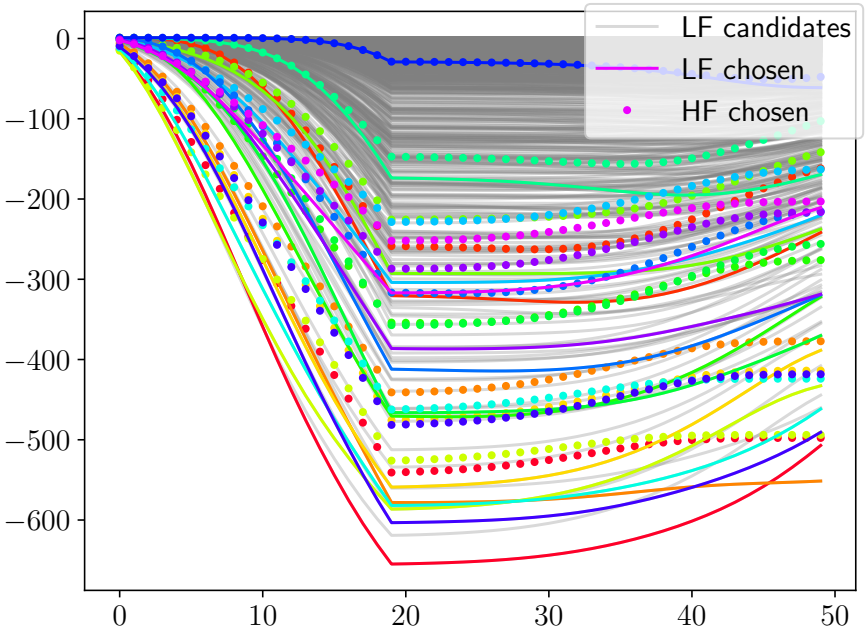


Figure 4.9: Water level over time. LF candidate outputs, and chosen LF and HF outputs.

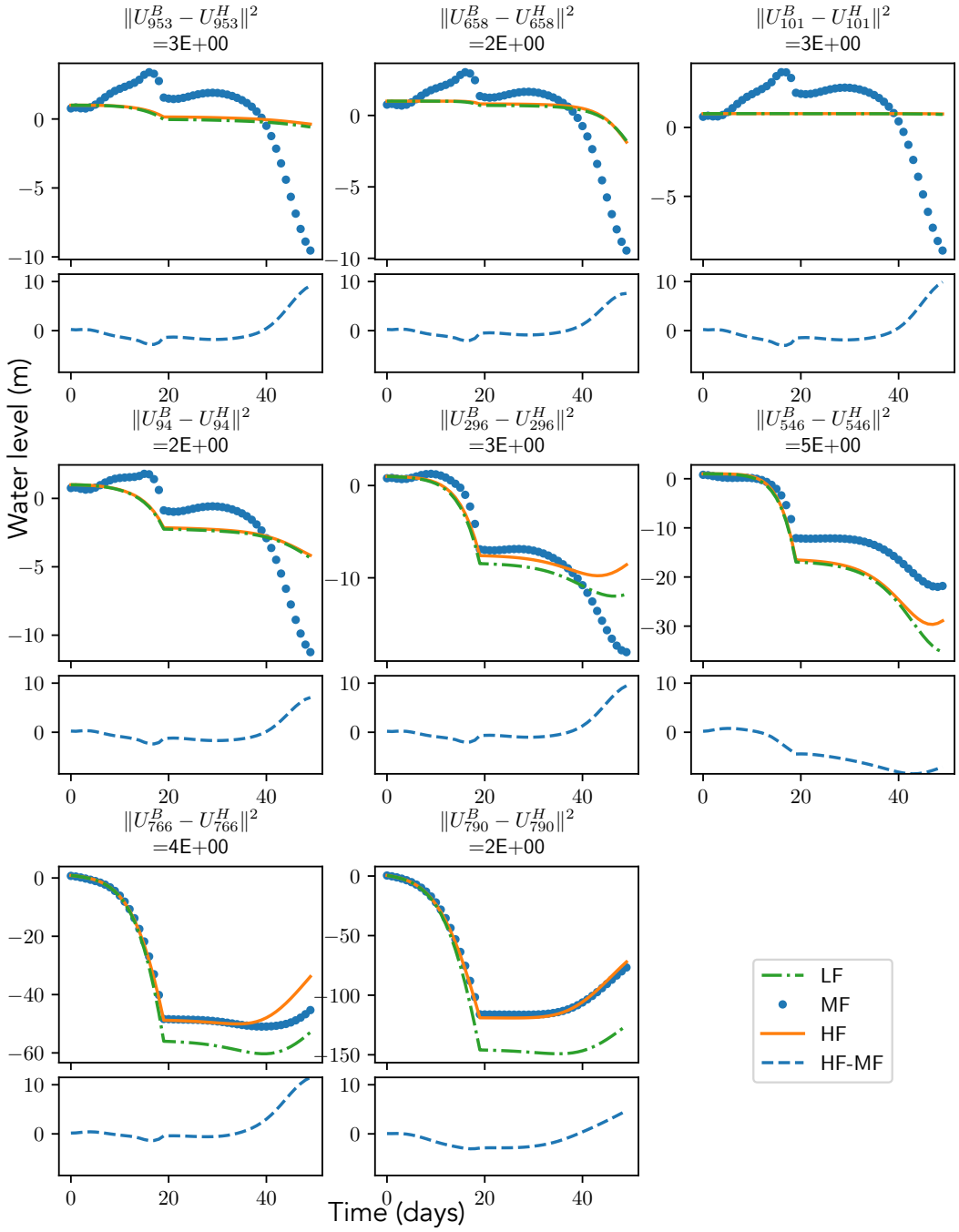


Figure 4.10: Water level versus time for LF, MF and HF for 8 random inputs. HF runs=28, mean L^2 for LF= 8.49e+01, mean L^2 for MF= 3.41e+00, mean relative L^2 for MF= 9.48e-01, condition number for plots=4e+11, HF condition number for plots=6e+12

4.4.3 Normalized timeseries LF

Here we consider an identical formulation to Section 4.4.2, with u^L again the coarse model timeseries output. However, to account for the aforementioned shortcoming of the greedy algorithm, the candidates $u^L(\Gamma)$ are normalized for the selection of the collocation nodes. Hence the Grammian $\mathbf{G}$ cannot be computed LL^T, but must be computed with the unnormalized $u^L(\gamma)^T u^L(\gamma)$. Figures 4.11, 4.12 and 4.13 correspond to Figures 4.8, 4.9 and 4.10 respectively, with a notable improvement in performance. The mean L^2 error in 4.11 converges rather than growing with more high-fidelity runs as it does in 4.8, and the error here ends an order of magnitude lower. One can see the distribution of the basis functions in 4.12 matches that of the underlying data better than in 4.9, where there is a clear bias towards vectors of greater magnitude. While the higher magnitude (latter) subplots of 4.13 have a similar fit to those of 4.10, there is a notable improvement in the earlier subplots with lower magnitude.

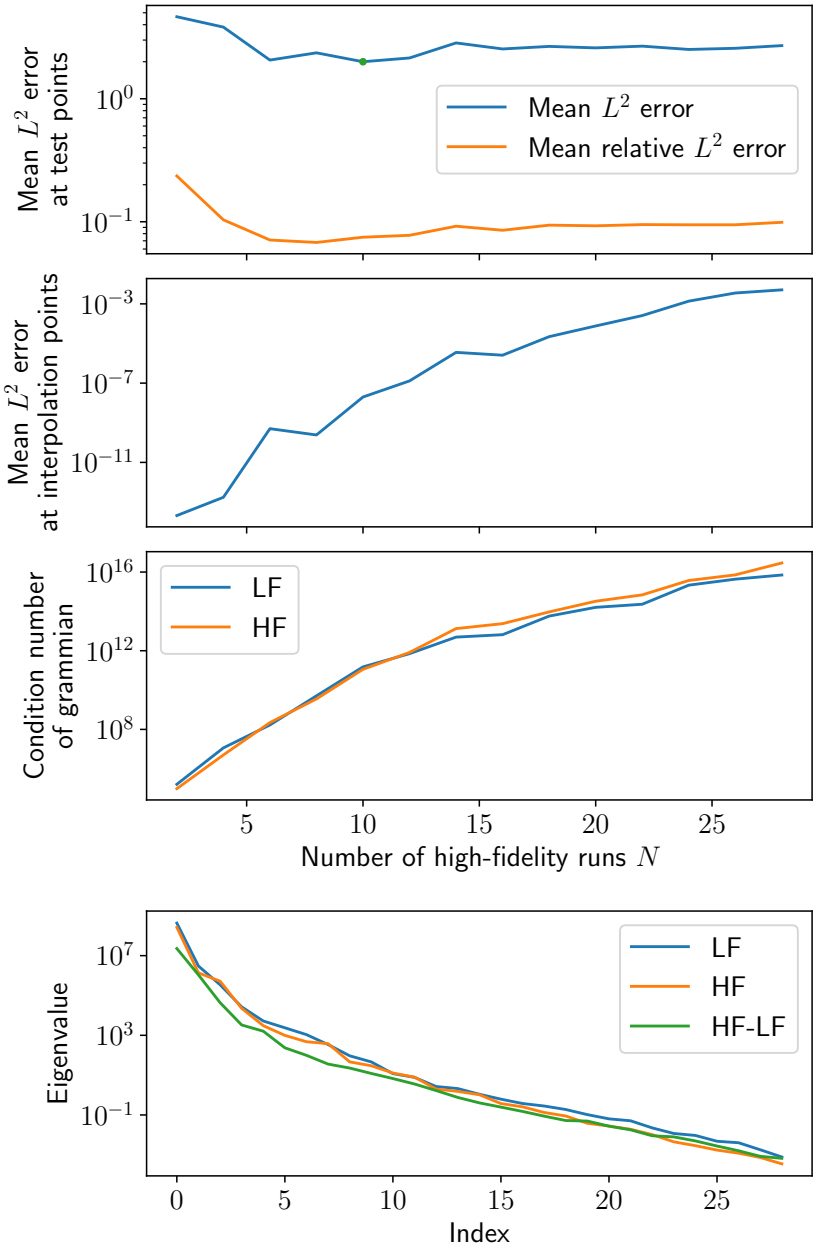


Figure 4.11: Error at 1000 test samples, error at collocation nodes, condition number of Grammian and eigenvalues of $u^L(\Gamma)$. High-fidelity cell size is $Q^H = 400$ meters. $Q^L = 2400$ meters. Candidate runs $M = 1000$.

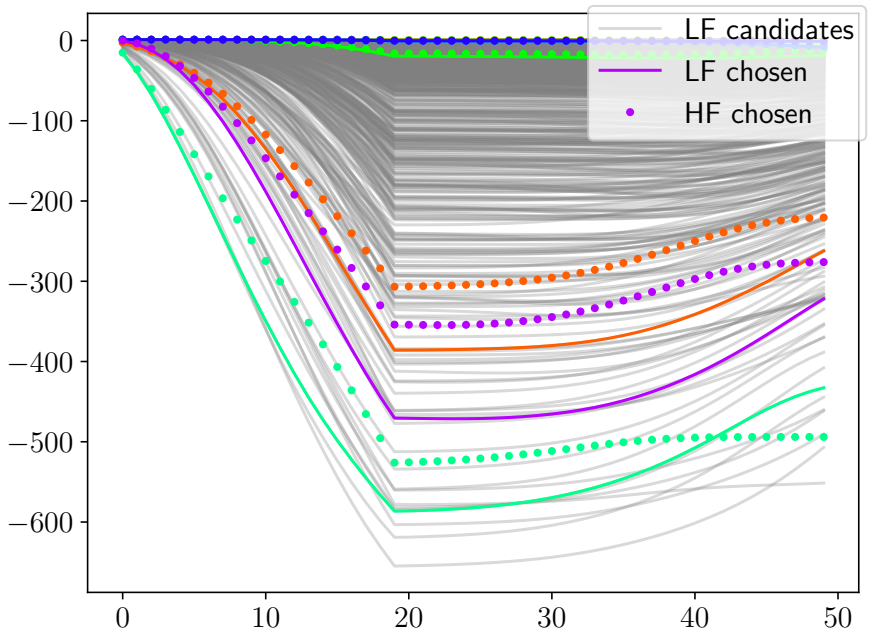


Figure 4.12: LF candidate outputs, and chosen LF and HF outputs.

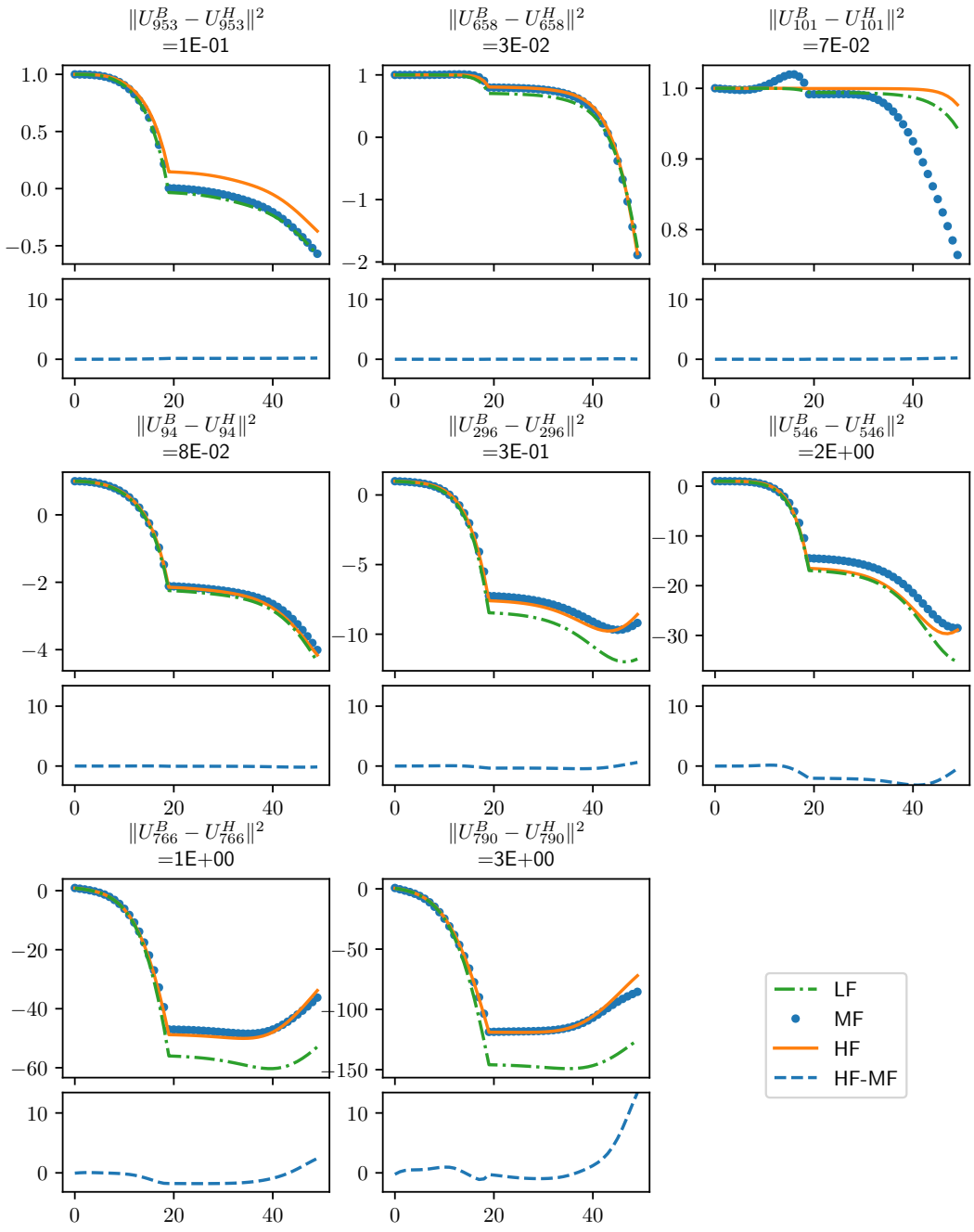


Figure 4.13: LF, MF and HF for 8 random inputs. HF runs=12, mean L^2 for LF= 8.49e+01, mean L^2 for MF= 1.97e+00, mean relative L^2 for MF= 5.65e-02, condition number for plots=2e+12, HF condition number for plots=1e+12

4.4.4 Normalized spatial and timeseries LF

In addition to normalizing the candidate vectors for node selection, we found that, in general, the algorithm improves when more information is added to u^L. An improvement on Section 4.4.2 was found by setting u^L as the spatial output of the coarse model in the top layer for a single timestep t_a

$$u^L(\mathbf{z}) = (h((0,0,0),t_a,\cdots,h((0,X_0,0),t_a,\mathbf{z}),\cdots,h((0,X_0,X_1),t_a,\mathbf{z})) \tag{4.21}$$

where the model has X_0 rows and X_1 columns. Note that this is a different output, but is based on the same underlying model. Thus storage requirements will increase, but computational time should not. Further improvement was noted by taking u^L as a combination of Equations 4.21 and 4.20, the results of which are given in Figures 4.14, 4.15 and 4.16, with a notable improvement in performance. Most notably the mean L^2 error is reduced by another order of magnitude for the selected number of high-fidelity runs and continues to decline, unlike the previous algorithm configurations which converged at this stage. Since we are now combining spatial and temporal data we plot not only the timeseries of head in 4.15 but also the 2D spatial head for the timestep t_a.

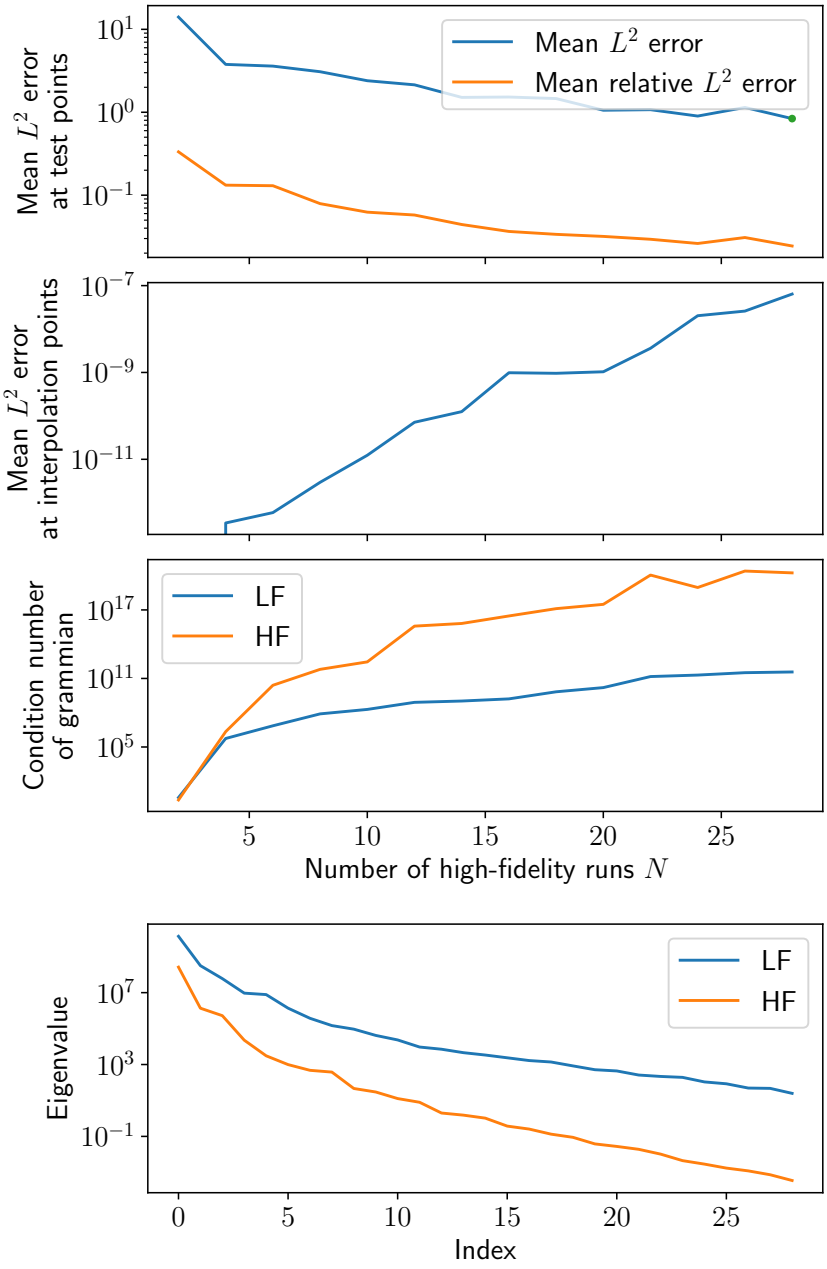


Figure 4.14: Error at 1000 test samples, error at collocation nodes, condition number of Grammian and eigenvalues of $u^L(\Gamma)$. High-fidelity cell size is $Q^H = 400$ meters. $Q^L = 2400$ meters. Candidate runs $M = 1000$.

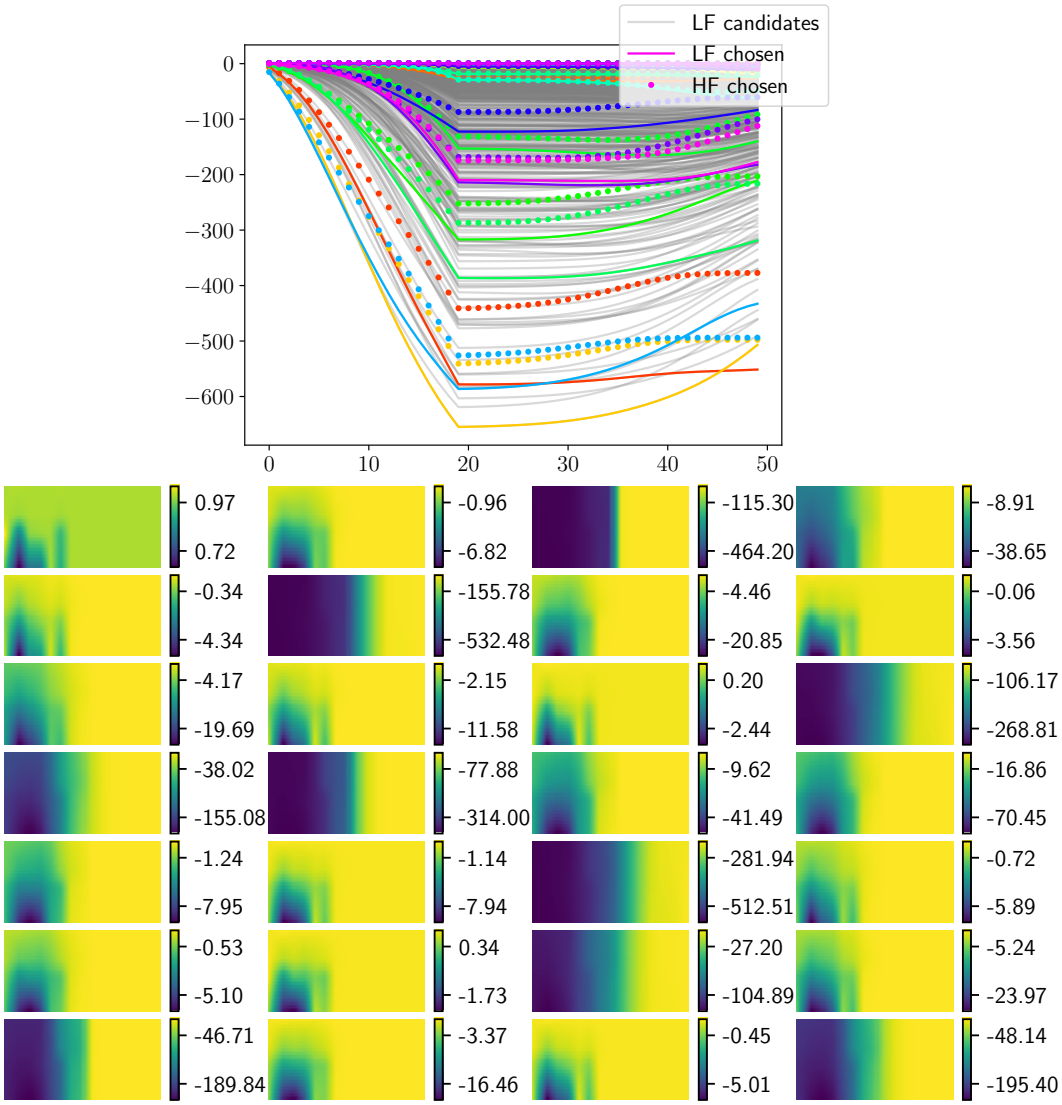


Figure 4.15: LF candidate outputs, and chosen LF and HF outputs. Top plots are water level over time. Bottom panel are 2D spatial cross sections of water level for t_a timestep. Previously (e.g. Figure 4.12 basis functions were only timeseries of water level, but now they are composed of the timeseries (top) concatenated with a flattened array of the 2D spatial cross section (bottom). The aim is to include more information in the basis functions.

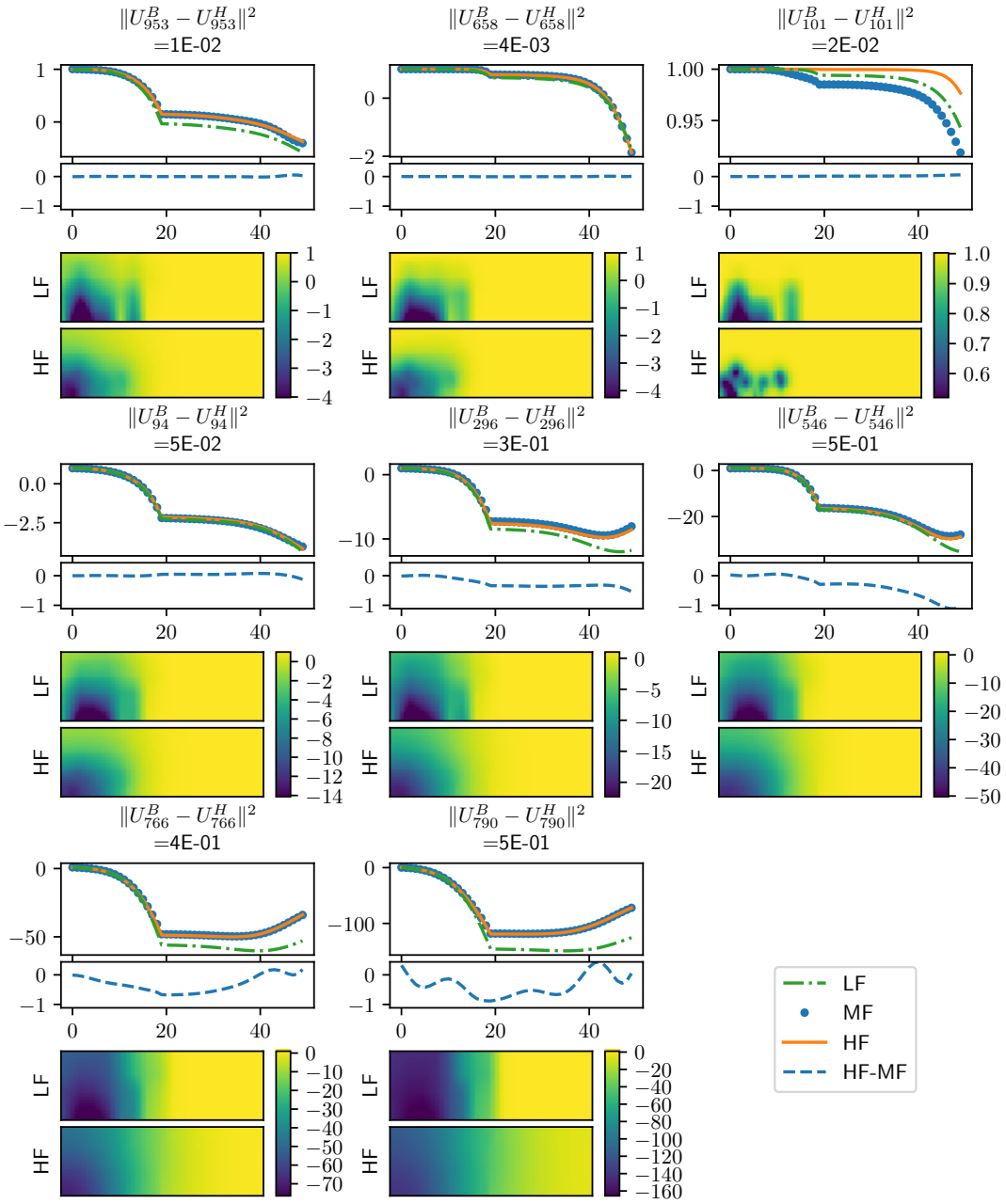


Figure 4.16: LF, MF and HF for 8 random inputs. For each sample we show the water level timeseries (top), error timeseries (middle) and 2D spatial maps of water level for the top layer at the selected timestep (bottom). HF runs=30, mean L^2 for LF= 1.00e+28, mean L^2 for MF= 7.83e-01, mean relative L^2 for MF= 1.76e-02, condition number for plots=6e+11, HF condition number for plots=2e+21

4.4.5 Comparison with LF equal to HF

To compare the above results with the potential of the method, we repeat the configuration in Section 4.4.4, but using the fine resolution simulator for both the low and high-fidelity model. The results are given in Figures 4.17 and 4.18. We omit the basis functions as the 82 chosen for these results in Figure 4.18 is an impractically large number to visualize. Note an improvement in the mean L^2 error from $7.83e-01$ to $1.14e-03$, and the relative error from $1.76e-02$ to $1.81e-05$. These results represent a theoretical maximum bound to the accuracy that could be achieved by this method. Since we are here using the HF for the LF when selecting the basis functions, it is not possible to get a higher fidelity LF than the HF itself. The conclusion here is that a more accurate surrogate could be developed by improving the lower fidelity representation or basis selection algorithm.

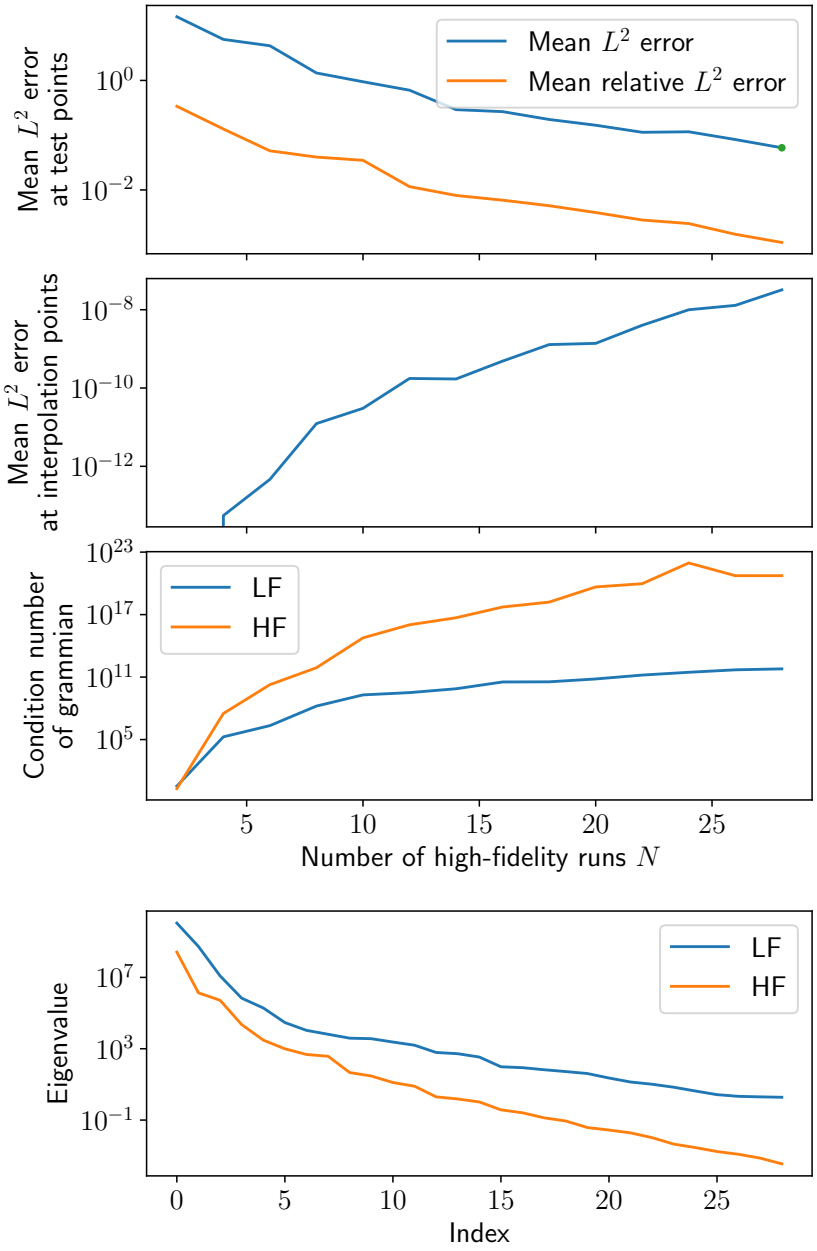


Figure 4.17: Error at 1000 test samples, error at collocation nodes, condition number of Grammian and eigenvalues of $u^L(\Gamma)$. High-fidelity cell size is $Q^H = Q^L = 400$ meters. Candidate runs $M = 1000$.

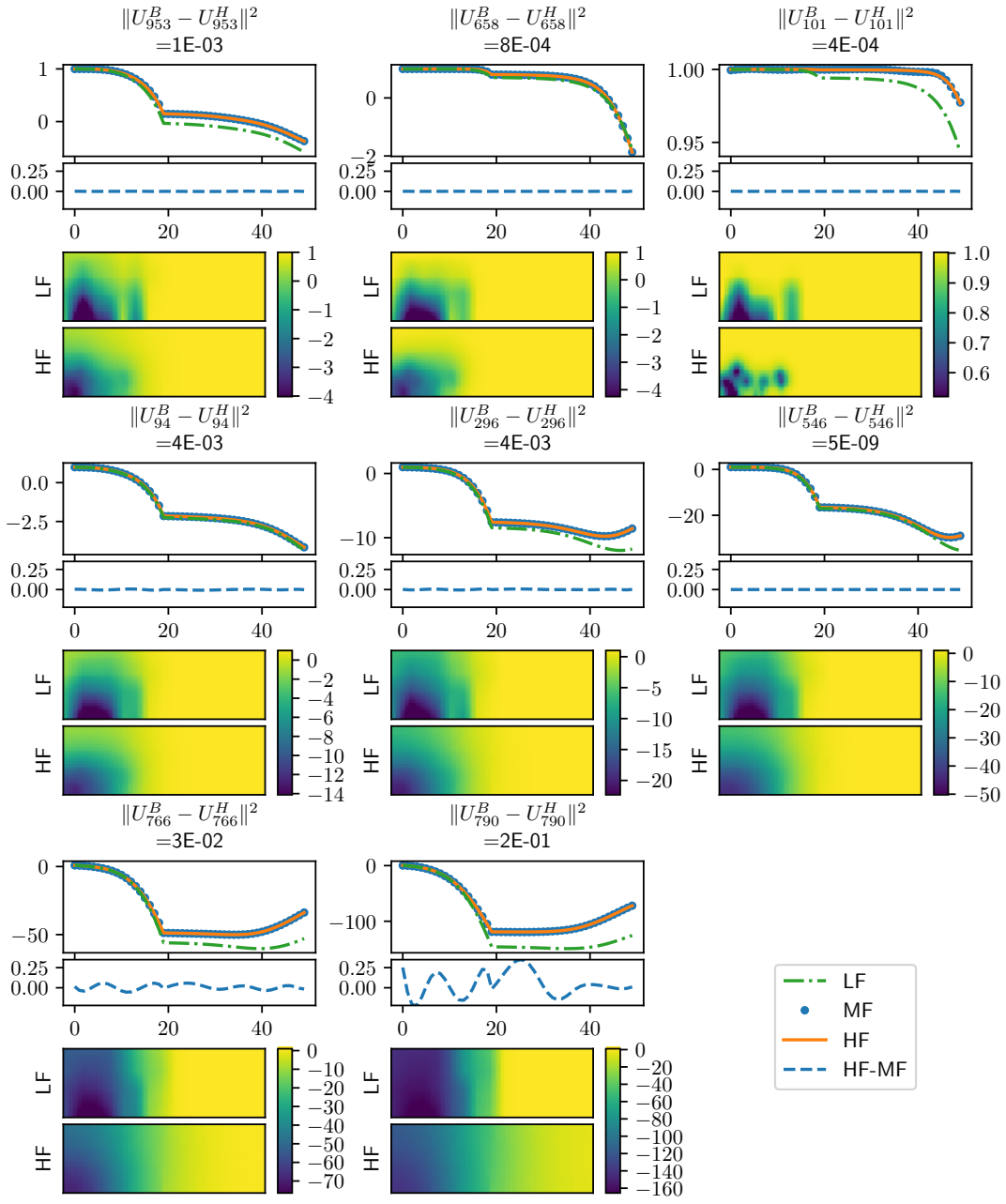


Figure 4.18: LF, MF and HF for 8 random inputs. For each sample we show the water level timeseries (top), error timeseries (middle) and 2D spatial maps of water level for the top layer at the selected timestep (bottom). HF runs=82, mean L^2 for LF= 1.00e+28, mean L^2 for MF= 1.14e-03, mean relative L^2 for MF= 1.81e-05, condition number for plots=3e+16, HF condition number for plots=4e+23

4.5 Conclusions

While lower fidelity, faster models may lack detail and accuracy, they often capture important features of the underlying system. This fact is exploited by a novel method, multi-fidelity stochastic collocation, to create an approximation of a high-fidelity model based on low-fidelity simulations by combining samples of both in a bi-fidelity surrogate. Multi-fidelity stochastic collocation is a valuable tool for efficient analysis of complex models, such as groundwater flow simulators. The method is non-intrusive and can be applied to a variety of lower-fidelity models. The surrogate can be used for demanding applications such as sensitivity analysis, uncertainty propagation, integrated modeling, optimization and decision support. It allows the efficient use of computational resources, as the required samples are independent and easily parallelizable. A major advantage of the approach is its robustness. The sampling is not adaptive (like for example the PCE algorithm used in Chapter 3, so samples for which the model fails to execute can be ignored. Furthermore, additional high and low-fidelity pairs can be added to the training set and the surrogate retrained at minimal cost. Since low fidelity runs are required for each surrogate run, there is a process based output to compare to the surrogate output. This is valuable to boost confidence in the surrogate enabled predictions where there is concern that the training set may not adequately capture variability present during prediction. The first step in the algorithm involves a large sample of low-fidelity runs to inform which samples are run through the high-fidelity model. This provides an additional layer of analysis for modellers, similar in function to k-mediods, identifying a few samples of the parameter space that span the variability present in the outputs of interest. The method is particularly well suited to the approximation of spatially and temporally varying output, from which particular quantities of interest can be computed.

As demonstrated by our case study (see Section 4.4), as few as 10-30 high-fidelity model runs may be required to satisfactorily construct the surrogate, orders of magnitude fewer than would be required for applications such as optimization, uncertainty analysis, or decision support. By satisfactorily we mean that the order of magnitude of error in the MF surrogates from the above results (around 1×10^{-5} for the relative

L^2 error) is negligible compared to error from other sources such as physical discretization (the low-fidelity outputs had a relative L^2 of 1×10^{25} for example). Thus despite the inaccuracy of the low-fidelity model, a bi-fidelity surrogate can achieve similar accuracy to higher-fidelity using a smaller number of simulations. Computational cost for each run required for such analysis can then be reduced to that of the low-fidelity model. In the above case study this is $1/50$ the cost of a high-fidelity simulation.

We have identified two novel improvements to the algorithm in practical settings. Firstly, low-fidelity outputs should be normalized for node selection so as to prevent bias towards outputs of larger magnitude. Second, as much information from the low-fidelity simulator should be included in the output as is practical. Significant improvement was noted by including more outputs of the same underlying model.

While the approach shows much promise, several limitations are of note. Accuracy of the surrogate depends on the discrepancy between low-fidelity and high-fidelity models. For a given number of high-fidelity simulations, the accuracy of bi-fidelity surrogate increases as the discrepancy decreases. The present algorithm relies on a large number (1000) of low-fidelity model runs. Even if these are significantly cheaper than high-fidelity runs, this cost may be prohibitive where a fast low-fidelity model cannot be found. As noted by Peherstorfer et al. [2018], complex relationships between models of different fidelities are common, limiting the utility of most multi-fidelity methods currently available.

The field remains ripe for further research. Few of the multi-fidelity methods surveyed by Peherstorfer et al. [2018] have been applied to groundwater simulation, and recent work [e.g. Man et al., 2020] is making significant progress towards emulating more complex models.

4.6 Appendix

```
"""
Boilerplate for running the multi-fidelity stochastic collocation algorithm using a simple diffusion example
authors: John Jakeman (jdjakem@sandia.gov) and Michael Asher (michael.james.asher@gmail.com)
date: August 2015
"""
import numpy
import matplotlib.pylab as plt
"""
Import
    * HF_model and LF_model: high-fidelity and low-fidelity models
    * sample_inputs: function to produce samples from input distributions (for building and testing)
    * select_nodes: function to select collocation nodes
    * synthesis_operator: function to compute surrogate
"""
from ellip import HF_model, LF_model, sample_inputs
from select_nodes import select_nodes
from synthesis_operator import synthesis_operator
# Settings
# number of initial candidates/snapshots for low-fidelity model
num_lf_candidates = 100
# number of low-fidelity, multi-fidelity, and high-fidelity runs to do for testing
num_test_samples = 100
# number of interpolations nodes/high-fidelity runs
num_hf_runs = 3
# 1. Evaluate the low-fidelity model u^L on a candidate set Γ.
candidate_inputs = sample_inputs(num_lf_candidates)
lf_candidate_values = LF_model( candidate_inputs )
# 2. Choose an ordered subset of N nodes γ using Algorithm 1.
pivots, L  = select_nodes(V=lf_candidate_values.copy(), N=num_hf_runs)
selected_inputs = candidate_inputs[:,pivots]
lf_selected_values = lf_candidate_values[:,pivots]
# 3. Evaluate the high-fidelity u_H model on γ.
hf_selected_values = HF_model( selected_inputs )
# 4. Use u_H(γ) to construct the interpolation operator
#    and evaluate at any z using Algorithm 2 with input data v = u^L(z).
test_inputs = sample_inputs(num_test_samples)
lf_test_values = LF_model( test_inputs )
mf_test_values, condition_number = synthesis_operator(lf_selected_values, hf_selected_values, L, lf_test_values)
hf_test_values = HF_model( test_inputs )
print('condition number= %.2e, 1/machine eps= %.2e' % (condition_number, 1./numpy.finfo(float).eps))
print("||HF-LF|| = %.2e" % (numpy.mean(numpy.linalg.norm(hf_test_values-lf_test_values, axis=0)) / numpy.sqrt(hf_test_values.shape[0])) )
print("||HF-MF|| = %.2e" % (numpy.mean(numpy.linalg.norm(hf_test_values-mf_test_values, axis=0)) / numpy.sqrt(hf_test_values.shape[0])))
# plot a few random LF, HF, and MF samples
rand_i = numpy.random.randint(num_test_samples, size=(6))
for i in range(len(rand_i)):
    plt.subplot(numpy.ceil(len(rand_i)/3.), 3, i+1)
    plt.plot(lf_test_values[:, rand_i[i]], '--', label="lf")
    plt.plot(hf_test_values[:, rand_i[i]], '-', label="hf")
    plt.plot(mf_test_values[:, rand_i[i]], 'o', label="mf")
    plt.title('sample '+str(rand_i[i]))
plt.legend()
plt.show()
# plot all LF candidates
for l in range(num_lf_candidates):
    plt.plot(lf_candidate_values[:,l], color='grey', alpha=0.3)
# plot chosen LF and HF
cm = plt.get_cmap('gist_rainbow')
for l in range(num_hf_runs):
    plt.plot(lf_selected_values[:,l], 'x', color=cm(1.*l/num_hf_runs))
    plt.plot(hf_selected_values[:,l], color=cm(1.*l/num_hf_runs))
plt.title("Selected nodes")
plt.show()
```

Figure 4.19: Boilerplate for running the multi-fidelity stochastic collocation algorithm using a simple diffusion example.

```
"""
Multi-fidelity solver for a 1D elliptic PDE with random coefficients and random forcing
Originally from Mike Giles' MCMC code.
authors: John Jakeman (jdjakem@sandia.gov) and Michael Asher (michael.james.asher@gmail.com)
date: August 2015
2nd order central difference discretization of (c u')'(x)= - 50 Z^2, u(0)=0, u(1)=0
where Z = N(0,1)                    (Normal with unit variance)
and c(x) = 1 + a*x with a = U(0,1)  (uniformly distributed on (0,1))
"""
import numpy
import scipy.sparse
import scipy.stats
class ellip():
  # set up equation matrices
  def __init__(self, nf, num_QOI):
    hf = 1./nf
    cf = numpy.ones((nf))
    A0f = hf**(-2)*scipy.sparse.spdiags([cf[1:], -cf[1:]-cf[0:-1], cf[0:-1]],[-1.,0.,1.],nf-1,nf-1)
    cf = (numpy.arange(1.,nf+1.)-0.5)*hf
    A1f = hf**(-2)*scipy.sparse.spdiags([cf[1:], -cf[1:]-cf[0:-1], cf[0:-1]],[-1.,0.,1.],nf-1,nf-1)
    self.cf = numpy.ones((nf-1))
    self.A0f = A0f
    self.A1f = A1f
    self.num_QOI = num_QOI
  # interpolate with grid of num_QOI cells
  def post_process(self, uf):
      nf = len(uf)+1
      hf = 1./nf
      n_HF = self.num_QOI + 1
      h_HF = 1./n_HF
      # note boundary conditions (0.) are left off ends, add them here
      return numpy.interp(x=[h_HF*(i+1) for i in range(n_HF-1)], xp=[hf*(i+1) for i in range(-1, nf)], fp=numpy.concatenate(([0.0], uf, [0.0]
  # solve equation for given random variables
  def run(self, x):
    U, Z = x
    uf = scipy.sparse.linalg.spsolve(-(self.A0f+U*self.A1f), (50.*Z**2*self.cf))
    return self.post_process(uf)
  # solve equation for each column
  def bulk_run(self, input_samples):
    num_samples = input_samples.shape[1]
    output_values = numpy.empty( (self.num_QOI, num_samples), float )
    for i in range( num_samples ):
      output_values[:,i] = self.run( input_samples[:,i] ).squeeze()
    return output_values
# set up two fidelities
n_HF  = 2 ** 9
h_HF  = 1./n_HF
n_LF = 2**(2)
h_LF  = 1./n_LF
num_QOI = n_HF - 1
num_dims = 2
HF_model = ellip(nf=n_HF, num_QOI=num_QOI).bulk_run
LF_model = ellip(nf=n_LF, num_QOI=num_QOI).bulk_run
# should return (num_dims, num_samples)
def sample_inputs(num_samples):
    input_samples = numpy.random.RandomState().uniform(0.,1.,(num_dims, num_samples))
    input_samples[1,:] = scipy.stats.distributions.norm(loc=0., scale=1.).ppf(input_samples[1,:])
    return input_samples
if __name__ == '__main__':
  num_samples = 6
  zs = sample_inputs(num_samples)
  HFs = HF_model(zs)
  LFs = LF_model(zs)
  import matplotlib.pylab as plt
  for i in range(num_samples):
    plt.subplot(numpy.ceil(num_samples/3.), 3, i+1)
    plt.plot(HFs[:,i], label="HF")
    plt.plot(LFs[:,i], '--', label="LF")
  plt.suptitle("Low and high fidelity elliptic equation solutions for a few random inputs.")
  plt.legend()
  plt.show()
```

Figure 4.20: Multi-fidelity solver for a 1D diffusion equaion with random coefficient and forcing.

Chapter 5

Deep Learning

5.1 Introduction

In recent years, Deep Learning (DL) models, composed of many Neural Network (NN) layers, have come to dominate the fields of artificial intelligence and machine learning [Schmidhuber, 2015]. The popularity of this branch of machine learning is surging as measured by listed jobs [Aiken et al., 2017], academic citations (eg. LeCun et al. [2015]) or the public activity of the worlds biggest companies, many of which have published papers and software on the topic recently (eg. Apple [2019]; Amazon [2019]; Google [2019]; Microsoft [2019]; Tencent [2019]; Baidu [2019]).

The success of these statistical methods is in large part due to software improvements made possible by huge communities of open source contributors. While their theoretical basis is poorly understood [Lin et al., 2017; Poggio and Banburski, 2020], DL frameworks provide simple interfaces to enormous hardware and software resources.

However, this progress has not been mirrored in the physics based modelling software typically used by applied mathematicians. A cursory investigation of public open source code libraries reveals at least a 100 to 1 ratio of users and contributors when comparing deep learning libraries [e.g. Google, 2019] to popular packages related to physics based modelling [e.g. Bakker et al., 2016; OpenTurns; OpenBLAS].

However while the field of applied mathematics has few state of the art software packages, it does have many decades of theoretical research, which is yet to be integrated into DL. For example optimization, regularization, functional forms and multiscale methods have been studied in great detail.

Marçais and de Dreuzy [2017] and Shen [2017] have noted the relevance of DL to hydrology, an area of research where process driven models have been the norm. While we agree with their suggestions about where DL can be applied, there are significant unmentioned possibilities. One of the aims of this paper is to identify specific ways in which applied mathematicians working in hydrology can engage with the DL community in a mutually beneficial manner.

It is worth noting that most DL applications have to date involved infilling missing data rather than extrapolating or predicting in unseen conditions [Marcus, 2018]. The seminal works cited by LeCun et al. [2015] all involve interpolation. Data for training and testing are randomly drawn from the same dataset without partitioning on any known variable. Both contain images of similar quality and subject.

A notable exception is the field of transfer learning, which uses models trained on data from one domain, as inputs to a model predicting in another domain. See Tan et al. [2018]'s survey for more details on the topic. It is also worth noting also that DL methods have performed well in scenarios that might appear to be extrapolation, such as a hydrological model predicting on catchments not present in the training data [Nearing et al., 2020]. The common approach to interpolating at external catchments is a form of ad-hoc machine-learning, selecting a similar catchment from the training dataset [Parajka et al., 2013].

Difficulty with extrapolation is of course a limitation of statistical methods [Marcus, 2018]. However, in our opinion data availability has outpaced model development in many fields, such as hydrology. This increases the relevance of data intensive approaches. Traditional models such as MODFLOW [Harbaugh, 2005] have not been developed to make full use of the gamut of data now available from for example the Sentinel program [El Hajj et al., 2017]. Further, one of the arguments we present here is that the hardware and software advances brought about by DL research are not limited to statistical approaches.

5.1.1 DL and Groundwater

Recently, DL techniques have been used to train surrogates of groundwater models. Zhu and Zabaras [2018] perform Bayesian Uncertainty Quantification on a 4225 parameter flow model using a CNN using only 512 runs of the underlying process based model. They employ Image-to-Image Regression, also known as fully convolution neural networks (FCN), which involves pixel level predictions not the more common task of image level classification. The surrogate has 241,164 parameters. A model setting with this ratio of parameters to data is clearly not focused on the principals of identifiability and parsimony [Guillaume et al., 2019]. The number of parameters is enormous compared to statistical approaches typical in hydrological modelling. Razavi et al. [2012b]'s review of surrogate modelling in hydrology found 85% of models in the literature have less than 20 parameters. However, this number of parameters is small in the DL literature. Forms of SegNet [Badrinarayanan et al., 2017] for example have over 1.6 million parameters. This gap highlights the difference between the two schools of thought. The DL community has focused on empirical results and intense computation. Numerical modellers have typically focused on systematic sensitivity and uncertainty analysis.

Zhu and Zabaras [2018] mapped permeability to steady state head, but suggested their approach can be applied to varying boundary conditions by simply appending additional input channels (or images). They suggested a similar approach for dealing with time as an additional input channel. In a follow up work, Mo et al. [2019] do not incorporate time in this way - instead approximating head at a given time as a function of head at the previous time, the hydraulic conductivity and the boundary conditions. All of which are represented as images in the spatial domain.

Much has been written (eg Nabian and Meidani [2018]) on creating DL based surrogates of physical models. The work of Mo et al. [2019] is of particular interest as it demonstrates a surrogate which emulates the full spatial-temporal nature of the original model as well as the key input-output relationships. It does this while maintaining the previously extolled virtues of DL surrogates - namely computational power and software quality. If such surrogates are broadly applicable they may be

used to replace the underlying models altogether. Recent work by Sun et al. [2020] provides a method for enforcing physical laws (PDEs) in DL architectures without using any simulation data. Since most DL approaches were developed for training on noisy "real-world" data. Future research will likely focus on models trained using both this data and physical systems based constraints.

5.1.2 The Band Wagon

NNs have been around since the 1950's [Rosenblatt, 1958], and have been applied in hydrogeological setting since at least the 1990's [Ranjithan et al., 1993]. Their potential is evident in their flexibility, Cybenko [1989] demonstrated that any continuous function could be approximated by a single hidden layer network. However, their recent popularity and successful application can be explained in large part by the

- Relative simplicity of use of DL libraries, eg PyTorch; Google [2019]. These "black-box" methods can be applied to most datasets with little modification. The libraries contain equation implementations, optimization algorithms, analysis tools and many examples of models and training data.
- Development of computational resources capable of training large networks, making them suitable for use with increasingly large datasets and computational resources.. In particular the aforementioned DL libraries provide an easy way to program GPUs [Ren et al., 2019].
- Availability of large training datasets, for example Deng et al. [2009] is a freely accessible database of millions of classified images.
- Research matching particular structures (interconnections, activation functions, loss functions and calibration methods) to particular problem domains. For example back propagation LeCun et al. [1990] advanced the performance of image classification networks considerably and convolution NNs were found to be easier to train for image recognition.
- Application of regularization techniques. For example dropout [Srivastava et al., 2014], unsupervised pre-training [Bengio et al., 2007; Erhan et al., 2010]

or vanishing gradient regularization [Pascanu et al., 2013] and early stopping.

- Techniques to artificially increase the size of a training dataset by deformation. Clearly no new information is added via this route, but it has been shown to assist training models on a given dataset Mikołajczyk and Grochowski [2018].

- "Knowledge transfer" from models calibrated on large datasets to others fine tuned on smaller datasets [Tan et al., 2018]. There is an inherent risk that such calibrated models can be misused without due diligence by others, and "knowledge" may be a misnomer for the coefficients of what are typically black box models. We believe that a tighter integration between classical numerical fields and DL research will shed some light on model internals.

5.1.3 Value of DL Methods

The aim of this chapter is to explore the potential for DL methods to address challenges faced by groundwater modellers in practice. Specifically the problem setting of simulating water level and flow based on readily available data. Commonly, what is available is water levels at sparse locations and proxies (eg rainfall, temperature) of boundary conditions and forcing conditions (eg neighboring systems, rivers, recharge, pumping). Data on aquifer properties is so sparse compared to aquifer heterogeneity that it is often used only to initialize or constrain calibrated values. Of special interest in this problem setting is forecasting into conditions for which data is not available (dramatically increased pumping, climate change, fracking, subsidence).

We find that there are three broad areas where DL methods stand out among other surrogate approaches of interest to groundwater practitioners.

Firstly, DL models have been shown to perform well (i.e. surrogate induced error is orders of magnitude less than that from other uncertainties) as "black-box" surrogates [Sun, 2018; Mo et al., 2020; Dagasan et al., 2020] which can be used to replace a slower model without changing the underlying code in a variety of prediction ad inversion settings. Notably, DL methods are capable of handling orders of magnitude more parameters than other surrogate approaches. Mo et al. [2019] demonstrate the use of a DL surrogate on a problem with 686 uncertain input parameters, whereas

Razavi et al. [2012b] found 85% of surveyed surrogate methods dealt with less than 20. In Section 5.4.1 we show the performance of a DL "black-box" surrogate.

Secondly, DL architectures vary far more so than other surrogate methods. This correlates with the huge quantity of publications on this subject in general (not yet on their application to hydrology and hydrogeology). The opportunity for groundwater modelling is that architectures can be adapted to particular model settings, to deal with for example a greater variety of boundary conditions or out-of-sample forecasting [Sun et al., 2020; Wang et al., 2020; Geneva and Zabaras, 2020]. In Section 5.4.2 we show the performance of a DL surrogate in a variety of modelling conditions on a practical groundwater model.

Thirdly, it has been speculated [Marçais and de Dreuzy, 2017; Shen, 2017; Nearing et al., 2020] that DL methods could be used beyond emulation to provide additional insight that process based models could not. However, to our knowledge, no studies have produced results in this area. In Section 5.4.3 we provide a couple of results showing the potential in opening up the "black-box" of a DL surrogate's inner workings.

5.1.4 Structure

Our major contributions of this chapter are as follows. We explain the particular aspects of DL methods, as opposed to statistical methods in general, that make them of interest to groundwater modelling. We enumerate opportunities for knowledge transfer between DL and physics based modelling in Section 5.2. Then in Section 5.3 we apply both simple and state of the art DL models as surrogates of groundwater problems showing 1) their strength as an emulation method, 2) some of their limitations in practice and 3) their potential to provide new kinds of insight not possible with existing approaches.

5.2 Opportunities

5.2.1 Software

In our view, the simplest, most productive approach for applied mathematicians who wish to benefit from DL is to use one of the software frameworks that are synonymous with the field. TensorFlow [Google, 2019], CNTK [Microsoft, 2019], PyTorch [PyTorch] and Caffe2 [Caffe], in descending order of popularity on Github at time of writing, are all open source software libraries built for DL. They allow models to be designed, calibrated and analyzed on state of the art computational resources (at present GPUs) using in the order of a hundred lines of a high level language (typically Python). This is a fundamental shift from, for example MODFLOW [Harbaugh, 2005], the customization of which might require editing thousands of lines of low level code. Current DL approaches cannot replace process driven models in many of the applications that require extrapolation, since in many situations there are insufficient methods to evaluate confidence in their predictions. For example they have not gained widespread adoption in weather forecasting Dueben and Bauer [2018]. Given the investment in DL research we posit this will change as the field matures, for example as we develop an understanding of the range of predictions that can be made from DL models with comparable measured performance.

DL software is of great use to applied mathematicians even if they have no interest in NNs. Apart from the NN specific code, each of these libraries provide generic functions which allow the implementation of arbitrary numerical algorithms. The frameworks are based on GPU accelerated tensor (multi-dimensional array) computation, but contain a great variety of numerical tools. Hence even a physics based model implemented in a DL framework, though it might not use NNs at all, will still benefit from simple to use software of a high quality, high performance computing and a large community of algorithms, models and datasets. For example, a simple PDE solver based on Burkardt [2013], re-implemented in PyTorch is attached in 5.11. A more complex example of coupling a physics based groundwater model with a DL approach is given in Section 5.4.2.

5.2.2 Remote Sensing

The potential of remote sensing data for hydrogeological modelling is largely unrealized [Brunner et al., 2007], and the same is true of many areas of geophysical applied mathematics. This is problematic as remote sensing datasets are the only source of data on the same physical scale as many of these models. Their application is often limited since remotely sensed variables do not directly map to the parameters of process based models. As mentioned in Section 5.3.1, Convolution NNs and Recurrent NNs have been successfully applied to imagery processing and timeseries modelling respectively. Many of the advances in training models on large datasets of imagery could be transferred to remote sensing, and indeed Zhang et al. [2016]; Zhu et al. [2017]; Yuan et al. [2020] have reviewed work in this area. We posit that the intersection between remote sensing data and deep learning models is the most promising area for research in hydrological modelling.

Continued increases in the number of sensors and open data initiatives has created global scale datasets of hyper-spectral imagery, elevation and synthetic-aperture radar, as well as continental datasets of airborne electromagnetics, magnetics, gravity, magnetotellurics, radiometrics and seismics. Progress has been made in using remote sensing in hydrological modelling. van Dijk and Renzullo [2011] review data assimilation and model evaluation methods which have used remote sensing data to improve accuracy and enhance spatial resolution. They reported varying degrees of success, but analysis is limited to "surface" level relationships such as the location of groundwater discharge zones, surface water levels, topsoil moisture content, soil hydraulic properties, snow cover, leaf area index, biomass and precipitation. Arguably the benefit of DL would be in discovering more complex interactions. One can sketch a rough relationship between reflectance values and soil moisture, but could one do the same between the reflectance values and geological classification?

A simple opportunity for applied mathematicians to utilize methods common to DL, is to preprocess or synthesize remote sensing imagery as inputs or parameters for models, regardless of that model's form (NN, statistical, PDE or otherwise). Relatively little has been done to investigate the possibilities here, for example in replacing

traditional interpolation methods such as kriging with a DL model incorporating remote sensing data to predict soil or aquifer properties.

5.2.3 PDE methods

Despite being framed as statistical methods, DL models in practice may share many of the same structures as physically based models Mehta and Schwab [2014]. Convolutions, the foundation of many DL libraries, are also a foundational data structure in the implementation of PDE based models. Marçais and de Dreuzy [2017]; Lin et al. [2017] conjecture that the success of DL models may be due to their ability to replicate physical processes. There are examples of using DL to solve PDEs [Rudd, 2013; Weinan et al., 2017] and applying PDE theory to improve DL models [Haber and Ruthotto, 2017; Chaudhari et al., 2017]. While these links are clearly conjecture, at least it makes clear the potential for PDE models to be implemented in DL related software - which would in and of itself yield significant benefits.

While the suite of computational advances labelled DL have had great practical success, their mathematical theory is not well developed Vidal et al. [2017]. Given the novelty of the field, there is likely great promise in the transfer of decades of applied mathematics research, particularly regarding PDEs, to the DL community. For example super-resolution methods Shi et al. [2016] have clear parallels with multifidelty methods such as those discussed in Chapter 4.

While still in their infancy, more complicated NN nodes would benefit disproportionately from this knowledge transfer. For example Amos and Kolter [2017] recently developed a NN where each node solves a constrained optimization (quadratic programming) problem. Such developments will increase the ease with which process driven models can be implemented in DL frameworks.

5.2.4 Model Structure

More speculative than areas noted above is the prospective use of DL to infer physical model structure. By design, DL architectures are flexible. The weights and activation functions turn internal nodes on and off during calibration. Internal nodes can be

composed of any function for which a gradient can be approximated. There is an open opportunity for research into new functional representations. By analyzing trained NNs, Marçais and de Dreuzy [2017] and Shen [2017] both note potential for DL to guide the choice of the structure of physical models. Shen [2017] gives examples where DL has already been used to identify patterns that are not represented in physical models.

In the broader DL literature, Bengio et al. [2007] demonstrated the effectiveness of unsupervised pre-training, implying the capture of an underlying structure, and LeCun et al. [2015] claim it will be an increasingly important area of research.

In some cases, DL may have the potential to improve the parameters of existing process driven models. For example Badrinarayanan et al. [2017] successfully used remote sensing data to spatially segment and classify an aquifer. The segmentation would be based on remote sensing values, and would not require knowledge of the aquifer properties. Current common practice is to use zones often hand drawn by experts based on historically available data.

More speculatively, it may be true that correlations unearthed by DL models trained on copious volumes of remote sensing data may encode internal variables that can be built into new or altered process based models [Nearing et al., 2020]. This is not as outrageous as it sounds. While hydraulic conductivity is a property that can be physically measured, any practicing hydro-geologist will note there is little in common with the measured property and the parameters of that name in existing groundwater models. A first step in future research could explore the potential of DL to automate and improve the prediction of spatial distributed model inputs (e.g. acquifer properties) from the availalbe data (e.g. geological facies maps and bore logs).

5.3 Methods

5.3.1 Neural Networks

A neural network model consists of J interconnected nodes (neurons), each of which can be written simply as

$$y_j = f\left(\sum_{i\in 1,...,I} w_{i,j}x_i + b_j\right) \qquad j \in 1,...,J \tag{5.1}$$

where y_j are the outputs, x_i the inputs, $w_{i,j}$ are the weights, b_j the bias and f is a non-linear function. Common examples of this so called activation function or so called "non-linearity" include the softmax function

$$f(x_j) = \frac{\exp x_j}{\sum_{i\in 1,...,J} \exp x_i} \qquad j \in 1,...,J \tag{5.2}$$

or the rather linear rectified linear unit (ReLU)

$$f(x_j) = max(0, x_j). \tag{5.3}$$

In a classic fully connected network, the nodes are arranged in layers where every node in a layer takes as input the outputs of all nodes in the previous layer. Deep Learning [LeCun et al., 2015] refers to the calibration of NN models with many (more than several) internal (hidden) layers between the model's inputs and outputs of interest. While this structure is nothing more than logistic regression models arranged in both series and parallel, particular structures of simple nodes have been developed to model complex problems in a number of domains.

Convolution NNs [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014] have had huge success in computer vision. In convolution NN layers, input sequences x_i are commonly pixels of an image, and the weights are sparse and shared between nodes in the same layer, such that Equation (5.1) amounts to a convolution of the vectors of inputs and weights.

Recurrent NNs, in particular Long Short-Term Memory networks [Sutskever et al., 2014], have produced unprecedented results for sequential data such as speech recog-

nition, language translation and timeseries modelling. Recurrent NNs create loops by adding nodes which maintain a state c_j, which is updated at each computation based on its previous value as well as the inputs. Layers are not typically fully connected nor are they linear, i.e. connections may skip layers.

5.3.2 Architectures

Much of the research into DL has focused on network architecture, and its impact on accuracy and training time for a given dataset. While there is no universal taxonomy of architectures as a whole, a number of well known submodules of several connected layers have surfaced and typically encoded in DL software packages such as Google [2019]; PyTorch. We focus here on image-to-image (or fully convolutional) architectures which have a multi-dimensional image as an input and output. Such configurations are common in literature relating to semantic segmentation [Minaee et al., 2020] and Generative Adversarial Networks (GAN) Gui et al. [2020]. These image-to-image methods appear most relevant to groundwater simulation where much of the data is multi-dimensional.

Here we draw from two popular image-to-image architectures. DenseED Zhu and Zabaras [2018] and ConvLSTM Shi et al. [2015]; Geneva and Zabaras [2020].

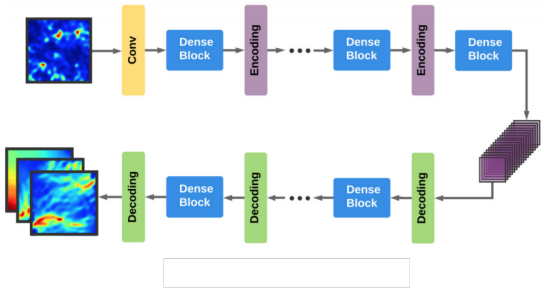
5.4 Results

5.4.1 DL as a simple surrogate

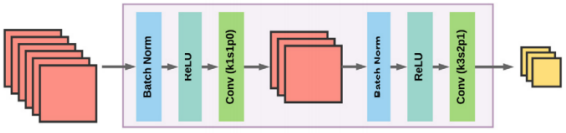
This section contains the training of a Convolution NN on two PDE based models, including a groundwater model implemented in MODFLOW Harbaugh [2005]. The purpose of experiment is to demonstrate the ease of use and power of DL software to encourage applied mathematicians to use the software for their own research.

First, we implement an explicit finite difference solution to the two dimensional heat equation

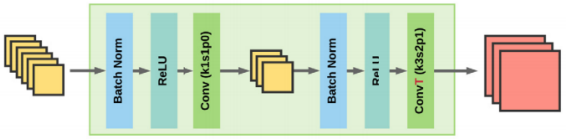
$$\frac{\partial h(x,t)}{\partial t} = \nabla_x^2 h(x,t) + f(x,t), \qquad x \in D, t \in (0,T] \tag{5.4}$$



(a) full network



(b) encoder



(c) decoder

Figure 5.1: The architecture of DenseEd, figures from Zhu and Zabaras [2018]

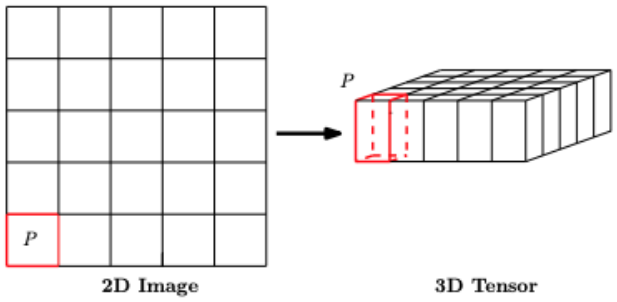
where $h(x,t)$ is temperature, though in application to groundwater systems, this is groundwater pressure.

We parameterize the initial head as the Karhunen-Loève expansion

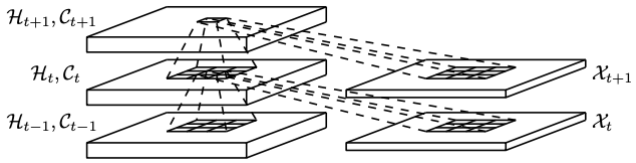
$$h(x) = \mu + \sigma \sum_{i=0}^{n} k_i(x)\lambda_i z_i, \tag{5.5}$$

where μ and σ are the mean and standard deviation of the field, $k_i(x)$ and λ_i are the eigenfunctions and eigenvalues, and z_i are the random weights.

The model in Equation (5.4), implemented in Figure 5.11 of the Appendix, is then solved for T timesteps for $10,000$ realizations of Equation (5.5) by sampling random values for the weights z_i. We then trained a convolution NN, implemented in Figure 5.12, with T identical layers. We used replication padding (extrapolating edge values to prevent convolution from reducing the image size) and a kernel diameter of three (a weighted sum of 3 x 3 neighbouring pixels is used in the convolution layer). Figure



(a) 2D image into a 3D tensor



(b) the inner structure of a ConvLSTM

Figure 5.2: The architecture of ConvLSTM, figures from Shi et al. [2015]

5.3 shows the convergence of the NN error with the number of training examples for $T = 1$ and $T = 5$. Two things of note from this example are that 1) the implementation of a convolution NN corresponds closely to that of a numerical PDE scheme and more significantly 2) that NN can be coded and trained on state of the art computers with unprecedented ease. The implementation of the PDE solver and NN are attached in Figure 5.11 and Figure 5.12 of the Appendix.

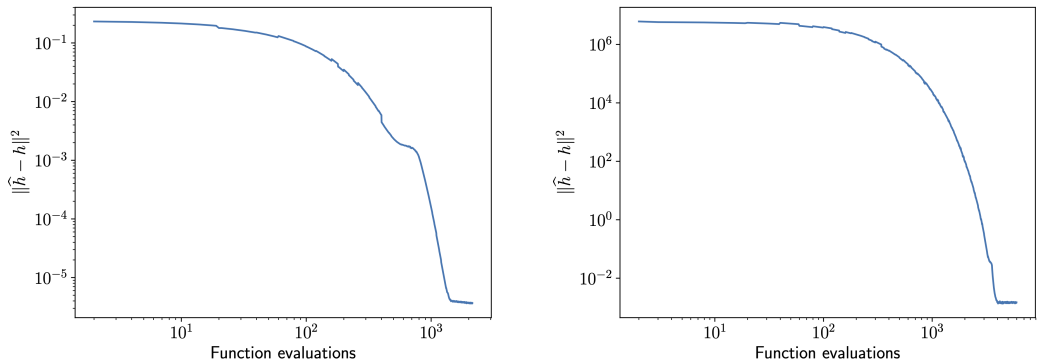


Figure 5.3: Convergence of convolution NN trained on heat model for one (left) and five (right) timesteps. The mean value of the solution was 2×10^{-2}.

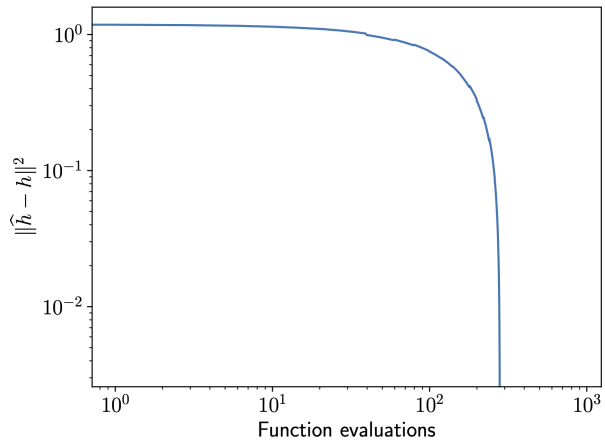


Figure 5.4: Convergence of convolution NN trained on MODFLOW model over 10 timesteps. The mean value of the solution was 2×10^2.

Similar convergence of errors can be seen on a surrogate of a simple MODFLOW model in 5.4. These results show the successful calibration of DL surrogates on PDE based models with hundreds of parameters and outputs, using simple "black-box" software, showing them to be a useful surrogate for groundwater practitioners.

5.4.2 Limitations of DL surrogates

The aim of this section is to extend the work of Mo et al. [2019] to a practical groundwater model. As a practical example we develop a model of similar complexity to those used in commercial or policy making settings. We construct a MODFLOW [Harbaugh, 2005] finite difference model of the Campaspe River basin in Victoria, Australia based on the work of Iwanaga et al. [2020]. Real data was incorporated for climate, elevation, groundwater levels and hydrological units.

Using the same popular FCN as Mo et al. [2019], but with additional channels, we predicted hydraulic head images at a given time based on images of hydraulic head at the previous timestep, hydraulic conductivity and spatial patterns of boundary conditions. To cover a range of scenarios we implemented a specified head boundary (Dirichlet or first-type) using the CHD package, specified flux boundary (Neumann or second-type) using the well package and head-dependent flux boundary (Robin

or mixed) using the river package. Figure 5.5 shows the images used to represent the spatial variability of the boundary conditions in the model. Hydraulic conductivity represented using a Karhunen-Loéve expansion. The model had 13 variable inputs including 10 Karhunen-Loéve modes and a scale for each of the three boundary conditions.

As seen in Figure 5.6, similar accuracies were achieved regardless of the boundary condition. The spikes in the convergence plots are an artefact of the random sampling during training. The specified flux model (middle) shows the least volatility - showing a simpler input-output relationship in this case. Further research is required into optimizing the hyper-parameters (parameters that control the calibration algorithm) that would lead to fast and reproducible calibration of DL surrogates in this setting. We improved the performance in all cases by adding the spatial variability of that boundary condition as an input - significantly so in the case of the fixed head boundary. This adds credibility not only to the use of FCN as surrogates but also to the idea that they may be able to represent the underlying physical processes in a similar fashion to the PDE based model. As seen in Figure 5.6, including the boundary condition as input only impacts the performance of this surrogate in the case of the fixed head boundary. In this model, only the fixed head boundary had a time varying component, which explains this artefact.

There are notable differences in the surrogate implemented here and those described in Chapter 2. Since this surrogate employs a DL package, [PyTorch], the author gained access to tested and professionally implemented software tools and algorithms. As Harris et al. [2020] note of Numpy, such tools play an essential role in research. Even more significantly, the efficiencies of the DL packages allow the calibration of a surrogate on a number of inputs many orders of magnitude higher than other approaches. Figure 5.7 shows the spatially distributed output of the surrogate compared with that of the full MODFLOW model. The mean of the absolute residual is 1.5 or about 1% of the head values. The mean head difference between the previous and current time steps was 13, so error was closer to 10% of that quantity.

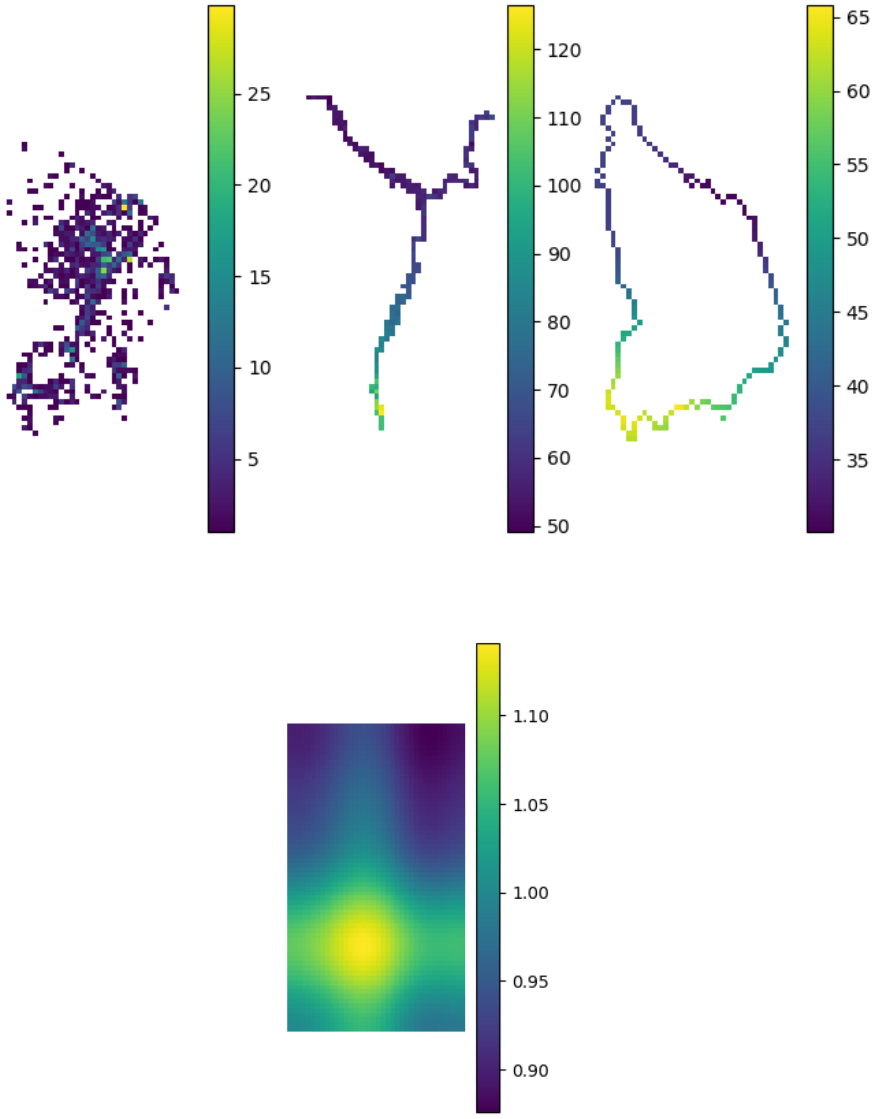


Figure 5.5: Top row: Spatial variability in well pumping, river stage and fixed head boundary conditions. Bottom row: An example spatial pattern of hydraulic conductivity.

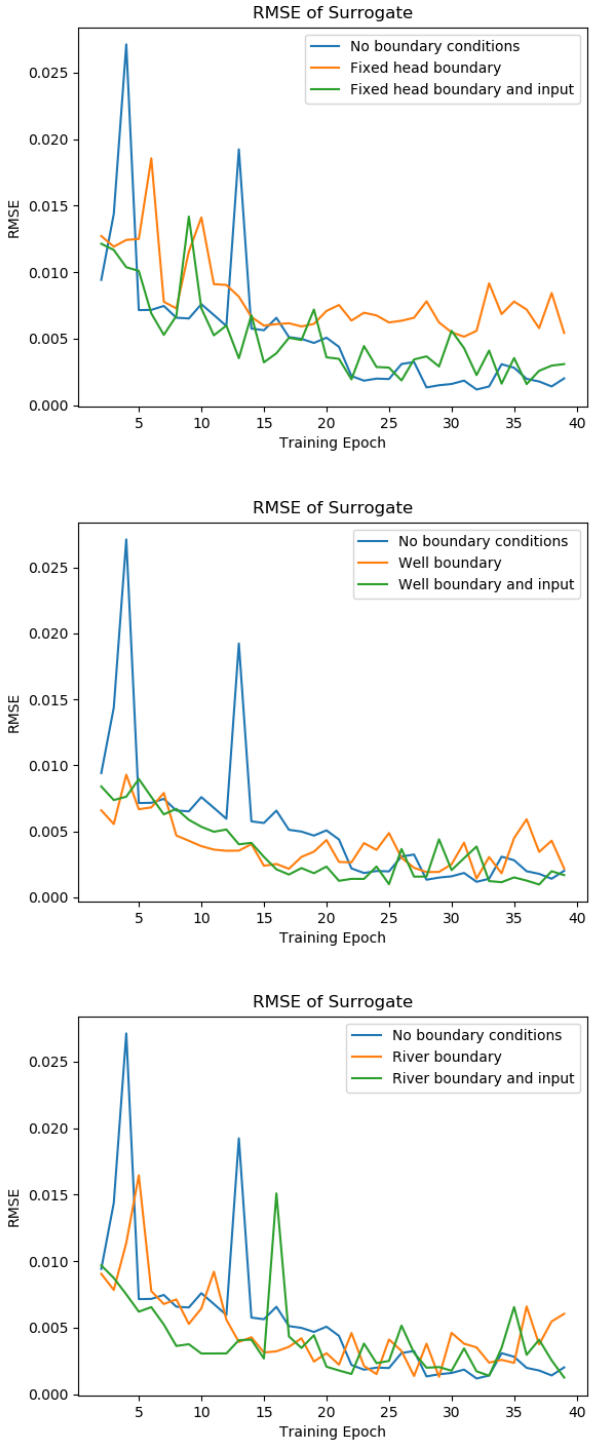


Figure 5.6: Convergence of surrogate root mean squared error (RMSE) with training epoch (set of 50 model input-output training runs). Shown here for a specified head boundary (Dirichlet type, top), specified flux boundary (Neumann type, middle) using the well package and head-dependent flux boundary (Robin type, bottom) using the river package.

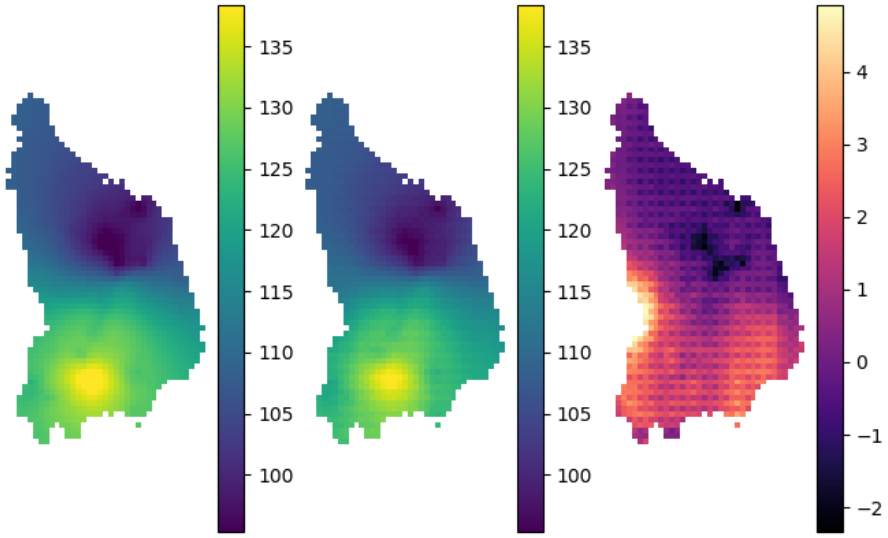


Figure 5.7: Example of hydraulic head observed (MODFLOW output) left, surrogate predicted (center) and the difference between the two (right).

5.4.3 What value do DL models provide beyond emulation?

As discussed above, DL methods stand out from other surrogate methods given their easy-to-use software, and ability to handle large numbers of uncertain inputs. In line with the previous literature on surrogate modelling, research continues into how certain network architectures perform better as surrogates for particular combinations of boundary conditions, quantities of interest and aquifer parameterizations. However, there is little except speculation [Marçais and de Dreuzy, 2017; Shen, 2017; Nearing et al., 2020] into how DL models might provide insight beyond their use as emulators. How can the recent surge in DL popularity translate into a step change in model ease-of-use, accuracy, reliability or resolution in earth systems modelling?

In this section, we provide a result that demonstrates how these methods can function fundamentally differently to other "black-box" surrogate methods. Due to the flexibility available in DL software packages, architectures can be implemented which mimic physical systems in ways the internals of other statistical methods

cannot.

The simplest example of this potential can be seen in calibrating a model with a single convolution layer. From the simple model in Section 5.4.1, with convergence results shown in Figure 5.3, we extract the internal weights of the model. The calibrated weights of the convolution layers are given in Figure 5.8. Ignoring the magnitude, which is adjusted by another factor in the model, the symmetrical shape of the weights correspond closely to the explicit finite difference scheme of the source model. This trivial example seeks to explain that the coefficients of DL models have the potential to represent more than simple black-box coefficients.

$$\begin{pmatrix} 0.0087 & 0.0853 & 0.0085 \\ 0.0851 & 0.6244 & 0.0855 \\ 0.0087 & 0.0853 & 0.0085 \end{pmatrix} \qquad \begin{pmatrix} 0.0097 & 0.0822 & 0.0097 \\ 0.0829 & 0.6310 & 0.0828 \\ 0.0090 & 0.0836 & 0.0090 \end{pmatrix} \tag{5.6}$$

Figure 5.8: Kernel of trained convolution NN for one (left) and five (right) timesteps.

A more practical example is constructed by fitting a state-of-the-art ConvLSTM [Shi et al., 2015] architecture to a MODFLOW model. Such recurrent architectures calibrate an internal state of the same dimensions as the input and output. By using images of head at subsequent timesteps as input and output, we fit a single ConvLSTM module model. The result, as shown in Figure 5.10 is that as the model error reduces, so increases the correlation between this internal state and the spatial distribution of hydraulic conductivity - the dominant aquifer property in this model. A simple "patchwork" spatial distribution of hydraulic conductivity (Figure 5.9 was chosen.

Such a method could be extended in practice to model the spatial distribution of unknown aquifer properties or boundary conditions based on the data which is most often regularly available - water level timeseries. This result is nebulous but an important step in providing the first (to our knowledge) feasible route for the use of DL methods beyond pure emulation. We speculate that truly novel modeling methods are possible by combining this approach with remote sensing data (digital elevation, magnetics, gravity, radar and hyper-spectral imagery).

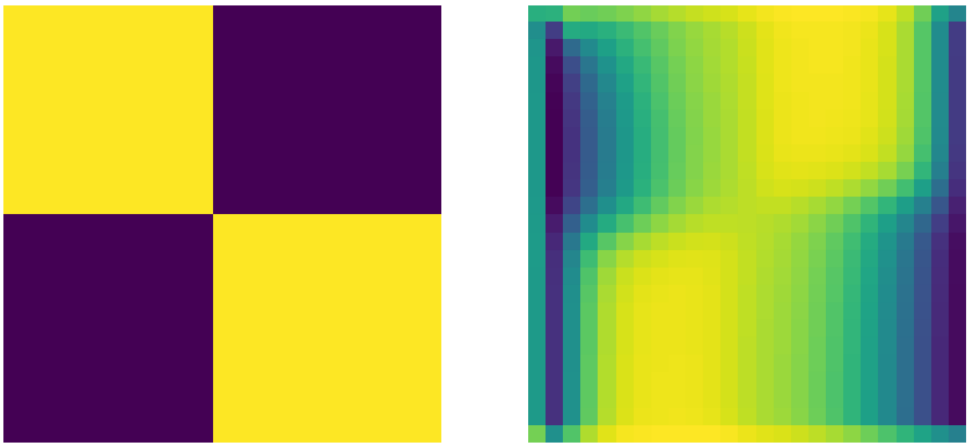


Figure 5.9: Images showing conductivity (left) and the internal state of the ConvLSTM model (right). Units are omitted as we only wish to show a correlation, which in this case has a Pearson coefficient of 0.5.

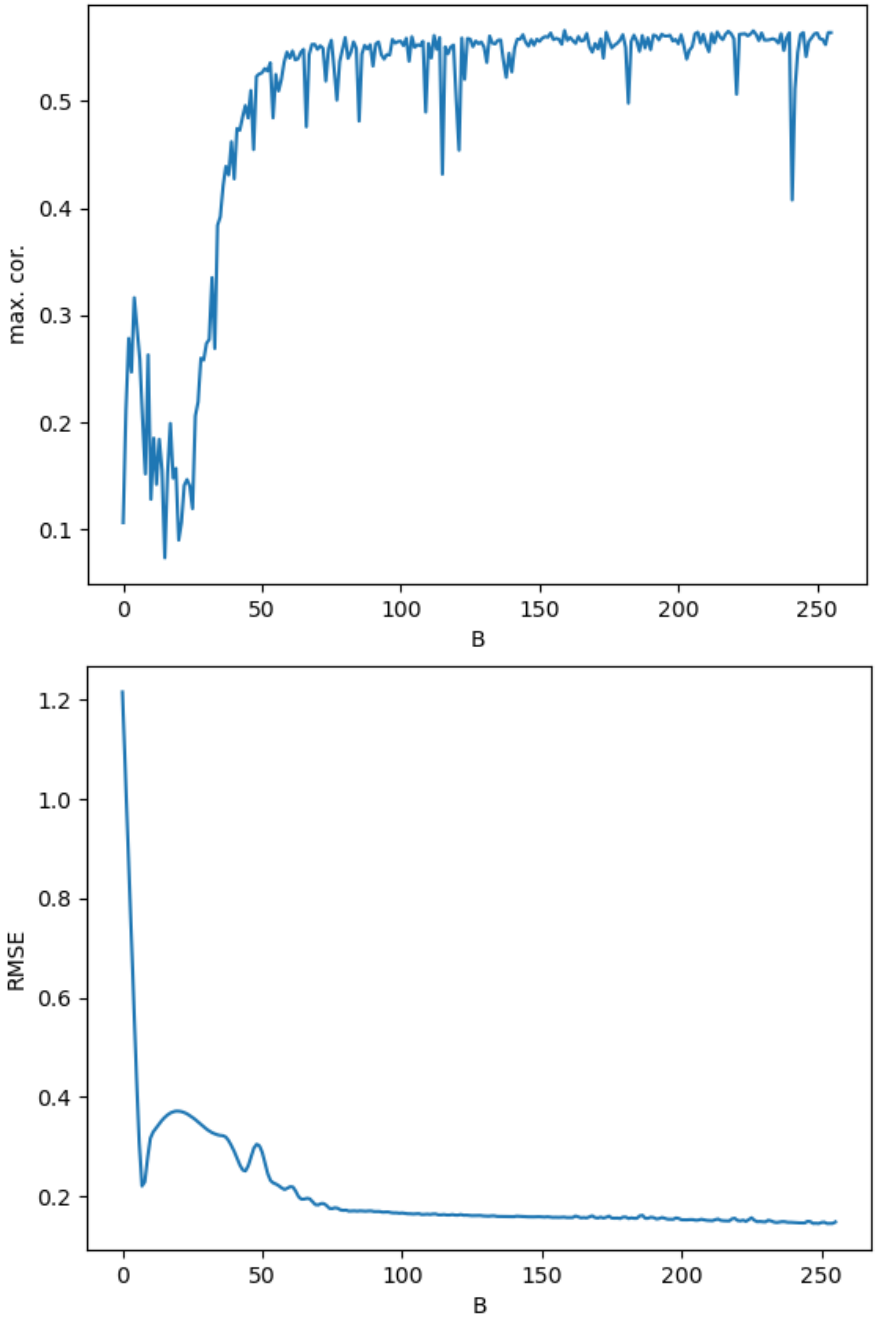


Figure 5.10: Convergence of the correlation (top) and error (bottom) of the ConvLSTM model with training batches, B, each of 100 data points. Correlation is measured by the Pearson coefficient, error by the root mean square error. The mean magnitude of the output is 3.5, the error converges around 6%.

5.5 Discussion and Conclusion

Recent advances associated with DL methods represent a significant opportunity for the applied mathematics community. To date, DL methods have been shown to be applicable as surrogates of physics based models, and are able to identify unrepresented mechanisms in these models. However, we argue that the greatest opportunity is in the use of the associated software frameworks. Both earth systems modelling and DL would benefit enormously from cross-pollination, and the best way to realize these opportunities is for applied mathematicians to employ DL software in their research.

While Marçais and de Dreuzy [2017] note the promise in using process based and statistical (DL) approaches in a complementary fashion, we argue that the biggest gains will come only if for process based modellers use and contribute to DL software. We predict this will accelerate advances in both fields by improving the computational resources available to physical modellers, the mathematical techniques available to DL practitioners, and the ease with which approaches can be combined.

A number of specific research challenges present themselves. Transfer learning, where models are first trained on large generic datasets before being refined on a smaller, more relevant ones, represents a great opportunity. The application of transfer learning to earth systems models, typically PDE based at present, has not been explored in great depth. The rapidly increasing availability of remote sensing data makes this a promising avenue for future research. The need is there: Clark et al. [2015] note that representing spatial variability remains the greatest challenge in hydrological modeling.

Another exciting prospect is the porting of methods from applied mathematics to the DL community. In particular uncertainty and sensitivity analysis methods encoded in packages such as PEST Doherty [2009], DAKOTA Eldred et al. [2013], and RAVEN Rabiti et al. [2014].

As demonstrated in Section 5.4, DL software can be used to implement and emulate process based models relatively easily. In Section 5.4.2 we developed DL surrogates which captured a range of common input-output relationships in ground-

water modelling. The number of inputs far exceeds what is typically possible in other surrogate approaches.

Our position is that the numerical modelling community, in particular hydrogeological modellers, would benefit enormously from using DL related software (even if they are not implementing NNs) and from methods such as that given in Section 5.4.2, a simple black box surrogate capable of approximating any reasonable number of input-output relationships. However, the aforementioned benefits alone justify greater engagement between the DL field of research and the numerical modeling community, and as noted in Section 5.2, there is a blue sky full of other innovations that may well come from this shift.

At present DL methods are largely black box, which ironically do little for human learning and provide limited insight into the confidence of their results. The apparent "power" in interpolating on a given dataset carries danger their results will be misconstrued elsewhere.

Currently, process driven surrogates such as the Multi-fidelity Stochastic Collocation approach described in Chapter 4 still provide an attractive alternative to groundwater modellers seeking a drop in surrogate to extrapolate, for example to a scenario with a fundamentally different climate to the data available for training. Further research into the the potential of DL methods to provide value to groundwater modellers in these scenarios is required.

5.6 Appendix

```python
import torch
import numpy as np
import util
def solve(A, rhs):
    LU_data, LU_pivots = torch.btrifact(torch.unsqueeze(A, 0))
    x = torch.btrisolve(torch.unsqueeze(rhs, 0), LU_data, LU_pivots)
    return torch.squeeze(x)
def implicit_heat_2d_torch(U, dx, dt, timesteps=1):
    n = U.shape[0]
    mu = 0.5 * dt / (dx * dx)
    # tridiagonal LHS
    du = -mu * np.ones((n - 1))
    dc = 1 + 2 * mu * np.ones((n))
    dl = -mu * np.ones((n - 1))
    A = torch.from_numpy(np.diag(dl, -1) + np.diag(dc) + np.diag(du, 1))
    for t in range(timesteps):
        # Peaceman-Rachford alternating direction implicit finite difference method for solving the 2d heat equation
        rhs = torch.from_numpy(np.zeros((n, n)))
        rhs[:, 0] = 0.
        rhs[:, n - 1] = 0.
        V = mu * U[:, 2:] + (1 - 2 * mu) * U[:, 1: n - 1] + mu * U[:, :-2] + mu * rhs[:, 1: n - 1]
        rhs[:, 1: n - 1] = V
        U_star = solve(A, rhs)
        U_star[:, 0] = 0.
        U_star[:, n - 1] = 0.
        rhs = torch.from_numpy(np.zeros((n, n)))
        rhs[0, :] = 0.
        rhs[n - 1, :] = 0.
        V = mu * U_star[2:, :] + (1 - 2 * mu) * U_star[1: n - 1, :] + mu * U_star[:-2, :] + mu * rhs[1: n - 1, :]
        rhs[1: n - 1, :] = V
        U[:, :] = solve(A, rhs)
        U[:, 0] = 0.
        U[:, n - 1] = 0.
        U[0, :] = 0.
        U[n - 1, :] = 0.
    return U
def explicit_heat_2d_torch(U, dx, dt, timesteps=1):
    r = dt / (dx * dx)
    assert r <= 1 / 4  # for stability
    for i in range(timesteps):
        dUx = U[2:, 1: -1] - 2 * U[1: -1, 1: -1] + U[:-2, 1: -1]
        dUy = U[1: -1, 2:] - 2 * U[1: -1, 1: -1] + U[1: -1, :-2]
        U[1: -1, 1: -1] = U[1: -1, 1: -1] + r * (dUx + dUy)
        U[0, :] = 0.
        U[-1, :] = 0.
        U[:, 0] = 0.
        U[:, -1] = 0.
    return U
if __name__ == '__main__':
  R = 32
  dx = 1.0 / R
  dt = 0.0001
  timesteps = 100
  U = np.meshgrid(np.ones(R), np.linspace(0, 1, R))[1]
  util.imshow('Initial', U)
  U_implicit_torch = implicit_heat_2d_torch(torch.from_numpy(U.copy()), dx, dt * 10, timesteps=timesteps // 10).numpy()
  util.imshow('Implicit', U_implicit_torch)
  U_explicit_torch = explicit_heat_2d_torch(torch.from_numpy(U.copy()), dx, dt, timesteps=timesteps).numpy()
  util.imshow('Explicit', U_explicit_torch)
```

Figure 5.11: Implemention of a PDE solver in PyTorch.

```
import numpy as np
import torch
import torch.nn as nn
from torch.autograd import Variable
from torchvision import datasets
import util
# test and training data loader
class Headset(torch.utils.data.Dataset):
    def __init__(self, base_dir, train_or_test):
        self.train_data = util.read_dir('%s/%s/%s' % (base_dir, train_or_test, 'in'))
        self.target_data = util.read_dir('%s/%s/%s' % (base_dir, train_or_test, 'out'))
        assert len(self.train_data) == len(self.target_data)
    def __getitem__(self, index):
        img = util.from_np_pickle(self.train_data[index])
        target = util.from_np_pickle(self.target_data[index])
        return img, target
    def __len__(self):
        return len(self.train_data)
# model structure
class Net(nn.Module):
    def __init__(self, timesteps=1):
        super(Net, self).__init__()
        # reuse the same instances here so there is only one set of conv parameters
        pad = nn.ReplicationPad2d(1)
        conv = nn.Conv2d(1, 1, kernel_size=3, bias=False)
        layers = []
        for i in range(timesteps):
            layers.append(pad)
            layers.append(conv)
        self.net = nn.Sequential(*layers)
    def forward(self, x):
        return self.net(x)
def loss_func(output, target):
    return (output[:, :, 1:-1, 1:-1] - target[:, :, 1:-1, 1:-1]).pow(2).sum()
```

Figure 5.12: Implemention and training of a convoluation NN in PyTorch.

Chapter 6

Conclusion

6.1 Contributions

A number of surrogate modelling methods have been developed to increase the efficiency of model based analysis of groundwater systems. We reviewed progress to date in Chapter 2 and developed three families of promising, state of the art surrogates of practical groundwater models in the following three chapters. In each case we analyzed the performance of the surrogate method in a range of practical groundwater modelling contexts where computational efficiency is a bottleneck to model based analysis. The models in Chapters 3 and 5 estimated water pressure at a number of locations based on uncertain boundary conditions and aquifer properties. This is a typical scenario for calibration and use of groundwater models where the main source of available data is water levels. The model of Chapter 4 estimated heads of a deep aquifer where little was known about its structure, hence the model's layer depths and properties were treated as uncertain inputs. All of the approaches provided significant runtime reductions. Once calibrated, the PCE and DL surrogates have a negligible computational cost to run online. The PCE approach took fewer than 200 full model runs to calibrate a surrogate of less than 1% error, and sensitivity indices can be derived from the expansion without further computational cost. The DL approach achieved similar results using 500 full model runs to calibrate. MFSC used far fewer full model runs (<50) but required many (1000) lower fidelity models to calibrate, and requires the lower fidelity model to be run online for each surrogate prediction.

All three methods can be applied simply as non-intrusive emulators in a number

of contexts, and have the benefit of adding an additional layer of understanding to the analysis. Non-intrusive methods can be applied without editing the underlying model, making such approaches simpler and more robust to apply. In many cases only a an input-output dataset is readily available so intrusive methods are not possible.

The PCE surrogate developed in Chapter 3 provides efficient sensitivity analysis of a model. The adaptive sparse grid calibration method trains a surrogate of a given accuracy using an order of magnitude fewer samples than conventional calibration approaches. The method stands out among all other approaches we analyzed as the fastest for moderate (<20) numbers of parameters. Surrogate introduced error is shown to quickly converge below that due to the spatial and temporal resolution. We achieve convergence of the error in the mean estimate below 1% (relitve RMSE) in fewer than 100 runs. Further, we developed a novel calibration method that makes use of nested polynomials implemented as layers in a DL software package. The approach has fewer parameters and converges faster than a conventional surrogate with a single polynomial of the same order. Of note too is that PCE theory [Xiu and Karniadakis, 2002] holds value in this new architecture. Pairing the polynomial used with the distribution of the input reduces the number of full model runs necessary for calibration of this nested surrogate.

The MFSC method was shown in Chapter 4 to be another promising emulation method. The approach is well suited to spatially and temporally distributed parameters, and has the added benefit of providing insight into the resolution-dependent errors of the underlying model. By using lower-fidelity model runs as the basis for the surrogate, the approach is better suited than purely statistical methods to make predictions in unseen conditions such as in different catchments or future climate scenarios. We developed two improvements to the algorithm in practical groundwater modelling settings. Both of these reduced error in the surrogate by an order of magnitude with the same training cost. Firstly, low-fidelity outputs should be normalized for node selection so as to prevent bias towards outputs of larger magnitude. Second, as much information from the low-fidelity simulator should be included in the output as is practical. Significant improvement was noted by including more outputs (e.g.

both spatial fields and timeseries) of the same underlying model.

In Chapter 5 we reviewed the rapidly developing field of DL surrogates and their recent application to groundwater modelling. We show two state of the art architectures perform well in a range of modelling contexts. Their major advantage as an emulator is their ability to handle orders of magnitude more model parameters than other surrogate methods. Further, these surrogates have the potential to provide more value than reducing computational requirements. We provided two examples of DL architectures which not only serve as emulators, but capture properties of the underlying physics-based model in their internal weights as they are calibrated. These represent innovative first steps towards using progress in DL not only to speed up existing process based groundwater models, but to create new models altogether.

6.2 Limitations

All surrogate techniques add a significant level of complexity to the modelling process, which in many cases outweigh the benefits in decreased modelling times. In some cases, making better use of computational resources (parallelism, faster languages, GPUs) is a better alternative. This would be true for many of the techniques reviewed in Chapter 2 and Razavi et al. [2012b] where computational gains reported between the original model and surrogate are less than a couple orders of magnitude. Or in analyses where the number of complex model runs necessary to calibrate the surrogate on a domain is not significantly lower than the number needed for the analysis itself. In summary, surrogates are best suited to scenarios such as integrated modelling, where online prediction speed is crucial.

Most approaches (MFSC and PCE included) are computationally intractable with large (>50) numbers of parameters. While DL methods can handle more parameters, published approaches to date have done so by including large samples of the underlying model for calibration of the surrogate. This requirement, along with relatively long training times is a major limitation of DL approaches in practice.

MFSC is among many novel algorithms with little publicly available software. While quality software implementations do exist for PCE [Eldred et al., 2013; Jakeman,

2020a], a lack of simple and flexible tools remains the biggest roadblock to both methods' widespread adoption.

While the adaptive calibration of the PCE surrogate outperforms naive sampling based approaches, it relies on a robust model which will not fail on any of the sampled inputs, which is seldom the case in practice.

Multi-fidelity methods such as MFSC are limited to scenarios where the fidelity (often the spatial resolution) can be altered without removing or fundamentally altering features of the model. This is not always possible, for example with certain boundary conditions with grid size [Mehl and Hill, 2010].

6.3 Future Work

In Chapter 2 we noted several generic areas of surrogate modelling warranting further research: quantification of surrogate induced uncertainty, multi-scale methods, localization in parameter space, and optimal snapshot selection. Research into each of these is required for practitioners to readily adopt model emulation. Surrogate induced uncertainty is typically treated as purely a function of stochastic resolution (how many model runs are required to train the surrogate). However as discussed in Chapter 3, stochastic and physical resolution both effect model runtime and error, and so should be considered together. A promising approach to doing so lies in multi-fidelity (or multi-scale) methods [e.g. Jakeman et al., 2019a] which quantify both stochastic and physical grid errors. There is a significant amount of contemporary research into multi-fidelity surrogates [Peherstorfer et al., 2018]. Conversely, there has been little progress (based on e.g. the work of Wan and Karniadakis [2005]) to automate the building of a different surrogates on sub-domains of the parameter space to better deal with discontinuities. Many surrogate methods rely on ad hoc approaches to sampling complex model snapshots on which to calibrate, despite this typically being the most computationally expensive aspect of the analysis. Adaptive polynomial chaos methods have been developed [e.g. Miller et al., 2018] which iteratively select new complex model runs based on existing knowledge of the space, a key innovation to the widespread adoption of these methods.

Traditionally, surrogate approaches have formed part of model analysis literature concerning parameter estimation, uncertainty quantification, and sensitivity analysis. During the writing of this thesis, deep learning has expanded in popularity rapidly in both research and industry. The latter field has fewer mathematical roots but has recently attracted a huge investment in software implementations. The software frameworks recently developed by the deep learning community are a promising avenue for researchers interested in surrogates. They provide simple interfaces to state of the art computational resources, libraries of algorithms and a burgeoning community of experts. There is significant overlap in the fields, such as in neural network techniques. However applied mathematicians, including groundwater modelers, can tap into these benefits not only by using NN surrogates or models, but by implementing process-driven simulators using these libraries. Such efforts would rapidly accelerate collaboration between two large fields of study and undoubtedly benefit both.

In this vein, we implemented a nested PCE surrogate using a DL framework in Chapter 3 and showed correlation of surrogate weights with underlying physical properties in Chapter 5. These are early results, but we believe there is great promise in continuing to explore this area.

Bibliography

Ababou, R.; McLaughlin, D.; Gelhar, L. W.; and Tompson, A. F. B., 1989. Numerical simulation of three-dimensional saturated flow in randomly heterogeneous porous media. *Transport in Porous Media*, 4, 6 (1989), 549–565. (cited on page 32)

Aiken, P.; Demchenko, Y.; McCann, S.; and Selker, L., 2017. THE QUANT CRUNCH: HOW THE DEMAND FOR DATA SCIENCE SKILLS IS DISRUPTING THE JOB MARKET. Technical report, Burning Glass Technologies. (cited on page 99)

Amazon, 2019. Amazon DSSTNE: Deep Scalable Sparse Tensor Network Engine. https://github.com/amzn/amazon-dsstne. (cited on page 99)

Amos, B. and Kolter, J. Z., 2017. OptNet: Differentiable Optimization as a Layer in Neural Networks. *arXiv preprint arXiv:1703.00443*, (2017). (cited on page 107)

Andreu, J.; Capilla, J.; and Sanchís, E., 1996. AQUATOOL, a generalized decision-support system for water-resources planning and operational management. *Journal of Hydrology*, 177, 3 (1996), 269–291. (cited on page 20)

Antoulas, A. C., 2005. An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control*, 29, 2 (2005), 181–190. doi:10.1016/j.arcontrol.2005.08.002. http://www.sciencedirect.com/science/article/pii/S1367578805000283. (cited on pages 7 and 19)

Apple, 2019. Apple Machine Learning Journal. https://machinelearning.apple.com/. (cited on page 99)

Aquaveo, L. L. C., 2011. Gms 8.0 tutorials. *Retrieved from Aquaveo Website (URL: http://www. aquaveo. com/gms-learning)*, (2011). (cited on pages 32 and 62)

Arbogast, T., 2002. Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow. *Computational Geosciences*, 6, 3-4 (2002), 453–481. (cited on page 9)

Ashby, S. F. and Falgout, R. D., 1996. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering*, 124, 1 (1996), 145–159. (cited on pages 9 and 27)

Asher, M. J.; Croke, B. F. W.; Jakeman, A. J.; and Peeters, L. J. M., 2015. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, (2015). (cited on pages 32 and 60)

Badrinarayanan, V.; Kendall, A.; and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39, 12 (2017), 2481–2495. (cited on pages 101 and 108)

Baidu, 2019. Institute of Deep Learning - Baidu Research. http://research.baidu.com/institute-of-deep-learning/. (cited on page 99)

Bakker, M.; Post, V.; Langevin, C. D.; Hughes, J. D.; White, J. T.; Starn, J. J.; and Fienen, M. N., 2016. Scripting MODFLOW Model Development Using Python and FloPy. *Groundwater*, 54, 5 (2016), 733–739. doi:https://doi.org/10.1111/gwat.12413. https://ngwa.onlinelibrary.wiley.com/doi/abs/10.1111/gwat.12413. (cited on pages 45 and 99)

Ballio, F. and Guadagnini, A., 2004. Convergence assessment of numerical Monte Carlo simulations in groundwater hydrology. *Water resources research*, 40, 4 (2004). (cited on page 34)

Barthelmann, V.; Novak, E.; and Ritter, K., 2000. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12, 4 (2000), 273–288. (cited on pages 40 and 41)

Bastian, P. and Reichenberger, V., 2000. Multigrid for higher order discontinuous

Galerkin finite elements applied to groundwater flow. Technical report, Universitat Heidelberg. (cited on page 23)

Baú, D. A., 2012. Planning of groundwater supply systems subject to uncertainty using stochastic flow reduced models and multi-objective evolutionary optimization. *Water Resources Management*, 26, 9 (2012), 2513–2536. (cited on page 21)

Baú, D. A. and Mayer, A. S., 2006. Stochastic management of pump-and-treat strategies using surrogate functions. *Advances in Water Resources*, 29, 12 (2006), 1901–1917. (cited on page 9)

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H., 2007. Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, 19, 1 (2007), 153. doi:citeulike-article-id:4640046. http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf. (cited on pages 102 and 108)

Bhatt, G.; Kumar, M.; and Duffy, C. J., 2014. A tightly coupled GIS and distributed hydrologic modeling framework. *Environmental Modelling and Software*, 62 (2014), 70–84. (cited on pages 32 and 62)

Bieker, H. P.; Slupphaug, O.; Johansen, T. A.; and Others, 2007. Real-time production optimization of oil and gas production systems: A technology survey. *SPE Production and Operations*, 22, 04 (2007), 382–391. (cited on page 6)

Blanning, R. W., 1975. The construction and implementation of metamodels. *Simulation*, 24, 6 (1975), 177–184. (cited on page 6)

Bower, K. M.; Gable, C. W.; and Zyvoloski, G. A., 2005. Grid resolution study of ground water flow and transport. *Groundwater*, 43, 1 (2005), 122–132. (cited on pages 32 and 33)

Boyce, S. E. and Yeh, W. W.-G., 2014. Parameter-independent model reduction of transient groundwater flow models: Application to inverse problems. *Advances in Water Resources*, 69 (2014), 168–180. (cited on pages 21 and 29)

Breiman, L., 2001. Random forests. *Machine Learning*, 45, 1 (2001), 5–32. (cited on page 13)

Brooks, R. J. and Tobias, A. M., 1996. Choosing the best model: Level of detail, complexity, and model performance. *Mathematical and computer modelling*, 24, 4 (1996), 1–14. (cited on page 33)

Brunner, P.; Franssen, H.-J. H.; Kgotlhang, L.; Bauer-Gottwein, P.; and Kinzelbach, W., 2007. How can remote sensing contribute in groundwater modeling? *Hydrogeology journal*, 15, 1 (2007), 5–18. (cited on page 106)

Buffa, A.; Maday, Y.; Patera, A. T.; Prudhomme, C.; and Turinici, G., 2012. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46, 03 (2012), 595–603. (cited on page 20)

Burkardt, J., 2013. FD2D_HEAT_STEADY - 2D Steady State Heat Equation in a Rectangle. https://people.sc.fsu.edu/{~}jburkardt/cpp{_}src/fd2d{_}heat{_}steady/fd2d{_}heat{_}steady.html. (cited on page 105)

Burrows, W. and Doherty, J., 2014. Efficient Calibration/Uncertainty Analysis Using Paired Complex/Surrogate Models. *Groundwater*, (2014). (cited on page 62)

Byrd, R. H.; Lu, P.; Nocedal, J.; and Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16, 5 (1995), 1190–1208. (cited on page 53)

Caffe. Caffe2 is a lightweight, modular, and scalable deep learning framework. https://github.com/caffe2/caffe2. (cited on page 105)

Cameron, R. H. and Martin, W. T., 1947. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, (1947), 385–392. (cited on page 35)

Cardoso, M. A.; Durlofsky, L. J.; and Sarma, P., 2009. Development and application of reduced-order modeling procedures for subsurface flow simulation. *International journal for numerical methods in engineering*, 77, 9 (2009), 1322–1350. (cited on page 21)

Carrara, E.; Sharples, J.; and Nation, E., 2017. Making Australian groundwater data accessible: the value of collaboration. *Water Practice and Technology*, 12, 3 (2017), 675–680. (cited on page 45)

Castelletti, A.; Galelli, S.; Ratto, M.; Soncini-Sessa, R.; and Young, P. C., 2012. A general framework for Dynamic Emulation Modelling in environmental problems. *Environmental Modelling and Software*, 34 (2012), 5–18. (cited on page 7)

Chaudhari, P.; Oberman, A.; Osher, S.; Soatto, S.; and Carlier, G., 2017. Deep Relaxation: partial differential equations for optimizing deep neural networks. *arXiv preprint arXiv:1704.04932*, (2017). (cited on page 107)

Chen, Y.; Hesthaven, J. S.; Maday, Y.; and Rodriguez, J., 2010. Certified Reduced Basis Methods and Output Bounds for the Harmonic Maxwell's Equations. In *SIAM Journal on Scientific Computing*, vol. 32, 970–996. SIAM publications. doi: 10.1137/09075250X. (cited on pages 9, 19, 20, and 28)

Cheng, W.-C.; Putti, M.; Kendall, D. R.; and Yeh, W. W.-G., 2011. A real-time groundwater management model using data assimilation. *Water Resources Research*, 47, 6 (2011), n/a—-n/a. doi:10.1029/2010WR009770. http://dx.doi.org/10.1029/2010WR009770. (cited on page 11)

Cheng, X.; Khomtchouk, B.; Matloff, N.; and Mohanty, P., 2018. Polynomial Regression As an Alternative to Neural Nets. *arXiv preprint arXiv:1806.06850*, (2018). (cited on page 57)

Chihara, T. S., 1978. *An introduction to orthogonal polynomials*. Gordon and Breach, New York, NY, USA. (cited on page 52)

Chinesta, F.; Ladeveze, P.; and Cueto, E., 2011. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18, 4 (2011), 395–404. (cited on pages 9, 19, and 20)

Ciriello, V.; Federico, V.; Riva, M.; Cadini, F.; Sanctis, J.; Zio, E.; and Guadagnini, A., 2013. Polynomial chaos expansion for global sensitivity analysis applied to a model of radionuclide migration in a randomly heterogeneous

aquifer. *Stochastic Environmental Research and Risk Assessment*, 27, 4 (2013), 945–954. doi:10.1007/s00477-012-0616-7. http://dx.doi.org/10.1007/s00477-012-0616-7. (cited on page 33)

Clark, M. P.; Nijssen, B.; Lundquist, J. D.; Kavetski, D.; Rupp, D. E.; Woods, R. A.; Freer, J. E.; Gutmann, E. D.; Wood, A. W.; Brekke, L. D.; and Others, 2015. A unified approach for process-based hydrologic modeling: 1. Modeling concept. *Water Resources Research*, 51, 4 (2015), 2498–2514. (cited on page 121)

Cliffe, K. A.; Giles, M. B.; Scheichl, R.; and Teckentrup, A. L., 2011. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14, 1 (2011), 3–15. (cited on page 24)

Connor, R., 2015. *The United Nations world water development report 2015: water for a sustainable world*, vol. 1. UNESCO publishing. (cited on page 1)

contributors, G., 2021. *{GDAL/OGR} Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. https://gdal.org. (cited on page 45)

Craig, J. R. and Read, W. W., 2010. The future of analytical solution methods for groundwater flow and transport simulation. In *XVIII International Conference on Computational Methods in Water Resources*, 1–8. International Center for Numerical Methods in Engineering. (cited on page 25)

Cukier, R. I.; Fortuin, C. M.; Shuler, K. E.; Petschek, A. G.; and Schaibly, J. H., 1973. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. *The Journal of Chemical Physics*, 59, 8 (1973), 3873–3878. (cited on page 42)

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2, 4 (1989), 303–314. (cited on page 102)

Dagasan, Y.; Juda, P.; and Renard, P., 2020. Using Generative Adversarial Networks as a Fast Forward Operator for Hydrogeological Inverse Problems. *Groundwater*,

58, 6 (nov 2020), 938–950. doi:10.1111/gwat.13005. https://onlinelibrary.wiley.com/doi/10.1111/gwat.13005. (cited on page 103)

Demissie, Y. K.; Valocchi, A. J.; Minsker, B. S.; and Bailey, B. A., 2009. Integrating a calibrated groundwater flow model with error-correcting data-driven models to improve predictions. *Journal of Hydrology*, 364, 3 (2009), 257–271. (cited on pages 6 and 14)

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE. (cited on page 102)

Diersch, H. J. G., 2005. FEFLOW finite element subsurface flow and transport simulation system. *Reference manual. Berlin, Germany: WASY GmbH Inst. Water Resour. Plan. Syst. Research*, (2005), 292. (cited on page 5)

Doherty, J., 2009. PEST: model-independent parameter estimation. *Watermark Numerical Computing, Brisbane, Australia*, (2009). (cited on pages 56 and 121)

Doherty, J. and Christensen, S., 2011. Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research*, 47, 12 (2011), W12534. (cited on pages 6, 24, 25, and 60)

Doherty, J. and Simmons, C., 2013. Groundwater modelling in decision support: reflections on a unified conceptual framework. *Hydrogeology Journal*, 21, 7 (2013), 1531–1537. doi:10.1007/s10040-013-1027-7. http://dx.doi.org/10.1007/s10040-013-1027-7. (cited on page 5)

Doherty, J. E. and Hunt, R. J., 2010. Approaches to highly parameterized inversion-A guide to using PEST for groundwater-model calibration. Technical report, US Geological Survey. (cited on page 56)

Dong, Y. and Li, G., 2009. A parallel PCG solver for MODFLOW. *Groundwater*, 47, 6 (2009), 845–850. (cited on page 27)

Dueben, P. D. and Bauer, P., 2018. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11, 10 (2018), 3999–4009. (cited on page 105)

Dunbar, W. S. and Woodbury, A. D., 1989. Application of the Lanczos algorithm to the solution of the groundwater flow equation. *Water Resources Research*, 25, 3 (1989), 551–558. (cited on pages 9, 17, and 19)

Efendiev, Y.; Datta-Gupta, A.; Ginting, V.; Ma, X.; and Mallick, B., 2005. An efficient two-stage Markov chain Monte Carlo method for dynamic data integration. *Water Resources Research*, 41, 12 (2005). doi:10.1029/2004WR003764. (cited on page 24)

Efendiev, Y.; Galvis, J.; Lazarov, R.; Moon, M.; and Sarkis, M., 2013. Generalized multiscale finite element method. Symmetric interior penalty coupling. *Journal of Computational Physics*, 255 (2013), 1–15. (cited on page 23)

Efendiev, Y.; Galvis, J.; and Thomines, F., 2012. A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications. *Multiscale Modeling and Simulation*, 10, 4 (2012), 1317–1343. (cited on pages 9 and 23)

Efendiev, Y. and Hou, T., 2007. Multiscale finite element methods for porous media flows and their applications. *Applied Numerical Mathematics*, 57, 5 (2007), 577–596. (cited on pages 23 and 60)

El Hajj, M.; Baghdadi, N.; Zribi, M.; and Bazzi, H., 2017. Synergic use of Sentinel-1 and Sentinel-2 images for operational soil moisture mapping at high spatial resolution over agricultural areas. *Remote Sensing*, 9, 12 (2017), 1292. (cited on page 100)

Eldred, M. S. and Burkardt, J., 2009. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. *AIAA paper*, 976, 2009 (2009), 1–20. (cited on page 39)

Eldred, M. S.; Giunta, A. A.; van Bloemen Waanders, B. G.; Wojtkiewicz, S. F.; Hart, W. E.; and Alleva, M. P., 2013. *DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.3.1 reference manual*, vol. SAND2010-2. Sandia National Laboratories Albuquerque, NM. (cited on pages 55, 121, and 127)

Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.-A.; Vincent, P.; and Bengio, S., 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, Feb (2010), 625–660. (cited on page 102)

Erhel, J.; de Dreuzy, J.-R.; Beaudoin, A.; Bresciani, E.; and Tromeur-Dervout, D., 2009. A parallel scientific software for heterogeneous hydrogeology. *Parallel Computational Fluid Dynamics 2007*, (2009), 39–48. (cited on page 27)

Ernst, O. G.; Mugler, A.; Starkloff, H.-J.; and Ullmann, E., 2012. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46, 02 (2012), 317–339. (cited on page 34)

Espig, M.; Hackbusch, W.; Litvinenko, A.; Matthies, H. G.; and Zander, E., 2013. Efficient analysis of high dimensional data in tensor formats. In *Sparse Grids and Applications, edited by J.Garcke and M. Griebel*, vol. 88, 31–56. Springer. (cited on page 29)

Espinet, A. J. and Shoemaker, C. A., 2013. Comparison of optimization algorithms for parameter estimation of multi-phase flow models with application to geological carbon sequestration. *Advances in Water Resources*, 54 (2013), 133–148. (cited on pages 13 and 29)

Estabragh, A. R.; Pereshkafti, M. R. S.; and Javadi, A. A., 2013. Comparison Between Analytical and Numerical Methods in Evaluating the Pollution Transport in Porous Media. *Geotechnical and Geological Engineering*, 31, 1 (2013), 93–101. (cited on page 25)

Fallah-Mehdipour, E.; Haddad, O. B.; and Mariño, M. A., 2013. Prediction and

simulation of monthly groundwater levels by genetic programming. *Journal of Hydro-Environment Research*, 7, 4 (2013), 253–260. (cited on pages 9 and 12)

Farthing, M. W.; Fowler, K. R.; Fu, X.; Davis, A.; and Miller, C. T., 2012. Effects of model resolution on optimal design of subsurface flow and transport problems. *Advances in Water Resources*, 38 (2012), 27–37. (cited on page 21)

Feinberg, J. and Langtangen, H. P., 2015. Chaospy: An open source tool for designing methods of uncertainty quantification. *Journal of Computational Science*, 11 (2015), 46–57. doi:https://doi.org/10.1016/j.jocs.2015.08.008. http://www.sciencedirect.com/science/article/pii/S1877750315300119. (cited on page 55)

Fienen, M. N. and Hunt, R. J., 2015. High-Throughput Computing Versus High-Performance Computing for Groundwater Applications. *Groundwater*, 53, 2 (2015), 180–184. (cited on page 27)

Fienen, M. N.; Masterson, J. P.; Plant, N. G.; Gutierrez, B. T.; and Thieler, E. R., 2013. Bridging groundwater models and decision support with a Bayesian network. *Water Resources Research*, 49, 10 (2013), 6459–6473. (cited on pages 9 and 11)

Fitzpatrick, C., 2012. Schlumberger Water Services launches Visual MODFLOW Flex. (2012). (cited on pages 32 and 62)

Forrester, A.; Keane, A.; and Others, 2008. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons. (cited on pages 7 and 12)

Forrester, A. I. J. and Keane, A. J., 2009. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45, 1 (2009), 50–79. (cited on pages 7, 12, 13, and 14)

Fox, R. L. and Miura, H., 1971. An approximate analysis technique for design calculations. *AIAA Journal*, 9, 1 (1971), 177–179. (cited on page 60)

Frangos, M.; Marzouk, Y.; Willcox, K.; and van Bloemen Waanders, B., 2010. Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems. *Large-Scale Inverse Problems and Quantification*

of Uncertainty, (2010), 123–149. doi:10.1002/9780470685853.ch7. http://dx.doi.org/10.1002/9780470685853.ch7. (cited on pages 7, 19, and 20)

Franssen, H. J. H.; Alcolea, A.; Riva, M.; Bakr, M.; der Wiel, N.; Stauffer, F.; and Guadagnini, A., 2009. A comparison of seven methods for the inverse modelling of groundwater flow. Application to the characterisation of well catchments. *Advances in Water Resources*, 32, 6 (2009), 851–872. (cited on page 27)

Franssen, H. J. H. and Kinzelbach, W., 2009. Ensemble Kalman filtering versus sequential self-calibration for inverse modelling of dynamic groundwater flow systems. *Journal of Hydrology*, 365, 3–4 (2009), 261–274. doi:10.1016/j.jhydrol.2008.11.033. http://www.sciencedirect.com/science/article/pii/S0022169408005908. (cited on page 26)

Galbally, D.; Fidkowski, K.; Willcox, K.; and Ghattas, O., 2010. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81, 12 (2010), 1581–1608. (cited on page 9)

Ganapathysubramanian, B. and Zabaras, N., 2007. Modeling diffusion in random heterogeneous media: data-driven models, stochastic collocation and the variational multiscale method. *Journal of Computational Physics*, 226, 1 (2007), 326–353. (cited on page 9)

Garcet, J. D. P.; Ordonez, A.; Roosen, J.; and Vanclooster, M., 2006. Metamodelling: theory, concepts and application to nitrate leaching modelling. *Ecological Modelling*, 193, 3 (2006), 629–644. (cited on pages 9 and 13)

Geneva, N. and Zabaras, N., 2020. Multi-fidelity generative deep learning turbulent flows. *arXiv preprint arXiv:2006.04731*, (2020). (cited on pages 104 and 110)

Ghanem, R., 1998. Scales of fluctuation and the propagation of uncertainty in random porous media. *Water Resources Research*, 34, 9 (1998), 2123–2136. doi:10.1029/98WR01573. http://dx.doi.org/10.1029/98WR01573. (cited on page 33)

Ghanem, R. and Dham, S., 1998. Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transport in Porous Media*, 32, 3 (1998), 239–262. (cited on pages 12 and 44)

Gheorghiu, C. I., 2007. *Spectral methods for differential problems*. Casa C{\u{a}}r{\c{t}}ii de {\c{S}}tiin{\c{t}}{\u{a}}. (cited on page 38)

Ghommem, M.; Presho, M.; Calo, V. M.; and Efendiev, Y., 2013. Mode decomposition methods for flows in high-contrast porous media. Global–local approach. *Journal of Computational Physics*, 253, 0 (2013), 226–238. doi:http://dx.doi.org/10.1016/j.jcp.2013.06.033. http://www.sciencedirect.com/science/article/pii/S0021999113004634. (cited on pages 9, 19, and 20)

Giles, M. B., 2008. Multilevel monte carlo path simulation. *Operations Research*, 56, 3 (2008), 607–617. (cited on page 60)

Google, 2019. TensorFlow – Google Open Source. https://opensource.google.com/projects/tensorflow. (cited on pages 99, 102, 105, and 110)

Guadagnini, A. and Neuman, S. P., 1999. Nonlocal and localized analyses of conditional mean steady state flow in bounded, randomly nonuniform domains: 1. Theory and computational approach. *Water Resources Research*, 35, 10 (1999), 2999–3018. doi:10.1029/1999WR900160. http://dx.doi.org/10.1029/1999WR900160. (cited on page 26)

Gugercin, S. and Antoulas, A. C., 2000. A comparative study of 7 algorithms for model reduction. In *Proceedings of the 39th IEEE Conference on Decision and Control.*, vol. 3, 2367–2372. IEEE. (cited on pages 7 and 19)

Gugercin, S. and Willcox, K., 2008. Krylov projection framework for Fourier model reduction. *Automatica*, 44, 1 (2008), 209–215. doi:10.1016/j.automatica.2007.05.007. http://www.sciencedirect.com/science/article/pii/S0005109807002610. (cited on page 19)

Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; and Ye, J., 2020. A review on generative adversarial

networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*, (2020). (cited on page 110)

Guillaume, J. H. A.; Jakeman, J. D.; Marsili-Libelli, S.; Asher, M.; Brunner, P.; Croke, B.; Hill, M. C.; Jakeman, A. J.; Keesman, K. J.; Razavi, S.; and Stigter, J. D., 2019. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environmental Modelling & Software*, 119 (2019), 418–432. doi:https://doi.org/10.1016/j.envsoft.2019.07.007. http://www.sciencedirect.com/science/article/pii/S1364815218307278. (cited on page 101)

Haber, E. and Ruthotto, L., 2017. Stable Architectures for Deep Neural Networks. *arXiv preprint arXiv:1705.03341*, (2017). (cited on page 107)

Hajibeygi, H.; Bonfigli, G.; Hesse, M. A.; and Jenny, P., 2008. Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227, 19 (2008), 8604–8621. (cited on page 22)

Harbaugh, A. W., 2005. *MODFLOW-2005, the US Geological Survey modular ground-water model: The ground-water flow process*. US Department of the Interior, US Geological Survey. (cited on pages 5, 16, 76, 100, 105, 110, and 113)

Harding, B. and Hegland, M., 2014. Robust solutions to PDEs with multiple grids. In *Sparse Grids and Applications-Munich 2012*, 171–193. Springer. (cited on page 60)

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J., 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16, 12 (2004), 2639–2664. (cited on page 64)

Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; and Oliphant, T. E., 2020. Array programming

with NumPy. *Nature*, 585, 7825 (2020), 357–362. doi:10.1038/s41586-020-2649-2. https://doi.org/10.1038/s41586-020-2649-2. (cited on page 114)

Hastie, T.; Tibshirani, R.; Friedman, J.; Hastie, T.; Friedman, J.; and Tibshirani, R., 2009. *The elements of statistical learning*, vol. 2. Springer. (cited on page 14)

Hay, A.; Akhtar, I.; and Borggaard, J. T., 2012. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 68, 1 (2012), 122–134. doi:10.1002/fld.2512. http://dx.doi.org/10.1002/fld.2512. (cited on page 20)

He, X.; Jiang, L.; and David Moulton, J., 2013. A stochastic dimension reduction multiscale finite element method for groundwater flow problems in heterogeneous random porous media. *Journal of Hydrology*, 478 (2013), 77–88. (cited on page 23)

Hegland, M., 2003. Adaptive sparse grids. In *ANZIAM J.*, vol. 44, C335—-C353. (cited on page 40)

Hemker, T.; Fowler, K. R.; Farthing, M. W.; and von Stryk, O., 2008. A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. *Optimization and Engineering*, 9, 4 (2008), 341–360. (cited on page 6)

Herzog, M.; Gilg, A.; Paffrath, M.; Rentrop, P.; and Wever, U., 2008. Intrusive versus Non-Intrusive Methods for Stochastic Finite Elements. In *From Nano to Space* (Eds. M. H. Breitner; G. Denk; and P. Rentrop), 161–174. Springer Berlin Heidelberg. (cited on page 12)

Hill, M. C., 2006. The practical use of simplicity in developing ground water models. *Groundwater*, 44, 6 (2006), 775–781. (cited on page 25)

Hou, T. Y. and Wu, X.-H., 1997. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134, 1 (1997), 169–189. (cited on page 9)

Hughes, T. J. R.; Feijóo, G. R.; Mazzei, L.; and Quincy, J.-B., 1998. The variational multiscale method: a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166, 1 (1998), 3–24. (cited on page 9)

Hunt, R. J. and Doherty, J., 2011. Interesting or important? Resetting the balance of theory and application. *Groundwater*, 49, 3 (2011), 301. (cited on page 33)

Hussain, M. F.; Barton, R. R.; and Joshi, S. B., 2002. Metamodeling: radial basis functions versus polynomials. *European Journal of Operational Research*, 138, 1 (2002), 142–154. (cited on page 9)

Isukapalli, S. S., 1999. *Uncertainty analysis of transport-transformation models*. Ph.D. thesis, Rutgers, The State University of New Jersey. (cited on page 39)

Iwanaga, T.; Partington, D.; Ticehurst, J.; Croke, B. F. W.; and Jakeman, A. J., 2020. A socio-environmental model for exploring sustainable water management futures: Participatory and collaborative modelling in the Lower Campaspe catchment. *Journal of Hydrology: Regional Studies*, 28 (2020), 100669. doi:https://doi.org/10.1016/j.ejrh.2020.100669. https://www.sciencedirect.com/science/article/pii/S2214581819303726. (cited on page 113)

Iwanaga, T.; Zare, F.; Croke, B.; Fu, B.; Merritt, W.; Partington, D.; Ticehurst, J.; and Jakeman, A., 2018. Development of an integrated model for the Campaspe catchment: a tool to help improve understanding of the interaction between society, policy, farming decision, ecology, hydrology and climate. *Proceedings of the International Association of Hydrological Sciences*, 379 (2018), 1–12. (cited on page 43)

Jakeman, A. J.; Letcher, R. A.; and Norton, J. P., 2006. Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling and Software*, 21, 5 (2006), 602–614. (cited on pages 8 and 56)

Jakeman, J., 2020a. PyApprox. https://github.com/sandialabs/pyapprox. (cited on page 127)

Jakeman, J. D., 2020b. PyApprox. (cited on page 55)

Jakeman, J. D.; Eldred, M. S.; Geraci, G.; and Gorodetsky, A., 2019a. Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis. *International Journal for Numerical Methods in Engineering*, (nov 2019). doi:10.1002/nme.6268. http://dx.doi.org/10.1002/nme.6268. (cited on pages 2, 40, 42, 45, 47, 55, and 128)

Jakeman, J. D.; Franzelin, F.; Narayan, A.; Eldred, M.; and Plfüger, D., 2019b. Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351 (2019), 643–666. doi:https://doi.org/10.1016/j.cma.2019.03.049. https://www.sciencedirect.com/science/article/pii/S0045782519301884. (cited on page 36)

Jakeman, J. D. and Roberts, S. G., 2013. Local and dimension adaptive stochastic collocation for uncertainty quantification. In *Sparse grids and applications, edited by J.Garcke and M. Griebel*, 181–203. Springer. (cited on pages 14 and 29)

Jenny, P.; Lee, S. H.; and Tchelepi, H. A., 2003. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187, 1 (2003), 47–67. (cited on page 22)

Jones, D. R.; Schonlau, M.; and Welch, W. J., 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, 4 (1998), 455–492. (cited on page 13)

Kavetski, D. and Kuczera, G., 2007. Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration. *Water Resources Research*, 43, 3 (2007), n/a—-n/a. doi:10.1029/2006WR005195. http://dx.doi.org/10.1029/2006WR005195. (cited on page 6)

Keating, E. H.; Doherty, J.; Vrugt, J. A.; and Kang, Q., 2010. Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality. *Water Resources Research*, 46, 10 (2010), n/a—-n/a. doi:10.1029/2009WR008584. http://dx.doi.org/10.1029/2009WR008584. (cited on pages 24 and 56)

Kennedy, M. C. and O'Hagan, A., 2001. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 3 (2001), 425–464. (cited on pages 9 and 10)

Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on page 53)

Kitanidis, P. K. and Lee, J., 2014. Principal Component Geostatistical Approach for large-dimensional inverse problems. *Water Resources Research*, 50, 7 (2014), 5428–5443. doi:10.1002/2013WR014630. http://dx.doi.org/10.1002/2013WR014630. (cited on page 27)

Knezevic, D. J. and Peterson, J. W., 2011. A high-performance parallel implementation of the certified reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 200, 13 (2011), 1455–1466. (cited on pages 9, 19, and 20)

Kourakos, G. and Mantoglou, A., 2009. Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. *Advances in Water Resources*, 32, 4 (2009), 507–521. (cited on page 9)

Krizhevsky, A.; Sutskever, I.; and Geoffrey E., H., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS2012)*, (2012), 1–9. doi:10.1109/5.726791. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. (cited on page 109)

Laloy, E.; Rogiers, B.; Vrugt, J. A.; Mallants, D.; and Jacques, D., 2013. Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion. *Water Resources Research*, 49, 5 (2013), 2664–2682. doi:10.1002/wrcr.20226. http://dx.doi.org/10.1002/wrcr.20226. (cited on pages 9, 12, 26, 33, and 56)

LaVenue, A. M. and Pickens, J. F., 1992. Application of a coupled adjoint sensitivity and kriging approach to calibrate a groundwater flow model. *Water Resources Research*, 28, 6 (1992), 1543–1569. (cited on page 27)

Le Gratiet, L.; Cannamela, C.; and Iooss, B., 2014. A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 2, 1 (2014), 336–363. (cited on page 60)

Le Maitre, O. P.; Knio, O. M.; Najm, H. N.; and Ghanem, R. G., 2004. Uncertainty propagation using Wiener–Haar expansions. *Journal of Computational Physics*, 197, 1 (2004), 28–57. (cited on page 14)

Le Maitre, O. P.; Reagan, M. T.; Najm, H. N.; Ghanem, R. G.; and Knio, O. M., 2002. A stochastic projection method for fluid flow: II. Random process. *Journal of computational Physics*, 181, 1 (2002), 9–44. (cited on page 39)

LeCun, Y.; Bengio, Y.; and Hinton, G., 2015. Deep learning. *Nature*, 521, 7553 (2015), 436–444. (cited on pages 99, 100, 108, and 109)

LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; and Jackel, L. D., 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 396–404. (cited on page 102)

Leube, P. C.; de Barros, F. P. J.; Nowak, W.; and Rajagopal, R., 2013. Towards optimal allocation of computer resources: Trade-offs between uncertainty quantification, discretization and model reduction. *Environmental Modelling & Software*, 50 (2013), 97–107. doi:http://dx.doi.org/10.1016/j.envsoft.2013.08.008. http://www.sciencedirect.com/science/article/pii/S1364815213001862. (cited on page 34)

Leube, P. C.; Nowak, W.; and Schneider, G., 2012. Temporal moments revisited: Why there is no better way for physically based model reduction in time. *Water Resources Research*, 48, 11 (2012), n/a—-n/a. doi:10.1029/2012WR011973. http://dx.doi.org/10.1029/2012WR011973. (cited on page 5)

Lieberman, C.; Willcox, K.; and Ghattas, O., 2010. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32, 5 (2010), 2523–2542. (cited on pages 9, 19, 20, 21, and 61)

Lin, H. W.; Tegmark, M.; and Rolnick, D., 2017. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168, 6 (2017), 1223–1247. (cited on pages 51, 99, and 107)

Liu, Z. and Narayan, A., 2021. On the computation of recurrence coefficients for univariate orthogonal polynomials. (cited on page 36)

Lunati, I. and Jenny, P., 2008. Multiscale finite-volume method for density-driven flow in porous media. *Computational Geosciences*, 12, 3 (2008), 337–350. (cited on page 22)

Man, J.; Zheng, Q.; Wu, L.; and Zeng, L., 2020. Adaptive multi-fidelity probabilistic collocation-based Kalman filter for subsurface flow data assimilation: numerical modeling and real-world experiment. *Stochastic Environmental Research and Risk Assessment*, 34 (2020), 1135–1146. (cited on page 94)

Mansour, M. M. and Spink, A. E. F., 2013. Grid Refinement in Cartesian Coordinates for Groundwater Flow Models Using the Divergence Theorem and Taylor's Series. *Groundwater*, 51, 1 (2013), 66–75. (cited on page 23)

Marçais, J. and de Dreuzy, J.-R., 2017. Prospective interest of deep learning for hydrological inference. *Groundwater*, 55, 5 (2017), 688–692. (cited on pages 99, 104, 107, 108, 117, and 121)

Marcus, G., 2018. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, (2018). (cited on page 100)

MarietHoz, G.; Renard, P.; and Caers, J., 2010. Bayesian inverse problems and optimization with iterative spatial resampling. *Water Resources Research*, 46, 11 (2010), n/a—-n/a. doi:10.1029/2010WR009274. http://dx.doi.org/10.1029/2010WR009274. (cited on page 26)

Marzouk, Y. M. and Najm, H. N., 2009. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228, 6 (2009), 1862–1902. (cited on pages 13 and 14)

Matott, L. S.; Babendreier, J. E.; and Purucker, S. T., 2009. Evaluating uncertainty in integrated environmental models: A review of concepts and tools. *Water Resources Research*, 45, 6 (2009), n/a—-n/a. doi:10.1029/2008WR007301. http://dx.doi.org/10.1029/2008WR007301. (cited on pages 26 and 32)

Matott, L. S. and Rabideau, A. J., 2008. Calibration of complex subsurface reaction models using a surrogate-model approach. *Advances in Water Resources*, 31, 12 (2008), 1697–1707. (cited on pages 6 and 13)

Matott, L. S.; Rabideau, A. J.; and Craig, J. R., 2006. Pump-and-treat optimization using analytic element method flow models. *Advances in water resources*, 29, 5 (2006), 760–775. (cited on page 25)

Matthies, H. G. and Keese, A., 2005. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194, 12 (2005), 1295–1331. (cited on page 14)

McPhee, J. and Yeh, W. W.-G., 2008. Groundwater management using model reduction via empirical orthogonal functions. *Journal of Water Resources Planning and Management*, 134, 2 (2008), 161–170. (cited on pages 9, 15, 19, and 37)

Mehl, S. and Hill, M. C., 2002. Development and evaluation of a local grid refinement method for block-centered finite-difference groundwater models using shared nodes. *Advances in Water Resources*, 25, 5 (2002), 497–511. (cited on page 62)

Mehl, S. and Hill, M. C., 2010. Grid-size dependence of Cauchy boundary conditions used to simulate stream–aquifer interactions. *Advances in water resources*, 33, 4 (2010), 430–442. (cited on pages 25, 32, 33, 62, and 128)

Mehl, S. W. and Hill, M. C., 2005. *MODFLOW-2005, the US Geological Survey Modular Ground-Water Model: Documentation of shared node local grid refinement (LGR) and the boundary flow and head (BFH) package*. US Department of the Interior, US Geological Survey. (cited on pages 23 and 45)

Mehta, P. and Schwab, D. J., 2014. An exact mapping between the variational

renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, (2014). (cited on page 107)

Menberg, K.; Bidarmaghz, A.; Gregory, A.; Choudhary, R.; and Girolami, M., 2020. Multi-fidelity approach to Bayesian parameter estimation in subsurface heat and fluid transport models. *Science of The Total Environment*, 745 (2020), 140846. (cited on page 2)

Microsoft, 2019. Microsoft Cognitive Toolkit (CNTK), an open source deep-learning toolkit. (2019). https://github.com/Microsoft/CNTK. (cited on pages 99 and 105)

Mikołajczyk, A. and Grochowski, M., 2018. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, 117–122. IEEE. (cited on page 103)

Miller, C. T.; Dawson, C. N.; Farthing, M. W.; Hou, T. Y.; Huang, J.; Kees, C. E.; Kelley, C. T.; and Langtangen, H. P., 2013. Numerical simulation of water resources problems: Models, methods, and trends. *Advances in Water Resources*, 51 (2013), 405–437. (cited on pages 25 and 27)

Miller, K. L.; Berg, S. J.; Davison, J. H.; Sudicky, E. A.; and Forsyth, P. A., 2018. Efficient uncertainty quantification in fully-integrated surface and subsurface hydrologic simulations. *Advances in Water Resources*, 111 (2018), 381–394. (cited on page 128)

Mills, R. T.; Hammond, G. E.; Lichtner, P. C.; Sripathi, V.; Mahinthakumar, G. K.; and Smith, B. F., 2009. Modeling subsurface reactive flows using leadership-class computing. In *Journal of Physics: Conference Series*, vol. 180, 12062. IOP Publishing. (cited on page 27)

Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; and Terzopoulos, D., 2020. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566*, (2020). (cited on page 110)

Mo, S.; Zabaras, N.; Shi, X.; and Wu, J., 2019. Deep Autoregressive Neural Networks for High-Dimensional Inverse Problems in Groundwater Contam-

inant Source Identification. *Water Resources Research*, 55, 5 (2019), 3856–3881. doi:10.1029/2018WR024638. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR024638. (cited on pages 101, 103, and 113)

Mo, S.; Zabaras, N.; Shi, X.; and Wu, J., 2020. Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities. *Water Resources Research*, 56, 2 (2020), e2019WR026082. (cited on page 103)

Moslehi, M.; Rajagopal, R.; and de Barros, F. P. J., 2015. Optimal allocation of computational resources in hydrogeological models under uncertainty. *Advances in Water Resources*, 83 (2015), 299–309. doi:http://dx.doi.org/10.1016/j.advwatres.2015.06.014. http://www.sciencedirect.com/science/article/pii/S0309170815001402. (cited on page 34)

Mugunthan, P.; Shoemaker, C. A.; and Regis, R. G., 2005. Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models. *Water Resources Research*, 41, 11 (2005), W11427. (cited on page 14)

Nabian, M. A. and Meidani, H., 2018. A Deep Neural Network Surrogate for High-Dimensional Random Partial Differential Equations. *CoRR*, abs/1806.0 (2018). http://arxiv.org/abs/1806.02957. (cited on page 101)

Najm, H. N., 2009. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41 (2009), 35–52. (cited on pages 3 and 14)

Nan, T. and Wu, J., 2011. Groundwater parameter estimation using the ensemble Kalman filter with localization. *Hydrogeology Journal*, 19, 3 (2011), 547–561. (cited on page 28)

Narayan, A.; Gittelson, C.; and Xiu, D., 2014. A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36, 2 (2014), A495—-A521. (cited on pages xvi, 24, 28, 61, 62, 67, 68, and 70)

Narayan, A. and Jakeman, J. D., 2014. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36, 6 (2014), A2952—-A2983. (cited on pages 45 and 51)

Nearing, G. S.; Kratzert, F.; Sampson, A. K.; Pelissier, C. S.; Klotz, D.; Frame, J. M.; Prieto, C.; and Gupta, H. V., 2020. What role does hydrological science play in the age of machine learning? *Water Resources Research*, (2020), e2020WR028091. (cited on pages 100, 104, 108, and 117)

Neuman, S. P., 2004. Stochastic groundwater models in practice. *Stochastic Environmental Research and Risk Assessment*, 18, 4 (2004), 268–270. http://dx.doi.org/10.1007/s00477-004-0192-6. (cited on page 33)

Neuman, S. P. and Di Federico, V., 2003. Multifaceted nature of hydrogeologic scaling and its interpretation. *Reviews of Geophysics*, 41, 3 (2003), n/a—-n/a. doi:10.1029/2003RG000130. http://dx.doi.org/10.1029/2003RG000130. (cited on pages 32 and 62)

Nouy, A., 2007. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 196, 45–48 (2007), 4521–4537. doi:10.1016/j.cma.2007.05.016. http://www.sciencedirect.com/science/article/pii/S004578250700223X. (cited on page 20)

O'Hagan, A., 2006. Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering & System Safety*, 91, 10 (2006), 1290–1300. (cited on page 6)

O'Hagan, A., 2013. Polynomial Chaos: A Tutorial and Critique from a Statistician's Perspective. *SIAM/ASA Journal of Uncertainty Quantification*, 20 (2013). (cited on page 13)

Oliver, D. S. and Chen, Y., 2011. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15, 1 (2011), 185–221. (cited on pages 18, 26, and 27)

OpenBLAS. xianyi/OpenBLAS: OpenBLAS is an optimized BLAS library based on GotoBLAS2 1.13 BSD version. https://github.com/xianyi/OpenBLAS. (cited on page 99)

OpenTurns. openturns/openturns: Uncertainty treatment library. https://github.com/openturns/openturns. (cited on page 99)

Pachepsky, Y. A.; Guber, A. K.; van Genuchten, M. T.; Nicholson, T. J.; Cady, R. E.; Simunek, J.; and Schaap, M. G., 2006. Model abstraction techniques for soil-water flow and transport, prepared for Division of Fuels, Engineering and Radiological Research, Office of Nuclear Regulatory Research. *US Nuclear Regulatory Commission, Rep. NUREG/CR-6884*, (2006). (cited on page 32)

Panday, S.; Huyakorn, P. S.; Therrien, R.; and Nichols, R. L., 2013. MODFLOW–USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow. *Advances in Water Resources*, 25, 5 (2013), 497–511. (cited on page 23)

Parajka, J.; Viglione, A.; Rogger, M.; Salinas, J. L.; Sivapalan, M.; and Blöschl, G., 2013. Comparative assessment of predictions in ungauged basins–Part 1: Runoff-hydrograph studies. *Hydrology and Earth System Sciences*, 17, 5 (2013), 1783–1795. (cited on page 100)

Pascanu, R.; Mikolov, T.; and Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 1310–1318. (cited on page 103)

Pasetto, D.; Guadagnini, A.; and Putti, M., 2011. POD-based Monte Carlo approach for the solution of regional scale groundwater flow driven by randomly distributed recharge. *Advances in Water Resources*, 34, 11 (2011), 1450–1463. (cited on page 7)

Pasetto, D.; Guadagnini, A.; and Putti, M., 2014. A reduced-order model for Monte Carlo simulations of stochastic groundwater flow. *Computational Geosciences*, 18, 2 (2014), 157–169. (cited on pages 21, 28, and 29)

Pasetto, D.; Putti, M.; and Yeh, W. W.-G., 2013. A reduced-order model for groundwater flow equation with random hydraulic conductivity: Application to Monte Carlo methods. *Water Resources Research*, 49, 6 (2013), 3215–3228. doi:10.1002/wrcr.20136. http://dx.doi.org/10.1002/wrcr.20136. (cited on page 21)

Peherstorfer, B.; Willcox, K.; and Gunzburger, M., 2018. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60, 3 (2018), 550–591. (cited on pages 94 and 128)

Poggio, T. and Banburski, A., 2020. An Overview of Some Issues in the Theory of Deep Networks. *IEEJ Transactions on Electrical and Electronic Engineering*, 15, 11 (2020), 1560–1571. doi:https://doi.org/10.1002/tee.23243. https://onlinelibrary.wiley.com/doi/abs/10.1002/tee.23243. (cited on page 99)

Pogson, M. and Smith, P., 2015. Effect of spatial data resolution on uncertainty. *Environmental Modelling and Software*, 63 (2015), 87–96. (cited on page 33)

Powell, M. J. D., 1987. Radial basis functions for multivariable interpolation: a review. In *Algorithms for approximation*, 143–167. Clarendon Press. (cited on page 60)

PyTorch. PyTorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration. (cited on pages 53, 54, 102, 105, 110, and 114)

Rabiti, C.; Alfonsi, A.; Cogliati, J.; Mandelli, D.; and Kinoshita, R., 2014. RAVEN, a new software for dynamic risk analysis. Technical report, Idaho National Laboratory (INL). (cited on page 121)

Ranjithan, S.; Eheart, J. W.; and Garrett, J. H., 1993. Neural network-based screening for groundwater reclamation under uncertainty. *Water Resources Research*, 29, 3 (1993), 563–574. (cited on page 102)

Razavi, S.; Tolson, B. A.; and Burn, D. H., 2012a. Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environmental Modelling and Software*, 34 (2012), 67–86. (cited on pages 7, 13, 14, and 28)

Razavi, S.; Tolson, B. A.; and Burn, D. H., 2012b. Review of surrogate modeling in water resources. *Water Resources Research*, 48, 7 (2012), n/a—-n/a. doi:10.1029/2011WR011527. http://dx.doi.org/10.1029/2011WR011527. (cited on pages 2, 6, 7, 8, 11, 13, 14, 27, 32, 101, 104, and 127)

Refsgaard, J. C.; van der Sluijs, J. P.; Højberg, A. L.; and Vanrolleghem, P. A., 2007. Uncertainty in the environmental modelling process–a framework and guidance. *Environmental Modelling and Software*, 22, 11 (2007), 1543–1556. (cited on page 32)

Regis, R. G. and Shoemaker, C. A., 2005. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31, 1 (2005), 153–171. (cited on pages 6 and 9)

Regis, R. G. and Shoemaker, C. A., 2007. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19, 4 (2007), 497–509. (cited on page 13)

Regis, R. G. and Shoemaker, C. A., 2012. A quasi-multistart framework for global optimization of expensive functions using response surface models. *Journal of Global Optimization*, (2012), 1–35. (cited on pages 27 and 29)

Ren, Y.; Yoo, S.; and Hoisie, A., 2019. Performance analysis of deep learning workloads on leading-edge systems. In *2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 103–113. IEEE. (cited on page 102)

Renard, P., 2007. Stochastic hydrogeology: what professionals really need? *Groundwater*, 45, 5 (2007), 531–541. (cited on page 33)

Rewieński Michałand White, J., 2006. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415, 2 (2006), 426–454. (cited on page 20)

Roach, J. and Tidwell, V., 2009. A Compartmental - Spatial System Dynamics Approach to Ground Water Modeling. *Groundwater*, 47, 5 (2009), 686–698. doi:

10.1111/j.1745-6584.2009.00580.x. http://dx.doi.org/10.1111/j.1745-6584.2009.00580.x. (cited on page 6)

Robinson, T. D.; Eldred, M. S.; Willcox, K. E.; and Haimes, R., 2008. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA Journal*, 46, 11 (2008), 2814–2822. (cited on pages 6, 24, and 25)

Rosenblatt, F., 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, (1958), 65–386. (cited on page 102)

Rudd, K., 2013. *Solving Partial Differential Equations Using Artificial Neural Networks*. Ph.D. thesis, Duke University. (cited on page 107)

Rumbaugh, J. O. and Rumbaugh, D. B., 2007. Groundwater Vistas. (cited on pages 32 and 61)

Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P.; and Others, 1989. Design and analysis of computer experiments. *Statistical Science*, 4, 4 (1989), 409–423. (cited on page 10)

Sahuquillo, A., 1983. An eigenvalue numerical technique for solving unsteady linear groundwater models continuously in time. *Water Resources Research*, 19, 1 (1983), 87–93. (cited on pages 17 and 20)

Saied, F. and Mahinthakumar, G., 1998. Efficient parallel multigrid based solvers for large scale groundwater flow simulations. *Computers and Mathematics with Applications*, 35, 7 (1998), 45–54. (cited on page 9)

Sangalli, G., 2003. Capturing small scales in elliptic problems using a residual-free bubbles finite element method. *Multiscale Modeling and Simulation*, 1, 3 (2003), 485–503. (cited on page 9)

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, 61 (2015), 85–117. (cited on page 99)

Schöbi, R.; Sudret, B.; and Wiart, J., 2015. Polynomial-Chaos-based Kriging. *International Journal for Uncertainty Quantification, Submitted*, (2015). (cited on page 13)

Shafii, M.; Tolson, B.; and Matott, L., 2014. Uncertainty-based multi-criteria calibration of rainfall-runoff models: a comparative study. *Stochastic Environmental Research and Risk Assessment*, 28, 6 (2014), 1493–1510. doi:10.1007/s00477-014-0855-x. http://dx.doi.org/10.1007/s00477-014-0855-x. (cited on page 26)

Shahane, S.; Aluru, N. R.; and Vanka, S. P., 2019. Uncertainty quantification in three dimensional natural convection using polynomial chaos expansion and deep neural networks. *International Journal of Heat and Mass Transfer*, 139 (2019), 613–631. doi:https://doi.org/10.1016/j.ijheatmasstransfer.2019.05.014. https://www.sciencedirect.com/science/article/pii/S0017931018363919. (cited on page 56)

Shen, C., 2017. A trans-disciplinary review of deep learning research for water resources scientists. *arXiv preprint arXiv:1712.02162*, (2017). (cited on pages 100, 104, 108, and 117)

Shi, L.; Zeng, L.; Zhang, D.; and Yang, J., 2012. Multiscale-finite-element-based ensemble Kalman filter for large-scale groundwater flow. *Journal of Hydrology*, (2012). (cited on page 9)

Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1874–1883. (cited on page 107)

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, (2015). (cited on pages xviii, 110, 112, and 118)

Siade, A. J.; Putti, M.; and Yeh, W. W.-G., 2010. Snapshot selection for groundwater

model reduction using proper orthogonal decomposition. *Water Resources Research*, 46, 8 (2010), W08539. (cited on page 21)

Siade, A. J.; Putti, M.; and Yeh, W. W.-G., 2012. Reduced order parameter estimation using quasilinearization and quadratic programming. *Water Resources Research*, 48, 6 (2012), n/a—-n/a. doi:10.1029/2011WR011471. http://dx.doi.org/10.1029/2011WR011471. (cited on page 9)

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, (2014). (cited on page 109)

Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Quarterly of applied mathematics*, 45, 3 (1987), 561–571. (cited on page 60)

Sobol, I. M., 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation*, 55, 1 (2001), 271–280. (cited on page 42)

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15, 1 (2014), 1929–1958. (cited on page 102)

Stone, N., 2011. *Gaussian process emulators for uncertainty analysis in groundwater flow*. Ph.D. thesis, University of Nottingham. (cited on pages 9 and 10)

Sudret, B., 2008. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93, 7 (2008), 964–979. (cited on pages 2, 13, and 42)

Sun, A. Y., 2008. STATE-OF-THE-ART MULTISCALE APPROACHES FOR FLOW AND TRANSPORT MODELING: A LITERATURE REVIEW. Technical report, U.S. Nuclear Regulatory Commission - Center for Nuclear Waste Regulatory Analyses - San Antonio, Texas. (cited on page 23)

Sun, A. Y., 2018. Discovering State-Parameter Mappings in Subsurface Models Using Generative Adversarial Networks. *Geophysical Research Letters*, 45, 20 (2018), 11,111–137,146. doi:https://doi.org/10.1029/2018GL080404. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL080404. (cited on page 103)

Sun, L.; Gao, H.; Pan, S.; and Wang, J.-X., 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361 (apr 2020), 112732. doi:10.1016/J.CMA.2019.112732. https://www.sciencedirect.com/science/article/pii/S004578251930622X. (cited on pages 102 and 104)

Sutskever, I.; Vinyals, O.; and Le, Q. V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112. (cited on page 109)

Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C., 2018. A survey on deep transfer learning. In *International conference on artificial neural networks*, 270–279. Springer. (cited on pages 100 and 103)

Tang, M.; Liu, Y.; and Durlofsky, L. J., 2020. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, (2020), 109456. (cited on page 2)

Tencent, 2019. Tencent AI Lab. http://ai.tencent.com/ailab/index.html. (cited on page 99)

Thum, P.; Hesch, W.; and Stüben, K., 2011. LMG2: Accelerating the SAMG Multigrid-Solver in MODFLOW. In *MODFLOW and MORE Conference Proceedings, Colorado, USA*. (cited on pages 9 and 27)

Tonkin, M. and Doherty, J., 2009. Calibration-constrained Monte Carlo analysis of highly parameterized models using subspace techniques. *Water Resources Research*, 45, 12 (2009). (cited on page 56)

Tonkin, M. J. and Doherty, J., 2005. A hybrid regularized inversion methodology

for highly parameterized environmental models. *Water Resources Research*, 41, 10 (2005). (cited on page 27)

Ungemach, P., 1975. Great Artesian Basin groundwater project—explanatory note on digital model package, Great Artesian Basin simulation model (GABSIM). *Geology and Geophysics Record*, 169 (1975). (cited on page 1)

van Dijk, A. I. J. M. and Renzullo, L. J., 2011. Water resource monitoring systems and the role of satellite observations. *Hydrology and Earth System Sciences*, 15, 1 (2011), 39–55. doi:10.5194/hess-15-39-2011. https://hess.copernicus.org/articles/15/39/2011/. (cited on page 106)

Vermeulen, P. T. M., 2013. iMOD version 2.7-Interactive MODeling. *Manual, Deltares*, (2013). (cited on page 62)

Vermeulen, P. T. M.; Te Stroet, C. B. M.; and Heemink, A. W., 2006. Limitations to upscaling of groundwater flow models dominated by surface water interaction. *Water resources research*, 42, 10 (2006). (cited on pages 25, 32, 33, and 62)

Viana, F. A. C.; Haftka, R. T.; and Steffen Jr, V., 2009. Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization*, 39, 4 (2009), 439–457. (cited on page 13)

Vidal, R.; Bruna, J.; Giryes, R.; and Soatto, S., 2017. Mathematics of Deep Learning. *arXiv preprint arXiv:1712.04741*, (2017). (cited on page 107)

Vilhelmsen, T. N.; Christensen, S.; and Mehl, S. W., 2012. Evaluation of MODFLOW-LGR in Connection with a Synthetic Regional-Scale Model. *Groundwater*, 50, 1 (2012), 118–132. (cited on page 23)

Villa-Vialaneix, N.; Follador, M.; Ratto, M.; and Leip, A., 2012. A comparison of eight metamodeling techniques for the simulation of {N2O} fluxes and N leaching from corn crops. *Environmental Modelling and Software*, 34, 0 (2012), 51–66. doi:10.1016/j.envsoft.2011.05.003. http://www.sciencedirect.com/science/article/pii/S1364815211001083. (cited on page 13)

Vrugt, J. A.; Gupta, H. V.; Bouten, W.; and Sorooshian, S., 2003. A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resources Research*, 39, 8 (2003). (cited on page 26)

Vuik, C.; Segal, A.; and Meijerink, J. A., 1999. An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. *Journal of Computational Physics*, 152, 1 (1999), 385–403. (cited on page 6)

Wan, X. and Karniadakis, G. E., 2005. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209, 2 (2005), 617–642. (cited on pages 14 and 128)

Wang, N.; Zhang, D.; Chang, H.; and Li, H., 2020. Deep learning of subsurface flow via theory-guided neural network. *Journal of Hydrology*, 584 (2020), 124700. (cited on page 104)

Watson, T. A.; Doherty, J. E.; and Christensen, S., 2013. Parameter and predictive outcomes of model simplification. *Water Resources Research*, 49, 7 (2013), 3952–3977. doi:10.1002/wrcr.20145. http://dx.doi.org/10.1002/wrcr.20145. (cited on page 6)

Weinan, E. and Engquist, B., 2003. Multiscale modeling and computation. *Notices of the American Mathematical Society*, 50, 9 (2003), 1062–1070. (cited on pages 6 and 21)

Weinan, E.; Engquist, B.; Li, X.; Ren, W.; and Vanden-Eijnden, E., 2007. Heterogeneous multiscale methods: a review. *Communications in Computational Physics*, 2, 3 (2007), 367–450. (cited on page 23)

Weinan, E.; Engquist, B.; and Others, 2002. The Heterogeneous Multi-Scale Method. *arXiv preprint physics/0205048*, (2002). (cited on page 9)

Weinan, E.; Han, J.; and Jentzen, A., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5, 4 (2017), 349–380. (cited on page 107)

Wen, X.-H. and Gómez-Hernández, J. J., 1996. Upscaling hydraulic conductivities in heterogeneous media: An overview. *Journal of Hydrology*, 183, 1 (1996), ix—-xxxii. (cited on pages 25, 32, 33, and 62)

Willcox, K. and Megretski, A., 2005. Fourier series for accurate, stable, reduced-order models in large-scale linear applications. *SIAM Journal on Scientific Computing*, 26, 3 (2005), 944–962. (cited on pages 9, 19, and 20)

Willcox, K. and Peraire, J., 2002. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40, 11 (2002), 2323–2330. (cited on page 6)

Winokur, J. G., 2015. *Adaptive Sparse Grid Approaches to Polynomial Chaos Expansions for Uncertainty Quantification*. Ph.D. thesis, Duke University. (cited on page 40)

Winston, R. B., 2009. ModelMuse-A graphical user interface for MODFLOW-2005 and PHAST: US Geological Survey techniques and methods 6-A29. *Reston (VA)*, (2009). (cited on pages 32 and 62)

Winter, C. L.; Tartakovsky, D. M.; and Guadagnini, A., 2003. Moment Differential Equations for Flow in Highly Heterogeneous Porous Media. *Surveys in Geophysics*, 24, 1 (2003), 81–106. doi:10.1023/A:1022277418570. http://dx.doi.org/10.1023/A{%}3A1022277418570. (cited on page 26)

Woodbury, A. D.; Dunbar, W. S.; and Nour-Omid, B., 1990. Application of the Arnoldi Algorithm to the solution of the advection-dispersion equation. *Water Resources Research*, 26, 10 (1990), 2579–2590. doi:10.1029/WR026i010p02579. http://dx.doi.org/10.1029/WR026i010p02579. (cited on pages 9 and 19)

Xia, C.-A.; Pasetto, D.; Hu, B. X.; Putti, M.; and Guadagnini, A., 2020. Integration of moment equations in a reduced-order modeling strategy for Monte Carlo simulations of groundwater flow. *Journal of Hydrology*, 590 (2020), 125257. (cited on page 2)

Xiu, D., 2009. Fast numerical methods for stochastic computations: a review. *Communications in computational physics*, 5, 2-4 (2009), 242–272. (cited on page 37)

Xiu, D. and Hesthaven, J. S., 2005. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27, 3 (2005), 1118–1139. (cited on pages 13 and 14)

Xiu, D. and Karniadakis, G. E., 2002. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24, 2 (2002), 619–644. (cited on pages 12, 36, 54, 60, and 126)

Xiu, D. and Karniadakis, G. E., 2003. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187, 1 (2003), 137–167. doi:http://dx.doi.org/10.1016/S0021-9991(03)00092-5. http://www.sciencedirect.com/science/article/pii/S0021999103000925. (cited on pages 3 and 34)

Xu, T.; Valocchi, A. J.; Choi, J.; and Amir, E., 2012. Improving groundwater flow model prediction using complementary data-driven models. In *Proceedings of 19th International Conference on Water Resources, Univ. of Ill., Urbana-Champaign, Ill.* (cited on page 6)

Yan, S. and Minsker, B., 2006. Optimal groundwater remediation design using an adaptive neural network genetic algorithm. *Water Resources Research*, 42, 5 (2006), W05407. (cited on pages 9 and 14)

Yoon, H.; Jun, S.-C.; Hyun, Y.; Bae, G.-O.; and Lee, K.-K., 2011. A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. *Journal of hydrology*, 396, 1 (2011), 128–138. (cited on page 9)

Young, P. C. and Ratto, M., 2011. Statistical Emulation of Large Linear Dynamic Models. *Technometrics*, 53, 1 (2011), 29–43. doi:10.1198/TECH.2010.07151. http://dx.doi.org/10.1198/TECH.2010.07151. (cited on pages 6, 9, and 11)

Yuan, Q.; Shen, H.; Li, T.; Li, Z.; Li, S.; Jiang, Y.; Xu, H.; Tan, W.; Yang, Q.; Wang, J.; and Others, 2020. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 241 (2020), 111716. (cited on page 106)

Zhang, L.; Zhang, L.; and Du, B., 2016. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4, 2 (2016), 22–40. (cited on page 106)

Zhou, H.; Gómez-Hernández, J. J.; and Li, L., 2014. Inverse methods in hydrogeology: evolution and recent trends. *Advances in Water Resources*, 63 (2014), 22–37. (cited on page 27)

Zhu, X.; Narayan, A.; and Xiu, D., 2014. Computational Aspects of Stochastic Collocation with Multifidelity Models. *SIAM/ASA Journal on Uncertainty Quantification*, 2, 1 (2014), 444–463. (cited on pages xvi, 61, 63, 64, 71, and 72)

Zhu, X. X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; and Fraundorfer, F., 2017. Deep learning in remote sensing: a review. *arXiv preprint arXiv:1710.03959*, (2017). (cited on page 106)

Zhu, Y. and Zabaras, N., 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366 (aug 2018), 415–447. doi:10.1016/JJCP.2018.04.018. https://www.sciencedirect.com/science/article/pii/S0021999118302341. (cited on pages xviii, 101, 110, and 111)

Zyvoloski, G. A. and Vesselinov, V. V., 2006. An investigation of numerical grid effects in parameter estimation. *Groundwater*, 44, 6 (2006), 814–825. (cited on page 33)