Acta Numerica (2021), pp. 445–554 Printed in the United Kingdom
doi:10.1017/S0962492921000064

Learning physics-based models from data: perspectives from inverse problems and model reduction

Omar Ghattas
Oden Institute for Computational Engineering & Sciences,
Departments of Geological Sciences and Mechanical Engineering,
The University of Texas at Austin, TX 78712, USA
E-mail: omar@oden.utexas.edu

Karen Willcox
Oden Institute for Computational Engineering & Sciences,
Department of Aerospace Engineering & Engineering Mechanics,
The University of Texas at Austin, TX 78712, USA
E-mail: kwillcox@oden.utexas.edu

This article addresses the inference of physics models from data, from the perspectives of inverse problems and model reduction. These fields develop formulations that integrate data into physics-based models while exploiting the fact that many mathematical models of natural and engineered systems exhibit an intrinsically low-dimensional solution manifold. In inverse problems, we seek to infer uncertain components of the inputs from observations of the outputs, while in model reduction we seek low-dimensional models that explicitly capture the salient features of the input–output map through approximation in a low-dimensional subspace. In both cases, the result is a predictive model that reflects data-driven learning yet deeply embeds the underlying physics, and thus can be used for design, control and decision-making, often with quantified uncertainties. We highlight recent developments in scalable and efficient algorithms for inverse problems and model reduction governed by large-scale models in the form of partial differential equations. Several illustrative applications to large-scale complex problems across different domains of science and engineering are provided.

CONTENTS

PART ONE
Learning physics-based models from data

1. Introduction

Computational science and engineering – the combination of mathematical modelling of physical phenomena with numerical analysis and scientific computing – historically has largely focused on the so-called *forward problem*. That is, given the inputs (*e.g.* initial conditions, boundary conditions, sources, geometry, system properties) to a model of a physical system, solve the forward problem to determine the system state and associated output quantities of interest. Significant progress has been made over the past century on solving the forward problem, resulting in powerful and sophisticated numerical discretizations and solvers tailored to a wide spectrum of models representing complex applications across science, engineering, medicine and beyond.

With the maturation of numerical methods for the forward problem – and with the explosion of observational, experimental and simulation data – interest in learning physics models from data has intensified in recent years. This interest has been

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

fuelled by rapid advances in machine learning representations and algorithms. Often lost amidst the excitement of machine learning approaches is the fact that inference of physics-based models from data has long been a subject of ‘classical’ applied mathematics. In particular, decades of research in the distinct but complementary fields of *inverse problems* and *model reduction* have led to rigorous, efficient and scalable methods for inferring uncertain components of complex models – or reduced models in their entirety – from complex and large-scale data. These methods integrate data into physics-based models (conservation laws, constitutive relations, closures, subgrid-scale models, *etc.*) to reduce their uncertainties and yield predictive outputs that can be used for design, control and more general decision-making, often with quantified uncertainties. It is unlikely that extracting generic models solely from data – the province of traditional machine learning – will be reliable and predictive outside the regime in which the data were acquired. Instead, we must learn from data through the lens of physics models.

The purpose of this article is to highlight recent developments in learning physics models from data, focusing on scalable and efficient algorithms for large-scale inverse problems (Part 2) and model reduction (Part 3). In this article we focus on systems governed by partial differential equations (PDEs), although the theory and methods (with appropriate modifications) apply broadly to other types of mathematical models, such as integral equations, ordinary differential equations and N-body problems. Large-scale inverse problems and model reduction are two ostensibly disparate approaches to inference of models, yet the two approaches are inherently connected: in both cases, we exploit the fact that many PDE models of natural and engineered systems exhibit an intrinsically low-dimensional solution manifold. That is, the map from inputs to outputs often admits a low-dimensional representation. This low-dimensionality stems from the common situation in which both the inputs and outputs are infinite-dimensional fields (high-dimensional after discretization), and the map from inputs to outputs is often smoothing or otherwise results in loss of information.

This fundamental property of the map is exploited in different ways. In inverse problems, in which we seek to infer uncertain components of the inputs from observations of the outputs, the intrinsic low-dimensionality is exploited to design fast preconditioned Newton–Krylov methods for solving deterministic inverse problems, and to construct Laplace approximations of Bayesian inverse solutions, both with dimension-independent complexity. In model reduction, we seek low-dimensional models that explicitly capture the salient features of the input–output map through approximation in a low-dimensional subspace. In both cases, we infer a model that reflects data-driven learning yet deeply embeds the underlying physics model and its associated mathematical properties. Indeed, we argue that for complex physical systems, it is only through the mathematical perspectives of inverse theory and model reduction that we can address the crucial challenges of ill-posedness, uncertainty quantification and under-sampling.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Notation

Throughout, we will use lower-case italic letters to denote scalar functions in $\mathbb{R}^2$ or $\mathbb{R}^3$, such as the state u, the parameter field m, the observed data d and the quantity of interest q. We denote the discretized versions of these quantities in $\mathbb{R}^n$ (with n as the discretization dimension) with lower-case upright boldface: $\mathbf{u}$, $\mathbf{m}$, $\mathbf{d}$ and $\mathbf{q}$. Vector functions in $\mathbb{R}^2$ or $\mathbb{R}^3$, such as the velocity field $\boldsymbol{v}$, are denoted by lower-case italic boldface. We use a calligraphic typeface to indicate both infinite-dimensional operators (*e.g.* the forward operator $\mathcal{A}$) and function spaces (*e.g.* the state space $\mathcal{U}$). The standard Sobolev spaces are an exception to this, and are denoted by upper-case italic letters such as H^1 or its vector counterpart $\boldsymbol{H}^1$. Discretized operators are denoted by upper-case upright boldface (*e.g.* the discretized forward operator $\mathbf{A}$). Finally, Greek letters denote scalars, except when convention dictates otherwise. Any exceptions to the above will be clear from the context. When we discuss specific PDE examples, we may unavoidably overload notation locally (*e.g.* p to denote pressure in the Euler equations versus p the adjoint state variable), but we will ensure that there is no ambiguity.

PART TWO
Large-scale inverse problems

Many PDE models of natural and engineered systems exhibit an intrinsically low-dimensional solution manifold. That is, the map from inputs to outputs often admits a low-dimensional representation. The inputs can represent distributed sources, initial conditions, boundary conditions, coefficients or geometry, and the outputs are some function of the state variables obtained by solving the PDEs governing the system – the *forward problem*. The low-dimensionality stems from the common situation in which both the inputs and outputs are often infinite-dimensional fields (high-dimensional after discretizations), and the map from inputs to outputs is often smoothing or otherwise results in loss of information. In the *inverse problem*, we attempt to infer the inputs – *i.e.* the *parameters* – from possibly noisy observations of the outputs – *i.e.* the data. This intrinsic low-dimensionality of the map implies that inference of the parameters from the data is unstable in the presence of noise in those parameter field components that are annihilated by the parameter-to-observable map: the inverse problem is *ill-posed*. The noise can stem from the observation process, or from the model uncertainty itself, and ill-posedness can be further amplified by sparse or incomplete data.

To address the challenges of learning models from data in the large-scale setting of PDEs, we must exploit the low-dimensional structure of the map from inputs to outputs. One way to do this is to construct a reduced model of the forward problem that parsimoniously exploits the low-dimensional solution manifold; this is discussed in Part 3. Alternatively, here in Part 2 we pose the learning-from-data

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

problem as an inverse problem governed by the forward (PDE) problem, and then exploit the low-dimensionality of the parameter-to-observable map to efficiently and scalably recover the informed components of the model at a cost – measured in forward model solutions – that scales independent of the parameter or data dimensions. Our singular focus is on scalable algorithms for large-scale inverse problems stemming from discretizations of PDEs with infinite-dimensional parameter fields (high-dimensional after discretization). For background and further information, the reader is directed to a number of monographs on inverse problems, including those by Banks and Kunisch (1989), Engl, Hanke and Neubauer (1996), Hansen (1998), Vogel (2002), Kaipio and Somersalo (2005), Tarantola (2005), Kirsch (2011), Mueller and Siltanen (2012), Smith (2013), Aster, Borchers and Thurbur (2013), Sullivan (2015), Asch, Bocquet and Nodet (2016), Hanke (2017), Tenorio (2017) and Bardsley (2018).

Section 3 addresses the deterministic inverse problem in the regularization framework; dimension-independence is provided by the inexact Newton–conjugate gradient method. The statistical inverse problem is addressed in the Bayesian framework in Section 4; low-rank approximation of the Hessian of the log likelihood exploits the low-dimensionality, also resulting in dimension-independence. In both cases, efficient computation of the action of the Hessian in an arbitrary direction is critical; the adjoint method for doing so is described in Section 5. We begin Part 2 with a discussion of ill-posedness, and several model elliptic, parabolic and hyperbolic inverse problems intended to illustrate the underlying concepts.

2. Ill-posedness of inverse problems

We begin with a general discussion of ill-posedness in inverse problems, and then give illustrations in the form of elliptic, parabolic and hyperbolic model problems that illustrate different manifestations of ill-posedness. These problems are simple enough to admit explicit characterization of the eigenfunctions and eigenvalues of the parameter-to-observable map, which provides intuition about the limited manner in which the model parameters influence the data, and thus the limited manner in which the model can be inferred from the data. The former property motivates the development of reduced-order models (Part 3) to represent the system behaviour, while the latter motivates the regularization and Bayesian inversion methods of Part 2.

We address the inverse problem of inferring the *model parameter* $m \in X$ from *observed data* $d \in Y$, where typically X and Y are normed spaces, and the relationship between the parameter and the data is represented by

$$\mathcal{F}(m) = d. \tag{2.1}$$

Here the mapping $\mathcal{F} \colon X \to Y$ is the *parameter-to-observable map* representing the process that predicts the data for a given parameter. For our purposes, this mapping is given by a (possibly non-linear) *observation operator* $\mathcal{B}(u) \colon U \to Y$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

that extracts the observables from the state $u \in U$, where u depends on m via solution of a PDE system known as the *forward problem* or the *state equation*, abstractly represented as

$$r(u, m) = 0, \quad r : U \times X \to Z. \tag{2.2}$$

The state equation residual r is assumed to be continuously Fréchet-differentiable. Its Jacobian is assumed to be a continuous linear operator with continuous inverse. The implicit function theorem then implies that $u = u(m)$ depends continuously on m. Thus, to compute $\mathcal{F}$ for a given m, we solve the forward problem to obtain the state, and then apply the observation operator to obtain the predicted observables.

The observation operator can involve localization in space and time (*e.g.* from sensors), imaging of a surface or along a path (*e.g.* from satellites), differentiating to obtain a flux, *etc*. We wish to find the parameter m such that the observables $\mathcal{F}(m)$ fit the observed data d. We focus on parameters that represent infinite-dimensional fields such as a distributed source, initial condition, boundary condition, PDE coefficients or geometry. The PDEs model any physical phenomenon of interest, such as heat conduction, wave propagation, elasticity, viscous flow, electromagnetics, transport or couplings thereof, and the observed state can represent the temperature, pressure, displacement, velocity, stress, electric field, species concentration and so on.

We assume that there exists additive noise η that represents the discrepancy between the data and the model output for the 'true' parameter m_{true}:

$$\mathcal{F}(m_{\text{true}}) + \eta = d. \tag{2.3}$$

This noise can be due to noise in the instrument or its environment, model error, or numerical error in approximating $\mathcal{F}$ on the computer. The precise form of η is typically not known; however, its statistical properties (*e.g.* mean, covariance) are often available. The fundamental difficulty of solving (2.1) is that in the typical case of $\mathcal{F}$ governed by PDEs with infinite-dimensional parameters, the inverse problem is *ill-posed*.

Hadamard (1923) postulated three conditions for a problem of the form (2.1) to be well-posed.

(i) *Existence.* For all $d \in Y$, there exists at least one $m \in X$ such that $F(m) = d$.

(ii) *Uniqueness.* For all $d \in Y$, there is at most one $m \in X$ such that $F(m) = d$.

(iii) *Stability.* The parameter m depends on d continuously.

If any of these three conditions is violated, the inverse problem is said to be ill-posed. Condition (i) can be violated when d fails to belong to the range space of $\mathcal{F}$, for example when the system is over-determined and noise is present. Condition (ii) may not be satisfied when d is finite-dimensional. In this case many different m may fit the data due to a non-trivial null space of $\mathcal{F}$. (For the discretized inverse problem, this condition amounts to fewer non-redundant data than parameters.)

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Finally, $\mathcal{F}^{-1}$, if it exists, may be unbounded, which leads to instability and a violation of condition (iii). Noise in the data may be amplified and pollute the solution. Below we illustrate these forms of ill-posedness via model elliptic, parabolic and hyperbolic inverse problems.

2.1. *Inference of the source term in a Poisson equation*

We begin with the most basic model problem: inference of the source term $m(x)$ of a Poisson equation with constant coefficient $k > 0$,

$$-k\frac{\partial^2 u}{\partial x^2} = m(x), \quad 0 < x < L, \tag{2.4}$$

$$u(0) = u(L) = 0, \tag{2.5}$$

from an observation $d(x)$ of the state $u(x)$ everywhere in the domain $(0, L)$. The parameter-to-observable map $\mathcal{F}$ maps the source $m(x)$ to the observable $u(x)$, and is defined by

$$\mathcal{F}(f) := u(x),$$

where $u(x)$ satisfies (2.4)–(2.5) for a given $m(x)$. It can be verified easily that $\mathcal{F}$ is self-adjoint and its eigenfunctions $v_j(x)$, $j = 1, 2, \ldots, \infty$, are given by

$$v_j(x) = \sqrt{\frac{2}{L}}\,\sin\left(\frac{j\pi x}{L}\right),$$

with corresponding eigenvalues

$$\lambda_j(\mathcal{F}) = \frac{1}{k}\left(\frac{L}{j\pi}\right)^2.$$

We see that $\lambda_j \to 0$ as $j \to \infty$, with increasingly oscillatory eigenfunctions $v_j(x)$. Thus $\mathcal{F}$ is a compact operator. It acts to damp highly oscillatory modes: it is a smoothing operator.

Let us attempt to solve for the source $m(x)$ given $d(x)$, the observation of $u(x)$. Then

$$\mathcal{F}(m) = d \implies m = \mathcal{F}^{-1}d.$$

Making use of the spectral decomposition of $\mathcal{F}$,

$$m = \mathcal{F}^{-1}d = \sum_{j=1}^{\infty}\frac{\langle v_j, d\rangle}{\lambda_j}v_j,$$

where the inner product $\langle v_j, d\rangle = \int_0^L v_j\, d\, \mathrm{d}\boldsymbol{x}$. In order for a solution m to exist, we see that the Fourier coefficients of the data, $\langle v_j, d\rangle$, must decay to zero faster than the eigenvalues λ_j. This is known as the Picard criterion.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

What is the relationship between the resulting $m(x)$ and the true source $m_{\text{true}}(x)$? Recalling the noise model (2.3), we can write

$$\begin{aligned} m &= \mathcal{F}^{-1} d, \\ &= \mathcal{F}^{-1}(\mathcal{F} m_{\text{true}} + \eta), \\ &= m_{\text{true}} + \sum_{j=1}^{\infty} \frac{\langle v_j, \eta \rangle}{\lambda_j} v_j. \end{aligned} \tag{2.6}$$

Or, defining the Fourier components of the noise, $\eta_j = \langle v_j, \eta \rangle$,

$$\|m - m_{\text{true}}\|^2 = \sum_{j=1}^{\infty} \frac{\eta_j^2}{\lambda_j^2}. \tag{2.7}$$

We see that the error in inferring the source, $m - m_{\text{true}}$, can be written as a linear combination of eigenfunctions of $\mathcal{F}$, with weights that depend on the Fourier coefficients of the noise and on the inverse of the corresponding eigenvalues. Since $\lambda_j^{-1} = O(j^2)$, the error grows like $O(j^2)$ in the mode number. The inference of $m(x)$ is thus unstable to small perturbations in the data: a perturbation η in a high-frequency direction $v_j(x)$ will be amplified by a large factor j^2. Modes v_j for which the Fourier coefficients of the noise are larger than λ_j cannot be reliably reconstructed. The inverse problem is thus ill-posed in the sense of Hadamard's instability condition.

For example (with $k\pi^2/L^2 = 1$), a component of the noise of magnitude 10^{-4} in the 100th eigenfunction direction contributes an $O(1)$ error in the inference of m in that mode. This also implies that two observations that differ by $O(10^{-4})$ in the direction of the 100th eigenfunction cannot be used to differentiate between two parameters that differ to $O(1)$. This is important, since in practice noise usually contains high-frequency components. So, in the presence of noise, we effectively lose uniqueness.

These properties carry over to the discretized problem as well. Let $\boldsymbol{F}$ denote the discrete form of $\mathcal{F}$ based on discretizing the Laplacian by a standard central difference three-point stencil on a uniform mesh with spacing h, and then inverting it. The eigenvalues $\lambda_j(\boldsymbol{F})$, $j = 1, 2, \ldots, L/h$, of $\boldsymbol{F}$ are then given by

$$\lambda_j(\boldsymbol{F}) = \frac{h^2}{4k} \csc^2\left(\frac{j\pi h}{2L}\right).$$

The corresponding eigenvectors $\boldsymbol{v}_j$ of $\boldsymbol{F}$ are just interpolants of the continuous eigenvectors $v_j(x)$, that is, the ith component of the jth eigenvector is given by

$$(\boldsymbol{v}_j)_i = \sqrt{\frac{2}{L}} \sin\left(\frac{ij\pi h}{L}\right).$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

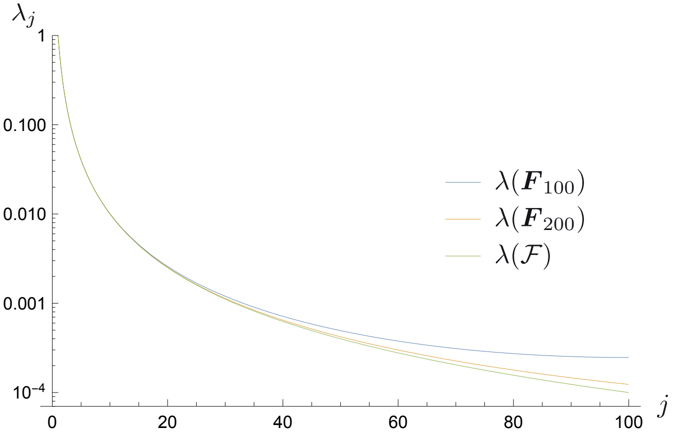


Figure 2.1. Spectrum of the continuous parameter-to-observable operator $\mathcal{F}$ (green) versus that of the discretized operator $\boldsymbol{F}$ for 100 (blue) and 200 (orange) mesh points, indicating four orders of magnitude decay over the first 100 eigenvalues.

In the discrete case, the error in the inference is now

$$\boldsymbol{m} - \boldsymbol{m}_{\text{true}} = \sum_{j=1}^{L/h} \lambda_j^{-1} (\boldsymbol{v}_j^T \boldsymbol{\eta}) \boldsymbol{v}_j.$$

We see that, similar to the continuous case, inference of rough components (*i.e.* those for which λ_j is small) is unstable: the error in the jth eigenvector direction is amplified by its Fourier coefficient divided by λ_j. When the mesh size h is sufficiently small relative to the frequency j, we can invoke a small angle approximation for cosecant to give the asymptotic expression

$$\lambda_j(\boldsymbol{F})) \approx \frac{1}{k} \left(\frac{L}{j\pi} \right)^2 \quad \text{for } jh \ll \frac{\pi}{2L},$$

which for fixed frequency shows that the discrete eigenvalues converge to those of the continuous operator:

$$\lambda_j(\boldsymbol{F}) \to \lambda_j(\mathcal{F}) \quad \text{as } h \to 0.$$

Figure 2.1 plots the first 100 eigenvalues of the continuous operator $\mathcal{F}$ as well as those of the discrete operator for two discretizations (100 and 200 mesh points). As can be seen, the discrete eigenvalues converge from above. Thus discretization has a regularizing effect, and for a sufficiently coarse mesh we might be able to stably infer $m(x)$, especially if the noise level is low. However, as the mesh is refined, the discrete eigenvalues approach their continuous counterparts, and the inference becomes unstable. This is true even for observations that are free of instrument noise: round-off errors alone are sufficient to trigger instabilities.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

This example demonstrates that attempting to infer rough components of the source of a Poisson equation from observations of the state is unstable. The data – even when infinite-dimensional and possessing small-amplitude noise – inform just a low-dimensional subspace of modes of the source. This is evidenced by the four-orders-of-magnitude eigenvalue reduction over the first 100 eigenvalues. Since the inferable components are smooth, they do not depend on the discretization (beyond a sufficiently fine mesh), and thus the data inform the model in an effectively finite-dimensional subspace, which remains independent of the parameter and data dimensions as they increase with mesh refinement.

While this is perhaps the simplest PDE-based inverse problem one can imagine, it does illustrate the futility of fully learning parameter fields (let alone entire models) from data, a property that characterizes many large-scale inverse problems that are governed by more complex forward models and more complex observation operators. Instead, we must be content to learn just the data-informed modes, which are low-dimensional. What constitutes the low-dimensionality and the informed modes will depend on the character of the parameter-to-observable map and the associated noise model. For the Poisson source inversion problem, the eigenvalues decay like j^{-2} for the one-dimensional problem. More generally, for a Poisson operator in ω space dimensions, the eigenvalues decay like $j^{-2/\omega}$. This algebraic decay rate is characteristic of other inverse problems governed by elliptic forward problems as well; these problems are referred to as *mildly* or *moderately ill-posed*. In the next section we will obtain a significantly worse decay rate for a parabolic inverse problem.

2.2. *Inference of initial condition in a heat equation*

Here we consider the problem of inferring the initial condition in the one-dimensional heat equation from observations of the temperature field at time T. Let $u(x,t)$ denote the temperature field and $u(x,0) = m(x)$ the initial temperature. Given the length of the rod L, the thermal diffusivity $k > 0$, the final time T and homogeneous Dirichlet boundary conditions, the parameter-to-observable map $\mathcal{F}(m)$ can be written as

$$\mathcal{F}(m) := u(x,T),$$

where for a given $m(x)$, the observable $u(x,T)$ is given by the solution at observation time T of the heat equation

$$\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = 0, \qquad 0 < x < L,\ 0 < t \le T, \tag{2.8}$$

$$u(x,0) = m(x), \qquad 0 < x < L, \tag{2.9}$$

$$u(0,t) = u(L,t) = 0, \qquad 0 < t \le T. \tag{2.10}$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

As with the Poisson source inversion problem, $\mathcal{F}$ is self-adjoint and compact with eigenfunctions v_j, $j = 1, \ldots, \infty$, given by

$$v_j(x) = \sqrt{\frac{2}{L}}\,\sin\left(\frac{j\pi x}{L}\right).$$

However, the eigenvalues are now given by

$$\lambda_j = \mathrm{e}^{-kT(\pi j/L)^2},$$

which decay exponentially. The rapid decay of the eigenvalues is a consequence of the information lost in diffusion of the initial temperature field. The more oscillatory modes of the initial temperature field decay more rapidly – the smaller eigenvalues correspond to more oscillatory eigenfunctions. Thus $\mathcal{F}$ is again a smoothing operator. In contrast with the Poisson source inversion problem, here we see nearly four orders of magnitude drop in the eigenvalues over just the first *three* eigenvalues (for $kT\pi^2/L^2 = 1$). Thus the noise in the third mode is amplified by a factor of $O(10^4)$. In fact, assuming *exact* observational data, rounding errors alone (of order 10^{-16} for double precision) will already corrupt the sixth mode. As a result, we can hope to reliably recover only a handful of modes. A larger diffusion coefficient or a later observation time will result in even more rapid decay in the eigenvalues and thus further deterioration in the ability to infer the initial condition. Inverse problems exhibiting exponential decay such as we see here for the inverse heat equation are termed *severely ill-posed*. This is typical of parabolic problems.

2.3. *Inference of initial condition in a wave equation*

Now we study the stability of the inverse problem governed by the one-dimensional wave equation. The forward problem is to find the transverse displacement $u(x, t)$ of a cable of length L, tension k and mass density ρ, and fixed at both ends. The cable is initially at rest and is plucked with an initial displacement of $u(x, 0) = m(x)$. The forward problem is then: Given $m(x)$, solve

$$\begin{aligned}
\frac{\partial u^2}{\partial t^2} - c^2\frac{\partial^2 u}{\partial x^2} &= 0, &\quad& 0 < x < L,\ 0 < t \leq T,\\
u(x,0) &= m(x), && 0 < x < L,\\
\frac{\partial u}{\partial t}(x,0) &= 0, && 0 < x < L,\\
u(0,t) = u(L,t) &= 0, && 0 < t \leq T,
\end{aligned}$$

for the displacement field $u(x, t)$, where $c := \sqrt{k/\rho}$ is the wave propagation speed. The inverse problem is to infer the initial displacement $m(x)$ from observation of

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

the cable's position at a later time $t = T$. Thus the parameter-to-observable map is given by $\mathcal{F}(m) := u(x, T)$.

Once again, the eigenfunctions v_j, $j = 1, \ldots, \infty$, of $\mathcal{F}$ are given by

$$v_j(x) = \sqrt{\frac{2}{L}}\, \sin\left(\frac{j\pi x}{L}\right).$$

However, the eigenvalues are now

$$\lambda_j = \cos\left(\frac{j\pi cT}{L}\right),$$

which do not decay but instead oscillate. In fact, if we choose the observation time to be $2L/c$, *i.e.* the time taken for the cable to return to its initial position, we see that the $\lambda_j = 1$ for $j = 1, \ldots, \infty$. This results in a perfect reconstruction of the initial displacement, and the inverse problem is well-posed.

However, this one-dimensional model problem fails to capture the ill-posedness characteristic of realistic inverse wave propagation problems (Colton and Kress 2019). For example, an important class of problems is seismic inversion (Symes 2009), in which one seeks to infer mechanical properties of Earth's interior (such as wave speeds) from reflected waves produced by seismic sources and measured at receiver locations on its surface. (This is a coefficient inverse problem, an example of which we will see in the next section.) In such cases, ill-posedness can arise for multiple reasons. First, the wavefield is observed at distinct receiver locations, not everywhere. Second, real Earth media are dissipative: wave amplitudes are attenuated, resulting in a loss of information. Third, the Earth is a semi-infinite medium, and geometric spreading of waves results in amplitude decay and again loss of information. Fourth, the subsurface typically contains multiple interfaces between distinct rock types, and primary reflections are often buried within multiple reflected signals, making it difficult for the inversion to disentangle the information contained within them. Fifth, features beneath high-contrast interfaces will be difficult to recover due to the limited ability of waves to reach them. And sixth, seismic waves are typically band-limited; feature sizes can be reconstructed at best to within an order of a wavelength, and thus sub-wavelength length scales belong to the null space of the parameter-to-observable map.

While ultimately most realistic inverse wave propagation problems are ill-posed, the model problem considered in this section at least highlights one important feature of hyperbolic inverse problems: preservation – or at least not significant loss – of information. This typically manifests as slowly decaying eigenvalues or singular values. In the context of seismic inversion, the rate of decay weakens as the numbers of sources and receivers increase, and as the source frequencies increase. This presents difficulties for the low-rank-based algorithms discussed in the next two sections. We will return to this point at the conclusion of Section 3.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

2.4. *Inference of coefficient of a Poisson equation*

The final model problem is to infer the coefficient of a Poisson equation (a nonlinear inverse problem) from observations of the entire solution as well as at points in the domain. The results in this section are from Flath (2013). We reparametrize the coefficient in terms of its logarithm $m(x)$ (to maintain its positivity). The forward problem is: Given $m(x)$, find the state $u(x)$ by solving

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(\mathrm{e}^{m}\frac{\mathrm{d}u}{\mathrm{d}x}\right) &= 0 \quad \text{for } x \in (0, L), \\ u(0) &= u_0, \\ u(L) &= u_L. \end{aligned}$$

The inverse problem is to infer the log coefficient $m(x)$ from observations of $u(x)$. We will consider two types of observations: (i) *full observations*, *i.e.* $u(x), x \in (0, L)$), and (ii) *point observations* at n_d equally spaced points $x_k = L/n_d(k - 1/2)$, $k = 1, \dots, n_d$. Although the forward problem is linear, the inverse problem is nonlinear, since the parameter appears as a (log) coefficient in the forward problem. So we linearize the parameter-to-observable map around a constant log coefficient m_0 and study its spectral properties. The linearized map $\mathcal{F}$ is no longer self-adjoint, but we can derive expressions for its singular values and functions from the eigenvalues and eigenfunctions of $\mathcal{F}^*\mathcal{F}$, which is known as the *Gauss–Newton Hessian* (more on this in Section 3). Here $\mathcal{F}^*$ is the adjoint of $\mathcal{F}$.

In the full observation case, the singular values and right singular functions (σ_j, v_j) of $\mathcal{F}$ are given for $j = 1, \dots, \infty$ by

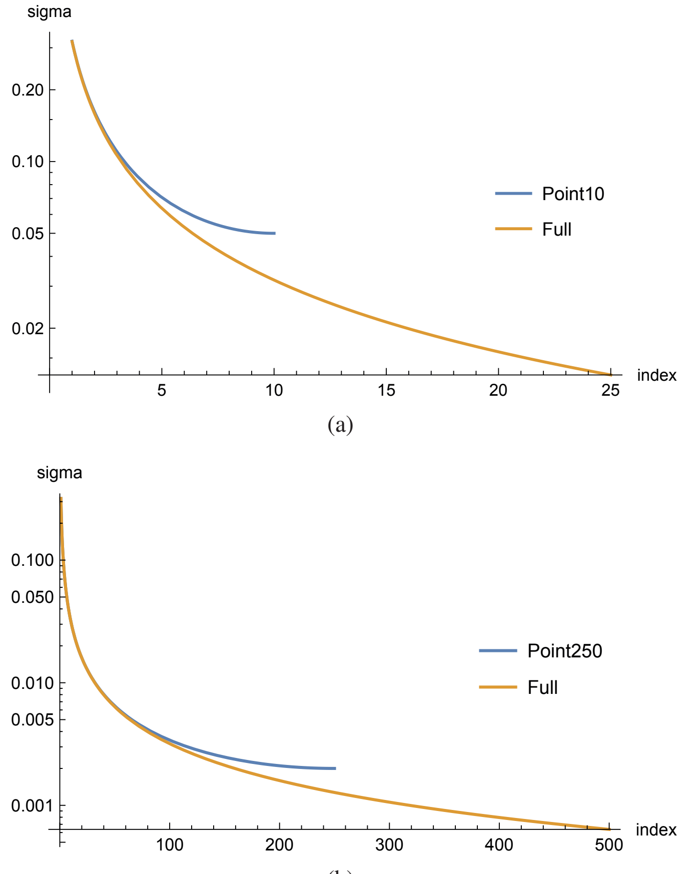
$$\begin{aligned} \sigma_j^{\text{full}} &= \frac{u_L - u_0}{j\pi}, \\ v_j^{\text{full}}(x) &= \sqrt{\frac{2}{L}}\cos\left(\frac{j\pi x}{L}\right). \end{aligned}$$

The singular values decay algebraically as for the Poisson source inversion problem, but at a reduced rate, $O(j^{-1})$. As with the previous elliptic and parabolic model problems, $\mathcal{F}$ damps more oscillatory modes more strongly, and is thus a smoothing operator.

In the case of point observations, the singular values of $\mathcal{F}$ are given by

$$\sigma_j^{\text{point}} = \begin{cases} \dfrac{u_L - u_0}{2n_d\,\sin(j\pi/2n_d)} & j = 1, \dots, n_d, \\ 0 & j = n_d + 1, \dots, \infty. \end{cases}$$

The singular values σ_j^{point} are seen to decay with j, though only n_d of them are nonzero. This makes sense, since there are only n_d point observations, and thus at most just n_d modes can be informed by the data. Thus $\mathcal{F}$ has a finite-dimensional range space. Since the parameter is infinite-dimensional, this introduces severe nonuniqueness to the inverse problem. Small angle approximation of the sine shows



(a)

(b)

Figure 2.2. (a) First 25 singular values of $\mathcal{F}$ for full observations (orange) versus those for 10 point observations (blue). (b) First 500 singular values of $\mathcal{F}$ for full observations (orange) versus those for 250 point observations (blue).

that the point observation singular values for which $j \ll n_d$ approximate those of the full observations. As the number of observations increases, the former converge to the latter, *i.e.* $\sigma_j^{\text{point}} \to \sigma_j^{\text{full}}$ as $n_d \to \infty$. (The difference in the squared singular values decays like n_d^{-2}.) The right singular functions turn out to be piecewise constant interpolants of the full-observation right singular functions, v_j^{full}, where the interpolation points are the midpoints between successive observation points x_j, and jump discontinuities occur at the x_j. Thus the singular functions converge as well.

It may be tempting to assume that in the case of n_d finite observations, we are entitled to infer n_d modes of the parameter, since the range space dimension of $\mathcal{F}$ is n_d. Certainly for small enough n_d this is true. However, increasing numbers of observations bring diminishing returns: less and less information is contained in the data, and we are back to the fundamental difficulty underlying the first three model problems, *i.e.* eigenvalues/singular values decaying to zero and leading to instability in the inference of the more oscillatory modes. Thus, for sufficiently large values of n_d, the smallest non-zero σ_j will amplify the noise (by a factor σ_j^{-1}) leading to an unstable reconstruction. So the number of inferrable modes may be just a fraction of n_d. To illustrate these points, Figure 2.2 plots the singular values of $\mathcal{F}$ for three cases: 10 point observations (plot (a)), 250 point observations (plot (b)), and full observations (both). The convergence of the σ_j^{point} to σ_j^{full} with increasing n_d can be seen. With 10 observations, the singular values (blue curve in (a)) remain sufficiently large that the inference will avoid pollution from observational noise of order 0.05. However, plot (b) shows that with 250 point observations, high-frequency noise of order 0.002 can lead to unstable reconstructions.

Thus, restricting the observations to a finite collection of points introduces severe ill-posedness in the now under-determined inverse problem, even when the full-observation case is only moderately (j^{-1}) ill-posed. Moreover, increasing the number of observations leads to diminishing returns, as the point observation singular values converge to their full observation counterparts and instability ensues for small singular values. This discussion of point observations is of course applicable to many other inverse problems.

2.5. *Summary*

The model elliptic, parabolic and hyperbolic inverse problems discussed above illustrate ill-posedness due to lack of existence, uniqueness or stability. (The unrealistically formulated hyperbolic inverse problem was in fact well-posed, but we discussed a number of features that make more realistic variants of the problem ill-posed.) Lack of existence (due to inconsistent data) and uniqueness (due to sparse data) can be addressed by solving (2.1) using the Moore–Penrose generalized inverse $\mathcal{F}^\dagger$ (Engl *et al.* 1996). The stability condition is more pernicious, and is the central cause of numerical difficulties in solving inverse problems. The lack of stability stems from the collapse of the spectrum of $\mathcal{F}$, the dominant eigenvalues of which correspond to modes of m that can be reliably inferred from the data. (In Section 4 we will make an information-theoretic connection between the spectrum of $\mathcal{F}$ and the information contained in the data.)

As (2.6) and (2.7) make evident, (inverses of) small eigenvalues amplify noise components in directions of the corresponding eigenfunctions, leading to unstable inference of these modes (for non-self-adjoint $\mathcal{F}$, the singular value decomposition replaces the spectral decomposition). The more rapid the decay of the spectrum of $\mathcal{F}$, and the higher the noise level, the fewer eigenvalues will be above the noise level, and the fewer modes of m that can be reliably inferred from the data. The model problems exhibit algebraic (in the elliptic case) or exponential (in the parabolic case) decay of the spectrum, accumulating at zero, reflecting different degrees of

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

severity of ill-posedness. Moreover, even for a finite number of observations n_d – meaning that the range space of $\mathcal{F}$ is finite-dimensional – if n_d is large enough, there can be significant decay of the spectrum (due to near-redundancy in information contained within the data) such that small eigenvalues can trigger instability.

The loss of information in $\mathcal{F}(m)$, the map from parameters to observables – and the resulting inability to recover the lost information via the inverse map – is a fundamental property of the inverse problem. No amount of mathematical wizardry can recover the lost information. The best we can hope to do is recover components of m lying in the 'effective' range space of $\mathcal{F}$, that is, those eigenfunction modes corresponding to eigenvalues that are large enough to dominate the noise. How to do this efficiently and scalably for problems characterized by high-dimensional parameters (resulting from discretization of m) and for expensive forward problems (involving solutions of PDEs) will be the subject of the next two sections, in the context of both regularization and Bayesian frameworks. The essential idea is to design algorithms that exploit the intrinsic low-dimensionality of the information contained within the data by requiring an amount of work (measured in PDE solves) that scales with the intrinsic 'information dimension', not the apparent dimension.

As we will see, these algorithms depend fundamentally on the rapid spectral decay (of $\mathcal{F}^*\mathcal{F}$). Rapid spectral decay has been demonstrated, not just for the model problems of this section, but for a broad set of inverse problems arising in science and engineering, either explicitly through low-rank approximation of the Hessian of the data misfit (Section 4) or implicitly through rapid convergence of conjugate gradients for the Hessian system (Section 3). These include *ice sheet dynamics* (Petra *et al.* 2012, Petra, Martin, Stadler and Ghattas 2014, Isaac, Petra, Stadler and Ghattas 2015, Zhu *et al.* 2016*b*, Babaniyi, Nicholson, Villa and Petra 2021), *shape and medium acoustic and electromagnetic scattering* (Akçelik, Biros and Ghattas 2002, Bui-Thanh and Ghattas 2012*a*,*b*, 2013, Chaillat and Biros 2012, Ambartsumyan *et al.* 2020, O'Leary-Roseberry, Villa, Chen and Ghattas 2020, Chen, Haberman and Ghattas 2021), *seismic wave propagation* (Akçelik *et al.* 2003*a*, Epanomeritakis, Akçelik, Ghattas and Bielak 2008, Martin, Wilcox, Burstedde and Ghattas 2012, Bui-Thanh *et al.* 2012*a*, Bui-Thanh, Ghattas, Martin and Stadler 2013, Zhu *et al.* 2016*a*), *mantle convection* (Worthen *et al.* 2014), *viscous incompressible flow* (Biros and Ghattas 1999, 2005*a*,*b*, Yang, Stadler, Moser and Ghattas 2011), *atmospheric transport* (Akçelik *et al.* 2003*b*, 2005, Bashir *et al.* 2008, Flath *et al.* 2011, Alexanderian, Petra, Stadler and Ghattas 2014, Wu, Chen and Ghattas 2020, Villa, Petra and Ghattas 2021), *ocean dynamics* (Kalmikov and Heimbach 2014), *turbulent combustion* (Chen, Villa and Ghattas 2019*a*), *poroelasticity* (Hesse and Stadler 2014, Alghamdi, Hesse, Chen and Ghattas 2020, Alghamdi *et al.* 2021), *infectious disease spread* (Chen and Ghattas 2020*a*,*b*), *tumour growth modelling* (Subramanian, Scheufele, Mehl and Biros 2020), *tsunami extreme events* (Tong, Vanden-Eijnden and Stadler 2020), *joint inversion* (Crestel, Stadler and Ghattas 2018) and *subsurface flow* (Alexanderian, Petra, Stadler and Ghattas 2016, 2017, Chen, Villa and Ghattas 2017, Chen and Ghattas 2019, 2020*c*).

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

3. Regularization framework and inexact Newton-CG

3.1. Regularization framework

We return to the task of solving the inverse problem (2.1), that is, attempting to infer the parameter $m \in X$ from given data $d \in Y$ from

$$\mathcal{F}(m) = d.$$

We saw in Section 2 that a fundamental feature of many ill-posed inverse problems is the decay of the spectrum of $\mathcal{F}$ to zero. As we saw in (2.6) and (2.7), small eigenvalues of $\mathcal{F}$ (smaller than the Fourier coefficients of the noise) amplify the noise in the data, rendering the inversion unstable for small enough eigenvalues and large enough noise. Since these eigenvalues correspond to eigenfunction modes that do not influence the data (above the noise threshold) – and are thus unrecoverable – a reasonable strategy is to annihilate them. When $\mathcal{F}$ is a linear operator, this can be accomplished by a *truncated singular value decomposition* (TSVD), in which the eigenvalues of $\mathcal{F}$ below a certain threshold are truncated, so that TSVD acts like a filter. A preferable alternative is *Tikhonov regularization*, which most strongly attenuates the smallest eigenvalues, with the amount of attenuation falling off with larger eigenvalues. Thus Tikhonov regularization acts like a 'softer' version of TSVD. Rather than apply the regularization as a filter (with varying filter factor), an equivalent formulation is to solve the optimization problem

$$\min_{m \in X} \phi(m) := \frac{1}{2}\|\mathcal{F}(m) - d\|^2 + \frac{\beta}{2}\|m\|^2, \tag{3.1}$$

where the observable $\mathcal{F}(m) \in Y$ depends on the parameter m via solution of the governing PDE (2.2), yielding $u \in U$ and application of an observation operator, that is,

$$\mathcal{F}(m) := \mathcal{B}(u(m)), \quad \text{where } u(m) \text{ solves } r(u, m) = 0 \text{ for given } m.$$

The regularization term $(\beta/2)\|m\|^2$ imposes a penalty on the norm of m, with the *regularization parameter* β controlling the strength of the penalty. Solution of the regularized least-squares problem (3.1) has several advantages over the filter formulation: (i) it avoids computation of the spectral decomposition or SVD of $\mathcal{F}$, (ii) it extends easily to non-linear parameter-to-observable maps, (iii) it allows for more general norms and regularization operators, (iv) it permits additional equality or inequality constraints (such as bounds) on m, and (v) it admits different norms besides L^2 in the data misfit term $\|\mathcal{F}(m) - d\|$, such as the more robust (to outliers) L^1.

The most popular regularization is the H^1 seminorm $\int \nabla m \cdot \nabla m \,\mathrm{d}\boldsymbol{x}$, which amounts to an increasing penalty on the more oscillatory modes of m. Its specific action for the model problems described in Section 2 is to damp the components of m in the eigenfunction directions $v_j(x)$ with increasing weights j^2 (the eigenvalues of the one-dimensional Laplacian underlying the H^1 seminorm), thereby stabilizing

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

the inference of those components. Many other choices of regularization terms have been employed, including other Hilbert space norms expressing different smoothness preferences, total variation (which expresses a preference for piecewise constant m), total generalized variation (piecewise smooth), sparsity-promoting and directional sparsity-promoting functionals, and a variety of statistical and data-driven regularizers. See the comprehensive accounts in Arridge, Maass, Öktem and Schönlieb (2019) and Benning and Burger (2018).

The regularization parameter β is chosen to balance the error due to instability caused by noise amplification and the error due to bias caused by the regularization. The tension is between too large a β, in which case modes that are well informed by the data are damped, and too small a β, in which case modes that are poorly informed by the data become unstable in the reconstruction. If $\delta = \|\eta\|$ represents the magnitude of the noise, it can be shown (*e.g.* Vogel 2002) that the choice $\beta = O(\delta^p)$, $0 < p < 2$ guarantees that both sources of error vanish as $\delta \to 0$. However, this *a priori* estimate does not provide a constructive formula for choosing β. Instead, we appeal to an *a posteriori* formula. The Morozov discrepancy principle is a popular choice: Find the largest value of β that satisfies

$$\|\mathcal{F}(m_\beta) - d\| \leq \delta, \tag{3.2}$$

where

$$m_\beta := \operatorname*{arg\,min}_{m \in X} \left\{ \frac{1}{2}\|\mathcal{F}(m) - d\|^2 + \frac{\beta}{2}\|m\|^2 \right\}.$$

Thus we seek the smallest regularization penalty $\|m\|^2$ that results in an m that fits the data to within the noise. The resulting choice of β^* can also be derived from the following optimization problem:

$$\begin{gathered} \min_{m \in X} \frac{1}{2}\|m\|^2, \\ \text{subject to } \|\mathcal{F}(m) - d\|^2 = \delta^2. \end{gathered}$$

The value of the Lagrange multiplier μ^* for the constraint at optimality is then related to Morozov's choice of β by $\beta^* = 1/(2\mu^*)$.

3.2. Inexact Newton–conjugate gradient methods

We now turn to the main topic of this section, which is efficient and scalable Newton algorithms for the optimization problem (3.1). This problem belongs to the class of *PDE-constrained optimization problems*; for discussion of the underlying theoretical, as well as other computational methods, see *e.g.* Biegler, Ghattas, Heinkenschloss and van Bloemen Waanders (2003), Gunzburger (2003), Biegler *et al.* (2007), Ito and Kunisch (2008), Hinze, Pinnau, Ulbrich and Ulbrich (2009), Tröltzsch (2010), Ulbrich (2011), Borzì and Schulz (2012), Leugering *et al.* (2012), De los Reyes (2015) and Antil, Kouri, Lacasse and Ridzal (2018). By virtue of the

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

implicit function theorem, we will consider the state u to be an implicit function of m via solution of the PDE $r(u, m) = 0$, resulting in the unconstrained optimization problem (3.1). While this greatly simplifies the optimization problem, this implicit dependence complicates the computation of the gradient and Hessian of the objective functional $\Phi(m)$, which are required for the Newton iterations. (A detailed concrete example of how to compute them is provided in Section 5.) For stationary and highly non-linear problems, significantly greater efficiency may be obtained by solving the inverse problem as a constrained optimization problem, which relaxes the PDE solution by imposing the PDEs as constraints, iterating toward both feasibility (satisfying the PDEs) and optimality (minimizing the objective) (*e.g.* Biros and Ghattas 2005*a*,*b*).

We assume (3.1) has been suitably discretized (see Section 5 for a discussion of discretization issues), resulting in the finite-dimensional optimization problem

$$\min_{\mathbf{m}\in\mathbb{R}^{n_m}} \Phi(\mathbf{m}) := \frac{1}{2}\|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|^2 + \frac{\beta}{2}\|\mathbf{m}\|^2_{\mathbf{H}^{\mathrm{reg}}}, \tag{3.3}$$

where $\mathbf{m} \in \mathbb{R}^{n_m}$ and $\mathbf{d} \in \mathbb{R}^{n_d}$ are the finite-dimensional representations of the parameter and data, $\mathbf{f}\colon \mathbb{R}^{n_m} \longrightarrow \mathbb{R}^{n_d}$ is the discrete parameter-to-observable map, Φ is the discrete regularized least-squares function, $\mathbf{H}^{\mathrm{reg}}$ is a regularization operator (for simplicity, we assume quadratic regularization) and $\mathbf{f}$ depends implicitly on $\mathbf{m}$ via solution of a discretized forward PDE problem.

In this section we will describe the inexact Newton–conjugate gradient method for solving (3.3) with specialization to large-scale inverse problems governed by PDEs. We describe the components of the method that are salient for the targeted problem class. See Kelley (1999) and Nocedal and Wright (2006) for further details and analysis. Newton's method is the gold standard for fast solution of discretized infinite-dimensional optimization problems such as (3.3). For a wide variety of problems, it converges in a mesh-independent (or nearly mesh-independent) number of iterations (Allgower, Böhmer, Potra and Rheinboldt 1986, Kelley and Sachs 1991, Heinkenschloss 1993). Our experience is that this is generally the most effective method for this class of problems, converging with far fewer forward PDE solves than methods that rely only on gradient information. Define $\mathbf{g}(\mathbf{m})$ as the gradient vector of $\Phi(\mathbf{m})$ and $\mathbf{H}(\mathbf{m})$ as its Hessian matrix, that is,

$$\begin{aligned}\mathbf{g}(\mathbf{m}) \in \mathbb{R}^{n_m} &:= \frac{D\Phi(\mathbf{m})}{D\mathbf{m}},\\ \mathbf{H}(\mathbf{m}) \in \mathbb{R}^{n_m\times n_m} &:= \frac{D^2\Phi(\mathbf{m})}{D\mathbf{m}^2},\end{aligned}$$

where $D/D\mathbf{m}$ denotes total derivative (taking into account implicit dependence of $\mathbf{f}$ on $\mathbf{m}$ via solution of the PDE). The kth Newton iteration solves the linear system

$$\mathbf{H}_k\mathbf{p}_k = -\mathbf{g}_k \tag{3.4}$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

for the *search direction* $\mathbf{p}_k$, followed by the parameter update

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{p}_k.$$

Here α_k is a step length chosen according to an appropriate line search strategy, and $\mathbf{g}_k := \mathbf{g}(\mathbf{m}_k)$ and $\mathbf{H}_k := \mathbf{H}(\mathbf{m}_k)$. Assuming Φ is twice Lipschitz continuously differentiable, then near a local minimizer $\mathbf{m}^*$ the Newton iteration (3.4) converges q-quadratically. Global convergence (*i.e.* from any starting point) to at least a stationary point ($\mathbf{g}(\mathbf{m}) = \mathbf{0}$) can be established under reasonable assumptions (which are frequently satisfied in practice) by:

(i) ensuring that $\mathbf{p}_k$ is a descent direction, *i.e.* $\mathbf{p}_k^\mathsf{T} \mathbf{g}_k < 0$,

(ii) taking a step length α_k that ensures sufficient decrease of $\Phi(\mathbf{m}_{k+1})$ and sufficiently large steps $\alpha_k \mathbf{p}_k$.

Condition (i) can be satisfied by maintaining a positive definite approximation to the Hessian. Condition (ii) can be satisfied by an *Armijo backtracking line search.*

1 Start with a trial step of $\alpha_k^0 = 1$.

2 Check for sufficient descent:

$$\frac{\Phi(\mathbf{m}_k + \alpha_k^i \mathbf{p}_k) - \Phi(\mathbf{m}_k)}{\alpha_k^i \mathbf{p}_k^\mathsf{T} \mathbf{g}_k} \geq c.$$

3 (a) If step 2 is satisfied, $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k^i \mathbf{p}_k$.
  (b) Else, backtrack using $\alpha_k^{i+1} = \gamma \alpha_k^i$ and repeat from 2.

Typical choices of constants are $c = 10^{-4}$ and $\gamma = 1/2$. Initializing the backtracking with $\alpha = 1$ asymptotically ensures the correct initial choice of step length, since Newton's quadratic approximation of Φ becomes increasingly accurate near a minimizer.

Several difficulties are encountered in specializing the globalized Newton optimization framework outlined above to large-scale inverse problems governed by PDEs.

- The major challenge faced by Newton methods for the targeted class of inverse problems is that explicit construction of the Hessian $\mathbf{H}(\mathbf{m})$ is typically intractable. Each column of the Hessian requires the solution of a pair of linearized forward/adjoint PDEs (see Section 5); these PDE solution costs overwhelmingly dominate all other costs of solving the optimization problem (such as linear algebra). For example, in the ice sheet flow inverse problem discussed in Section 6, solving a single linearized forward problem requires about one minute on 1024 processor cores. Thus constructing a single Hessian for the one million parameters that characterize this problem would require about two years of supercomputing time! How can we enjoy the benefits of Newton's asymptotic quadratic convergence without explicitly forming this Hessian?

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

- Unless the inverse problem is linear, the usual situation is that $\Phi(\mathbf{m})$ is nonconvex. As a result, the Hessian can be indefinite away from a local minimizer. To ensure a descent direction far from a local minimizer, we must maintain a positive definite approximation to the Hessian, $\tilde{\mathbf{H}}$. How can this be done when the Hessian itself cannot even be formed? Moreover, how can we avoid modifying the Hessian in the vicinity of a local minimizer so as not to interfere with the asymptotic convergence rate?
- Since manipulating the Hessian is so expensive (in terms of PDE solves), can we lessen our reliance on Newton steps far from a minimizer (when Newton is only linearly converging)?
- The Hessian can be decomposed into the Hessian of the data misfit functional and the Hessian of the regularization functional:

  $$\mathbf{H} = \mathbf{H}^{\text{data}} + \beta \mathbf{H}^{\text{reg}}.$$

  As we saw in the model problems in Section 2, typically for ill-posed inverse problems, $\mathbf{H}^{\text{data}}$ is a (discretized) compact operator. On the other hand, typically it is the highly oscillatory modes of $\mathbf{m}$ that are poorly determined by the data and require regularization, and as a result the regularization is typically smoothing. Thus the regularization operator $\mathbf{H}^{\text{reg}}$ is often chosen as a elliptic differential operator. Can we take advantage of this 'compact + differential' structure in addressing the two challenges listed above?

The *inexact Newton–conjugate gradient* (Newton-CG) method addresses all of these issues. The basic idea is to solve (3.4) with the (linear) CG method in a matrix-free manner, terminating early to enforce positive definiteness and avoid over-solving (Dembo, Eisenstat and Steihaug 1982, Eisenstat and Walker 1996).

1 Solve $\mathbf{H}_k \mathbf{p}_k = -\mathbf{g}_k$ iteratively using preconditioned CG.

   (a) At each CG iteration where $\mathbf{d}^i$ is the ith CG search direction, form the product of $\mathbf{H}_k$ with $\mathbf{d}^i$ in a matrix-free manner via a pair of 'incremental' forward/adjoint PDE solves (see Section 5).

   (b) Terminate CG iteration when a direction of negative curvature is encountered, *i.e.* when

   $$\mathbf{d}^{i^{\mathsf{T}}} \mathbf{H}_k \mathbf{d}^i \leq 0,$$

   and use the current CG solution $\mathbf{p}_k^i$ as the Newton search direction (unless it is the first CG iteration, in which case use the steepest descent direction).

   (c) Also terminate the CG iterations when the Newton system is solved 'accurately enough', *i.e.* when

   $$\|\mathbf{H}_k \mathbf{p}_k^i + \mathbf{g}_k\| \leq \eta_k \|\mathbf{g}_k\|,$$

   where the tolerance with which this system is solved tightens as the minimizer is approached ($\|\mathbf{g}_k\| \to 0$ as $k \to \infty$). The choice of the *forcing term* η_k affects the Newton convergence rate:

$$\eta_k = \begin{cases} 0.5 & \text{linear,} \\ \min(0.5, \sqrt{\|\mathbf{g}_k\|}) & \text{superlinear,} \\ \min(0.5, \|\mathbf{g}_k\|) & \text{quadratic.} \end{cases}$$

(d) Precondition the CG iterations with regularization operator $\mathbf{H}^{\text{reg}}$.

2 Perform Armijo backtracking line search with $\mathbf{p}_k^i$ as Newton search direction.

3 Update $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{p}_k^i$ and repeat from step 1 until convergence.

In step 1(a), the Hessian-vector product is carried out without explicitly forming the Hessian, forming only its action on a vector (CG is a matrix-free algorithm, and requires only the matrix action on a given vector). This can be done at the cost of solving a pair of linearized PDEs, one forward and one adjoint. Section 5 illustrates how this can done in the context of a specific non-linear, non-self-adjoint, time-dependent PDE. (In the time-dependent case, several additional PDE solves will be required in the memory-bound setting due to the necessity of checkpointing; this will be explained in Section 5). This avoids the need to store $\mathbf{H}_k$, which is (formally) dense and would require $O(n_m^2)$ storage. More important, it avoids the prohibitive cost of explicitly forming $\mathbf{H}_k$ column by column. To be successful, of course, few Hessian actions – *i.e.* few CG iterations – should be required. We will return to this shortly.

By terminating CG when negative curvature is encountered, step 1(b) ensures that a descent direction is maintained. This is because each CG iterate $\mathbf{p}_k^i$ is a linear combination of the negative gradient and the previous iterate $\mathbf{p}_k^{i-1}$, back to the first iterate, which itself is taken as the negative gradient. Thus the Newton-CG method is globally convergent (with appropriate line search). (The CG termination criteria can be easily extended to allow for a trust region globalization.)

Step 1(c) is intended to avoid 'over-solving'. Since Newton's method entails minimization of a quadratic approximation of Φ, and this quadratic approximation may be a poor predictor of the actual change in Φ for large steps far from a local minimizer, there is no point in solving $\mathbf{H}_k \mathbf{p}_k = -\mathbf{g}_k$ accurately early in the iterations. The linear system is solved more accurately as the iterations progress by choosing $\eta_k < 1$. Quadratic convergence can be preserved by taking η_k to be of the order of the norm of the gradient. On the other hand, a constant η_k results in a reduction to linear convergence. A more rapidly shrinking η_k will lead to faster Newton convergence, at the expense of increased CG iterations, while the converse is true with a more slowly shrinking η. A good compromise is often $\eta_k = O(\|\mathbf{g}_k\|^{1/2})$, which tends to result in the fewest overall CG iterations and thus Hessian actions.

As remarked above, it is crucial that CG take few iterations, since each iteration entails a pair of linearized forward/adjoint PDE solves (see Section 5). Can we

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

be assured that this is indeed the case? CG has the property that it minimizes the energy norm of the error at the ith iteration with respect to a polynomial of degree i. The energy norm of the error $\mathbf{e}^i$ can be written as (Shewchuk 1994, Nocedal and Wright 2006)

$$\|\mathbf{e}^i\|_{\mathbf{H}}^2 = \sum_{j=1}^{n_m} \lambda_j [P^i(\lambda_j)]^2 \xi_j^2, \tag{3.5}$$

where

- $\mathbf{e}^i = \mathbf{p}_k^* - \mathbf{p}_k^i$, with $\mathbf{p}_k^*$ as the exact solution to (3.4),
- ξ_j is the participation factor of the jth eigenvector $\mathbf{v}_j$ in the initial error, *i.e.* $\mathbf{e}^0 = \sum_{j=1}^{n_m} \mathbf{v}_j \xi_j$, and
- $P^i(\lambda)$ is an ith-order polynomial in λ, with the property that $P^0(\lambda) = 1$ and the constraint $P^i(0) = 1$.

Since CG minimizes the left-hand side of (3.5) with respect to P^i, it also minimizes the right-hand side with respect to P^i, that is,

$$P^{i^*}(\lambda) = \underset{P^i(\lambda)}{\arg\min} \sum_{j=1}^{n_m} \lambda_j [P^i(\lambda_j)]^2 \xi_j^2. \tag{3.6}$$

From this expression, we see that at the ith iteration, CG finds the ith degree polynomial that best fits the eigenvalues on the positive real axis, in a weighted least-squares sense. It minimizes the squared misfit at each eigenvalue, $[P^i(\lambda_j)]^2$, weighted by the magnitude of the eigenvalue λ_j, and by the squared component of the initial error in the direction of the corresponding eigenvalue, ξ_j^2. We see that CG favours the elimination of error components associated with eigenvectors that (i) correspond to large eigenvalues, (ii) correspond to clustered eigenvalues (since $P^i(\lambda)$ is small in their vicinity), and (iii) are most aligned with the initial error.

As illustrated in the examples of Section 2, the spectrum of a typical data misfit Hessian $\mathbf{H}^{\text{data}}$ collapses, with smaller eigenvalues corresponding to more oscillatory eigenvectors. What happens when CG is applied to such an operator? Initially it eliminates error components associated with the large eigenvalues (thus behaving as a smoother), and then begins to attack the smaller eigenvalues that accumulate at zero. In fact, as an alternative to Tikhonov regularization, early termination of CG iterations can be used as a regularization (Hanke 1995): initially, error components of the solution in smooth (data-informed) eigenfunction directions are eliminated, and the CG iteration is then terminated (*e.g.* by the Morozov criterion (3.2)) as noise-amplified unstable modes begin to be picked up.

However, here we are interested in the use of CG for solving the Newton system with the regularized Hessian, $\mathbf{H} := \mathbf{H}^{\text{data}} + \beta \mathbf{H}^{\text{reg}}$, since often we want to incorporate additional prior information on $\mathbf{m}$ via the regularization term. Toward this end, we

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

recognize that the regularization operator $\mathbf{H}^{\text{reg}}$ is typically an elliptic differential operator (possibly heterogeneous and anisotropic). This suggests preconditioning by $\mathbf{H}^{\text{reg}}$, so that the preconditioned Hessian operator is $\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{H}_k$, and the Newton system becomes

$$\big(\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{H}_k^{\text{data}} + \beta\mathbf{I}\big)\mathbf{p}_k = -\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{g}_k. \tag{3.7}$$

In the common case where $\mathbf{H}^{\text{reg}}$ is an elliptic differential operator, its inverse is compact, and thus the product of its inverse with $\mathbf{H}^{\text{data}}$, the *regularization-preconditioned data misfit Hessian*, is also compact. Then the operator

$$(\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{H}_k^{\text{data}} + \beta\mathbf{I})$$

has the form of a (discretized) Fredholm second-kind integral operator. For such problems, the conjugate gradient method (as well as other Krylov methods) is known to converge superlinearly; moreover, for n_m sufficiently large, the number of iterations required to solve (3.7) to a given accuracy is constant, independent of the parameter dimension n_m (Fortuna 1979, Flores 1993, Campbell, Ipsen, Kelley and Meyer 1994, Atkinson 1997, Axelsson and Karatson 2007, Herzog and Sachs 2015).

This mesh-independence of the (inner) CG iterations when using regularization preconditioning can be understood to be a consequence of the effectively finite-dimensional and mesh-independent range space of both the Hessian of $\mathcal{F}^*\mathcal{F}$ as well as the inverse regularization operator. CG takes $O(r)$ iterations to eliminate error components associated with the r dominant eigenvalues (*i.e.* those for which $\lambda_i > \beta$) of $\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{H}_k^{\text{data}}$, and then very quickly dispenses with the remaining error components associated with eigenvalues that cluster around β. Since the dominant cost of the CG iterations is the PDE solves associated with the Hessian action at each iteration, fully solving (3.7) requires about r forward/adjoint PDE solves, where usually $r \ll n_m$. Moreover, the inexactness of the method (*i.e.* early termination of CG when far from a local minimum) means that CG converges in just a handful of iterations in the early Newton iterations.

This combination of mesh-independence of (outer) Newton iterations, mesh-independence of (inner) CG iterations and inexactness to prevent oversolving results in a method for solving PDE-based inverse problems that requires a parameter-dimension-independent number of PDE solves. Thus the method is *scalable*. When the eigenvalues of the regularization-preconditioned data misfit Hessian decay sufficiently rapidly, r is small, and the method is then also *efficient*, in the sense that the required number of PDE solves is small. To understand how rapidly the eigenvalues decay, consider the model problems of Section 2. For the parabolic inverse problem, the exponential decay of the eigenvalues of the parameter-to-observable map $\mathcal{F}$ renders r very small, on the order of a handful of modes. For the elliptic inverse problem, the eigenvalues of $\mathcal{F}$ decay only algebraically, $O(j^{-2})$ for the specific problem considered. However, the spectrum of the data misfit Hessian

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

squares this to $O(j^{-4})$. Moreover, the common choice of H^1 regularization leads to preconditioning by $-\Delta^{-1}$, which in this case has the same eigenfunctions as $\mathcal{F}^*\mathcal{F}$, and eigenvalues that decay (in one dimension) like j^{-2}. Thus, overall, the eigenvalues of $\mathbf{H}_k^{\text{reg}^{-1}}\mathbf{H}_k^{\text{data}}$ decay like j^{-6}, which is very rapid decay indeed. Thus, while this operator's effective range space dimension r for elliptic inverse problems is typically large relative to those of parabolic problems, one often still has sufficiently rapid spectral decay that the average number of CG iterations per Newton iteration is in the 10s–100s, even for inverse problems with millions of parameters. Combined with a typically modest number of Newton iterations, this often means solution of large-scale inverse problems at a cost of 100s–1000s of linearized PDE solves. Section 6 presents a case study that illustrates this behaviour for inexact Newton-CG solution of a large-scale regularized inverse problem for the flow of the Antarctic ice sheet with $O(10^6)$ parameters.

To conclude this section, we note an important class of inverse problems for which r, the number of dominant eigenvalues of the regularization-preconditioned data misfit Hessian, is large in absolute terms, even if it is small or moderate relative to the parameter dimension n_m. In such problems, the required number of PDE solves using regularization preconditioning may be unacceptably large. This can result from a low data noise level, which reduces the regularization parameter β, allowing us to extract more information from the data (up to a point of course: ultimately instability takes over). By exposing additional smaller eigenvalues above the β threshold, CG is forced to iterate longer to resolve the associated modes. Another source of large r is more informative data, which results in a spectrum of $\mathbf{H}^{\text{data}}$ that decays more slowly. As suggested by the hyperbolic inverse problem of Section 2, this can stem from inverse problems in which the loss of information in the parameter-to-observable map is more muted. Examples include flows that are increasingly advection-dominated, wave propagation with higher frequencies and more sources, and more generally problems with large numbers of informative observations and/or informative experiments. For these and other problems, finding effective Hessian preconditioners beyond the regularization operator remains an active area of research; see Battermann and Heinkenschloss (1998), Arian and Ta'asan (1999), Gunzburger, Heinkenschloss and Lee (2000), Dreyer, Maar and Schulz (2000), Haber and Ascher (2001), Battermann and Sachs (2001), Akçelik *et al.* (2002), Borzì (2003), Ascher and Haber (2003), Akçelik *et al.* (2005), Borzì and Griesse (2005), Heinkenschloss (2005), Biros and Ghattas (2005*a*), Akçelik *et al.* (2006), Heinkenschloss and Nguyen (2006), Heinkenschloss and Herty (2007), Schöberl and Zulehner (2007), Drăgănescu and Dupont (2008), Herrmann, Moghaddam and Stolk (2008), Biros and Doğan (2008), Adavani and Biros (2008), Borzì and Schulz (2009), Adavani and Biros (2010), Herzog and Sachs (2010), Rees, Dollar and Wathen (2010*a*), Nielsen and Mardal (2010), Rees, Stoll and Wathen (2010*b*), Benzi, Haber and Taralli (2011), Takacs and Zulehner (2011), Rees and Wathen (2011), Chaillat and Biros (2012), Pearson, Stoll and

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Wathen (2012), Pearson and Wathen (2012), Borzì and Schulz (2012), Demanet *et al.* (2012), Nielsen and Mardal (2012), Pearson, Stoll and Wathen (2014), Schiela and Ulbrich (2014), Benner, Onwunta and Stoll (2016), Barker, Rees and Stoll (2016), Gholami, Mang and Biros (2016), Zhu *et al.* (2016*a*), Alger, Villa, Bui-Thanh and Ghattas (2017), Mardal, Nielsen and Nordaas (2017), Ambartsumyan *et al.* (2020) and Alger *et al.* (2019).

4. Bayesian framework and Laplace approximation

The regularization framework of Section 3 – while admitting scalable and efficient inverse solvers – provides just a point estimate of the inverse solution, and is not capable of quantifying the uncertainty in the inferred parameters. For a well-posed problem, this may be acceptable. But for an ill-posed problem, particularly one with non-negligible noise, where many parameter choices are consistent with the data to within the noise, this point estimate is not very useful. The Bayesian framework, on the other hand, states the inverse problem as one of statistical inference over the space of uncertain parameters (Tarantola 2005, Kaipio and Somersalo 2005, Stuart 2010, Dashti and Stuart 2017). The resulting solution to this statistical inverse problem is a *posterior probability distribution* that describes the probability of the parameter conditioned on the data. It combines a *prior distribution*, which encodes any prior knowledge or assumptions about the parameters before data are collected, with a *likelihood*, which explicitly represents the probability that a given set of parameters might give rise to the observed data. The central task of Bayesian solution then is to characterize or explore the posterior: drawing samples, estimating the mean, covariance or higher moments, determining credible intervals, or evaluating the posterior probabilities of particular events.

Unfortunately, Bayesian solution of inverse problems – *i.e.* fully characterizing the posterior – is often intractable for expensive PDE forward models and in high parameter dimensions, as results from discretization of infinite-dimensional parameter fields. This stems from the need to sample a posterior distribution in high dimensions, where each sample requires (at least) evaluation of the posterior, and thus solution of the forward problem. Markov chain Monte Carlo (MCMC) methods are usually the method of choice for sampling complex distributions, but in their black-box forms will often require many millions of evaluations of the posterior to produce reliable statistical estimates, making their use prohibitive for PDE-based inverse problems. In recent years, a number of methods for Bayesian inverse problems governed by PDEs have emerged that exploit structure of parameter space including geometry (locally approximated by the Hessian and higher derivatives) and intrinsic low-dimensionality (dictated by the compactness of the data-misfit Hessian), mirroring the structure-exploiting properties of inexact Newton-CG methods (Section 3) for deterministic inverse problems. These range from Hessian-aware Gaussian process approximation of the parameter-to-observable map (Bui-Thanh, Ghattas and Higdon 2012*b*), to projection-type forward model reduction (Galbally,

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Fidkowski, Willcox and Ghattas 2010, Lieberman, Willcox and Ghattas 2010, Cui, Marzouk and Willcox 2015), to polynomial chaos approximations of the stochastic forward problem (Badri Narayanan and Zabaras 2004, Ghanem and Doostan 2006, Marzouk and Najm 2009), to Markov chain Monte Carlo (MCMC) proposals exploiting log-likelihood Hessian approximations (Girolami and Calderhead 2011, Flath *et al.* 2011, Martin *et al.* 2012, Bui-Thanh *et al.* 2012*a*, 2013, Bui-Thanh and Girolami 2014, Petra *et al.* 2014, Cui *et al.* 2014, Cui, Law and Marzouk 2016, Beskos *et al.* 2017), to randomize-then-optimize sampling methods (Oliver, He and Reynolds 1996, Bardsley, Solonen, Haario and Laine 2014, Wang *et al.* 2017, Oliver 2017, Wang, Bui-Thanh and Ghattas 2018, Bardsley, Cui, Marzouk and Wang 2020, Ba, de Wiljes, Oliver and Reich 2021), to adaptive sparse quadrature combined with reduced basis approximations (Schillings and Schwab 2013, Chen and Schwab 2015, 2016*a*,*b*, Chen *et al.* 2017), to optimal transport-based variational methods using parametric and non-parametric transport maps (Moselhy and Marzouk 2012, Liu and Wang 2016, Marzouk, Moselhy, Parno and Spantini 2016, Detommaso *et al.* 2018, Chen *et al.* 2019*b*, Chen and Ghattas 2021, 2020*a*).

These methods have shown considerable promise by tackling certain PDE-governed Bayesian inverse problems with exploitable structure. Further developments in such structure-exploiting methods – in conjunction with the model reduction methods described in Part 3 – offer hope that full Bayesian solution of complex PDE-governed high-dimensional inverse problems will become more common over the coming decade. These methods are undergoing rapid development today and we will not discuss them further. However, we will describe in detail the *Laplace approximation* of the posterior, which becomes tractable to compute – and even scalable and efficient – when one exploits the same properties that made inexact Newton-CG so powerful for the regularization formulation. Thus the methods of this section are intimately connected to those of the previous one and the properties of the model inverse problems highlighted in Section 2. Moreover, the Laplace approximation and related tools often play an important role in the sampling, variational and transport methods mentioned above.

The discussion below is based on the finite-dimensional form of the Bayesian inverse problem, *i.e.* after parameter space has been discretized. Conditions on the prior and the parameter-to-observable map that result in a well-defined and well-posed Bayesian inverse problem in infinite dimensions are presented in Stuart (2010) and Dashti and Stuart (2017), and summarized in Arridge *et al.* (2019). Infinite-dimension-consistent discretizations are discussed in Bui-Thanh *et al.* (2013) and Petra *et al.* (2014).

4.1. *Bayesian formulation*

Denote by $\pi_{\text{pr}}(\mathbf{m})$ the prior probability density of the model parameters $\mathbf{m} \in \mathbb{R}^{n_m}$, $\pi_{\text{like}}(\mathbf{d}|\mathbf{m})$ the likelihood of the data $\mathbf{d} \in \mathbb{R}^{n_d}$ given the parameters, and $\pi_{\text{post}}(\mathbf{m}) := \pi(\mathbf{m}|\mathbf{d})$ the posterior density reflecting the probability of the parameters conditioned

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

on the data. Then Bayes' rule can be written as

$$\pi_{\text{post}}(\mathbf{m}) = \frac{1}{Z}\,\pi_{\text{like}}(\mathbf{d}|\mathbf{m})\,\pi_{\text{pr}}(\mathbf{m}), \tag{4.1}$$

where the normalizing constant Z is given by

$$Z = \int \pi_{\text{like}}(\mathbf{d}|\mathbf{m})\,\pi_{\text{pr}}(\mathbf{m})\,\mathrm{d}\mathbf{m},$$

and is often called the *evidence*.

We restrict our discussion to a Gaussian prior $\mathbf{m} \sim \mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{\Gamma}_{\text{pr}})$, that is,

$$\pi_{\text{pr}}(\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{pr}}\|^2_{\mathbf{\Gamma}_{\text{pr}}^{-1}}\right\}, \tag{4.2}$$

where $\mathbf{m}_{\text{pr}} \in \mathbb{R}^{n_m}$ and $\mathbf{\Gamma}_{\text{pr}} \in \mathbb{R}^{n_m \times n_m}$ are respectively the mean and covariance of the prior distribution. Here, $\mathbf{\Gamma}_{\text{pr}}$ must be chosen to be sufficiently smoothing in order to ensure a well-posed Bayesian inverse problem; this is the Bayesian equivalent of regularization. One attractive choice is for $\mathbf{\Gamma}_{\text{pr}}$ to be taken as the discretization of the inverse of the νth power of an elliptic differential operator:

$$\mathbf{\Gamma}_{\text{pr}} = (-\gamma\Delta_h + \delta I_h)^{-\nu}, \tag{4.3}$$

where $\gamma > 0$ and $\delta > 0$ are positive constants and h indicates discretization. For well-posedness, $\nu > \omega/2$, where ω is the spatial dimension (Stuart 2010). So in one dimension $\omega = 1$ is sufficient, whereas in two and three dimensions we often take $\omega = 2$ to avoid solving a fractional PDE. The parameters γ and δ are chosen to impose prior knowledge of the correlation length ρ (distance for which the two-point correlation coefficient is 0.1) and the pointwise variance σ^2 of the prior operator; specifically $\rho \propto \sqrt{\gamma/\delta}$ and $\sigma^2 \propto \delta^{-\nu}\rho^{-\omega}$. This form of prior is equivalent to a subset of the Matérn covariance family (Lindgren, Rue and Lindström 2011). To incorporate anisotropy in correlation lengths, the Laplacian in (4.3) can be replaced by an anisotropic Poisson operator. These choices ensure that the prior covariance $(-\gamma\Delta + \delta I)^{-\nu}$ is a trace-class operator, leading to bounded pointwise variance and a well-posed infinite-dimensional Bayesian inverse problem (Stuart 2010). Appropriate boundary conditions can be chosen to mitigate the influence of the boundary (Daon and Stadler 2018).

To construct the likelihood, we consider the discretized form of the additive noise model (2.3),

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_{\text{noise}}),$$

with the noise $\boldsymbol{\eta} \in \mathbb{R}^{n_d}$ taken to be a normally distributed random variable with mean zero and covariance $\mathbf{\Gamma}_{\text{noise}} \in \mathbb{R}^{n_d \times n_d}$, and as before $\mathbf{f}(\mathbf{m})$ is the discretized parameter-to-observable map. The likelihood then becomes

$$\pi_{\text{like}}(\mathbf{d}|\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{f}(\mathbf{m})-\mathbf{d}\|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}}\right\}.$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

(See Dunlop 2019 for extensions to other noise models.)

With this choice of Gaussian prior and Gaussian additive noise, we arrive at the expression for the posterior,

$$\pi_{\text{post}}(\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{f}(\mathbf{m})-\mathbf{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{pr}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{pr}}}\right\}. \tag{4.4}$$

After discretization, the posterior (4.4) is a probability density in n_m dimensions, which can be in the millions (or much higher) for complex PDE models, such as the example in Section 6. Moreover, evaluating $\pi_{\text{post}}(\mathbf{m})$ at any point in parameter space requires solution of the forward PDE problem embedded within the parameter-to-observable map $\mathbf{f}(\mathbf{m})$. As such, exploring the posterior to, for example, compute statistics of quantities of interest (mean, variance, quantiles) is typically prohibitive, due to the high-dimensionality of the parameters and the expense of solving the forward PDE problems. We will restrict ourselves here to the Laplace approximation, which is a Gaussian centred on the *maximum a posteriori* (MAP) point, with covariance equal to the inverse of the Hessian of the negative logarithm of the posterior evaluated at the MAP point. The Laplace approximation is exact when the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ is linear, and often provides a reasonable approximation to the posterior, since the error is of the order of the departure from linearity of the parameter-to-observable map (Helin and Kretschmann 2020). Moreover, in the limit of small noise or large data, the Laplace approximation converges to the posterior (Schillings, Sprungk and Wacker 2020).

4.2. *Finding the MAP point*

The maximum *a posteriori* point $\mathbf{m}_{\text{map}}$ is the point that maximizes the posterior density with respect to $\mathbf{m}$,

$$\begin{aligned}\mathbf{m}_{\text{map}} &:= \underset{\mathbf{m}\in\mathbb{R}^{n_m}}{\arg\max}\ \frac{1}{Z}\pi_{\text{post}}(\mathbf{m}) \\ &= \underset{\mathbf{m}\in\mathbb{R}^{n_m}}{\arg\min}\ -\log\pi_{\text{post}}(\mathbf{m}) \\ &= \underset{\mathbf{m}\in\mathbb{R}^{n_m}}{\arg\min}\ \underbrace{\frac{1}{2}\|\mathbf{f}(\mathbf{m})-\mathbf{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{pr}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{pr}}}}_{\Phi(\mathbf{m})}. \end{aligned} \tag{4.5}$$

Notice the equivalence between finding the MAP point and solving the regularized inverse problem (3.3) by minimizing $\Phi(\mathbf{m})$. The prior term is analogous to a regularization (with regularization operator taken as $\boldsymbol{\Gamma}^{-1}_{\text{pr}}$), and the data misfit is weighted by the noise covariance $\boldsymbol{\Gamma}^{-1}_{\text{noise}}$, but otherwise the two are equivalent. This means the MAP point can be found efficiently and scalably using the machinery introduced for the regularized inverse problem in Section 3: inexact Newton-CG with early termination, regularization preconditioning, and adjoint-based gradient and matrix-free Hessian actions.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

4.3. The Laplace approximation

The MAP point, while useful, is still a point estimate and does not provide us with any estimate of uncertainty in the inverse solution. We define the Laplace approximation $\pi_{\text{post}}^{\mathcal{L}}(\mathbf{m})$ of the posterior as

$$\pi_{\text{post}}(\mathbf{m}) \simeq \pi_{\text{post}}^{\mathcal{L}}(\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{map}}\|^2_{\mathbf{H}(\mathbf{m}_{\text{map}})}\right\}.$$

The Laplace approximation is a Gaussian, centred at the MAP point $\mathbf{m}_{\text{map}}$, with covariance equal to the inverse of the Hessian of the negative log posterior, $-\log \pi_{\text{post}}(\mathbf{m})$, *i.e.* the Hessian of $\Phi(\mathbf{m})$. This can be seen by expanding $\Phi(\mathbf{m})$ in a Taylor series to second order, around the MAP point:

$$\begin{aligned}\Phi(\mathbf{m}) \simeq \Phi^{\mathcal{Q}}(\mathbf{m}) &:= \Phi(\mathbf{m}_{\text{map}}) + \mathbf{g}(\mathbf{m}_{\text{map}})^{\mathsf{T}}(\mathbf{m}-\mathbf{m}_{\text{map}}) \\ &\quad + \frac{1}{2}(\mathbf{m}-\mathbf{m}_{\text{map}})^{\mathsf{T}}\mathbf{H}(\mathbf{m}_{\text{map}})(\mathbf{m}-\mathbf{m}_{\text{map}}),\end{aligned}$$

where as before $\mathbf{g}(\mathbf{m})$ is the gradient of $\Phi(\mathbf{m})$ with respect to $\mathbf{m}$, and $\mathbf{H}(\mathbf{m})$ is its Hessian. Since $\mathbf{g}(\mathbf{m}_{\text{map}}) = \mathbf{0}$ (*i.e.* $\mathbf{m}_{\text{map}}$ minimizes $\Phi(\mathbf{m})$), and defining $\mathbf{H}_{\text{map}} := \mathbf{H}(\mathbf{m}_{\text{map}})$, we obtain

$$\Phi^{\mathcal{Q}}(\mathbf{m}) = \Phi(\mathbf{m}_{\text{map}}) + \frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{map}}\|^2_{\mathbf{H}_{\text{map}}}.$$

With this approximation,

$$\begin{aligned}\pi_{\text{post}}^{\mathcal{L}}(\mathbf{m}) &\propto \exp\{-\Phi^{\mathcal{Q}}(\mathbf{m})\} \\ &\propto \underbrace{\exp\{-\Phi(\mathbf{m}_{\text{map}})\}}_{\substack{\text{constant that}\\ \text{can be absorbed}\\ \text{into normalization}}} \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{map}}\|^2_{\mathbf{H}_{\text{map}}}\right\}, \\ &\propto \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{map}}\|^2_{\mathbf{H}_{\text{map}}}\right\}.\end{aligned}$$

Thus we obtain a Gaussian $\mathcal{N}(\mathbf{m}_{\text{map}}, \mathbf{H}_{\text{map}}^{-1})$. Since this is a Gaussian, we can compute its normalizing constant to give

$$\pi_{\text{post}}^{\mathcal{L}}(\mathbf{m}) := \frac{\det(\mathbf{H}_{\text{map}}^{1/2})}{(2\pi)^{n_m/2}} \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{map}}\|^2_{\mathbf{H}_{\text{map}}}\right\}. \tag{4.6}$$

Here n_m is the dimension of $\mathbf{m}$. The covariance $\mathbf{H}_{\text{map}}^{-1}$ should be positive definite, which is assured since the Hessian is evaluated at $\mathbf{m}_{\text{map}}$ (making it at least positive semidefinite), and assuming the prior covariance has been chosen appropriately to annihilate zero eigenvalues of $\mathbf{H}$, *i.e.* to provide prior information (or regularization) in directions in parameter space that are not informed by the data.

As stated above, when the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ is linear – *i.e.* we have a linear inverse problem – then $\Phi(\mathbf{m})$ is quadratic, and so the quadratic

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Taylor approximation is exact. Thus the Laplace approximation is exact for a linear inverse problem. For non-linear but unimodal posteriors, it may still provide useful information, since $-\log \pi_{\text{post}}^{\mathcal{L}}$ and $-\log \pi_{\text{post}}$ agree to two derivatives at $\mathbf{m}_{\text{map}}$. Finally, the Laplace approximation is accurate not only in poorly data-informed directions (since the posterior reverts to the Gaussian prior in those directions) but also in highly data-informed directions, since the small posterior variance in those directions implies that linearization of the parameter-to-observable map over the support of the posterior is viable.

The expression above for $\pi_{\text{post}}^{\mathcal{L}}(\mathbf{m})$ is easy to write, but it becomes clear that there are significant challenges to computing it. In particular, we are interested in the following operations on the Laplace approximation of the posterior.

1 Compute posterior covariance $\mathbf{\Gamma}_{\text{post}}(\mathbf{m}) := \mathbf{H}_{\text{map}}^{-1}$.

2 Compute normalizing constant, requiring $\det(\mathbf{H}_{\text{map}}^{1/2})$.

3 Sample from the posterior, *i.e.* compute $\mathbf{H}_{\text{map}}^{-1/2}\mathbf{x}, \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

4 Compute pointwise variance field, $\operatorname{diag}(\mathbf{H}_{\text{map}}^{-1})$.

Recall that we cannot even construct the Hessian $\mathbf{H}$ (which would require as many linearized forward model solves as we have parameters, *i.e.* millions). So how are we going to compute $\mathbf{H}_{\text{map}}^{-1}$, $\operatorname{diag}(\mathbf{H}_{\text{map}}^{-1})$, $\mathbf{H}_{\text{map}}^{-1/2}$, and $\det(\mathbf{H}_{\text{map}}^{1/2})$?

The key is to recognize that $\mathbf{H}$ has special structure – the same structure that allows conjugate gradients to converge in a small and dimension-independent number of iterations. Namely, the Hessian is composed of a compact operator (with eigenvalues that collapse to zero) and a differential operator (when $\mathbf{\Gamma}_{\text{pr}}$ is taken as the inverse of an elliptic differential operator):

$$\mathbf{H}_{\text{map}} = \mathbf{H}_{\text{map}}^{\text{data}} + \mathbf{\Gamma}_{\text{pr}}^{-1},$$

where as before

$$\mathbf{H}^{\text{data}}(\mathbf{m}) := \frac{D^2}{D\mathbf{m}^2}\,\frac{1}{2}\|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}}$$

is the Hessian of the data misfit. We have seen that for ill-posed inverse problems, $\mathbf{H}^{\text{data}}$ often has eigenvalues that decay rapidly to zero. In this case we can make a low-rank approximation of $\mathbf{H}_{\text{map}}^{\text{data}}$ by solving the generalized eigenvalue problem

$$\mathbf{H}_{\text{map}}^{\text{data}}\mathbf{v}_j = \lambda_j \mathbf{\Gamma}_{\text{pr}}^{-1}\mathbf{v}_j \tag{4.7}$$

for the first r eigenvalues and eigenvectors, $r \ll N$. Define

$$\begin{aligned}\mathbf{V}_r \in \mathbb{R}^{n_m \times r} &:= [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r],\\ \mathbf{\Lambda}_r \in \mathbb{R}^{r \times r} &:= \operatorname{diag}(\lambda_j),\end{aligned}$$

where $\mathbf{V}_r$ is $\mathbf{\Gamma}_{\text{pr}}^{-1}$-orthogonal, *i.e.* $\mathbf{V}_r^{\top}\mathbf{\Gamma}_{\text{pr}}^{-1}\mathbf{V}_r = \mathbf{I}_r$. With this truncated spectral decomposition, and using the Sherman–Morrison–Woodbury formula, it can be

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

shown that (Isaac *et al.* 2015)

$$\begin{aligned}\mathbf{\Gamma}^{\mathcal{L}}_{\text{post}} &:= \mathbf{H}^{-1}_{\text{map}} \\ &= (\mathbf{H}^{\text{data}}_{\text{map}} + \mathbf{\Gamma}^{-1}_{\text{pr}})^{-1} \\ &= \mathbf{\Gamma}_{\text{pr}} - \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^{\mathsf{T}} + O\Bigg(\sum_{j=r+1}^{n_m} \frac{\lambda_j}{1+\lambda_j}\Bigg),\end{aligned}$$

where $\mathbf{D}_r = \operatorname{diag}(\lambda_j/\lambda_{j+1})$. The error term, which depends on the discarded eigenvalues λ_j, $j = r+1, \dots, n_m$, suggests that when the eigenvalues decay rapidly (as is typical of ill-posed inverse problems), truncating the spectrum when λ_j is small relative to 1 incurs small and controllable error. Thus we can approximate (with arbitrary accuracy) the posterior covariance of the Laplace approximation by

$$\mathbf{\Gamma}^{\mathcal{L}}_{\text{post}} := \mathbf{H}^{-1}_{\text{map}} \simeq \mathbf{\Gamma}_{\text{pr}} - \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^{\mathsf{T}}. \tag{4.8}$$

The posterior covariance can be compactly represented by a low-rank perturbation (the data-informed modes) of the inverse of an elliptic operator (the prior). The availability of the low-rank decomposition $\mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^{\mathsf{T}}$ permits us to compute or compactly represent $\mathbf{H}^{-1}_{\text{map}}$, $\operatorname{diag}(\mathbf{H}^{-1}_{\text{map}})$, $\mathbf{H}^{-1/2}_{\text{map}}$ and $\det(\mathbf{H}^{1/2}_{\text{map}})$. See Villa *et al.* (2021), which describes algorithms provided by the hIPPYlib software framework to compute these and other quantities relevant to exploring the Laplace approximation posterior for PDEs.

All of this assumes we can solve the generalized eigenvalue problem (4.7) for the dominant eigenpairs. How can this be done when we cannot even compute $\mathbf{H}^{\text{data}}_{\text{map}}$? The answer is a randomized eigensolver (see *e.g.* Halko, Martinsson and Tropp 2011, Villa *et al.* 2021). This family of algorithms has the following properties.

(i) The matrix $\mathbf{H}^{\text{data}}$ is not needed explicitly; only its action in a random direction is needed. This requires solution of a linearized forward problem and a linearized adjoint problem, *i.e.* two linearized PDE solves per Hessian action. This will be illustrated in Section 5.

(ii) The prior inverse $\mathbf{\Gamma}^{-1}_{\text{pr}}$ is not needed; only the ability to solve systems involving $\mathbf{\Gamma}^{-1}_{\text{pr}}$ is required, and this is efficiently done with the choice of prior covariance (4.3), since this involves solving elliptic PDEs with operators given by powers of $-\gamma\Delta + \delta I$, which can be done in $O(n_m)$ work using a multigrid method (at least when the power is an even integer).

(iii) The number of necessary products of $\mathbf{H}^{\text{data}}_{\text{map}}$ with a vector is just $O(r)$ (in fact it is r plus a few additional oversampling Hessian-vector products to control error). Recall that in the usual case of ill-posed inverse problems, $r \ll N$.

(iv) As illustrated in Section 2, for many ill-posed inverse problems, the dominant eigenvalues (λ_i, $i = 1, \dots, r$) correspond to smooth eigenfunctions (*i.e.* the data are informative about smooth components of the parameter field). Thus, once these eigenfunctions are well-resolved on a given grid (which is usually coarse, since they are smooth), further grid refinement does not affect them. Thus the effective rank r is independent of mesh size. So we can construct an accurate approximation of the posterior covariance (of the Laplace approximation) at a cost, measured in the number of (linearized) forward/adjoint PDE solves (r), that is independent of the parameter dimension, as well as the data dimension, and depends only on the information dimension.

Thus, once the MAP point has been found, we can compactly represent the posterior covariance via (4.8), sample from the posterior distribution, and compute the posterior variance field, all at a cost of r pairs of linearized forward/adjoint PDEs and r prior elliptic PDE solves, where r is typically small and dimension-independent. There are several randomized generalized eigensolver variants depending on desired accuracy and work; see Villa *et al.* (2021).

4.4. *Optimal experimental design*

One of the attractions of the Bayesian framework for inverse problems is the ability to pose the following meta-question: How do we optimally acquire the data in the first place? In other words, what, where, when and under which conditions should we observe in order to minimize the uncertainty in the inferred parameters, or in some predictive goal depending on the parameters? This is the optimal experimental design (OED) problem. The Bayesian solution – in particular the Laplace approximation – allows us to pose the OED problem as a minimization problem involving the posterior covariance, reflecting uncertainty in the parameters.

Given experimental design variables $\mathbf{z} \in \mathbb{R}^{n_z}$ (*e.g.* locations of sensors), we can pose the OED problem as minimizing with respect to $\mathbf{z}$ some invariant of the posterior covariance $\mathbf{\Gamma}_{\text{post}}$, which is approximated by the inverse of the Hessian at the MAP point. Two popular choices are minimizing the trace of the inverse Hessian, *i.e. A-optimal design*,

$$\min_{\mathbf{z}\in\mathbb{R}^{n_z}} \operatorname{tr}(\mathbf{H}(\mathbf{m}_{\text{map}}(\mathbf{z}), \mathbf{z})^{-1}), \tag{4.9}$$

and minimizing its determinant, *i.e. D-optimal design*,

$$\min_{\mathbf{z}\in\mathbb{R}^{n_z}} \det(\mathbf{H}(\mathbf{m}_{\text{map}}(\mathbf{z}), \mathbf{z})^{-1}). \tag{4.10}$$

For linear inverse problems, the D-optimal design is equivalent to maximizing the *expected information gain* (EIG) from the data – that is, the expectation over the data of the Kullback–Leibler divergence of prior to posterior. The availability of the low-rank-based approximation of the posterior covariance (4.8) facilitates not only the direct computation of the trace in (4.9) (Alexanderian *et al.* 2016),

$$\operatorname{tr}(\mathbf{H}_{\text{map}}^{-1}) \simeq \sum_{j=1}^{r} \frac{\lambda_j}{1+\lambda_j},$$

but also the computation of the EIG for linear inverse problems (Alexanderian,

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Gloor and Ghattas 2015),

$$\mathrm{EIG} := \mathbb{E}_{\mathbf{d}}[D_{\mathrm{KL}}(\pi_{\mathrm{post}}(\mathbf{m}|\mathbf{d})\|\pi_{\mathrm{pr}}(\mathbf{m}))] \simeq \sum_{j=1}^{r} \log(1+\lambda_j).$$

Here the λ_j are the dominant eigenvalues of the generalized eigenvalue problem (4.7). For non-linear inverse problems under the Laplace approximation, similar easily computable approximations of the EIG are available once (4.7) has been solved (Wu *et al.* 2020).

Note from (4.9) and (4.10) that the Hessian's dependence on the experimental design variables $\mathbf{z}$ is complicated. For example, if $\mathbf{z}$ represents the sensor locations, then the observation operator and hence the Hessian depend on $\mathbf{z}$ explicitly. But the MAP point $\mathbf{m}_{\mathrm{map}}$ also depends on the sensor locations, so the Hessian depends on $\mathbf{z}$ implicitly via solution of the optimization problem (4.5). Thus (4.9) and (4.10) are bilevel optimization problems, where the inner optimization problem finds the MAP point and expresses the eigenvalue problem (4.7) at this point, and the outer optimization problem maximizes the EIG or minimizes the trace or determinant of the posterior covariance.

Moreover, unless the inverse problem is linear, the Hessian depends on the actual data $\mathbf{d}$, which are not yet available when designing the experiment or data acquisition strategy. This problem is typically addressed by synthesizing predicted data from a prior parameter model (which could stem from an existing observation network in a sequential design strategy). The resulting optimization problem is fantastically difficult to solve. It is bad enough that traces or determinants of inverses of Hessians are required at each optimization iteration; worse, efficient solution requires gradients of (4.9) and (4.10) with respect to $\mathbf{z}$, which means differentiating through these traces/determinants, Hessian inverses and inner inverse problems to find the MAP point. Because of these difficulties, OED for large-scale inverse problems governed by PDE forward models with large numbers of design variables remained out of reach. In the last several years, a number of advances have made the solution of such OED problems tractable, at least for simpler PDE systems (the OED problem is still several orders of magnitude more expensive than the inverse problem, which in turn is several orders of magnitude more expensive than the forward problem). The critical ingredients often end up being the tools described in this section and the previous one: regularization-preconditioned inexact Newton-CG with adjoint-based Hessian actions for finding the MAP point; Laplace approximation for characterizing the posterior along with a randomized eigensolver to reveal the dominant spectrum and induce a low-rank approximation, facilitating computation of traces or determinants and their derivatives.

While the resulting algorithms for OED can require a dimension-independent number of PDE solves, the absolute number can be very large – several orders of magnitude greater than required for the 'inner' inverse problem. Thus OED remains a frontier problem for large-scale inverse problems governed by PDEs.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Despite the computational challenges, the promise of designing optimal data acquisition systems and campaigns for complex inverse problems governed by PDEs in a rigorous and systematic way is motivating much current research (Huan and Marzouk 2013, 2014, Attia, Alexanderian and Saibaba 2018, Alexanderian and Saibaba 2018, Feng and Marzouk 2019, Koval, Alexanderian and Stadler 2020, Wu *et al.* 2020, Alexanderian 2020, Herman, Alexanderian and Saibaba 2020, Jagalur-Mohan and Marzouk 2020, Wu, Chen and Ghattas 2021, Alexanderian, Petra, Stadler and Sunseri 2021).

5. Computing the Hessian action

As we have seen, the linchpin of the Newton-CG and randomized eigensolver algorithms of the previous sections is the Hessian action. This is the operator that enables scalable and efficient solution of inverse problems in the regularized or Bayesian frameworks. Scalability implies that the number of forward/adjoint PDE solves remains independent of the parameter or data dimensions; efficiency means that this number is small relative to the parameter dimension n_d.

Unfortunately, explicit construction of the discretized Hessian requires n_d linearized PDE solves (one per column), which is intractable for anything other than the simplest problems. Fortunately the Newton-CG and randomized eigensolver algorithms are matrix-free: what is needed is not the Hessian itself, but its action in an arbitrary direction. Adjoint methods along with the Lagrangian formalism can be used to efficiently form the Hessian action, at the cost of just a pair of linearized forward/adjoint PDE solves. In this section we will illustrate the derivation of the Hessian action in the context of a specific problem, described in Section 5.1. The gradient – needed for the Newton algorithm as well as to find the MAP point for the Laplace approximation – is derived in Section 5.2. Finally, the Hessian action is derived in Section 5.3. Checkpointing and discretization issues are also discussed.

5.1. *An advection–diffusion–reaction inverse problem*

To illustrate the derivation of the gradient and the Hessian action, we consider an inverse problem governed by a time-dependent advection–diffusion–reaction (ADR) model with a simple cubic reaction term. This forward problem is chosen since it elaborates the roles of time-dependence, non-self-adjoint operators and non-linearity. To contrast the structure of the gradient and Hessian action for coefficient (parameter) estimation with those of state estimation, we will jointly infer the unknown diffusion coefficient field m_κ, and the unknown initial condition field m_0. Tikhonov H^1 regularization is employed for both. The inverse problem is stated as

$$\begin{aligned} &\min_{(m_\kappa, m_0)\in H^1(\Omega)} \phi(m_\kappa, m_0) \qquad (5.1)\\ &:= \frac{1}{2}\int_0^T \int_\Omega (\mathcal{B}u - d)^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \frac{\beta_\kappa}{2}\int_\Omega \nabla m_\kappa \cdot \nabla m_\kappa \,\mathrm{d}\boldsymbol{x} + \frac{\beta_0}{2}\int_\Omega \nabla m_0 \cdot \nabla m_0 \,\mathrm{d}\boldsymbol{x}, \end{aligned}$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

where the state $u(t, \boldsymbol{x})$ implicitly depends on the parameters (m_κ, m_0) through the solution of the *forward advection–diffusion–reaction initial-boundary* value problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{v} \cdot \nabla u - \nabla \cdot (m_\kappa \nabla u) + c u^3 &= f && \text{in } \Omega \times (0, T) && \text{(PDE)}, && (5.2)\\ u &= 0 && \text{on } \partial\Omega \times (0, T) && \text{(BC)}, && (5.3)\\ u|_{t=0} &= m_0 && \text{in } \Omega && \text{(IC)}, && (5.4) \end{aligned}$$

where (5.2) represents the ADR PDE, (5.3) the (homogeneous Dirichlet) boundary condition and (5.4) the initial condition. Here

- $\phi(m_\kappa, m_0)\colon H^1(\Omega) \times H^1(\Omega) \longrightarrow \mathbb{R}$ is the regularized data misfit functional,
- $\mathcal{B}\colon L^2((0, T); H^1_0(\Omega)) \longrightarrow \mathcal{D}$ is the space-time observation operator, which prescribes locations in space and instants in time at which observations are made,
- $d \in \mathcal{D}$ is the corresponding observed data at these locations/instants and $\mathcal{D}$ is the corresponding data space,
- $(0, T)$ is the observation time window,
- $u(t, \boldsymbol{x}) \in L^2((0, T); H^1_0(\Omega))$ is the state concentration field,
- $m_\kappa(\boldsymbol{x}) \in H^1(\Omega)$ is the diffusion coefficient inversion parameter field,
- $m_0(\boldsymbol{x}) \in H^1(\Omega)$ is the initial condition inversion parameter field,
- β_κ is the regularization parameter for m_κ,
- β_0 is the regularization parameter for m_0,
- $\boldsymbol{v}(t, \boldsymbol{x}) \in L^2((0, T); \boldsymbol{H}^1(\Omega))$ is the advection velocity field,
- $c(\boldsymbol{x}) \in L^2(\Omega)$ is the reaction coefficient,
- $f(t, \boldsymbol{x}) \in L^2((0, T); H^{-1}(\Omega))$ is the source term, and
- Ω is a bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, and $\partial\Omega$ is its boundary.

Here we have employed standard Sobolev spaces: $L^2(\Omega)$ is the space of square-integrable functions over Ω, $H^1(\Omega)$ is the space of functions whose derivatives belong to $L^2(\Omega)$, $\boldsymbol{H}^1(\Omega)$ is the space of vector-valued functions whose components belong to $H^1(\Omega)$, and $H^1_0(\Omega)$ is the space of functions in $H^1(\Omega)$ that vanish on $\partial\Omega$. We shall sometimes omit Ω if no ambiguity results. We shall remain agnostic to the particular form of the observation operator $\mathcal{B}$, which depends on the data acquisition system; typically it will involve local or path integrals, including mollified Dirac deltas, of the state or its flux, or more general functions of the state.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Let us define the spaces

$$\mathcal{U} = \{u \in L^2((0,T); H_0^1(\Omega))\},$$
$$\mathcal{P} = \{p \in L^2((0,T); H^1(\Omega))\}.$$

To derive the gradient via the Lagrangian, we require the space–time weak form of the forward ADR problem. Multiplying the residual of the PDE (5.2) by a test function $p(t, \boldsymbol{x}) \in \mathcal{P}$, integrating the diffusion term by parts, and weakly incorporating the initial condition (5.4) via another test function $q(\boldsymbol{x}) \in L^2(\Omega)$, we obtain the space–time weak form of the forward ADR problem: Find $u \in \mathcal{U}$ such that

$$\int_0^T \int_\Omega \left[p\frac{\partial u}{\partial t} + p\boldsymbol{v} \cdot \nabla u + m_\kappa \nabla u \cdot \nabla p + cpu^3 - pf \right] \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$
$$- \int_0^T \int_{\partial\Omega} p\, m_\kappa \nabla u \cdot \boldsymbol{n}\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t + \int_\Omega q\,[u(0,\boldsymbol{x}) - m_0]\, \mathrm{d}\boldsymbol{x} \quad \text{for all } p \in \mathcal{P}. \tag{5.5}$$

It is important to enforce the initial condition weakly, since we wish to 'expose' the initial condition inversion parameter m_0 to the Lagrangian (and hence the gradient and Hessian), rather than build it into the admissible space for u. In the derivation of the gradient and Hessian action, p will play the role of the adjoint state variable (*adjoint* from now on), and thus for now we do not assume it vanishes on the boundary and instead take it to be arbitrary on $\partial\Omega$. In the next section, the boundary condition for the adjoint ADR problem will emerge from stationarity of the Lagrangian. Notice that this weak form is different from the usual weak forms used in (spatial) finite element methods for time-dependent problems, in which the test function p is defined in space only.

5.2. *The gradient*

Here we derive expressions for the gradient of ϕ with respect to m_κ and m_0, using the formal Lagrange method. We begin by forming the Lagrangian functional $\mathcal{L}^g$ that combines the regularized data misfit $\phi(m_\kappa, m_0)$ in (5.1) with the weak form of the ADR problem (5.5),

$$\mathcal{L}^g(u, p, q, m_\kappa, m_0) := \frac{1}{2} \int_0^T \int_\Omega (\mathcal{B}u - d)^2\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$
$$+ \frac{\beta_\kappa}{2} \int_\Omega \nabla m_\kappa \cdot \nabla m_\kappa\, \mathrm{d}\boldsymbol{x} + \frac{\beta_0}{2} \int_\Omega \nabla m_0 \cdot \nabla m_0\, \mathrm{d}\boldsymbol{x}$$
$$+ \int_0^T \int_\Omega \left[p\frac{\partial u}{\partial t} + p\boldsymbol{v} \cdot \nabla u + m_\kappa \nabla u \cdot \nabla p + cpu^3 - pf \right] \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$
$$- \int_0^T \int_{\partial\Omega} p\, m_\kappa \nabla u \cdot \boldsymbol{n}\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t + \int_\Omega q\,[u(0,\boldsymbol{x}) - m_0]\, \mathrm{d}\boldsymbol{x} \tag{5.6}$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

for functions $(u, p, m_\kappa, m_0, q) \in \mathcal{U} \times \mathcal{P} \times H^1(\Omega) \times H^1(\Omega) \times L^2(\Omega)$. The superscript g in $\mathcal{L}^g$ indicates that this is the Lagrangian used to derive the gradient (a different Lagrangian will be employed for the Hessian in the next section).

To derive an expression for the gradient of ϕ with respect to (m_κ, m_0), we seek conditions that make the Lagrangian $\mathcal{L}^g$ stationary with respect to its arguments (u, p, m_κ, m_0, q). Taking variations of $\mathcal{L}^g$ with respect to $p \in \mathcal{P}$ and $q \in L^2(\Omega)$ and requiring them to vanish for all admissible variations simply recovers the weak forms of the forward ADR equation and its initial condition,

$$\begin{aligned} \delta_p \mathcal{L}^g &= 0 \quad \text{for all } \hat{p} \in \mathcal{P} &&\Longrightarrow \quad \text{weak form of PDE,} \\ \delta_q \mathcal{L}^g &= 0 \quad \text{for all } \hat{q} \in L^2(\Omega) &&\Longrightarrow \quad \text{weak form of IC,} \end{aligned}$$

where $\hat{p}$ is the variation of p, and $\hat{q}$ is the variation of q. Restating this *weak form of the forward ADR problem*, now in the context of the inverse problem: Given $(m_\kappa, m_0) \in H^1(\Omega) \times H^1(\Omega)$, find $(u, q) \in \mathcal{U} \times L^2(\Omega)$ such that

$$\begin{aligned} &\int_0^T \int_\Omega \left[\hat{p} \frac{\partial u}{\partial t} + \hat{p} \boldsymbol{v} \cdot \nabla u + m_\kappa \nabla u \cdot \nabla \hat{p} + c\hat{p}u^3 - \hat{p} f \right] \mathrm{d}\boldsymbol{x}\, \mathrm{d}t \qquad (5.7) \\ &\quad - \int_0^T \int_{\partial\Omega} \hat{p}\, m_\kappa \nabla u \cdot \boldsymbol{n}\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t + \int_\Omega \hat{q}\, [u(0, \boldsymbol{x}) - m_0]\, \mathrm{d}\boldsymbol{x} \quad \text{for all } (\hat{p}, \hat{q}) \in \mathcal{P} \times L^2(\Omega). \end{aligned}$$

Next we require that variations of $\mathcal{L}^g$ with respect to u vanish for all admissible variations $\hat{u} \in \mathcal{U}$,

$$\delta_u \mathcal{L}^g = 0 \quad \text{for all } \hat{u} \in \mathcal{U} \quad \Longrightarrow \quad \text{weak form of adjoint PDE.}$$

This condition yields the *weak form of the adjoint advection–diffusion–reaction problem*: Given $(m_\kappa, m_0) \in H^1(\Omega) \times H^1(\Omega)$ and $(u, q) \in \mathcal{U} \times L^2(\Omega)$, find $p \in \mathcal{P}$ such that

$$\begin{aligned} &\int_0^T \int_\Omega \left[p \frac{\partial \hat{u}}{\partial t} + p \boldsymbol{v} \cdot \nabla \hat{u} + m_\kappa \nabla \hat{u} \cdot \nabla p + 3cpu^2 \hat{u} + (\mathcal{B}u - d)\mathcal{B}\hat{u} \right] \mathrm{d}\boldsymbol{x}\, \mathrm{d}t \\ &\quad - \int_0^T \int_{\partial\Omega} p\, m_\kappa \nabla \hat{u} \cdot \boldsymbol{n}\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t + \int_\Omega q\, \hat{u}(0, \boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \quad \text{for all } \hat{u} \in \mathcal{U}. \qquad (5.8) \end{aligned}$$

To recover the strong form of the adjoint problem (5.8), we integrate the time derivative term $p(\partial \hat{u}/\partial t)$ by parts in time, and the advection and diffusion terms $p\boldsymbol{v} \cdot \nabla \hat{u}$ and $m_\kappa \nabla \hat{u} \cdot \nabla p$ by parts in space, to remove derivatives of $\hat{u}$; then, combining terms, we obtain

$$\begin{aligned} &\int_0^T \int_\Omega \hat{u} \underbrace{\left[-\frac{\partial p}{\partial t} - \nabla \cdot (p\boldsymbol{v}) - \nabla \cdot (m_\kappa \cdot \nabla p) + 3cpu^2 + \mathcal{B}^*(\mathcal{B}u - d) \right]}_{\text{A}} \mathrm{d}\boldsymbol{x}\, \mathrm{d}t \\ &\quad + \int_\Omega \hat{u}(T, \boldsymbol{x})\, \underbrace{p(T, \boldsymbol{x})}_{\text{B}}\, \mathrm{d}\boldsymbol{x} + \int_\Omega \hat{u}(0, \boldsymbol{x})\, \underbrace{[q - p(0, \boldsymbol{x})]}_{\text{C}}\, \mathrm{d}\boldsymbol{x} \end{aligned}$$

$$
\begin{aligned}
&- \int_0^T \int_{\partial\Omega} (m_\kappa \nabla \hat{u} \cdot \boldsymbol{n}) \underbrace{\ p\ }_{\mathrm{D}} \,\mathrm{d}\boldsymbol{s}\,\mathrm{d}t \\
&+ \int_0^T \int_{\partial\Omega} \hat{u}\,(p\boldsymbol{v} \cdot \boldsymbol{n} + m_\kappa \nabla p \cdot \boldsymbol{n})\,\mathrm{d}\boldsymbol{s}\,\mathrm{d}t = 0 \quad \text{for all } \hat{u} \in \mathcal{U}, \qquad (5.9)
\end{aligned}
$$

where $\mathcal{B}^* : \mathcal{D} \to \mathcal{U}^*$ is the adjoint of the observation operator. Since $\hat{u} \in \mathcal{U}$, $\hat{u}$ vanishes on $\partial\Omega \times (0, T)$, and the last term in (5.9) vanishes. Now we proceed to make arguments about the arbitrariness of $\hat{u}$ on different portions of the space–time cylinder $\Omega \times (0, T)$ to recover the adjoint ADR equation and its boundary and terminal conditions (TC):

$$
\begin{aligned}
\hat{u} \text{ is arbitrary in } \Omega \times (0,T) \quad &\Longrightarrow \quad \mathrm{A} = 0 \quad \text{in } \Omega \times (0,T) \quad &&\text{(PDE)},\\
\hat{u} \text{ is arbitrary on } \Omega \times \{t = T\} \quad &\Longrightarrow \quad \mathrm{B} = 0 \quad \text{on } \Omega \times \{t = T\} \quad &&\text{(TC)},\\
m_\kappa \nabla \hat{u} \cdot \boldsymbol{n} \text{ is arbitrary on } \partial\Omega \times (0,T) \quad &\Longrightarrow \quad \mathrm{D} = 0 \quad \text{on } \partial\Omega \times (0,T) \quad &&\text{(BC)}.
\end{aligned}
$$

This leads to the *strong form of the adjoint advection–diffusion–reaction terminal-boundary value problem*:

$$
\begin{aligned}
-\frac{\partial p}{\partial t} - \nabla \cdot (p\boldsymbol{v}) - \nabla \cdot (m_\kappa \nabla p) + 3cu^2 p = -\mathcal{B}^*(\mathcal{B}u - d) \quad &\text{in } \Omega \times (0,T), &&(5.10)\\
p = 0 \quad &\text{on } \partial\Omega \times (0,T), &&(5.11)\\
p|_{t=T} = 0 \quad &\text{in } \Omega. &&(5.12)
\end{aligned}
$$

Note the following properties of the adjoint problem (5.10)–(5.12), which are shared more generally with many other systems.

- The condition (5.12) is a terminal condition (it holds at $t = T$), which means the adjoint problem (5.10)–(5.12) is solved backward in time for $p(t, \boldsymbol{x})$ from $t = T$ to $t = 0$.
- The sign of the time derivative and of the advective velocity are reversed. The equation remains stable, since the time reversal and sign change make the adjoint equation's operator equivalent to that of the forward problem (modulo the non-linear term, and for a divergence-free velocity).
- The adjoint problem is linear in the adjoint variable p, and depends non-linearly on the state u. Thus the forward-in-time computed state is required to backward-in-time compute the adjoint. Typically main memories of computers are not sized to store the entire space–time state, and remote memories are too slow to retrieve it at each time step. Rather than store the entire state, one can periodically store checkpoints of the state, and then recompute the state forward over each time interval, starting from each checkpoint, while integrating the adjoint equation backward in time. This is the simplest of a family of *checkpointing methods*; it avoids storage of the entire space–time

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

state at the cost of an additional solve of the forward problem. More sophisticated schemes that trade off additional forward solves for reduced storage are also possible (Griewank 1992, Griewank and Walther 2008).

- The adjoint problem inherits a homogeneous form of the forward problem's Dirichlet boundary condition.
- The source term for the adjoint PDE is given by $\mathcal{B}^*(\mathcal{B}u - d)$, *i.e.* the data misfit mapped back to the PDE residual space. This is the only source term for the adjoint problem. Thus, when the predicted observables fit the data, the adjoint variable $p(t, \boldsymbol{x})$ is identically zero.

Finally, there is one more term we have not used in (5.9), and that is the one involving expression C. Since the initial condition was imposed weakly, $\hat{u}$ is arbitrary on $\Omega \times \{t = 0\}$, which implies that C = 0 on $\Omega \times \{t = 0\}$. This provides an identity for $q \in L^2(\Omega)$, the Lagrange multiplier for the initial condition:

$$q(\boldsymbol{x}) = p(0, \boldsymbol{x}) \quad \text{in } \Omega \times \{t = 0\}.$$

This identity plays no role in the adjoint problem, but we will need it in the expression for the gradient.

At last we are in a position to derive expressions for the gradient, or rather the Fréchet derivative of ϕ, which we denote $\mathfrak{D}\phi$, with respect to both m_κ and m_0. This is given by variations of the Lagrangian $\mathcal{L}^g$ with respect to the diffusion coefficient $m_\kappa \in H^1$ and the initial condition $m_0 \in H^1$:

$$\begin{aligned}\delta_{m_\kappa}\mathcal{L}^g &= \mathfrak{D}_{m_\kappa}\phi := \text{Fréchet derivative of } \phi \text{ at } m_\kappa \text{ in direction } \hat{m}_\kappa \text{ for all } \hat{m}_\kappa \in H^1,\\ \delta_{m_0}\mathcal{L}^g &= \mathfrak{D}_{m_0}\phi := \text{Fréchet derivative of } \phi \text{ at } m_0 \text{ in direction } \hat{m}_0 \text{ for all } \hat{m}_0 \in H^1,\end{aligned}$$

where $\hat{m}_\kappa$ is the variation of m_κ, and $\hat{m}_0$ is the variation of m_0.

Thus, given $(m_\kappa, m_0) \in H^1 \times H^1$, *the Fréchet derivative of* $\phi(m_\kappa, m_0)$ *with respect to* m_κ *in an arbitrary direction* $\hat{m}_\kappa$ *evaluated at* (m_κ, m_0) is given by $\delta_{m_\kappa}\mathcal{L}^g$, namely

$$\begin{aligned}&\mathfrak{D}_{m_k}\phi(m_k, m_0, \hat{m}_k)\\ &\quad := \beta_\kappa \int_\Omega \nabla\hat{m}_\kappa \cdot \nabla m_\kappa \,\mathrm{d}\boldsymbol{x} + \int_0^T \int_\Omega \hat{m}_\kappa \nabla u \cdot \nabla p \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \quad \text{for all } \hat{m}_\kappa \in H^1, \qquad (5.13)\end{aligned}$$

where $u \in \mathcal{U}$ satisfies the forward ADR problem (5.7) and $p \in \mathcal{U}$ satisfies the adjoint ADR problem (5.8) for given (m_κ, m_0). Note that in $\delta_{m_\kappa}\mathcal{L}^g$ we have discarded the term

$$-\int_0^T \int_{\partial\Omega} \hat{m}_\kappa \nabla u \cdot \boldsymbol{n}\, p \,\mathrm{d}\boldsymbol{s}\,\mathrm{d}t$$

due to the adjoint Dirichlet boundary condition $p = 0$ on $\partial\Omega \times (0, T)$.

The gradient $\mathcal{G}$ with respect to m_k is defined as the Riesz representer of the Fréchet derivative of ϕ in (5.13) with respect to a chosen inner product:

$$(\mathcal{G}_\kappa(m_\kappa, m_0), \hat{m}_\kappa) := \mathfrak{D}_{m_k}\phi(m_k, m_0, \hat{m}_k).$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

For the L^2 inner product, $\mathcal{G}_\kappa$ can be extracted from the Fréchet derivative by integrating the regularization term by parts to remove the derivative off of $\hat{m}_\kappa$, which yields

$$
\begin{aligned}
(\mathcal{G}_\kappa(m_\kappa, m_0), \hat{m}_\kappa) := \int_\Omega \hat{m}_\kappa \left[-\beta_\kappa\, \Delta m_\kappa + \int_0^T \nabla u \cdot \nabla p \,\mathrm{d}t \right] \mathrm{d}\boldsymbol{x} \\
+ \int_{\partial\Omega} \hat{m}_\kappa\, \beta_\kappa \nabla m_\kappa \cdot \boldsymbol{n} \,\mathrm{d}\boldsymbol{s} \quad \text{for all } \hat{m}_\kappa \in H^1.
\end{aligned}
$$

Arguing that $\hat{m}_\kappa$ is arbitrary in Ω and on $\partial\Omega$ leads to

$$
\mathcal{G}_\kappa(m_\kappa, m_0) := \begin{cases} -\beta_\kappa \Delta m_\kappa + \displaystyle\int_0^T \nabla u \cdot \nabla p \,\mathrm{d}t & \text{in } \Omega, \\ \beta_\kappa \nabla m_\kappa \cdot \boldsymbol{n} & \text{on } \partial\Omega. \end{cases} \tag{5.14}
$$

The Neumann boundary term for m_κ is tied to the choice of the regularization: we assume that either m_κ is known on the boundary, in which case the gradient of ϕ is defined as zero on $\partial\Omega$, or else the normal derivative of m_κ vanishes. We have made the latter choice.

Similarly $(m_\kappa, m_0) \in H^1 \times H^1$, *the Fréchet derivative of* $\phi(m_\kappa, m_0)$ *with respect to* m_0 *in an arbitrary direction* $\hat{m}_0$ *evaluated at* (m_κ, m_0), is given by $\delta_{m_0}\mathcal{L}^g$, namely

$$
\mathfrak{D}_{m_0}\phi(m_k, m_0, \hat{m}_0) := \beta_0 \int_\Omega \nabla \hat{m}_0 \cdot \nabla m_0 \,\mathrm{d}\boldsymbol{x} + \int_\Omega \hat{m}_0\, p(0, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \quad \text{for all } \hat{m}_0 \in H^1, \tag{5.15}
$$

where again $u \in \mathcal{U}$ satisfies the forward ADR problem (5.7) and $p \in \mathcal{U}$ satisfies the adjoint ADR problem (5.8), for given (m_κ, m_0). Note that in $\delta_{m_\kappa}\mathcal{L}^g$ we have made use of the identity for the Lagrange multiplier for the initial condition, $q = p(0, \boldsymbol{x})$.

The gradient $\mathcal{G}_0$ with respect to m_0 is again defined as the Riesz representer of the Fréchet derivative in (5.15). It can be extracted by integrating the regularization term by parts and arguing that m_0 is arbitrary in Ω and on $\partial\Omega$, to obtain

$$
\mathcal{G}_0(m_\kappa, m_0) := \begin{cases} -\beta_\kappa \Delta m_\kappa + p(0, \boldsymbol{x}) & \text{in } \Omega, \\ \beta_\kappa \nabla m_\kappa \cdot \boldsymbol{n} & \text{on } \partial\Omega. \end{cases} \tag{5.16}
$$

In summary, to compute the gradients $\mathcal{G}_\kappa$ and $\mathcal{G}_0$ at (m_κ, m_0), we do the following.

1 Solve the forward ADR problem (5.7) for u, given (m_κ, m_0).

2 Solve the adjoint ADR problem (5.8) for p, given (m_κ, m_0) and u.

3 Evaluate Fréchet derivatives (5.13) and (5.15) given u, p and (m_κ, m_0).

A number of observations can be made about the structure of the Fréchet derivatives (5.13) and (5.15) or their gradient counterparts (5.14) and (5.16) (to which we refer below as the ‘gradient’, accepting some abuse of terminology in exchange for simplicity of presentation).

- As can be seen from the gradient expressions, the cost of computing the gradient (beyond solving the forward problem as required by the objective) is just solution of the adjoint ADR problem. When the forward problem is non-linear, the additional cost of solving the (linear) adjoint problem, and thus evaluating the gradient, may be negligible. (Here our unit of cost is the PDE solve, since all remaining linear algebra is negligible for large-scale problems.) What makes the adjoint ADR solve even cheaper is the fact that the adjoint operator is just the adjoint of the forward operator, and thus preconditioner construction or factorization are done just once for the forward problem and are available for free in the adjoint problem. (More discussion about amortizing forward problem computations will follow after the Hessian discussion in Section 5.3.) So adjoint-based gradient computation can essentially come for free in the case of a highly non-linear forward solve, but is at most twice the cost of the forward solve when the forward problem is linear and explicitly integrated (but see the discussion on checkpointing below). This situation is far superior to the direct method for computing gradients (using *sensitivity equations*, *e.g.* Gunzburger 2003), which costs as many linearized forward PDE solves as there are parameters, n_m. Moreover, finite difference approximation of the gradient would require at least n_m non-linear forward solves, and would likely produce unreliable gradients for a sufficiently complex forward problem. Automatic differentiation of the computer code provides another route to the gradient; see the discussion below after the Hessian derivation.

- As can be seen in the expressions for the gradient (5.13) or (5.14), the terms that come from the data misfit (the ones involving the adjoint p) vanish when $p = 0$, *i.e.* when the data misfit is zero. However, the terms in the gradient stemming from the regularization functional (the ones involving β_κ and β_0) prevent the gradients $\mathcal{G}_\kappa$ and $\mathcal{G}_0$ from vanishing with $p = 0$ for $\beta_\kappa, \beta_0 > 0$. The regularization parameters act to avoid overfitting (*i.e.* fitting the noise) by balancing the data misfit with the smoothing of m_κ and m_0.

- The gradient of the data misfit term with respect to the initial condition m_0 is given simply by the value of the adjoint variable p at time $t = 0$. Of course one needs to integrate the adjoint ADR equation backward in time to arrive at this value, which in turn requires solving the forward ADR equation, forward in time. Were it not for the non-linear reaction term cu^3, one could avoid storage of the full space–time state u and record only the state at observation locations, which are then used to drive the adjoint ADR equation backward in time. This is in contrast to the gradient with respect to the diffusion coefficient m_κ, in which the state u and adjoint p are accumulated over time to form the gradient, and thus both are required to be available at each time instant. As mentioned above, checkpointing circumvents the storage difficulties this causes, at the cost of an additional forward solve.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

- In the state estimation case, m_0 appears linearly in the forward ADR problem (5.2). Thus, if the forward problem is linear ($c = 0$), then the state u depends linearly on m_0, the adjoint p depends linearly on u and thus m_0, and the gradient (5.15), which is linear in p, depends linearly on m_0. Since in this case the gradient is linear in m_0, the inverse problem ($\mathcal{G}_0^d = 0$) is linear. (Linearity of the inverse problem is of course lost when the forward ADR problem is non-linear.) The same cannot be said about the coefficient estimation case: even when the forward problem is linear, the diffusion term is bilinear in u and m_κ, u thus depends on m_κ non-linearly, and the inverse problem is non-linear.
- With the gradient now determined, we could solve the joint inverse problem in the regularization framework (5.1) using the steepest descent method, by searching in a direction $(\tilde{m}_\kappa^i, \tilde{m}_0^i) \in H^1 \times H^1$ such that

$$\begin{Bmatrix} (\tilde{m}_\kappa^i, \hat{m}_\kappa) \\ (\tilde{m}_0^i, \hat{m}_0) \end{Bmatrix} = -\begin{Bmatrix} (\mathcal{G}_\kappa(m_\kappa^i, m_0^i), \hat{m}_\kappa) \\ (\mathcal{G}_0(m_\kappa^i, m_0^i), \hat{m}_0) \end{Bmatrix} \quad \text{for all } (\hat{m}_\kappa, \hat{m}_0) \in H^1 \times H^1, \tag{5.17}$$

  and then updating the parameter fields by

$$\begin{Bmatrix} m_\kappa^{i+1} \\ m_0^{i+1} \end{Bmatrix} = -\begin{Bmatrix} m_\kappa^i + \alpha^i \tilde{m}_\kappa^i \\ m_0^i + \alpha^i \tilde{m}_0^i \end{Bmatrix}$$

  for an α^i obtained by a suitable line search. However, steepest descent is a poor choice for solving inverse problems governed by PDEs, typically requiring a discretization-dependent and very large number of iterations. Instead, the Newton-CG method typically converges in a discretization-independent and small number of iterations. In addition to the gradient, it requires the action of the Hessian in a given direction. How to obtain this Hessian action for the inverse problem (5.1) is discussed next. Of course, as explained in Section 4, efficient methods for Laplace approximation of the Bayesian inverse solution require the Hessian action as well.

5.3. *The Hessian action*

Explicit construction of the (discrete) Hessian itself is prohibitive, since it requires as many linearized forward solves as there are parameters. Instead, as we will show in this section, the action of the Hessian of ϕ in an arbitrary direction $(\tilde{m}_\kappa, \tilde{m}_0)$ can be computed at the cost of a pair of linearized forward/adjoint solves.

We will follow a path similar to that used for the gradient, except now the Fréchet derivatives (5.13) and (5.15) themselves are the functions we seek to differentiate in order to form the Hessian actions. That is, we construct a Lagrangian functional $\mathcal{L}^h$ that consists of the Fréchet derivatives of ϕ with respect to m_κ and m_0 in directions $\tilde{m}_\kappa \in H^1$ and $\tilde{m}_0 \in H^1$, respectively (so we replace $\hat{m}_\kappa$ and $\hat{m}_0$ in (5.13) and (5.15) with $\tilde{m}_\kappa$ and $\tilde{m}_0$), along with residuals of the PDEs that must be satisfied to evaluate the Fréchet derivatives. These include not only the forward ADR problem but also the adjoint ADR problem. The forward ADR problem (5.7) is enforced in $\mathcal{L}^h$ via

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

the Lagrange multiplier $\tilde{p} \in \mathcal{P}$, which replaces p in (5.7). Similarly, the adjoint ADR problem (5.8) is enforced in $\mathcal{L}^h$ via the Lagrange multiplier $\tilde{u} \in \mathcal{U}$, which replaces $\hat{u}$ in (5.8). (We wish to retain the notation $\hat{u}$ and $\hat{p}$ for variations of u and p, respectively.) We will refer to the Lagrange multipliers $\tilde{u}$ and $\tilde{p}$ as the *incremental state* and the *incremental adjoint* variables, respectively, and below we will derive PDEs that govern their behaviour. This superscript h in $\mathcal{L}^h$ denotes the role of this Lagrangian in deriving the Hessian.

The Lagrangian for the Hessian then reads

$$\begin{aligned}
&\mathcal{L}^h(u, p, m_\kappa, m_0, \tilde{u}, \tilde{p}, \tilde{m}_\kappa, \tilde{m}_0) \\
&\quad := \underbrace{\beta_\kappa \int_\Omega \nabla \tilde{m}_\kappa \cdot \nabla m_\kappa \,\mathrm{d}\boldsymbol{x} + \int_0^T \int_\Omega \tilde{m}_\kappa \nabla u \cdot \nabla p \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t}_{\text{Fréchet derivative with respect to } m_\kappa \text{ in direction } \tilde{m}_\kappa} \\
&\qquad + \underbrace{\beta_0 \int_\Omega \nabla \tilde{m}_0 \cdot \nabla m_0 \,\mathrm{d}\boldsymbol{x} + \int_\Omega \tilde{m}_0\, p(0, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}}_{\text{Fréchet derivative with respect to } m_0 \text{ in direction } \tilde{m}_0} \\
&\qquad + \underbrace{\int_0^T \int_\Omega \left[\tilde{p} \frac{\partial u}{\partial t} + \tilde{p}\boldsymbol{v} \cdot \nabla u + m_\kappa \nabla u \cdot \nabla \tilde{p} + c\tilde{p}u^3 - \tilde{p}f \right] \mathrm{d}\boldsymbol{x}\,\mathrm{d}t}_{\text{weak form of forward ADR equation}} \qquad (5.18) \\
&\qquad + \underbrace{\int_0^T \int_\Omega \left[p \frac{\partial \tilde{u}}{\partial t} + p\boldsymbol{v} \cdot \nabla \tilde{u} + m_\kappa \nabla \tilde{u} \cdot \nabla p + 3cpu^2\tilde{u} + (\mathcal{B}u - d)\mathcal{B}\tilde{u} \right] \mathrm{d}\boldsymbol{x}\,\mathrm{d}t}_{\text{weak form of adjoint ADR equation}} \\
&\qquad + \underbrace{\int_\Omega \tilde{p}(0, \boldsymbol{x})[u(0, \boldsymbol{x}) - m_0] \,\mathrm{d}\boldsymbol{x}}_{\text{weak forward initial condition}} + \underbrace{\int_\Omega p(0, \boldsymbol{x})\, \tilde{u}(0, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}}_{\text{forward initial condition from adjoint equation}},
\end{aligned}$$

where $(u, p, m_\kappa, m_0, \tilde{u}, \tilde{p}, \tilde{m}_\kappa, \tilde{m}_0) \in \mathcal{U} \times \mathcal{U} \times H^1 \times H^1 \times \mathcal{U} \times \mathcal{P} \times H^1 \times H^1$. Note that in the expression for $\mathcal{L}^h$ above, we have not included the boundary terms

$$-\int_0^T \int_{\partial\Omega} p\, m_\kappa \nabla u \cdot \boldsymbol{n} \,\mathrm{d}s\,\mathrm{d}t$$

from the forward ADR problem (5.7) and

$$-\int_0^T \int_{\partial\Omega} p\, m_\kappa \nabla \hat{u} \cdot \boldsymbol{n} \,\mathrm{d}s\,\mathrm{d}t$$

from the adjoint ADR problem (5.8), since we deduced that p satisfies the homogeneous adjoint Dirichlet boundary condition $p(t, \boldsymbol{x}) = 0$ on $\partial\Omega \times (0, T)$ while deriving the adjoint ADR problem. As a consequence, we have updated our knowledge of the admissible space for p, and have written $p \in \mathcal{U}$. Notice also that

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

we have replaced q, the Lagrange multiplier for the initial condition in (5.7), with $p(0,\boldsymbol{x})$, as deduced in the derivation of the adjoint ADR equation.

To derive an expression for the action of the Hessian of ϕ with respect to m_κ, m_0 in a direction $\tilde{m}_\kappa, \tilde{m}_0$, we take variations of the Lagrangian $\mathcal{L}^h$ (5.18) with respect to its arguments (u, p, m_κ, m_0). First, requiring variations of $\mathcal{L}^h$ with respect to the adjoint p to vanish for all admissible variations $\hat{p} \in \mathcal{P}$ yields the so-called incremental forward problem,

$$\delta_p \mathcal{L}^h = 0 \quad \text{for all } \hat{p} \in \mathcal{P} \quad \Longrightarrow \quad \text{weak form of incremental forward problem.}$$

The *weak form of the incremental forward advection–diffusion–reaction problem* then takes the form: Given $(m_\kappa, m_0) \in H^1 \times H^1$, $(\tilde{m}_\kappa, \tilde{m}_0) \in H^1 \times H^1$ and $u \in \mathcal{U}$, find the incremental state $\tilde{u} \in \mathcal{U}$ such that

$$\begin{aligned}\int_0^T \int_\Omega \biggl[\hat{p}\frac{\partial \tilde{u}}{\partial t} + \hat{p}\boldsymbol{v}\cdot\nabla\tilde{u} + m_\kappa \nabla\hat{p}\cdot\nabla\tilde{u} + 3c\hat{p}u^2\tilde{u} + \tilde{m}_\kappa\nabla\hat{p}\cdot\nabla u\biggr]\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t& \\ + \int_\Omega \hat{p}(0,\boldsymbol{x})[\tilde{u}(0,\boldsymbol{x}) + \tilde{m}_0]\,\mathrm{d}\boldsymbol{x} = 0 \quad \text{for all } \hat{p} \in \mathcal{P}.& \qquad (5.19)\end{aligned}$$

Integrating by parts the two diffusion terms to remove the derivatives from $\hat{p}$, invoking the condition $\hat{p} = 0$ in $\partial\Omega \times (0,T)$ and arguing that $\hat{p}$ is arbitrary in $\Omega \times (0,T)$ and on $\Omega \times \{t = 0\}$, yields the *strong form of the incremental forward advection–diffusion–reaction problem*:

$$\begin{aligned}\frac{\partial \tilde{u}}{\partial t} + \boldsymbol{v}\cdot\nabla\tilde{u} - \nabla\cdot(m_\kappa\nabla\tilde{u}) + 3cu^2\tilde{u} &= \nabla\cdot(\tilde{m}_\kappa\nabla u) && \text{in } \Omega\times(0,T),\\ \tilde{u} &= 0 && \text{on } \partial\Omega\times(0,T), \qquad (5.20)\\ \tilde{u}|_{t=0} &= -\tilde{m}_0 && \text{in } \Omega.\end{aligned}$$

Similarly, taking variations of $\mathcal{L}^h$ with respect to $u \in \mathcal{U}$ and requiring them to vanish for all admissible variations yields the incremental adjoint problem,

$$\delta_u \mathcal{L}^h = 0 \quad \text{for all } \hat{u} \in \mathcal{U} \quad \Longrightarrow \quad \text{weak form of incremental adjoint problem.}$$

The *weak form of the incremental adjoint advection–diffusion–reaction problem* then takes the form: Given $(m_\kappa, m_0) \in H^1 \times H^1$, $(\tilde{m}_\kappa, \tilde{m}_0) \in H^1 \times H^1$, $u \in \mathcal{U}$, $p \in \mathcal{P}$ and $\tilde{u} \in \mathcal{U}$, find the incremental adjoint $\tilde{p} \in \mathcal{P}$ such that

$$\begin{aligned}\int_0^T \int_\Omega \biggl[\tilde{p}\frac{\partial \hat{u}}{\partial t} + \tilde{p}\,\boldsymbol{v}\cdot\nabla\hat{u} + m_\kappa\nabla\hat{u}\cdot\nabla\tilde{p} + 3cu^2\tilde{p}\hat{u} + 6cpu\tilde{u}\hat{u}&\\ + \mathcal{B}\hat{u}\,\mathcal{B}\tilde{u} + \tilde{m}_\kappa\nabla\hat{u}\cdot\nabla p\biggr]\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_\Omega \tilde{p}(0,\boldsymbol{x})\,\hat{u}(0,\boldsymbol{x}) = 0 \quad \text{for all } \hat{u} \in \mathcal{U}.& \quad (5.21)\end{aligned}$$

Integrating by parts the time derivative, advection and two diffusion terms to remove the derivatives from $\hat{u}$, invoking the condition $\hat{u} = 0$ in $\partial\Omega \times (0,T)$ and arguing that $\hat{u}$ is arbitrary in $\Omega \times (0,T)$ and on $\Omega \times \{t = T\}$ yields the *strong form*

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

of the incremental adjoint advection–diffusion–reaction problem:

$$\begin{aligned}
-\frac{\partial \tilde{p}}{\partial t} - \nabla \cdot (\tilde{p}\boldsymbol{v}) - \nabla \cdot (m_\kappa \nabla \tilde{p}) + 3cu^2\tilde{p}& \\
= -\mathcal{B}^*\mathcal{B}\tilde{u} - 6cpu\tilde{u} + \nabla \cdot (\tilde{m}_\kappa \nabla p) &\quad \text{in } \Omega \times (0,T), \\
\tilde{p} = 0 &\quad \text{on } \partial\Omega \times (0,T), \\
\tilde{p}|_{t=T} = 0 &\quad \text{in } \Omega. \qquad (5.22)
\end{aligned}$$

We make the following observations about the incremental forward ADR problem (5.19) or (5.20), and the incremental adjoint ADR problem (5.21) or (5.22), both defined for a given direction $(\tilde{m}_\kappa, \tilde{m}_0)$ in which we wish to compute the Hessian action.

- The incremental forward problem (5.20) is a linearization of the forward problem (5.2)–(5.4) with respect to both the state u and the parameters (m_κ, m_0), with $\tilde{u}$ and $(\tilde{m}_\kappa, \tilde{m}_0)$ playing the role of the infinitesimals. Its operator is thus the linearized forward operator or the adjoint of the adjoint operator. It inherits the homogeneous Dirichlet condition from the forward problem, and its source term is the (negative of the) variation of the residual of the forward problem in the direction $(\tilde{m}_\kappa, \tilde{m}_0)$. Even if the inverse problem is non-linear (because the inversion parameter enters non-linearly into the forward problem), if the forward problem is linear in the state, then the incremental forward problem has the same operator as the forward problem.

- The incremental adjoint problem (5.22) is a linearization of the adjoint problem (5.10)–(5.12) with respect to u, p and (m_κ, m_0), with $\tilde{u}$, $\tilde{p}$ and $(\tilde{m}_\kappa, \tilde{m}_0)$ as infinitesimals. Its operator is thus identical to that of the adjoint problem, since that problem is already linear in the adjoint p. The incremental adjoint problem inherits the homogeneous Dirichlet condition from the adjoint problem, along with the adjoint problem's terminal condition, leading to $\tilde{p}|_{t=T} = 0$. Its source term is given by the (negative of the) linearization of the adjoint equation residual with respect to u and (m_κ, m_0). In the incremental adjoint ADR problem (5.22), the observation operator $\mathcal{B}$ appears in the source term (as it does for the adjoint equation), as a consequence of our assumption that the observations are acquired in $\Omega \times (0,T)$. Had they been acquired on the boundary $\partial\Omega$ or at final time $t = T$, the observation operator would have shown up respectively as a source term for the Dirichlet boundary condition or the terminal condition on $\tilde{p}$,

- The incremental forward problem requires the state u, while the incremental adjoint problem requires the state u, the adjoint p and the incremental state $\tilde{u}$. Since u and $\tilde{u}$ are solved forward in time and p is solved backward in time, a more involved checkpointing strategy will be required to solve backward-in-time for $\tilde{p}$. We defer this discussion of checkpointing until after we have derived the form of the Hessian action below.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Finally $\mathfrak{D}^2\phi$, the second Fréchet derivative of ϕ, *i.e.* the action of the Hessian of ϕ with respect to (m_κ, m_0) in a direction $(\tilde{m}_\kappa, \tilde{m}_0) \in H^1 \times H^1$, for all $(\hat{m}_\kappa, \hat{m}_0) \in H^1 \times H^1$, can be determined from variations of the Lagrangian $\mathcal{L}^h$ with respect to (m_κ, m_0). Since we are simultaneously inverting for the two parameter fields, the Hessian is a block operator, that is,

$$\mathcal{H}(m_\kappa, m_0) := \begin{bmatrix} \mathcal{H}_{\kappa\kappa} & \mathcal{H}_{\kappa 0} \\ \mathcal{H}_{0\kappa} & \mathcal{H}_{00} \end{bmatrix}.$$

Thus, given (m_κ, m_0), the action of the first block row of $\mathcal{H}$ in the direction $(\tilde{m}_\kappa, \tilde{m}_0)$, evaluated at (m_κ, m_0), for all $\hat{m}_\kappa$, is given by variations of $\mathcal{L}^h$ with respect to m_κ, yielding

$$\begin{aligned}\delta_{m_\kappa}\mathcal{L}^h &:= (\hat{m}_\kappa, \mathcal{H}_{\kappa\kappa}\,\tilde{m}_\kappa) + (\hat{m}_\kappa, \mathcal{H}_{\kappa 0}\,\tilde{m}_0) := \beta_\kappa \int_\Omega \nabla\hat{m}_\kappa \cdot \nabla\tilde{m}_\kappa \,\mathrm{d}\boldsymbol{x} \\ &\quad + \int_0^T \int_\Omega [\hat{m}_\kappa \nabla u \cdot \nabla\tilde{p} + \hat{m}_\kappa \nabla\tilde{u} \cdot \nabla p] \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \quad \text{for all } \hat{m}_\kappa \in H^1. \qquad (5.23)\end{aligned}$$

Similarly, the action of the second block row of $\mathcal{H}$ in the direction $(\tilde{m}_\kappa, \tilde{m}_0)$, evaluated at (m_κ, m_0), for all m_0, is given by variations of $\mathcal{L}^h$ with respect to m_0, yielding

$$\begin{aligned}\delta_{m_0}\mathcal{L}^h &:= (\hat{m}_0, \mathcal{H}_{0\kappa}\,\tilde{m}_\kappa) + (\hat{m}_0, \mathcal{H}_{00}\,\tilde{m}_0) \\ &:= \beta_0 \int_\Omega \nabla\hat{m}_0 \cdot \nabla\tilde{m}_0 \,\mathrm{d}\boldsymbol{x} - \int_\Omega \hat{m}_0\,\tilde{p}(0, \boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \quad \text{for all } \hat{m}_0 \in H^1. \qquad (5.24)\end{aligned}$$

In the Hessian action expression (5.23) (corresponding to m_κ), u satisfies the forward ADR problem (5.5), p satisfies the adjoint ADR problem (5.8), $\tilde{u}$ satisfies the incremental forward ADR problem (5.19) and $\tilde{p}$ satisfies the incremental adjoint problem (5.21). On the other hand, the Hessian action (5.24) (corresponding to m_0) depends only on $\tilde{p}$, though this variable in turn depends on the u, p and $\tilde{u}$ through the solution of the incremental adjoint ADR problem.

As with the gradients, ‘strong’ forms of the Hessian action in the direction $(\tilde{m}_\kappa, \tilde{m}_0)$ can be extracted from the second Fréchet derivatives (5.23) and (5.24), leading to the strong form of the action of the first block row,

$$\mathcal{H}_{\kappa\kappa}\tilde{m}_\kappa + \mathcal{H}_{\kappa 0}\tilde{m}_0 := \begin{cases} -\beta_\kappa \Delta\tilde{m}_\kappa + \displaystyle\int_0^T [\nabla u \cdot \nabla\tilde{p} + \nabla\tilde{u} \cdot \nabla p] \,\mathrm{d}t & \text{in } \Omega, \\ \beta_\kappa \nabla\tilde{m}_\kappa \cdot \boldsymbol{n} & \text{on } \partial\Omega, \end{cases} \qquad (5.25)$$

and the strong form of the action of the second block row,

$$\mathcal{H}_{0\kappa}\tilde{m}_\kappa + \mathcal{H}_{00}\tilde{m}_0 := \begin{cases} -\beta_0 \Delta\tilde{m}_0 + \tilde{p}(0, \boldsymbol{x}) & \text{in } \Omega, \\ \beta_0 \nabla\tilde{m}_0 \cdot \boldsymbol{n} & \text{on } \partial\Omega. \end{cases} \qquad (5.26)$$

In summary, to compute the Hessian action at a point in parameter space (m_κ, m_0) in the direction $(\tilde{m}_\kappa, \tilde{m}_0)$, we do the following.

1 Solve the forward ADR problem (5.7) for u, given (m_κ, m_0).

2 Solve the adjoint ADR problem (5.8) for p, given (m_κ, m_0) and u.

3 Solve the incremental forward ADR problem (5.19) for $\tilde{u}$, given (m_κ, m_0), $(\tilde{m}_\kappa, \tilde{m}_0)$ and u.

4 Solve the incremental adjoint ADR problem (5.21) for $\tilde{p}$, given (m_κ, m_0), $(\tilde{m}_\kappa, \tilde{m}_0)$, u, p and $\tilde{u}$.

5 Evaluate the Hessian actions (5.23) and (5.24), given (m_κ, m_0), $(\tilde{m}_\kappa, \tilde{m}_0)$, u, p, $\tilde{u}$ and $\tilde{p}$.

In contrast with the steepest descent iteration (5.17) and with the Hessian actions (5.23) and (5.24) now elaborated, we can now can state a *Newton method for solving the joint inverse problem* (5.1): Search in a direction $(\tilde{m}_\kappa^i, \tilde{m}_0^i) \in H^1 \times H^1$ such that

$$\begin{Bmatrix} (\hat{m}_\kappa, \mathcal{H}_{\kappa\kappa}\, \tilde{m}_\kappa) + (\hat{m}_\kappa, \mathcal{H}_{\kappa 0}\, \tilde{m}_0) \\ (\hat{m}_0, \mathcal{H}_{0\kappa}\, \tilde{m}_\kappa) + (\hat{m}_0, \mathcal{H}_{00}\, \tilde{m}_0) \end{Bmatrix} = -\begin{Bmatrix} (\mathcal{G}_\kappa(m_\kappa^i, m_0^i), \hat{m}_\kappa) \\ (\mathcal{G}_0(m_\kappa^i, m_0^i), \hat{m}_0) \end{Bmatrix} \tag{5.27}$$

for all $(\hat{m}_\kappa, \hat{m}_0) \in H^1 \times H^1$, and then update the parameter fields by

$$\begin{Bmatrix} m_\kappa^{i+1} \\ m_0^{i+1} \end{Bmatrix} = -\begin{Bmatrix} m_\kappa^i + \alpha^i \tilde{m}_\kappa^i \\ m_0^i + \alpha^i \tilde{m}_0^i \end{Bmatrix}$$

for an α^i obtained by a suitable line search.

A number of observations can be made about the Hessian action $\mathcal{H}\tilde{m}$ (5.23) and (5.24) or its strong form counterpart (5.25) and (5.26), both in the context of the advection–diffusion–reaction problem and for more general PDEs.

Structure of the Hessian action. The action of the first block row ((5.23) or (5.25)), which represents a linearization of the gradient $\mathcal{G}_\kappa$ ((5.13) or (5.14)) in the direction $\tilde{m}_\kappa$, resembles $\mathcal{G}_\kappa$ but with a linearized $\nabla u \cdot \nabla p$ term (integrated over time). Similarly, the action of the second block row ((5.24) or (5.26)), which represents a linearization of the gradient $\mathcal{G}_0$ in the direction $\tilde{m}_0$ ((5.15) or (5.16)), resembles $\mathcal{G}_0$. No linearization is needed in this case, since the initial condition parameter m_0 appears linearly in the gradient, and the data misfit portion of the Hessian action involves just the evaluation of $\tilde{p}$ at $t = 0$ (after integrating backward from $t = T$).

The compute-bound case. As mentioned above, both Hessian action expressions depend on all four fields u, p, $\tilde{u}$ and $\tilde{p}$ (explicitly for the first block row, and implicitly through the dependence of $\tilde{p}$ on u, p and $\tilde{u}$ for the second block row). If sufficient (fast) memory is available, the simplest and most computationally efficient approach is to store all four fields when solving the corresponding state, adjoint, incremental state and incremental adjoint equations, and then evaluate the Hessian actions using these fields. The forward and adjoint problems would be solved just once to compute the gradient, and then their solution incorporated into solutions of the incremental forward and incremental adjoint problems for each Hessian action required in the algorithms of Sections 3 and 4. Since the incremental problems are linear, the cost of the two linearized PDE solves comprising the Hessian action may be substantially less than that of the gradient, depending on the non-linearity of the forward problem. When the forward problem is linear, all four systems have the same operator or its adjoint, and thus the costs of the gradient and the Hessian action are comparable (modulo the possibly more complex source terms of the incremental problems, which involve variations of the forward PDE residual with respect to the parameters).

The memory-bound case. However, as mentioned in the discussion of the gradient, main memories of computers are typically not sized to store entire space–time fields. Instead we must resort to checkpointing schemes. The most straightforward strategy is as follows. Solve the forward (5.2)–(5.4) and incremental forward (5.20) (which depends on u) problems simultaneously forward in time to yield u and $\tilde{u}$, checkpointing them at regular intervals based on available memory. Then solve the adjoint (5.10)–(5.12) (which depends on u) and incremental adjoint (5.22) (which depends on u, p and $\tilde{u}$) problems backward in time, re-solving the forward and incremental forward problems forward in time from their checkpoints over each interval to supply the adjoint and incremental adjoint equations with the needed fields u and $\tilde{u}$ within that interval. The four fields at each time step are accumulated over time to compute the $\int_0^T [\nabla u \cdot \nabla \tilde{p} + \nabla \tilde{u} \cdot \nabla p] \, \mathrm{d}t$ term in the Hessian action. On the other hand, the term $p(0, \boldsymbol{x})$ is directly extracted at the end of the backward-in-time incremental adjoint solution. Storage of u and $\tilde{u}$ over a typical time interval must be taken into account when deciding on the frequency of checkpoints. Altogether, two (non-linear) forward problems, two (linear) incremental forward problems, one (linear) adjoint problem and one (linear) incremental adjoint problem must be solved for each Hessian action. For highly non-linear forward problems, the Hessian action will be close to twice the cost of a gradient evaluation when the non-linear forward PDE solve dominates. For linear forward problems, it is three times the cost (these are all asymptotic costs, measured in units of PDE solves).

The Gauss–Newton approximation. Motivated by the fact that the only source term in the adjoint problem (5.10)–(5.12) is the data misfit $\mathcal{B}^*(\mathcal{B}u - d)$, and this adjoint problem determines the adjoint variable p, the *Gauss–Newton approximation* of the Hessian is obtained by setting $p = 0$ in the incremental adjoint problem and in the Hessian action expression. When the model fits the data exactly at the inverse solution or MAP point (as is the case for a sufficiently well-parametrized model in the absence of data noise and model error), then quadratic convergence to the solution can be expected (Kelley 1999). With real data, a zero data misfit is unrealistic; instead, if the data misfit is small, we can expect fast linear convergence. One benefit of this approximation is that the resulting Gauss–Newton Hessian is guaranteed, with appropriate regularization, to be positive definite everywhere (while the full Hessian is positive definite only in the vicinity of a local minimum) (Kelley 1999, Nocedal and Wright 2006). This guarantees a descent direction, and

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

global convergence with a suitable globalization strategy. However, as discussed in Section 3, the inexact Newton-CG method terminates the inner CG iteration early when it detects a negative curvature direction, and this allows it to maintain a descent direction. So the Gauss–Newton approximation does not offer an advantage in this sense. On the other hand, an advantage of the Gauss–Newton approximation is that it avoids having to solve the adjoint problem when solving the incremental adjoint problem (since $p = 0$), thereby reducing to five the number of PDE solves required in the basic checkpointing scheme described above. Moreover, when the forward problem is linear (in the state u), the incremental adjoint problem no longer depends on u, and since its dependence on p is suppressed, we no longer have to re-solve the forward problem from its checkpoints when solving the incremental adjoint problem. This reduces the number of PDE solves per Hessian action to just four with the Gauss–Newton approximation. Whether the trade-off of fewer PDE solves per Hessian action is worth the reduction to linear convergence (and what the convergence constant will be) will depend on the particular problem. Finally, a special case is when the forward problem is both linear in the state and in the parameter (and thus the inverse problem is linear). For example, this is the case for the example problem (5.1)–(5.4) when the reaction term vanishes ($c = 0$) and when we invert for the initial condition m_0 (*i.e.* $\tilde{m}_\kappa = 0$). In this case the Gauss–Newton approximation is exact, since p no longer appears in the incremental adjoint problem (5.22). However, this does not change the required number of PDE solves in the checkpointing scheme.

Summary of PDE solves for each Hessian action. Summarizing the above, we can distinguish the number of non-linear PDE solves (for the forward problem) and linear PDE solves (for the adjoint, incremental forward and incremental adjoint problem) for each Hessian action, based on the Hessian approximation and the non-linearity of the forward problem as follows, assuming the basic checkpointing scheme described above is employed.

- *Full Hessian, non-linear time-dependent forward problem.* Here, *two non-linear and four linear PDEs* must be solved: two forward, two incremental forward, one adjoint and one incremental adjoint.

- *Full Hessian, linear time-dependent forward problem.* In this case the forward problem is linear, so *six linear PDEs* must be solved, distributed as for the non-linear forward case above.

- *Gauss–Newton Hessian, non-linear time-dependent forward problem.* The Gauss–Newton approximation eliminates the adjoint solve, so *two non-linear and three linear PDEs* must be solved: two forward, two incremental forward and one incremental adjoint.

- *Gauss–Newton Hessian, linear time-dependent forward problem.* In the linear forward problem case, the incremental adjoint problem does not depend on u, further eliminating a (non-linear) forward solve. Thus *four linear PDEs*

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

must be solved: one forward, two incremental forward and one incremental adjoint.

- *Time-independent forward problem.* In this case no checkpointing is required, all four fields can be stored, and the Hessian action simply requires *two linear PDE solves* for the forward and adjoint pair of incremental problems.

Amortizing the cost of incremental PDE solves over multiple Hessian actions. The discussion above counts each PDE solve required across Hessian actions (for a single Newton step) as an independent solve. However, all of these PDE solves share the same operator or its adjoint, which is evaluated at the same point in parameter space. (In the case of a non-linear forward problem, the linearized forward operator at the solution is identical to the incremental forward operator, and its adjoint identical to the adjoint and incremental adjoint operators.) Moreover, these operators are also independent of the direction $(\tilde{m}_\kappa, \tilde{m}_0)$ in which the Hessian action is sought, which appears in the source terms of the forward and adjoint incremental PDE problems. Thus there is an opportunity to exploit the commonality of the PDE operator to greatly reduce the cost of solving the PDEs across the r Hessian actions required by CG in Section 3 and randomized eigensolvers in Section 4. This can be achieved in several ways. The most effective is when a direct linear solver is viable; in this case the triangular factorization of the forward operator can be carried out once, and the factors re-used across the multiple right-hand sides during the triangular solves. With iterative solvers, the opportunity for amortizing the offline computations is not as great as with triangular factorization, but opportunities still do exist in re-using the preconditioner construction. For domain decomposition preconditioners, the local subdomain factorizations, as well as coarse grid factorization, can be amortized across the right-hand sides. For multigrid preconditioners, the set-up of the coarse grid hierarchy can be re-used (*e.g.* prolongation/restriction operators, coarse grid factorization, AMG hierarchy). The opportunities for solver re-use, ranging from none to full, are distinguished by the following problem classes.

- *Explicitly integrated time-dependent forward problems.* No linear solves are executed in explicit integration, so no solver re-use is possible. There still exist opportunities to exploit simultaneous multiple right-hand sides in the randomized eigensolver algorithm, by blocking together a batch of right-hand sides, which increases the ratio of flops to memory accesses and thus improves locality and cache performance. So speed-ups can be obtained, but not at the scale of those described below.

- *Implicitly integrated time-dependent forward problems.* In this case solver re-use is possible. However, the operators of the adjoint and two incremental PDEs, although they are linear, do depend on the state u, which varies from time step to time step. It would be prohibitive to store the factors (or preconditioner) of the time-stepping operator at each time step (we usually cannot

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

store even the state at all time steps). In the special case of a *linear forward problem*, the adjoint and two incremental operators are independent of u, so there is just a single time-stepping operator. Thus storing its factors or preconditioner and re-using them within the related PDE solves during a Hessian action, and across all r Hessian actions, becomes attractive. For a direct solver, the entire Newton step within Newton-CG, or the truncated spectral decomposition for randomized eigensolver, comes nearly for free, since triangular solves are asymptotically negligible relative to the factorization. The savings are not as dramatic for iterative PDE solution, depending on how much is invested in the preconditioner construction, but can still be substantial for a sophisticated preconditioner.

- *Time-independent forward problem.* Regardless of whether the forward problem is linear or non-linear, in the time-independent case, the problem of storing factors or preconditioners at each time step is avoided, and the single operator factorization or preconditioner construction can be amortized over r Hessian actions. Thus again, depending on the offline versus online cost, the cost of the Hessian system solve or spectral decomposition can be negligible once the gradient has been computed (which is where the factorization or preconditioner is constructed).

Discretize-then-optimize versus optimize-then-discretize. Ultimately the expressions for the gradient and Hessian presented above must be discretized for computer implementation. The most natural choice is to directly discretize them using suitably chosen bases for the field variables (u, p, m_κ, m_0, $\tilde{u}$, $\tilde{p}$, $\tilde{m}_\kappa$, $\tilde{m}_0$) and their associated variations/test functions ($\hat{u}$, $\hat{p}$, $\hat{m}_\kappa$, $\hat{m}_0$). This transforms the infinite-dimensional Newton iteration (5.27) into the finite-dimensional method discussed in Section 3. However, another approach could be taken: first discretize the optimization problem (5.1) and its governing initial-boundary value problem (5.2)–(5.4), and then derive the corresponding finite-dimensional gradient and Hessian expressions, employing a finite-dimensional Lagrangian approach analogous to the one described in this section. The resulting finite-dimensional gradient and Hessian expressions then lead to a finite-dimensional Newton iteration as in Section 3. The former approach is often referred to as *optimize-then-discretize* (OTD), and the latter as *discretize-then-optimize* (DTO). Unfortunately the two approaches are not generally guaranteed to yield the same gradient and Hessian. Whether differentiation and discretization commute depends on the discretization method used. For example, use of a Galerkin or least-squares method in space and a Crank–Nicolson or four-stage fourth-order Runge–Kutta method in time are commutative, but discontinuous Galerkin or SUPG in space, or backward Euler or certain Runge–Kutta methods in time, may not be. Since a particular discretization method such as Galerkin may not be appropriate for a given problem (*e.g.* one prefers a stablilized method or discontinuous Galerkin for a convection-dominated problem), what should one do? DTO has the advantage that it leads to a consistent gradient (and

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Hessian), that is, the gradient that is computed is the gradient of the discretized objective function. Lack of consistency in the gradient, as can happen with OTD in the pre-asymptotic regime, may lead to a search direction that is not a descent direction with respect to the discretized objective, and thus lack of convergence of the optimization solver (Gunzburger 2003, Wilcox, Stadler, Bui-Thanh and Ghattas 2015). This would seem to favour DTO. However, deriving the gradient and Hessian with DTO can be tedious and error-prone, since one is differentiating through all of the components of the discretization (*e.g.* for finite element methods this includes element shape functions, numerical quadrature, imposition of boundary conditions, assembly, *etc.*). In contrast, the variationally based infinite-dimensional derivations illustrated in this section are more compact and 'clean' due to their differentiating the underlying PDEs. Indeed, there exist variationally based finite element frameworks such as FEniCS and Firedrake and inversion libraries built on top of them such as hIPPYlib that provide the functionality for automating the variational calculus needed to derive OTD-based gradients and Hessians, making the process even faster and less error-prone. More importantly, use of an inconsistent discretization may lead to deterioration in rates of convergence of the DTO-based discrete gradient to the infinite-dimensional gradient, or even non-convergence (Collis and Heinkenschloss 2002, Heinkenschloss and Leykekhman 2010). Since this gradient characterizes the first-order necessary condition for an optimum, this is problematic. This would seem to favour OTD. But, as mentioned above, one often prefers certain discretization methods that are well suited to the problem at hand, for example stabilized methods for hyperbolic systems. This has led to a search for discretization methods that are both 'optimal' for the PDE at hand as well as gradient-consistent (*e.g.* in the context of stabilization methods (Becker and Vexler 2007, Braack 2009, Leykekhman 2012, Yücel, Heinkenschloss and Karasözen 2013) or time integration of the semi-discrete system (Hager 2000)), or else aim to recover the lost convergence rate through such devices as regularization (Liu and Wang 2019).

Automatic differentiation. While a large body of work on automatic differentiation (AD) theory and tools is available to automate the DTO process (Griewank 2003, Griewank and Walther 2008, Naumann 2012), ultimately AD differentiates the code, as opposed to the model, meaning that it will differentiate through preconditioner and solver construction and non-linear iterations. This is not necessary for the adjoint, incremental state and incremental adjoint solves, which are linear problems and share the same coefficient matrix or its transpose with each other and with the linearized forward problem. Thus, while AD does deliver exact derivatives, it does so at a significant increase in cost. Moreover, the development of automatic differentiation tools for parallel distributed memory code remains challenging (Heimbach, Hill and Giering 2002, Utke *et al.* 2009). Nevertheless, AD may be the only realistic option for legacy codes or those with complex internal representations (such as look-up tables or subgrid-scale models).

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

6. Case study: an inverse problem for the Antarctic ice sheet

In this section we illustrate the application of the large-scale inversion methods of Section 3 (regularization-based) and Section 4 (Bayesian) to an inverse problem for the flow of the Antarctic ice sheet. A brief description of this problem and illustrative results are provided below; a more detailed presentation can be found in Isaac *et al.* (2015). The mass flux of ice from the Antarctic and Greenland ice sheets to the polar oceans is expected to be the dominant contributor to sea level rise this century. There is great uncertainty in the projections of land ice contributions to sea level rise from ice sheet models (Meehl *et al.* 2007). Satellite observational data can be used to reduce uncertainties in the ice sheet parameters via solution of an inverse problem. The greatest uncertainty in ice sheet models lies in the basal friction parameter, a heterogeneous field at the base of the ice sheet representing the resistance to sliding due to various difficult-to-model phenomena such as frictional behaviour of the ice, roughness of the bedrock and presence of water. We seek to infer this field from interferometric synthetic aperture radar (InSAR) satellite observations of the ice flow velocity on the top surface of the ice sheet, along with an ice flow model.

6.1. *Forward and inverse ice flow*

The ice is modelled as a creeping, viscous, incompressible, non-Newtonian fluid with strain-rate- and temperature-dependent viscosity (Hutter 1983). For an ice sheet occupying a three-dimensional volume Ω, with top surface Γ_t and basal surface Γ_b, the forward problem can be stated as a non-linear Stokes problem:

$$-\boldsymbol{\nabla}\cdot[\mu_{\text{eff}}(\boldsymbol{u})(\boldsymbol{\nabla}\boldsymbol{u}+\boldsymbol{\nabla}\boldsymbol{u}^{\mathsf{T}})-\boldsymbol{I}p]=\rho\boldsymbol{g} \quad \text{in } \Omega, \tag{6.1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \quad \text{in } \Omega, \tag{6.2}$$

$$\boldsymbol{\sigma}\boldsymbol{n}=\boldsymbol{0} \quad \text{on } \Gamma_t, \tag{6.3}$$

$$\boldsymbol{u}\cdot\boldsymbol{n}=0,\ \boldsymbol{T}\boldsymbol{\sigma}\boldsymbol{n}+\exp(m)\boldsymbol{T}\boldsymbol{u}=\boldsymbol{0} \quad \text{on } \Gamma_b, \tag{6.4}$$

where $\boldsymbol{u}$, p, ρ and μ_{eff} are the velocity, pressure, mass density and effective viscosity of the ice, respectively, $\boldsymbol{g}$ is the gravitational acceleration, $\boldsymbol{n}$ is the outward unit normal, $\boldsymbol{T} := \boldsymbol{I} - \boldsymbol{n}\otimes\boldsymbol{n}$ is the projection operator onto the tangent plane on the boundary, and m is the log basal friction parameter. The ice rheology is shear-thinning with stress tensor $\boldsymbol{\sigma}$ related to the strain rate tensor $\dot{\boldsymbol{\varepsilon}} := \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{u}+\boldsymbol{\nabla}\boldsymbol{u}^{\mathsf{T}})$ via the constitutive law

$$\boldsymbol{\sigma} := 2\mu_{\text{eff}}(\boldsymbol{u})\dot{\boldsymbol{\varepsilon}} - \boldsymbol{I}p, \quad \text{with } \mu_{\text{eff}}(\boldsymbol{u}) := \frac{1}{2}A^{-1/n}\left(\frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\varepsilon}}^2)\right)^{(1-n)/(2n)},$$

where $n \geq 1$ is the Glen's flow law exponent, typically taken as 3, and A is a flow rate factor accounting for temperature-dependence of the ice viscosity, here represented by the Paterson–Budd relationship (Paterson and Budd 1982); μ_{eff} is regularized to prevent the vanishing of the viscosity when $\dot{\boldsymbol{\varepsilon}} = 0$. In this model, (6.1) represents the balance of linear momentum, (6.2) the balance of mass, (6.3) the traction-free condition on the top surface and (6.4) the basal boundary conditions. On the basal boundary, we assume zero normal velocity, and a Robin condition in the tangential direction. This Robin condition is a phenomenological relationship that mediates between no-slip ($\exp(m) \approx \infty$) and free-slip ($\exp(m) \approx 0$) boundary conditions. Thus the distribution of $\exp(m)$ along the basal surface can represent any intermediate point between these two extremes. The challenge is that this boundary is not directly observable; instead we must infer the basal friction from satellite observations of the surface velocity.

To solve the inverse problem, we express the negative log posterior as

$$\phi(m) := \frac{1}{2}\|\mathcal{C}_{\text{noise}}^{-1/2}(\mathcal{B}\boldsymbol{u}(m) - \boldsymbol{d})\|^2_{L^2(\Gamma_t)} + \frac{1}{2}\|\mathcal{A}(m - m_{\text{pr}})\|^2_{L^2(\Gamma_b)}, \tag{6.5}$$

where $\boldsymbol{u}$ is a solution of the non-linear Stokes forward problem (6.1)–(6.4) for a given log basal friction m, $\mathcal{C}_{\text{noise}}$ is the noise covariance operator, $\mathcal{B}$ is the observation operator that restricts the velocity to the top surface of the ice sheet, $\boldsymbol{d}$ is the InSAR-image-based top surface velocity, $\mathcal{C}_{\text{pr}} := \mathcal{A}^{-2}$ is the prior covariance operator and m_{pr} is the prior mean of the log basal friction. The differential operator $\mathcal{A}$ is defined by

$$\mathcal{A}(m) := \begin{cases} -\gamma\Delta_\Gamma m + \delta m & \text{in } \Gamma_b, \\ (\gamma\boldsymbol{\nabla}_\Gamma m)\cdot\boldsymbol{\nu} & \text{on } \partial\Gamma_b, \end{cases}$$

where Δ_Γ is the Laplace–Beltrami operator on the basal boundary, the roles of $\gamma > 0$ and $\delta > 0$ are described just below (4.3), $\boldsymbol{\nabla}_\Gamma$ is the tangential gradient and $\boldsymbol{\nu}$ is the outward unit normal on $\partial\Gamma_b$, the boundary of Γ_b. This choice of prior is sufficiently smoothing to ensure that $\mathcal{C}_{\text{pr}}$ is trace-class, leading to bounded pointwise variance and a well-posed infinite-dimensional Bayesian inverse problem (Stuart 2010).

Expressions for the adjoint-based gradient, and incremental forward/adjoint-based Hessian action, can be found in Isaac *et al.* (2015). To solve this inverse problem numerically, all field quantities and operators are discretized by Galerkin finite elements, resulting in 3 785 889 unknowns for each state-like field (state, adjoint, incremental state, incremental adjoint) and 409 545 uncertain basal friction parameters. A prior mean of zero and covariance parameters $\gamma = 10$, $\delta = 10^{-5}$ are chosen. A diagonal noise covariance matrix with noise given by 10% of the observed signal is prescribed. This results in the finite-dimensional posterior

$$\pi_{\text{post}}(\mathbf{m}) \propto \exp(-\boldsymbol{\Phi}(\mathbf{m})). \tag{6.6}$$

6.2. *The MAP point*

Our first task is to find the maximum *a posteriori* (MAP) point,

$$\mathbf{m}_{\text{map}} := \arg\min_{\mathbf{m}\in\mathbb{R}^{n_m}} \boldsymbol{\Phi}(\mathbf{m}). \tag{6.7}$$

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Table 6.1. Performance and scalability of inexact Newton-CG for Pine Island Glacier inverse problem. The columns report the number of state variable unknowns (#sdof), basal friction parameter unknowns (#pdof), Newton iterations (#N), total and average (per Newton iteration) number of CG iterations (#CG, avgCG) and total number of linear(ized) Stokes solves (from forward, adjoint and incremental forward and adjoint problems) (#Stokes). The iterations are terminated when the norm of the gradient is decreased by a factor of 10^5. The superlinear choice of the forcing term is made, *i.e.* $\eta_k = \|\mathbf{g}_k\|^{0.5}$.

#sdof	#pdof	#N	#CG	avgCG	#Stokes
95 796	10 371	42	2718	65	7031
233 834	25 295	39	2342	60	6440
848 850	91 787	39	2577	66	6856
3 372 707	364 649	39	2211	57	6193
22 570 303	1 456 225	40	1923	48	5376

We employ the inexact Newton-CG method of Section 3 to solve the optimization problem (6.7), which is equivalent to solving a regularized inverse problem. We begin by studying the scalability of this method for an inverse problem posed on a portion of the Antarctic ice sheet, the region around the Pine Island Glacier. This scalability test is run for synthetic data (contaminated by 10% additive noise). Table 6.1 reports the computational effort as the parameter mesh is refined (along with the mesh representing the state-like quantities). The numbers of Newton iterations, total CG iterations and average CG iterations per Newton iteration are shown in the table. As can be seen by the essentially constant number of iterations as the parameter dimension increases from 10K to 1.5M, the inexact Newton-CG method results in dimension-independent convergence. Since the top surface mesh is refined along with the volume and parameter meshes, the InSAR image is represented on successively finer meshes as well. Thus the results indicate scaling independent of the data dimension, in addition to the parameter dimension.

Note that despite the extrinsic dimension of the problem (up to 1.5 million parameters), on average just ~ 60 CG iterations are required per Newton iteration. This is a consequence of the relatively small intrinsic 'information' dimension of the problem, *i.e.* the number of modes in parameter space that are informed by the data, filtered through the prior. This is also a consequence of the inexactness of the method that provides early termination of CG in the pre-asymptotic phase (the superlinear choice of the forcing term, $\eta_k = O(\|\mathbf{g}_k\|^{1/2})$, is taken).

Having demonstrated scalability, we next proceed to scaling up the inversion to the full Antarctic ice sheet with its 410K basal friction parameters, using real InSAR data from 900 satellite tracks over the period 2007–2009 (Rignot, Mouginot

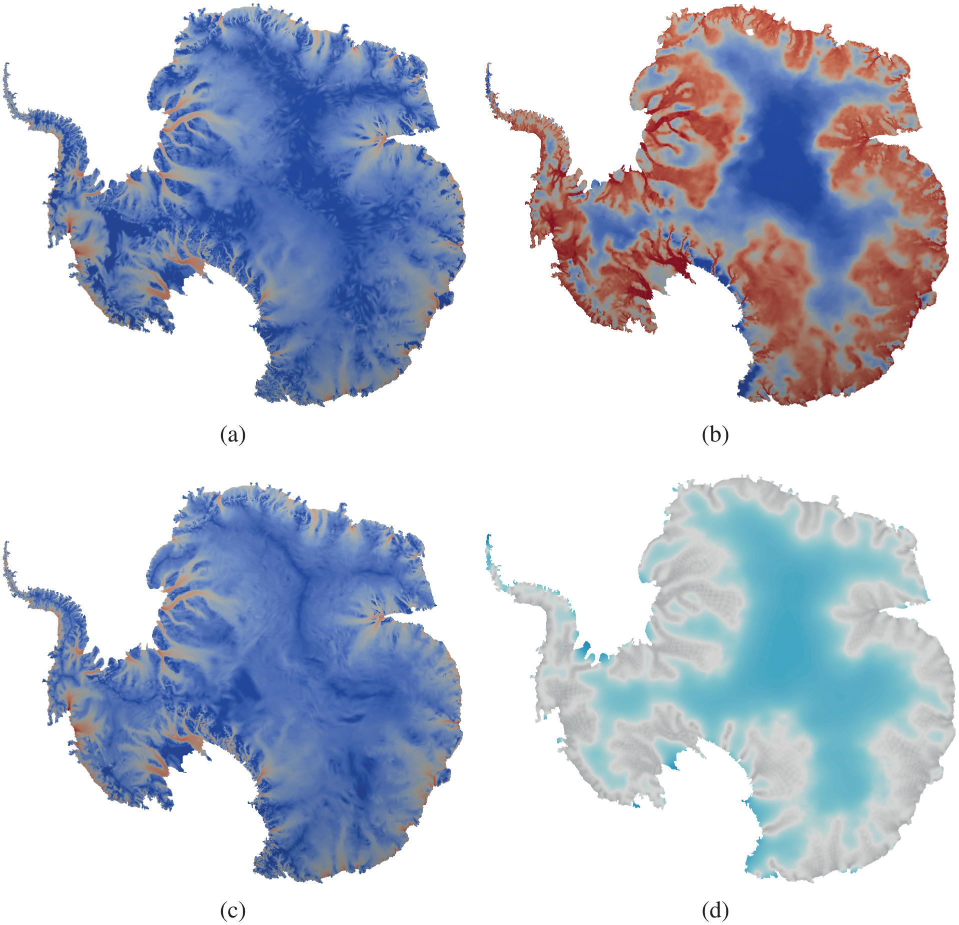


Figure 6.1. Inference of basal friction in Antarctic ice sheet. (a) InSAR observed velocity field on top surface, ranging from several m/yr (dark blue) to several km/yr (bright red). (b) MAP point of inferred basal friction field, ranging from an essentially no-slip (dark blue) to an essentially free slip (dark red) boundary condition, representing nine orders of magnitude variation in basal friction. (c) Reconstructed surface velocity. (d) Laplace approximation-based standard deviation of basal friction, with two orders of magnitude variation from grey (low) to blue (high). (Adapted from Isaac *et al.* 2015, with permission. Copyright © 2015 Elsevier.)

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

and Scheuchl 2011). To find the MAP point, we minimize $\Phi(\mathbf{m})$ in (6.7). We terminate after a 1000-fold reduction in the norm of the gradient of Φ, which requires 213 Newton iterations – a reflection of the strong non-linearity of the inverse problem. Each Newton iteration required an average of 239 CG iterations. We shall see below (in Figure 6.2) that the prior-preconditioned data misfit Hessian has ∼5000 data-informed eigenmodes. So 239 CG iterations is about as few as one can hope for with regularization preconditioning. Nevertheless, finding the MAP point entails a total of 107 578 linear(ized) Stokes solves (executed while solving the non-linear forward problem at each Newton iteration and during the line search, as well during adjoint, incremental forward and incremental adjoint solves). This is quite substantial. Since a typical (non-linear Stokes) forward solve might require ∼10 Stokes solves, the inverse problem is about 10^4 times as costly as the forward problem. This is the price of inversion. Only through the use of a parallel computer for the Stokes solves (in this case 1024 processor cores), could the MAP point be computed in acceptable time (and even then the regularization parameter was chosen based on the smaller Pine Island problem). So, while the inexact Newton-CG method is observed to scale in an ideal manner, and the number of CG iterations per Newton iteration is relatively small ($239 \ll 5000 \ll 409\,545$), improving on the regularization preconditioning described in Section 3 is an active area of research (see the discussion at the end of Section 3).

Figure 6.1 depicts the results of inversion. Figure 6.1(a) is the InSAR observed velocity field on the top surface of the Antarctic ice sheet, which ranges from a few metres per year in the interior of the ice sheet (dark blue) to several kilometres per year in the fast-flowing ice streams (dark red). Figure 6.1(b) shows the distribution of the basal friction at the MAP point, ranging from near no-slip in the dark blue regions to near free-slip in the dark red, which implies essentially no resistance to basal sliding. Figure 6.1(c) depicts the reconstructed ice surface velocity (computed by forward-propagating the inferred basal friction to the surface velocity via solution of the forward problem), showing excellent fits to the data in ice stream regions, which play a critical role in the sea level rise, with somewhat lower fidelity to the data in the interior (though still within the range of the InSAR noise level). While the MAP solution allows us to find a basal friction field that results in a good fit to the data, it does not characterize uncertainty in the inversion, which is the subject of the next section.

6.3. *Computing the Laplace approximation of the posterior*

In this section we construct the Laplace approximation of the posterior, made tractable by the low-rank-based representation (4.8). We begin by computing the dominant eigenvalues of the generalized eigenvalue problem (4.7), which identify the data-informed modes of the basal friction field. A randomized generalized eigensolver is used (Villa *et al.* 2021, Isaac *et al.* 2015). Figure 6.2 plots the dominant spectrum of this eigenvalue problem for two meshes: the baseline, with 410K

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

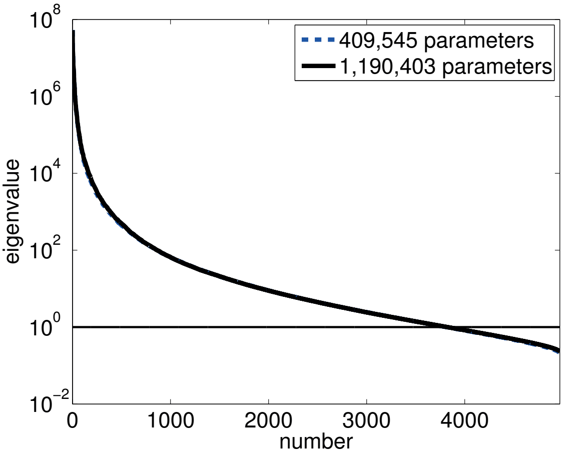


Figure 6.2. Spectrum of the generalized eigenvalue problem (4.7) on successively refined meshes with 410K and 1.19M basal friction parameters, capturing 5000 dominant data-informed modes. Spectrum decays like j^{-3}. (Adapted from Isaac *et al.* 2015, with permission. Copyright © 2015 Elsevier.)

basal friction parameters, and a refined mesh, with 1.19M parameters. The two spectra lie on top of each other, suggesting dimension-independence of the data-informed modes. The eigenvalues are observed to decay like j^{-3}. We truncate the eigensolver at ~5000 modes, at which point the eigenvalues have decayed by nine orders of magnitude. Thus we retain just 0.42%–1.2% of the modes, depending on the parameter dimension. This corresponds to an expected information gain in the 5000th mode that is ~0.5% of that in the first mode $(\ln(\lambda_1 + 1)/\ln(\lambda_{5000} + 1))$. The number of required Hessian actions is ~5000, which corresponds to ~10 000 linear(ized) Stokes solves in the incremental forward and incremental adjoint problems. Thus finding the MAP point (107 578 Stokes solves) is an order of magnitude more costly than constructing the low-rank, Laplace-based approximation of the posterior covariance matrix (10 000 Stokes solves). A parameter-to-observable map with a lower degree of non-linearity combined with more informative data might yield the opposite result.

Figure 6.3 displays select eigenfunctions (namely 1, 7, 10, 100, 1000, 4000) corresponding to dominant eigenvalues of Figure 6.2. As can be seen, these dominant eigenfunctions start off very smooth, and become more oscillatory as the eigenvalues decrease in magnitude, and as a result the information gain in the eigenfunctions decreases. The decreasing length scales in the eigenfunctions corresponding to smaller eigenvalues reflect the smaller feature sizes that can be inferred from the data in that eigenfunction direction in parameter space. But there are diminishing returns: by the 5000th eigenfunction, the information provided by the (noisy) data is dominated by the prior information, and so the inference of these

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

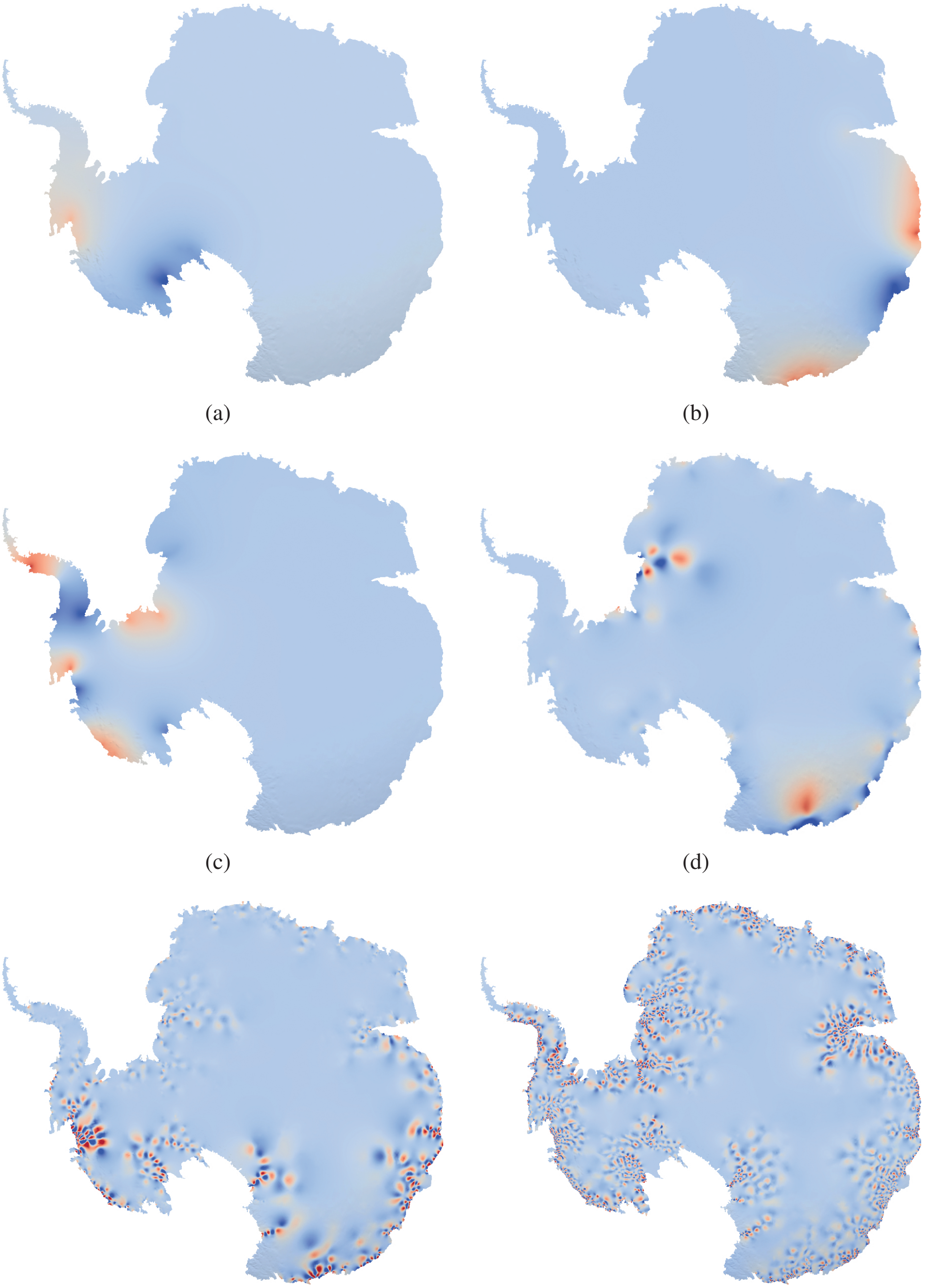


Figure 6.3. Eigenfunctions 1 (a), 7 (b), 10 (c), 100 (d), 1000 (e) and 4000 (f) of the generalized eigenvalue problem (4.7). (Adapted from Isaac *et al.* 2015, with permission. Copyright © 2015 Elsevier.)

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

smallest scales becomes unreliable and the spectrum is truncated in favour of the prior.

Note that the dominant eigenfunctions are (relatively) smooth, and thus once the mesh is sufficiently fine to resolve them, further mesh refinement (and thus a higher-dimensional parameter space) does not permit the extraction of additional information from the data. Hence the inverse problem is effectively finite-dimensional, despite the infinite-dimensional parameter space, and the effectively infinite-dimensional data space (the satellite images are represented on a much finer grid than the velocity field).

The eigenfunctions in Figure 6.3 are seen to be highly heterogeneous and localized; the smallest basal friction length scales occur beneath the fast-flowing ice streams nearer to the coast, with intermediate length scales in the catchments of these ice streams, and the largest length scales in the interior of the continent. The eigenfunctions can also be seen to be highly anisotropic: zooming into the ice stream regions in the electronic version of the figure, one can see oscillations in the basal friction developing in the streamwise directions of the ice streams in the higher modes, while the spanwise directions remain smooth. In contrast, the eigenfunctions are essentially flat in the interior for most of the dominant eigenvalues, and only in the highest of the dominant modes does one begin to see their support penetrating into the interior of the ice sheet. Thus most of the information gained from the data about the basal friction, as indicated by the dominant eigenfunctions, is restricted to the coastal regions, and particularly along the base of fast-flowing ice streams.

These highly heterogeneous, localized and anisotropic dominant eigenfunctions suggest that a Karhunen–Loève approximation – which retains the smoothest eigenfunctions of the prior covariance $\mathcal{C}_{\text{pr}}$ – would be a poor choice for a parameter reduction strategy, since these modes amount to the smoothest modes of the (homogeneous, isotropic) Laplacian in the usual case of constant γ and δ. In contrast, the dominant eigenfunctions of the generalized eigenvalue problem (4.7) are informed not just by prior information but importantly also by the parameter-to-observable map. They are the modes that are both observable and consistent with prior knowledge, with the balance between the two mediated by the noise in the data, the strength of the prior knowledge, and the decay rate of the singular values of the linearized map (which in turn depends on the observation operator, how the parameter field enters into the forward PDE, and the smoothing nature of the forward problem). The generalized eigenvalue problem (4.7) combines all of these factors together, and its dominant eigenfunctions describe the intrinsic low-dimensional subspace in which we learn from the data.

Finally, Figure 6.1(d) displays the posterior standard deviation of the basal friction field, which is taken as the diagonal of the posterior covariance based on the Laplace approximation. Fully exploring the $O(10^5–10^6)$-dimensional posterior using state-of-the-art MCMC methods is prohibitive for this problem: the nonlinear Stokes forward problem alone requires ~1 minute on 1024 cores, and many

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

millions of forward solves would be required. Instead, the Laplace approximation is constructed here at the cost of $\sim 10^4$ forward solves. Experience with a model two-dimensional ice sheet inverse problem suggests that this approximation may be a reasonable one (Petra *et al.* 2014). The standard deviation field in Figure 6.1 indicates two orders of magnitude variation in σ, ranging from high values in the interior of the ice sheet (blue) to the lowest values in ice streams (grey) and intermediate values in their catchments (white). Thus, while the inference of the basal friction in the interior is unreliable, significantly higher credibility can be assigned to the inference of the basal friction in the fast-flowing ice streams and their catchment regions nearer to the coast – which are the regions in which the basal friction has the greatest influence on ice mass flux into the ocean, and thus potential sea level rise (Joughin, Alley and Holland 2012).

In summary, this case study of a large-scale, complex ice sheet flow inverse problem has demonstrated that the combination of an inexact Newton-CG method for finding the MAP point – providing dimension-independent Newton iterations – with an effectively low-dimensional manifold in which the data inform the model parameters – providing dimension-independence of both CG iterations and posterior covariance construction – leads to a scalable method that can tractably infer the $O(10^5\text{–}10^6)$-dimensional basal friction field and quantify its uncertainty under the Laplace approximation. Nevertheless, despite the resulting ability to solve the inverse problem at a cost, measured in forward problem solves, that scales independent of the parameter and data dimensions, the number of (equivalent) non-linear Stokes forward solves, $\sim 10^4$, remains large. This is due to both the non-linearity of the inverse problem and the fact that the number of data-informed modes in parameter space, while just a fraction of the overall parameter dimension ($\sim 1\%$), remains large (5000) in absolute terms. This motivates research in methods that can tame the non-linearities, such as non-linear preconditioning (*e.g.* Cai and Keyes 2002) and full space PDE-constrained optimization methods (*e.g.* Biros and Ghattas 2005*a*,*b*), as well as methods that can better tame the Hessian in the data-informed regime, such as more effective preconditioners for, and more compact representations of, the Hessian (see the references at the end of Section 3).

PART THREE
Model reduction

In Part 3 we approach the task of exploitation of system structure from a complementary viewpoint and consider the task of deriving a computationally efficient surrogate model. Many different surrogate modelling methods exist. For decades the engineering and science communities have used surrogate models – also called metamodels and emulators – to reduce computational burden in applications such as design, control, optimization and uncertainty quantification. Examples

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

include polynomial response surfaces (Kaufman *et al.* 1996, Giunta *et al.* 1997, Venter, Haftka and Starnes 1998, Eldred, Giunta and Collis 2004), radial basis functions (Wild, Regis and Shoemaker 2008), Gaussian process models (Kennedy and O'Hagan 2001, Rasmussen and Williams 2006), Kriging models (Simpson, Mauery, Korte and Mistree 2001) and stochastic spectral approximations (Ghanem and Spanos 1991, Xiu and Karniadakis 2002, Marzouk, Najm and Rahn 2007). Surrogate models are receiving increased attention with the growing popularity of machine learning.

Here we focus on reduced-order models as one specific class of surrogate models. Reduced-order modelling differs from other surrogate modelling approaches in that it explicitly appeals to the structure of the physical problem being modelled. Just as in the inverse problem formulations of Part 2, this structure manifests through the governing equations and the mathematical properties of their operators. Reduced-order modelling embeds this structure in the constructed surrogate models.

We begin Part 3 by presenting the projection framework of model reduction in Section 7. We discuss its mathematical properties as a general framework for deriving surrogate models and present reduced models derived using proper orthogonal decomposition combined with Galerkin projection for time-dependent and parametrized systems. Then in Section 8 we show how this classical projection viewpoint paves the way for a class of structure-preserving non-intrusive surrogate modelling approaches, and we present Operator Inference – where the task of deriving a reduced model is viewed as an inverse problem in which we infer the reduced operators – as a bridge between model reduction and machine learning. We close Part 3 with a discussion of the state of the art and open challenges in non-linear model reduction in Section 9.

7. Projection-based model reduction

This section presents the general mathematical framework for low-dimensional approximation of continuous and semi-discrete systems via projection onto a low-dimensional subspace. As a concrete example, we present computation of the low-dimensional subspace using the proper orthogonal decomposition (POD) method. We discuss approaches for embedding parametric dependence into projection-based reduced models, which is essential for applications in inverse problems, optimization and uncertainty quantification where the reduced model will be invoked over a range of parameter values. We close this section by reminding the reader that the availability of error estimates for a large class of problems is one of the advantages of projection-based reduced models over other data-driven surrogate models.

7.1. Low-dimensional approximation via projection

Projection is broadly applicable to linear and non-linear systems, but in order to highlight its structure-exploiting properties, let us begin with a linear system.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Consider the linear PDE defined on the domain Ω and time horizon $(0, t_f)$,

$$\frac{\partial u}{\partial t} = \mathcal{A}(u) \quad \text{in } \Omega \times (0, t_f), \tag{7.1}$$

with appropriate boundary and initial conditions, and where $u(x, t) \in \mathcal{U}$ is the state at spatial location x and time t, and $\mathcal{A}\colon \mathcal{U} \to \mathcal{U}^*$ is the linear PDE operator with $\mathcal{U}^*$ the dual of $\mathcal{U}$. The projection-based reduced model of (7.1) is defined by restricting the state to lie in a rank r subspace $\mathcal{U}_r \subset \mathcal{U}$. Define r basis vectors $v_1(x), \dots, v_r(x)$ that span $\mathcal{U}_r$ and that form an orthonormal set, *i.e.* $\langle v_i, v_j \rangle = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta and $\langle \cdot \rangle$ the appropriate inner product or duality pairing. Then

$$u(x, t) \approx \sum_{j=1}^{r} v_j(x) \hat{u}_j(t), \tag{7.2}$$

where $\hat{u}_j$, $j = 1, \dots, r$ are the reduced model's coefficients of expansion in the basis v_j. Substituting the approximation (7.2) into the governing equation (7.1) yields the residual

$$r(x, t) = \sum_{j=1}^{r} v_j \frac{\mathrm{d}\hat{u}_j}{\mathrm{d}t} - \sum_{j=1}^{r} \mathcal{A}(v_j) \hat{u}_j. \tag{7.3}$$

We will use a Galerkin projection to define the reduced model (noting that Petrov–Galerkin projection is also possible and is desirable in some settings), meaning that we impose the condition $\langle r, v_i \rangle = 0$, $i = 1, \dots, r$. This yields the reduced model

$$\frac{\mathrm{d}\hat{u}_i}{\mathrm{d}t} = \sum_{j=1}^{r} \hat{\mathcal{A}}_{ij} \hat{u}_j, \quad i = 1, \dots, r, \tag{7.4}$$

where $\hat{\mathcal{A}}_{ij} = \langle v_i, \mathcal{A}(v_j) \rangle$ is the reduced linear operator, which can be precomputed once the basis is defined.

The preservation of linear structure in the reduced model can be seen by comparing (7.1) and (7.4). This structure preservation is not just limited to linear systems. Consider the linear-quadratic PDE

$$\frac{\partial u}{\partial t} = \mathcal{A}(u) + \mathcal{H}(u, u) \quad \text{in } \Omega \times (0, t_f), \tag{7.5}$$

where we now introduce the bilinear operator $\mathcal{H}\colon \mathcal{U} \times \mathcal{U} \to \mathcal{U}^*$ which represents quadratic terms in the PDE such as u^2 and $u(\partial u / \partial x)$. As in the linear case, we restrict the state to lie in a rank r subspace $\mathcal{U}_r \subset \mathcal{U}$ and employ Galerkin projection. This leads to the reduced model

$$\frac{\mathrm{d}\hat{u}_i}{\mathrm{d}t} = \sum_{j=1}^{r} \hat{\mathcal{A}}_{ij} \hat{u}_j + \sum_{j=1}^{r} \sum_{k=1}^{r} \hat{\mathcal{H}}_{ijk} \hat{u}_j \hat{u}_k, \quad i = 1, \dots, r, \tag{7.6}$$

where $\hat{\mathcal{H}}_{ijk} = \langle v_i, \mathcal{H}(v_j, v_k) \rangle$ is the reduced quadratic operator. Comparing (7.5) and (7.6), again we see the preservation of the linear-quadratic structure in the reduced model. Furthermore, the reduced linear and quadratic operators can be precomputed, which will lead to efficient solution of the reduced model. Following similar steps for systems with higher-order polynomial terms shows that structure preservation holds similarly for cubic and higher-order polynomial terms.

This preservation of polynomial structure has been well known for decades in the projection-based model reduction community. Indeed, many of the earliest reduced-order modelling papers exploited this property. For example, Graham, Peraire and Tang (1999) exploited the quadratic structure of the incompressible Navier–Stokes equations to derive a POD reduced model. Even though well known, the projection's preservation of polynomial structure has arguably become under-appreciated in the face of more recent work for handling general non-linear terms. Yet the preservation of structure is fundamental, because this structure directly reflects the underlying physics – such as the linear form of a diffusion term or the quadratic form of the convective transport of momentum. And so while general-purpose non-linear approximations are needed, and have been developed as we discuss in Section 9, the structure of the underlying governing physics equations should not be overlooked when seeking a physics-informed approximation. We return to this discussion in Section 9.3.

7.2. General projection framework for semi-discrete systems

We now present the general projection framework in the ODE setting, which directly parallels the PDE setting just presented. Consider the linear system of ODEs

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}\mathbf{u}, \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{7.7}$$

and the linear-quadratic system of ODEs

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}\mathbf{u} + \mathbf{H}(\mathbf{u}\otimes\mathbf{u}), \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{7.8}$$

where $\mathbf{u}(t) \in \mathbb{R}^n$ is the state at time t, with dimension n, and $\mathbf{u}_0 \in \mathbb{R}^n$ is the specified initial condition. $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the linear ODE operator and $\mathbf{H} \in \mathbb{R}^{n^2\times n}$ is the quadratic ODE operator. Here $\otimes$ denotes the Kronecker product (following the notation in Kolda and Bader 2009), which for an n-dimensional column vector $\mathbf{u} = [u_1, u_2, \dots, u_n]^\mathsf{T}$ is given by

$$\mathbf{u}\otimes\mathbf{u} = [u_1^2, u_1u_2, \dots, u_1u_n, u_2u_1, u_2^2, \dots, u_2u_n, \dots, u_n^2] \in \mathbb{R}^{n^2}.$$

In some applications, the governing equations manifest directly as systems of ODEs. In many cases of interest, the systems (7.7) and (7.8) result from spatial discretization of PDEs of the general form (7.1) and (7.5), respectively. In such cases, the state dimension is large, typically ranging from $n \sim 10^3$ to $n \sim 10^9$.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

To form reduced models of (7.7) and (7.8), the state is approximated in an r-dimensional orthonormal basis,

$$\mathbf{u}(t) \approx \sum_{j=1}^{r} \mathbf{v}_j \hat{u}_j(t) = \mathbf{V}\hat{\mathbf{u}}(t), \tag{7.9}$$

where $\mathbf{v}_j \in \mathbb{R}^n$ is the jth basis vector and $\mathbf{V} \in \mathbb{R}^{n\times r}$ is the basis matrix that has $\mathbf{v}_j$ as its jth column. The column space of $\mathbf{V}$ defines an r-dimensional subspace of the full state space $\mathbb{R}^n$. The reduced state $\hat{\mathbf{u}}(t) \in \mathbb{R}^r$ represents the coefficients of expansion in the basis $\mathbf{V}$. Using a Galerkin projection, we obtain the reduced model of (7.7) as

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \hat{\mathbf{A}}\hat{\mathbf{u}}, \quad \hat{\mathbf{u}}(0) = \mathbf{V}^{\mathsf{T}}\mathbf{u}_0, \tag{7.10}$$

and the reduced model of (7.8) as

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \hat{\mathbf{A}}\hat{\mathbf{u}} + \hat{\mathbf{H}}(\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}), \quad \hat{\mathbf{u}}(0) = \mathbf{V}^{\mathsf{T}}\mathbf{u}_0, \tag{7.11}$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{H}} \in \mathbb{R}^{r\times r^2}$ are the reduced-order operators, defined by projection of the full-order operators onto the subspace defined by $\mathbf{V}$:

$$\hat{\mathbf{A}} = \mathbf{V}^{\mathsf{T}}\mathbf{A}\mathbf{V}, \tag{7.12}$$

$$\hat{\mathbf{H}} = \mathbf{V}^{\mathsf{T}}\mathbf{H}(\mathbf{V} \odot \mathbf{V}). \tag{7.13}$$

Here $\odot$ denotes the Khatri–Rao product of two matrices (which is also known as the column-wise Kronecker product; see Kolda and Bader 2009), which for a matrix $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_r] \in \mathbb{R}^{n\times r}$ is given by

$$\mathbf{V} \odot \mathbf{V} = [\mathbf{v}_1 \otimes \mathbf{v}_1\ \mathbf{v}_2 \otimes \mathbf{v}_2\ \cdots\ \mathbf{v}_r \otimes \mathbf{v}_r] \in \mathbb{R}^{n^2\times k}.$$

The reduced-order operators (7.12) and (7.13) can be precomputed, so that solution of the reduced models (7.10) and (7.11) does not scale with the full-order dimension n.

The ODE setting clearly shows the preservation of linear structure in the reduced model (7.10) of (7.7) and quadratic structure in the reduced model (7.11) of (7.8). The parallel definition of the reduced models between continuous and semi-discrete setting can also be seen, for example, in the analogous definitions of $\hat{\mathcal{A}}_{ij}$ in (7.4) and $\hat{\mathbf{A}}$ in (7.12), and of $\hat{\mathcal{H}}_{ijk}$ in (7.6) and $\hat{\mathbf{H}}$ in (7.13). For the methods and algorithms presented in the remainder of Part 3, we will work with the semi-discrete ODE form because its linear algebraic form lends a more compact presentation. However, it is important to note that the approaches apply to systems arising from PDEs; indeed, large-scale simulations of PDEs are the most common target for model reduction.

7.3. Proper orthogonal decomposition

Proper orthogonal decomposition (POD) (Lumley 1967, Sirovich 1987, Berkooz, Holmes and Lumley 1993), which also goes by the names Karhunen–Loève expansions in stochastic process modelling (Loève 1955, Kosambi 1943), principal component analysis (PCA) in statistical analysis (Hotelling 1933, Jolliffe 2005) and empirical orthogonal eigenfunctions in atmospheric modelling (North, Bell, Cahalan and Moeng 1982), is one method to define the basis matrix $\mathbf{V}$. The POD basis is computed empirically from training data in the form of system solutions, and thus applies to both linear and non-linear systems. POD can be applied over both time-varying and parametrically varying conditions. To compute the POD basis, consider a set of n_s snapshots, $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$, which are state solutions computed at different instants in time and/or different parameter values, with $\mathbf{u}_j \in \mathbb{R}^n$ denoting the jth snapshot. Define the snapshot matrix $\mathbf{U} \in \mathbb{R}^{n \times n_s}$ whose jth column is the snapshot $\mathbf{u}_j$. The singular values of $\mathbf{U}$ are denoted $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_s} \geq 0$. The POD basis vectors are the left singular vectors of $\mathbf{U}$ corresponding to the r largest singular values. This yields an orthonormal basis, *i.e.* $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$.

The POD basis minimizes the least-squares error of snapshot reconstruction. That is, over all r-dimensional orthonormal basis choices $\tilde{\mathbf{V}}$, the POD basis is given by

$$\mathbf{V} = \underset{\substack{\tilde{\mathbf{V}} \in \mathbb{R}^{n \times r} \\ \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}} = \mathbf{I}}}{\arg\min} \|\mathbf{U} - \tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\mathbf{U}\|_F^2 = \underset{\substack{\tilde{\mathbf{V}} \in \mathbb{R}^{n \times r} \\ \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}} = \mathbf{I}}}{\arg\min} \sum_{i=1}^{n_s} \|\mathbf{u}_i - \tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\mathbf{u}_i\|_2^2. \tag{7.14}$$

The familiar singular value decomposition result yields the snapshot reconstruction error as being the sum of the squares of the singular values corresponding to those left singular vectors not included in the POD basis,

$$\|\mathbf{U} - \mathbf{V}\mathbf{V}^\mathsf{T}\mathbf{U}\|_F^2 = \sum_{i=1}^{n_s} \|\mathbf{u}_i - \mathbf{V}\mathbf{V}^\mathsf{T}\mathbf{u}_i\|_2^2 = \sum_{i=r+1}^{n_s} \sigma_i^2. \tag{7.15}$$

The size of the POD basis is thus typically chosen using the singular values as a guide. A typical approach is to choose r so that

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} > \kappa, \tag{7.16}$$

where κ is a user-specified tolerance, often taken to be 99.9% or greater (see *e.g.* Algorithm 1). The left-hand side of (7.16) is often referred to as the 'relative cumulative energy' captured by the r POD modes.

The performance of POD reduced models depends heavily on the choice of the snapshots. In any given setting, the snapshots should be representative of the conditions over which the reduced model is expected to issue predictions. For time-dependent systems, the snapshots are typically generated by sampling trajectories generated by different initial conditions and/or different forcing scenarios. In the case of a parametrized system (discussed further in Section 7.4), the snapshots must be generated over different parameter values. For problems where the varying condition (parameter, initial condition, forcing) is represented with low dimension and the variation in dynamics is relatively benign, it is typically tractable to use

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Algorithm 1 Computing the POD basis.

```
Inputs: snapshots $\mathbf{U} \in \mathbb{R}^{n \times n_s}$, POD cumulative energy threshold $\kappa$
1: Centre and/or scale snapshots {Especially important if each snapshot contains multiple physical quantities}
2: Compute the SVD of $\mathbf{U}$
3: $r \leftarrow$ choose $r$ such that $(\sum_{i=1}^{r} \sigma_i^2)/(\sum_{i=1}^{n_s} \sigma_i^2) > \kappa$ {Determine the dimension of the POD basis}
4: $\mathbf{V} \leftarrow$ the $r$ leading left singular vectors of $\mathbf{U}$ {Compute the POD basis}
5: return $\mathbf{V}, r$
```

brute-force sampling (*e.g.* grid sampling) over the desired range of conditions. For higher-dimensional and/or more complex problems, it is essential to use a tailored snapshot selection strategy. For parametrized problems, the common approach is adaptive greedy sampling, discussed in Section 7.4. Kunisch and Volkwein (2010) considered optimal snapshot selection for non-parametric POD, where the time locations of snapshots are chosen to minimize the error between the POD solution and the trajectory of the original dynamical system. For initial condition problems, Bashir *et al.* (2008) showed that the eigenvectors of the Hessian yield an optimal POD sampling strategy.

Another practical consideration is that of scaling. In practice, one must pay close attention to the scaling of the snapshots in order to obtain an adequate basis. Typically the snapshot set is centred (*i.e.* shifted to have zero mean). This centering may be combined with particular solutions to account for inhomogeneous boundary conditions and other imposed conditions (Swischuk, Mainini, Peherstorfer and Willcox 2019). Preprocessing of the snapshot set to address scaling and to extract particular solutions is often overlooked, but is an essential element of achieving accurate and efficient POD representations. Scaling is particularly important when the snapshots contain multiple physical variables (*e.g.* velocity, pressure, temperature, concentration).

Extended versions of POD expand the snapshot set to include adjoint solutions (Lall, Marsden and Glavaski 2002, Willcox and Peraire 2002), sensitivity information (Hinze and Volkwein 2008, Hay, Borggaard and Pelletier 2009) and time derivatives of the states (Kunisch and Volkwein 2001). Iliescu and Wang (2014) showed that including snapshots of the time derivatives when computing the POD basis (as proposed in Kunisch and Volkwein 2001) is needed in order to achieve an optimal convergence rate of the error of the reduced model with respect to the number of POD basis functions. The gappy POD is another variant of POD that deals with so-called gappy (incomplete/missing) data. The gappy POD was introduced by Everson and Sirovich (1995) for facial image reconstruction and was first applied to PDEs in the aerodynamic modelling examples of Bui-Thanh, Damodaran and Willcox (2004).

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Other methods exist to compute the basis $\mathbf{V}$ (see Benner, Gugercin and Willcox 2015 for a review), most notably the reduced basis method (which is similar to POD but with a prescribed adaptive approach for generating snapshots), rational interpolation methods and balanced truncation. We summarize here only POD, because it is the most common, flexible and easily applied of the methods, but note that in some settings there may be advantages to using the other basis computation methods. In particular, Antoulas, Beattie and Gugercin (2020) provided an in-depth presentation of the state of the art in rational interpolation methods, a powerful set of approaches with deep connections to system-theoretic approximation methods.

7.4. General projection framework for parametrized systems

Parametric model reduction considers the dependence of the governing equations on one or more parameters and reflects this dependence explicitly in the reduced model. This is important for design, uncertainty quantification, control and inversion applications, where the reduced model will be invoked over a range of parameter values. Benner *et al.* (2015) surveyed parametric model reduction methods in depth; here we discuss the two main challenges in creating a parametrized reduced model. The first challenge is to expose the parametric dependence in the reduced model in a way that leads to efficient computations. The second challenge is to ensure that the reduced-order representation is sufficiently rich to yield accurate predictions over the desired range of parameter values.

To see how the first challenge manifests, consider first the linear system with dependence on n_m parameters $\mathbf{m} \in \mathbb{R}^{n_m}$,

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}(\mathbf{m})\,\mathbf{u}, \quad \mathbf{u}(0) = \mathbf{u}_0. \tag{7.17}$$

This system might arise, for example, from semi-discretization of the parametrized PDE $\partial u/\partial t = \mathcal{A}(u; m)$, where m could be a spatial field, $m(x)$, or a finite-dimensional vector containing multiple parameters. The state $\mathbf{u}(t;\mathbf{m})$ is now a function of both time t and discretized parameters $\mathbf{m}$. Following the same steps as in Section 7.2, the Galerkin-reduced model of (7.17) is

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \mathbf{V}^{\mathsf{T}}\mathbf{A}(\mathbf{m})\mathbf{V}\,\hat{\mathbf{u}}, \quad \hat{\mathbf{u}}(0) = \mathbf{V}^{\mathsf{T}}\mathbf{u}_0. \tag{7.18}$$

Equation (7.18) reveals the first challenge of parametric model reduction: in the non-parametric case, the reduced operator $\hat{\mathbf{A}} = \mathbf{V}^{\mathsf{T}}\mathbf{A}\mathbf{V}$ could be precomputed, avoiding $O(n)$ computations in the solution of the reduced model; however, this is not true for the parametric case without additional treatment because of the dependence on $\mathbf{m}$.

As discussed in Benner *et al.* (2015), there are several strategies to address this challenge. In many cases, the governing equations admit an affine representation of the system's dependence on the parameters $\mathbf{m}$. In such cases, the parametrized linear system (7.17) may be written as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \sum_{i=1}^{n_a} \vartheta_i(\mathbf{m})\mathbf{A}_i\,\mathbf{u} \tag{7.19}$$

and a parametrized quadratic system as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \sum_{i=1}^{n_a} \vartheta_i(\mathbf{m})\mathbf{A}_i\,\mathbf{u} + \sum_{i=1}^{n_h} \varpi_i(\mathbf{m})\mathbf{H}_i(\mathbf{u}\otimes\mathbf{u}), \tag{7.20}$$

where there are n_a terms in the affine representation of $\mathbf{A}(\mathbf{m})$ and n_h terms in the affine representation of $\mathbf{H}(\mathbf{m})$. The scalar functions $\vartheta_i(\mathbf{m})$, $i = 1,\ldots,n_a$ and $\varpi_i(\mathbf{m})$, $i = 1,\ldots,n_h$ capture the parametric dependence, and the operators $\mathbf{A}_i \in \mathbb{R}^{n\times n}$, $i = 1,\ldots,n_a$ and $\mathbf{H}_i \in \mathbb{R}^{n^2\times n}$, $i = 1,\ldots,n_h$ are parameter-independent. The reduced models of (7.19) and (7.20) are respectively

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \sum_{i=1}^{n_a} \vartheta_i(\mathbf{m})\hat{\mathbf{A}}_i\,\hat{\mathbf{u}} \tag{7.21}$$

and

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \sum_{i=1}^{n_a} \vartheta_i(\mathbf{m})\hat{\mathbf{A}}_i\,\hat{\mathbf{u}} + \sum_{i=1}^{n_h} \varpi_i(\mathbf{m})\hat{\mathbf{H}}_i(\hat{\mathbf{u}}\otimes\hat{\mathbf{u}}), \tag{7.22}$$

where the reduced operators $\hat{\mathbf{A}}_i = \mathbf{V}^{\mathsf{T}}\mathbf{A}_i\mathbf{V} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{H}}_i = \mathbf{V}^{\mathsf{T}}\mathbf{H}_i(\mathbf{V}\odot\mathbf{V}) \in \mathbb{R}^{r\times r^2}$ can be precomputed, meaning that efficient solution of the parametric reduced model is recovered.

If the operators of the governing equations do not directly admit an affine parametric representation, it can be introduced approximately as in Drohmann, Haasdonk and Ohlberger (2012) for PDE operators using the empirical interpolation method (EIM) of Barrault, Maday, Nguyen and Patera (2004) or as in Benner *et al.* (2015) for ODE operators using the discrete empirical interpolation method (DEIM) of Chaturantabut and Sorensen (2010). An alternative strategy is to precompute multiple reduced models at selected parameter points and then interpolate these reduced models to issue predictions at a new parameter point (Amsallem, Cortial, Carlberg and Farhat 2009, Degroote, Vierendeels and Willcox 2010, Lohmann and Eid 2009, Panzer, Mohring, Eid and Lohmann 2010, Amsallem and Farhat 2011).

The second main challenge to be addressed in parametric model reduction is ensuring that the reduced-order basis is sufficiently rich so that its span allows the projected model to yield accurate predictions over the desired range of parameter values. One approach is to compute a global basis with span encompassing the desired range of dynamics. This can be achieved, for example, using a POD basis where the snapshots input to Algorithm 1 are sampled in both time and parameter space. For systems with just one or two parameters, it may be feasible to generate these snapshots in a single pass by dense sampling over the entire parameter space. For higher-dimensional parameters, it is essential to use some kind of intelligent

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Algorithm 2 Greedy sampling for a linear system with affine parametric dependence.

Inputs: Initial snapshot set $\mathbf{U} \in \mathbb{R}^{n \times n_s}$, POD cumulative energy threshold κ, desired reduced model accuracy level $\bar{\varepsilon}$, full-order model matrices $\{\mathbf{A}_i \in \mathbb{R}^{n \times n}\}_{i=1,\ldots,n_a}$ (or function computing action of $\mathbf{A}_i$ on a vector)

```
$\mathbf{V} \leftarrow$ compute the POD basis using Algorithm 1 {Initialize the POD basis}
$\hat{\mathbf{A}}_i \leftarrow \mathbf{V}^\mathsf{T}\mathbf{A}_i\mathbf{V},\ i = 1, \ldots, n_a$ {Initialize the reduced model (7.21)}
$\mathbf{m}^* \leftarrow$ the parameter value for which the reduced model has maximum indicated error {Worst-case prediction of the current reduced model using an error indicator or error estimator, found using grid search (low-dimensional parameters) or scalable optimization ($n_m > 3$)}
$\varepsilon \leftarrow$ reduced model error evaluated at $\mathbf{m}^*$
while $\varepsilon > \bar{\varepsilon}$ do
    $\mathbf{U} \leftarrow$ augment $\mathbf{U}$ with new snapshots evaluated at $\mathbf{m}^*$
    $\mathbf{V} \leftarrow$ compute the POD basis using Algorithm 1 {Update the POD basis (can be done with incremental updates)}
    $\hat{\mathbf{A}}_i \leftarrow \mathbf{V}^\mathsf{T}\mathbf{A}_i\mathbf{V},\ i = 1, \ldots, n_a$ {Update the reduced model}
    $\mathbf{m}^* \leftarrow$ the parameter value for which the reduced model has maximum indicated error
    $\varepsilon \leftarrow$ reduced model error evaluated at $\mathbf{m}^*$
end while
return $\mathbf{V}, \{\hat{\mathbf{A}}_i\}_{i=1,\ldots,n_a}$
```

and/or adaptive sampling strategy. For example, greedy sampling (Prud'homme *et al.* 2002, Rozza, Huynh and Patera 2008) is a common method for adaptive selection of snapshots using an error estimator or error indicator to identify the next sample location in parameter space. As the number of parameters increases, it becomes essential to combine greedy sampling with a scalable optimization search method as in Bui-Thanh, Willcox and Ghattas (2008*a*).

Algorithm 2 summarizes the adaptive greedy parameter sampling approach for a linear system with affine parametric dependence, which has full-order model (7.19) and reduced model (7.21). Figure 7.1 shows model reduction results for a system of this form from Bui-Thanh, Willcox and Ghattas (2008*b*) for forced response of a subsonic rotor blade undergoing rigid body motions. The problem is characterized by $n_m = 10$ parameters representing blade geometric variations that arise from manufacturing variability. The parametric reduced model is derived using greedy parameter sampling via the scalable optimization formulation of Bui-Thanh *et al.* (2008*a*). The histograms show representative results of the work per cycle for 5000 randomly sampled geometries computed using a linearized discontinuous Galerkin computational fluid dynamics (CFD) analysis ($n = 103\,008$ dof (degrees of freedom)) and a parametric POD reduced model ($r = 307$ dof).

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

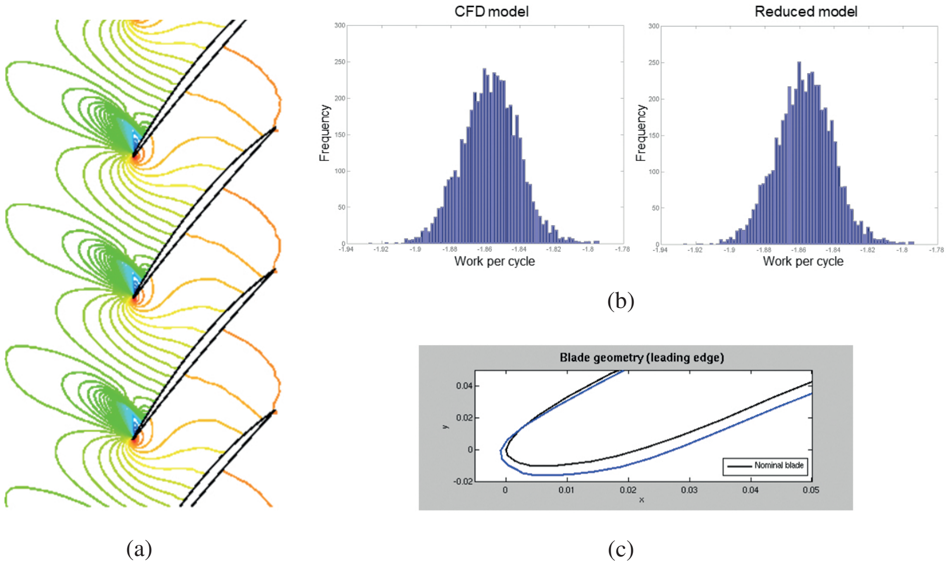


Figure 7.1. Model reduction results for compressor blade example of Bui-Thanh *et al.* (2008*b*). (a) Representative CFD pressure fields. (b) Histograms of work per cycle computed over 5000 different blade geometries using linearized CFD ($n = 103\,008$ dof) and a parametric POD reduced model ($r = 307$ dof). (c) Close-up of one sampled blade geometry compared to the nominal blade.

An alternative to a global basis is to employ multiple local bases throughout the parameter space. The localized strategy may be particularly appropriate as the number of parameters increases, since a single global basis may require the reduced dimension r to be large, which compromises the computational efficiency of the reduced model. As discussed in detail in Benner *et al.* (2015), there are a number of strategies to achieve localized reduced models. These strategies include interpolation of localized bases (Amsallem and Farhat 2008), interpolation of localized reduced model operators (Amsallem *et al.* 2009, Degroote *et al.* 2010, Lohmann and Eid 2009, Panzer *et al.* 2010, Amsallem and Farhat 2011), adaptive parameter domain partitioning (Haasdonk, Dihlmann and Ohlberger 2011, Eftang and Stamm 2012), partitioning into multiple reduced models by clustering and classification (Amsallem, Zahr and Farhat 2012, Peherstorfer, Butnaru, Willcox and Bungartz 2014) and adaptive updating of the reduced model through trust region model management (Arian, Fahl and Sachs 2002, Qian, Grepl, Veroy and Willcox 2017). We do not attempt to describe all these approaches here, but rather point the reader to Section 4 of Benner *et al.* (2015) for a comparative discussion.

7.5. *Model generalization, error estimates, stability guarantees and other metrics of success*

Surrogate modelling, whether achieved through projection-based model reduction, machine learning or response surface fitting, typically leads to models that provide significant computational savings. But how good are the surrogate models? Can they be used with confidence, especially in settings where predictions inform high-consequence decisions? In machine learning this notion is referred to as *generalization*: the ability of the model to issue predictions for previously unseen data. In computational science this notion is often referred to as *predictive modelling* or *predictive computational science*, defined in Oden, Babuška and Faghihi (2017) as 'the scientific discipline concerned with assessing the predictability of mathematical and computational models of physical events in the presence of uncertainties'.

The availability of rigorous error bounds or error estimates for a large class of systems is one advantage of projection-based reduced models over other surrogate modelling approaches. These bounds/estimators depend on exploiting the properties of the underlying system being approximated (*e.g.* the properties of the operators $\mathcal{A}(u)$ and $\mathbf{A}$) combined with the properties of the projection-based approximation. The reduced basis methodology in particular has had a strong emphasis on the derivation of *a posteriori* error estimates, including for elliptic (Patera and Rozza 2006, Rozza *et al.* 2008, Veroy and Patera 2005, Veroy, Prud'homme, Rovas and Patera 2003), parabolic (Grepl and Patera 2005) and hyperbolic (Haasdonk and Ohlberger 2008) parametrized PDEs. These approaches have paved the way to error estimation for a broader class of problems, including general linear dynamical systems (Haasdonk and Ohlberger 2011), parametrized saddle point problems (Gerner and Veroy 2012), variational inequalities (Zhang, Bader and Veroy 2016) and 4D-Var data assimilation (Kaercher, Boyaval, Grepl and Veroy 2018). Key in all of these works is that the error estimates are computed without recourse to the full model – that is, they can be computed cheaply so that the reduced model can rapidly issue both predictions and a rigorous estimate of the quality of those predictions.

8. Non-intrusive model reduction

The vast majority of model reduction methods are intrusive, but there is growing recognition of the importance of non-intrusive methods. This section first defines the properties of intrusive, non-intrusive and black-box methods. We then present Operator Inference (OpInf), a non-intrusive projection-based model reduction method. We close this section with a discussion of the relationship between model reduction and machine learning.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

8.1. Non-intrusive versus black-box methods

Consider the task of deriving a reduced model of a particular high-fidelity model. The high-fidelity model comprises a problem definition that includes the domain, governing equations, parameters, boundary conditions, initial conditions, *etc.* and a numerical implementation that solves the specified problem. The numerical implementation of the high-fidelity model is sometimes referred to as the full-order model.

In the context of model reduction, we put forward the following definitions.

- An *intrusive* method computes the reduced model by applying the basis expansions and projections to the operators implemented in the full-order model. For example, computing the reduced operators $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ in (7.12) and (7.13) requires access to the full-order operators $\mathbf{A}$ and $\mathbf{H}$ (or access to their actions on a vector). This typically requires intrusive access to the source code that implements the full-order model.
- A *black-box* method computes the reduced model without using *a priori* or explicit knowledge of the form of the high-fidelity problem definition or its numerical implementation. A black-box method operates on outputs of the full-order model, but does not require access to the full-order operators and does not use knowledge of their particular structure. For example, one black-box formulation is to fit a generic surrogate model (*e.g.* a polynomial response surface or a neural network) to the POD coefficients of sampled snapshot data (Ly and Tran 2001, Mainini and Willcox 2015, Swischuk *et al.* 2019, Wang, Hesthaven and Ray 2019).
- A *non-intrusive* method computes the reduced model using outputs of the full-order model but without access to the full-order operators (or to their action on a vector). Black-box methods are non-intrusive, but not all non-intrusive methods are necessarily black-box: a non-intrusive method can exploit knowledge of the high-fidelity problem definition and the full-order model structure, even though it does not access the full-order operators themselves. McQuarrie, Huang and Willcox (2021) refer to this as a ‘glass-box method’: one in which the form of the governing equations is known (*e.g.* the partial differential equations that define the problem of interest) but we do not have internal access to the code that produces the simulation data. The OpInf method of Peherstorfer and Willcox (2016), described in more detail in Section 8.2, is an example of a non-intrusive model reduction approach that is not black-box. Another example is the dynamic mode decomposition DMD (Schmid 2010, Kutz, Brunton, Brunton and Proctor 2016).

The distinction between non-intrusive and black-box methods is particularly important in the model reduction setting, where the goal is to obtain an approximation of a *known* high-fidelity model. Black-box methods are appealing because they are easy to apply, but they offer little in the way of scalability or mathematical guarantees of quality: the best one can do is hope that the training snapshots have been sufficiently rich and that the approximate representation is sufficiently expressive. Intrusive projection approaches are more difficult to apply, especially when the full-order model involves an industrial or legacy code, but have the advantage of error estimates for a large class of parametrized PDEs (Veroy *et al.* 2003, Veroy and Patera 2005, Grepl and Patera 2005, Rozza *et al.* 2008). Non-intrusive approaches offer a viable middle ground – providing ease of implementation in situations where an intrusive reduced model is not possible, but exploiting knowledge of the underlying problem to retain some form of mathematical accuracy guarantees.

8.2. Non-intrusive model reduction via Operator Inference

The OpInf approach proposed in Peherstorfer and Willcox (2016) uses the structure-preserving projection lens of Section 7 but infers the reduced model directly from snapshot data rather than computing the reduced operators via intrusive projection as in (7.12) and (7.13). More specifically, OpInf solves a regression problem to find the reduced operators that yield the reduced model that best matches projected snapshot data in a minimum residual sense.

Consider the task of determining a reduced model of (7.8). We first present OpInf in this quadratic setting and then discuss the more general non-linear setting. As before, we compute the POD basis by collecting the snapshots $\mathbf{u}_1, \ldots, \mathbf{u}_{n_s}$ and computing the dominant r left singular vectors of the snapshot matrix $\mathbf{U}$ (Algorithm 1). The next step in OpInf is to compute the representation of each snapshot in the POD coordinates, $\hat{\mathbf{u}}_j = \mathbf{V}^\top \mathbf{u}_j$, $j = 1, \ldots, n_s$. In addition, we collect the time derivative of each snapshot and compute its representation in the POD coordinates, $\dot{\hat{\mathbf{u}}}_j = \mathbf{V}^\top \dot{\mathbf{u}}_j$, $j = 1, \ldots, n_s$. Defining $\hat{\mathbf{U}}$ as the matrix of projected snapshots, with $\hat{\mathbf{u}}_j$ as its jth column, and $\dot{\hat{\mathbf{U}}}$ as the matrix of projected snapshot time derivatives, with $\dot{\hat{\mathbf{u}}}_j$ as its jth column, OpInf solves the least-squares problem

$$\min_{\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{H}} \in \mathbb{R}^{r\times r^2}} \sum_{j=1}^{n_s} \|\hat{\mathbf{A}}\hat{\mathbf{u}}_j + \hat{\mathbf{H}}(\hat{\mathbf{u}}_j \otimes \hat{\mathbf{u}}_j) - \dot{\hat{\mathbf{u}}}_j\|_2^2. \tag{8.1}$$

This regression problem seeks the reduced operators $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ that minimize the residual of the snapshot data evaluated in the form of the reduced model defined by (7.11). In other words, if the high-fidelity model to be approximated has the form (7.5) or (7.8), a projection-based reduced model has the form (7.6) or (7.11), and thus (8.1) seeks a model of that quadratic form that minimizes the residual of projected snapshot data. Another way to view this is that we are defining the functional form of the system input–output map and then inferring from data the representation of the map in the POD coordinate system.

As shown in Peherstorfer and Willcox (2016), the residual formulation employed in (8.1) can be separated into r independent least-squares problems that solve for

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

each row of $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ independently. We write the full OpInf regression problem in matrix form as

$$\min_{\mathbf{O}} \|\mathbf{D}\mathbf{O}^{\mathsf{T}} - \mathbf{R}^{\mathsf{T}}\|_F^2, \tag{8.2}$$

where

$$\begin{aligned}
\mathbf{O} &= \begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{H}} \end{bmatrix} \in \mathbb{R}^{r \times d(r)} && \text{(unknown operators)},\\
\mathbf{D} &= \begin{bmatrix} \hat{\mathbf{U}}^{\mathsf{T}} & (\hat{\mathbf{U}} \odot \hat{\mathbf{U}})^{\mathsf{T}} \end{bmatrix} \in \mathbb{R}^{n_s \times d(r)} && \text{(snapshot training data)},\\
\hat{\mathbf{U}} &= \begin{bmatrix} \hat{\mathbf{u}}_1 & \hat{\mathbf{u}}_2 & \cdots & \hat{\mathbf{u}}_{n_s} \end{bmatrix} \in \mathbb{R}^{r \times n_s} && \text{(projected snapshots)},\\
\mathbf{R} &= \begin{bmatrix} \dot{\hat{\mathbf{u}}}_1 & \dot{\hat{\mathbf{u}}}_2 & \cdots & \dot{\hat{\mathbf{u}}}_{n_s} \end{bmatrix} \in \mathbb{R}^{r \times n_s} && \text{(snapshot time derivatives)}.
\end{aligned}$$

The r independent least-squares problems are then given by

$$\min_{\mathbf{o}_i \in \mathbb{R}^{d(r)}} \|\mathbf{D}\mathbf{o}_i - \mathbf{r}_i\|_2, \quad i = 1, \ldots, r, \tag{8.3}$$

where $\mathbf{o}_i$ is a column of $\mathbf{O}^{\mathsf{T}}$ and $\mathbf{r}_i$ is a column of $\mathbf{R}^{\mathsf{T}}$. Each least-squares problem in (8.3) has $d(r)$ reduced operator coefficients to be inferred. Enforcing symmetry of $\hat{\mathbf{H}}$, we have $d(r) = r + r(r+1)/2$.[1]

The forms of (8.2) and (8.3) show clearly that the OpInf least-squares problem is linear in the coefficients of the unknown reduced operators. Equation (8.3) has a unique solution if the data matrix $\mathbf{D}$ has full rank (noting again that the redundant terms in $(\hat{\mathbf{U}} \odot \hat{\mathbf{U}})$ are eliminated). This implies that the number of snapshots must satisfy $n_s \geq d(r)$. Even when the data matrix has full rank, for some problems the OpInf problem may require regularization to avoid overfitting. There are three sources of noise that affect the solution of (8.3): numerical error in the estimates of the time derivatives $\dot{\hat{\mathbf{u}}}_j$, closure error from the neglected POD modes, and model misspecification error (*i.e.* if the full-order model used to generate the snapshots does not have the exact quadratic form of (7.8)). Tikhonov regularization has been shown to be effective (Swischuk, Kramer, Huang and Willcox 2020, McQuarrie *et al.* 2021), yielding a regularized OpInf problem of the form

$$\min_{\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}, \hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}} \sum_{j=1}^{n_s} \|\hat{\mathbf{A}}\hat{\mathbf{u}}_j + \hat{\mathbf{H}}(\hat{\mathbf{u}}_j \otimes \hat{\mathbf{u}}_j) - \dot{\hat{\mathbf{u}}}_j\|_2^2 + \lambda_1 \|\hat{\mathbf{A}}\|_F^2 + \lambda_2 \|\hat{\mathbf{H}}\|_F^2, \tag{8.4}$$

where λ_1 and λ_2 are regularization parameters, here chosen to weight $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ differently (because they typically have different scales).

Algorithm 3 presents the steps of the regularized OpInf approach. A key point to emphasize is that the reduced model operators $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ are computed without needing explicit access to the original high-dimensional operators $\mathbf{A}$ and $\mathbf{H}$. All

[1] To keep the notation simple, we write $\mathbf{O} = [\hat{\mathbf{A}}\ \hat{\mathbf{H}}]$, which suggests a dimension of $r + r^2$, but note that in implementation we do not solve for the redundant terms in $\hat{\mathbf{H}}$. Similarly we eliminate the redundant terms in the Khatri–Rao product defining the data matrix $\mathbf{D}$.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Algorithm 3 Non-intrusive regularized OpInf approach for a quadratic system (Peherstorfer and Willcox 2016, McQuarrie *et al.* 2021).

```
Inputs: snapshots $\mathbf{U} \in \mathbb{R}^{n \times n_s}$, snapshot time derivatives $\dot{\mathbf{U}} \in \mathbb{R}^{n \times n_s}$, POD cumulative energy threshold $\kappa$
1: Compute the SVD of $\mathbf{U}$
2: $r \leftarrow$ choose $r$ such that $(\sum_{i=1}^{r} \sigma_i^2)/(\sum_{i=1}^{n_s} \sigma_i^2) > \kappa$ {Determine the reduced model dimension}
3: $\mathbf{V} \leftarrow$ the $r$ leading left singular vectors of $\mathbf{U}$ {Compute the POD basis from the snapshots}
4: $\hat{\mathbf{U}} \leftarrow \mathbf{V}^{\mathsf{T}}\mathbf{U}$ {Project snapshots onto the $r$-dimensional POD subspace}
5: $\mathbf{R} \leftarrow \mathbf{V}^{\mathsf{T}}\dot{\mathbf{U}}$ {Time derivatives of the projected snapshots}
6: $\lambda_1, \lambda_2 \leftarrow$ Set regularization parameters by hyper-parameter optimization
7: $\hat{\mathbf{A}}, \hat{\mathbf{H}} \leftarrow$ solve regularized OpInf regression (8.4) via $r$ independent least-squares problems
8: return $\hat{\mathbf{A}}, \hat{\mathbf{H}}$
```

steps of Algorithm 3 can be computed non-intrusively, that is, by operating only on snapshots output by the full-order model. However, Algorithm 3 uses knowledge of the equations being solved by the full-order model to postulate the form of the reduced model. Thus Algorithm 3 is non-intrusive but it is *not* black-box.

Algorithm 3 presents the regularized OpInf approach of McQuarrie *et al.* (2021) for the quadratic system (7.8). It is straightforward to see how the approach extends to systems with forcing terms, and cubic and higher-order polynomial state-dependence. In these cases, the least-squares optimization problems remain linear, but the number of coefficients to be inferred in each least-squares problem, $d(r)$, grows. For a cubic system, $d(r) \sim O(r^3)$ and it becomes increasingly difficult to maintain a well-conditioned least-squares problem. Benner *et al.* (2020) have developed an extension of OpInf to systems with spatially local non-polynomial non-linear terms. We also note that the OpInf approach is presented here using the POD basis, but any low-dimensional coordinate system can be used as long as the corresponding projected snapshot data can be computed.

What can we say about the properties of an OpInf reduced model, particularly its accuracy in issuing predictions beyond the training data? The OpInf method, while non-intrusive itself, is set up to mimic the approximations employed in intrusive projection-based model reduction. The original OpInf paper (Peherstorfer and Willcox 2016) shows that, subject to having sufficient snapshot data, the intrusive POD reduced models can be recovered asymptotically. Of more practical significance, Peherstorfer (2020) shows that if the snapshot data have Markovian dynamics in the reduced space, OpInf achieves exact recovery of the intrusive reduced model pre-asymptotically. Peherstorfer (2020) derives a data generation scheme that achieves this pre-asympotic recovery. These recovery results are important because they guarantee that the non-intrusive OpInf reduced models

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

inherit the properties of intrusive projection-based models, which are well studied for a large class of systems as discussed in Section 7.5. Furthermore, the polynomial structure of an OpInf reduced model makes it more amenable to stability analysis. For example, Benner and Breiten (2015) and Kramer (2020) analyse the stability of quadratic-bilinear reduced models.

8.3. Model reduction and its relationship to machine learning

What is the connection between model reduction and machine learning? Model reduction methods have grown from the *computational science* community, with a focus on reducing high-dimensional models that arise from physics-based modelling, whereas machine learning has grown from the *computer science* community, with a focus on learning models from black-box data streams. The perspectives are traditionally different, yet there are clear connections between the two fields. These connections are becoming particularly pertinent with the growing interest in applying machine learning to problems in science and engineering, where there is a need to embed physics constraints within the machine learning method.

Broadly speaking, the three main algorithmic steps of projection-based model reduction are as follows.

1 *Training data generation.* Solve the full-order model to generate training data. These training data are typically in the form of snapshots.

2 *Identify structure.* Define a low-dimensional representation based on the training data and, in some cases, based also on the properties of the full-order model. Many, but not all, methods define the reduced-order representation through a low-dimensional basis that defines a linear subspace of the full-order state space.

3 *Reduction.* Derive the reduced model as a projection of the full-order model onto the reduced space.

These three steps manifest in different ways for different model reduction methods. In step 1, POD and reduced basis model reduction methods compute the basis entirely from snapshot training data, while rational interpolation methods employ full-order solves at system-specific optimal interpolation points (Gugercin, Antoulas and Beattie 2008, Antoulas *et al.* 2020). In step 2 there are many different ways to compute the low-dimensional representation, with POD discussed in this paper being the most widely applied. Step 3 has classically been implemented by explicit projection of the full-order operators onto the reduced space, but emerging data-driven methods such as OpInf instead cast step 3 as an inference task. It is also important to note that in many cases these steps are performed iteratively. This is the case, for example, in adaptive snapshot generation with greedy sampling (Prud'homme *et al.* 2002) and in the iterative rational Krylov algorithm (IRKA) (Gugercin *et al.* 2008). In both cases the three steps are repeated iteratively until the desired accuracy levels are achieved.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Casting model reduction in this way reveals both the commonalities and the differences with machine learning. The first two steps – training data generation and identification of structure – are shared. Not only are they common in philosophy but they are also common at a deeper algorithmic level. For example, the workhorses of structure identification via linear subspaces – POD and PCA – are equivalent methods, both based on singular value decomposition. Step 3 – reduction – is where machine learning and model reduction methods differ. This point of departure is due to their respective origins in computer science and computational science, as discussed above. However, as increasing attention is given to the areas of *scientific machine learning* and *physics-informed machine learning*, which seek to embed physics constraints within machine learning methods, it is important not to overlook the relevance of model reduction theory and methods. Indeed, viewing the reduction step as an inference problem, as in the OpInf method, brings model reduction much closer to the perspectives of machine learning: the chosen representation is informed by the physics (projected operators of a PDE or a state-space system, rather than a generic neural network representation) and the loss function is the residual of the governing equations represented in the low-dimensional space.

9. Non-linear model reduction

We discuss the challenges of model reduction for non-linear systems and then discuss different strategies to address these challenges. We first discuss the use of hyper-reduction methods such as DEIM to approximate non-linear terms. We then discuss alternative strategies based on variable transformations and present the Lift & Learn approach, which is based on the OpInf method of Section 8.2.

9.1. *Challenges of model reduction for non-linear systems*

Let us now return to the general projection framework, but consider a non-linear system. Consider the non-linear system of ODEs

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{9.1}$$

where as before $\mathbf{u}(t) \in \mathbb{R}^n$ is the state at time t, with dimension n, and the non-linear function $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ maps the state $\mathbf{u}(t)$ to its derivative $\mathrm{d}\mathbf{u}/\mathrm{d}t$. A typical case of interest is where the system (9.1) results from the spatial discretization of a system of non-linear PDEs. A general non-linear PDE defined on the domain Ω and time horizon $(0, t_f)$ may be written

$$\frac{\partial u}{\partial t} = \mathcal{F}(u) \quad \text{in } \Omega \times (0, t_f), \tag{9.2}$$

with appropriate boundary and initial conditions, where $\mathcal{F}\colon \mathcal{U} \to \mathcal{U}^*$ is the non-linear PDE operator.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Working with the ODE representation (9.1), we approximate the state in an r-dimensional orthonormal basis as in Section 7, $\mathbf{u}(t) \approx \sum_{j=1}^{r} \mathbf{v}_j \hat{u}_j(t) = \mathbf{V}\hat{\mathbf{u}}(t)$. Using a Galerkin projection, we obtain the reduced model of (9.1) as

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \mathbf{V}^{\mathsf{T}}\mathbf{f}(\mathbf{V}\hat{\mathbf{u}}), \quad \hat{\mathbf{u}}(0) = \mathbf{V}^{\mathsf{T}}\mathbf{u}_0. \tag{9.3}$$

While (9.3) is of reduced dimension, its solution still requires computations that scale with the full-order dimension n, because of the non-linear term $\mathbf{f}(\cdot)$ on the right-hand side of the state equation. Therefore an extra layer of approximation is required to achieve a reduced model that is efficient to solve.

One approach is to approximate the non-linear system by a set of piecewise linear models (Rewienski and White 2003). Each linear model can then be reduced by straightforward projection as in Section 7, and the resulting piecewise linear reduced model is computationally efficient. A second class of methods retain the form of the non-linear reduced model (9.3) but introduce an extra layer of approximation to make the right-hand side computationally efficient. This extra layer of approximation is sometimes referred to as 'hyper-reduction'. The empirical interpolation method (EIM) (Barrault *et al.* 2004) and its discrete variant, the discrete empirical interpolation method (DEIM) (Chaturantabut and Sorensen 2010), achieve hyper-reduction by interpolation on a low-dimensional basis for the non-linear term. Other hyper-reduction methods include the missing point estimation (MPE) (Astrid, Weiland, Willcox and Backx 2008), the masked projection formulation of Galbally *et al.* (2010) and Gauss–Newton with approximated tensors (GNAT) (Carlberg, Farhat, Cortial and Amsallem 2013), which all employ the gappy POD method of Everson and Sirovich (1995).

9.2. *Discrete empirical interpolation method*

As one example of a non-linear model reduction method that includes hyper-reduction, we describe the widely used POD-DEIM approach (Chaturantabut and Sorensen 2010). This requires the following additional steps over the POD model reduction described in Section 7. First, during the snapshot generation phase, we collect snapshots of the non-linear term $\mathbf{f}(\mathbf{u})$, in addition to snapshots of the states. This leads to the non-linear term snapshots $\mathbf{f}(\mathbf{u}_1), \ldots, \mathbf{f}(\mathbf{u}_{n_s})$. Define the non-linear term snapshot matrix $\mathbf{F} \in \mathbb{R}^{n \times n_s}$ whose jth column is the non-linear term snapshot $\mathbf{f}(\mathbf{u}_j)$. Note that we have written these non-linear term snapshots as being evaluated at the state snapshots; this does not necessarily need to be the case. That is, one could in principle generate the state snapshots and non-linear term snapshots at different conditions, but doing so would be computationally inefficient because it would require additional runs of the expensive full-order model.

The second step in the POD-DEIM approach is to apply POD to the non-linear term snapshot set to compute the DEIM basis. That is, we compute r_d DEIM basis vectors as the left singular vectors of $\mathbf{F}$. This leads to the DEIM basis vectors, which

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Algorithm 4 Computing the POD-DEIM reduced model.

```
Inputs: POD basis $\mathbf{V} \in \mathbb{R}^{n\times r}$ computed using Algorithm 1, non-linear term snapshots $\mathbf{F} \in \mathbb{R}^{n\times n_s}$, DEIM basis cumulative energy threshold $\kappa_d$
1: Compute the SVD of $\mathbf{F}$
2: $r_d \leftarrow$ choose $r_d$ such that $(\sum_{i=1}^{r_d} \sigma_i^2)/(\sum_{i=1}^{n_s} \sigma_i^2) > \kappa_d$ {Determine the dimension of the DEIM basis}
3: $\mathbf{W} \leftarrow$ the $r$ leading left singular vectors of $\mathbf{F}$ {Compute the DEIM basis}
4: $\{p_1, \dots, p_{r_d}\} \leftarrow$ choose $r_d$ interpolation points $p_1, \dots, p_{r_d} \in \{1, \dots, n\}$ {Choose DEIM interpolation points, e.g. using the point selection approach in Chaturantabut and Sorensen (2010)}
5: $\mathbf{P} \leftarrow [\mathbf{e}_{p_1}, \dots, \mathbf{e}_{p_{r_d}}]$ {Define the DEIM points selection matrix}
6: POD-DEIM reduced model $\leftarrow \mathrm{d}\hat{\mathbf{u}}/\mathrm{d}t = \mathbf{V}^\top\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{P}^\top\mathbf{f}(\mathbf{V}\hat{\mathbf{u}})$
7: return $\mathbf{W}, \mathbf{P}, r_d$
```

we denote $\mathbf{w}_1, \dots, \mathbf{w}_{r_d}$. We define the DEIM basis matrix $\mathbf{W} \in \mathbb{R}^{n\times r_d}$, which has $\mathbf{w}_j$ as its jth column. The number of DEIM basis vectors r_d is typically chosen according to the decay of the singular values of $\mathbf{W}$, using a condition analogous to (7.16).

The third step is to select a set of r_d DEIM interpolation points. These are the elements of $\mathbf{f}(\cdot)$ at which the full-order model's non-linear function must be evaluated in order to construct the hyper-reduction approximation. Consider r_d pairwise distinct interpolation points $p_1, \dots, p_{r_d} \in \{1, \dots, n\}$. The DEIM interpolation points matrix is then defined as $\mathbf{P} = [\mathbf{e}_{p_1}, \dots, \mathbf{e}_{p_{r_d}}] \in \mathbb{R}^{n\times r_d}$, where $\mathbf{e}_i \in \{0, 1\}^n$ is the ith canonical unit vector. This interpolation point selection can be done using a greedy selection as in Chaturantabut and Sorensen (2010) or via QR decomposition as in Drmac and Gugercin (2016).

The DEIM approximation of the non-linear function $\mathbf{f}$ evaluated at the state vector $\mathbf{u}$ is then

$$\mathbf{f}(\mathbf{u}) \approx \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{P}^\top\mathbf{f}(\mathbf{u}), \tag{9.4}$$

where $\mathbf{P}^\top\mathbf{f}(\mathbf{u})$ samples the non-linear function at r_d components only, and thus requires $r_d < n$ function evaluations. The DEIM interpolation points matrix $\mathbf{P}$ and the DEIM basis $\mathbf{W}$ are selected such that the matrix $(\mathbf{P}^\top\mathbf{W})^{-1} \in \mathbb{R}^{r_d\times r_d}$ has full rank. Employing the approximation (9.4) in the Galerkin-reduced model (9.3), we obtain the POD-DEIM reduced model as

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \mathbf{V}^\top\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{P}^\top\mathbf{f}(\mathbf{V}\hat{\mathbf{u}}). \tag{9.5}$$

Algorithm 4 summarizes the derivation of a POD-DEIM reduced model.

For systems where the non-linear term has a local dependence on the state, *i.e.* where the ith element of $\mathbf{f}$ depends only on the ith element of $\mathbf{u}$, we can write the

POD-DEIM reduced model as

$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \mathbf{V}^{\mathsf{T}}\mathbf{W}(\mathbf{P}^{\mathsf{T}}\mathbf{W})^{-1}\mathbf{P}^{\mathsf{T}}\mathbf{f}(\mathbf{P}^{\mathsf{T}}\mathbf{V}\hat{\mathbf{u}}). \tag{9.6}$$

In this case all terms involving the full-order dimension n can be computed offline, and the cost of evaluating terms online in the POD-DEIM model scales with the number of POD basis vectors, r, and the number of DEIM basis vectors, r_d, but not with n.

Peherstorfer, Drmac and Gugercin (2020) showed the potential numerical advantages of oversampling with DEIM, that is, choosing more than r_d points, which leads to hyper-reduction via regression rather than interpolation.

POD-DEIM and the other hyper-reduction methods have been successfully applied across a broad range of problems with different forms of non-linear structure. An incomplete list of compelling examples includes data assimilation for geophysical flows (Ştefănescu, Sandu and Navon 2015), viscoplasticity in solid mechanics (Ghavamian, Tiso and Simone 2017), electrocardiology (Yang and Veneziani 2017) and rocket combustion (Huang, Duraisamy and Merkle 2019). However, it remains a challenge to achieve efficient model reduction for highly non-linear systems, especially those that exhibit behaviour across a range of time scales. In such cases, the complexity of the hyper-reduction term (*e.g.* the dimension of the DEIM basis r_d) can become so large that the reduced models are not computationally efficient. For example, Huang, Xu, Duraisamy and Merkle (2018) demonstrated the difficulties of reducing the complex dynamics in rocket combustion applications.

9.3. Exploiting variable transformations in non-linear model reduction

POD-DEIM and other hyper-reduction methods target generic non-linear systems of the form (9.1) and achieve efficiency through selective sampling of $\mathbf{f}(\mathbf{u})$ but without otherwise exploiting known structure of $\mathbf{f}(\cdot)$. An alternative approach is to instead employ variable transformations to manipulate the form of (9.1), thereby exposing structure that is amenable to projection-based approximation without the need for hyper-reduction.

The idea of variable transformations to promote system structure has a long history in many different fields. For example, McCormick (1976) used variable substitutions to solve non-convex optimization problems. In operations research, the reformulation of a problem in higher dimensions is referred to as an 'extended formulation' or 'lifting' (Balas 2005). Hughes, Franca and Mallet (1986) used the entropy variables to derive finite element methods for solving the Euler and Navier–Stokes equations that automatically satisfy the second law of thermodynamics and thus guarantee stability of the discrete solution. Kerner (1981) showed how general non-linear ODEs can be written as 'polynomial ordinary differential systems (PODS)' through the introduction of additional variables. In biology, variable transformations called 'recasting' are used to transform non-linear ODEs to

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

the so-called S-system form, a polynomial form that is faster to solve numerically (Savageau and Voit 1987). Approaches based on the Koopman operator lift a non-linear dynamical system to an infinite-dimensional space in which the dynamics are linear (Mezić 2013, Korda and Mezić 2018). It is important to emphasize that extended formulations, lifting and recasting employ problem reformulation, *not* problem approximation.

Gu (2011) introduced the idea of reformulating non-linear dynamical systems in quadratic form for model reduction and showed that the number of auxiliary variables needed to lift a system to quadratic-bilinear form is linear in the number of elementary non-linear functions in the original state equations. The work in (Gu 2011) shows that a large class of non-linear terms that appear in engineering systems (including monomial, sinusoidal and exponential terms) may be lifted to quadratic form. Benner and Breiten (2015), Kramer and Willcox (2019, 2021) and Qian, Kramer, Peherstorfer and Willcox (2020) extended lifting to model reduction of problems governed by PDEs, and show that it is a competitive alternative to hyper-reduction methods.

The relevance of variable transformations to model reduction can be seen through an illustrative example. Consider the Euler equations describing the dynamics of inviscid compressible flow. In one spatial dimension, the Euler equations are written in conservative form (here in strong form for the simple case of no source terms and an ideal gas) as

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho v\\ \rho e\end{pmatrix} = -\frac{\partial}{\partial x}\begin{pmatrix}\rho v\\ \rho v^2 + p\\ (\rho e + p)v\end{pmatrix}, \tag{9.7}$$

where t is time, x is the spatial dimension, and the conservative state variables $u = (\rho \;\; \rho v \;\; \rho e)^\mathsf{T}$ are the density ρ, specific momentum ρv and total energy ρe. The equation of state

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2 \tag{9.8}$$

relates energy and pressure, p, via the heat capacity ratio γ. The structure of the system can be seen in the PDE operators in (9.7). For example, the first equation $\partial\rho/\partial t = -\partial\rho v/\partial x$ describing conservation of mass is linear in the conservative state variables ρ and ρv; however, the other two equations are non-linear and, while some structure is apparent, the non-linear terms are non-polynomial. Working with this form of the governing equations would require hyper-reduction, a piecewise linear representation or some other form of approximation in order to make the reduced model efficient.

It is well known that the Euler equations can be alternatively formulated in the specific volume form, using the state $\tilde{u} = (v \;\; p \;\; \zeta)^\mathsf{T}$, where $\zeta = 1/\rho$ is the specific volume, v the velocity and p the pressure. The conservation equations expressed

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

in (9.7) then take the form

$$\frac{\partial}{\partial t}\begin{pmatrix} \upsilon \\ p \\ \zeta \end{pmatrix} = -\frac{\partial}{\partial x}\begin{pmatrix} \upsilon\dfrac{\partial \upsilon}{\partial x} + \zeta\dfrac{\partial p}{\partial x} \\ \gamma p\dfrac{\partial \upsilon}{\partial x} + \upsilon\dfrac{\partial p}{\partial x} \\ \upsilon\dfrac{\partial \zeta}{\partial x} - \zeta\dfrac{\partial \upsilon}{\partial x} \end{pmatrix}, \tag{9.9}$$

where it can now be seen that the equations contain only quadratic non-linear dependences on the transformed state $\tilde{u}$ and its spatial derivatives. This example illustrates that the choice of state variables impacts the structure of the resulting equations – by no means a surprising result, but one that has been under-appreciated in model reduction methods. For this illustrative example of the Euler equations, working with the conservative variables leads to a system of the general non-linear form (9.2) while working with the specific volume variables leads to the quadratic form (9.9). The former is most common in computational fluid dynamics (CFD) codes, while the latter has clear advantages for model reduction. By discretizing the conservative form of the equations in a CFD method, one can craft a numerical solution method that has desirable properties (such as respecting conservation laws). However, if the goal is not to create a physics-based solver but rather to use the physical equations as a way to identify, represent and exploit system structure, then we can see immediately that the specific volume form (9.9) yields a more attractive set of equations to work with because of the quadratic structure.

The Euler equations provide an illustrative example that highlights the power of variable transformations, but in many cases the governing PDEs do not obviously admit a variable transformation that yields a polynomial representation. This is where the idea of *lifting* comes in: the introduction of auxiliary variables that expose polynomial structure in the system. Classical system theory defines a system state as a set of quantities that provide a minimal representation of the system while containing enough information to determine the system's future behaviour under known input conditions. Lifting introduces ('unnecessary') auxiliary variables and their evolution equations, with the key idea that the dynamics of the lifted system are polynomial in the expanded set of variables.[2] To illustrate lifting, consider the simple non-linear PDE

$$\frac{\partial u}{\partial t} = u - \mathrm{e}^{u}, \tag{9.10}$$

with scalar state $u(x,t)$. To lift (9.10) to quadratic form, define the auxiliary variable $-\mathrm{e}^{u}$ and the lifted state $\tilde{u} = (\tilde{u}_1 \;\; \tilde{u}_2)^{\mathsf{T}} = (u \quad -\mathrm{e}^{u})^{\mathsf{T}}$. The governing equations can

[2] We loosely call the expanded set of variables the 'lifted state' even though technically this is not a 'state' in the system-theoretic sense.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

then be written

$$\frac{\partial}{\partial t}\begin{pmatrix}\tilde{u}_1\\ \tilde{u}_2\end{pmatrix} = \begin{pmatrix}1\\ -\mathrm{e}^u\end{pmatrix}\frac{\partial u}{\partial t} = \begin{pmatrix}1\\ -\mathrm{e}^u\end{pmatrix}(u - \mathrm{e}^u) = \begin{pmatrix}\tilde{u}_1 + \tilde{u}_2\\ \tilde{u}_1\tilde{u}_2 + (\tilde{u}_2)^2\end{pmatrix}. \tag{9.11}$$

The lifted equation (9.11) is quadratic in the lifted state $\tilde{u}$.

Qian *et al.* (2020) defines lifting in a general form for systems of PDEs. Consider the non-linear PDE (9.2) where the (continuous) state comprises n_u different quantities. That is, $u(x,t) = (u_1(x,t), \ \ldots, \ u_{n_u}(x,t))^\intercal$ and $\mathcal{F}(u) = (\mathcal{F}_1(u), \ \ldots, \ \mathcal{F}_{n_u}(u))^\intercal$. Define the lifting map, $\mathcal{T}$, as the map that lifts the original state $u(x,t)$ to the lifted state $\tilde{u}(x,t)$ with lifted state dimension $n_{\tilde{u}} \geq n_u$. That is, $\tilde{u}(x,t) = \mathcal{T}(u(x,t))$. Qian *et al.* (2020) define the map $\mathcal{T}$ to be a *quadratic lifting* of (9.2) if the following conditions are met.

(i) The map $\mathcal{T}$ is differentiable with respect to u with bounded derivative, that is, if $\mathcal{J}(u)$ is the Jacobian of $\mathcal{T}$ with respect to u, then

$$\sup_{u\in\mathcal{U}} \|\mathcal{J}(u)\| \leq c, \tag{9.12}$$

for some constant $c > 0$.

(ii) The lifted state $\tilde{u}$ satisfies

$$\frac{\partial \tilde{u}}{\partial t} = \frac{\partial \mathcal{T}(u)}{\partial t} = \mathcal{J}(u)\mathcal{F}(u) = \mathcal{A}(\tilde{u}) + \mathcal{H}(\tilde{u},\tilde{u}), \tag{9.13}$$

where

$$\mathcal{A}(\tilde{u}) = \begin{pmatrix}\mathcal{A}_1(\tilde{u})\\ \vdots\\ \mathcal{A}_{n_{\tilde{u}}}(\tilde{u})\end{pmatrix}, \quad \mathcal{H}(\tilde{u},\tilde{u}) = \begin{pmatrix}\mathcal{H}_1(\tilde{u},\tilde{u})\\ \vdots\\ \mathcal{H}_{n_{\tilde{u}}}(\tilde{u},\tilde{u})\end{pmatrix}, \tag{9.14}$$

for some linear functions $\mathcal{A}_j$ and quadratic functions $\mathcal{H}_j$, $j = 1, 2, \ldots, n_{\tilde{u}}$.

Equation (9.13) is the lifted PDE, which contains only quadratic non-linear terms. In other words we have identified a lifted state $\tilde{u}$ for which the governing PDEs could be written in quadratic form.

Conditions (9.12)–(9.14) define a map $\mathcal{T}$ that leads to a quadratic lifting. Condition (9.13) can be modified to define maps to cubic and higher-order polynomial forms by adding the appropriate polynomial functions of $\tilde{u}$.

As an illustrative example, consider the non-adiabatic tubular reactor model with Arrhenius single reaction terms of Heinemann and Poore (1981). We present the lifting of this model from Kramer and Willcox (2019). The governing equations describe the evolution of the species concentration $\psi(x,t)$ and temperature $\theta(x,t)$ with spatial variable $x \in (0,1)$ and time $t > 0$, via the coupled set of PDEs

$$\frac{\partial \psi}{\partial t} = \frac{1}{Pe}\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial \psi}{\partial x} - \mathcal{D}\psi\, \mathrm{e}^{\gamma-\gamma/\theta}, \tag{9.15}$$

$$\frac{\partial \theta}{\partial t} = \frac{1}{Pe}\frac{\partial^2 \theta}{\partial x^2} - \frac{\partial \theta}{\partial x} - \beta(\theta - \theta_{\text{ref}}) + \mathcal{B}\mathcal{D}\psi\, \mathrm{e}^{\gamma-\gamma/\theta}. \tag{9.16}$$

Here the parameters are the Damköhler number $\mathcal{D}$, Péclet number Pe and known constants $\mathcal{B}$, β, θ_{ref} and γ. Robin boundary conditions are imposed on the left boundary of the domain and Neumann boundary conditions on the right:

$$\begin{aligned} \frac{\partial \psi}{\partial x}(0,t) &= Pe(\psi(0,t) - 1), & \frac{\partial \theta}{\partial x}(0,t) &= Pe(\theta(0,t) - 1), \\ \frac{\partial \psi}{\partial x}(1,t) &= 0, & \frac{\partial \theta}{\partial x}(1,t) &= 0. \end{aligned}$$

The initial conditions are prescribed as $\psi(x,0) = \psi_0(x)$ and $\theta(x,0) = \theta_0(x)$.

To lift the equations (9.15) and (9.16), we introduce the auxiliary variables $\tilde{u}_3 = \mathrm{e}^{\gamma-\gamma/\theta}$, $\tilde{u}_4 = 1/\theta^2$ and $\tilde{u}_5 = 1/\theta$. The lifting map is thus

$$\mathcal{T}\colon \begin{pmatrix} \psi \\ \theta \end{pmatrix} \mapsto \begin{pmatrix} \psi \\ \theta \\ \mathrm{e}^{\gamma-\gamma/\theta} \\ \dfrac{1}{\theta^2} \\ \dfrac{1}{\theta} \end{pmatrix} \equiv \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{pmatrix}. \tag{9.17}$$

Working with the five-dimensional lifted state $\tilde{u}$ and applying the chain rule to derive the auxiliary equations leads to the following set of lifted PDEs, in which all non-linear terms have polynomial form in the lifted variables:

$$\frac{\partial \psi}{\partial t} = \underbrace{\frac{1}{Pe}\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial \psi}{\partial x}}_{\text{linear}} - \underbrace{\mathcal{D}\psi\tilde{u}_3}_{\text{quadratic}}, \tag{9.18}$$

$$\frac{\partial \theta}{\partial t} = \underbrace{\frac{1}{Pe}\frac{\partial^2 \theta}{\partial x^2} - \frac{\partial \theta}{\partial x} - \beta(\theta - \theta_{\text{ref}})}_{\text{linear}} + \underbrace{\mathcal{B}\mathcal{D}\psi\tilde{u}_3}_{\text{quadratic}}, \tag{9.19}$$

$$\frac{\partial \tilde{u}_3}{\partial t} = \underbrace{\gamma \tilde{u}_3 \tilde{u}_4 \frac{\partial \theta}{\partial t}}_{\text{quartic}}, \tag{9.20}$$

$$\frac{\partial \tilde{u}_4}{\partial t} = \underbrace{-2\tilde{u}_4 \tilde{u}_5 \frac{\partial \theta}{\partial t}}_{\text{quartic}}, \tag{9.21}$$

$$\frac{\partial \tilde{u}_5}{\partial t} = \underbrace{-\tilde{u}_4 \frac{\partial \theta}{\partial t}}_{\text{cubic}}, \tag{9.22}$$

where we compactly write $\partial\theta/\partial t$ on the right-hand sides of (9.20)–(9.22) but recognize that this term is given by (9.19). At the level of the continuous formulation, the lifted system (9.18)–(9.22) is equivalent to the original equations (9.15)–(9.16). The polynomial structure of the lifted equations could be directly exploited in the projection-based model reduction framework of Section 7, while the original equations would require hyper-reduction or some other approximate treatment of the non-linear terms.

In practice, a quartic non-linear term would likely prove to be computationally unmanageable, since the dimension of the reduced fourth-order tensor would be $r \times r^4$. The tubular reactor system can be further lifted by the introduction of three additional auxiliary variables: $\tilde{u}_6 = \psi\tilde{u}_3$, $\tilde{u}_7 = \tilde{u}_4\tilde{u}_5$ and $\tilde{u}_8 = \tilde{u}_3\tilde{u}_4$. The system with eight-dimensional lifted state $\tilde{u} = (\psi \quad \theta \quad \tilde{u}_3 \quad \tilde{u}_4 \quad \tilde{u}_5 \quad \tilde{u}_6 \quad \tilde{u}_7 \quad \tilde{u}_8)^{\mathsf{T}}$ has only quadratic and linear terms, but the three new equations are algebraic:

$$0 = \tilde{u}_6 - \tilde{u}_3\psi, \tag{9.23}$$

$$0 = \tilde{u}_7 - \tilde{u}_4\tilde{u}_5, \tag{9.24}$$

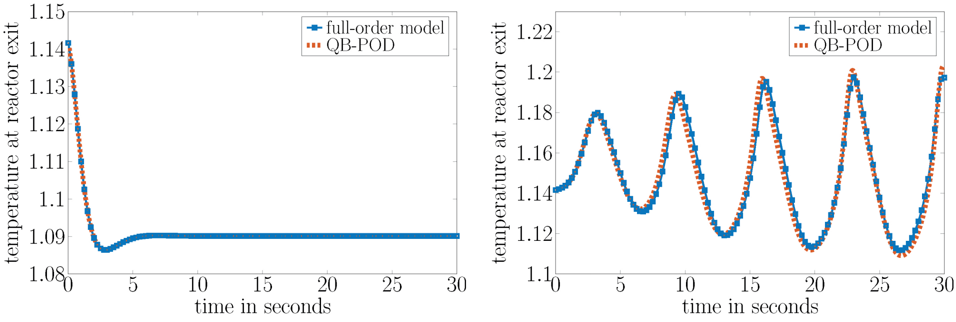
$$0 = \tilde{u}_8 - \tilde{u}_3\tilde{u}_4. \tag{9.25}$$

This quadratic system can still be directly reduced via projection, but special care is needed with the differential algebraic equation (DAE) structure; see *e.g.* Benner and Stykel (2017) for a discussion of the challenges and methods for reduction of DAE systems.

We have presented this tubular reactor example in some detail to emphasize the point that the choice of state variables plays a central role in defining problem structure. With the derivations presented, we now have three options: reduce the original non-linear system (9.15)–(9.16) using hyper-reduction or a trajectory piecewise polynomial approximation, or introduce three auxiliary variables and reduce (9.18)–(9.22) exploiting the quartic structure, or introduce six auxiliary variables and reduce (9.18)–(9.25) exploiting the quadratic structure while treating the DAE form.

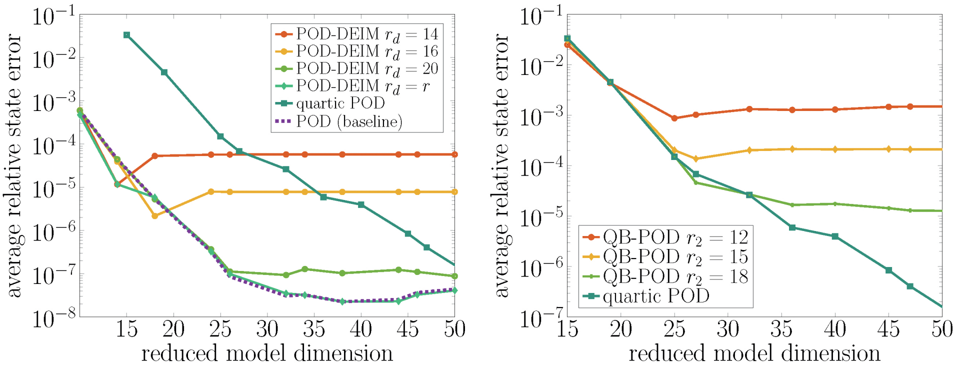
Kramer and Willcox (2019) presented a numerical study of this tubular reactor example, comparing POD model reduction of the lifted quartic system (9.18)–(9.22), POD model reduction of the lifted quadratic DAEs (9.18)–(9.25) and

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press



(a) (b)

Figure 9.1. Model reduction results for tubular reactor of Kramer and Willcox (2019, Figure 5). The reduced model uses lifting to quadratic-bilinear (QB) DAE form, $r_1 = 30$ POD modes to approximate the differential equations and $r_2 = 9$ POD modes to approximate the algebraic equations. (a) $\mathcal{D} = 0.162$ (stable case). (b) $\mathcal{D} = 0.167$ (unstable case).



(a) (b)

Figure 9.2. Model reduction results for tubular reactor of Kramer and Willcox (2019, Figure 6), showing the relative state errors for several different reduced models as a function of the reduced model dimension. Here $\mathcal{D} = 0.167$.

POD-DEIM model reduction of the original non-linear system (9.15)–(9.16). Figure 9.1 shows an illustrative result of a reduced model prediction of the temperature at the reactor exit compared to the full-order model prediction. In this case the reduced model is a POD model of the lifted quadratic DAEs (9.18)–(9.25) with $r_1 = 30$ POD modes to approximate the differential equations and $r_2 = 9$ POD modes to approximate the algebraic equations. Predictions for two different Damköhler numbers are shown, one in the stable regime and one that leads to a limit cycle oscillation.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Figure 9.2 shows the relative state errors as a function of reduced model dimension for $\mathcal{D} = 0.167$. Plot (a) compares POD-DEIM approximations with $r_d = 14, 16, 20$ DEIM interpolation points, the lifted quartic POD reduced model, and a POD baseline with no hyper-reduction. Plot (b) compares the lifted quartic POD model and lifted quadratic POD models using $r_2 = 12, 15, 18$ POD modes to approximate the algebraic equations. The levelling-off behaviour of the error in the plots is a typical result for non-linear model reduction. In plot (a) it is due to the error introduced by the DEIM approximation, which does not reduce if r_d is fixed. In plot (b) it is due to the error introduced by the approximation of the algebraic equations that appear in the quadratic lifting.

For this tubular reactor problem, hyper-reduction via DEIM is more effective than lifting. We see that the POD-DEIM models recover the performance of POD applied directly to the non-linear system (*i.e.* the inefficient baseline POD model as in (9.3)), provided that number of DEIM points r_d is chosen to be sufficiently large. For the results in Figure 9.2, choosing $r_d = r$ yields excellent results. In other complex PDE models, lifting has been found to have benefits over hyper-reduction, especially when the nature of the non-linear dynamics requires $r_d > r$. This includes rocket engine combustion (Swischuk *et al.* 2020, McQuarrie *et al.* 2021) and solidification in additive manufacturing (Khodabakhshi and Willcox 2021).

The lifting map $\mathcal{T}$ is not unique, and for any given non-linear system there may be a myriad of options to reformulate it in a polynomial form. These options often reveal trade-offs between the number of auxiliary variables, the order of the polynomial terms in the lifted systems, and other system properties such as sparsity and algebraic structure. Current lifting strategies are manual, based on inspection of the governing equations. The automated identification of optimal lifting strategies is an open research question. Lastly, we comment on the important question of translating lifted formulations into numerical implementations. In most situations it will not be viable, or even mathematically sound, to discretize the lifted equations. Discretization intricacies, such as stabilization terms, may destroy the polynomial structure. Numerical approximation errors will destroy the equivalence of the lifted state variables and will make the reverse lifting map ill-posed (*e.g.* in (9.17) we can see that issues may arise when the computations yield $\tilde{u}_5 \neq 1/\tilde{u}_2$). Therefore in the next section we discuss how the non-intrusive algorithms of OpInf can be employed to take lifting from a valuable thought exercise to a practical and scalable approach for non-linear model reduction.

9.4. Non-intrusive model reduction via Lift & Learn

The Lift & Learn method of Qian *et al.* (2020) combines lifting with OpInf to achieve model reduction of non-linear systems. The conditions (9.12)–(9.14) define a quadratic lifting map $\mathcal{T}$ that permits the non-linear governing equations (9.2) to be written in the quadratic form (9.13). As illustrated in Section 9.3, such a lifting map can be derived analytically for many classes of non-linear PDEs;

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Algorithm 5 Lift & Learn for non-intrusive model reduction of a non-linear system of PDEs (Qian *et al.* 2020).

```
Inputs: snapshots $\mathbf{U} \in \mathbb{R}^{n \times n_s}$, snapshot time derivatives $\dot{\mathbf{U}} \in \mathbb{R}^{n \times n_s}$, POD cumulative energy threshold $\kappa$
Analyse (pencil and paper) the form of the governing PDEs to identify a lifting map $\mathcal{T}$ that defines a polynomial (e.g. quadratic) lifting at the PDE level
Apply the lifting map to the snapshot data: $\tilde{\mathbf{U}} = \mathcal{T}(\mathbf{U})$, $\dot{\tilde{\mathbf{U}}} = \mathcal{T}(\dot{\mathbf{U}})$
Compute the SVD of $\tilde{\mathbf{U}}$
$r \leftarrow$ choose $r$ such that $(\sum_{i=1}^{r} \sigma_i^2)/(\sum_{i=1}^{n_s} \sigma_i^2) > \kappa$ {Determine the reduced model dimension}
$\mathbf{V} \leftarrow$ the $r$ leading left singular vectors of $\tilde{\mathbf{U}}$ {Compute the POD basis from the lifted snapshots}
$\hat{\mathbf{U}} \leftarrow \mathbf{V}^{\mathsf{T}}\tilde{\mathbf{U}}$ {Project lifted snapshots onto the $r$-dimensional POD subspace}
$\mathbf{R} \leftarrow \mathbf{V}^{\mathsf{T}}\dot{\tilde{\mathbf{U}}}$ {Time derivatives of the projected lifted snapshots}
$\lambda_1, \lambda_2 \leftarrow$ Set regularization parameters by hyper-parameter optimization
$\hat{\mathbf{A}}, \hat{\mathbf{H}} \leftarrow$ solve regularized OpInf regression (8.4) via $r$ independent least-squares problems
return $\hat{\mathbf{A}}, \hat{\mathbf{H}}, \mathcal{T}$
```

however, from a numerical standpoint it will be difficult/impossible to compute the discretized lifted operators needed to derive a reduced model of the lifted systems. The non-intrusive formulation of OpInf provides an alternative path: we define the quadratic (or other polynomial) lifting map $\mathcal{T}$ by analysing the PDEs, as discussed in Section 9.3. But rather than apply this lifting map to the PDE operators, we apply it to *snapshot data* computed using the original PDE simulation. The lifted snapshot data are then used within the OpInf approach of Algorithm 3 to learn the reduced model operators. In other words, *lifting* defines the polynomial form of the reduced model we seek, and OpInf permits us to *learn* that reduced model directly from snapshot data, without any modifications to the original full-order simulation. Algorithm 5 summarizes the Lift & Learn approach for the example of a system lifted to quadratic form.

Swischuk *et al.* (2019), McQuarrie *et al.* (2021) and Qian (2021) present detailed studies that combine variable transformations and OpInf for a single-injector combustion process. In all three works, the snapshot data are generated using the General Equation and Mesh Solver (GEMS), a finite-volume based CFD solver (Harvazinski *et al.* 2015). McQuarrie *et al.* (2021) consider the two-dimensional Navier–Stokes equations with a global one-step chemical reaction model. The GEMS snapshot data are transformed to provide snapshots of the learning variables $\tilde{u} = (p \;\; v_x \;\; v_y \;\; T \;\; \zeta \;\; c_1 \;\; c_2 \;\; c_3 \;\; c_4)$, which are respectively the pressure, x-velocity, y-velocity, temperature, specific volume and species molar concentrations of CH_4, O_2, H_2O and CO_2. As discussed in McQuarrie *et al.* (2021), this transformation makes many, but not all, terms in the governing equations quadratic.

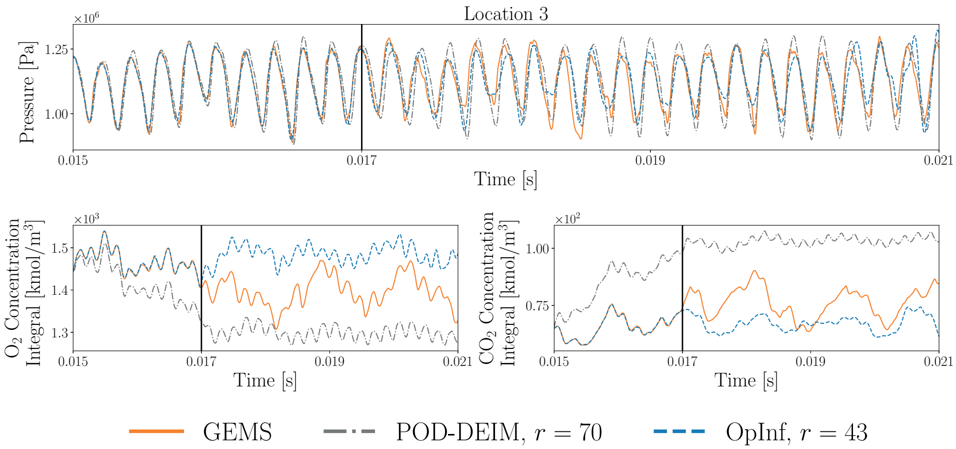


Figure 9.3. Model reduction results for the single-injector rocket combustion process of McQuarrie *et al.* (2021, Figure 6). Pressure trace at a monitor location (top) and spatially integrated O_2 and CO_2 molar concentrations (bottom), computed by GEMS (308 184 dof), a POD-DEIM reduced model (70 dof), and an OpInf reduced model (43 dof). POD basis computation and reduced model training used the first two milliseconds of data. (Figure reproduced with permission. Copyright © 2021 Taylor & Francis Ltd., www.tandfonline.com.)

Figures 9.3 and 9.4 show representative results comparing the performance of OpInf and POD-DEIM on this challenging problem. Both OpInf and POD-DEIM reduced models are able to accurately predict pressure dynamics, while transport-dominated quantities such as temperature and chemical concentrations are more challenging to capture in a reduced-order model. The non-intrusive OpInf approach compares favourably in predictive accuracy to the intrusive POD-DEIM approach for this problem. OpInf reduced models are faster to solve than POD-DEIM models by several orders of magnitude (less than one second compared to approximately 30 minutes) because DEIM requires repeated queries of the GEMS code to evaluate the non-linear function as in (9.6), whereas the OpInf quadratic reduced model is defined entirely by $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ and thus can be solved completely independently of GEMS.

A larger-scale three-dimensional combustion example is considered in Qian (2021). OpInf reduced models for the three-dimensional Navier–Stokes with flamelet progress variable model for the chemistry show six orders of magnitude speed-ups (reducing the dimension from 18M dof in the GEMS model to $\sim 10^2$ dof in the OpInf reduced models) while achieving accurate predictions of the pressure field and the large-scale structures in the temperature and chemistry variables. Representative results for this test problem are shown in Figure 9.5.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

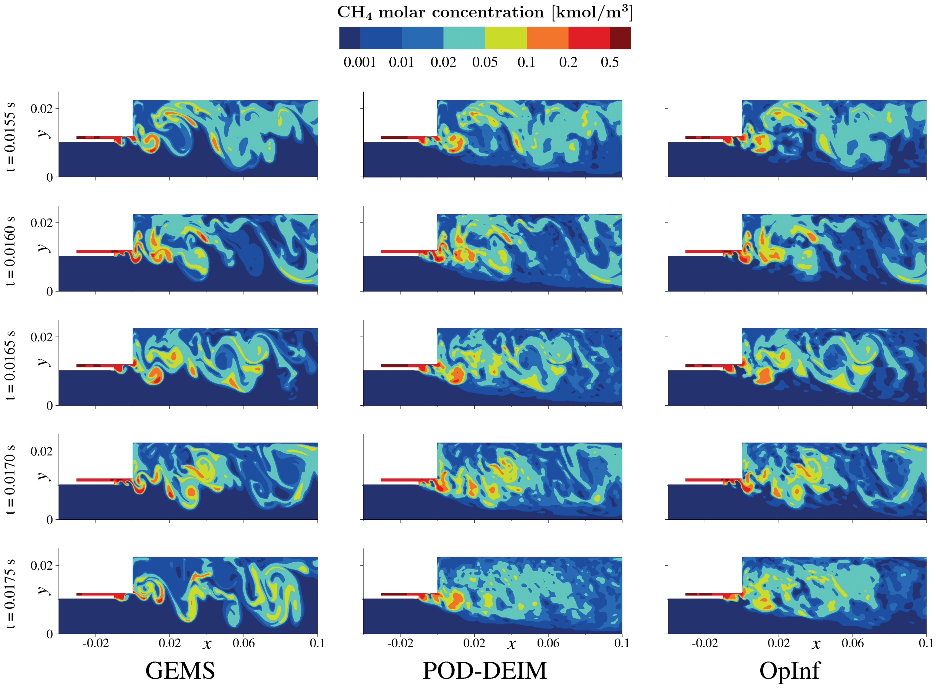


Figure 9.4. Model reduction results for the single-injector rocket combustion process of McQuarrie *et al.* (2021, Figure 8). Molar concentrations of CH$_4$ produced by GEMS (308 184 dof, left column), POD-DEIM (70 dof, middle column), and OpInf (43 dof, right column). POD basis computation and reduced model training used the first two milliseconds of data. (Figure reproduced with permission. Copyright © 2021 Taylor & Francis Ltd., www.tandfonline.com.)

We close this section by observing some of the relative advantages and disadvantages of the Lift & Learn approach compared to hyper-reduction approaches. Lift & Learn is informed by the structure of the lifted PDE, but makes no attempt to mimic the function of the full-order discretized model. Lift & Learn is agnostic to the numerical discretization, other than how the discretization is reflected in the snapshot data. In contrast, a hyper-reduction method such as DEIM interpolates the non-linear function through selective evaluations of $\mathbf{f}(\cdot)$ as in (9.5). In doing so, it explicitly embeds elements of the numerical discretization into the reduced model approximation. Neither approach is clearly better than the other – both introduce additional approximations to the model reduction process. Lift & Learn has a clear advantage in being non-intrusive, which in turn broadens its potential to be applied in industrial and other settings that use commercial and legacy codes.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

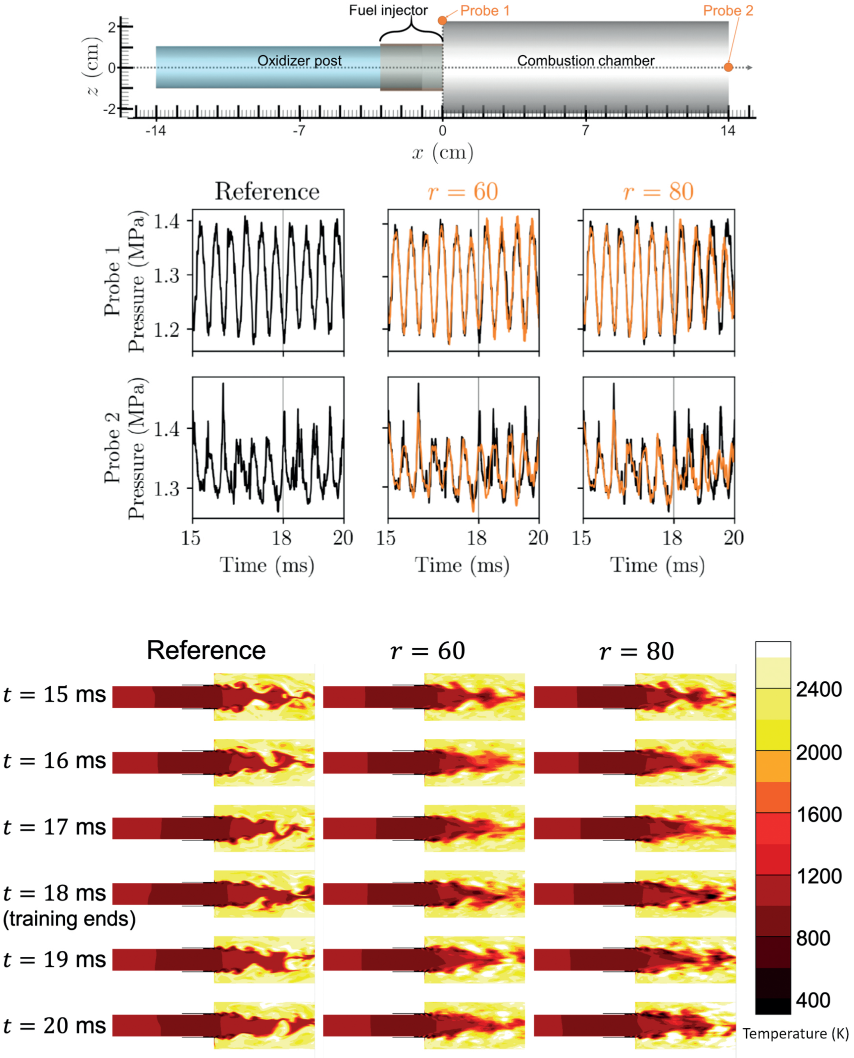


Figure 9.5. Model reduction results for the CVRC combustor from Qian (2021). OpInf reduced models are derived from snapshots generated using the General Equation and Mesh Solver (GEMS) to solve the Navier–Stokes equations with a flamelet model. Here we show representative results comparing GEMS (18M dof) with OpInf reduced models of dimension $r = 60$ and $r = 80$ for pressure traces and temperature fields.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Acknowledgements

We gratefully acknowledge the numerous contributions of students, postdoctoral researchers, research scientists and collaborators who have contributed to the work described here. While we cannot list them all, we wish to acknowledge by name the students and postdoctoral researchers whose results have appeared in this article: T. Bui-Thanh, T. Isaac, B. Kramer, S. McQuarrie, B. Peherstorfer, N. Petra, E. Qian and G. Stadler. We also gratefully acknowledge US Department of Energy grants SC0019303 and DE-SC0021239, US Air Force Office of Scientific Research grants FA9550-21-1-0084 and FA9550-17-1-0195, US Advanced Research Projects Agency–Energy grant DE-AR0001208 and US National Science Foundation cooperative agreement OAC-1854828.

References

S. S. Adavani and G. Biros (2008), Multigrid algorithms for inverse problems with linear parabolic PDE constraints, *SIAM J. Sci. Comput.* **31**, 369–397.

S. S. Adavani and G. Biros (2010), Fast algorithms for source identification problems with elliptic PDE constraints, *SIAM J. Imag. Sci.* **3**, 791–808.

V. Akçelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. J. Kim, J. Lopez, D. R. O'Hallaron, T. Tu and J. Urbanic (2003*a*), High resolution forward and inverse earthquake modeling on terascale computers, in *SC '03: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, IEEE, p. 52.

V. Akçelik, G. Biros and O. Ghattas (2002), Parallel multiscale Gauss–Newton–Krylov methods for inverse wave propagation, in *SC '02: Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, IEEE, pp. 41–41.

V. Akçelik, G. Biros, A. Drăgănescu, O. Ghattas, J. Hill and B. van Bloemen Waanders (2005), Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants, in *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, IEEE, p. 43.

V. Akçelik, G. Biros, O. Ghattas, J. Hill, D. Keyes and B. van Bloemen Waanders (2006), Parallel PDE-constrained optimization, in *Parallel Processing for Scientific Computing* (M. Heroux, P. Raghaven and H. Simon, eds), Society for Industrial and Applied Mathematics (SIAM).

V. Akçelik, G. Biros, O. Ghattas, K. R. Long and B. van Bloemen Waanders (2003*b*), A variational finite element method for source inversion for convective-diffusive transport, *Finite Elem. Anal. Des.* **39**, 683–705.

A. Alexanderian (2020), Optimal experimental design for Bayesian inverse problems governed by PDEs: A review. Available at arXiv:2005.12998.

A. Alexanderian and A. K. Saibaba (2018), Efficient D-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems, *SIAM J. Sci. Comput.* **40**, A2956–A2985.

A. Alexanderian, P. J. Gloor and O. Ghattas (2015), On Bayesian A- and D-optimal experimental designs in infinite dimensions, *Bayesian Anal.* **11**, 671–695.

A. Alexanderian, N. Petra, G. Stadler and O. Ghattas (2014), A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized ℓ_0-sparsification, *SIAM J. Sci. Comput.* **36**, A2122–A2148.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

A. Alexanderian, N. Petra, G. Stadler and O. Ghattas (2016), A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems, *SIAM J. Sci. Comput.* **38**, A243–A272.

A. Alexanderian, N. Petra, G. Stadler and O. Ghattas (2017), Mean-variance risk-averse optimal control of systems governed by PDEs with random parameter fields using quadratic approximations, *SIAM/ASA J. Uncertain. Quantif.* **5**, 1166–1192.

A. Alexanderian, N. Petra, G. Stadler and I. Sunseri (2021), Optimal design of large-scale Bayesian linear inverse problems under reducible model uncertainty: Good to know what you don't know, *SIAM/ASA J. Uncertain. Quantif.* **9**, 163–184.

N. Alger, V. Rao, A. Meyers, T. Bui-Thanh and O. Ghattas (2019), Scalable matrix-free adaptive product-convolution approximation for locally translation-invariant operators, *SIAM J. Sci. Comput.* **41**, A2296–A2328.

N. Alger, U. Villa, T. Bui-Thanh and O. Ghattas (2017), A data scalable augmented Lagrangian KKT preconditioner for large scale inverse problems, *SIAM J. Sci. Comput.* **39**, A2365–A2393.

A. Alghamdi, M. Hesse, J. Chen and O. Ghattas (2020), Bayesian poroelastic aquifer characterization from InSAR surface deformation data, I: Maximum *a posteriori* estimate, *Water Resour. Res.* **56**, e2020WR027391.

A. Alghamdi, M. Hesse, J. Chen, U. Villa and O. Ghattas (2021), Bayesian poroelastic aquifer characterization from InSAR surface deformation data, II: Quantifying the uncertainty, *Water Resour. Res.* Submitted.

E. L. Allgower, K. Böhmer, F. A. Potra and W. C. Rheinboldt (1986), A mesh-independence principle for operator equations and their discretizations, *SIAM J. Numer. Anal.* **23**, 160–169.

I. Ambartsumyan, W. Boukaram, T. Bui-Thanh, O. Ghattas, D. Keyes, G. Stadler, G. Turkiyyah and S. Zampini (2020), Hierarchical matrix approximations of Hessians arising in inverse problems governed by PDEs, *SIAM J. Sci. Comput.* **42**, A3397–A3426.

D. Amsallem and C. Farhat (2008), Interpolation method for the adaptation of reduced-order models to parameter changes and its application to aeroelasticity, *AIAA J.* **46**, 1803–1813.

D. Amsallem and C. Farhat (2011), An online method for interpolating linear parametric reduced-order models, *SIAM J. Sci. Comput.* **33**, 2169–2198.

D. Amsallem, J. Cortial, K. Carlberg and C. Farhat (2009), A method for interpolating on manifolds structural dynamics reduced-order models, *Internat. J. Numer. Methods Engng* **80**, 1241–1258.

D. Amsallem, M. J. Zahr and C. Farhat (2012), Nonlinear model order reduction based on local reduced-order bases, *Internat. J. Numer. Methods Engng* **92**, 891–916.

H. Antil, D. P. Kouri, M. Lacasse and D. Ridzal, eds (2018), *Frontiers in PDE-Constrained Optimization*, Springer.

A. C. Antoulas, C. A. Beattie and S. Gugercin (2020), *Interpolatory Methods for Model Reduction*, Society for Industrial and Applied Mathematics (SIAM).

E. Arian and S. Ta'asan (1999), Analysis of the Hessian for aerodynamic optimization: Inviscid flow, *Comput. Fluids* **28**, 853–877.

E. Arian, M. Fahl and E. Sachs (2002), Trust-region proper orthogonal decomposition models by optimization methods, in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 3300–3305.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

S. Arridge, P. Maass, O. Öktem and C.-B. Schönlieb (2019), Solving inverse problems using data-driven models, in *Acta Numerica*, Vol. 28, Cambridge University Press, pp. 1–174.

M. Asch, M. Bocquet and M. Nodet (2016), *Data Assimilation: Methods, Algorithms, and Applications*, Society for Industrial and Applied Mathematics (SIAM).

U. M. Ascher and E. Haber (2003), A multigrid method for distributed parameter estimation problems, *Electron. Trans. Numer. Anal.* **15**, 1–17.

R. C. Aster, B. Borchers and C. H. Thurbur (2013), *Parameter Estimation and Inverse Problems*, Academic Press.

P. Astrid, S. Weiland, K. Willcox and T. Backx (2008), Missing point estimation in models described by proper orthogonal decomposition, *IEEE Trans. Automat. Control* **53**, 2237–2251.

K. E. Atkinson (1997), *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press.

A. Attia, A. Alexanderian and A. K. Saibaba (2018), Goal-oriented optimal design of experiments for large-scale Bayesian linear inverse problems, *Inverse Problems* **34**, 095009.

O. Axelsson and J. Karatson (2007), Mesh independent superlinear PCG rates via compact-equivalent operators, *SIAM J. Numer. Anal.* **45**, 1495–1516.

Y. Ba, J. de Wiljes, D. S. Oliver and S. Reich (2021), Randomised maximum likelihood based posterior sampling. Available at arXiv:2101.03612.

O. Babaniyi, R. Nicholson, U. Villa and N. Petra (2021), Inferring the basal sliding coefficient field for the Stokes ice sheet model under rheological uncertainty, *Cryosphere* **15**, 1731–1750.

V. A. Badri Narayanan and N. Zabaras (2004), Stochastic inverse heat conduction using a spectral approach, *Internat. J. Numer. Methods Engng* **60**, 1569–1593.

E. Balas (2005), Projection, lifting and extended formulation in integer and combinatorial optimization, *Ann. Oper. Res.* **140**, 125.

H. T. Banks and K. Kunisch (1989), *Estimation Techniques for Distributed Parameter Systems*, Systems & Control: Foundations & Applications, Birkhäuser.

J. M. Bardsley (2018), *Computational Uncertainty Quantification for Inverse Problems*, Society for Industrial and Applied Mathematics (SIAM).

J. M. Bardsley, T. Cui, Y. M. Marzouk and Z. Wang (2020), Scalable optimization-based sampling on function space, *SIAM J. Sci. Comput.* **42**, A1317–A1347.

J. Bardsley, A. Solonen, H. Haario and M. Laine (2014), Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems, *SIAM J. Sci. Comput.* **36**, A1895–A1910.

A. Barker, T. Rees and M. Stoll (2016), A fast solver for an H_1 regularized PDE-constrained optimization problem, *Commun. Comput. Phys.* **19**, 143–167.

M. Barrault, Y. Maday, N. C. Nguyen and A. T. Patera (2004), An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations, *C. R. Math. Acad. Sci. Paris* **I**, 339–667.

O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders and J. Hill (2008), Hessian-based model reduction for large-scale systems with initial condition inputs, *Internat. J. Numer. Methods Engng* **73**, 844–868.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

A. Battermann and M. Heinkenschloss (1998), Preconditioners for Karush–Kuhn–Tucker systems arising in the optimal control of distributed systems, in *Optimal Control of Partial Differential Equations* (W. Desch, F. Kappel and K. Kunisch, eds), Vol. 126 of International Series of Numerical Mathematics, Birkhäuser, pp. 15–32.

A. Battermann and E. W. Sachs (2001), Block preconditioners for KKT systems in PDE-governed optimal control problems, in *Fast Solution of Discretized Optimization Problems (Berlin, 2000)* (K.-H. Hoffmann, R. H. W. Hoppe and V. Schulz, eds), Vol. 138 of International Series of Numerical Mathematics, Birkhäuser, pp. 1–18.

R. Becker and B. Vexler (2007), Optimal control of the convection–diffusion equation using stabilized finite element methods, *Numer. Math.* **106**, 349–367.

P. Benner and T. Breiten (2015), Two-sided projection methods for nonlinear model order reduction, *SIAM J. Sci. Comput.* **37**, B239–B260.

P. Benner and T. Stykel (2017), Model order reduction for differential-algebraic equations: A survey, in *Surveys in Differential-Algebraic Equations IV* (A. Ilchmann and T. Reis, eds), Springer, pp. 107–160.

P. Benner, P. Goyal, B. Kramer, B. Peherstorfer and K. Willcox (2020), Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, *Comput. Methods Appl. Mech. Engng* **372**, 113433.

P. Benner, S. Gugercin and K. Willcox (2015), A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.* **57**, 483–531.

P. Benner, A. Onwunta and M. Stoll (2016), Block-diagonal preconditioning for optimal control problems constrained by PDEs with uncertain inputs, *SIAM J. Matrix Anal. Appl.* **37**, 491–518.

M. Benning and M. Burger (2018), Modern regularization methods for inverse problems, in *Acta Numerica*, Vol. 27, Cambridge University Press, pp. 1–111.

M. Benzi, E. Haber and L. Taralli (2011), A preconditioning technique for a class of PDE-constrained optimization problems, *Adv. Comput. Math.* **35**, 149–173.

G. Berkooz, P. Holmes and J. L. Lumley (1993), The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.* **25**, 539–575.

A. Beskos, M. Girolami, S. Lan, P. E. Farrell and A. M. Stuart (2017), Geometric MCMC for infinite-dimensional inverse problems, *J. Comput. Phys.* **335**, 327–351.

L. T. Biegler, O. Ghattas, M. Heinkenschloss and B. van Bloemen Waanders, eds (2003), *Large-Scale PDE-Constrained Optimization*, Vol. 30 of Lecture Notes in Computational Science and Engineering, Springer.

L. T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes and B. van Bloemen Waanders, eds (2007), *Real-Time PDE-Constrained Optimization*, Society for Industrial and Applied Mathematics (SIAM).

G. Biros and G. Doğan (2008), A multilevel algorithm for inverse problems with elliptic PDE constraints, *Inverse Problems* **24**, 034010.

G. Biros and O. Ghattas (1999), Parallel Newton–Krylov methods for PDE-constrained optimization, in *SC '99: Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*, IEEE, pp. 28–28.

G. Biros and O. Ghattas (2005*a*), Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization, I: The Krylov–Schur solver, *SIAM J. Sci. Comput.* **27**, 687–713.

G. Biros and O. Ghattas (2005*b*), Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization, II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows, *SIAM J. Sci. Comput.* **27**, 714–739.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

A. Borzì (2003), Multigrid methods for parabolic distributed optimal control problems, *J. Comput. Appl. Math.* **157**, 365–382.

A. Borzì and R. Griesse (2005), Experiences with a space-time multigrid method for the optimal control of a chemical turbulence model, *Internat. J. Numer. Methods Fluids* **47**, 879–885.

A. Borzì and V. Schulz (2009), Multigrid methods for PDE optimization, *SIAM Rev.* **51**, 361–395.

A. Borzì and V. Schulz (2012), *Computational Optimization of Systems Governed by Partial Differential Equations*, Society for Industrial and Applied Mathematics (SIAM).

M. Braack (2009), Optimal control in fluid mechanics by finite elements with symmetric stabilization, *SIAM J. Control Optim.* **48**, 672–687.

T. Bui-Thanh and O. Ghattas (2012*a*), Analysis of the Hessian for inverse scattering problems, I: Inverse shape scattering of acoustic waves, *Inverse Problems* **28**, 055001.

T. Bui-Thanh and O. Ghattas (2012*b*), Analysis of the Hessian for inverse scattering problems, II: Inverse medium scattering of acoustic waves, *Inverse Problems* **28**, 055002.

T. Bui-Thanh and O. Ghattas (2013), Analysis of the Hessian for inverse scattering problems, III: Inverse medium scattering of electromagnetic waves, *Inverse Probl. Imaging* **7**, 1139–1155.

T. Bui-Thanh and M. A. Girolami (2014), Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo, *Inverse Problems* **30**, 114014.

T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler and L. C. Wilcox (2012*a*), Extreme-scale UQ for Bayesian inverse problems governed by PDEs, in *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, IEEE, pp. 1–11.

T. Bui-Thanh, M. Damodaran and K. Willcox (2004), Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, *AIAA J.* **42**, 1505–1516.

T. Bui-Thanh, O. Ghattas and D. Higdon (2012*b*), Adaptive Hessian-based nonstationary Gaussian process response surface method for probability density approximation with application to Bayesian solution of large-scale inverse problems, *SIAM J. Sci. Comput.* **34**, A2837–A2871.

T. Bui-Thanh, O. Ghattas, J. Martin and G. Stadler (2013), A computational framework for infinite-dimensional Bayesian inverse problems, I: The linearized case, with application to global seismic inversion, *SIAM J. Sci. Comput.* **35**, A2494–A2523.

T. Bui-Thanh, K. Willcox and O. Ghattas (2008*a*), Model reduction for large-scale systems with high-dimensional parametric input space, *SIAM J. Sci. Comput.* **30**, 3270–3288.

T. Bui-Thanh, K. Willcox and O. Ghattas (2008*b*), Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications, *AIAA J.* **46**, 2520–2529.

X. C. Cai and D. E. Keyes (2002), Nonlinearly preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* **24**, 183–200.

S. L. Campbell, I. C. F. Ipsen, C. T. Kelley and C. D. Meyer (1994), GMRES and the minimal polynomial, *BIT* **36**, 664–675.

K. Carlberg, C. Farhat, J. Cortial and D. Amsallem (2013), The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, *J. Comput. Phys.* **242**, 623–647.

S. Chaillat and G. Biros (2012), FaIMS: A fast algorithm for the inverse medium problem with multiple frequencies and multiple sources for the scalar Helmholtz equation, *J. Comput. Phys.* **231**, 4403–4421.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

S. Chaturantabut and D. Sorensen (2010), Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.* **32**, 2737–2764.

P. Chen and O. Ghattas (2019), Hessian-based sampling for high-dimensional model reduction, *Internat. J. Uncertain. Quantif.* **9**, 103–121.

P. Chen and O. Ghattas (2020*a*), Projected Stein variational gradient descent, in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)* (H. Larochelle *et al.*, eds), Vol. 33, Curran Associates, pp. 1947–1958. Available at https://proceedings.neurips.cc/paper/2020.

P. Chen and O. Ghattas (2020*b*), Stochastic optimization of heterogeneous epidemic models: Application to COVID-19 spread accounting for long-term care facilities. Preprint.

P. Chen and O. Ghattas (2020*c*), Taylor approximation for chance constrained optimization problems governed by partial differential equations with high-dimensional random parameters. Available at arXiv:2011.09985.

P. Chen and O. Ghattas (2021), Stein variational reduced basis Bayesian inversion, *SIAM J. Sci. Comput.* **43**, A1163–A1193.

P. Chen and C. Schwab (2015), Sparse-grid, reduced-basis Bayesian inversion, *Comput. Methods Appl. Mech. Engng* **297**, 84–115.

P. Chen and C. Schwab (2016*a*), Adaptive sparse grid model order reduction for fast Bayesian estimation and inversion, in *Sparse Grids and Applications (Stuttgart 2014)*, Vol. 109 of Lecture Notes in Computational Science and Engineering, Springer, pp. 1–27.

P. Chen and C. Schwab (2016*b*), Sparse-grid, reduced-basis Bayesian inversion: Nonaffine-parametric nonlinear equations, *J. Comput. Phys.* **316**, 470–503.

P. Chen, M. Haberman and O. Ghattas (2021), Optimal design of acoustic metamaterial cloaks under uncertainty, *J. Comput. Phys.* **431**, 110114.

P. Chen, U. Villa and O. Ghattas (2017), Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems, *Comput. Methods Appl. Mech. Engng* **327**, 147–172.

P. Chen, U. Villa and O. Ghattas (2019*a*), Taylor approximation and variance reduction for PDE-constrained optimal control under uncertainty, *J. Comput. Phys.* **385**, 163–186.

P. Chen, K. Wu, J. Chen, T. O'Leary-Roseberry and O. Ghattas (2019*b*), Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions, in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)* (H. Wallach *et al.*, eds), Curran Associates. Available at https://proceedings.neurips.cc/paper/2019.

S. S. Collis and M. Heinkenschloss (2002), Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. Report TR02–01, Department of Computational and Applied Mathematics, Rice University, Houston, TX.

D. Colton and R. Kress (2019), *Inverse Acoustic and Electromagnetic Scattering Theory*, Vol. 93 of Applied Mathematical Sciences, fourth edition, Springer.

B. Crestel, G. Stadler and O. Ghattas (2018), A comparative study of regularizations for joint inverse problems, *Inverse Problems* **35**, 024003.

T. Cui, K. J. H. Law and Y. M. Marzouk (2016), Dimension-independent likelihood-informed MCMC, *J. Comput. Phys.* **304**, 109–137.

T. Cui, J. Martin, Y. M. Marzouk, A. Solonen and A. Spantini (2014), Likelihood-informed dimension reduction for nonlinear inverse problems, *Inverse Problems* **30**, 114015.

T. Cui, Y. M. Marzouk and K. E. Willcox (2015), Data-driven model reduction for the Bayesian solution of inverse problems, *Internat. J. Numer. Methods Engng* **102**, 966–990.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Y. Daon and G. Stadler (2018), Mitigating the influence of the boundary on PDE-based covariance operators, *Inverse Probl. Imaging* **12**, 1083–1102.

M. Dashti and A. M. Stuart (2017), The Bayesian approach to inverse problems, in *Handbook of Uncertainty Quantification* (R. Ghanem, D. Higdon and H. Owhadi, eds), Springer, pp. 311–428.

J. C. De los Reyes (2015), *Numerical PDE-Constrained Optimization*, Springer.

J. Degroote, J. Vierendeels and K. Willcox (2010), Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis, *Internat. J. Numer. Methods Fluids* **63**, 207–230.

L. Demanet, P.-D. Létourneau, N. Boumal, H. Calandra, J. Chiu and S. Snelson (2012), Matrix probing: A randomized preconditioner for the wave-equation Hessian, *Appl. Comput. Harmon. Anal.* **32**, 155–168.

R. S. Dembo, S. C. Eisenstat and T. Steihaug (1982), Inexact Newton methods, *SIAM J. Numer. Anal.* **19**, 400–408.

G. Detommaso, T. Cui, Y. M. Marzouk, A. Spantini and R. Scheichl (2018), A Stein variational Newton method, in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)* (S. Bengio *et al.*, eds), Curran Associates, pp. 9187–9197.

T. Dreyer, B. Maar and V. Schulz (2000), Multigrid optimization in applications, *J. Comput. Appl. Math.* **120**, 67–84.

Z. Drmac and S. Gugercin (2016), A new selection operator for the discrete empirical interpolation method: Improved *a priori* error bound and extensions, *SIAM J. Sci. Comput.* **38**, A631–A648.

M. Drohmann, B. Haasdonk and M. Ohlberger (2012), Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM J. Sci. Comput.* **34**, 937–969.

A. Drăgănescu and T. Dupont (2008), Optimal order multilevel preconditioners for regularized ill-posed problems, *Math. Comp.* **77**, 2001–2038.

M. M. Dunlop (2019), Multiplicative noise in Bayesian inverse problems: Well-posedness and consistency of MAP estimators. Available at arXiv:1910.14632.

J. L. Eftang and B. Stamm (2012), Parameter multi-domain 'hp' empirical interpolation, *Internat. J. Numer. Methods Engng* **90**, 412–428.

S. C. Eisenstat and H. F. Walker (1996), Choosing the forcing terms in an inexact Newton method, *SIAM J. Sci. Comput.* **17**, 16–32.

M. Eldred, A. Giunta and S. Collis (2004), Second-order corrections for surrogate-based optimization with model hierarchies, in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA paper 2004-4457.

H. W. Engl, M. Hanke and A. Neubauer (1996), *Regularization of Inverse Problems*, Vol. 375 of Mathematics and its Applications, Springer.

I. Epanomeritakis, V. Akçelik, O. Ghattas and J. Bielak (2008), A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion, *Inverse Problems* **24**, 034015.

R. Everson and L. Sirovich (1995), The Karhunen–Loève procedure for gappy data, *J. Optical Soc. Amer.* **12**, 1657–1664.

C. Feng and Y. M. Marzouk (2019), A layered multiple importance sampling scheme for focused optimal Bayesian experimental design. Available at arXiv:1903.11187.

P. H. Flath (2013), Hessian-based response surface approximations for uncertainty quantification in large-scale statistical inverse problems, with applications to groundwater flow. PhD thesis, The University of Texas at Austin.

P. H. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders and O. Ghattas (2011), Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations, *SIAM J. Sci. Comput.* **33**, 407–432.

J. D. Flores (1993), The conjugate gradient method for solving Fredholm integral equations of the second kind, *Int. J. Comput. Math.* **48**, 77–94.

Z. Fortuna (1979), Some convergence properties of the conjugate gradient method in Hilbert space, *SIAM J. Numer. Anal.* **16**, 380–384.

D. Galbally, K. Fidkowski, K. Willcox and O. Ghattas (2010), Nonlinear model reduction for uncertainty quantification in large-scale inverse problems, *Internat. J. Numer. Methods Engng* **81**, 1581–1608.

A.-L. Gerner and K. Veroy (2012), Certified reduced basis methods for parametrized saddle point problems, *SIAM J. Sci. Comput.* **34**, A2812–A2836.

R. G. Ghanem and A. Doostan (2006), On the construction and analysis of stochastic models: Characterization and propagation of the errors associated with limited data, *J. Comput. Phys.* **217**, 63–81.

R. G. Ghanem and P. D. Spanos (1991), *Stochastic Finite Elements: A Spectral Approach*, Springer.

F. Ghavamian, P. Tiso and A. Simone (2017), POD–DEIM model order reduction for strain-softening viscoplasticity, *Comput. Methods Appl. Mech. Engng* **317**, 458–479.

A. Gholami, A. Mang and G. Biros (2016), An inverse problem formulation for parameter estimation of a reaction–diffusion model of low grade gliomas, *J. Math. Biol.* **72**, 409–433.

M. Girolami and B. Calderhead (2011), Riemann manifold Langevin and Hamiltonian Monte Carlo methods, *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **73**, 123–214.

A. A. Giunta, V. Balabanov, M. Kaufman, S. Burgee, B. Grossman, R. T. Haftka, W. H. Mason and L. T. Watson (1997), Variable-complexity response surface design of an HSCT configuration, in *Multidisciplinary Design Optimization: State of the Art* (N. M. Alexandrov and M. Y. Hussaini, eds), Society for Industrial and Applied Mathematics (SIAM).

W. R. Graham, J. Peraire and K. Y. Tang (1999), Optimal control of vortex shedding using low-order models, I: Open-loop model development, *Internat. J. Numer. Methods Engng* **44**, 945–972.

M. Grepl and A. Patera (2005), *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations, *ESAIM Math. Model. Numer. Anal.* **39**, 157–181.

A. Griewank (1992), Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, *Optim. Methods Software* **1**, 35–54.

A. Griewank (2003), A mathematical view of automatic differentiation, in *Acta Numerica*, Vol. 12, Cambridge University Press, pp. 321–398.

A. Griewank and A. Walther (2008), *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, second edition, Society for Industrial and Applied Mathematics (SIAM).

C. Gu (2011), QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* **30**, 1307–1320.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

S. Gugercin, A. C. Antoulas and C. A. Beattie (2008), $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems, *SIAM J. Matrix Anal. Appl.* **30**, 609–638.

M. D. Gunzburger (2003), *Perspectives in Flow Control and Optimization*, Society for Industrial and Applied Mathematics (SIAM).

M. D. Gunzburger, M. Heinkenschloss and H. K. Lee (2000), Solution of elliptic partial differential equations by an optimization based domain decomposition method, *Appl. Math. Comput.* **113**, 111–139.

B. Haasdonk and M. Ohlberger (2008), Reduced basis method for finite volume approximations of parametrized linear evolution equations, *ESAIM Math. Model. Numer. Anal.* **42**, 277–302.

B. Haasdonk and M. Ohlberger (2011), Efficient reduced models and *a posteriori* error estimation for parametrized dynamical systems by offline/online decomposition, *Math. Comput. Model. Dyn. Syst.* **17**, 145–161.

B. Haasdonk, M. Dihlmann and M. Ohlberger (2011), A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space, *Math. Comput. Model. Dyn. Syst.* **17**, 423–442.

E. Haber and U. Ascher (2001), Preconditioned all-at-once methods for large, sparse parameter estimation problems, *Inverse Problems* **17**, 1847–1864.

J. Hadamard (1923), *Lectures on the Cauchy Problem in Linear Partial Differential Equations*, Yale University Press.

W. W. Hager (2000), Runge–Kutta methods in optimal control and the transformed adjoint system, *Numer. Math.* **87**, 247–282.

N. Halko, P. G. Martinsson and J. A. Tropp (2011), Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.* **53**, 217–288.

M. Hanke (1995), *Conjugate Gradient Type Methods for Ill-Posed Problems*, Vol. 327 of Pitman Research Notes in Mathematics, Longman Scientific & Technical.

M. Hanke (2017), *A Taste of Inverse Problems: Basic Theory and Examples*, Society for Industrial and Applied Mathematics (SIAM).

P. C. Hansen (1998), *Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Society for Industrial and Applied Mathematics (SIAM).

M. E. Harvazinski, C. Huang, V. Sankaran, T. W. Feldman, W. E. Anderson, C. L. Merkle and D. G. Talley (2015), Coupling between hydrodynamics, acoustics, and heat release in a self-excited unstable combustor, *Phys. Fluids* **27**, 045102.

A. Hay, J. T. Borggaard and D. Pelletier (2009), Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition, *J. Fluid Mech.* **629**, 41–72.

P. Heimbach, C. Hill and R. Giering (2002), Automatic generation of efficient adjoint code for a parallel Navier–Stokes solver, in *Computational Science: ICCS 2002* (P. M. A. Sloot *et al.*, eds), Vol. 2330 of Lecture Notes in Computer Science, Springer, pp. 1019–1028.

R. F. Heinemann and A. B. Poore (1981), Multiplicity, stability, and oscillatory dynamics of the tubular reactor, *Chem. Engng Sci.* **36**, 1411–1419.

M. Heinkenschloss (1993), Mesh independence for nonlinear least squares problems with norm constraints, *SIAM J. Optim.* **3**, 81–117.

M. Heinkenschloss (2005), Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problems, *J. Comput. Appl. Math.* **173**, 169–198.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

M. Heinkenschloss and M. Herty (2007), A spatial domain decomposition method for parabolic optimal control problems, *J. Comput. Appl. Math.* **201**, 88–111.

M. Heinkenschloss and D. Leykekhman (2010), Local error estimates for SUPG solutions of advection-dominated elliptic linear-quadratic optimal control problems, *SIAM J. Numer. Anal.* **47**, 4607–4638.

M. Heinkenschloss and H. Nguyen (2006), Neumann–Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems, *SIAM J. Sci. Comput.* **28**, 1001–1028.

T. Helin and R. Kretschmann (2020), Non-asymptotic error estimates for the Laplace approximation in Bayesian inverse problems. Available at arXiv:2012.06603.

E. Herman, A. Alexanderian and A. K. Saibaba (2020), Randomization and reweighted ℓ_1-minimization for A-optimal design of linear inverse problems, *SIAM J. Sci. Comput.* **42**, A1714–A1740.

F. J. Herrmann, P. Moghaddam and C. C. Stolk (2008), Sparsity- and continuity-promoting seismic image recovery with curvelet frames, *Appl. Comput. Harmon. Anal.* **24**, 150–173.

R. Herzog and E. Sachs (2010), Preconditioned conjugate gradient method for optimal control problems with control and state constraints, *SIAM J. Matrix Anal. Appl.* **31**, 2291–2317.

R. Herzog and E. Sachs (2015), Superlinear convergence of Krylov subspace methods for self-adjoint problems in Hilbert space, *SIAM J. Numer. Anal.* **53**, 1304–1324.

M. Hesse and G. Stadler (2014), Joint inversion in coupled quasistatic poroelasticity, *J. Geophys. Research: Solid Earth* **119**, 1425–1445.

M. Hinze and S. Volkwein (2008), Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition, *Comput. Optim. Appl.* **39**, 319–345.

M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich (2009), *Optimization with PDE Constraints*, Vol. 23 of Mathematical Modelling: Theory and Applications, Springer.

H. Hotelling (1933), Analysis of a complex of statistical variables with principal components, *J. Educ. Psychol.* **24**, 417–441,498–520.

X. Huan and Y. M. Marzouk (2013), Simulation-based optimal Bayesian experimental design for nonlinear systems, *J. Comput. Phys.* **232**, 288–317.

X. Huan and Y. M. Marzouk (2014), Gradient-based stochastic optimization methods in Bayesian experimental design, *Internat. J. Uncertain. Quantif.* **4**, 479–510.

C. Huang, K. Duraisamy and C. L. Merkle (2019), Investigations and improvement of robustness of reduced-order models of reacting flow, *AIAA J.* **57**, 5377–5389.

C. Huang, J. Xu, K. Duraisamy and C. Merkle (2018), Exploration of reduced-order models for rocket combustion applications, in *AIAA Aerospace Sciences Meeting*. AIAA paper 2018-1183.

T. J. R. Hughes, L. P. Franca and M. Mallet (1986), A new finite element formulation for computational fluid dynamics, I: Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics, *Comput. Methods Appl. Mech. Engng* **54**, 223–234.

K. Hutter (1983), *Theoretical Glaciology*, Mathematical Approaches to Geophysics, Reidel.

T. Iliescu and Z. Wang (2014), Are the snapshot difference quotients needed in the proper orthogonal decomposition?, *SIAM J. Sci. Comput.* **36**, A1221–A1250.

T. Isaac, N. Petra, G. Stadler and O. Ghattas (2015), Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet, *J. Comput. Phys.* **296**, 348–368.

K. Ito and K. Kunisch (2008), *Lagrange Multiplier Approach to Variational Problems and Applications*, Society for Industrial and Applied Mathematics (SIAM).

J. Jagalur-Mohan and Y. M. Marzouk (2020), Batch greedy maximization of non-submodular functions: Guarantees and applications to experimental design. Available at arXiv:2006.04554.

I. Jolliffe (2005), Principal component analysis, in *Encyclopedia of Statistics in Behavioral Science*, Wiley.

I. Joughin, R. B. Alley and D. M. Holland (2012), Ice-sheet response to oceanic forcing, *Science* **338** (6111), 1172–1176.

M. Kaercher, S. Boyaval, M. A. Grepl and K. Veroy (2018), Reduced basis approximation and *a posteriori* error bounds for 4D-Var data assimilation, *Optim. Engng* **19**, 663–695.

J. Kaipio and E. Somersalo (2005), *Statistical and Computational Inverse Problems*, Vol. 160 of Applied Mathematical Sciences, Springer.

A. G. Kalmikov and P. Heimbach (2014), A Hessian-based method for uncertainty quantification in global ocean state estimation, *SIAM J. Sci. Comput.* **36**, S267–S295.

M. Kaufman, V. Balabanov, A. A. Giunta, B. Grossman, W. H. Mason, S. L. Burgee, R. T. Haftka and L. T. Watson (1996), Variable-complexity response surface approximations for wing structural weight in HSCT design, *Comput. Mech.* **18**, 112–126.

C. T. Kelley (1999), *Iterative Methods for Optimization*, Society for Industrial and Applied Mathematics (SIAM).

C. T. Kelley and E. W. Sachs (1991), Mesh independence of Newton-like methods for infinite dimensional problems, *J. Integral Equations Appl.* **3**, 549–573.

M. C. Kennedy and A. O'Hagan (2001), Bayesian calibration of computer models, *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **63**, 425–464.

E. H. Kerner (1981), Universal formats for nonlinear ordinary differential systems, *J. Math. Phys.* **22**, 1366–1371.

P. Khodabakhshi and K. Willcox (2021), Non-intrusive data-driven model reduction for differential algebraic equations derived from lifting transformations. Oden Institute Report 21-08, University of Texas at Austin.

A. Kirsch (2011), *An Introduction to the Mathematical Theory of Inverse Problems*, second edition, Springer.

T. G. Kolda and B. W. Bader (2009), Tensor decompositions and applications, *SIAM Rev.* **51**, 455–500.

M. Korda and I. Mezić (2018), Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica* **93**, 149–160.

D. D. Kosambi (1943), Statistics in function space, *J. Indian Math. Soc.* **7**, 76–88.

K. Koval, A. Alexanderian and G. Stadler (2020), Optimal experimental design under irreducible uncertainty for linear inverse problems governed by PDEs, *Inverse Problems* **36**, 075007.

B. Kramer (2020), Stability domains for quadratic-bilinear reduced-order models. Available at arXiv:2009.02769.

B. Kramer and K. Willcox (2019), Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, *AIAA J.* **57**, 2297–2307.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

B. Kramer and K. Willcox (2021), Balanced truncation model reduction for lifted nonlinear systems, in *Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas* (C. Beattie *et al.*, eds), Springer. To appear.

K. Kunisch and S. Volkwein (2001), Galerkin proper orthogonal decomposition methods for parabolic problems, *Numer. Math.* **90**, 117–148.

K. Kunisch and S. Volkwein (2010), Optimal snapshot location for computing POD basis functions, *ESAIM Math. Model. Numer. Anal.* **44**, 509–529.

J. N. Kutz, S. L. Brunton, B. W. Brunton and J. L. Proctor (2016), *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, Society for Industrial and Applied Mathematics (SIAM).

S. Lall, J. E. Marsden and S. Glavaski (2002), A subspace approach to balanced truncation for model reduction of nonlinear control systems, *Internat. J. Robust Nonlinear Control* **12**, 519–535.

G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich and S. Ulbrich, eds (2012), *Constrained Optimization and Optimal Control for Partial Differential Equations*, Birkhäuser.

D. Leykekhman (2012), Investigation of commutative properties of discontinuous Galerkin methods in PDE constrained optimal control problems, *J. Sci. Comput.* **53**, 483–511.

C. Lieberman, K. Willcox and O. Ghattas (2010), Parameter and state model reduction for large-scale statistical inverse problems, *SIAM J. Sci. Comput.* **32**, 2523–2542.

F. Lindgren, H. Rue and J. Lindström (2011), An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach, *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **73**, 423–498.

J. Liu and Z. Wang (2019), Non-commutative discretize-then-optimize algorithms for elliptic PDE-constrained optimal control problems, *J. Comput. Appl. Math.* **362**, 596–613.

Q. Liu and D. Wang (2016), Stein variational gradient descent: A general purpose Bayesian inference algorithm, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (D. D. Lee *et al.*, eds), Curran Associates, pp. 2378–2386.

M. Loève (1955), *Probability Theory*, Van Nostrand.

B. Lohmann and R. Eid (2009), Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models, in *Methoden und Anwendungen der Regelungstechnik: Erlangen-Münchener Workshops 2007 und 2008* (G. Roppencker and B. Lohmann, eds), Shaker, pp. 27–36.

J. L. Lumley (1967), The structure of inhomogeneous turbulent flow, in *Atmospheric Turbulence and Radio Wave Propagation* (A. M. Yaglom and V. I. Tartarsky, eds), Nauka, pp. 166–178.

H. V. Ly and H. T. Tran (2001), Modeling and control of physical processes using proper orthogonal decomposition, *J. Math. Comput. Model.* **33**, 223–236.

L. Mainini and K. Willcox (2015), Surrogate modeling approach to support real-time structural assessment and decision making, *AIAA J.* **53**, 1612–1626.

K.-A. Mardal, B. Nielsen and M. Nordaas (2017), Robust preconditioners for PDE-constrained optimization with limited observations, *BIT* **57**, 405–431.

J. Martin, L. C. Wilcox, C. Burstedde and O. Ghattas (2012), A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, *SIAM J. Sci. Comput.* **34**, A1460–A1487.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

Y. M. Marzouk and H. N. Najm (2009), Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, *J. Comput. Phys.* **228**, 1862–1902.

Y. M. Marzouk, T. Moselhy, M. Parno and A. Spantini (2016), Sampling via measure transport: An introduction, in *Handbook of Uncertainty Quantification* (R. Ghanem *et al.*, eds), Springer, pp. 1–41.

Y. M. Marzouk, H. N. Najm and L. A. Rahn (2007), Stochastic spectral methods for efficient Bayesian solution of inverse problems, *J. Comput. Phys.* **224**, 560–586.

G. P. McCormick (1976), Computability of global solutions to factorable nonconvex programs, I: Convex underestimating problems, *Math. Program.* **10**, 147–175.

S. A. McQuarrie, C. Huang and K. Willcox (2021), Data-driven reduced-order models via regularised operator inference for a single-injector combustion process, *J. Royal Soc. New Zealand* **51**, 194–211.

G. A. Meehl, T. F. Stocker, W. D. Collins, A. T. Friedlingstein, A. T. Gaye, J. M. Gregory, A. Kitoh, R. Knutti, J. M. Murphy, A. Noda, S. C. B. Raper, I. G. Watterson, A. J. Weaver and Z.-C. Zhao (2007), Global climate projections, in *Climate Change 2007: The Physical Science Basis* (S. D. Solomon *et al.*, eds), Cambridge University Press, pp. 747–845.

I. Mezić (2013), Analysis of fluid flows via spectral properties of the Koopman operator, *Annu. Rev. Fluid Mech.* **45**, 357–378.

T. A. E. Moselhy and Y. M. Marzouk (2012), Bayesian inference with optimal maps, *J. Comput. Phys.* **231**, 7815–7850.

J. L. Mueller and S. Siltanen (2012), *Linear and Nonlinear Inverse Problems with Practical Applications*, Society for Industrial and Applied Mathematics (SIAM).

U. Naumann (2012), *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*, Society for Industrial and Applied Mathematics (SIAM).

B. Nielsen and K.-A. Mardal (2010), Efficient preconditioners for optimality systems arising in connection with inverse problems, *SIAM J. Control Optim.* **48**, 5143–5177.

B. Nielsen and K.-A. Mardal (2012), Analysis of the minimum residual method applied to ill posed optimality systems, *SIAM J. Sci. Comput.* **35**, A785–A814.

J. Nocedal and S. J. Wright (2006), *Numerical Optimization*, second edition, Springer.

G. R. North, T. L. Bell, R. F. Cahalan and F. J. Moeng (1982), Sampling errors in the estimation of empirical orthogonal functions, *Mon. Weather Rev.* **110**, 699–706.

J. T. Oden, I. Babuška and D. Faghihi (2017), Predictive computational science: Computer predictions in the presence of uncertainty, in *Encyclopedia of Computational Mechanics*, second edition, Wiley Online Library, pp. 1–26.

T. O'Leary-Roseberry, U. Villa, P. Chen and O. Ghattas (2020), Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs. Available at arXiv:2011.15110.

D. S. Oliver (2017), Metropolized randomized maximum likelihood for improved sampling from multimodal distributions, *SIAM/ASA J. Uncertain. Quantif.* **5**, 259–277.

D. S. Oliver, H. He and A. C. Reynolds (1996), Conditioning permeability fields to pressure data, in *5th European Conference on the Mathematics of Oil Recovery (ECMOR V)*, European Association of Geoscientists & Engineers, pp. 1–11.

H. Panzer, J. Mohring, R. Eid and B. Lohmann (2010), Parametric model order reduction by matrix interpolation, *Automatisierungstechnik* **58**, 475–484.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

A. T. Patera and G. Rozza (2006), Reduced basis approximation and *a posteriori* error estimation for parametrized partial differential equations. Version 1.0, Copyright MIT.

W. S. B. Paterson and W. F. Budd (1982), Flow parameters for ice sheet modeling, *Cold Reg. Sci. Technol.* **6**, 175–177.

J. W. Pearson and A. Wathen (2012), A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, *Numer. Linear Algebra Appl.* **19**, 816–829.

J. W. Pearson, M. Stoll and A. Wathen (2012), Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, *SIAM J. Matrix Anal. Appl.* **33**, 1126–1152.

J. W. Pearson, M. Stoll and A. J. Wathen (2014), Preconditioners for state constrained optimal control problems with Moreau–Yosida penalty function, *Numer. Linear Algebra Appl.* **21**, 81–97.

B. Peherstorfer (2020), Sampling low-dimensional Markovian dynamics for preasymptotically recovering reduced models from data with operator inference, *SIAM J. Sci. Comput.* **42**, A3489–A3515.

B. Peherstorfer and K. Willcox (2016), Data-driven operator inference for nonintrusive projection-based model reduction, *Comput. Methods Appl. Mech. Engng* **306**, 196–215.

B. Peherstorfer, D. Butnaru, K. Willcox and H. J. Bungartz (2014), Localized discrete empirical interpolation method, *SIAM J. Sci. Comput.* **36**, A168–A192.

B. Peherstorfer, Z. Drmac and S. Gugercin (2020), Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points, *SIAM J. Sci. Comput.* **42**, A2837–A2864.

N. Petra, J. Martin, G. Stadler and O. Ghattas (2014), A computational framework for infinite-dimensional Bayesian inverse problems, II: Stochastic Newton MCMC with application to ice sheet flow inverse problems, *SIAM J. Sci. Comput.* **36**, A1525–A1555.

N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes and O. Ghattas (2012), An inexact Gauss–Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model, *J. Glaciology* **58**, 889–903.

C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. T. Patera and G. Turinici (2002), Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods, *J. Fluids Engng* **124**, 70–80.

E. Qian (2021), A scientific machine learning approach to learning reduced models for nonlinear partial differential equations. PhD thesis, Massachusetts Institute of Technology.

E. Qian, M. Grepl, K. Veroy and K. Willcox (2017), A certified trust region reduced basis approach to PDE-constrained optimization, *SIAM J. Sci. Comput.* **39**, S434–S460.

E. Qian, B. Kramer, B. Peherstorfer and K. Willcox (2020), Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems, *Phys. D* **406**, 132401.

C. E. Rasmussen and C. K. Williams (2006), *Gaussian Processes for Machine Learning*, MIT Press.

T. Rees and A. Wathen (2011), Preconditioning iterative methods for the optimal control of the Stokes equations, *SIAM J. Sci. Comput.* **33**, 2903–2926.

T. Rees, S. H. Dollar and A. J. Wathen (2010*a*), Optimal solvers for PDE-constrained optimization, *SIAM J. Sci. Comput.* **32**, 271–298.

T. Rees, M. Stoll and A. Wathen (2010*b*), All-at-once preconditioning in PDE-constrained optimization, *Kybernetika* **46**, 341–360.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

M. Rewienski and J. White (2003), A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* **22**, 155–170.

E. Rignot, J. Mouginot and B. Scheuchl (2011), Ice flow of the Antarctic ice sheet, *Science* **333** (6048), 1427–1430.

G. Rozza, D. B. P. Huynh and A. T. Patera (2008), Reduced basis approximation and *a posteriori* error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics, *Arch. Comput. Methods Engng* **15**, 229–275.

M. A. Savageau and E. O. Voit (1987), Recasting nonlinear differential equations as S-systems: A canonical nonlinear form, *Math. Biosci.* **87**, 83–115.

A. Schiela and S. Ulbrich (2014), Operator preconditioning for a class of inequality constrained optimal control problems, *SIAM J. Optim.* **24**, 435–466.

C. Schillings and C. Schwab (2013), Sparse, adaptive Smolyak quadratures for Bayesian inverse problems, *Inverse Problems* **29**, 065011.

C. Schillings, B. Sprungk and P. Wacker (2020), On the convergence of the Laplace approximation and noise-level-robustness of Laplace-based Monte Carlo methods for Bayesian inverse problems, *Numer. Math.* **145**, 915–971.

P. J. Schmid (2010), Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.* **656**, 5–28.

J. Schöberl and W. Zulehner (2007), Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems, *SIAM J. Matrix Anal. Appl.* **29**, 752–773.

J. Shewchuk (1994), An introduction to the conjugate gradient method without the agonizing pain. Report, Carnegie Mellon University.

T. W. Simpson, T. M. Mauery, J. J. Korte and F. Mistree (2001), Kriging models for global approximation in simulation-based multidisciplinary design optimization, *AIAA J.* **39**, 2233–2241.

L. Sirovich (1987), Turbulence and the dynamics of coherent structures, 1: Coherent structures, *Quart. Appl. Math.* **45**, 561–571.

R. C. Smith (2013), *Uncertainty Quantification: Theory, Implementation, and Applications*, Society for Industrial and Applied Mathematics (SIAM).

R. Ştefănescu, A. Sandu and I. M. Navon (2015), POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation, *J. Comput. Phys.* **295**, 569–595.

A. M. Stuart (2010), Inverse problems: A Bayesian perspective, in *Acta Numerica*, Vol. 19, Cambridge University Press, pp. 451–559.

S. Subramanian, K. Scheufele, M. Mehl and G. Biros (2020), Where did the tumor start? An inverse solver with sparse localization for tumor growth models, *Inverse Problems* **36**, 045006.

T. J. Sullivan (2015), *Introduction to Uncertainty Quantification*, Springer.

R. Swischuk, B. Kramer, C. Huang and K. Willcox (2020), Learning physics-based reduced-order models for a single-injector combustion process, *AIAA J.* **58**, 2658–2672.

R. Swischuk, L. Mainini, B. Peherstorfer and K. Willcox (2019), Projection-based model reduction: Formulations for physics-based machine learning, *Comput. Fluids* **179**, 704–717.

W. W. Symes (2009), The seismic reflection inverse problem, *Inverse Problems* **25**, 123008.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

S. Takacs and W. Zulehner (2011), Convergence analysis of multigrid methods with collective point smoothers for optimal control problems, *Comput. Vis. Sci.* **14**, 131–141.

A. Tarantola (2005), *Inverse Problem Theory and Methods for Model Parameter Estimation*, Society for Industrial and Applied Mathematics (SIAM).

L. Tenorio (2017), *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems*, Society for Industrial and Applied Mathematics (SIAM).

S. Tong, E. Vanden-Eijnden and G. Stadler (2020), Extreme event probability estimation using PDE-constrained optimization and large deviation theory, with application to tsunamis. Available at arXiv:2007.13930.

F. Tröltzsch (2010), *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Vol. 112 of Graduate Studies in Mathematics, American Mathematical Society.

M. Ulbrich (2011), *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, Society for Industrial and Applied Mathematics (SIAM).

J. Utke, L. Hascoët, P. Heimbach, C. Hill, P. Hovland and U. Naumann (2009), Toward adjoinable MPI, in *IEEE International Symposium on Parallel & Distributed Processing (IPDPS 2009)*, IEEE, pp. 1–8.

G. Venter, R. Haftka and J. H. Starnes (1998), Construction of response surface approximations for design optimization, *AIAA J.* **36**, 2242–2249.

K. Veroy and A. Patera (2005), Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: Rigorous reduced-basis *a posteriori* error bounds, *Internat. J. Numer. Methods Fluids* **47**, 773–788.

K. Veroy, C. Prud'homme, D. V. Rovas and A. T. Patera (2003), *A posteriori* error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, in *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*. AIAA paper 2003-3847.

U. Villa, N. Petra and O. Ghattas (2021), hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs, I: Deterministic inversion and linearized Bayesian inference, *ACM Trans. Math. Software* **47**, 16.

C. R. Vogel (2002), *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM).

K. Wang, T. Bui-Thanh and O. Ghattas (2018), A randomized maximum *a posteriori* method for posterior sampling of high dimensional nonlinear Bayesian inverse problems, *SIAM J. Sci. Comput.* **40**, A142–A171.

Q. Wang, J. S. Hesthaven and D. Ray (2019), Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem, *J. Comput. Phys.* **384**, 289–307.

Z. Wang, J. M. Bardsley, A. Solonen, T. Cui and Y. M. Marzouk (2017), Bayesian inverse problems with l_1 priors: A randomize-then-optimize approach, *SIAM J. Sci. Comput.* **39**, S140–S166.

L. C. Wilcox, G. Stadler, T. Bui-Thanh and O. Ghattas (2015), Discretely exact derivatives for hyperbolic PDE-constrained optimization problems discretized by the discontinuous Galerkin method, *J. Sci. Comput.* **63**, 138–162.

S. M. Wild, R. G. Regis and C. A. Shoemaker (2008), ORBIT: Optimization by radial basis function interpolation in trust-regions, *SIAM J. Sci. Comput.* **30**, 3197–3219.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press

K. Willcox and J. Peraire (2002), Balanced model reduction via the proper orthogonal decomposition, *AIAA J.* **40**, 2323–2330.

J. Worthen, G. Stadler, N. Petra, M. Gurnis and O. Ghattas (2014), Towards adjoint-based inversion for rheological parameters in nonlinear viscous mantle flow, *Phys. Earth Planet. Inter.* **234**, 23–34.

K. Wu, P. Chen and O. Ghattas (2020), A fast and scalable computational framework for large-scale and high-dimensional Bayesian optimal experimental design. Available at arXiv:2010.15196.

K. Wu, P. Chen and O. Ghattas (2021), A fast and scalable computational framework for goal-oriented linear Bayesian optimal experimental design: Application to optimal sensor placement. Available at arXiv:2102.06627.

D. Xiu and G. E. Karniadakis (2002), The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.* **24**, 619–644.

H. Yang and A. Veneziani (2017), Efficient estimation of cardiac conductivities via POD–DEIM model order reduction, *Appl. Numer. Math.* **115**, 180–199.

S. Yang, G. Stadler, R. Moser and O. Ghattas (2011), A shape Hessian-based boundary roughness analysis of Navier–Stokes flow, *SIAM J. Appl. Math.* **71**, 333–355.

H. Yücel, M. Heinkenschloss and B. Karasözen (2013), Distributed optimal control of diffusion–convection–reaction equations using discontinuous Galerkin methods, in *Numerical Mathematics and Advanced Applications 2011* (A. Cangiani *et al.*, eds), Springer, pp. 389–397.

Z. Zhang, E. Bader and K. Veroy (2016), A slack approach to reduced-basis approximation and error estimation for variational inequalities, *Comptes Rendus Mathématique* **354**, 283–289.

H. Zhu, S. Li, S. Fomel, G. Stadler and O. Ghattas (2016*a*), A Bayesian approach to estimate uncertainty for full waveform inversion with *a priori* information from depth migration, *Geophysics* **81**, R307–R323.

H. Zhu, N. Petra, G. Stadler, T. Isaac, T. J. R. Hughes and O. Ghattas (2016*b*), Inversion of geothermal heat flux in a thermomechanically coupled nonlinear Stokes ice sheet model, *Cryosphere* **10**, 1477–1494.

https://doi.org/10.1017/S0962492921000064 Published online by Cambridge University Press