Hierarchical Approaches for Efficient and Scalable Solution of Inverse Problems Governed by Partial Differential Equations

by

Tucker Andrew Hartland

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

in the

Graduate Division

of the

University of California, Merced

Committee in charge:

Noémi Petra, Chair
Arnold Kim
Roummel Marcia
Cosmin Petra
Georg Stadler

December 2022

The dissertation of Tucker Andrew Hartland, titled Hierarchical Approaches for Efficient and Scalable Solution of Inverse Problems Governed by Partial Differential Equations, is approved:

Chair ______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

University of California, Merced

Hierarchical Approaches for Efficient and Scalable Solution of Inverse Problems
Governed by Partial Differential Equations

Abstract

Hierarchical Approaches for Efficient and Scalable Solution of Inverse Problems
Governed by Partial Differential Equations

by

Tucker Andrew Hartland

Doctor of Philosophy in Applied Mathematics

University of California, Merced

Noémi Petra, Chair

Inverse problems governed by partial differential equations (PDEs) is a means to learn, from data, unknown or uncertain aspects of PDE-based mathematical models. PDE-based models often are constructed from scientific principles and are pervasive in science and engineering. The improvement of such a model can be accomplished by determining a "best" set of model parameters, or by reducing the uncertainty of the parameters that characterize the model. Ultimately, the goal is to improve the predictive capacity of such models and quantitatively understand the limitations of model-based predictions.

In this dissertation, we describe algorithmic approaches for the Newton-based solution of large-scale computational inverse problems governed by PDEs. We present and motivate hierarchical matrix approximations of the Hessian, a key-component of Newton's method, as a means to exploit localized sensitivities of underlying elliptic PDE operators and which perform well for inverse problems with highly-informative data. To circumvent the computational challenges associated with generating hierarchical matrix approximations of matrix-free operators, such as the Hessian, we describe a local point spread function methodology, whose computational cost is independent of the problem discretization. It is numerically demonstrated that hierarchical matrix approximations of the Hessian can be effective for large-scale ice sheet inverse problems with highly informative data and thin ice sheets. Lastly, we present a scalable means to solve PDE- and bound-constrained optimization problems by an interior-point filter line-search strategy that leverages performant algebraic multigrid linear solvers. Bound constraints arise naturally in many inverse problems as a means to enforce sign-definiteness as dictated by e.g., physical principles. The inclusion of bound constraints does add particular computational challenges such as non-smooth complementarity conditions as part of the conditions for optimality. The Newton

linear system solve strategy described in Chapter 4 is accelerated due to the inclusion of bound-constraints, wherein the performance of many other strategies is negatively impacted by such bound-constraints. A theoretical analysis is provided which demonstrates the algorithmic scaling of the approach. In addition, we demonstrate algorithmic and parallel scaling of the described approach by applying the framework to an example elliptic PDE- and bound-constrained problem.

To my family and soon-to-be wife.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vi**

1 Introduction and Background **1**
1.1 Full- and reduced-space methodologies 2
1.2 Adjoint-based gradients 3
1.3 Newton linear system 5
1.4 Role and structure of the Hessian in inverse problems governed by PDEs 7

2 HODLR Approximation of Hessians in Inverse Problems, with Application to Ice Sheet Model Initialization **10**
2.1 Introduction 10
2.2 Preliminaries 11
2.3 HODLR matrices in inverse problems governed by PDEs 15
2.4 Bayesian inverse ice sheet problems 18
2.5 Example I: Two-dimensional ISMIP-HOM benchmark 20
2.6 Example II: Humboldt glacier and Greenland ice sheet 25
2.7 Conclusion 29

3 A Point Spread Function Method to Further Enable Hierarchical Hessian Approximation in Inverse Problems **31**
3.1 Introduction 31
3.2 Preliminaries 35
3.3 Summary of algorithmic methodology 38
3.4 Numerical results 40
3.5 Conclusion 47

4 A Multigrid Interior-Point Approach for PDE- and Bound-Constrained Optimization **49**
4.1 Introduction 49

4.2 Interior-point methods for nonlinear PDE-constrained optimization 50
4.3 Scalable IP-Newton linear system solution computation 56
4.4 Problem setup 62
4.5 Numerical results 62
4.6 Conclusion 68

5 Conclusion 69

A Randomized Compression Algorithms and Additional Analysis 71
A.1 Randomized compression algorithms 71
A.2 Global HODLR approximation error from low-rank off-diagonal block approximation 77
A.3 Analysis of posterior-covariance error due to prior-preconditioned data-misfit error 78
A.4 Generalized eigenvalue ordering 79

Bibliography 81

List of Figures

2.1 Rank-structure illustration of a HODLR matrix. 14
2.2 Parameter-to-observable sensitivity cone illustration. 16
2.3 Schematic of a two-dimensional rectangular slab of ice. 21
2.4 Parameter and data reconstructions (Example I). 21
2.5 Prior and (HODLR Gaussianized) posterior samples and means. 22
2.6 Computational costs of HODLR and global low-rank Hessian approximation for various aspect ratios of the problem domain. 23
2.7 Computational costs of HODLR and global low-rank Hessian approximation for various discretized parameter dimensions. 24
2.8 Computational costs of HODLR and global low-rank Hessian approximation for various data set dimensions. 25
2.9 Parameter and data reconstructions Greenland ice-sheet and Humboldt glacier (Example II). 27
2.10 Block spectra of the Greenland and Humboldt data-misfit Hessians. 28
2.11 Computational costs of HODLR and global low-rank Greenland ice-sheet and Humboldt glacier Hessian approximation. 29
2.12 Off-diagonal block spectra of the Humboldt glacier data-misfit Hessian, wherein the basis for the Hessian matrix is determined by a kd-tree and by the finite element discretization 30

3.1 Batches of impulse responses for a Stokes data-misfit Gauss-Newton Hessian 33
3.2 Illustration of the local mean displacement invariance of the impulse responses 34
3.3 Inverse ice mountain problem geometry. 41
3.4 Inverse ice mountain true parameter and flow-field. 42
3.5 Ice mountain log-basal friction parameter field snapshots for various Newton iterations. 44
3.6 Left: convergence history of solving a Newton linear system by PCG using different preconditioners. Right: numerical study of the robustness, with respect to regularization, of various Newton linear system preconditioners. 46

3.7 Left: spectral clustering of the preconditioned Hessian for various preconditioning strategies. Right: condition number of the preconditioned Hessian and preconditioned Gauss-Newton Hessian for various preconditioning strategies. 47

4.1 Reconstructed fields from the interior-point multigrid based framework on an example elliptic PDE- and bound-constrained optimization problem. 66

4.2 Strong and weak scaling of the interior-point multigrid based framework on an example elliptic PDE- and bound-constrained optimization problem. 66

A.1 Global low-rank approximation of a steady-heat data-misfit Hessian using double-pass, symmetric double-pass and single-pass randomized algorithms. 76

List of Tables

2.1 Specifications for the discretized Humboldt glacier and Greenland ice-sheet problems (Example II). 26

3.1 Convergence history of solving the ice mountain Stokes inverse problem by inexact Newton-PCG with three different preconditioning strategies: point spread function with five batches of impulse responses, regularization and no preconditioning. 45

4.1 Block Gauss-Seidel interior-point Newton system preconditioner performance as the minimizer is approached. 63

4.2 Dependence of the block Gauss-Seidel interior-point Newton system preconditioner performance on level of mesh refinement. 64

4.3 Algorithmic scaling of the interior-point multigrid based framework on an example elliptic PDE- and bound-constrained optimization problem. 67

Acknowledgments

I have been so fortunate to have worked and learned from so many talented people that have done a great deal in helping shape me into a better computational mathematician. I am indebted to my advisor Dr. Noémi Petra for not only a seemingly immeasurable amount of scientifically valuable feedback but also her kindness and confidence in me which contributed to an environment that allowed me to flourish. I thank my committee member and coauthor Dr. Georg Stadler for having shared so many mathematical insights which often were in the form of simple and precise explanations for seemingly complex mathematical and physical concepts. It has been a great pleasure to work with the talented and kind Dr. Mauro Perego and Dr. Kim Liegeois, during both a summer internship experience and ongoing collaborations. The PyAlbany framework that they helped make possible is a essential reason for why this thesis contains continental-scale ice sheet inverse problems.

It has been inspiring to collaborate with Dr. Nick Alger, who so soon after earning his Ph.D. conducted high quality research and with an impressive ability to work independently. Through two summer internships and ongoing research my committee member and coauthor Dr. Cosmin Petra has influenced the ways in which I fundamentally approach research as a mathematician along with having significantly contributed to my understanding of numerical linear algebra and large-scale constrained optimization. I would also like to thank my committee members Dr. Arnold Kim and Dr. Roummel Marcia for their engagement and consistently having insightful questions and comments that helped me gain a variety of perspectives on the work contained in this thesis.

I could not have asked for a better academic home than that found in the UC Merced applied mathematics community. I sincerely appreciate the many interactions with current and former members of the department of applied mathematics at UC Merced. The unbounded support of my fiancé and graduate student office mate Tanya Tafolla dramatically impacted the richness of my graduate student life and through our many scientific conversations has helped me more than she appreciates. There are few for which mathematical discussions feel more like a natural and casual conversation that my mathematical brother Dr. Radoslav Vuchkov. I would also like to mention Dr. Ruanai Nicholson, Dr. Ki-Tae Kim, Dr. Olalekan Babaniyi, Irabiel Romero and Alex Ho for the many perspectives that they shared with me.

The UC Merced optimization and SAMPLe seminar series have been academically rich platforms that have refined my ability to present scientific findings. The content of this thesis has been made richer due to foundational courses such as: Optimization with Dr. Miguel Carreira-Perpinan, Numerical Analysis with Dr. Mayya Tokman and Parallel Computing with Dr. Dong Li. Having participated in the UC Merced SIAM Student Chapter helped me understand how we can actively enrich and intentionally shape the scientific communities that we are part of. I acknowledge the larger SIAM community which has allowed me to present my research at various conferences. The

2018 Gene Golub SIAM Summer School on Inverse Problems: Systematic Integration of Data with Uncertainty could not have come at a better time for me with regard to my growth as a computational inverse problems researcher. This summer school was a pivotal event in which I gained so much from the mature scientific perspectives that were shared and thus I acknowledge the organizers and instructors that made this summer school possible: Drs. Omar Ghattas, Youssef Marzouk, Matthew Parno, Noémi Petra, Georg Stadler and Umberto Villa. As I pass from one academic environment to another I acknowledge a few particularly influential people from my past undergraduate studies at CSU, Chico: Dr. Louis Buchholtz and Dr. Vladimir Rosenhaus.

I would not be where I am today without the love and support of my mother Lory, sister Abbey and grandma Mary. I appreciate the many questions that they often ask on topics that they are yet to be familiar with. Their questioning spirit reflects a natural scientific curiosity that they possess and which has left an indelible mark on me. I am grateful to my father Gary and my grandpa Lonnie whose quiet and persistent efforts has been a great example.

I am grateful for the financial support that I received as a graduate student at UC Merced which allowed me to focus and create a substantially better body of research. I received funding through NSF grants CAREER-1654311, DMS-1840265 and ACI-1550547, the DOE under contract DE-AC52-07NA27344, the DOE SciDAC project ProSPect and the Lawrence Livermore National Laboratory summer internship program. Regarding Chapter 2 of this thesis, I thank Trevor Hillebrand from Los Alamos National Laboratory for help set up the Humboldt and Greenland ice-sheet grids and datasets. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231, under NERSC award ERCAP0020130.

VITAE

Education

2017–2022 Ph.D. Applied Mathematics, University of California, Merced
2013–2016 B.Sc. Applied Mathematics and Physics, California State University, Chico

Internship Experiences

1. Computing Scholar Intern, May 2022–August 2022
   Center for Applied Scientific Computing,
   Advised by: Dr. Cosmin G. Petra,
   Lawrence Livermore National Laboratory, Livermore, CA
2. Graduate R&D Intern, May 2021–August 2021
   Computer Science Research Institute,
   Advised by: Dr. Mauro Perego,
   Sandia National Laboratories, Albuquerque, NM
3. Computing Scholar Intern, May 2020–August 2020
   Center for Applied Scientific Computing,
   Advised by: Dr. Cosmin G. Petra,
   Lawrence Livermore National Laboratory, Livermore, CA
4. Givens Scholar Intern, May 2018–August 2018
   Mathematics and Computer Science Division,
   Advised by: Drs. Oana Marin and Emil Constantinescu,
   Argonne National Laboratory, Lemont, IL

Teaching Experiences

1. Teaching Assistant
   University of California, Merced. Merced, CA.

i	Numerical Analysis (Math 130)	Fall 2020
ii	Vector Calculus (Math 23)	Fall 2018
iii	Probability and Statistics (Math 32)	Spring 2018
iv	Linear Algebra and Differential Equations (Math 24)	Fall 2017

2. Physics Laboratory Instructor
   California State University, Chico. Chico, CA.

i	General Physics I (Phys 202A)	Spring 2017
ii	General Physics II (Phys 202B)	Spring 2017
iii	Electricity and Magnetism (Phys 204B)	Spring 2017

3. Teaching Assistant
   Emory University. Atlanta, GA.

i	Intro to Physics I (Phys 141)	Fall 2016

Articles

1. *Hierarchical Off-Diagonal Low-Rank Approximation of Hessians in Inverse Problems, with Application to Ice Sheet Model Initialization.* Tucker Hartland, Georg Stadler, Mauro Perego, Kim Liegeois, Noémi Petra (submitted to Inverse Problems)

2. *PyAlbany: A Python Interface to the C++ Multiphysics Solver Albany.* Kim Liegeois, Mauro Perego, Tucker Hartland (submitted to the Journal of Computational and Applied Mathematics)

3. *Further Analysis of Multilevel Stein Variational Gradient Descent with an Application to the Bayesian Inference of Glacier Ice Models.* Terrence Alsup, Tucker Hartland, Benjamin Peherstorfer, Noémi Petra (submitted to Inverse Problems)

4. *Efficient Matrix-Free Point Spread Function of Operators with Locally Supported Non-Negative Integral Kernel, with Application to Hessians in PDE Constrained Inverse Problems.* Nick Alger, Tucker Hartland, Noémi Petra (to be submitted)

5. *An Interior-Point Multigrid Approach for a Class of Partial Differential Equation and Bound-Constrained Optimization Problems.* Tucker Hartland, Cosmin G. Petra, Noémi Petra, Jingyi Wang (in preparation)

6. *Indecomposable Vector-Valued Modular Forms and Periods of Modular Curves.* Luca Candelori, Tucker Hartland, Christopher Marks, Diego Yépez, Research in Number Theory, (2018) 4:17

7. *Two-Length-Scale Turbulence Model for Self-Similar Buoyancy-, Shock-, and Shear-Driven Mixing.* Brandon E. Morgan, Oleg Schilling, Tucker Hartland, Physical Review E 97, 013104 (2018)

8. *Linear Long Wave Propagation Over Discontinuous Submerged Shallow Water Topography.* Ravi Shankar, Yan Sheng, Megan Golbek, Tucker Hartland, Peter Gerrodette, Sergei Fomin, Vladimir Chugunov. Applied Mathematics and Computation 252 (2015) 27-44

Technical Reports

1. *Inversion of the Basal Sliding Coefficient for the Humboldt Glacier in an Uncertain Ice Sheet Model.* Tucker Hartland, Noémi Petra, Mauro Perego, Computer Science Research Institute 2021 Proceedings.

2. *Bound Constrained Partial Differential Inverse Problem Solution by the Semi-Smooth Newton Method.* Tucker Hartland, Cosmin G. Petra, Noémi Petra, Jingyi Wang, LLNL-TR-819385.

Contributed Talks

1. *Exploiting hierarchical low-rank structure in ice sheet inverse problems*, Tucker Hartland, Nick Alger, Georg Stadler, Omar Ghattas and Noémi Petra. SIAM Conference on the Mathematics of Data Science. San Diego, CA. September 2022.

2. *Exploiting hierarchical low-rank structure in ice sheet inverse problems using FEniCS and hIPPYlib*, Tucker Hartland, Nick Alger, Georg Stadler, Omar Ghattas and Noémi Petra. FEniCS 2022 Conference. San Diego, CA. August 2022.

3. *Hierarchical Hessian approximation for large-scale inverse problems governed by ice sheet model equations*, Tucker Hartland. CSU Fresno Applied Mathematics Junior Seminar Series. Online. April 2022.

4. *Efficient characterization of model uncertainty for ice sheet flow problems.* Tucker Hartland, Mauro Perego, Georg Stadler and Noémi Petra. SIAM Conference on Uncertainty Quantification. Hybrid. April 2022.

5. *Inversion for the basal sliding coefficient field under uncertainty for the Humboldt glacier.* Tucker Hartland, Noémi Petra and Mauro Perego. American Geophysical Union Fall Meeting. Online. December 2021.

6. *Inferring the basal coefficient in continental scale ice sheet models under additional uncertainty.* Tucker Hartland, Georg Stadler, Mauro Perego and Noémi Petra. DFP-SPP 1962 Summer School on Optimization under Uncertainty. Online. September 2021.

7. *Hierarchical off-diagonal Hessian approximation for Bayesian inference with application to ice sheet models.* Tucker Hartland, Georg Stadler, Mauro Perego, Kim Liegeois and Noémi Petra. International School on Inverse Problems in Geophysics on the shore of the Lario Lake. Online. August 2021.

8. *Hierarchical off-diagonal Hessian approximation for Bayesian inference with application to ice sheet models.* Tucker Hartland, Georg Stadler and Noémi Petra. SIAM Conference on Mathematical and Computational Issues in the Geosciences. Online. June 2021.

9. *Hierarchical off-diagonal low-rank approximation for Hessians in Bayesian inference with application to ice sheet models.* Tucker Hartland, Noémi Petra and Georg Stadler. SIAM Conference on Computational Science and Engineering. Online. March 2021.

10. *Hierarchical off-diagonal low-rank approximation for Hessians in Bayesian inference.* Tucker Hartland, Ruanui Nicholson and Noémi Petra. SIAM Conference on Computational Science and Engineering. Spokane, WA. February 2019.

11. *Three dimensional spaces of vector valued modular forms.* Tucker Hartland. Chico Topology Conference. Chico, CA. May 2016.

12. *Using the inviscid Burgers equation and the nonlinear shallow water equations to compute the time of wave breaking.* Tucker Hartland, Ravi Shankar and Sergei Fomin. Joint Mathematics Meetings. San Antonio, TX. January 2015.

Academic Service

1. Vice President 2021–present
   President 2020–2021
   Vice President 2019–2020
   Treasurer 2018–2019
   SIAM Student Chapter, University of California, Merced

2. Vice President 2014–2015
   Society of Physics Students, California State University, Chico Chapter

3. President 2012–2013
   Shasta College Science Club

Selected Training

1. *MFEM Community Workshop.* Online. October 2021, 2022.

2. *CMake Training.* Exascale Computing Project, Online, August 2021.

3. *Randomized Algorithms for Scientific Computing (RASC) Workshop.* Online, December 2020.

4. *MIT Uncertainty Quantification (MUQ) Workshop.* Online. March 2020.

5. *Inverse Problems: Systematic Integration of Data with Models under Uncertainty.* Gene Golub SIAM Summer School, Breckenridge, CO. June 2018.

Chapter 1

Introduction and Background

Newton's laws of motion is one systematic means by which knowledge of the system that an entity interacts with, can be encoded into a (system of) differential equation(s), and whose solution contain physics-based model predictions of the entity state. The information which the entity state refers to depends on the model. The entity state can e.g., be the location of an object through time, as in various time-dependent differential equations or the steady-state configuration of an extended body as in various time-independent partial differential equations [1]. However, having detailed system knowledge is uncommon, rather such knowledge is often incomplete and the knowledge that is available is always uncertain. In this thesis, it is assumed that this described system knowledge uncertainty is mathematically manifest through a model differential equation, of a known functional form, with an uncertain spatially distributed parameter field $m(x)$.

An inverse problem is a mathematical means to utilize data to update and potentially improve system knowledge. Inverse problems governed by PDEs can be formulated as PDE-constrained optimization problems [2], wherein one seeks to minimize a (regularized) data-misfit functional

$$\min J(u,m) = \underbrace{\frac{1}{2}\|\boldsymbol{\mathcal{B}}u - \boldsymbol{d}\|^2}_{\text{data-misfit}} + \underbrace{\frac{1}{2}R(m)}_{\text{regularization}}, \text{ such that } u = F(m). \tag{1.1}$$

The functional $J(u,m)$ measures a regularized discrepancy between observation data $\boldsymbol{d}$ and the associated model predictions $\boldsymbol{\mathcal{B}}u$, where $F(m)$ is the parameter-to-PDE-solution mapping, $u(x)$ is referred to as the state variable or PDE-solution, and $\boldsymbol{\mathcal{B}}$ is the observation operator which maps the PDE-solution to model predictions associated to the data $\boldsymbol{d}$. We note that Equation (1.1) is intentionally ambiguously defined and clarity is provided in Section 1.1. An example inverse problem that is described in more detail in Chapters 2 and 3 of this thesis, is that of determining a basal friction field from available surface velocity observation data in PDE-based ice sheet models [3, 4, 5]. The basal friction field locally quantifies the ease by which an ice sheet can slide across the ground which supports it. Model based predictions of sea level rise are influenced by the basal friction field as it impacts the predicted rate at which the ice flows out to sea. Determining this basal friction field and how it is impacted by data uncertainty, is then of critical importance.

Computing solutions of inverse problems governed by PDEs is often computationally intensive, especially so for parameter fields that vary continuously over a spatial domain Ω. The central challenges are that under mesh refinement, PDE-solution computation requires more computational resources and the discretized parameter dimension can be made arbitrarily large. It is then imperative that the employed numerical optimization methods [6] perform well in this large-scale setting, namely that they are efficient and not adversely impacted by changes in the underlying discretization of the differential equation model, that is the methods are *mesh-independent*. Due to these considerations, a natural choice for the optimization algorithm is one based on Newton's method, due to its rapid mesh-independent rate of convergence. Newton's method may converge rapidly, but it is not without challenges as each "outer" Newton optimization step requires the (inexact) solution of a system of linear equations. Such linear systems can be large, ill-conditioned and, as discussed in Section 1.3, there may not even be direct access to elements of the system matrix, which precludes the use of a significant number of linear system solution methods. In this thesis, we focus on the design of preconditioners for efficient Krylov-subspace based solution of Newton linear systems in inverse problems governed by PDEs. By reducing the cost to compute these linear system solutions further enables our ability to solve large-scale inverse problems governed by PDEs. The preconditioners that are discussed in this thesis are obtained by exploiting the rich underlying PDE-based problem structure that underlies inverse problems governed by PDEs.

1.1 Full- and reduced-space methodologies

We next focus on two distinct approaches to solve the optimization problem described by Equation (1.1), namely the so-called full- and reduced-space approaches. In the full-space approach the optimization problem

$$(u^\star, m^\star) = \arg\min_{(u,m)} J(u, m) = \frac{1}{2}\|\boldsymbol{\mathcal{B}}u - \boldsymbol{d}\|^2 + \frac{1}{2}R(m), \text{ such that } u = F(m), \quad (1.2)$$

is to determine the primal optimization variable $(u^\star, m^\star)$. During an iterative full-space optimization algorithm, the PDE constraint $u = F(m)$ is not necessarily satisfied, however the sequence of iterates generated by the optimization algorithm is expected to converge to a point where the constraint $u = F(m)$, is satisfied. In the reduced-space approach, the PDE-solution u, is formally eliminated, so that $u = u(m) := F(m)$, and the resultant optimization problem

$$m^\star = \arg\min_m J(m) = J(u(m), m) = \frac{1}{2}\|\boldsymbol{\mathcal{B}}u(m) - \boldsymbol{d}\|^2 + \frac{1}{2}R(m), \quad (1.3)$$

is absent of explicit constraints. With the reduced-space approach the primal optimization variable is only m, and so the optimum $m^\star$ is sought in a smaller space, thus the terminology reduced-space. A computational challenge of the reduced-space approach is that evaluating the reduced-space objective $J(m)$ requires the evaluation of the potentially expensive parameter-to-PDE-solution map F. An attractive feature of the reduced-space approach is that the objective functional is a merit functional, a critical object for ensuring robust convergence to a local minimizer and which provides a criterion to accept or reject trial steps based on whether a proposed trial step provides at least the minimum required reduction of the merit functional, hence indicating progress towards the optimizer.

Solving Newton linear systems for large-scale inverse problems, using either the reduced- or full-space approach is computationally challenging. Direct methods have unfavorable computational complexity and so their use is precluded for large-scale problems and instead here we pursue Krylov-subspace methods [7], for which effective preconditioners are needed. The full-space Newton system matrix (see Equation (1.19)) is a symmetric indefinite saddle-point matrix, and developing preconditioners for such matrices is challenging and an active area of research [8]. The reduced-space Newton system matrix i.e., Hessian (see Equation (1.24)) is a matrix-free, formally dense, operator for which there is a means to compute its action on vectors but not a cheap means to evaluate its entries. It is not feasible to use a significant number of preconditioning strategies for such matrix-free operators, which is an additional challenge for reduced-space methods.

1.2 Adjoint-based gradients

Here, for completeness we describe the adjoint-method [2, 9, 10] to compute a gradient of a scalar-valued function $\Phi(\boldsymbol{u}(\boldsymbol{m}), \boldsymbol{m})$, with respect to $\boldsymbol{m} \in \mathbb{R}^{n_1}$, wherein Φ both explicitly depends on a variable $\boldsymbol{m}$ and implicitly on $\boldsymbol{m}$ through an intermediate variable $\boldsymbol{u} \in \mathbb{R}^{n_2}$, that is defined by $\boldsymbol{c}(\boldsymbol{u}(\boldsymbol{m}), \boldsymbol{m}) = \boldsymbol{0}$. Here $\boldsymbol{c} : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{n_2}$, is assumed to have a nonsingular Jacobian $\partial \boldsymbol{c}/\partial \boldsymbol{u}$ and admits a unique solution $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{m})$ for each $\boldsymbol{m}$. We begin by differentiating on the manifold defined by $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{m})$,

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\boldsymbol{m}} = \frac{\partial \Phi}{\partial \boldsymbol{m}} + \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{m}}\right)^\top \frac{\partial \Phi}{\partial \boldsymbol{u}}, \tag{1.4}$$

$$\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}\boldsymbol{m}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}} + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{m}} = \boldsymbol{0}. \tag{1.5}$$

We note that one could follow a sensitivity approach [10], wherein each column of the sensitivity matrix $\mathrm{d}\boldsymbol{u}/\mathrm{d}\boldsymbol{m}$ is computed as

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{m}_i} = -\left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}_i}.$$

Each column of the sensitivity matrix requires the solution a system of linear equations with system matrix $\partial\boldsymbol{c}/\partial\boldsymbol{u}$, which is a computational bottleneck for large problems. The adjoint-based approach avoids the $\dim(\boldsymbol{u})$ dependent number of linear system solves with system matrix $\partial\boldsymbol{c}/\partial\boldsymbol{u}$ by first formally eliminating the sensitivity matrix

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{m}} = -\left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}}, \tag{1.6}$$

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\boldsymbol{m}} = \frac{\partial \Phi}{\partial \boldsymbol{m}} - \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}}\right)^{\top} \left[\left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\right)^{-\top} \frac{\partial \Phi}{\partial \boldsymbol{u}}\right], \tag{1.7}$$

and then defining the adjoint variable, $\boldsymbol{\lambda} \in \mathbb{R}^{n_2}$ as the unique solution of the adjoint equation

$$\left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\right)^{\top} \boldsymbol{\lambda} := \frac{\partial \Phi}{\partial \boldsymbol{u}}. \tag{1.8}$$

Having computed the adjoint variable, it is utilized to compute the adjoint-based gradient

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\boldsymbol{m}} = \frac{\partial \Phi}{\partial \boldsymbol{m}} - \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}}\right)^{\top} \boldsymbol{\lambda}. \tag{1.9}$$

We note that computing $\mathrm{d}\Phi/\mathrm{d}\boldsymbol{m}$, by the adjoint-based approach requires a single linear system solve with the adjoint system matrix $(\partial\boldsymbol{c}/\partial\boldsymbol{u})^{\top}$, as opposed, with the sensitivity approach, to n_2 linear system solves with the system matrix $\partial\boldsymbol{c}/\partial\boldsymbol{u}$. Lastly, we note that the conditions for stationarity of $\Phi(\boldsymbol{u}(\boldsymbol{m}), \boldsymbol{m})$

$$\frac{\partial \Phi}{\partial \boldsymbol{u}} - \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\right)^{\top} \boldsymbol{\lambda} = \boldsymbol{0}, \tag{1.10}$$

$$\frac{\partial \Phi}{\partial \boldsymbol{m}} - \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{m}}\right)^{\top} \boldsymbol{\lambda} = \boldsymbol{0}, \tag{1.11}$$

$$\boldsymbol{c} = \boldsymbol{0}, \tag{1.12}$$

are identical to that obtained by the Lagrangian formalism with Lagrangian $\mathcal{L}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}) = \Phi(\boldsymbol{u}, \boldsymbol{m}) - \boldsymbol{\lambda}^{\top}\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{m})$ and Lagrange multiplier $\boldsymbol{\lambda}$.

1.3 Newton linear system

We next detail the infinite-dimensional Newton linear system, which is a key component for Newton-based solution of inverse problem governed by PDEs. We note that by "infinite-dimensional" we refer to the determination of an object, a parameter field, that is an element of an infinite-dimensional function space. The forward problem is formulated as: for a given parameter $m(x) \in \mathcal{M}$, determine $u(x) \in \mathcal{V}$, such that

$$c(u, m, \lambda) = 0, \quad \forall \lambda \in \mathcal{V}^0. \tag{1.13}$$

Here $\mathcal{M}$, $\mathcal{V}$ and $\mathcal{V}^0$ are appropriately defined sets of functions and λ is a test function. $\mathcal{M}$ is the space of admissible functions $m(x)$, and $\mathcal{V}^0$ is a homogeneous version of the space $\mathcal{V}$, that is $(u+v) \in \mathcal{V}$ for any $u \in \mathcal{V}$ and $v \in \mathcal{V}^0$. Next, we invoke the Lagrangian formalism to solve

$$\min_{(u,m)\in\mathcal{V}\times\mathcal{M}} J(u,m) = \frac{1}{2}\|\boldsymbol{\mathcal{B}}u - \boldsymbol{d}\|^2 + \frac{1}{2}R(m), \text{ such that } c(u,m,\lambda) = 0, \quad \forall \lambda \in \mathcal{V}^0, \tag{1.14}$$

by first defining the Lagrangian functional

$$\mathcal{L}(u, m, \lambda) = J(u, m) + c(u, m, \lambda), \tag{1.15}$$

where λ now acts as a Lagrange multiplier associated to the partial differential equality constraint in Equation (1.13). The first-order necessary conditions for optimality are then

$$\mathcal{L}_u \tilde{u} = 0, \quad \forall \tilde{u} \in \mathcal{V}^0, \tag{1.16a}$$
$$\mathcal{L}_m \tilde{m} = 0, \quad \forall \tilde{m} \in \mathcal{M}, \tag{1.16b}$$
$$\mathcal{L}_\lambda \tilde{\lambda} = 0, \quad \forall \tilde{\lambda} \in \mathcal{V}^0, \tag{1.16c}$$

where $\mathcal{L}_u \tilde{u}$ indicates the variational derivative [11]

$$\mathcal{L}_u \tilde{u} := \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{L}(u + \varepsilon\tilde{u}, m, p)\right]_{\varepsilon=0}, \tag{1.17}$$

and $\mathcal{L}_m \tilde{m}$, $\mathcal{L}_\lambda \tilde{\lambda}$ are likewise defined. In the reduced-space approach Equation (1.16) should be understood as the state equation (1.16c) being a condition that the PDE constraint is satisfied, the adjoint Equation (1.16a) being a condition for determining the adjoint variable λ and $\mathcal{L}_m \tilde{m}$ then being the gradient which at an optimum is required to vanish by Equation (1.16b). The optimality conditions (see Equation (1.16)) is a system of nonlinear equations, which we choose to solve by Newton's method which requires the linearization [12]

$$\mathcal{L}_{u,u}(\tilde{u}, \hat{u}) + \mathcal{L}_{u,m}(\tilde{u}, \hat{m}) + \mathcal{L}_{u,\lambda}(\tilde{u}, \hat{\lambda}) = -\mathcal{L}_u \tilde{u}, \quad \forall \tilde{u} \in \mathcal{V}^0, \tag{1.18a}$$
$$\mathcal{L}_{m,u}(\tilde{m}, \hat{u}) + \mathcal{L}_{m,m}(\tilde{m}, \hat{m}) + \mathcal{L}_{m,\lambda}(\tilde{m}, \hat{\lambda}) = -\mathcal{L}_m \tilde{m}, \quad \forall \tilde{m} \in \mathcal{M}, \tag{1.18b}$$
$$\mathcal{L}_{\lambda,u}(\tilde{\lambda}, \hat{u}) + \mathcal{L}_{\lambda,m}(\tilde{\lambda}, \hat{m}) + \mathcal{L}_{\lambda,\lambda}(\tilde{\lambda}, \hat{\lambda}) = -\mathcal{L}_\lambda \tilde{\lambda}, \quad \forall \tilde{\lambda} \in \mathcal{V}^0, \tag{1.18c}$$

of Equation (1.16). Upon discretization by finite elements Equation (1.18) becomes

$$\begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{u}}^{\top} \\ \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{m}}^{\top} \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{J}_{\boldsymbol{m}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{m}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{\boldsymbol{u}} \\ \boldsymbol{r}_{\boldsymbol{m}} \\ \boldsymbol{r}_{\boldsymbol{\lambda}} \end{bmatrix}, \tag{1.19}$$

where e.g., $\hat{\boldsymbol{u}}$ is the vector of finite element coefficients of $\hat{u}(x) = \sum_i \hat{\boldsymbol{u}}_i \phi_i(x)$, and $\{\phi_i\}_i$ is a basis for the finite element space $\mathcal{V}_h^0 \subset \mathcal{V}^0$ defined over a mesh with characteristic discretization length scale h. Operators are likewise defined, for instance the blocks of the Hessian of the Lagrangian with respect to u and m are given by

$$(\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}})_{i,j} = \mathcal{L}_{u,u}(\phi_i, \phi_j), \quad (\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}})_{i,j} = \mathcal{L}_{u,m}(\phi_i, \psi_j), \tag{1.20}$$

$$(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}})_{i,j} = \mathcal{L}_{m,u}(\psi_i, \phi_j), \quad (\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}})_{i,j} = \mathcal{L}_{m,m}(\psi_i, \psi_j), \tag{1.21}$$

and $\boldsymbol{J}_{\boldsymbol{u}}$ and $\boldsymbol{J}_{\boldsymbol{m}}$ are the Jacobians of the PDE constraint with respect to u and m respectively

$$(\boldsymbol{J}_{\boldsymbol{u}})_{i,j} = c_{\lambda,u}(\phi_i, \phi_j), \quad (\boldsymbol{J}_{\boldsymbol{m}})_{i,j} = c_{\lambda,m}(\phi_i, \psi_j), \tag{1.22}$$

where $\{\psi_i\}_i$ is a basis for the space $\mathcal{M}_h \subset \mathcal{M}$, from which we approximate $\hat{m}(x) = \sum_i \hat{\boldsymbol{m}}_i \psi_i(x)$. Finally the linear system residuals are $(\boldsymbol{r}_{\boldsymbol{u}})_i = \mathcal{L}_u \phi_i$, $(\boldsymbol{r}_{\boldsymbol{m}})_i = \mathcal{L}_m \psi_i$ and $(\boldsymbol{r}_{\boldsymbol{\lambda}})_i = c(u, m, \phi_i)$. We note that Equation (1.19) is the Karush-Kuhn-Tucker (KKT) linear system [6], which is an indefinite saddle-point linear system [8]. In the reduced-space approach, the state u and adjoint λ, are computed so that $\boldsymbol{r}_{\boldsymbol{u}} = \boldsymbol{r}_{\boldsymbol{\lambda}} = 0$, and $\boldsymbol{r}_{\boldsymbol{m}}$ is the adjoint-based reduced-space gradient $\boldsymbol{g}$. Furthermore, the Hessian $\boldsymbol{H}$ of the reduced-space objective is obtained by taking the Schur complement of the $2,2$ block of the system matrix in Equation (1.19) and allows for the description of the reduced-space Newton system

$$\boldsymbol{H}\hat{\boldsymbol{m}} = -\boldsymbol{g}, \tag{1.23}$$

$$\boldsymbol{H} = \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}} + {\boldsymbol{J}_{\boldsymbol{m}}}^{\top} {\boldsymbol{J}_{\boldsymbol{u}}}^{-\top} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} {\boldsymbol{J}_{\boldsymbol{u}}}^{-1} \boldsymbol{J}_{\boldsymbol{m}} - (\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} {\boldsymbol{J}_{\boldsymbol{u}}}^{-1} \boldsymbol{J}_{\boldsymbol{m}} + {\boldsymbol{J}_{\boldsymbol{m}}}^{\top} {\boldsymbol{J}_{\boldsymbol{u}}}^{-\top} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}}). \tag{1.24}$$

Due to the Hessian component inverse operators ${\boldsymbol{J}_{\boldsymbol{u}}}^{-1}$, ${\boldsymbol{J}_{\boldsymbol{u}}}^{-\top}$, which lack sparsity, the reduced-space Hessian cannot be formed directly for large-scale problems, as an excessive number of floating point operations and CPU memory is required for the computation and storage of such inverses. It is then computationally necessary that the Hessian is kept in a matrix-free format, for which we only have access to it through matrix-vector products each of which costs two linearized PDE model solves, specifically the incremental-state and incremental-adjoint solves described in Algorithm 1.

Algorithm 1 Hessian-vector product computation.
Input: Hessian $\boldsymbol{H}$ and vector $\hat{\boldsymbol{m}}$.
Output: Hessian-vector product $\boldsymbol{y} = \boldsymbol{H}\hat{\boldsymbol{m}}$.

```
1: Solve $\boldsymbol{J}_{\boldsymbol{u}}\hat{\boldsymbol{u}} = -\boldsymbol{J}_{\boldsymbol{m}}\hat{\boldsymbol{m}}$ for $\hat{\boldsymbol{u}}$    {incremental-state equation}
2: Solve $\boldsymbol{J}_{\boldsymbol{u}}^{\top}\hat{\boldsymbol{\lambda}} = -(\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}\hat{\boldsymbol{u}} + \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}}\hat{\boldsymbol{m}})$ for $\hat{\boldsymbol{\lambda}}$    {incremental-adjoint equation}
3: Compute $\boldsymbol{y} = \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\hat{\boldsymbol{u}} + \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}\hat{\boldsymbol{m}} + \boldsymbol{J}_{\boldsymbol{m}}^{\top}\hat{\boldsymbol{\lambda}}$    {Hessian-vector product}
```

1.4 Role and structure of the Hessian in inverse problems governed by PDEs

The availability of the Hessian is an algorithmic prerequisite for reduced-space Newton-based optimization methods. The Hessian is useful for additional tasks, such as in the Bayesian statistical setting [13] wherein the Hessian locally characterizes uncertainty of the inverse problem solution that is due to both prior knowledge and data uncertainty. In particular, the Hessian inverse is the posterior covariance when the parameter-to-observable map $\boldsymbol{\mathcal{F}}$ is linear, the additive data noise and prior distributions are Gaussian [14]. Throughout this thesis the parameter-to-observable map $\boldsymbol{\mathcal{F}}$ is defined as the composition of a (potentially nonlinear) parameter-to-PDE-solution map $u(m)$ and a linear observation operator $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{F}}(m) = \boldsymbol{\mathcal{B}}u(m)$.

It is critical to have computationally efficient means to manipulate the Hessian, in order to determine important quantities such as Newton-based inverse problem solutions and posterior uncertainties [15]. However, as described in Section 1.3, the Hessian is a formally dense matrix that is only accessible through it's action on vectors. Given the matrix-free form of the Hessian, a natural choice for solving linear systems with Hessian system matrices is a Krylov-subspace method [7, 16], as such methods only require a means to compute system matrix-vector products. However, the number of Krylov-subspace iterations, needed to solve a linear system depends strongly on the system matrix eigenstructure; the number of Krylov-subspace iterations needed to determine the linear system solution is small when the system matrix has clustered eigenvalues.

The number of Krylov-subspace iterations is of particular relevance for large-scale inverse problems governed by PDEs as each iteration requires at least one Hessian-vector product and two linearized PDE solves are needed for each Hessian-vector product computation [9, 2]. Preconditioners are a standard means to reduce the number of Krylov iterations [7], by solving a mathematically equivalent preconditioned linear system, wherein more computation is generally required per Krylov iteration. A more established means to generate preconditioners for reduced-space Newton systems in inverse problems governed by PDEs is by exploiting the ill-posed nature of unregularized inverse problems [4, 17, 18, 14, 19]. The Hessian of the data misfit-component of the objective functional (Equation (1.1)) has eigenvalues that rapidly decay and furthermore the eigenvalues of the regularization preconditioned data-misfit Hessian has been proved in particular applications and shown numerically in others to be *mesh-independent*, e.g., [20] and references within. The convergence of the preconditioned conjugate-gradient (CG) Krylov-subspace method is consequentially mesh-independent as the number of iterations required to resolve the k most dominant modes of the error is proportional to k. Furthermore, one can invoke a low-rank approximation of the regularization preconditioned data-misfit Hessian to generate a preconditioner which further reduces the number of Kyrlov-subspace iterations. The cost to generate such low-rank approximations is also mesh-independent due to the aforementioned mesh-independence of the regularization preconditioned data-misfit Hessian eigenvalues.

While regularization preconditioning and methods that leverage low-rank data-misfit approximations of the Hessian are mesh-independent, the number of Hessian applies to complete a designated task may still be too large. The size of the aforementioned costs may be large because such methods, while they scale well with respect to the discretized parameter dimension, they do not scale well with respect to the amount of information contained in the data [21, Chapter 4]. An exorbitant amount of PDE solves may thus be needed to solve inverse problems with highly informative data. It is then crucially important to develop problem structure exploiting methods that are computationally effective for Hessians in inverse problems in the highly informative data regime [21]. In Chapter 2 of this thesis we exploit localized sensitivity structure, contained in a large class of problems that is due to the underlying PDEs, in large-scale ice sheet inverse problems by generating hierarchical off-diagonal low-rank (HODLR) matrix approximations of the data-misfit Hessian. It is found that in the data informative regime, the HODLR matrix format can be more suitable than the global low-rank format for approximating data-misfit Hessians.

While there has been some recent advances [22] in mesh-independent black-box hierarchical matrix compression algorithms, the cost to generate HODLR Hessian approximations by purely algebraic peeling methods, i.e., a class of methods that only require Hessian-vector products, is mesh-dependent. Hierarchical matrices with more complex hierarchical partitioning patterns are expected to be better suited to exploit localized sensitivity problem structure. Unfortunately, the cost to compute a hierarchical matrix approximation of a matrix-free operator by a peeling method grows with the complexity of the hierarchical partitioning. To overcome these challenges in Chapter 3 of this thesis we summarize a method, which obviates the need to compute hierarchical matrices by peeling. This method exploits local mean translation invariance of localized point spread functions associated to the Hessian. It is shown, in a ice sheet model inverse problem, that the number of PDE solves needed to solve the inverse problem by an inexact Newton-CG method can be significantly reduced with the outlined local point spread function formalism.

In Chapters 2 and 3 the parameter of interest, the basal friction field, is a quantity which can only be positive, as friction only acts to oppose motion and mathematically is sign-definite. Furthermore, parameters which violate specified inequality constraints can cause the optimization algorithm to fail, for instance in an inverse problem in which a distributed diffusion coefficient is sought in a steady-state heat equation. In this problem, having upper and lower bound constraints on the diffusion coefficient can bound the condition number of the matrix of the linear system whose solution is the state u. Including constraints on the parameter to account for sign-definiteness of a friction field makes the necessary conditions for optimality significantly more complex and so it not uncommon to reparametrize the problem and solve for the sign-indefinite *log* basal friction field. If one does not modify the regularization term to have greater complexity for the log basal friction parameter field then the reparametrized inverse problem is not equivalent to the inverse problem from which it was derived.

In Chapter 4 of this thesis ongoing research is presented on the computationally scalable solution of PDE-constrained optimization problems which include additional bound constraints. Here, both the optimizer and linear system solutions are computed in the full-space. We utilize a robust interior-point approach to handle the bound-constraints through a regularizing log-barrier penalty term. The log-barrier term penalizes the parameter from being too close to the problem bounds and is then homotopically relaxed therein permitting the parameter to approach the bounds along what is commonly referred to as the central path [6, Chapter 14]. The log-barrier subproblem is solved with a filter line-search so that a proposed iterate is accepted if it properly reduces either the partial differential equality constraint violation or log-barrier objective. To solve the Newton linear systems that arise in this context we utilize multigrid. Multigrid is a linear system solution technique that exploits the relationship between PDE-solutions for different levels of resolution as determined by the degree of mesh refinement. For large systems of equations which arise from the discretization of an elliptic PDE, multigrid is computationally attractive due to its optimal performance, with respect to computational complexity. Here we use a generalized minimal residual (GMRES) Krylov-subspace solver for the interior-point Newton linear system, with a block Gauss-Seidel preconditioner. The eigenvalues of the block Gauss-Seidel preconditioned interior-point Newton system matrix depend on the informativeness of the data, which is mesh-independent, and so the number of Krylov-subspace iterations is also mesh-independent. We leverage available mature multigrid solvers for the scalable application of the block Gauss-Seidel preconditioner. Finally with the given framework we achieve a scalable means to solve an example PDE- and bound-constrained optimization problem.

Chapter 2

HODLR Approximation of Hessians in Inverse Problems, with Application to Ice Sheet Model Initialization

2.1 Introduction

Model-based simulation of complex physical systems plays an essential role in understanding real world phenomena. These models are often characterized by partial differential equations (PDEs), and are typically subject to uncertainties stemming from unknown coefficient fields, constitutive laws, source terms, initial and/or boundary conditions, geometries, etc. When observation data exist, these parameters can be estimated by solving an inverse problem governed by the underlying model (e.g., PDE). It is well known that uncertainty is a fundamental feature of inverse problems, therefore in addition to inferring the parameters of interest, we need to quantify the uncertainty associated with this inference. This uncertainty quantification can be done via Bayesian inference. Solving Bayesian inverse problems governed by complex PDEs can be extremely challenging due to high-dimensional parameter spaces that stem from discretization of infinite-dimensional parameter fields and the need to repeatedly solve the underlying PDEs. To overcome these computational challenges, it is essential to exploit problem structure, when possible. For example, the underlying PDE-solution operator is often diffusive, that observation data may be sparse or only contain limited information about the parameter field. These particularities give rise to a low-rank structure in the second derivative of the data-misfit component of the inverse problem objective (or of the negative log likelihood), hereafter referred to as the data-misfit Hessian. In previous work [4, 15] this low-rank structure in the context of inverse ice sheet flow problems was exploited. However, for cases when this "low-rank" is in fact large, as is the case for many inverse problems of practical interest, where the observation data are highly informative, low-rank approximation is insufficient. In this article, we exploit the local sensitivity of model predictions to parameters, which gives rise to an off-diagonal low-rank structure. We do so by invoking hierarchical off-diagonal low-rank (HODLR) matrix approximations and detail how they can be used to reduce the computational cost to solve large-scale PDE-based inverse problems.

Related work Global low-rank approximation of Hessians in inverse problems have been successfully utilized in [4, 17, 18, 14, 19], with deterministic and randomized methods [23, 14] being available to generate said approximations. However, some problems, specifically those with highly informative observation data, are not amenable to global low-rank approximation, and thus other structure-exploiting strategies are needed such as those based on local translation invariance and localized sensitivities [24, 25, 26]. Here we focus on hierarchical low-rank methods for which convenient randomized methods are available [27, 28].

Hierarchical matrices have been demonstrated in [29, 30] to be an effective means to approximate covariance matrices associated to large-scale Gaussian processes. In [31], hierarchical matrix approximations with general hierarchical partitioning patterns are utilized for the construction of explicit representations of Hessian inverses. In one of the examples studied, the authors find that the diffusivity of the parameter-to-PDE-solution map and the informativeness of the observation data impact whether the data-misfit Hessian is more suited for compression with hierarchical or global low-rank formats. Here, we build on this study and focus on a specific inverse problem arising in land ice modeling.

Contributions The main contributions of this work are as follows. (1) We motivate the use of HODLR compression for data-misfit Hessians in inverse problems governed by PDEs, and present a detailed study for large-scale ice sheet inverse problems, such as the Greenland ice sheet. (2) We describe a strategy that leverages the fast manipulation of HODLR matrices to efficiently generate approximate samples from a Gaussian posterior distribution for uncertainty quantification. (3) We numerically study the influence of various problem setups on the off-diagonal low-rank structure of the data-misfit Hessian. The results show the effectiveness of the HODLR approximation for various problem scales including for a Greenland ice sheet inverse problem, which has a discretized parameter dimension of 3.2×10^5.

2.2 Preliminaries

In this section, we summarize background material regarding the solution of discretizations of infinite-dimensional inverse problems. We also briefly review HODLR matrices. Specifically, we define HODLR matrices, list some of their properties and summarize the computational complexities of computing symmetric HODLR matrix approximations of symmetric operators that are only available through their application on vectors. We refer to [32, 33] for a more thorough discussion of hierarchical matrices and to [28] for more detail on HODLR matrices.

2.2.1 Bayesian Inverse Problems

A means to account for uncertainty in parametric inference is to employ the Bayesian approach to inverse problems [34, 13, 35], which takes as input observation data $\boldsymbol{d}$, i.e., the data, prior knowledge of the parameter and a model for the likelihood of data conditional to β. Prior knowledge of the discretized parameter $\boldsymbol{\beta}$ is typically determined by the expertise of domain scientists and mathematically encoded in a probability density function $\pi_{\text{prior}}\left(\boldsymbol{\beta}\right)$. The likelihood $\pi\left(\boldsymbol{d}|\boldsymbol{\beta}\right)$ involves the data uncertainty and the mathematical model for the parameter-to-observable process. The solution of a Bayesian inverse problem is a probability density function for the discretized parameter $\boldsymbol{\beta}$, that is conditioned on the observation data according to Bayes formula

$$\pi_{\text{post}}\left(\boldsymbol{\beta}\right) = \pi\left(\boldsymbol{\beta}|\boldsymbol{d}\right) \propto \pi_{\text{prior}}\left(\boldsymbol{\beta}\right)\pi\left(\boldsymbol{d}|\boldsymbol{\beta}\right),$$

which provides a formal expression for the posterior distribution. Here, "$\propto$" means equal up to a normalization constant. For a problem with Gaussian prior $\mathcal{N}\left(\overline{\boldsymbol{\beta}}, \boldsymbol{\Gamma}_{\text{prior}}\right)$ and data noise $\boldsymbol{\eta}$ described by the zero mean Gaussian $\mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Gamma}_{\text{noise}}\right)$, $\pi_{\text{post}}(\cdot)$ has the following form

$$\pi_{\text{post}}\left(\boldsymbol{\beta}\right) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{\mathcal{F}}(\beta) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{1}{2}\|\boldsymbol{\beta} - \overline{\boldsymbol{\beta}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\right), \tag{2.1}$$

where $\boldsymbol{\mathcal{F}}$ is the parameter-to-observable map. The notation $\|\cdot\|_{\boldsymbol{A}}$ means that the norm is weighted with the positive-definite matrix $\boldsymbol{A}$, i.e., $\|\boldsymbol{v}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}}$. The parameter-to-PDE-solution map is typically nonlinear, and consequently the posterior distribution is not a Gaussian. One characteristic of the posterior distribution is the point at which it is maximized, or equivalently the point which minimizes the negative log-posterior, the so-called maximum a posteriori (MAP) point,

$$\boldsymbol{\beta}^\star := \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) := \frac{1}{2}\|\boldsymbol{\mathcal{F}}(\beta) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|\boldsymbol{\beta} - \overline{\boldsymbol{\beta}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}. \tag{2.2}$$

A means to compute the MAP point is to employ a (Gauss) Newton method for optimization [6], which critically relies on the availability of the (Gauss-Newton) Hessian. Since, J is defined implicitly in terms of the parameter-to-observable map, which involves a PDE-solution operator, we utilize the adjoint method [2, 9, 10] to compute it's gradient and Hessian-applies.

To fully explore posterior distributions, Markov chain Monte-Carlo (MCMC) techniques [36, 37] can be used. Such techniques require a proposal distribution that ideally approximates the posterior and is easily sampled from. One method to generate a

Gaussian proposal distribution is through the Laplace approximation of the posterior about $\boldsymbol{\beta}_k$ (or around the MAP point)

$$\tilde{\pi}_{\text{post}}\left(\boldsymbol{\beta}, \boldsymbol{\beta}_k\right) \propto \exp\left(-\frac{1}{2}\langle \boldsymbol{\beta} - \boldsymbol{\mu}_k, \boldsymbol{H}_k\left(\boldsymbol{\beta} - \boldsymbol{\mu}_k\right)\rangle_{\ell^2}\right), \boldsymbol{\mu}_k = \boldsymbol{\beta}_k - \boldsymbol{H}_k^{-1}\boldsymbol{g}_k,$$

where $\boldsymbol{g}_k$, $\boldsymbol{H}_k$ are the gradient and Hessian of the log-posterior $J(\boldsymbol{\beta})$ at $\boldsymbol{\beta}_k$. Another MCMC sampling approach is the generalized preconditioned Crank-Nicholson (gpCN) method [38, 39]. An attractive choice for the preconditioner is the Hessian at the MAP point, [40].

For these and other MCMC samplers, one typically needs to apply the inverse Hessian $\boldsymbol{H}_k^{-1}$ or its square root $\boldsymbol{H}_k^{-1/2}$ repeatedly and efficiently, which also motivates the study presented in this paper. In particular, in Section 2.3.2 we discuss how HODLR approximations can be used for the fast application of the Hessian square root.

2.2.2 Symmetric HODLR Matrices

A HODLR matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, is a matrix equipped with a depth $L \in \mathbb{N}$, hierarchical partitionings of the index set $\mathcal{I} = \{1, 2, \ldots, N\}$ into continguous subsets and low-rank off-diagonal blocks defined by the partition, which is described in greater detail in e.g. [28]. The block rank-structure of a HODLR matrix for various hierarchical depths is illustrated in Figure 2.1. An HODLR matrix must satisfy two additional properties.

1. The depth of the hierarchical partitioning scales with the logarithm of the size of the matrix, i.e.,
$$L = \mathcal{O}\left(\log\, N\right).$$
2. The maximum rank of each hierarchical level ℓ off-diagonal block, r_ℓ, is bounded above by a number r that is independent of the problem size N, for each level ℓ
$$\max_{1\leq \ell \leq L} r_\ell \leq r = \mathcal{O}\left(1\right).$$

Such matrices are referred to as data-sparse since the low-rank blocks allow for them to be represented computationally with less than $\mathcal{O}\left(N^2\right)$ floating point numbers. In particular, the storage of an HODLR matrix is $\mathcal{O}\left(N\, \log\, N\right)$, $\mathcal{O}(N\, \log\, N)$ flops are needed to compute a HODLR matrix-vector product [23], and $\mathcal{O}(N\, \log^2\, N)$ flops are required for direct methods to compute an inverse HODLR matrix-vector product [41], as well as square root and inverse square root matrix-vector products [42].

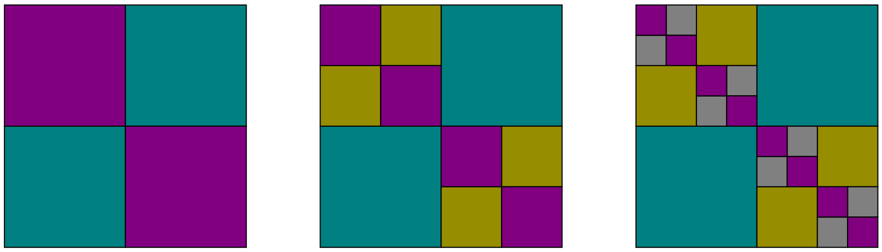


Figure 2.1: Rank-structure of a matrix $\boldsymbol{A}$ with hierarchical depths $L = 1$ (left), $L = 2$ (middle) and $L = 3$ (right). Off-diagonal blocks are assumed to be low-rank.

Compression We aim to generate HODLR approximations of data-misfit Hessians in inverse problems. For large-scale problems, the data-misfit Hessian is available only as a matrix-free operator. In order to construct HODLR approximations of symmetric matrix-free operators, we employ previously developed randomized linear algebraic routines which only require the matrix-free action on a limited number of random vectors with specified null entries, referred to as *structured* random vectors. The Hessian action on these structured random vectors is used to sample row and column spaces of off-diagonal Hessian submatrices and allow for randomized approximate truncated singular value decompositions of the aforementioned off-diagonal submatrices. More details can be found in the appendix, see Algorithm 5.

For the results that we present in Section 2.5 a rank-adaptive symmetric matrix-free [43, 44], hierarchical compression algorithm is utilized, that is based on [28]. A similar algorithm is presented in [45], wherein the hierarchical partitioning is more general and the low-rank blocks have nested bases. The rank-adaptivity provides a high probability means of resolving the off-diagonal blocks to a desired level of accuracy. By utilizing available matrix-vector product information and the Rayleigh quotient, a rank adaptive relative tolerance algorithm is made possible.

Computational Cost of Generating HODLR Approximations The number of matrix-vector products ζ, needed to compress a symmetric matrix using d oversampling vectors, into a level L HODLR matrix with off-diagonal ranks $\{r_\ell\}_{\ell=1}^{L}$ is given by

$$\zeta = 2\left(\langle r \rangle + d\right) L + N/2^L, \text{ where } \langle r \rangle := \frac{1}{L}\sum_{\ell=1}^{L} r_\ell. \tag{2.3}$$

Equation 2.3 can be understood from Algorithm 5 in Appendix A.1, as for each level ℓ one needs to compute $r_\ell + d$ Hessian vector products, in order to compute $\boldsymbol{Y}$ (line 7 of Algorithm 5) and $r_\ell + d$ Hessian vector products to compute $\boldsymbol{Z}$ (line 14 of Algorithm 5). The remaining $N/2^L$ Hessian vector products arise from the need to determine the diagonal subblocks, which is detailed in [23]. We note that with an adaptive procedure to determine an approximate basis $\boldsymbol{Q}$, such as that in [44], for a block matrix column space, the cost is reduced to $\zeta_{\text{adaptive}} = 2\left(\langle r \rangle + d/2\right) L + N/2^L$ but with the additional computational burden of extra orthogonalization routine calls. We note that $\zeta = \mathcal{O}(\log N)$ matrix-vector products are needed to generate an HODLR approximation of a matrix with HODLR structure. For sufficiently large problems HODLR compression is not expected to be more computationally efficient than global low-rank (LR) compression, as $\zeta^{\text{LR}} = r + d$, the number of matrix-vector products to generate a rank r compression by the single-pass algorithm [28] with d oversampling vectors is independent of the problems size. However, for problems of substantial size, we observe that the HODLR format does offer computational savings (see Section 2.6).

2.3 HODLR matrices in inverse problems governed by PDEs

Here, we illustrate why data-misfit Hessians in inverse problems governed by PDEs may contain numerically low-rank off-diagonal blocks, describe how one can permute parameters to expose this HODLR structure, and show how HODLR approximations can be leveraged to draw samples from Gaussian approximations of Bayesian posterior distributions.

2.3.1 Motivation

Consider the following data-misfit cost functional

$$J_{\text{misfit}}\left(\beta\right) := \frac{1}{2}\|\boldsymbol{\mathcal{F}}(\beta) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}, \quad \text{with} \quad \boldsymbol{\mathcal{F}}(\beta) = \boldsymbol{\mathcal{B}}u,$$

where $\boldsymbol{\mathcal{B}}$ linearly maps the PDE-solution $u = u(\beta)$, for the spatially-distributed parameter field β, to the model predictions associated to the data $\boldsymbol{d}$. Moreover, $\boldsymbol{\Gamma}_{\text{noise}}$ is the covariance matrix describing the Gaussian noise of the observational data. For illustration purposes, we assume that the parameter function β is defined on a region Γ_1 and the data $\boldsymbol{d}$ is observed on a region Γ_2, which may or may not be distinct. These quantities are related through the solution of the governing PDE and the measurement operator $\boldsymbol{\mathcal{B}}$. The characteristics of this relation depends on properties of the governing PDE. In the following, we assume that a spatially (or temporally) localized perturbation in the β field leads to a predominantly localized effect in the PDE-solution u, and thus the model predictions $\boldsymbol{\mathcal{B}}u$. This property is illustrated in Figure 2.2, where we use a sensitivity cone to illustrate the influence of a local perturbation in β, defined over Γ_1, on the PDE-solution u in Γ_2. It is well known that for an elliptic PDE, local perturbations influence the solution globally, but depending on the geometry of the domain and the equation, this global effect may rapidly

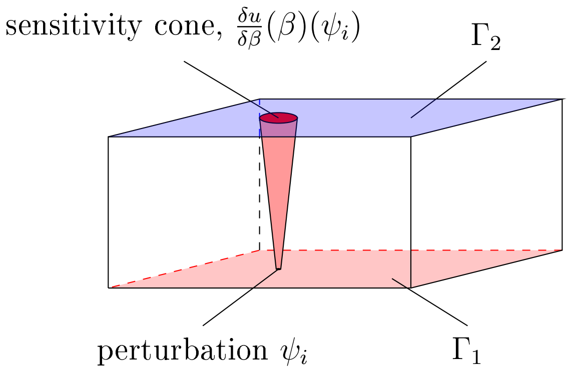

sensitivity cone, $\frac{\delta u}{\delta \beta}(\beta)(\psi_i)$
Γ_2
perturbation ψ_i
Γ_1

Figure 2.2: Sketch illustrating a case where the influence of changes in the parameter β on the PDE-solution u in Γ_2 is focused in a small area. To illustrate this, we show a sensitivity cone, i.e., the PDE solution u is predominantly impacted in a cone about the support of the localized parameter perturbation.

decay outside a subset of Γ_2 that captures the main effects of the perturbation. For instance, in a problem as in Figure 2.2, the influence of perturbations in β on u is likely to become more localized when the distance between Γ_1 and Γ_2 decreases.

We next discuss the relationship between properties of the PDE as discussed above and off-diagonal blocks in the Hessian matrix (or its Gauss-Newton variant). The data-misfit Hessian, i.e., the Hessian of the data-misfit part of the cost functional, can be derived using the adjoint method [2, 9, 10]. However, we find that the HODLR structure of the data-misfit Hessian is most easily seen by studying a formal expression of it in terms of the first and second order sensitivities $\delta u/\delta\beta$, $\delta^2 u/\delta\beta^2$

$$\begin{aligned} \frac{\delta^2}{\delta\beta^2} J_{\text{misfit}}\left(\beta\right)\left(\beta_1,\beta_2\right) &= \left(\boldsymbol{\mathcal{B}}u-\boldsymbol{d}\right)^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\left(\boldsymbol{\mathcal{B}}\frac{\delta^2 u}{\delta\beta^2}\left(\beta\right)\left(\beta_1,\beta_2\right)\right) + \\ &\quad \left(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\beta_1\right)\right)^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\left(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\beta_2\right)\right), \end{aligned}$$

where $\delta u/\delta\beta\left(\beta\right)\left(\beta_1\right)$ is the first variation [11] of u with respect to β in direction β_1, and $\delta^2 u/\delta\beta^2\left(\beta\right)\left(\beta_1,\beta_2\right)$ is the second variation of u with respect to β in directions β_1,β_2, that is,

$$\begin{aligned} \frac{\delta u}{\delta\beta}\left(\beta\right)\left(\beta_1\right) &:= \left[\frac{\mathrm{d}}{\mathrm{d}\epsilon}u\left(\beta+\epsilon\beta_1\right)\right]_{\epsilon=0}, \\ \frac{\delta^2 u}{\delta\beta^2}\left(\beta\right)\left(\beta_1,\beta_2\right) &:= \left[\frac{\mathrm{d}}{\mathrm{d}\epsilon}\frac{\delta u}{\delta\beta}\left(\beta+\epsilon\beta_2\right)\left(\beta_1\right)\right]_{\epsilon=0}. \end{aligned}$$

Upon discretizing β with finite elements we obtain the following formal expression for the (i,j)-entry of the data-misfit Hessian $\boldsymbol{H}_{\text{misfit}}$ and of the Gauss-Newton data-misfit

Hessian $\boldsymbol{H}_{\text{misfit}}^{\text{GN}}$

$$\left(\boldsymbol{H}_{\text{misfit}}\right)_{i,j} = \frac{\delta^2}{\delta\beta^2}\Big(J_{\text{misfit}}\left(\beta\right)\Big)\left(\psi_i, \psi_j\right), \tag{2.4}$$

$$\left(\boldsymbol{H}_{\text{misfit}}^{\text{GN}}\right)_{i,j} = \left(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\psi_i\right)\right)^{\top}\boldsymbol{\Gamma}_{\text{noise}}^{-1}\left(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\psi_j\right)\right), \tag{2.5}$$

where $\{\psi_j\}_{j=1}^{N}$ is a basis for the nodal finite-element space, which is used to approximate β.

When sensitivities are predominantly local as discussed above and when the support of two finite element basis functions ψ_i, ψ_j are well separated, the terms

$$\left(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\psi_i\right)\right)^{\top}\boldsymbol{\Gamma}_{\text{noise}}^{-1}\Big(\boldsymbol{\mathcal{B}}\frac{\delta u}{\delta\beta}\left(\beta\right)\left(\psi_j\right)\Big) \quad \text{and} \quad \boldsymbol{\mathcal{B}}\Big(\frac{\delta^2 u}{\delta\beta^2}\left(\beta\right)\left(\psi_i, \psi_j\right)\Big),$$

are rather small (assuming diagonally dominant noise covariance matrices). This is, e.g., due to $\boldsymbol{\mathcal{B}}\delta u/\delta\beta(\beta)(\psi_i)$ having small values when $\boldsymbol{\mathcal{B}}\delta u/\delta\beta(\beta)(\psi_j)$ is large. Now, let $\mathcal{I}, \mathcal{J}$ be disjoint index subsets of $\{1, 2, \ldots, N\}$, then the entries in the matrix block $\{(\boldsymbol{H}_{\text{misfit}})_{i\in\mathcal{I}, j\in\mathcal{J}}\}$ of the data-misfit Hessian are relatively small whenever $\cup_{i\in\mathcal{I}}\text{supp}\left(\psi_i\right)$ and $\cup_{j\in\mathcal{J}}\text{supp}\left(\psi_j\right)$ are well separated. Such Hessian blocks are well suited for approximation by low-rank matrices. When the degrees of freedom corresponding to the finite element basis functions ψ_i are ordered such that $\mathcal{I}, \mathcal{J}$ are contiguous, $(\boldsymbol{H}_{\text{misfit}})_{\mathcal{I},\mathcal{J}}$ is an off-diagonal subblock of $\boldsymbol{H}_{\text{misfit}}$ and $\boldsymbol{H}_{\text{misfit}}$ tends to have HODLR structure as defined in Section 2.2.2. The Gauss-Newton data-misfit Hessian may have HODLR structure for the same reasons. In both cases, the order of the basis functions and thus the degrees of freedom influence this structure. Ideally, one wants an order that maintains locality, i.e., consecutive indices correspond to basis functions that are close to each other, and as a consequence, basis function with significantly different indices are far from each other such that the corresponding off-diagonal blocks have small entries and can be well approximated using a low-rank matrix approximation. We defer to Section 2.6.2 for a discussion of methods and numerical experiments regarding the order of the degrees of freedom.

2.3.2 Application of HODLR structure for fast sampling of Gaussian posterior approximations

In [15], the following expressions of the Gaussianized posterior covariance are provided,

$$\begin{aligned}
\boldsymbol{\Gamma}_{\text{post}} &= \left(\boldsymbol{H}_{\text{misfit}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1} = \boldsymbol{\Gamma}_{\text{prior}}^{1/2}\left(\boldsymbol{H}'_{\text{misfit}} + \boldsymbol{I}\right)^{-1}\boldsymbol{\Gamma}_{\text{prior}}^{\top/2}, \\
\boldsymbol{H}'_{\text{misfit}} &:= \boldsymbol{\Gamma}_{\text{prior}}^{\top/2}\boldsymbol{H}_{\text{misfit}}\boldsymbol{\Gamma}_{\text{prior}}^{1/2}, \\
\boldsymbol{\Gamma}_{\text{post}}^{1/2} &= \boldsymbol{\Gamma}_{\text{prior}}^{1/2}\left(\boldsymbol{H}'_{\text{misfit}} + \boldsymbol{I}\right)^{-1/2},
\end{aligned}$$

where the matrix square-root $\boldsymbol{A}^{1/2}$ is such that $\boldsymbol{A} = \boldsymbol{A}^{1/2}\left(\boldsymbol{A}^{1/2}\right)^{\top}$. For Bayesian inverse problems with a parameter field that is distributed spatially over a bounded subset of $\mathbb{R}^m$, $m = 2, 3$, a reasonable choice is to use the square of an inverse elliptic PDE operator for the prior covariance [35], which permits a means of obtaining a symmetric square root of $\boldsymbol{\Gamma}_{\text{prior}}$. In previous works such as [4, 17, 18, 14, 19], the prior-preconditioned data-misfit Hessian $\boldsymbol{H}'_{\text{misfit}}$, was approximated by global low-rank compression. This strategy provides an efficient means of approximating the posterior covariance matrix in inverse problems with data sets that contain sufficiently small amounts of information. Here we propose to exploit HODLR problem structure and generate approximate posterior covariance matrices by HODLR approximations of the prior-preconditioned data-misfit $\tilde{\boldsymbol{H}}'_{\text{misfit}}$, see Appendix A.3 for an analysis on how such an approximation impacts the accuracy of the approximate posterior covariance

$$\tilde{\boldsymbol{\Gamma}}_{\text{post}} = \boldsymbol{\Gamma}_{\text{prior}}^{1/2}\left(\tilde{\boldsymbol{H}}'_{\text{misfit}} + \boldsymbol{I}\right)^{-1}\boldsymbol{\Gamma}_{\text{prior}}^{\top/2}.$$

A symmetric square-root factorization of $\tilde{\boldsymbol{H}}'_{\text{misfit}}+\boldsymbol{I}$ is then generated with $\mathcal{O}\left(N \log^2 N\right)$ flops [42]. The symmetric factorization allows for a $\mathcal{O}\left(N \log N\right)$ means of applying both the square root and inverse square root.

2.4 Bayesian inverse ice sheet problems

The simulation of the dynamics of ice sheets (e.g., the Greenland or Antarctic ice sheets) is an important component of coupled climate simulations. Such simulations require estimation of a present state of the ice that is consistent with available observations, a process sometimes referred to as model initialization. This estimation problem can be formulated either as a deterministic inverse problem (i.e., as nonlinear least squares optimization governed by PDEs) or as a Bayesian inverse problem (i.e., as a statistical problem which aims to characterize a distribution of states). The latter approach, while more expensive, provides uncertainty estimates in addition to determining a best parameter fit.

Ice sheet dynamics [46] is typically governed by nonlinear Stokes equations or simplifications thereof, such as the first-order equations (see e.g., [47]). Generally, the most uncertain component in ice sheet simulations is the basal boundary condition, i.e., how the ice sheet interacts with the rock, sand, water or a mix thereof at its base. Estimating an ice sheet's effective boundary condition from velocity observations on the top surface, the ice sheet's geometry and a model for its dynamics is thus an important problem that can mathematically formulated as an inverse problem [4, 48, 49, 50, 3].

We summarize the formulation of this inverse problem next. As common in the literature, we use *a snapshot* optimization approach, where all the data are assumed to be collected over a short period of time during which changes in the ice geometry are negligible. We denote the bounded domain covered by ice by $\Omega \subset \mathbb{R}^m$, $m \in \{2,3\}$, and the basal, lateral and top parts of the domain boundary $\partial\Omega$ by Γ_b, Γ_l, and Γ_t, as illustrated in Figure 2.3.

The governing equations are nonlinear incompressible Stokes equations whose solution is the ice flow velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^m$ and the pressure $p : \Omega \to \mathbb{R}$ given as follows:

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{u}} &= \rho \boldsymbol{g} \quad \text{in } \Omega, && (2.6)\\ \nabla \cdot \boldsymbol{u} &= 0 \qquad \text{in } \Omega, && (2.7)\\ \boldsymbol{\sigma}_{\boldsymbol{u}} \boldsymbol{n} &= \boldsymbol{0} \qquad \text{on } \Gamma_t, && (2.8)\\ \boldsymbol{u}\cdot\boldsymbol{n} = 0 \text{ and } \boldsymbol{T}\left(\boldsymbol{\sigma}_{\boldsymbol{u}}\boldsymbol{n} + \exp\left(\beta\right)\boldsymbol{u}\right) &= \boldsymbol{0} \qquad \text{on } \Gamma_b, && (2.9) \end{aligned}$$

along with additional lateral boundary conditions. Here, β is a basal sliding parameter field, $\rho\boldsymbol{g}$ the body force density, where ρ is the mass density of the ice and $\boldsymbol{g}$ the acceleration due to gravity. Equation 2.6 describes the conservation of momentum, 2.7 the conservation of mass, and 2.8 are stress-free boundary conditions for the top surface (the ice-air interface). In normal direction, Equation 2.9 states a non-penetration condition, i.e., the ice cannot flow into the rock/sand layer which supports it (here $\boldsymbol{n}$ denotes the outward unit normal to the boundary $\partial\Omega$ and $\boldsymbol{T}$ the tangential operator, $\boldsymbol{T}\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{n}(\boldsymbol{n}^\top \boldsymbol{v})$). In tangential direction, Equation 2.9 specifies a tangential sliding condition that relates the fraction of tangential sliding and tangential stress through the (logarithmic) basal sliding field $\beta = \beta(x)$, $x \in \Gamma_b$. We employ Glen's flow law [51], a constitutive law for ice that relates the stress tensor $\boldsymbol{\sigma}_{\boldsymbol{u}}$ and the strain rate tensor $\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{u}} = \frac{1}{2}\left(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^\top\right)$,

$$\boldsymbol{\sigma}_{\boldsymbol{u}} = 2\eta\left(\boldsymbol{u}\right)\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{u}} - \boldsymbol{I}p, \text{ with } \eta\left(\boldsymbol{u}\right) = \frac{1}{2}A^{-1/n}\dot{\boldsymbol{\varepsilon}}_{\text{II}}^{\frac{1-n}{2n}}, \tag{2.10}$$

where η is the effective viscosity, $\boldsymbol{I}$ is the unit matrix, $\dot{\boldsymbol{\varepsilon}}_{\text{II}} = \text{tr}\left(\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{u}}^2\right)$ is the second invariant of the strain rate tensor, A is a flow rate factor, and n is Glen's exponent. Ice is typically modeled using $n \approx 3$, which corresponds to a shear-thinning constitutive relation, here we use $n = 3$.

As discussed above, the parameter containing the largest uncertainty is the (logarithmic) basal sliding field $\beta = \beta(x)$. Thus, it is usually the parameter inferred from (typically, satellite) observation data $\boldsymbol{d}$, here in the form of surface velocity measurements. Using an appropriate point observation operator $\boldsymbol{\mathcal{B}}$ that extracts point data from the solution $\boldsymbol{u}$ of the governing equations 2.6-2.9, and assuming additive observation errors $\boldsymbol{\eta}$, the relationship between model and data is now of the typical form

$$\boldsymbol{d} = \boldsymbol{\mathcal{B}}\boldsymbol{u} + \boldsymbol{\eta}. \tag{2.11}$$

Assuming that the observation errors $\boldsymbol{\eta}$ and the prior for the parameter field β follow Gaussian distributions, we are in the framework of Bayesian inverse problems summarized in Section 2.2.1.

2.5 Example I: Two-dimensional ISMIP-HOM benchmark

We first study the prospects of compressing the Gauss-Newton data-misfit Hessian in a problem inspired by the ISMIP-HOM collection of ice sheet simulation benchmark problems [52]. This problem set was used to explore inverse ice sheet problems in [50, 3]. After a short description of the problem setup, we present results such as the MAP point estimate $\boldsymbol{\beta}^{\star}$ and approximate Gaussianized posterior samples using an HODLR compression of the posterior covariance. Then, we study the impact that various problem features have on the suitability of the Gauss-Newton data-misfit Hessian for compression to the HODLR and global low-rank formats.

2.5.1 Problem setup

This problem setup consists of a rectangular piece of ice on a slope, as sketched in Figure 2.3. This simple example allows us to study the influence of the domain aspect ratio, the number of observations and the level of mesh refinement on the properties of the Gauss-Newton data-misfit Hessian matrix. Unless otherwise specified, the domain has a width of $W = 10^4$ [m] and a height of $H = 10^2$ [m]. Periodic boundary conditions are employed along the lateral boundaries such that the setup models an infinite slab of ice on a slope. The governing equations and other boundary conditions are as discussed in Equations 2.6-2.9.

The Stokes equations are discretized using Taylor-Hood finite elements on a mesh over $\Omega = [0, W] \times [0, H]$ of 256×10 rectangles, each subdivided into two triangles. To compute a MAP estimate, we generate synthetic surface velocity data using the true logarithmic basal sliding field, $\beta_{\text{true}}(x) := \log\left(1\,200 + 1\,100 \sin\left(\frac{2\pi x}{W}\right)\right)$. Given this basal sliding field, we solve Equations 2.6-2.9, extract the tangential velocity component at 100 uniformly distributed points on the top boundary Γ_t, and add 1% relative Gaussian noise to each data point, resulting in the synthetic data $\boldsymbol{d}$.

It remains to define the prior distribution for the parameter field β. The average value of β_{true} is used as constant prior mean $\overline{\beta}(x) = 6.73315 \approx \frac{1}{W}\int_0^W \beta_{\text{true}}(s)\,\mathrm{d}s$. The prior covariance matrix $\boldsymbol{\Gamma}_{\text{prior}}$ is a discretization of the covariance PDE operator $\mathcal{C} := (\delta I - \gamma\Delta)^{-1}$, with $\gamma = 6 \times 10^2$ and $\delta = 2.4 \times 10^{-3}$, with Robin boundary conditions [53]. These values are chosen in order to provide a relatively large prior correlation length of 10^3 [m] [54]. Next, we summarize the computation of the MAP point and the compression of the Gauss-Newton data-misfit Hessian matrix at the MAP point.

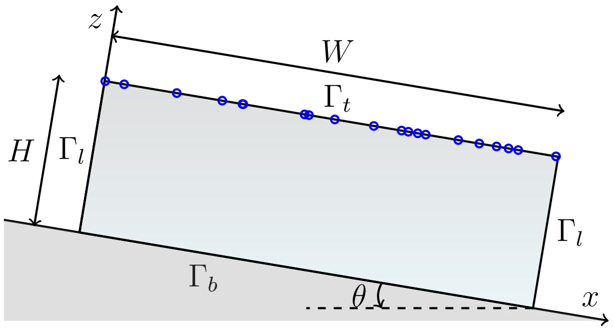


Figure 2.3: Schematic of two-dimensional slab of ice used for Example I in Section 2.5. The blue circles show representative (random) measurement locations. The angle θ is the slope of the ice slab.

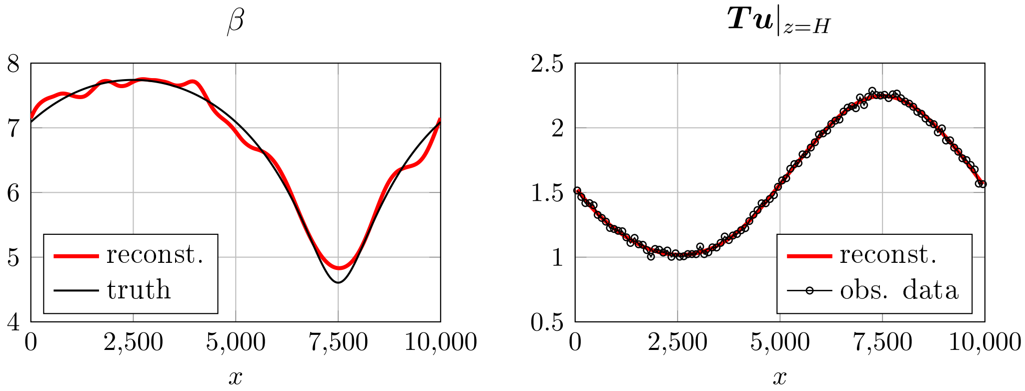


Figure 2.4: Shown for Example I are on the left the MAP point $\beta^{\star}$ (red) and the truth basal sliding parameter β_{true} (black) used to generate synthetic observations of the tangential velocity component on the upper surface Γ_t. Shown on the right are noisy synthetic observations (black dots) used for computing the MAP point and the associated tangential surface velocity reconstruction (red).

2.5.2 MAP point and HODLR Gaussianized posterior

The nonlinear optimization problem for finding the MAP estimate is solved using an inexact Gauss-Newton minimization method with backtracking linesearch [6], where the linear systems are iteratively solved by the conjungate gradient method. The resulting MAP point is shown in Figure 2.4. The MAP parameter field $\beta^{\star}$, closely resembles the true parameter β_{true}, which is a consequence of the large amount of available data and relatively small noise level.

Next, we use the Gaussianized posterior distribution with a compressed prior-

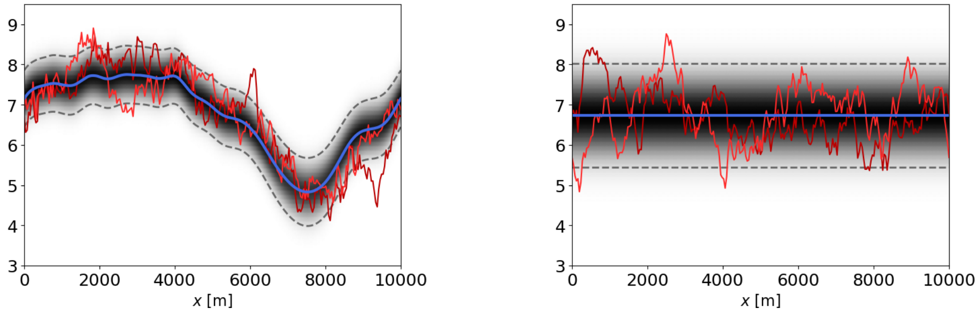


Figure 2.5: Results for Example I: Two random samples (red), mean $\overline{\beta}$ (blue) and boundaries of the region $R = \{(x, y)$ such that $0 \leq x \leq W$ and $\overline{\beta}(x) - 2\sigma(x) \leq y \leq \overline{\beta}(x) + 2\sigma(x)\}$ (dashed black) are shown for the prior (left) and a HODLR Gaussianized posterior using the scheme described in Section 2.3.2 (right).

preconditioned data-misfit Hessian $\boldsymbol{H}'_{\text{misfit}}$ to generate approximate samples from the posterior distribution. Upon construction of the HODLR compression of the prior-preconditioned data-misfit Hessian (details and comparisons can be found below in Section 2.5.3), we draw samples from the HODLR Gaussianized posterior as outlined in Section 2.3.2. In Figure 2.5, we compare the mean, pointwise standard deviation and samples from the prior and the posterior distributions. As expected, we find that the data updates our belief about the spatially distributed parameter field and reduces the uncertainty. In particular, the 2σ bounds on the one-dimensional point marginals $\sigma(x)$, $\boldsymbol{\sigma}_i = [\boldsymbol{\Gamma}_{i,i}]^{-1/2}$ of the Gaussianized posterior and the prior distributions are shown, in order to verify that the samples are largely contained within two standard deviations of their respective means. The prior-preconditioned data-misfit Hessian $\boldsymbol{H}'_{\text{misfit}}$, is compressed using a relative tolerance of 10^{-6}, that is $\|\boldsymbol{H}'_{\text{misfit}} - \tilde{\boldsymbol{H}}'_{\text{misfit}}\|_2 / \|\boldsymbol{H}'_{\text{misfit}}\|_2 \leq 10^{-6}$, with high probability.

2.5.3 Dependence of Hessian block spectra on problem setting

Next, we study how problem features impact the numerical suitability of using global low-rank and HODLR compressions to approximate the Gauss-Newton data-misfit Hessian. In this and subsequent sections we measure the cost to generate the matrix compression in terms of Hessian vector products, which we also describe as Hessian applies, as each said vector product requires two linearized PDE solves and thus dominates the computational cost. We use the result of Appendix A.2, to claim ε absolute error in a level L HODLR approximation, when there is no more than ε/L absolute error in each off-diagonal block. What is particular to this section, is that

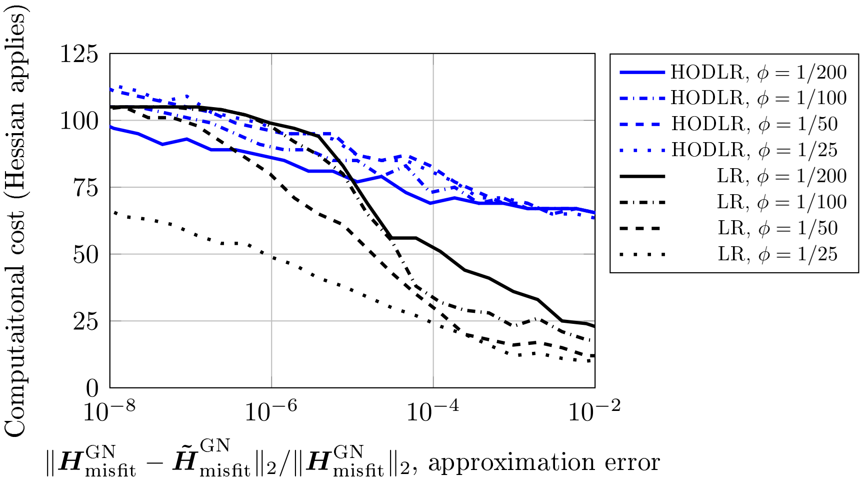


Figure 2.6: Comparison of HODLR and global low-rank (LR) compression costs of the Gauss-Newton data-misfit Hessian $\boldsymbol{H}^{\mathrm{GN}}_{\mathrm{misfit}}$, for Example I with ice sheet aspect ratio ϕ. This figure shows that for low aspect ratios, HODLR becomes more efficient than global low-rank for medium levels of target accuracy.

adaptive single-pass and HODLR algorithms are used to generate global low-rank and HODLR approximations, based on absolute tolerance criteria. The absolute tolerance algorithmic input is scaled by the largest global low-rank singular value in order to report relative approximation errors. We note that additional errors are neglected in the reported approximation error such as that incurred in the peeling process [27, 28] and additional approximation assumptions in the single-pass algorithm, both of which are not expected to be significant.

Influence of aspect ratio Here, we vary the aspect ratio of the domain $\phi = H/W$, where H and W are the domain height and width respectively, in order to study how it influences the block spectra of the Gauss-Newton data-misfit Hessian and ultimately the computational cost. Figure 2.6 shows that the global spectrum is more sensitive to changes in the relative length scale ϕ than the spectra of the off-diagonal blocks. Low-rank approximations of the off-diagonal blocks become computationally cheaper as ϕ decreases as a result of the sensitivity cones becoming increasingly localized as the ice sheet thickness decreases. Global low-rank approximations become more expensive as ϕ decreases, a result of the data being more informative. We note that realistic problems, such as the Humboldt glacier and the Greenland ice sheet studied later in Section 2.6, have small aspect ratios and are thus expected to have data-misfit Hessians that are less amenable to global low-rank approximation.

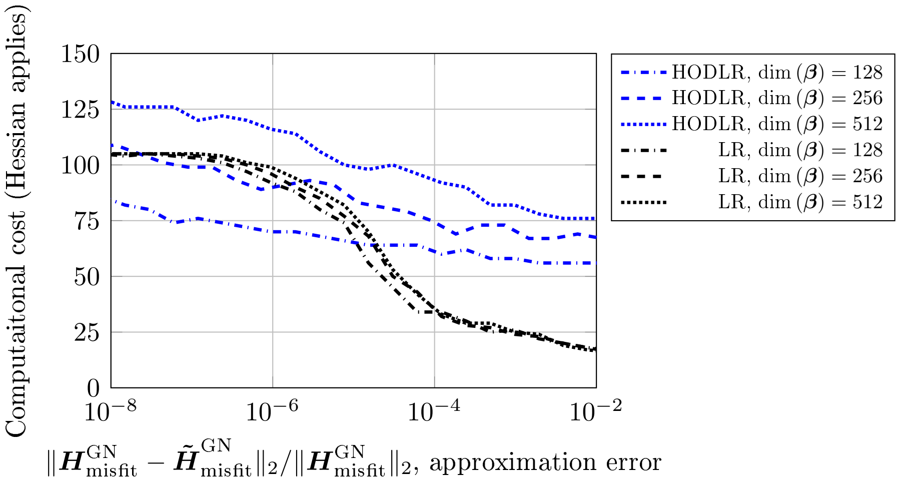


Figure 2.7: Dependence of HODLR and global low-rank (LR) compression costs of the Gauss-Newton data-misfit Hessian on dim $(\boldsymbol{\beta})$, the dimension of the discretized logarithmic basal sliding field for Example I. The cost of global low-rank compression is almost constant, while the cost of HODLR compression increases as the mesh is refined.

Influence of the parameter dimension We now vary the level of mesh discretization refinement in order to study the influence of data informativeness, through the discretized parameter dimension $N = \dim(\boldsymbol{\beta})$, on the computational cost to generate HODLR and global low-rank approximations of the Gauss-Newton data-misfit Hessian. The hierarchical depth L is incremented for every doubling of the discretized parameter dimension, in order that the hierarchical depth scales with the logarithm of the size of the Hessian matrix, a condition described in Section 2.2.2. Figure 2.7 provides computational evidence of the claim made in Section 2.2.2, that the number of applies needed to hierarchically compress an operator with HODLR structure is $\mathcal{O}(\log N)$. On the contrary, the number of applies to generate the global low-rank approximation is rather insensitive to the level of mesh refinement.

Influence of the data dimension Figure 2.8 shows that the global rank grows with the number of observations points and thus global low-rank compression tends to be less efficient for problems with strongly informative observation data. The rate of spectral decay of the (Gauss-Newton) data-misfit Hessian is related to the degree of ill-posedness of the unregularized inverse problem. As the number of observations increases, these associated model predictions are increasingly sensitive to small scale variations in the basal sliding field. Thus, more data, generally makes the data set more informative about the parameter and the (Gauss-Newton) data-misfit Hessian

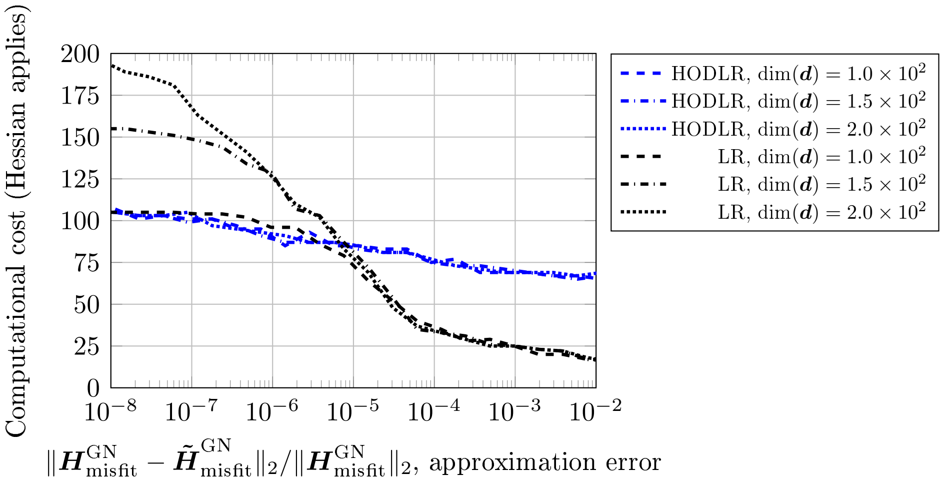


Figure 2.8: Dependence of HODLR and global low-rank (LR) compression costs of the Gauss-Newton data-misfit Hessian on $\dim(\boldsymbol{d})$, the data dimension, for Example I. The computational cost for global low-rank approximation increases with the number of observations, while the cost for HODLR compression is rather insensitive.

have a weaker rate of spectral decay.

2.6 Example II: Humboldt glacier and Greenland ice sheet

Here, we study the scalability of the proposed methods using large-scale ice sheet problems which are typically used in climate simulations. Namely, we focus on the Humboldt glacier in North-West Greenland, and the entire Greenland ice sheet. For these simulations, we use the ice sheet model MALI, [55], which relies on Albany, [56], a C++ multi-physics library for the implementation of the so-called first-order approximation of Stokes equations. This first-order approximation is based on scaling arguments motivated by the shallow nature of ice sheets and uses the incompressibility condition to reduce the unknows to the horizontal velocities. We use PyAlbany [57] a convenient Python interface to the Albany package, which in turn builds upon Trilinos [58]. Albany is designed to support parallel and scalable finite-element discretized PDE solvers and various analysis capabilities. Details about the parameter, state, data dimensions as well as the number of cores and hierarchical levels used in the computations is provided in Table 2.1.

The following study is partially motivated by findings made in the Section 2.5, namely that the role of the aspect ratio between the vertical and horizontal directions

	Humboldt	Greenland
$\dim(\boldsymbol{\beta})$	11 608	320 116
$\dim(\boldsymbol{u})$	255 376	7 042 552
$\dim(\boldsymbol{d})$	23 216	640 232
# of cores	120	2 048
L	8	10

Table 2.1: Problem specifications for the Humboldt glacier and Greenland ice-sheet problems (Example II). Dimension of the discretized parameter field $\dim(\boldsymbol{\beta})$, dimension of the discretized velocity field $\dim(\boldsymbol{u})$, dimension of the observations $\dim(\boldsymbol{d})$, processors employed for computations and L depth of HODLR hierarchical partitioning.

(see Section 2.5.3) influences the ability to use global low-rank compression and favors HODLR compression. We generate HODLR and global low-rank approximations and then based on the computed spectra, Equation 2.3 and $\zeta^{\mathrm{LR}} = r + d$, we estimate the computational cost. Additionally, we study how the ordering of the degrees of freedom impacts the spectral decay for off-diagonal blocks of the data-misfit Hessian. We present results for both, the Humboldt glacier, which expands about 4×10^2 [km] laterally, and the Greenland ice sheet, which expands about 1.8×10^3 [km]. The ice is at most 3.4 [km] thick, resulting in approximate aspect ratios of 8.5×10^{-3} for Humboldt and 1.9×10^{-3} for Greenland. We use a nonuniform triangulation of the Greenland ice sheet, with mesh size ranging from 1 to 10 [km], and we then extrude it in the vertical direction, obtaining a 3D mesh having 10 layers of prismatic elements. The velocity observations at the top surface of the Greenland ice sheet are obtained from satellite observations [59]. The MAP basal sliding field and the temperature fields are obtained as part of the initialization process, using a numerical optimization approach to match the ice velocity observations and constrained by the first-order flow model coupled with a temperature model [5]. Additional details about the mesh geometries and data, in particular regarding the Humboldt glacier, can be found in [60].

In Figure 2.9, we show the observed surface velocity $\boldsymbol{d}$ in [m/yr], the MAP estimates of the logarithmic basal sliding field $\beta^{\star}$ ($\exp(\beta^{\star})$ is in [kPa yr/m]) and surface velocity in [m/yr] generated by the model.

2.6.1 HODLR compressability

We next generate global low-rank approximations of a Greenland and Humboldt data-misfit Hessian as well as low-rank approximations of various off-diagonal blocks. Plots of the estimated singular values are provided in Figure 2.10. We observe that

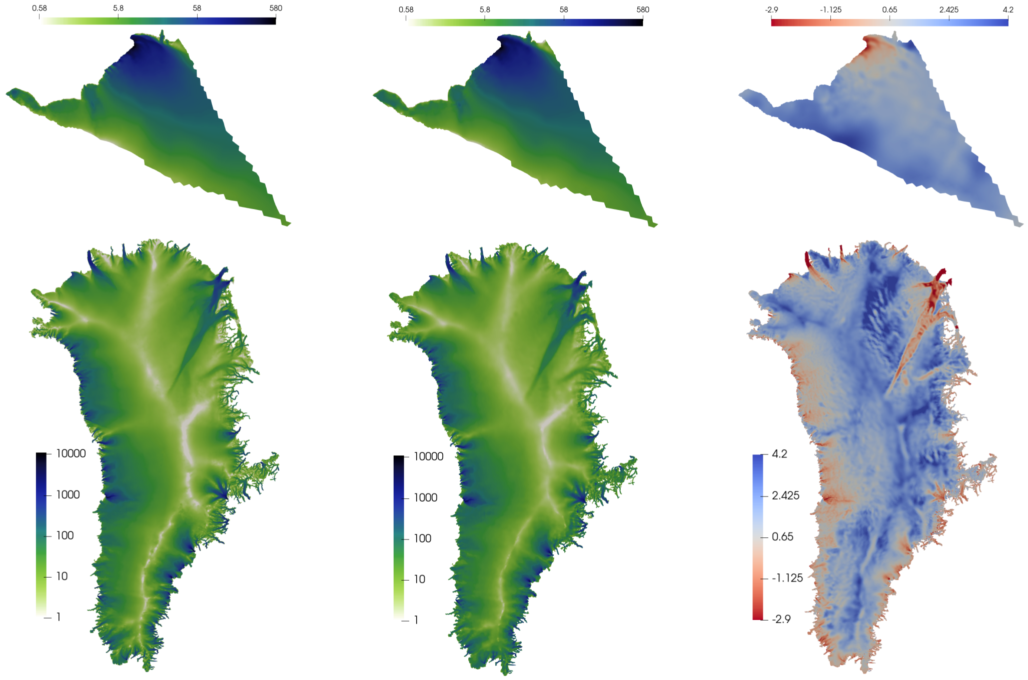


Figure 2.9: Data and MAP estimates for Example II. Shown are the surface velocity observation data (left), and the reconstructed surface velocity field (middle) that is based on the MAP estimate of the logarithmic basal sliding field (right). Top row is for the Humboldt glacier and bottom row for the Greenland ice sheet.

the spectrum of the Greenland ice sheet decays substantially slower than the one for the Humboldt glacier. Besides the different sizes of these two discretized problems, this is also due to the different aspect ratios. Having estimated singular values of the data-misfit Hessians and the appropriate off-diagonal blocks, one is able to estimate computational costs to compress them into the global low-rank and HODLR matrix formats. The computational cost as a function of Hessian approximation target accuracy is given in Figure 2.11, wherein it is demonstrated that the HODLR compression format can offer a favorable means to approximate data-misfit Hessians for large-scale inverse problems governed by complex ice-sheet models.

2.6.2 Impact of parameter degree of freedom ordering

We seek to ensure that the off-diagonal blocks, determined by the hierarchical partitioning described in Section 2.2.2, of the data-misfit Hessian are low-rank. For this reason, the nodes $\{\boldsymbol{x}_i\}_i$ associated to the degrees of freedom (dofs) are ordered according to a kd-tree, i.e., a recursive hyperplane splitting. The ordering provided by

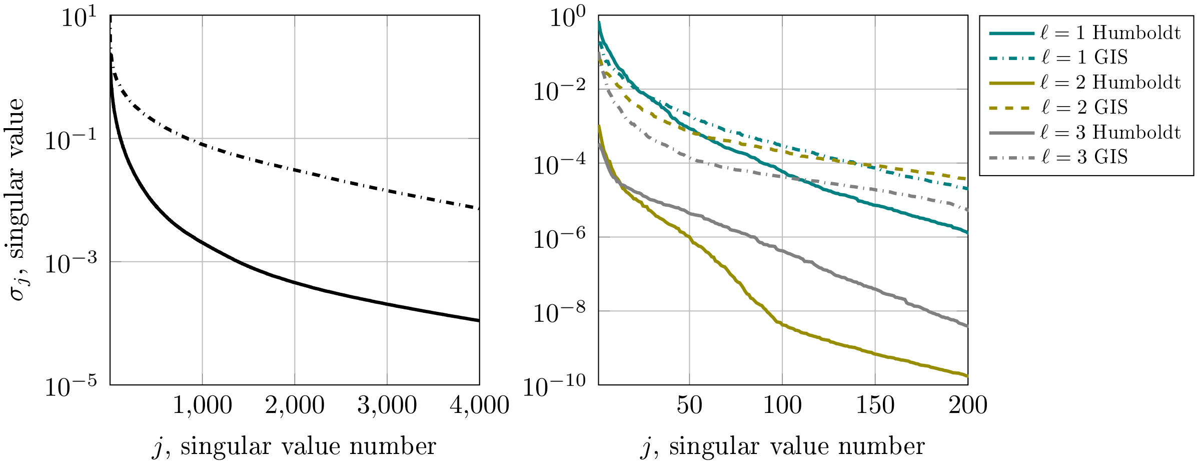


Figure 2.10: Singular values of the data-misfit Hessian (left figure) and various off-diagonal blocks of the data-misfit Hessian (right figure) for Example II. The color-scheme in the right most figure is consistent with Figure 2.1. On the left, the singular values of the Humboldt and Greenland data-misfit Hessians are shown using a solid and dash-dotted line, respectively. On the right, we show the singular values of the upper most blocks, that is $\boldsymbol{A}_{1,2}^{(\ell)}$ as defined in Section A.2.

the kd-tree is such that the (i, j)-entry of the distance matrix $\boldsymbol{D}_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$, is typically small whenever $|i - j|$ is small, that is the dof ordering preserves some notion of locality (see Section 2.3.1). In particular, a sparse permutation matrix $\boldsymbol{B}$, is determined, whose action reorders the dofs from the default ordering provided by the finite element discretization to that specified by the kd-tree. The data-misfit Hessian with respect to the kd-tree ordering, $\boldsymbol{H}_{\text{misfit}}^{\text{kd}} := \boldsymbol{B}\boldsymbol{H}_{\text{misfit}}\boldsymbol{B}^{\top}$, is then amenable to HODLR compression. Subsequently, $\boldsymbol{B}^{\top}\tilde{\boldsymbol{H}}_{\text{misfit}}^{\text{kd}}\boldsymbol{B}$ is an approximation of the data-misfit Hessian with respect to the default ordering.

The dof ordering has no impact on a matrix's global numerical rank but does indeed impact the numerical rank of its numerous submatrices that are defined by a fixed partitioning scheme, such as the off-diagonal blocks of an HODLR matrix (see Section 2.2.2). Here, we study the HODLR compressibility of the Humboldt glacier data-misfit Hessian by comparing the rate of decay of an off-diagonal block's singular values using the default ordering provided by Albany and the ordering obtained by a kd-tree recursive hyperplane splitting. As observed in Figure 2.12, the rate at which the singular values of the level-1 off-diagonal block decay, strongly depends on the dof ordering. This is because the ordering given by the kd-tree better preserves locality, and as a consequence, by the argument provided in Section 2.3.1, the singular values decay much faster when using the kd-tree ordering. The kd-tree ordering therefore

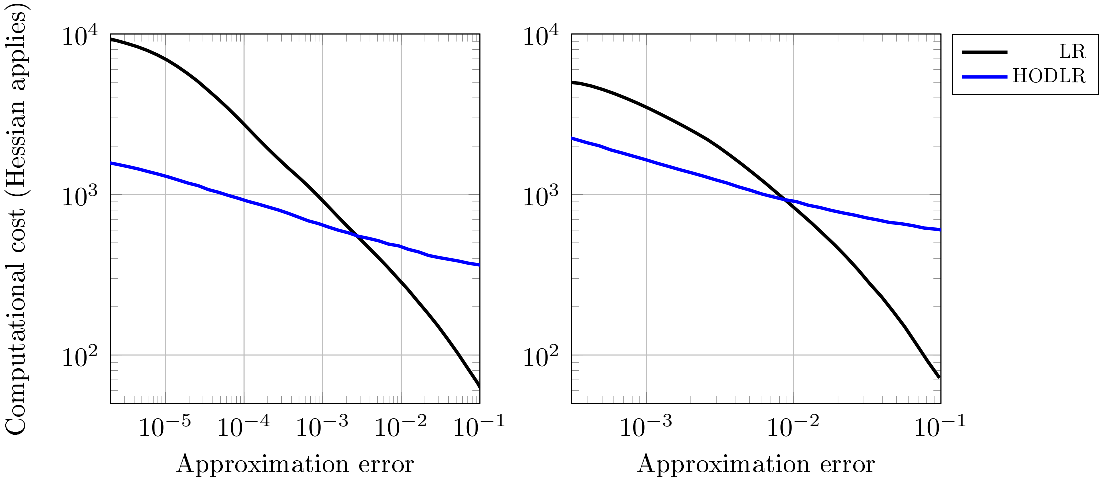


Figure 2.11: Estimated computational costs (measured by the number of Hessian applies) to compress the Humboldt glacier (left) and Greenland ice-sheet (right) data-misfit Hessians into the global low-rank (LR) and hierarchical off-diagonal low-rank (HODLR) formats as a function of the approximation error $\|\boldsymbol{H}_{\text{misfit}} - \tilde{\boldsymbol{H}}_{\text{misfit}}\|_2/\|\boldsymbol{H}_{\text{misfit}}\|_2$.

provides a substantially computationally cheaper means to generate an HODLR approximation of the data-misfit Hessian. Figure 2.12 also shows distance matrices for the default and kd-tree bases. These show the improved locality for the kd-orderings. Note that data-misfit Hessian matrices are expected to follow a similar structure as these distance matrices, which explains why the former's off-diagonal blocks can be compressed more effectively in the kd-order than in the default order of dofs.

2.7 Conclusion

In this work, we motivated why data-misfit Hessians which arise from a class of inverse problems governed by PDEs have HODLR matrix structure. HODLR matrices can efficiently be inverted and factorized, operations needed for solving inverse problems governed by PDEs by Newton's method, for constructing Gaussian approximations and for Markov chain Monte Carlo sampling methods. We study inverse ice sheet problems, for which, under certain regimes, HODLR matrices provide a more computationally efficient approximation format than the global low-rank matrix format. These problems are those with highly informative data and small aspect ratio ice sheets. While global low-rank matrices are favorable for large discretized parameter dimension and small data dimension, we find that HODLR matrices can offer compu-

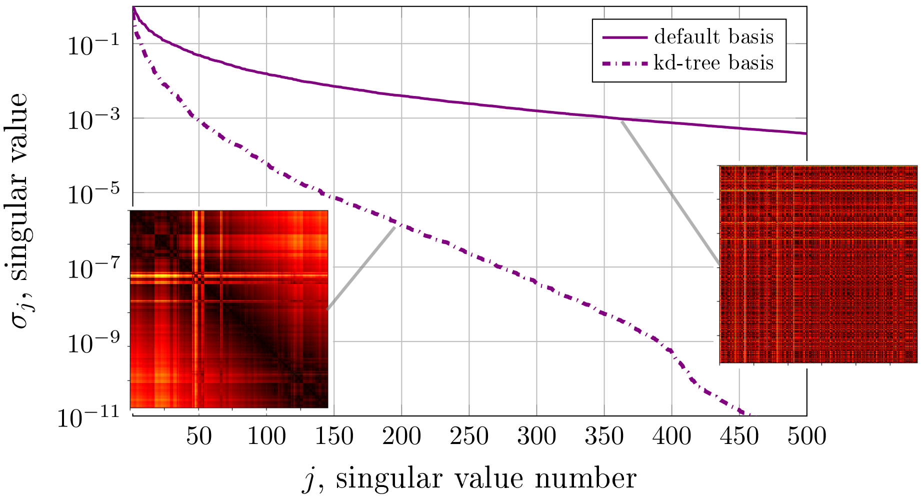


Figure 2.12: Singular values of the hierarchical level 1 off-diagonal block, $\boldsymbol{A}_{1,2}^{(1)}$, of the Humboldt glacier data-misfit Hessian, when expressed in a kd-tree basis and the default basis. Shown also are heat maps of the distance matrices $\boldsymbol{D}_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$, wherein the nodes $\{\boldsymbol{x}_i\}_i$, associated to the finite element degrees of freedom have been ordered according to a default standard and a kd-tree.

tational savings for large-scale inverse problems such as a Greenland ice sheet inverse problem with satellite observational data and a discretized parameter dimension that exceeds 10^5.

The accuracy of the data-misfit Hessian can be increased by utilizing more complex hierarchical matrix partitionings that satisfy a strong admissibility condition [33], as they are better suited to exploit data-misfit Hessian structure. However, the number of Hessian vector products required to generate a hierarchical matrix approximation by the peeling method [27, 28], grows with the complexity of the hierarchical partitioning. Ultimately, to further reduce the computational cost of Hessian approximations in inverse problems governed by PDEs, exploiting further problem structure is essential and that is precisely the content of the fourth chapter of this thesis.

Chapter 3

A Point Spread Function Method to Further Enable Hierarchical Hessian Approximation in Inverse Problems

3.1 Introduction

Our motivation, as before, is Hessian approximation for spatially distributed parameter inverse problems governed by PDEs. The Hessian is an operator that is central to efficient methods for solving such inverse problems in both the deterministic and Bayesian settings, but information about the Hessian is only accessible via application of the Hessian to vectors [21, 20]. The most popular matrix-free Hessian approximation methods are based on low-rank approximation [61, 62, 63, 64, 65] because there are matrix-free methods for constructing low-rank matrix approximations [66, 43] and that a number of inverse problems contain prior-preconditioned data-misfit Hessians with small and discretization independent numerical rank. However, in many inverse problems of practical interest this Hessian is high-rank [67, 68] and the computational cost to determine a low-rank matrix approximation grows with the numerical rank. Moreover, the numerical rank of the data-misfit Hessian grows as the data in the inverse problem become more informative about the unknown parameter. Low-rank approximation methods therefore suffer from a data predicament—if the data are highly informative about the unknown parameter, then the numerical rank of the data-misfit Hessian is large, so a large number of operator applies are required to form an accurate “low-rank” approximation [69, 21, 70]. Alternatively, the cost of hierarchical-matrix, e.g., HODLR (see Chapter 2), data-misfit Hessian approximation is largely insensitive to the informativeness of the data and so offers a data-scalable alternative to global low-rank approximation.

Hierarchical matrices that satisfy a strong-admissibility criteria, i.e. $\mathcal{H}$-matrices [71, 72], are a natural matrix format for exploiting the narrow sensitivity structure of elliptic PDEs that gives rise to off-diagonal low-rank Hessian matrices in inverse problems (see Section 2.3.1). In this context, a strong admissibility criteria refers to a geometric means of choosing which off-diagonal blocks are approximated by low-rank matrices. For inverse problems discretized by nodal finite elements, the rows and columns of an off-diagonal block corresponds to clusters of points. An off-diagonal block of an $\mathcal{H}$-matrix is approximated by a low-rank matrix when the cluster of points associated to it's row is sufficiently well separated from the cluster of points associated to it's columns. Hierarchical matrices with a geometric based strong-admissibility criteria are better suited to exploit the narrow-sensitivity structure of governing elliptic PDEs than hierarchical matrices which do not satisfy this criteria, such as HODLR matrices. Unfortunately, matrix partitioning patterns of $\mathcal{H}$-matrices are generally fairly complex and the number of matrix-vector products required to determine a hierarchical matrix approximation of a matrix-free operator by a randomized peeling process [73, 74, 75, 76], grows with the complexity of the hierarchical partitioning pattern. Classical methods [71, 72] to construct $\mathcal{H}$-matrices require evaluation of matrix-entries and thus means to approximately evaluate Hessian entries are needed to construct $\mathcal{H}$-matrix approximations of Hessians.

We present an efficient matrix-free point spread function (PSF) method for approximating operators, $H : L^2(\Omega) \to L^2(\Omega)'$, that have locally supported non-negative integral kernels. Here $\Omega \subset \mathbb{R}^d$ is a bounded domain, and $L^2(\Omega)'$ is the space of real-valued continuous linear functionals on $L^2(\Omega)$. Such operators arise as Hessian operators in inverse problems governed by PDEs [77, 78] where

$$(H\,\beta_1)(\beta_2) = \frac{\delta^2 J}{\delta\beta^2}(\beta_1, \beta_2),$$

for an objective functional $J(\beta)$ and $\delta^2 J/\delta\beta^2(\beta_1, \beta_2)$ is the second variation of the objective functional at and with respect to β in directions β_1 and β_2. Such operators also arise as Schur complements in Schur complement methods for solving PDEs [79, 80], Poincare-Steklov operators in domain decomposition methods (e.g., Dirichlet-to-Neumann maps), covariance operators in spatial statistics [81, 82, 83], and blurring operators in imaging [84, 85]. By matrix-free we mean that there is a black box computational procedure to apply H and its adjoint[1], $H^\dagger$, to arbitrary functions,

$$\beta_1 \mapsto H\beta_1 \qquad \text{and} \qquad \beta_2 \mapsto H^\dagger\beta_2, \tag{3.1}$$

but there is not a simple means to evaluate H's integral kernel. In applications, evaluating the maps in (3.1) may require a costly computational procedure, such as solving a large linear system.

[1] The adjoint operator $H^\dagger : L^2(\Omega) \to L^2(\Omega)'$ is the unique operator satisfying $(H\beta_1)(\beta_2) = \left(H^\dagger\beta_2\right)(\beta_1)$ for all $\beta_1, \beta_2 \in L^2(\Omega)$, where $H\beta_1 \in L^2(\Omega)'$ is the result of applying H to $\beta_1 \in L^2(\Omega)$, and $(H\beta_1)(\beta_2)$ is the result of applying that linear functional to $\beta_2 \in L^2(\Omega)$, and similar for operations with $H^\dagger$.

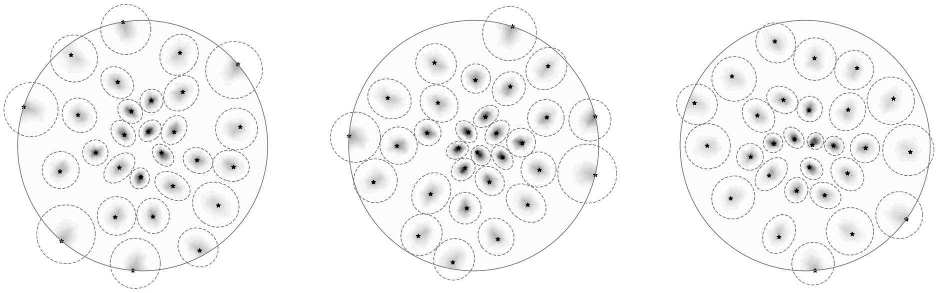


Figure 3.1: Batches, η_b, of normalized impulses, ϕ_x, for the Stokes inverse problem data-misfit Gauss-Newton Hessian (Section 3.4). Black stars are point source locations. Dashed gray ellipses are estimated impulse support ellipsoids. The large circle is $\partial\Omega$.

We use impulse interpolation to form a high-rank operator approximation using a small number of operator applies. The impulse, ϕ_x, associated with a point x is the Riesz representation[2] of the linear functional that results from applying H to a delta distribution (i.e., point source, impulse) centered at x. We compute batches of impulses by applying H to linear combinations of delta distributions associated with batches of points scattered throughout the domain (see Figure 3.1). Then we interpolate translated and scaled versions of these impulses to approximate entries of the operator's integral kernel. Picking the batches of points requires us to estimate the supports of the impulses ϕ_x *before* we compute them. The idea of estimating the supports of the functions ϕ_x a-priori was inspired by techniques from resolution analysis in seismic imaging. There, $H^\dagger$ is applied to a random noise function, and the width of ϕ_x is estimated to be the autocorrelation length of the resultant function near x [86, 87]. We use polynomial functions instead of random noise functions (see Section 3.3). Our method estimates the support of ϕ_x more accurately and reliably than random noise probing but is applicable only to those operators H which have non-negative integral kernels. Our impulse interpolation may be categorized as a PSF method that is loosely based on product convolution (PC) approximations, which are approximations of an operator by weighted sums of convolution operators with spatially varying weights. PC and PSF methods have a long history dating back several decades. We note the following papers (among many others), [88, 77, 89, 90, 91, 92, 93, 85, 26], in which the convolution kernels are constructed from sampling impulses of the operator to scattered point sources. For background on PC and PSF methods, we recommend the following papers: [94, 95, 96]. While PC approximations

[2] The Riesz representative of a functional $\rho \in L^2(\Omega)'$ with respect to the L^2 inner product is the unique function $\rho^* \in L^2(\Omega)$ such that $\rho(w) = (\rho^*, w)_{L^2(\Omega)}$ for all $w \in L^2(\Omega)$.

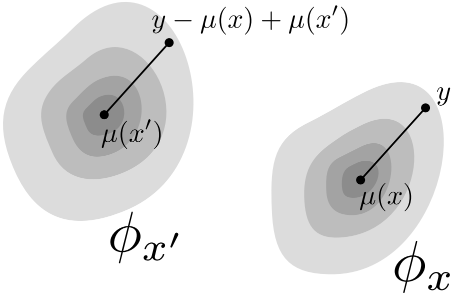


Figure 3.2: H is locally mean displacement invariant if $\phi_x(y) = \phi_{x'}\left(y - \mu(x) + \mu(x')\right)$ when $\|x - x'\|$ is small. Here $\mu(z)$ denotes the mean of ϕ_z.

are based on an assumption of local translation invariance, the method discussed in this Chapter is based on a local mean displacement invariance property, which is a generalization of local translation invariance.

The ability to rapidly approximate entries of H's integral kernel allows us to form a $\mathcal{H}$-matrix [71, 72] approximation of a discretized version of H. $\mathcal{H}$-matrices constitute a matrix format in which the rows and columns of the matrix are reordered, then the matrix is recursively subdivided into blocks, in such a way that many off-diagonal blocks are low-rank, even though the matrix as a whole may be high-rank. $\mathcal{H}$-matrix methods permit us to perform matrix-vector products cheaply, and perform other useful linear algebra operations that cannot be done easily using the original operator. These operations include matrix-matrix addition, matrix-matrix multiplication, matrix factorization, and matrix inversion. The work and memory required to perform these operations for an $N \times N$ $\mathcal{H}$-matrix scales *nearly linearly* in N (i.e., $o(N^{1+\epsilon})$ for any $\epsilon > 0$). The exact cost depends on the type of $\mathcal{H}$-matrix used, the operation being performed, and the rank of the off-diagonal blocks [97]. The ability to perform these operations permits, for example, fast solution of Newton linear systems in PDE-constrained optimization, fast sampling of ill-conditioned posterior distribution in Bayesian inverse problems, and construction of high-rank surrogate models that can be used for uncertainty quantification. There are matrix-free $\mathcal{H}$-matrix construction methods based on a peeling process [73, 74, 75, 76], which are not used in this Chapter. These alternative methods have been applied to form $\mathcal{H}$-matrix representations of Hessians in inverse problems governed by PDEs [67, 98]. Methods based on the peeling process are asymptotically scalable, (typically the number of matrix-vector products required scales as $\mathcal{O}(\log N)$), but in practice the required number of matrix-vector products is large.

3.2 Preliminaries

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain (typically $d = 1$, 2, or 3). We seek to approximate integral operators $H : L^2(\Omega) \to L^2(\Omega)'$ of the form

$$(H\beta_1)(\beta_2) := \int_\Omega \int_\Omega \beta_2(y)\Phi(y,x)\beta_1(x)\mathrm{d}x\mathrm{d}y. \tag{3.2}$$

Here the linear functional $H\beta_1 \in L^2(\Omega)'$ is the result of applying H to $\beta_1 \in L^2(\Omega)$, the scalar $(H\beta_1)(\beta_2)$ is the result of applying that linear functional to $\beta_2 \in L^2(\Omega)$. The integral kernel, $\Phi : \Omega \times \Omega \to \mathbb{R}$, exists in principle but is not easily accessible. In this section we describe how to extend the action of H to distributions, which allows us to define impulse responses (Section 3.2.1), we state the conditions on H that our method requires (Section 3.2.2), and detail finite element discretization (Section 3.2.3).

3.2.1 Distributions and impulses

The action of H may be extended to distributions if Φ is sufficiently regular. Let $\rho \in L^2(\Omega)'$, and let $\rho^* \in L^2(\Omega)$ denote the Riesz representative of ρ with respect to the $L^2(\Omega)$ inner product. We have

$$\begin{aligned}(H\rho^*)(w) &= \int_\Omega \int_\Omega w(y)\Phi(y,x)\rho^*(x)\mathrm{d}x\ \mathrm{d}y && \text{(3.3a)}\\ &= \int_\Omega w(y)\int_\Omega \Phi(y,x)\rho^*(x)\mathrm{d}x\ \mathrm{d}y = \int_\Omega w(y)\rho\left(\Phi(y,\cdot)\right)\mathrm{d}y, && \text{(3.3b)}\end{aligned}$$

where $\Phi(y,\cdot)$ denotes the function $x \mapsto \Phi(y,x)$. Now let $\mathcal{D}(\Omega) \subset L^2(\Omega)$ be a suitable space of test functions and let $\rho : \mathcal{D}(\Omega) \to \mathbb{R}$ be a distribution. In this case, the Riesz representative ρ^* may not exist, so the derivation in (3.3) is not valid. However, if Φ is sufficiently regular such that the function $y \mapsto \rho\left(\Phi(y,\ \cdot\ )\right)$ is well defined for almost all $y \in \Omega$, and if this function is in $L^2(\Omega)$, then the right hand side of (3.3b) is well defined. Hence, we *define* the action of H on the distribution ρ to be the right hand side of (3.3b). We denote this action by "$H\rho^*$," even if ρ^* does not exist.

Let δ_x denote the delta distribution(i.e., point source, impulse) centered at the point $x \in \Omega$. The *impulse* of H associated with x is the function $\phi_x : \Omega \to \mathbb{R}$,

$$\phi_x := \left(H\delta_x^*\right)^*, \tag{3.4}$$

that is formed by applying H to δ_x, then taking the Riesz representation of the resulting linear functional. Using (3.3b) and the definition of the delta distribution, we see that ϕ_x may also be written as the function $\phi_x(y) = \Phi(y,x)$.

3.2.2 Required conditions

We focus on approximating operators that satisfy the following conditions:

1. The kernel Φ is sufficiently regular so that ϕ_x is well defined for all $x \in \Omega$.
2. The supports of the impulses ϕ_x are contained in localized regions.
3. The integral kernel is non-negative in the sense that

   $$\Phi(y, x) \geq 0$$

   for all $(y, x) \in \Omega \times \Omega$.[3]

Our method may still perform well if these conditions are relaxed slightly. It is fine if the support of ϕ_x is not perfectly contained in a localized region, so long as the bulk of the "mass" of ϕ_x is contained in a localized region. The integral kernel does not need to be non-negative for all pairs of points $(y, x) \in \Omega \times \Omega$, so long as it is non-negative for the vast majority of pairs of points (y, x), and as long as the negative numbers are comparatively small. If these conditions are violated, our method will incur additional error. If these conditions are violated too much, our method may fail.

3.2.3 Finite element discretization

In computations, functions are discretized and replaced by finite dimensional vectors, and operators mapping between infinite dimensional spaces are replaced by operators mapping between finite dimensional spaces. We discretize using continuous Galerkin finite elements satisfying the Kronecker property (defined below). With minor modifications, our method could be used with more general finite element methods, or other discretization schemes such as finite differences or finite volumes.

Let $\psi_1, \psi_2, \dots, \psi_N$ be a set of continuous Galerkin finite element basis functions used to discretize the problem on a mesh with mesh size parameter h, let $\mathcal{V}_h := \operatorname{Span}(\psi_1, \psi_2, \dots, \psi_N)$ be the corresponding finite element space under the L^2 inner product, and let $\zeta_i \in \mathbb{R}^d$, $i = 1, \dots, N$ be the Lagrange nodes associated with the functions ψ_i. We assume that the finite element basis satisfies the Kronecker property, i.e., $\psi_i(\zeta_i) = 1$ and $\psi_i(\zeta_j) = 0$ when $i \neq j$. For $\beta_h \in \mathcal{V}_h$ we write $\boldsymbol{\beta} \in \mathbb{R}^m_{\boldsymbol{M}}$ to denote the coefficient vector for β_h with respect to the finite element basis, i.e., $\beta_h(x) = \sum_{i=1}^N \boldsymbol{\beta}_i \psi_i(x)$. Linear functionals $\rho_h \in \mathcal{V}_h'$ have coefficient dual vectors $\boldsymbol{\rho} \in \mathbb{R}^m_{\boldsymbol{M}^{-1}}$, with entries $\boldsymbol{\rho}_i = \rho_h(\psi_i)$ for $i = 1, \dots, m$. Here $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ denotes the sparse finite element mass matrix which has entries $\boldsymbol{M}_{i,j} = \int_\Omega \psi_i(x)\psi_j(x)dx$ for $i, j = 1, \dots, N$.

[3] Note that having a non-negative integral kernel is different from positive semi-definiteness. The operator H need not be positive semi-definite to use our method, and positive semi-definite operators need not have a non-negative integral kernel.

The space $\mathbb{R}^N_{\boldsymbol{M}}$ is $\mathbb{R}^N$ with the inner product $(\boldsymbol{u}, \boldsymbol{w})_{\boldsymbol{M}} := \boldsymbol{u}^\top \boldsymbol{M} \boldsymbol{w}$, and $\mathbb{R}^N_{\boldsymbol{M}^{-1}}$ is the analogous space with $\boldsymbol{M}^{-1}$ replacing $\boldsymbol{M}$. Direct calculation shows that $\mathbb{R}^N_{\boldsymbol{M}}$ and $\mathbb{R}^N_{\boldsymbol{M}^{-1}}$ are isomorphic to $\mathcal{V}_h$ and $\mathcal{V}_h'$ as Hilbert spaces, respectively.

After discretization, the operator $H : L^2(\Omega) \to L^2(\Omega)'$ is replaced by an operator $H_h : \mathcal{V}_h \to \mathcal{V}_h'$, which becomes an operator

$$\boldsymbol{H} : \mathbb{R}^N_{\boldsymbol{M}} \to \mathbb{R}^N_{\boldsymbol{M}^{-1}}$$

under the isomorphism discussed above. Our method is agnostic to the computational procedure for approximating H with $\boldsymbol{H}$. What is important is that we do not have direct access to matrix entries $\boldsymbol{H}_{i,j}$. Rather, we have a computational procedure that allows us to compute matrix-vector products $\boldsymbol{\beta} \mapsto \boldsymbol{H}\boldsymbol{\beta}$ and $\boldsymbol{w} \mapsto \boldsymbol{H}^\top \boldsymbol{w}$, where computing these matrix-vector products is assumed to be costly. Of course, matrix entries can be computed via matrix-vector products as $\boldsymbol{H}_{i,j} = \left(\boldsymbol{H}\boldsymbol{e}^{(j)}\right)_i$, where $\boldsymbol{e}^{(j)} = (0, \dots, 0, 1, 0, \dots, 0)^\top$ is the length N unit vector with one in the jth coordinate and zeros elsewhere. But computing the matrix-vector product $\boldsymbol{e}^{(j)} \mapsto \boldsymbol{H}\boldsymbol{e}^{(j)}$ is costly, and therefore wasteful if we do not use other matrix entries in the jth column of $\boldsymbol{H}$. Hence, methods for approximating $\boldsymbol{H}$ are computationally intractable if they require accessing scattered matrix entries from many different rows and columns of $\boldsymbol{H}$.

The operator $H_h : \mathcal{V}_h \to \mathcal{V}_h'$ can be written in integral kernel form, (3.2), but with Φ replaced by a slightly different integral kernel, Φ_h, which we do not know, and which differs from Φ due to discretization error. Since the functions in $\mathcal{V}_h$ are continuous at x, the delta distribution δ_x is a continuous linear functional on $\mathcal{V}_h$, which has a discrete dual vector $\boldsymbol{\delta}_x \in \mathbb{R}^N_{\boldsymbol{M}^{-1}}$ with entries $(\boldsymbol{\delta}_x)_i = \psi_i(x)$ for $i = 1, \dots, N$. Additionally, it is straightforward to verify that the Riesz representation, $\rho_h^* \in \mathcal{V}_h$, of a functional $\rho \in \mathcal{V}_h'$ has coefficient vector $\boldsymbol{\rho}^* = \boldsymbol{M}^{-1}\boldsymbol{\rho}$. Therefore, the formula for the impulse from (3.4) becomes $\boldsymbol{\phi}_x = (A_h \delta_x^*)^* = \boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{\delta}_x$, and the (y, x) kernel entry of Φ_h may be written as $\Phi_h(y, x) = \boldsymbol{\delta}_y^\top \boldsymbol{\phi}_x = \boldsymbol{\delta}_y^\top \boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{\delta}_x$. Now define $\boldsymbol{\Phi} \in \mathbb{R}^{N \times N}$ to be the following dense matrix of kernel entries evaluated at all pairs of Lagrange nodes:

$$\boldsymbol{\Phi}_{i,j} := \Phi_h(\zeta_i, \zeta_j). \tag{3.5}$$

Because of the Kronecker property of the finite element basis, we have $\boldsymbol{\delta}_{\zeta_i} = \boldsymbol{e}_i$. Thus, we have $\Phi_h(\zeta_i, \zeta_j) = \left(\boldsymbol{M}^{-1}\boldsymbol{H}\boldsymbol{M}^{-1}\right)_{i,j}$, which implies

$$\boldsymbol{H} = \boldsymbol{M}\boldsymbol{\Phi}\boldsymbol{M}. \tag{3.6}$$

Broadly, our method will construct an $\mathcal{H}$-matrix approximation of $\boldsymbol{H}$ by forming $\mathcal{H}$-matrix approximations of $\boldsymbol{\Phi}$ and $\boldsymbol{M}$ (or a lumped mass version of $\boldsymbol{M}$), then multiplying these matrices per (3.6) using $\mathcal{H}$-matrix methods. Classical $\mathcal{H}$-matrix construction methods require access to arbitrary matrix entries $\boldsymbol{\Phi}_{i,j}$, but these matrix entries are not easily accessible. The bulk of our method is therefore dedicated to forming approximations of these matrix entries that can be evaluated rapidly.

Lumped mass matrix At the continuum level, Φ is assumed to be non-negative. However, entries of Φ_h, involve inverse mass matrices, which typically contain negative numbers. If there are too many negative numbers, or if the negative numbers are too large, our algorithm will be less robust. We therefore recommend replacing the mass matrix, $\boldsymbol{M}$, with a positive diagonal *lumped mass* approximation. In our numerical results, we use the lumped mass matrix constructed by replacing off-diagonal entries of the mass matrix by zero. Other mass lumping techniques may be used [99].

3.3 Summary of algorithmic methodology

We next summarize the algorithmic approach, whose goal is to have an means to quickly approximate entries of H's integral kernel by approximating the impulse response functions $\phi_x(y)$. Due to the locality of the impulse responses (see Figure 3.1), it is feasible that they can be computed in batches,

$$\sum_i \phi_{x_i}(y) = \left(H \sum_i \delta^*_{x_i}\right)^* (y).$$

In order to determine which points x_i are to be included in the Dirac comb $\sum_i \delta_{x_i}$, it is important that the supports of the associated impulse responses $\phi_{x_i}(y)$ do not overlap much. We make use of the identity

$$\left(H^\dagger w\right)^* (x) = \int_\Omega \Phi(y,x) w(y) dy = \left(\phi_x, w\right)_{L^2(\Omega)}, \tag{3.7}$$

for which we see that by computing one operator application of $H^\dagger$ to w, we simultaneously compute the L^2 inner product of each and every impulse ϕ_x with w. By taking w to be C, L^i, and $Q^{i,j}$, the following constant, linear, and quadratic functions:

$$C(x) := 1, \qquad L^i(x) := x^i, \qquad Q^{i,j}(x) := x^i x^j$$

for $i,j = 1,\dots,d$, we are able to determine the low-order moments of each and every impulse response ϕ_x, that is

$$V(x) = \left(H^\dagger C\right)^* (x) = \left(\phi_x, C\right)_{L^2(\Omega)}, \tag{3.8a}$$

$$\mu^i(x) = \left(H^\dagger L^i\right)^* (x)/V(x) = \left(\phi_x, L^i\right)_{L^2(\Omega)}, \tag{3.8b}$$

$$\Sigma^{i,j}(x) = \left(H^\dagger Q^{i,j}\right)^* (x)/V(x) - \mu^i(x) \cdot \mu^j(x) = \left(\phi_x, Q^{i,j}\right)_{L^2(\Omega)}, \tag{3.8c}$$

for $i,j = 1,\dots,d$. Having estimated these low-order moments we then approximate the support of ϕ_x within the ellipsoid

$$E_x := \{x' \in \Omega : (x' - \mu(x))^\top \Sigma(x)^{-1} (x' - \mu(x)) \leq \tau^2\}, \tag{3.9}$$

where τ is a fixed constant, for the numerical results in Section 3.4, $\tau = 3$. With knowledge of the approximate support of each and every impulse response we then choose, via an ellipsoid packing problem, a batches sample points, S_k, such any two distinct impulse responses ϕ_{x_i}, ϕ_{x_j}, associated to a batch of sample points, $x_i, x_j \in S_k$, have supports that do not overlap. This is an efficient means to generate many impulse responses with a small number of applications of $H^\dagger$, it is however not scalable to compute each and every impulse response in batches, which we denote by η_b.

We next describe a means to approximate unknown impulse responses ϕ_x from those impulse responses $\phi_{x'}$ which have already been computed. One approximation scheme assumes that the underlying PDEs are locally translation invariant, that is that the impulse responses are identical up to a shift given by the displacement in the center's of the generating Dirac distributions i.e., $\phi_x(y) \approx \phi_{x'}(y - x + x')$. Here, to account for PDE models with nonuniform advection in addition to multiple scales we utilize an approximation scheme that assumes that the impulse responses satisfy a scaled local mean displacement invariance property

$$\phi_x(y)/V(x) \approx \phi_{x'}\left(y - \mu(x) + \mu(x')\right)/V(x'). \tag{3.10}$$

Once the impulse batches η_b are computed, we approximate the integral kernel $\Phi(y, x)$ at arbitrary points (y, x) by interpolation of translated and scaled versions of the computed impulses. The key idea behind the interpolation is the scaled local mean displacement invariance assumption. Specifically, we approximate $\Phi(y, x) = \phi_x(y)$ by a weighted linear combination of the values $\frac{V(x)}{V(x_i)}\phi_{x_i}(y - \mu(x) + \mu(x_i))$ for a small number of sample points x_i near x. The weights are determined by interpolation.

The ability to rapidly evaluate approximate kernel entries $\Phi(y, x)$ allows us to construct an $\mathcal{H}$-matrix approximation, $\mathbf{\Phi}_{\mathcal{H}} \approx \mathbf{\Phi}$, using the conventional adaptive cross $\mathcal{H}$-matrix construction method. In this method, one forms low-rank approximations of off-diagonal blocks of the matrix by sampling rows and columns of those blocks. We then use $\mathcal{H}$-matrix methods to convert $\mathbf{\Phi}_H$ into an $\mathcal{H}$-matrix approximation $\boldsymbol{H}_{\mathcal{H}} \approx \boldsymbol{H}$. The complete algorithm for constructing $\boldsymbol{H}_{\mathcal{H}}$ is shown in Algorithm 2.

When H is symmetric positive semi-definite, $\boldsymbol{H}_{\mathcal{H}}$ may be non-symmetric and indefinite due to errors in the approximation. In this case, one may (optionally) modify the $\mathcal{H}$-matrix representation of $\boldsymbol{H}_{\mathcal{H}}$ to make it symmetric positive semi-definite using a rational function method that is described in [25].

Algorithm 2 Localpsf $\mathcal{H}$-matrix approximation of the matrix-free Hessian.
Input: Linear operator H, parameter n_b
Output: $\mathcal{H}$-matrix $\boldsymbol{H}_{\mathcal{H}}$ approximation of $\boldsymbol{H}$

```
Compute $V, \mu$, and $\Sigma$
for $k = 1, 2, \ldots, n_b$ do
    Choose a batch of sample points, $S_k$
    Compute a batch of impulse response $\eta_k$ by applying $H$ to a Dirac comb for $S_k$
end for
Form $\mathcal{H}$-matrix approximation $\boldsymbol{\Phi}_{\mathcal{H}}$ of integral kernel
Form $\mathcal{H}$-matrix approximation $\boldsymbol{H}_{\mathcal{H}}$ of $H$
(optional) Modify $\boldsymbol{H}_{\mathcal{H}}$ to make it symmetric positive definite
```

3.4 Numerical results

3.4.1 Problem setup

We use our method to approximate the data-misfit Gauss-Newton Hessian in an inverse problem governed by a Stokes PDE (2.6)-(2.9), described in Chapter 2, which models steady state ice sheet flow [100]. The domain geometry is chosen so as to model the flow of ice down an irregular geophysical entity such as a mountain (see Figure 3.3). As in Chapter 2 of this thesis, given observations of the tangential component of the ice velocity on the top surface of the ice, we seek to invert for the logarithm of the unknown spatially varying basal friction parameter which quantifies the local resistance to ice flow across the ground which supports the ice sheet. The domain that is filled with ice is denoted by $U \subset \mathbb{R}^3$. The basal, lateral and top parts of ∂U are denoted by Γ_b, Γ_l, and Γ_t, respectively. The governing equations are the linear incompressible Stokes equations,

$$-\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{u}} = \boldsymbol{f} \quad \text{in } U, \tag{3.11a}$$
$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } U, \tag{3.11b}$$
$$\boldsymbol{\sigma}_{\boldsymbol{u}} \boldsymbol{n} = \boldsymbol{0} \quad \text{on } \Gamma_t, \tag{3.11c}$$
$$\boldsymbol{u} \cdot \boldsymbol{n} = 0 \text{ and } \boldsymbol{T}\left(\boldsymbol{\sigma}_{\boldsymbol{u}} \boldsymbol{n} + \exp\left(\beta\right) \boldsymbol{u}\right) = \boldsymbol{0} \quad \text{on } \Gamma_b, \tag{3.11d}$$
$$\boldsymbol{\sigma}_{\boldsymbol{u}} \boldsymbol{n} + s \boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_l. \tag{3.11e}$$

The solution to these equations is the pair $(\boldsymbol{u}, p)$, where $\boldsymbol{u}$ is the ice flow velocity field, and p is the pressure field. Here, β is the unknown logarithmic basal sliding parameter field which is defined on the 2D surface Γ_b. The quantity $\boldsymbol{f}$ is the body force density due to gravity, $s = 10^6$ is a Robin boundary condition constant, $\boldsymbol{n}$ is the outward normal and $\boldsymbol{T}$ is the tangential projection operator that restricts a vector field to its tangential component along the boundary. We employ Glen's flow law [51],

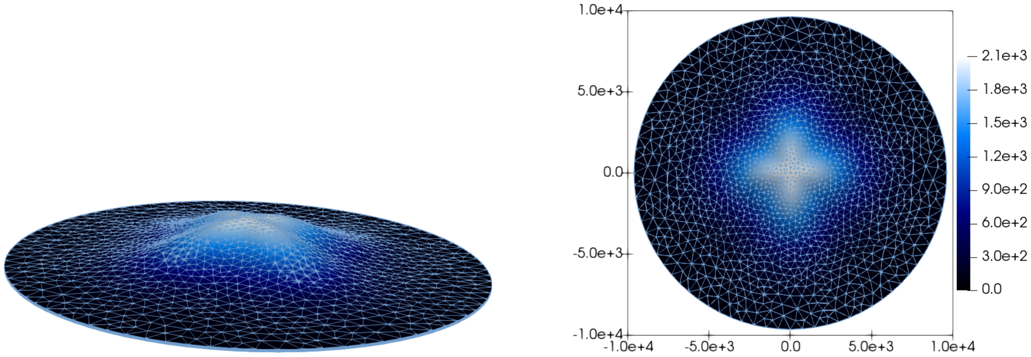


Figure 3.3: The ice sheet, discretized by a mesh of tetrahedra, bird's eye view (left) and top down view (right). Color indicates the height of the base of the ice sheet (i.e., the mountain topography). The radius of the domain is 10^4 meters, the maximum height of the mountain is 2.1×10^3 meters, and the average thickness of the ice sheet is 250 meters.

$\boldsymbol{\sigma}_{\boldsymbol{u}} = 2\eta\dot{\boldsymbol{\varepsilon}} - \boldsymbol{I}p$, which is a constitutive law for ice that relates the stress tensor, $\boldsymbol{\sigma}_{\boldsymbol{u}}$, to the strain rate tensor, $\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\top\right)$. Here η is a constant effective viscosity and $\boldsymbol{I}$ is the unit matrix. Glen's exponent has been chosen to be one, which provides for a linear Stokes model. Note that while the PDE is linear, the parameter-to-solution map, $\beta \mapsto (\boldsymbol{u}, p)$, is nonlinear.

The pressure, p, is discretized with first order scalar continuous Galerkin finite elements defined on a mesh of tetrahedra. The velocity, $\boldsymbol{u}$, is discretized with second order continuous Galerkin finite elements on the same mesh. The parameter β is discretized with first order scalar continuous Galerkin finite elements on the mesh of triangles that results from restricting the tetrahedral mesh to the basal boundary, Γ_b. Note that Γ_b is a 2D surface embedded in 3D due to the mountain topography. We also generate a flattened version of Γ_b, denoted by $\Omega \subset \mathbb{R}^2$, by ignoring the height coordinate. The parameter β is dually viewed as a function on Γ_b for the purpose of solving the Stokes equations, and as a function on Ω for the purpose of building our Hessian approximations and defining the regularization. The true parameter field, β_{true}, and corresponding velocity field, $\boldsymbol{u}_{\text{true}}$, are shown in Figure 3.4. We generate synthetic observations by restricting the tangential component of $\boldsymbol{u}_{\text{true}}$ to the top surface to get

$$y_{\text{true}} = \boldsymbol{T}\boldsymbol{u}_{\text{true}}|_{\Gamma_t}.$$

We add 1% multiplicative Gaussian noise to all components of the finite element vector, $\boldsymbol{y}_{\text{true}}$, corresponding to the function y_{true}, which yields noisy synthetic obser-

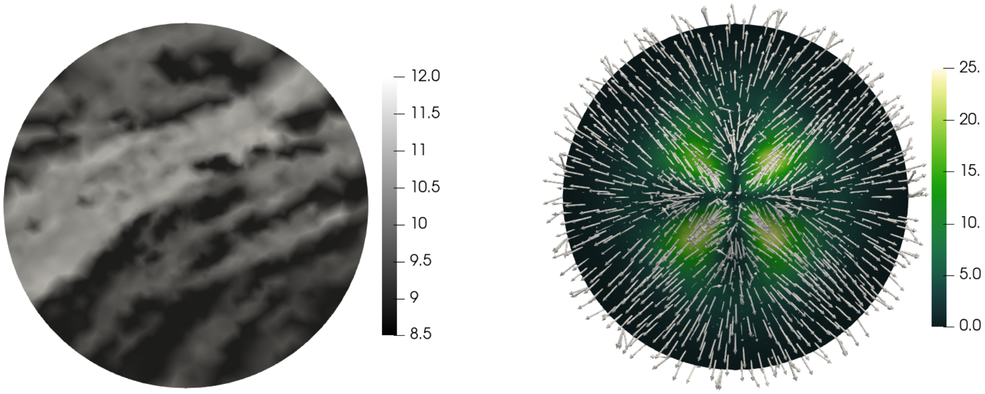


Figure 3.4: True parameter, β_{true}, (left) and true velocity $\boldsymbol{u}_{\text{true}}$ (right). In the right plot, arrows indicate the direction of $\boldsymbol{u}_{\text{true}}$ and color indicates the magnitude of $\boldsymbol{u}_{\text{true}}$.

vations, y_{obs} which are used for the inversion. That is,

$$(\boldsymbol{y}_{\text{obs}})_j := (\boldsymbol{y}_{\text{true}})_j + 0.01 \cdot \boldsymbol{\xi}_j \cdot |\,(\boldsymbol{y}_{\text{true}})_j\,|,$$

where j ranges over all entries of the vector of coefficients representing the finite element functions y_{true} and y_{obs}, and $\boldsymbol{\xi}_j$ are independent and identically distributed (i.i.d) random numbers drawn from the standard normal distribution. We also define the noise function, ξ, to be the finite element function with jth vector entry $\boldsymbol{\xi}_j$.

The reconstructed parameter field β is found as the solution to an optimization problem of the following form:

$$\min_{\beta} \quad J(\beta) := J_{\text{misfit}}(\beta) + R(\beta). \tag{3.12}$$

The first term in the objective function is

$$J_{\text{misfit}}(\beta) := \frac{1}{2}\int_{\Gamma_t} \|y_{\text{obs}} - \boldsymbol{T}\boldsymbol{u}\|_2^2 \, \mathrm{d}S.$$

We call this term the data-misfit, because it measures the difference between the observed data and the predicted data based on a candidate parameter β. Here $\boldsymbol{u} = \boldsymbol{u}(\beta)$ denotes the velocity field solving Equation (3.11) for the given parameter β. The second term,

$$R(\beta) := \frac{1}{2}\int_{\Omega} |\mathcal{K}(\beta - \beta_0)|^2 \, \mathrm{d}x,$$

is a Bilaplacian regularization term. Specifically, β_0 is the constant function $\beta_0(x) = 10.5$, and $\mathcal{K}$ is the inverse of the solution operator for the following elliptic PDE:

$$\begin{aligned} -\gamma\Delta u + \delta\, u &= f \quad \text{in } \Omega, \\ \gamma\nabla u \cdot \boldsymbol{n} + s\, u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Recall that Ω is the 2D flattened version of the basal surface Γ_b. Here f is a generic forcing term, and $s = \sqrt{\delta\gamma}/1.42$ is a Robin boundary condition coefficient [101], and these quantities are different from $\boldsymbol{f}$ and s in the Stokes equation (3.11). In all results except for the regularization robustness study in Figure 3.6, γ is chosen so that the Morozov discrepancy principle is satisfied, i.e.,

$$\int_{\Gamma_t} \|y_{\text{obs}} - \boldsymbol{T}\boldsymbol{u}\|_2^2 \, dS = \int_{\Gamma_t} \|\xi\|_2^2 \, \mathrm{d}S.$$

The value of γ satisfying the Morozov discrepancy principle is $\gamma = 7.3 \times 10^3$. The exception is Figure 3.6, where we vary γ to study how changing γ impacts the effectiveness of our preconditioner. The constant δ is chosen so that the correlation length of functions drawn from the normal distribution with covariance $\mathcal{K}^{-2}$, given by $L = \sqrt{\gamma/\delta}$, is 1/10th the radius of the domain.

We solve the inverse problem with an inexact Newton preconditioned conjugate gradient (PCG) scheme [102, Section 7.1], using β_0 as the initial guess. The discretized parameter, $\boldsymbol{\beta}$ is updated iteratively according to the formula $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha_k \widehat{\boldsymbol{\beta}}_k$, where $\widehat{\boldsymbol{\beta}}_k$ approximately solves one of the following Newton linear systems,

$$\boldsymbol{H}\widehat{\boldsymbol{\beta}}_k = -\boldsymbol{g}_k \qquad \text{or} \qquad \boldsymbol{H}^{\text{GN}}\widehat{\boldsymbol{\beta}}_k = -\boldsymbol{g}_k, \tag{3.13}$$

and the step length $\alpha_k \in \mathbb{R}$ is chosen by line-search to satisfy the Armijo conditions [102, Section 3.1]. Here $\boldsymbol{g}_k$ is the discretized gradient of J, $\boldsymbol{H}$ is the discretized Hessian, and $\boldsymbol{H}^{\text{GN}}$ is the discretized Gauss-Newton Hessian, and these derivatives are evaluated at $\boldsymbol{\beta}_k$. We use $\boldsymbol{H}^{\text{GN}}$ for the first five iterations, and $\boldsymbol{H}$ for all subsequent iterations. The Newton iteration terminates when $\|\boldsymbol{g}_k\|/\|\boldsymbol{g}_0\| < 10^{-8}$. Systems (3.13) are solved inexactly using an inner PCG iteration, which terminates when the norm of the residual for the linear system is less than $\min\left(0.5, \sqrt{\|\boldsymbol{g}_k\|}\right)\|\boldsymbol{g}_k\|$. PCG only requires applying $\boldsymbol{H}$ (or $\boldsymbol{H}^{\text{GN}}$) to vectors. Each application of $\boldsymbol{H}$ (or $\boldsymbol{H}^{\text{GN}}$) to a vector requires solving one incremental forward and one incremental adjoint Stokes PDE of the form (3.11), but with different right hand sides.

3.4.2 Numerical results

We use the framework described in this Chapter to generate Hessian preconditioners. We build $\mathcal{H}$-matrix approximations of the data-misfit Gauss-Newton Hessian (the term in $\boldsymbol{H}^{\text{GN}}$ that arises from the data-misfit) using 1, 5, and 25 impulse batches. These approximations are denoted by PSF (1), PSF (5), and PSF (25), respectively. The resulting $\mathcal{H}$-matrices, which may be indefinite due to approximation error, are modified to be symmetric positive definite. We approximate $\boldsymbol{H}^{\text{GN}}$ rather than $\boldsymbol{H}$ because $\boldsymbol{H}$ more often has negative values in its integral kernel, especially when one is away from the optimal point. Our results show that our preconditioners are

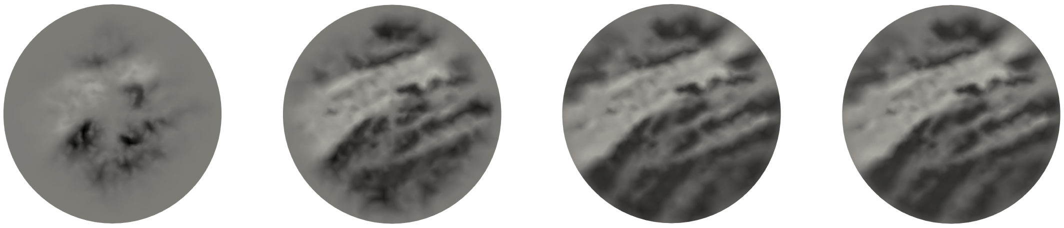


Figure 3.5: The log basal friction parameter after 1 (left), 3 (second from left), and 4 (third from left) Newton iterations, and the final Newton iterate (right). The PSF (5) preconditioner is constructed between Newton iterations 3 and 4.

also good preconditioners for $\boldsymbol{H}$. The $\mathcal{H}$-matrix approximation of the data-misfit Gauss-Newton Hessian is added to a $\mathcal{H}$-matrix approximation of the Hessian of the regularization term to form a $\mathcal{H}$-matrix approximation of $\boldsymbol{H}^{\text{GN}}$. Straightforward calculation shows that the discretized Hessian of the regularization term is given by $(\gamma\boldsymbol{K}+\delta\boldsymbol{M})\boldsymbol{M}^{-1}(\gamma\boldsymbol{K}+\delta\boldsymbol{M})$ where $\boldsymbol{K}$ is a finite element stiffness matrix with appropriate boundary conditions, and $\boldsymbol{M}$ is the finite element mass matrix. These matrices are sparse, so we form $\mathcal{H}$-matrix representations of these matrices and combine them using standard sparse $\mathcal{H}$-matrix techniques. We factor the overall Gauss-Newton Hessian approximation, denoted $\tilde{\boldsymbol{H}}$, using fast $\mathcal{H}$-matrix methods, then and use the factorization as a preconditioner.

Table 3.1 shows the performance of our preconditioner for accelerating the solution of optimization problem (3.12). We build the PSF (5) preconditioner in the third Gauss-Newton iteration, and reuse it for all subsequent Gauss-Newton and Newton iterations. No preconditioning is used in the Gauss-Newton iterations before the PSF (5) preconditioner is built. We compare our method with the most commonly used existing preconditioners: no preconditioning (NONE), and preconditioning by the regularization term in the Hessian (REG). Using PSF (5) reduces the total number of Stokes PDE solves by a factor of six compared to no preconditioning, and by a factor of nine compared to regularization preconditioning[4]. In Figure 3.5 we show select Newton iterates using our PSF (5) preconditioner.

Next, we build PSF (1), PSF (5), and PSF (25) preconditioners based on the Gauss-Newton Hessian evaluated at the converged solution $\boldsymbol{\beta}$. We use PCG to solve a linear system with the Hessian as the coefficient operator and a right hand vector with random i.i.d. entries drawn from the standard normal distribution. In the left most figure of Figure 3.6 we compare the convergence of PCG for solving this linear system using the PSF (1), PSF (5), PSF (25), REG, and NONE preconditioners. PCG converges fastest with the PSF preconditioners, with PSF (25) converging

[4] Interestingly, we observe that no preconditioning outperforms regularization preconditioning here because the noise level (and hence γ) is small.

	PSF (5)			REG			NONE		
Iter	#CG	#Stokes	$\|\boldsymbol{g}\|$	#CG	#Stokes	$\|\boldsymbol{g}\|$	#CG	#Stokes	$\|\boldsymbol{g}\|$
0	1	4	2.0e+7	1	4	2.0e+7	1	4	2.0e+7
1	2	6	5.8e+6	2	6	7.9e+6	2	6	5.8e+6
2	4	10	2.4e+6	5	12	3.9e+6	4	10	2.4e+6
3	2	6+22	5.8e+5	13	28	1.6e+6	13	28	5.8e+5
4	5	12	5.6e+4	42	86	4.8e+5	33	68	9.1e+4
5	10	22	3.5e+3	84	170	7.7e+4	57	116	6.2e+3
6	14	30	2.7e+1	125	252	5.0e+3	76	154	1.1e+2
7	0	2	4.1e-2	194	390	7.9e+1	117	236	2.3e-1
8	—	—	—	0	2	1.6e-1	0	2	4.1e-2
Total	38	114	—	466	950	—	303	624	—

Table 3.1: Convergence history for solving the Stokes inverse problem using preconditioned inexact Newton PCG. Preconditioners shown are our method with five impulse batches (PSF (5)) constructed at the third iteration, regularization preconditioning (REG), and no preconditioning (NONE). Columns titled #CG show the number of PCG iterations used to solve the Newton system for $\hat{\boldsymbol{\beta}}_k$. Columns titled $\|\boldsymbol{g}\|$ show the ℓ_2 norm of the gradient at $\boldsymbol{\beta}_k$. Columns titled #Stokes show the total number of Stokes PDE solves performed in each Newton iteration. This consists of Stokes solves for u performed during the linesearch going from $\boldsymbol{\beta}_{k-1}$ to $\boldsymbol{\beta}_k$, plus one adjoint Stokes solve to compute the gradient at $\boldsymbol{\beta}_k$, plus one incremental forward and one incremental adjoint Stokes solve per PCG iteration for solving the Newton system. In the PSF (5) portion of row 3, we write $6 + 22$ to indicate that 6 Stokes solves were used during the standard course of the iteration, and 22 Stokes solves were used to build the PSF (5) preconditioner.

fastest, followed by PSF (5), followed by PSF (1), as expected. PCG converges much slower with no preconditioning and regularization preconditioning than it does with PSF preconditioning, with no preconditioning outperforming regularization preconditioning. In the right most figure of 3.6, we perform PCG solves on linear systems of the same form, except now we vary γ. The performance of our PSF preconditioners is good and relatively stable over a wide range of γ's. All PSF preconditioners perform the same for medium and large values of γ. For small values of γ PSF (25) performs slightly better than PSF (5), which performs slightly better than PSF (1). As expected, regularization preconditioning performs well for large γ and poorly for small γ. Our PSF preconditioners outperform no preconditioning and regularization preconditioning for medium and small γ, and perform similarly to regularization preconditioning for large γ.

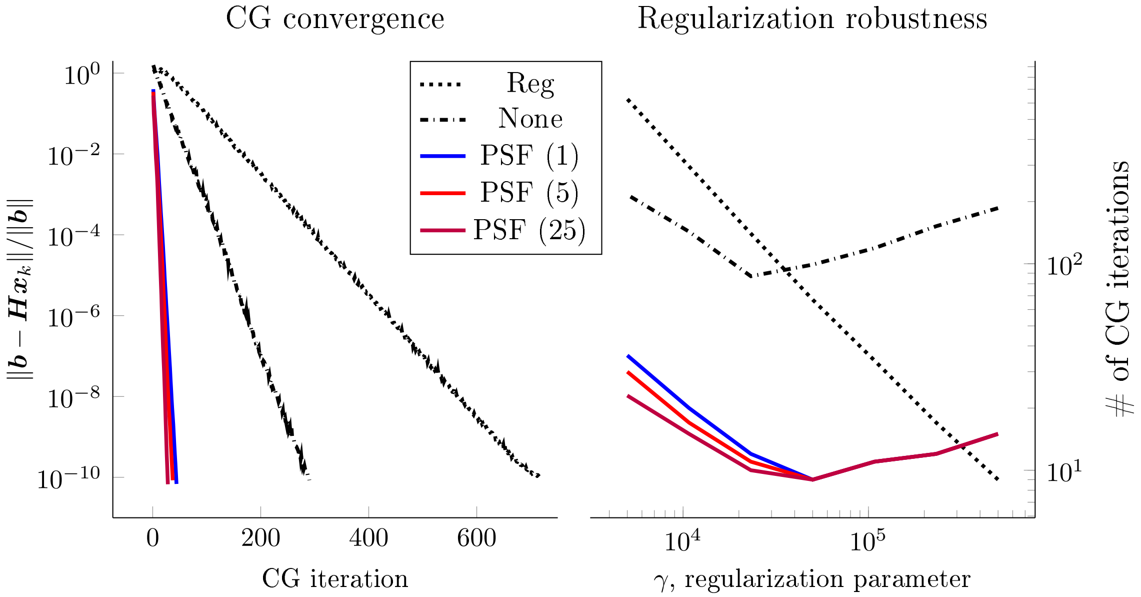
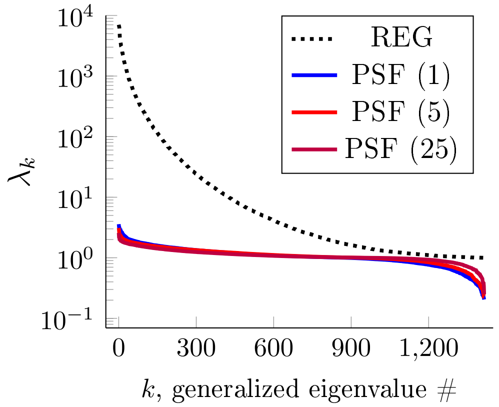


Figure 3.6: Left shows the convergence history for solving $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{b}$ using PCG, where $\boldsymbol{b}$ has i.i.d. random entries drawn from the standard normal distribution. Here we use $\gamma = 7.3 \times 10^3$, which is determined by the Morozov discrepancy principle. Right shows the number of PCG iterations required to achieve $\|\boldsymbol{b} - \boldsymbol{H}\boldsymbol{x}_k\|/\|\boldsymbol{b}\| < 10^{-6}$, where $\boldsymbol{x}_k$ is the kth PCG iterate, for a range of different γ. Results in these figures are shown for our preconditioner with 1, 5, and 25 batches of impulses (PSF (1), PSF (5), and PSF(25), respectively), regularization preconditioning (REG), and no preconditioning (NONE). The preconditioner is constructed using $\boldsymbol{H}^{\mathrm{GN}}$. For a fair comparison, even as γ changes, $\boldsymbol{H}$ and $\boldsymbol{H}^{\mathrm{GN}}$ are always evaluated at the same $\boldsymbol{\beta}$, which is the optimal point for the inverse problem with $\gamma = 7.3 \times 10^3$.

The left most figure of Figure 3.7 shows the generalized eigenvalues for the generalized eigenvalue problem $\boldsymbol{H}\boldsymbol{u} = \lambda\tilde{\boldsymbol{H}}\boldsymbol{u}$. Here $\boldsymbol{H}$ is the Hessian evaluated at the reconstructed parameter $\boldsymbol{\beta}$ for a regularization parameter γ chosen to satisfy the Morozov discrepancy principle. The matrix $\tilde{\boldsymbol{H}}$ is one of the PSF (1), PSF (5), or PSF (25) Gauss-Newton Hessian approximations constructed at that $\boldsymbol{\beta}$, or the regularization Hessian. The PSF preconditioners cluster the eigenvalues of the Hessian near one, with more batches tending to yield better clustering. The regularization preconditioner clusters the trailing eigenvalues but amplifies leading eigenvalues. The right most figure of Figure 3.7 we show the condition numbers of $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}^{\mathrm{GN}}$, for various preconditioners $\tilde{\boldsymbol{H}}$. The PSF preconditioned Hessians have the the smallest condition numbers, wherein more batches tend to improve the conditioning. Regularization preconditioning and no preconditioning yield preconditioned Hessians with



	$\kappa(\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H})$	$\kappa(\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}_{\text{gn}})$
REG	7.02e+3	7.02e+3
NONE	8.71e+2	9.22e+2
PSF (1)	1.75e+1	1.76e+1
PSF (5)	1.39e+1	1.39e+1
PSF (25)	1.03e+1	1.06e+1

Figure 3.7: Left: eigenvalues for generalized eigenvalue problem $\boldsymbol{H}\boldsymbol{u}_k = \lambda_k \tilde{\boldsymbol{H}}\boldsymbol{u}_k$. Here $\boldsymbol{H}$ is the Hessian and $\tilde{\boldsymbol{H}}$ is the PSF approximation constructed using the Gauss-Newton Hessian, $\boldsymbol{H}^{\text{GN}}$, for 1, 5, and 25 batches (PSF (1), PSF (5), and PSF (25), respectively), or the regularization Hessian (REG). Right: condition number, κ, for $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}^{\text{GN}}$ for these different preconditioners, and no preconditioner (NONE). All operators are evaluated at the $\boldsymbol{\beta}$ that solves the inverse problem for $\gamma = 7.3 \times 10^3$, as determined by the Morozov discrepancy principle.

larger condition numbers. The condition numbers are similar for both $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}^{-1}\boldsymbol{H}^{\text{GN}}$, demonstrating that the a PSF preconditioner built based on the Gauss-Newton Hessian $\boldsymbol{H}^{\text{GN}}$ can effectively precondition the Hessian $\boldsymbol{H}$.

3.5 Conclusion

We presented an efficient matrix-free PSF method for approximating operators with locally supported non-negative integral kernels. The method only requires access to the operator via application of the operator to a small number of vectors. The idea of the method is to compute batches of impulses by applying the operator to Dirac combs of scattered point sources, then interpolate these impulses to approximate entries of the operator's integral kernel. The interpolation is based on a local mean displacement invariance principle, which generalizes and improves upon local translation invariance. The ability to quickly approximate arbitrary integral kernel entries permits us to form a $\mathcal{H}$-matrix approximation of the operator. Fast $\mathcal{H}$-matrix arithmetic is then used to perform further linear algebra operations that cannot be performed easily with the original operator, such as matrix factorization and inversion. The supports of the impulses are estimated to be contained in ellipsoids, which are determined a-priori via a new procedure that involves applying the operator to a small number of polynomial functions. Point source locations for the impulse batches are chosen using a greedy ellipsoid packing procedure, in which we choose as many impulses per batch as possible, while ensuring that the corresponding ellipsoids do not overlap. We applied the method to approximate the Gauss-Newton Hessian in a Stokes ice sheet inverse problem, and saw that the approximation substantially outperforms existing Hessian approximation methods. Although the method we presented is not applicable to all Hessians, it is applicable to many Hessians of practical interest. For these Hessians, our method offers a *data scalable* alternative to conventional low-rank approximation, because our method can form high-rank approximations of an operator using a small number of operator applies.

Chapter 4

A Multigrid Interior-Point Approach for PDE- and Bound-Constrained Optimization

4.1 Introduction

In this Chapter we target PDE-constrained optimization problems with an additional bound constraint on the parameter $m(\boldsymbol{x})$. Formally, the determination or inversion of a spatially distributed state $u(\boldsymbol{x})$ and parameter $m(\boldsymbol{x})$ can be done by solving a PDE- and bound-constrained optimization problem in the form of

$$\min_{(u,m)\in\mathcal{V}\times\mathcal{M}} f(u,m), \tag{4.1}$$

$$\text{such that } c(u,m,\lambda)=0, \ \ \forall\lambda\in\mathcal{V}^0, \tag{4.2}$$

$$\text{and } m(\boldsymbol{x})\geq m_\ell(\boldsymbol{x}), \ \ \forall\boldsymbol{x}\in\Omega. \tag{4.3}$$

Here, $f(u,m)$ is the objective functional whose minimizer is saught, c defines the weak form of a partial differential equation, which in the context of the optimization problem is a partial differential equality constraint, m_ℓ is a pointwise parameter lower-bound, and $\mathcal{V}^0$ is a homogeneous version of the space $\mathcal{V}$, that is $(u+v)\in\mathcal{V}$ for any $u\in\mathcal{V}$ and $v\in\mathcal{V}^0$. Here it is assumed, as before that the objective functional $f(u,m)$ is the sum of misfit and regularization components $f(u,m)=f_{\text{misfit}}(u)+R(m)$.

In the reduced-space approach to solve optimization problems with PDE constraints [9, 2], as in Chapters 2 and 3 of this dissertation, one formally eliminates the state-variable u, by making use of the implicitly defined mapping from the parameter m to the PDE-solution $u(m)$. With this approach, evaluating various quantities such as the reduced-space functional $f(m):=f(u(m),m)$, or its various derivatives by the adjoint method [9, 103], can be computationally expensive, as they require the solution of (nonlinear) PDEs and optimizer estimation may require a significant number of such evaluations. Here we use a full-space approach in which functional evaluation and gradient computations are less computationally expensive as they do not require PDE-solution computation. An additional challenge of the full-space approach is that the Jacobian of the perturbed first-order optimality conditions, whose inverse action is needed for Newton-based methods, is an indefinite saddle-point matrix, for which effective preconditioners are challenging to construct and for which direct linear system solution methods, which poorly scale, are often applied. A central contribution of this work is a preconditioner (see Section 4.3.1) for such problems, which relies on the ill-posedness of the unregularized inverse problem and efficient algebraic multigrid schemes to solve sparse elliptic linear systems. The methodology presented in this Chapter is a hierarchical approach as the title of this thesis suggests, as hierarchies of algebraically defined grids are leveraged to exploit relations among multiple problem resolutions for efficient elliptic linear system solution by algebraic multigrid [104, 105].

4.1.1 Notation

Bold faced symbols generally indicate discretized quantitities, *e.g.*, vector and matrices. The scalar $\boldsymbol{x}_i$ denotes the ith component of the vector $\boldsymbol{x}$ and $\boldsymbol{A}_{i,j}$ the entry of $\boldsymbol{A}$ that resides in its the i-th row and j-th column. The symbol $\mathbf{1}$ refers to a vector with all entries equal to one, $\mathbf{0}$ refers to a vectors with all entries equal to zero, and $\boldsymbol{I}_n$ refers to the $n \times n$ identity matrix. The elementwise Hadamard product is indicated by $\odot$, that is $(\boldsymbol{x} \odot \boldsymbol{y})_i = \boldsymbol{x}_i \boldsymbol{y}_i$. The standard Euclidean vector norm $\sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, is denoted by $\|\boldsymbol{x}\|_2$, whereas weighted inner products, e.g., the $\boldsymbol{M}$-weighted inner product $\sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$ is denoted by $\|\boldsymbol{x}\|_{\boldsymbol{M}}$, where $\boldsymbol{M}$ is assumed to be symmetric positive definite. The closure of a set $\mathcal{S}$ is denoted by $\overline{\mathcal{S}}$, and $\emptyset$ is the empty set.

4.2 Interior-point methods for nonlinear PDE-constrained optimization

Our optimization approach falls under the umbrella of interior-point methods. Specifically, we employ a filter line-search interior-point method (IPM) [106] that has emerged recently as one of the most robust methods for nonlinear nonconvex optimization. This method also possesses best-in-class global and local convergence properties [107, 108].

An interior-point method involves solving a sequence of log-barrier subproblems, whose optimality conditions can be viewed as smoothed versions of the (nonsmooth) optimality conditions of (4.1)-(4.3). Mathematically, a so-called log-barrier parameter

μ is introduced and a sequence of problems

$$\min_{(u,m)\in\mathcal{V}\times\mathcal{M}} \varphi(u,m) := f(u,m) - \mu\int_\Omega \log(m - m_\ell)\mathrm{d}\boldsymbol{x}, \tag{4.4}$$

$$\text{such that } c(u,m,\lambda) = 0, \ \ \forall\lambda\in\mathcal{V}^0, \tag{4.5}$$

for $\mu \to 0^+$, is approximately solved. The inequality constraints $m \geq m_\ell$ are enforced implicitly in the form of $m > m_\ell$ via a line-search along a Newton direction, as detailed later in this section.

4.2.1 Discretization and derivatives calculations

In order to obtain a computational solution of Equations (4.4)-(4.5), said equations must be discretized. In this work, the discretization of the state equation (4.5) as well as all continuous fields are obtained by the finite element method with first-order continuous elements, but other finite elements or discretization methods, such as finite differences [109, 110] can be used as well. A careful analysis is needed to discretize the log-barrier term (see Equation (4.4)) to ensure consistency with the underlying finite element Hilbert space for m as shown in [111]. A careful analysis is also needed to make appropriate choices for the finite element representations for the dual variables (*i.e.*, $\boldsymbol{\lambda}$, $\boldsymbol{z}_\ell$ in Equations (4.12)-(4.15) below), discretized inner products and norms, and discretized stopping criteria. If the underlying finite element spaces are not properly accounted for then the IPM is prone to mesh dependence performance [111], such as requiring a mesh dependent number of optimization steps. This is especially pervasive when the meshes are nonuniform, for example, as the result of adaptive mesh refinement. We refer the reader to [111] for details and a comparative discussion of the differences between the discretized IPM of this work and the finite element-oblivious IPM from [106].

Upon discretization by finite elements, Equations (4.4)-(4.5) become

$$\min_{(\boldsymbol{u},\boldsymbol{m})\in\mathbb{R}^{n_1}\times\mathbb{R}^{n_2}} \varphi_h(\boldsymbol{u},\boldsymbol{m}) := f_h(\boldsymbol{u},\boldsymbol{m}) - \mu\mathbf{1}^\top\boldsymbol{M}\log(\boldsymbol{m}-\boldsymbol{m}_\ell), \tag{4.6}$$

$$\text{such that } \boldsymbol{c}(\boldsymbol{u},\boldsymbol{m}) = \mathbf{0}. \tag{4.7}$$

Here the finite element approximation $(u_h(\boldsymbol{x}),\, m_h(\boldsymbol{x})) \in \mathcal{V}_h\times\mathcal{M}_h$ of $(u(\boldsymbol{x}),\, m(\boldsymbol{x})) \in \mathcal{V}\times\mathcal{M}$ is given in terms of the basis elements $\{\phi_i(\boldsymbol{x})\}_{i=1}^{n_1}\subset\mathcal{V}_h$, $\{\psi_i(\boldsymbol{x})\}_{i=1}^{n_2}\subset\mathcal{M}_h$ as

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{n_1}\boldsymbol{u}_i\phi_i(\boldsymbol{x}),\ m_h(\boldsymbol{x}) = \sum_{i=1}^{n_2}\boldsymbol{m}_i\psi_i(\boldsymbol{x}), \tag{4.8}$$

furthermore $\boldsymbol{M}_{i,j} = \int_\Omega \psi_i(\boldsymbol{x})\,\psi_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is the mass matrix with respect to the space $\mathcal{M}_h$ used to discretize the parameter $m(\boldsymbol{x})$ and the discretized partial differential equality constraint $\boldsymbol{c}$ is given by

$$\left[\boldsymbol{c}(\boldsymbol{u},\boldsymbol{m})\right]_i = c(u_h, m_h, \phi_i), \quad 1\leq i\leq n_1.$$

It is to be noted that $\log(\boldsymbol{m}-\boldsymbol{m}_\ell)$ is a vector representation of a nodal discretization of $\log(m-m_\ell)$, rather than the logarithm of the difference of nodal discretizations of m and m_ℓ. We choose the former approach due to its simplicity and remark that the resultant discretized vectors from these two approaches converge to one another under mesh refinement. Formally, the IPM requires the solution of a sequence of subproblems described in Equations (4.6)-(4.7), each said subproblem solution, for a given μ, must necessarily satisfy the first-order optimality conditions

$$\boldsymbol{\nabla}_{\boldsymbol{u}}\varphi_h + \boldsymbol{J}_{\boldsymbol{u}}^\top\boldsymbol{\lambda} = \boldsymbol{\nabla}_{\boldsymbol{u}} f_h + \boldsymbol{J}_{\boldsymbol{u}}^\top\boldsymbol{\lambda} = \boldsymbol{0}, \tag{4.9}$$

$$\boldsymbol{\nabla}_{\boldsymbol{m}}\varphi_h + \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{\lambda} = \boldsymbol{\nabla}_{\boldsymbol{m}} f_h + \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{\lambda} - \boldsymbol{M}_L(\mu\boldsymbol{1}/(\boldsymbol{m}-\boldsymbol{m}_\ell)) = \boldsymbol{0}, \tag{4.10}$$

$$\boldsymbol{c} = \boldsymbol{0}, \tag{4.11}$$

where for the sake of compactness, function arguments have been omitted. Above, $\boldsymbol{\lambda}$ is a Lagrange multiplier associated to the equality constraint (4.11), $\boldsymbol{M}_L = \mathrm{diag}(\boldsymbol{M}\boldsymbol{1})$ is a diagonal lumped mass matrix, $\boldsymbol{J}_{\boldsymbol{m}} \in \mathbb{R}^{n_1\times n_2}$ and $\boldsymbol{J}_{\boldsymbol{u}} \in \mathbb{R}^{n_1\times n_1}$ are the Jacobians of the (discretized) state constraint function from (4.11) with respect to $\boldsymbol{m}$ and $\boldsymbol{u}$ respectively; furthermore, $\boldsymbol{\nabla}_{\boldsymbol{u}}, \boldsymbol{\nabla}_{\boldsymbol{m}}$ are the gradient operators with respect to $\boldsymbol{u}$ and $\boldsymbol{m}$, with which we utilize the forms of $f(u,m)$ and $c(u,m,\lambda)$ to compute quantities needed for a complete description of the optimality system (4.9)-(4.11), i.e.,

$$\begin{aligned}
(\boldsymbol{\nabla}_{\boldsymbol{u}} f_h)_i &= \frac{\delta}{\delta u_h} f(u_h,m_h)(\phi_i) := \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon} f(u_h+\varepsilon\phi_i, m_h)\right]_{\varepsilon=0},\\
(\boldsymbol{\nabla}_{\boldsymbol{m}} f_h)_i &= \frac{\delta}{\delta m_h} f(u_h,m_h)(\psi_i) := \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon} f(u_h, m_h+\varepsilon\psi_j)\right]_{\varepsilon=0},\\
(\boldsymbol{J}_{\boldsymbol{u}})_{i,j} &= \frac{\delta}{\delta u_h} c(u_h,m_h,\phi_i)(\phi_j) := \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon} c(u_h+\varepsilon\phi_j, m_h, \phi_i)\right]_{\varepsilon=0},\\
(\boldsymbol{J}_{\boldsymbol{m}})_{i,j} &= \frac{\delta}{\delta m_h} c(u_h,m_h,\phi_i)(\psi_j) := \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon} c(u_h, m_h+\varepsilon\psi_j, \phi_i)\right]_{\varepsilon=0}.
\end{aligned}$$

Finally, we note that the term $\boldsymbol{M}_L(\mu\boldsymbol{1}/(\boldsymbol{m}-\boldsymbol{m}_\ell))$, is the gradient of the log term $\mu\boldsymbol{1}^\top\boldsymbol{M}\log(\boldsymbol{m}-\boldsymbol{m}_\ell)$, of the discretized log-barrier objective.

We formally introduce the dual variable $\boldsymbol{z}_\ell := \mu\boldsymbol{1}/(\boldsymbol{m}-\boldsymbol{m}_\ell)$, associated to the bound constraint. With this formulation, the first-order optimality conditions require that a primal-dual solution $(\boldsymbol{u}^\star, \boldsymbol{m}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{z}_\ell^\star)$ of the log-barrier subproblem (4.6)-(4.7) satisfies the nonlinear system of equations

$$\boldsymbol{\nabla}_{\boldsymbol{u}} f_h + \boldsymbol{J}_{\boldsymbol{u}}^\top\boldsymbol{\lambda}^\star = \boldsymbol{0}, \tag{4.12}$$

$$\boldsymbol{\nabla}_{\boldsymbol{m}} f_h + \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{\lambda}^\star - \boldsymbol{M}_L\boldsymbol{z}_\ell^\star = \boldsymbol{0}, \tag{4.13}$$

$$\boldsymbol{c} = \boldsymbol{0}, \tag{4.14}$$

$$\boldsymbol{z}_\ell^\star \odot (\boldsymbol{m}^\star - \boldsymbol{m}_\ell) - \mu\boldsymbol{1} = \boldsymbol{0}, \tag{4.15}$$

where for $\mu > 0$, (4.15) guarantees that $\boldsymbol{z}_\ell^\star = \mu\boldsymbol{1}/(\boldsymbol{m}^\star - \boldsymbol{m}_\ell)$.

4.2.2 Newton iterations

The primal-dual nonlinear optimality system for the log-barrier subproblem (4.12)-(4.15) is solved by using a damped Newton's method. This is achieved by first linearizing Equations (4.12)-(4.15), which results in

$$\begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{u}}^\top & \boldsymbol{0} \\ \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{m}}^\top & -\boldsymbol{M}_L \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{J}_{\boldsymbol{m}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \operatorname{diag}(\boldsymbol{z}_\ell) & \boldsymbol{0} & \operatorname{diag}(\boldsymbol{m}-\boldsymbol{m}_\ell) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{m}} \\ \hat{\boldsymbol{\lambda}} \\ \hat{\boldsymbol{z}}_\ell \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{\boldsymbol{u}} \\ \boldsymbol{r}_{\boldsymbol{m}} \\ \boldsymbol{r}_{\boldsymbol{\lambda}} \\ \boldsymbol{r}_{\boldsymbol{z}_\ell} \end{bmatrix}, \tag{4.16}$$

where

$$\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} = \boldsymbol{\nabla}^2_{\boldsymbol{u},\boldsymbol{u}}(f_h + \boldsymbol{\lambda}^\top \boldsymbol{c}),\ (\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}})_{i,j} = \frac{\delta^2}{\delta u_h^2}(f(u_h,m_h) + c(u_h,m_h,\lambda_h))(\phi_i,\phi_j), \tag{4.17a}$$

$$\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}} = \boldsymbol{\nabla}^2_{\boldsymbol{m},\boldsymbol{m}}(f_h + \boldsymbol{\lambda}^\top \boldsymbol{c}),\ (\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}})_{i,j} = \frac{\delta^2}{\delta m_h^2}(f(u_h,m_h) + c(u_h,m_h,\lambda_h))(\psi_i,\psi_j), \tag{4.17b}$$

$$\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}} = \boldsymbol{\nabla}^2_{\boldsymbol{u},\boldsymbol{m}}(f_h + \boldsymbol{\lambda}^\top \boldsymbol{c}), \tag{4.17c}$$

$$\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}^\top = \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}},\ (\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}})_{i,j} = \frac{\delta^2}{\delta u_h \delta m_h}(f(u_h,m_h) + c(u_h,m_h,\lambda_h))(\phi_i,\psi_j). \tag{4.17d}$$

Furthermore, the right-hand side of the Newton linearization system (4.16) is given by

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{u}} &= \boldsymbol{\nabla}_{\boldsymbol{u}} f_h + \boldsymbol{J}_{\boldsymbol{u}}^\top \boldsymbol{\lambda}, \\ \boldsymbol{r}_{\boldsymbol{m}} &= \boldsymbol{\nabla}_{\boldsymbol{m}} f_h + \boldsymbol{J}_{\boldsymbol{m}}^\top \boldsymbol{\lambda} - \boldsymbol{M}_L \boldsymbol{z}_\ell, \\ \boldsymbol{r}_{\boldsymbol{\lambda}} &= \boldsymbol{c}, \\ \boldsymbol{r}_{\boldsymbol{z}_\ell} &= \boldsymbol{z}_\ell \odot (\boldsymbol{m} - \boldsymbol{m}_\ell) - \mu \boldsymbol{1}. \end{aligned}$$

The solution $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{m}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{z}}_\ell)$ of Equation (4.16) is the so-called Newton direction and is used to update a given optimizer estimate $(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell)$ and obtain an updated estimate $(\boldsymbol{u}^+, \boldsymbol{m}^+, \boldsymbol{\lambda}^+, \boldsymbol{z}_\ell^+)$ based on a linear update in the form of

$$\begin{bmatrix} \boldsymbol{u}^+ & \boldsymbol{m}^+ & \boldsymbol{\lambda}^+ \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{m} & \boldsymbol{\lambda} \end{bmatrix} + \alpha_p \begin{bmatrix} \hat{\boldsymbol{u}} & \hat{\boldsymbol{m}} & \hat{\boldsymbol{\lambda}} \end{bmatrix}, \tag{4.18}$$

$$\boldsymbol{z}_\ell^+ = \boldsymbol{z}_\ell + \alpha_d \, \hat{\boldsymbol{z}}_\ell, \tag{4.19}$$

where the primal and dual step-lengths α_p and α_d are computed using the convergence-enforcing filter line-search algorithm presented in Section 4.2.3.

We algebraically eliminate $\hat{\boldsymbol{z}}_\ell$ to obtain a symmetric saddle-point system of equations, $\boldsymbol{AX} = \boldsymbol{b}$ in the form

$$\underbrace{\begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{u}}^\top \\ \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{m}}^\top \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{J}_{\boldsymbol{m}} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{m}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix}}_{\boldsymbol{X}} = \underbrace{\begin{bmatrix} \boldsymbol{b}_{\boldsymbol{u}} \\ \boldsymbol{b}_{\boldsymbol{m}} \\ \boldsymbol{b}_{\boldsymbol{\lambda}} \end{bmatrix}}_{\boldsymbol{b}}, \tag{4.20}$$

where

$$\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} = \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}} + \boldsymbol{M}_L \mathrm{diag}(\boldsymbol{z}_\ell / (\boldsymbol{m} - \boldsymbol{m}_\ell)), \tag{4.21}$$

$$\boldsymbol{b}_{\boldsymbol{m}} = -(\boldsymbol{r}_{\boldsymbol{m}} + \boldsymbol{M}_L \mathrm{diag}(\boldsymbol{m} - \boldsymbol{m}_\ell) \boldsymbol{r}_{\boldsymbol{z}_\ell}), \tag{4.22}$$

$$\boldsymbol{b}_{\boldsymbol{u}} = -\boldsymbol{r}_{\boldsymbol{u}}, \quad \boldsymbol{b}_{\boldsymbol{\lambda}} = -\boldsymbol{r}_{\boldsymbol{\lambda}}. \tag{4.23}$$

We remark that once a solution to the linear system described in Equation (4.20) has been found, the bound multiplier search direction can be computed as

$$\hat{\boldsymbol{z}}_\ell = -\left[\boldsymbol{z}_\ell + \mathrm{diag}(\boldsymbol{m} - \boldsymbol{m}_\ell)^{-1}(\boldsymbol{z}_\ell \odot \hat{\boldsymbol{m}} - \mu \boldsymbol{1})\right].$$

It is common practice to symmetrize IPM linear systems (see Equation (4.16)) since direct $\boldsymbol{LDL}^\top$ factorizations can be used and are generally more stable and more efficient than $\boldsymbol{LU}$. Symmetrization is also favorable for Krylov subspace methods as generally having reduced computational requirements due to all system matrix eigenvalues being real. The reduction of Equation (4.16) to Equation (4.20) provides a symmetric linear system of reduced size at minimal computation since the computation of the second-term in $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$, $\boldsymbol{b}_{\boldsymbol{m}}$, and $\hat{\boldsymbol{z}}_\ell$ requires only elementwise vector operations. A potential disadvantage of this approach is the deterioration of the condition number of the 4 × 4 system (4.16) caused by the diagonal term from (4.21), whose condition number can grow unbounded as the IPM progresses towards the optimal solution [6, Chapter 14]. Other symmetrization strategies for (4.16) are possible. Recent work by Ghannad et al. [112] (also see related previous theoretical work by Greig et al. [113] that bound the eigenvalues and condition number of various formulations of IPM linear systems) indicates that symmetrizations of (4.16) that keep the 4 × 4 structure have better condition number than our reduced symmetric linear system (4.20) for convex problems with proper regularizations. Likely this applies to our nonconvex setup; in fact, the state-of-the-art solver Ipopt [106] works with one such 4 × 4 symmetric linear system. However, in the case of the problems considered in this work, the ill-conditioning of the reduced symmetric 3 × 3 linear system (4.20) can be effectively factored out by algebraic multigrid preconditioning, as we elaborate in Section 4.3.

4.2.3 Robust Newton IPM convergence

To address the potential lack of converge from arbitrary initial points, we use a line-search method and inertia regularization. For the line-search, we use the algorithm of Wächter and Biegler [107] for nonlinear nonconvex optimization that is based on filters [114]. In the context of the log-barrier problem (4.6)-(4.7), the salient idea behind the filter line-search is to accept the trial point (4.19) whenever the log-barrier objective (4.6) or the norm of the constraint (4.7) are smaller than their counterparts at the previous iterations. Whenever this condition is not satisfied, the step-length α_p is reduced. The so-called dual step-length α_d is chosen using a fraction-to-the-boundary rule to maintain the positiveness of the dual variables $\boldsymbol{z}_\ell$.

A notable difference from the filter line-search algorithm presented in [107] is that our method uses the inertia-free regularization technique from [115]. Inertia regularization is necessary for line-search IPMs for nonconvex problems to ensure that the solution of (4.20) is a descent direction of the log-barrier objective for points that nearly satisfy the equality constraint (4.14), and is critical to ensure global convergence of the IPM [107]. Inertia regularization essentially consists of altering the diagonal of the blocks $\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}$, $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ from (4.20) whenever the system matrix $\boldsymbol{A}$ does not have the desired inertia. Since we target large-scale problems by utilizing a linear solve strategy for Equation (4.20) based on Krylov-subspace methods, in this setting, it is not computationally feasible to compute the inertia of the system matrix. Instead, we use the inertia-free regularization from [115] that repeatedly perturbs diagonal blocks of the system matrix until the line-search passes carefully designed "curvature" tests; these tests require only matrix-vector products (in addition to adding a diagonal to the system matrix). We refer the reader to [106] for details; the previously mentioned work by the authors [111] contains the discretization considerations needed in the context of PDE-constrained optimization problems.

4.2.4 Stopping criteria

Next various metrics are presented that we use to measure progress towards solving a discretized optimization problem (4.1)-(4.3) as well as a discretized log-barrier subproblem (4.6)-(4.7). As in [111] we utilize norms, which measure proximity towards a local optimum and that are expected to be well-defined with respect to an infinite-dimensional problem formulation. Such norms are used in order to avoid discretization dependent performance of the optimization algorithm. To measure stationarity we use

$$e^{rr}_{stat}(\boldsymbol{u},\boldsymbol{m},\boldsymbol{\lambda},\boldsymbol{z}_\ell) := \sqrt{\|\boldsymbol{\nabla}_{\boldsymbol{u}} f_h + \boldsymbol{J}_{\boldsymbol{u}}^\top\boldsymbol{\lambda}\|^2_{\boldsymbol{M}_{\boldsymbol{u}}^{-1}} + \|\boldsymbol{\nabla}_{\boldsymbol{m}} f_h + \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{\lambda} - \boldsymbol{M}_L\boldsymbol{z}_\ell\|^2_{\boldsymbol{M}^{-1}}},$$

and

$$e^{rr}_{\text{feas}}(\boldsymbol{u},\boldsymbol{m}) := \|\boldsymbol{c}(\boldsymbol{u},\boldsymbol{m})\|_{\boldsymbol{M}_{\boldsymbol{u}}^{-1}},$$

to measure feasibility, where $\boldsymbol{M}_{\boldsymbol{u}}$ is the mass matrix with respect to the space $\mathcal{V}_h$, i.e., $(\boldsymbol{M}_{\boldsymbol{u}})_{i,j} = \int_\Omega \phi_i(\boldsymbol{x})\,\phi_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and as before $\boldsymbol{M}$ is the mass matrix with respect to the space $\mathcal{M}_h$. To measure complementarity we make use of

$$e^{rr}_{compl}(\boldsymbol{m}, \boldsymbol{z}_\ell; \mu) := \mathbf{1}^\top \boldsymbol{M}(\boldsymbol{z}_\ell \odot (\boldsymbol{m} - \boldsymbol{m}_\ell) - \mu \mathbf{1}).$$

These measures are combined into the single optimality measure

$$e^{rr}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell; \mu) := \max\left\{e^{rr}_{\text{stat}}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell)/s_d, e^{rr}_{\text{feas}}(\boldsymbol{u}, \boldsymbol{m}), e^{rr}_{\text{compl}}(\boldsymbol{m}, \boldsymbol{z}_\ell; \mu)/s_c\right\},$$

where the scaling values s_c and s_d are given by

$$s_c = \max\left\{s_{\max}, \|\boldsymbol{z}_\ell\|_{\boldsymbol{M}^{-1}}\right\}/s_{\max},$$
$$s_d = \max\left\{\frac{1}{2}\|\boldsymbol{\lambda}\|_{\boldsymbol{M}_{\boldsymbol{u}}^{-1}} + \frac{1}{2}\|\boldsymbol{z}_\ell\|_{\boldsymbol{M}^{-1}}, s_{\max}\right\}/s_{\max},$$

where $s_{\max}$ is a constant equal to 100 in all numerical results in this Chapter. We note that $e^{rr}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell; \mu)$ is an optimality measure for the log-barrier subproblem (4.6)-(4.7) and $e^{rr}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell; 0)$ is an optimality measure for the (discretized) PDE- and bound-constrained optimization problem (4.1)-(4.3).

4.3 Scalable IP-Newton linear system solution computation

A common approach to solve indefinite saddle-point interior-point Newton systems is a direct solver such as in [116]. Due to the undesirably high asymptotic computational cost of direct linear solves, such an approach severely limits the scalability for problems such as that studied here, wherein the dimension of the discretized optimization variable can be made arbitrarily large under mesh refinement. Here we take advantage of problem specific structure and propose a multigrid-based preconditioner, whose scaling relies only on underlying multigrid linear solves of block matrices which are known to be amenable to such treatment. The described framework achieves algorithmic scalability, in the sense that the number of outer optimization iterations and inner Krylov subspace iterations is largely independent of discretization. Due to the aforementioned algorithmic scalability and an implementation that leverages scalable scientific software libraries, we obtain a scalable means of minimizer computation.

4.3.1 Block Gauss-Seidel preconditioner performance

Here, we introduce the block Gauss-Seidel preconditioner

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{0} & \boldsymbol{J}_{\boldsymbol{u}}^\top \\ \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{m}}^\top \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{4.24}$$

to accelerate the Krylov-subspace solution of linear system (4.20). We determine, under mild assumptions for PDE-constrained optimization problems, that the block Gauss-Seidel preconditioned GMRES solves converge at a discretization independent rate. This result is obtained by utilizing the spectrum of the diagonalizable block Gauss-Seidel preconditioned IP-Newton system matrix $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$. This discretization independent linear solve convergence rate is numerically verified and reported in Table 4.3.

To make the block triangular structure clear to the reader and facilitate a quantitative discussion of the preconditioner's properties, it is useful to permute the IPM linear system (4.20) so that $\boldsymbol{A}$ becomes

$$\boldsymbol{A}_{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{V}^\top \\ \boldsymbol{V} & \boldsymbol{W} \end{bmatrix}, \tag{4.25}$$

where

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{J}_{\boldsymbol{u}}^\top \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2n_1 \times 2n_1},$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{J}_{\boldsymbol{m}}^\top \end{bmatrix} \in \mathbb{R}^{n_2 \times 2n_1},$$

and $\boldsymbol{W} = \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} \in \mathbb{R}^{n_2 \times n_2}$. The corresponding permutation of the preconditioner $\tilde{\boldsymbol{A}}$ from (4.24) is

$$\tilde{\boldsymbol{A}}_p = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{V} & \boldsymbol{W} \end{bmatrix}, \tag{4.26}$$

which makes clear why we call $\tilde{\boldsymbol{A}}$ a block Gauss-Seidel preconditioner. The rationale behind our choice of the preconditioner is that the preconditioned matrix $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$ enjoys spectral properties that are favorable for use in Krylov-subspace methods. For example, it is known that in exact arithmetic the preconditioner

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{V} & \left(\boldsymbol{W} - \boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^\top\right) \end{bmatrix}, \tag{4.27}$$

causes GMRES to converge in two iterations since the minimum polynomial for the preconditioned matrix $\boldsymbol{P}^{-1}\boldsymbol{A}_p$ is of degree two [8, Section 10.1.2]. The preconditioner $\tilde{\boldsymbol{A}}_p$ can be viewed as an approximation of the preconditioner $\boldsymbol{P}$, wherein the Schur complement $\boldsymbol{W} - \boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^\top$ is approximated by $\boldsymbol{W}$. A number of papers advocate this and related strategies for building efficient preconditioners for saddle-point linear systems e.g., [117, 118, 119] and references therein. For our block triangular preconditioner $\tilde{\boldsymbol{A}}_p$, a similar characterization is possible as inspection of the preconditioned matrix reveals that

$$\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p = \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{U}^{-1}\boldsymbol{V}^\top \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} - \boldsymbol{W}^{-1}\left(\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^\top\right) \end{bmatrix}. \tag{4.28}$$

We note that $\boldsymbol{W}^{-1}\left(\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^{\top}\right) = -\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}$, where $\hat{\boldsymbol{H}}_{\boldsymbol{d}} := (\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}})^{\top}\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}(\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}}) - \left[\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}} + (\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}})^{\top}\right]$ is the reduced-space data-misfit Hessian. By (4.28) we see that $\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p$ contains $2n_1$ eigenvalues that are equal to 1. We next show that the preconditioned matrix $\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p$ is diagonalizable and that the n_2 eigenvalues that have yet to be characterized, decay rapidly to 1, and such decay rate is independent of the discretization. This implies by [120, Proposition 4], assuming a certain character of the eigenvectors of $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$, that the block Gauss-Seidel preconditioned GMRES solver will converge at a rapid discretization independent rate.

Let $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\top} = -\boldsymbol{W}^{-1/2}\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^{\top}\boldsymbol{W}^{-1/2}$ be the (symmetric) Schur decomposition [121, Theorems 7.1.3., 8.1.1.], where $\boldsymbol{Q}$ is unitary ($\boldsymbol{Q} = \boldsymbol{Q}^{-\top}$) and $\boldsymbol{\Lambda}$ is a diagonal matrix. We now show that the diagonalized form

$$\begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} + \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q} \end{bmatrix}^{-1}, \tag{4.29}$$

where $\boldsymbol{X} = -\boldsymbol{U}^{-1}\boldsymbol{V}^{\top}(\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^{\top})^{-1}\boldsymbol{W}^{1/2}\boldsymbol{Q}$, is equal to $\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p$. We proceed by multiplying all three matrices of the diagonalized form (4.29)

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} + \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{2n_1} & -\boldsymbol{X}\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \\ \boldsymbol{0} & \boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{2n_1} & -\boldsymbol{X}\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \\ \boldsymbol{0} & (\boldsymbol{I}_{n_2} + \Lambda)\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{2n_1} & -\boldsymbol{X}\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} + \boldsymbol{X}(\boldsymbol{I}_{n_2} + \Lambda)\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q}(\boldsymbol{I} + \Lambda)\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X}\Lambda\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} + \boldsymbol{W}^{-1/2}\boldsymbol{Q}\Lambda\boldsymbol{Q}^{\top}\boldsymbol{W}^{1/2} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{U}^{-1}\boldsymbol{V}^{\top} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} - \boldsymbol{W}^{-1}\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^{\top} \end{bmatrix} \\
&= \tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p,
\end{aligned}$$

hence $\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p$ is diagonalizable. Given that the Gauss-Seidel preconditioned IP-Newton system matrix is diagonalizable we now utilize [120, Proposition 4] to bound the residual norm $\|\boldsymbol{r}^{(k+1)}\|_2$ at the kth step of GMRES as

$$\|\boldsymbol{r}^{(k+1)}\|_2 \leq \kappa(\boldsymbol{Y})\|\boldsymbol{r}^{(0)}\|_2 \min_{p\in\mathbb{P}_k, p(0)=1} \max_{1\leq j\leq n_2+2n_1} p(\lambda_j(\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A})),$$

where $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_{2n_1} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{W}^{-1/2}\boldsymbol{Q} \end{bmatrix}$ is the matrix of eigenvectors of $\tilde{\boldsymbol{A}}_p^{-1}\boldsymbol{A}_p$ and $\mathbb{P}_k$ is the set of all polynomials whose degree does not exceed k. We assume $k < n_2$ and that the eigenvalues of $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$ are given in descending order as follows

$$1 \leq \lambda_j := \lambda_j(\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}) = \begin{cases} 1 + \lambda_j(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}) & 1 \leq j \leq n_2, \\ 1 & n_2 + 1 \leq j \leq n_2 + 2n_1, \end{cases}$$

where $\lambda_1(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}) \geq \lambda_2(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}) \geq \cdots \geq \lambda_{n_2}(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})$. We choose, similarly to [120, Theorem 5], $q \in \mathbb{P}_k$ with $q(0) = 1$ as

$$q(\lambda) = \prod_{j=1}^{k}\left(1 - \frac{\lambda}{\lambda_j}\right),$$

that is then utilized to bound $\|\boldsymbol{r}^{(k+1)}\|_2$

$$\min_{p \in \mathbb{P}_k, p(0)=1} \max_{1 \leq j \leq n_2+2n_1} p(\lambda_j) \leq \max_{1 \leq j \leq n_2+2n_1} q(\lambda_j) = \max_{k+1 \leq j \leq n_2+2n_1} q(\lambda_j),$$

as by construction $q(\lambda_j) = 0$ for $j = 1, 2, \ldots, k$. We note that $(1 - \frac{\lambda}{\lambda_j})$ is a nonnegative decreasing function of λ on $[1, \lambda_j]$, so that

$$\begin{aligned} \max_{k+1 \leq j \leq n_2+2n_1} q(\lambda_j) \leq q(1) &= \prod_{j=1}^{k}\left(1 - \frac{1}{\lambda_j}\right) = \prod_{j=1}^{m}\left(\frac{\lambda_j - 1}{\lambda_j}\right) \\ &= \prod_{j=1}^{k}\left(\frac{\lambda_j(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})}{1 + \lambda_j(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})}\right) \\ &\leq \prod_{j=1}^{k}\left(\frac{\lambda_j(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})}{1 + \lambda_j(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})}\right) =: \delta_k. \end{aligned}$$

To get a sense of the behavior of δ_k we note that for a large class of PDE-constrained optimization problems the eigenvalues of the regularization preconditioned reduced-space data-misfit Hessian $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}$ decay rapidly to zero and in a mesh-independent manner [63, 78, 20], which is related to the informativeness of the data [21, Chapter 4]. Assuming $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}$, $\hat{\boldsymbol{H}}_{\boldsymbol{d}}$, $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ are symmetric semidefinite, that $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}$ are invertible and $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ and $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} \geq \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}$ one can show (see Appendix A.4) that

$$0 \leq \lambda_j(\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}) \leq \lambda_j(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}), \quad j = 1, 2, \ldots, n_2,$$

which allows one to conclude that the eigenvalues of $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}$, like the eigenvalues of $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}}$, rapidly decay to zero. Finally we have that the eigenvalues $\lambda_j(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}})$ decay rapidly and in a mesh-independent manner from which we conclude that for k large-enough, but independent of the problem discretization $\delta_k \ll 1$. Assuming that, at worst, $\kappa(\boldsymbol{Y})$ has mild growth then by $\|\boldsymbol{r}^{(k+1)}\|_2 \leq (\delta_k\,\kappa(\boldsymbol{Y}))\,\|\boldsymbol{r}^{(0)}\|_2$, we see that a small and largely mesh-independent number of Gauss-Seidel preconditioned GMRES iterations are required to achieve a specified relative reduction of the residual-norm.

There are symmetric variants of the chosen block Gauss-Seidel preconditioner, whose application only relies on similar block solves, such as

$$\tilde{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}} & \boldsymbol{J}_{\boldsymbol{u}}^\top \\ \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{0} \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \text{ and } \tilde{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}} & \boldsymbol{0} & \boldsymbol{J}_{\boldsymbol{u}}^\top \\ \boldsymbol{0} & \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} & \boldsymbol{0} \\ \boldsymbol{J}_{\boldsymbol{u}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

In practice, we find that $\tilde{\boldsymbol{A}}_1$ and $\tilde{\boldsymbol{A}}_2$ do not perform as well as the proposed preconditioner $\tilde{\boldsymbol{A}}$ for IP-Newton GMRES solves and while they are symmetric, they are guaranteed to not be positive definite [8, Theorem 3.5] and thus do not allow for use of other Krylov-subspace solvers which exploit symmetry such as MINRES [16]. A means to utilize an iterative solver that exploits symmetry is to first reformulate $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{b}$, by taking the Schur complement with respect to $\boldsymbol{m}$. The resultant linear system is $\hat{\boldsymbol{H}}\boldsymbol{x}_{\boldsymbol{m}} = \hat{\boldsymbol{b}}$, where $\hat{\boldsymbol{H}} = \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}} + \hat{\boldsymbol{H}}_{\boldsymbol{d}}$ is the reduced Hessian. Furthermore, $\hat{\boldsymbol{H}}_d$ is the reduced data-misfit Hessian and $\hat{\boldsymbol{b}}$ is the reduced right hand side, namely

$$\hat{\boldsymbol{H}}_{\boldsymbol{d}} := (\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}})^\top \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}(\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}}) - \left[\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}} + (\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{m}})^\top\right], \tag{4.30}$$

$$\hat{\boldsymbol{b}} := \boldsymbol{b}_{\boldsymbol{m}} - \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{b}_{\boldsymbol{\lambda}} - \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{J}_{\boldsymbol{u}}^{-\top}(\boldsymbol{b}_{\boldsymbol{u}} - \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{b}_{\boldsymbol{\lambda}}). \tag{4.31}$$

The reduced Hessian $\hat{\boldsymbol{H}}$ is symmetric but not necessarily positive definite, e.g., far from an optimal point, however for some problems, such as that studied here, $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ is symmetric and positive definite. Thus, one can solve the reduced Newton system $\hat{\boldsymbol{H}}\boldsymbol{x}_{\boldsymbol{m}} = \hat{\boldsymbol{b}}$ with a MINRES solver and a $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ preconditioner. The Gauss-Seidel preconditioned GMRES solve of the IP-Newton system utilized in this work and a $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ preconditioned MINRES of the reduced Newton system both require one $\boldsymbol{J}_{\boldsymbol{u}}$ solve, one $\boldsymbol{J}_{\boldsymbol{u}}$ solve and one $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ solve per Krylov subspace iteration. As seen from the discussion in Section 4.3.1 and that $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}} = \boldsymbol{I} + \boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}\hat{\boldsymbol{H}}_{\boldsymbol{d}} = \boldsymbol{I} - \boldsymbol{W}^{-1}\boldsymbol{V}\boldsymbol{U}^{-1}\boldsymbol{V}^\top$, both preconditioning strategies have identical eigenvalues which differ from unity, so that the rates of convergence for the linear solvers do not differ due to having distinct eigenvalue distributions. MINRES is a more favorable solver due to, for instance, smaller memory requirements but here we use GMRES due to its robustness.

4.3.2 Block Gauss-Seidel preconditioner cost

Having established that a GMRES solve of the block Gauss-Seidel preconditioned interior-point Newton linear system is expected to converge in a discretization independent number of Krylov-subspace iterations, we now discuss a scalable means to apply the block Gauss-Seidel preconditioner. A scalable means to apply the block

Gauss-Seidel preconditioner enables the scalable solution of the PDE- and bound-constrained optimization example problem described in Section 4.4 as reported in Figure 4.2.

As elucidated in Algorithm 3 and as expected of a block triangular matrix, applying $\tilde{\boldsymbol{A}}^{-1}$ to a vector requires a sequence of block solves. Particularly, the computational cost to apply $\tilde{\boldsymbol{A}}^{-1}$ to a vector, critically depends on the cost to apply $\boldsymbol{J}_{\boldsymbol{u}}^{-1}$, $\boldsymbol{J}_{\boldsymbol{u}}^{-\top}$ and $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}^{-1}$. The Jacobian of the PDE constraint with respect to $\boldsymbol{u}$, $\boldsymbol{J}_{\boldsymbol{u}}$, is amenable to an algebraic multigrid treatment, e.g., when the constraint (4.11) describes a discretized elliptic PDE. If $\boldsymbol{c}$ depends linearly on the parameter $\boldsymbol{m}$, then $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ is the sum of the objective Hessian with respect to the parameter and the Hessian of the log-barrier term $\boldsymbol{M}_L\text{diag}(\boldsymbol{z}_\ell/(\boldsymbol{m}-\boldsymbol{m}_\ell))$. While, the ill-conditioning of the Hessian of the log-barrier term deteriorates the performance of many linear system solution strategies, here the convergence of the algebraic multigrid preconditioned conjugate gradient (CG-AMG) $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ solve is accelerated by the (diagonal) positive-definite Hessian of the log-barrier term. For the example problem described in Section 4.4, $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}$ is the Hessian of the regularization term of the objective and the regularization is a linear combination of the squared L^2 norm of m and the squared L^2 norm of the gradient of m, for this reason the Hessian of the regularization is a linear combination of mass and stiffness matrices, hence an invertible discretized second order elliptic PDE operator. The structure of $\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}$ is part of what makes CG-AMG a performant choice for to solve linear systems with a $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ system matrix.

```
Algorithm 3 Block Gauss-Seidel preconditioner action $\boldsymbol{X} = \tilde{\boldsymbol{A}}^{-1}\boldsymbol{b}$.
Input: $\boldsymbol{b} = \begin{bmatrix}\boldsymbol{b}_{\boldsymbol{u}}^\top & \boldsymbol{b}_{\boldsymbol{m}}^\top & \boldsymbol{b}_{\boldsymbol{\lambda}}^\top\end{bmatrix}^\top$.
Output: Solution $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_{\boldsymbol{u}}^\top & \boldsymbol{x}_{\boldsymbol{m}}^\top & \boldsymbol{x}_{\boldsymbol{\lambda}}^\top\end{bmatrix}^\top$ of the linear system $\tilde{\boldsymbol{A}}\boldsymbol{X} = \boldsymbol{b}$.
Solve $\boldsymbol{J}_{\boldsymbol{u}}\boldsymbol{x}_{\boldsymbol{u}} = \boldsymbol{b}_{\boldsymbol{\lambda}}$    {AMG preconditioned CG solve}
Solve $\boldsymbol{J}_{u}^\top\boldsymbol{x}_{\boldsymbol{\lambda}} = \boldsymbol{b}_{\boldsymbol{u}} - \boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}\boldsymbol{x}_{\boldsymbol{u}}$    {AMG preconditioned CG solve}
Solve $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}\boldsymbol{x}_{\boldsymbol{m}} = \boldsymbol{b}_{\boldsymbol{m}} - \boldsymbol{H}_{\boldsymbol{m},\boldsymbol{u}}\boldsymbol{x}_{\boldsymbol{u}} - \boldsymbol{J}_{\boldsymbol{m}}^\top\boldsymbol{x}_{\boldsymbol{\lambda}}$    {AMG preconditioned CG solve}
```

The ill-conditioned positive-definite diagonal log-barrier Hessian $\boldsymbol{M}_L\text{diag}(\boldsymbol{z}_\ell/(\boldsymbol{m}-\boldsymbol{m}_\ell))$, which is a component of $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$, significantly deteriorates the computational performance of many Krylov-subspace based strategies for the IP-Newton system (4.20). However, our approach is robust against such ill-conditioning due to the following

1. The condition number of the block Gauss-Seidel preconditioned interior-point Newton system matrix, as detailed in Section 4.3.1 and demonstrated numerically in Table 4.1, weakly depends on the ill-conditioning present in $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$.

2. Generally, the convergence rate of an iterative scheme such as Jacobi iteration or a Krylov-subspace solver with a smoothing preconditioner, such as Jacobi, will increase as the diagonal dominance of the linear system matrix. Specifically,

the diagonal dominance of $\boldsymbol{W}_{\boldsymbol{m},\boldsymbol{m}}$ is increased due to presence of the log-barrier Hessian and thus the smoothing components of a multigrid scheme tend to reduce the residual norm faster.

4.4 Problem setup

Here we detail a PDE- and bound-constrained optimization problem, to determine fields $u(\boldsymbol{x})$, $m(\boldsymbol{x})$ defined over the unit square $\Omega = (0,1) \times (0,1)$, used to test the computational performance of the framework detailed in this Chapter and from which we obtain the results that are presented in Section 4.5. The objective functional is a linear combination of so-called data-misfit f_{misfit} and regularization R terms

$$f(u,m) = f_{\text{misfit}}(u) + R(m),$$

where f_{misfit} provides a discrepancy measure between u and a given data field u_d. The regularization term

$$R(m) = \frac{\gamma_1}{2}\int_\Omega m^2(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \frac{\gamma_2}{2}\int_\Omega \boldsymbol{\nabla}_{\boldsymbol{x}} m(\boldsymbol{x}) \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

acts to penalize the squared $L^2(\Omega)$ norms of $m(\boldsymbol{x})$ and $\boldsymbol{\nabla}_{\boldsymbol{x}} m(\boldsymbol{x})$, and is chosen as such to reduce the sensitivity of the solution of the optimization problem to random noise contained in the data u_d. The partial differential equality constraint, c, is the weak form of an elliptic PDE that when expressed in strong form reads

$$-\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (m \boldsymbol{\nabla}_{\boldsymbol{x}} u) + u = g, \quad \text{in } \Omega, \tag{4.32a}$$

$$m \boldsymbol{\nabla}_{\boldsymbol{x}} u \cdot \boldsymbol{n} = 0, \quad \text{on } \partial\Omega, \tag{4.32b}$$

where $\partial\Omega$ is the boundary of the spatial domain Ω, $\boldsymbol{n}$ is the outward normal to $\partial\Omega$ and g is a forcing term. The forcing term $g(\boldsymbol{x}) = -\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (m_{\text{true}}\, \boldsymbol{\nabla}_{\boldsymbol{x}} \hat{u}_d) + \hat{u}_d$, is chosen so that the noise free data $\hat{u}_d(\boldsymbol{x}) = \cos(\pi\, \boldsymbol{x}_1)\, \cos(\pi\, \boldsymbol{x}_2)$ solves the PDE for the given "true" parameter field $m_{\text{true}}(\boldsymbol{x}) = 0.5 + \boldsymbol{1}^\top \boldsymbol{x}$. The data u_d is generated by adding random 1% pointwise relative random noise to $\hat{u}_d$. It is to be noted that m_{true} does not satisfy the lower-bound constraint $m \geq m_\ell = 1.0$, and so it is expected that the optimizer $m^\star$ is active on a subset of Ω and that the inclusion of the bound-constraint has a nontrivial impact on the optimization problem. In the following, the parameters γ_1, γ_2 that define the regularization regularization function $R(m)$ are both set to 10^{-1}.

4.5 Numerical results

We next discuss two minor variants of the problem detailed in Section 4.4, which only differ by the data-misfit component f_{misfit} of the objective functional f. In

Section 4.5.1 we present a study of the condition number of the IP-Newton system matrix and the block Gauss-Seidel preconditioned IP-Newton system matrix. In Section 4.5.2 we demonstrate both the algorithmic scaling of the framework but also that scalable IPM solves are achieved due to the aforementioned favorable algorithmic scaling and the utilization of mature high-performant parallel finite element and algebraic multigrid code bases.

4.5.1 IP-Newton condition number study

In this subsection, the presented results are obtained with a serial Python implementation of the IPM framework. We utilize FEniCS [122] for the finite element discretization and all sparse finite element matrices are converted to scipy [123] sparse matrices for ease of manipulation. Furthermore, all block solves required of a block Gauss-Seidel preconditioner apply are done with sparse direct linear solvers. We utilize a pointwise observation operator available in the Inverse Problems PYthon library hIPPYlib [124], in order to define a pointwise data-misfit

$$f_{\text{misfit}}(u) = \frac{1}{2}\sum_{i=1}^{k}\left(u(\boldsymbol{x}^{(i)}) - u_d(\boldsymbol{x}^{(i)})\right)^2,$$

on a set of k points $\{\boldsymbol{x}^{(i)}\}_{i=1}^{k} \subset \overline{\Omega}$. Here and in Section 4.5.2, $\boldsymbol{x}^{(i)}$ are 25 evenly distributed points on $\overline{\Omega}$. We defer further discussion of convergence, plots of the solution fields to Section 4.5.2 and instead focus on the condition of the IP-Newton system matrix $\boldsymbol{A}$ and the block Gauss-Siedel preconditioned matrix $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$. Table 4.1 provides computational evidence that, as expected, the IP-Newton system matrix grows increasingly ill-conditioned, as the IPM progresses, due to the log-barrier Hessian and further suggests that the block Gauss-Seidel preconditioner is an effective preconditioner for this example problem. Table 4.2 shows that the condition number of the IP-Newton system matrix is relatively stable with respect to uniform mesh refinement.

ε_{opt}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
$\kappa(\boldsymbol{A})$	1.61×10^5	1.23×10^6	1.86×10^8	2.32×10^{10}
$\kappa(\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A})$	4.07×10^1	4.29×10^1	4.69×10^1	4.75×10^1

Table 4.1: Condition number $\kappa(\boldsymbol{A})$ of $\boldsymbol{A}$ for dim($\boldsymbol{m}$)= 441, and the condition number of the GS-preconditioned IP-Newton system matrix $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$, as a function of the optimality error ε_{opt}, which sets the termination condition $\varepsilon_{\text{opt}} > e^{rr}(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{\lambda}, \boldsymbol{z}_\ell; 0)$ of the IPM.

$\dim(\boldsymbol{m})$	441	961	1 681	2 601
$\kappa(\boldsymbol{A})$	1.86×10^8	2.31×10^8	2.33×10^8	6.44×10^8
$\kappa(\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A})$	4.69×10^1	4.93×10^1	5.95×10^1	6.19×10^1

Table 4.2: Condition number $\kappa(\boldsymbol{A})$ of $\boldsymbol{A}$ for $\varepsilon_{\text{opt}} = 10^{-6}$, and the condition number of the GS-preconditioned IP-Newton system matrix $\tilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$, as a function of the level of mesh refinement as indicated by the dimension of $\boldsymbol{m}$, $\dim(\boldsymbol{m})$.

4.5.2 Scaling of the IPM framework

In this subsection we present results via a performant distributed memory parallel C++ optimization package that we developed, which makes significant use of the modular finite elements library MFEM [125] for finite-element discretization and contains wrappers to hypre [126], scalable linear algebra solvers, which we specifically use for it's AMG preconditioned Krylov-subspace solvers. The package includes

- `optimizationProblem` abstract parent class which provides methods to derived classes to e.g., evaluate the gradient of the log-barrier objective from a derived class method for evaluating the objective function.
- Various helper functions, such as a `GSPreconditioner` object which inherits from `MFEM::Solver` and provides the action of $\tilde{\boldsymbol{A}}^{-1}$, with hypre based CG-AMG block solves and can be used as a preconditioner for a preconditioned Krylov-subspace solver such as GMRES.
- `interiorPtSolver` class, which utilizes an `optimizationProblem` to determine a minimizer by the interior-point filter line-search approach detailed in this Chapter.

Here, the data-misfit term of the objective functional is given by

$$f_{\text{misfit}}(u) = \frac{1}{2}\int_{\Omega} w(\boldsymbol{x})\left(u(\boldsymbol{x}) - u_d(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x},$$

where $w(\boldsymbol{x}) > 0$ provides a weighted $L^2(\Omega)$ distance between the state $u(\boldsymbol{x})$ and a data set $u_d(\boldsymbol{x})$. The weighting function $w(\boldsymbol{x})$ is chosen as a linear combination of k normalized Gaussian distributions, with variance σ^2, which are each radially symmetric about their respective means $\boldsymbol{x}^{(i)}$

$$\begin{aligned} w(\boldsymbol{x}) &= \sum_{i=1}^{k} w_i(\boldsymbol{x}), \\ w_i(\boldsymbol{x}) &= C_i (2\pi\sigma^2)^{-r/2} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\|_2^2}{2\sigma^2} \right), \end{aligned}$$

where r is the geometric dimension of Ω. The scaling factor $C_i = 4$ if $\boldsymbol{x}^{(i)}$ is a corner of Ω, $C_i = 2$ when $\boldsymbol{x}^{(i)}$ lies on a edge but not a corner and $C_i = 1$ when $\boldsymbol{x}^{(i)}$ is interior to Ω. The weights C_i are chosen as such so that $\int_\Omega w_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx 1$, for $i = 1, 2, \dots, k$. The weight function $w(\boldsymbol{x})$ is defined as it is in order that we mimic

$$\int_\Omega w(\boldsymbol{x}) \left(u(\boldsymbol{x}) - u_d(\boldsymbol{x}) \right)^2 \mathrm{d}\boldsymbol{x} \approx \sum_{i=1}^{k} \left(u(\boldsymbol{x}^{(i)}) - u_d(\boldsymbol{x}^{(i)}) \right)^2,$$

the data-misfit term detailed in Section 4.5.1. Figure 4.1 shows a representative state and parameter reconstruction, along with the bound-constraint multiplier z_ℓ, whose significantly nonzero values indicate, when strict complementarity holds, at which locations the lower-bound constraint is active. Table 4.1 shows that the IP-Newton system matrix $\boldsymbol{A}$, driven by the log-barrier Hessian, grows increasingly ill-conditioned as the interior-point optimization procedure progresses and furthermore that the Gauss-Seidel preconditioner stabilizes the conditioning of $\boldsymbol{A}$. Table 4.3 shows, by the reported number of IP-Newton linear system solves that the outer optimization loop is mesh-independent, by the average number of GMRES iterations per Newton solve we see that the performance of the GS-preconditioner does not appear to degrade under uniform mesh refinement.

Figure 4.2 shows both strong and weak scaling for computing the solution of the PDE- and bound-constrained optimization problem discussed in this Section on various numbers of Intel Xeon E5-2695 CPU's. Here it is seen that the framework has strong and weak scaling that favorably compares to the ideal strong and weak scaling.

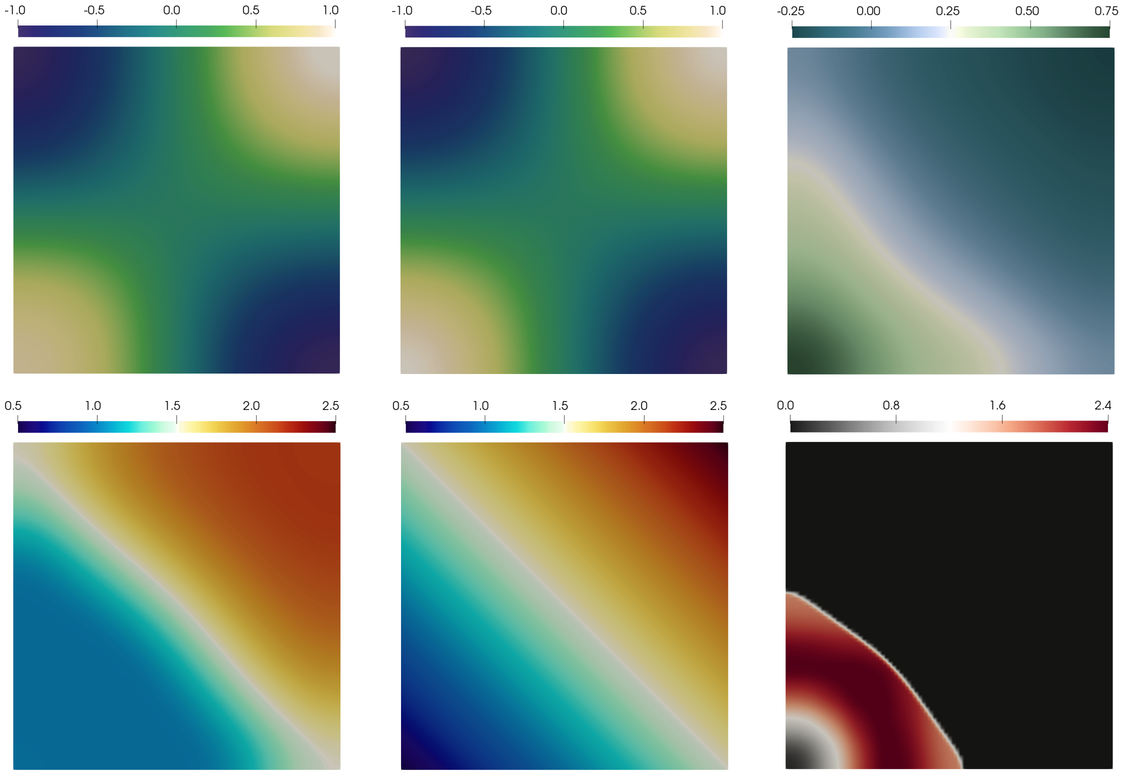
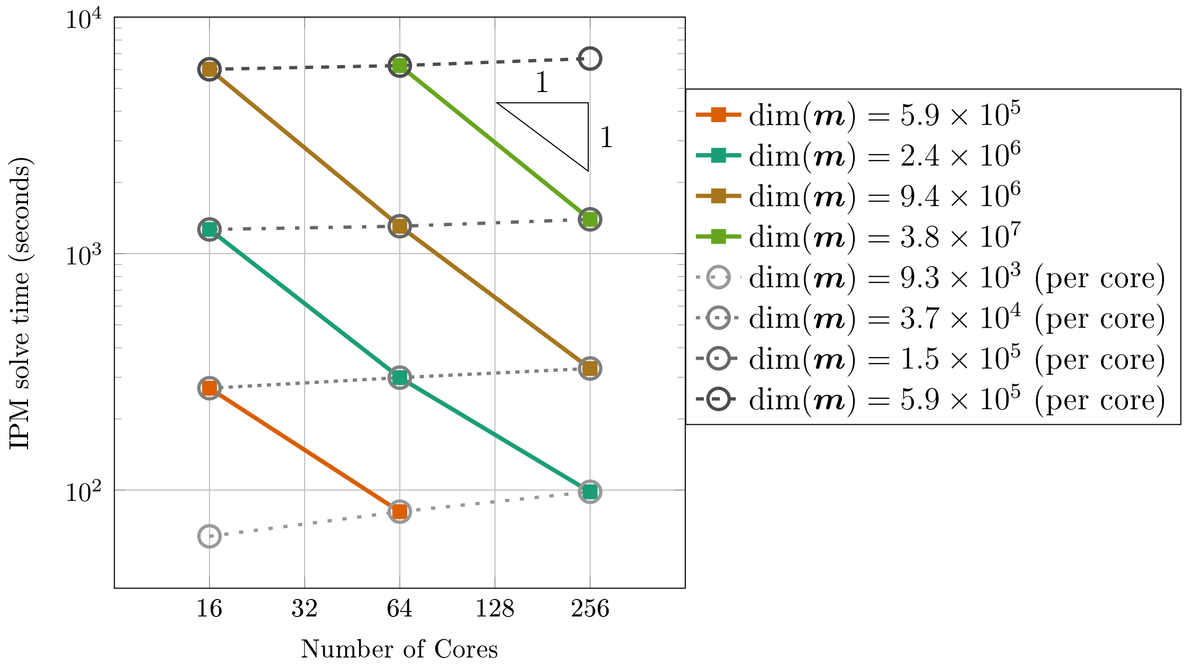


Figure 4.1: Top row: State reconstruction $u^\star$ (left), true state field $\hat{u}_d$ (middle), adjoint reconstruction $\lambda^\star$, bottom row: $m^\star$ reconstruction (left), infeasible "true" parameter m_{true} (middle), bound constraint multiplier reconstruction $z_\ell^\star$ (right). Here the dimension of each of the discretized fields is 7 921.



	IP-Newton linear solves	Preconditioned GMRES iterations
dim($\boldsymbol{m}$)	per optimizer computation	per IP-Newton linear solve
148 225	24	20.58
591 361	25	20.60
2 362 369	25	20.93
9 443 329	25	20.93
37 761 025	25	21.10
151 019 521	25	21.60

Table 4.3: Algorithmic performance metrics for the interior-point optimizer with block Gauss-Seidel preconditioning for GMRES solves of the interior-point Newton linear systems. The outer optimization tolerance is set to 10^{-6} and a GMRES solve is terminated when the norm of the preconditioned linear system residual is smaller than 10^{-14} or is reduced by a factor of 10^{-10} relative to the initial preconditioned linear system residual. The algorithmic performance is tested on a sequence of meshes with different levels of uniform refinement, as indicated by the dimension of the discretized parameter dim($\boldsymbol{m}$).

4.5.3 Example problem discretization

The partial differential equaltiy constraint whose strong form is given in Equation (4.32) is described as find $u_h \in \mathcal{V}_h$ such that

$$c(u_h, m_h, \lambda_h) = \int_\Omega (m_h \boldsymbol{\nabla}_{\boldsymbol{x}} u_h \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} \lambda_h + u_h\, \lambda_h - g\, \lambda_h)\, \mathrm{d}\boldsymbol{x} = 0, \quad \forall \lambda_h \in \mathcal{V}_h,$$

wherein $\mathcal{V}_h = \mathcal{M}_h$ are finite-dimensional subspaces of of the Sobolev space $H^1(\Omega)$. As before $\{\phi_i\}_{i=1}^{n_1}$ is a basis for $\mathcal{V}_h$ and $\{\psi_i\}_{i=1}^{n_2}$ is a basis for $\mathcal{M}_h$. The discrete constraint is then

$$\left[\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{m})\right]_i = c(u_h, m_h, \phi_i),$$

with Jacobians

$$\begin{aligned}(\boldsymbol{J_u})_{i,j} &= \int_\Omega (m_h \boldsymbol{\nabla}_{\boldsymbol{x}} \phi_i \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} \phi_j + \phi_i\, \phi_j) \mathrm{d}\boldsymbol{x}, \\ (\boldsymbol{J_m})_{i,j} &= \int_\Omega \psi_j \boldsymbol{\nabla}_{\boldsymbol{x}} u_h \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} \phi_j \mathrm{d}\boldsymbol{x}.\end{aligned}$$

The remaining IP-Newton system matrix blocks are via Equations (4.17)

$$\begin{aligned}
\left(\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{u}}\right)_{i,j} &= \int_\Omega w\,\phi_i\,\phi_j \mathrm{d}\boldsymbol{x}, \\
\left(\boldsymbol{H}_{\boldsymbol{m},\boldsymbol{m}}\right)_{i,j} &= \int_\Omega \left(\gamma_1\,\psi_i\,\psi_j + \gamma_2 \boldsymbol{\nabla}_{\boldsymbol{x}}\psi_i \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\psi_j\right) \mathrm{d}\boldsymbol{x}, \\
\left(\boldsymbol{H}_{\boldsymbol{u},\boldsymbol{m}}\right)_{i,j} &= \int_\Omega \psi_j \boldsymbol{\nabla}_{\boldsymbol{x}}\phi_i \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\lambda_h \,\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

4.6 Conclusion

We have presented a scalable, with respect to mesh refinement, means to solve elliptic PDE- and bound-constrained optimization problems that ultimately leverages the limited amount of information contained in the data that defines the optimization problem. We utilized an interior-point method to smoothen the nonsmooth, complementarity conditions which are a component of the optimality conditions. The nonsmooth complementarity equations were smoothened and then homotopy was applied in order to approach the solution of the nonsmooth equations by traversing the central path. To solve the nonlinear interior-point optimality condition subproblem, we utilized a Newton method with a globalizing filter line-search. The Newton linear system matrices are large, indefinite, of saddle-point type and poorly conditioned. Effective preconditioners are thus essential in order that the number of Krylov-subspace iterations required of a Newton linear system solve is not excessively large. We proposed a block Gauss-Seidel preconditioner and demonstrated that the spectrum of the preconditioned interior-point Newton system matrix decays rapidly to one and in a mesh-independent manner, which indicates that the number of Krylov-subspace iterations is mesh-independent. Finally, on an example problem we demonstrated the algorithmic scalability, of both the outer optimization and inner Krylov-subspace linear solutions of interior-point Newton systems. Furthermore, by utilizing mature scalable algebraic multigrid accelerated Krylov-subspace solvers for the block solves required to apply the inverse of the Gauss-Seidel preconditioner, we generated strong and weak scaling results for the time to compute the optimizer which compares well with the ideal strong and weak scaling. The work presented in this Chapter is intended to be submitted soon.

Chapter 5

Conclusion

In this thesis, we detailed Newton-based optimization methods to solve large-scale inverse problems governed by PDEs and computationally efficient means of approximating Newton linear system matrices. In particular, we utilized hierarchical matrices as a data-scalable means to approximate Hessians in inverse problems and demonstrated that such methods are effective in large-scale ice sheet inverse problems (Chapters 2 and 3). To alleviate computational challenges of generating hierarchical matrix approximations of matrix-free operators by black-box randomized peeling methods, as discussed in Chapter 2, we presented a local point spread function formalism in Chapter 3, for which a mesh-independent number of Hessian-vector products are required to generate hierarchical matrix approximations which satisfy a strong-admissibility criterion. In Chapter 4, we presented a scalable means to solve elliptic PDE- and bound-constrained optimization problems, using a full-space interior-point filter line search approach. The Newton linear systems were solved with the Krylov-subspace method GMRES and preconditioned by a block Gauss-Seidel preconditioner. Our analysis shows that the performance of the block Gauss-Seidel preconditioner does not degrade as the spatial discretization is refined or as bound-constraints are approached along the central path as it ultimately only relies on the limited, mesh-independent amount of information contained in the data that defines the inverse problem. We now outline potential future avenues of research.

- HODLR approximation of matrix-free Hessian operators by black-box peeling processes can be more computationally efficient than global low-rank Hessian approximations, as shown in Chapter 2. The cost to generate such HODLR approximations is scalable with respect to the data-dimension but unfortunately is not scalable with dimension of the discretized parameter. There have been recent advances in randomized black-box compression algorithms [127], in which the number of matrix-vector products required to generate such approximations is independent of the problem dimension. Such a compression scheme has potential to further reduce the computational cost to generate hierarchical approximations of Hessians in e.g., large-scale ice sheet inverse problems.

- The framework outlined in Chapter 4 has been shown to perform well on the presented elliptic PDE- and bound-constrained example problem. It is clear to the author that incorporating a fixed, mesh-independent number of general inequality constraints could be easily handled by a small modification of the existing framework. Furthermore applying such a framework to a broader range of objective functionals would be of interest and relevant for applications such as topology optimization [128].

Appendix A

Randomized Compression Algorithms and Additional Analysis

A.1 Randomized compression algorithms

We now describe randomized matrix-free double-pass global low-rank and HODLR compression algorithms as well as symmetric double-pass and single-pass algorithmic variants of the double-pass algorithm. The essential ideas of the randomized double-pass low-rank algorithm [43] are

1. the application of a vector $\boldsymbol{\omega} \in \mathbb{R}^N$ with random entries to a matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, yields a vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\omega}$, which is likely aligned with the dominant left singular vectors of $\boldsymbol{A}$;

2. a matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times k}$, $k \leq N$ whose columns are nearly aligned with the dominant left singular vectors of $\boldsymbol{A}$, can be used to construct an accurate low-rank approximation $\tilde{\boldsymbol{A}} = \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{A}$ of $\boldsymbol{A}$, when the singular values of $\boldsymbol{A}$ decay sufficiently fast.

Before detailing said randomized algorithms, we first provide more details concerning item 1. above. Any matrix $\boldsymbol{A}$ possesses a singular value decomposition $\boldsymbol{U}^\star \boldsymbol{\Sigma}^\star (\boldsymbol{V}^\star)^\top$ [121, Section 2.5], where $\boldsymbol{U}^\star$ is an orthonormal matrix whose columns are the left-singular vectors of $\boldsymbol{A}$, $\boldsymbol{V}^\star$ is an orthonormal matrix whose columns are the right-singular vectors and $\boldsymbol{\Sigma}^\star$ is a diagonal matrix whose diagonal elements are the singular values of $\boldsymbol{A}$. If the entries of $\boldsymbol{\omega}$ are i.i.d. and drawn from a standard normal distribution then so are the entries of $\boldsymbol{\chi}$ where $\boldsymbol{\omega} = \boldsymbol{V}^\star \boldsymbol{\chi}$. We now verify the claim that the $\boldsymbol{A}\boldsymbol{\omega}$ is likely aligned with the dominant left-singular vectors of $\boldsymbol{A}$ by determining a few statistical moments of the random variable $\Xi_k := |(\boldsymbol{u}^{(k)})^\top \boldsymbol{A}\boldsymbol{\omega}|$, the magnitude of the projection of $\boldsymbol{A\omega}$ on $\boldsymbol{u}^{(k)}$ the kth column of $\boldsymbol{U}^{\star}$

$$\mathbb{E}\left(\Xi_k\right) = \frac{2}{\pi}\sigma_k, \tag{A.1}$$

$$\mathrm{Var}\left(\Xi_k\right) = \left(1 - \frac{2}{\pi}\right)\sigma_k^2, \tag{A.2}$$

where σ_k is the kth singular value of $\boldsymbol{A}$. Equation A.1 shows that on average the magnitude of the component of $\boldsymbol{A\omega}$ along the left singular vector $\boldsymbol{u}^{(k)}$ is proportional to the associated singular value σ_k and thus $\boldsymbol{A\omega}$ is likely aligned with the dominant left singular vectors of $\boldsymbol{A}$. Equation A.2 shows that statistical fluctuations of Ξ_k are larger for the dominant left-singular vectors, which motivates the use of strategies such as oversampling, for a much more thorough analysis we refer the reader to [43].

The randomized double-pass algorithm that is presented in Algorithm 4 does not significantly differ from that in [43], specifically lines $7, 8$ and 9 are distinct. This minor modification frees us from the need to compute a (parallel) singular value decomposition (SVD) of a (distributed) $N \times k$ matrix, such as $\boldsymbol{Z}$. Here, we only need to compute an SVD of the smaller $k \times k$ matrix $\boldsymbol{R}_{\boldsymbol{Z}}$. In the distributed memory parallelism setting of Section 2.6, this algorithmic modification allows us to only require the invocation of serial SVD routines, as $\boldsymbol{R}_{\boldsymbol{Z}}$, which is typically small, is available on each processor.

Algorithm 4 Double-pass randomized SVD.
Input: $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $r \in \mathbb{N}$ desired rank and oversampling parameter $d \in \mathbb{N}$.
Output: low-rank approximation $\tilde{\boldsymbol{A}}$ of $\boldsymbol{A}$

```
 1: $k = r + d$
 2: $\boldsymbol{\Omega} = \texttt{randn}(N, k)$                  {Initiate random matrix}
 3: $\boldsymbol{Y} = \boldsymbol{A\Omega}$                         {Sample column space}
 4: $\boldsymbol{Q}_{\boldsymbol{Y}} = \texttt{orthog}(\boldsymbol{Y})$            {Orthogonalize column samples}
 5: $\boldsymbol{Z} = \boldsymbol{A}^{\top}\boldsymbol{Q}_{\boldsymbol{Y}}$                       {Sample row space}
 6: $\boldsymbol{Q}_{\boldsymbol{Z}} = \texttt{orthog}(\boldsymbol{Z})$            {Orthogonalize row samples}
 7: $\boldsymbol{R}_{\boldsymbol{Z}} = {\boldsymbol{Q}_{\boldsymbol{Z}}}^{\top}\boldsymbol{Z}$                   {Compress row samples}
 8: $\boldsymbol{R}_{\boldsymbol{Z}} = \hat{\boldsymbol{V}}\boldsymbol{\Sigma}\hat{\boldsymbol{U}}^{\top}$     {SVD of $k \times k$ compressed row sample matrix}
 9: $\boldsymbol{V} = \boldsymbol{Q}_{\boldsymbol{Z}}\hat{\boldsymbol{V}}$              {Project row space information}
10: $\boldsymbol{U} = \boldsymbol{Q}_{\boldsymbol{Y}}\hat{\boldsymbol{U}}$              {Project column space information}
11: $\tilde{\boldsymbol{A}} = \boldsymbol{U\Sigma V}^{\top}$             {Form low-rank approximation}
```

The randomized hierarchical off-diagonal low-rank algorithm proceeds by compressing off-diagonal blocks by the double-pass algorithm. The larger off-diagonal blocks are compressed prior to the compression of smaller off-diagonal blocks, via a peeling procedure [27]. Here, both $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$ are assumed to be symmetric as we seek compression of symmetric operators and computation of symmetric approximants.

Algorithm 5 Symmetric matrix-free randomized HODLR.
Input: symmetric $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, hierarchical depth $L \in \mathbb{N}$, $r_1, \dots, r_L$ desired ranks of the off-diagonal blocks at each hierarchical depth and oversampling parameter d.
Output: symmetric HODLR approximation $\tilde{\boldsymbol{A}}$ of $\boldsymbol{A}$

```
for $\ell = 1, 2, \dots, L$ do
    $k_\ell = r_\ell + d$
    $\boldsymbol{\Omega} = \texttt{zeros}(N, k_\ell)$
    for $j = 1, \dots, 2^{\ell-1}$ do
        $\boldsymbol{\Omega}(\mathcal{I}_{2j}^{(\ell)}, :) = \texttt{randn}(|\mathcal{I}_{2j}^{(\ell)}|, k_\ell)$    {Initiate structured random matrix}
    end for
    $\boldsymbol{Y} = \left(\boldsymbol{A} - \sum_{j=1}^{\ell-1} \boldsymbol{A}^{(j)}\right) \boldsymbol{\Omega}$    {Sample off-diagonal block column spaces}
    for $j = 1, \dots, 2^{\ell-1}$ do
        $\boldsymbol{Y}^{(j)} = \text{zeros}(N, k_\ell)$
        $\boldsymbol{Y}^{(j)}(\mathcal{I}_{2j-1}^{(\ell)}, :) = \boldsymbol{Y}(\mathcal{I}_{2j-1}^{(\ell)}, :)$
        $\boldsymbol{Q_Y}^{(j)} = \texttt{orthog}(\boldsymbol{Y}^{(j)})$    {Orthogonalize column samples of the level $\ell$
        off-diagonal blocks}
    end for
    $\boldsymbol{Q_Y} = \sum_{j=1}^{2^{\ell-1}} \boldsymbol{Q_Y}^{(j)}$    {Row space sampling matrix}
    $\boldsymbol{Z} = \left(\boldsymbol{A} - \sum_{j=1}^{\ell-1} \boldsymbol{A}^{(j)}\right) \boldsymbol{Q_Y}$    {Sample off-diagonal block row spaces}
    for $j = 1, \dots, 2^{\ell-1}$ do
        $\boldsymbol{Z}^{(j)} = \boldsymbol{Z}(\mathcal{I}_{2j}^{(\ell)}, :)$
        $\boldsymbol{Q_Z}^{(j)} = \texttt{orthog}(\boldsymbol{Z}^{(j)})$    {Orthogonalize row samples of the level $\ell$
        off-diagonal blocks}
        $\boldsymbol{R_Z}^{(j)} = \left(\boldsymbol{Q_Z}^{(j)}\right)^\top \boldsymbol{Z}^{(j)}$    {Compress level $\ell$ off-diagonal block row samples}
        $\boldsymbol{R_Z}^{(j)} = \hat{\boldsymbol{V}}_{2j-1}^{(\ell)} \boldsymbol{\Sigma}_{2j-1}^{(\ell)} \hat{\boldsymbol{U}}_{2j-1}^{(\ell)}$    {SVD of $k_\ell \times k_\ell$ compressed row sample matrix}
        $\boldsymbol{V}_{2j-1}^{(\ell)} = \boldsymbol{Q_Z}^{(j)} \hat{\boldsymbol{V}}_{2j-1}^{(\ell)}$    {Project row space information}
        $\boldsymbol{U}_{2j-1}^{(\ell)} = \boldsymbol{Q_Y}^{(j)} \hat{\boldsymbol{U}}_{2j-1}^{(\ell)}$    {Project column space information}
        $\boldsymbol{V}_{2j}^{(\ell)} = \boldsymbol{U}_{2j-1}^{(\ell)}$
        $\boldsymbol{U}_{2j}^{(\ell)} = \boldsymbol{V}_{2j-1}^{(\ell)}$
        $\boldsymbol{\Sigma}_{2j}^{(\ell)} = \boldsymbol{\Sigma}_{2j-1}^{(\ell)}$
    end for
    $\boldsymbol{A}^{(\ell)} = \sum_{j=1}^{2^\ell} \boldsymbol{U}_j^{(\ell)} \boldsymbol{\Sigma}_j^{(\ell)} \left(\boldsymbol{V}_j^{(\ell)}\right)^\top$
end for
obtain block diagonal $\boldsymbol{D}$ of $\boldsymbol{A}$ by sampling $\boldsymbol{A} - \sum_{j=1}^{L} \boldsymbol{A}^{(j)}$
$\tilde{\boldsymbol{A}} = \boldsymbol{D} + \sum_{\ell=1}^{L} \boldsymbol{A}^{(\ell)}$
```

We now describe a few randomized global low-rank algorithmic variants of the double-pass algorithm. Once more, we mention that if the columns of $\boldsymbol{Q}$, obtained from random column samples of $\boldsymbol{A}$ are nearly aligned with the dominant eigenvectors of $\boldsymbol{A}$, then $\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{A}$ can well approximate $\boldsymbol{A}$. Symmetry is frequently a desirable property of a matrix and when $\boldsymbol{A}$ is symmetric, by choosing the double-pass approximant as $\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^\top$, the symmetry is preserved by the approximation scheme and does not require any additional $\boldsymbol{A}$-vector products. This symmetric low-rank approximation scheme does however incur a modest additional error as $\|\boldsymbol{A}-\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^\top\|_2 \leq 2\|\boldsymbol{A}-\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{A}\|_2$, for any symmetric matrix $\boldsymbol{A}$ and $\boldsymbol{Q}$ with orthonormal columns. In Algorithm 6 we present a means of generating an explicit approximate truncated eigenvalue decomposition of said approximation.

Algorithm 6 Double-pass randomized SVD (symmetric variant).
Input: Symmetric $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $r \in \mathbb{N}$ desired rank and oversampling parameter $d \in \mathbb{N}$.
Output: symmetric low-rank approximation $\tilde{\boldsymbol{A}}$ of $\boldsymbol{A}$

```
$k = r + d$
$\boldsymbol{\Omega} = \texttt{randn}(N, k)$                    {Initiate random matrix}
$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Omega}$                    {Sample column space}
$\boldsymbol{Q}_{\boldsymbol{Y}} = \texttt{orthog}(\boldsymbol{Y})$                    {Orthogonalize column samples}
$\boldsymbol{A}_c = \boldsymbol{Q}^\top(\boldsymbol{A}\boldsymbol{Q})$                    {Compress operator}
$\boldsymbol{A}_c = \hat{\boldsymbol{U}}\boldsymbol{\Lambda}\hat{\boldsymbol{U}}^\top$                    {SVD of compressed $k \times k$ matrix}
$\boldsymbol{U} = \boldsymbol{Q}\hat{\boldsymbol{U}}$                    {Project compressed eigenvectors}
$\tilde{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$                    {Form low-rank approximation}
```

The single-pass algorithm is an extension of the symmetric double-pass algorithm in which an approximate compressed matrix $\tilde{\boldsymbol{A}}_c$ is generated in such a way that avoids additional $\boldsymbol{A}$-vector products, thus only requiring a single-pass of matrix-vector product computations, $\boldsymbol{A}\boldsymbol{\Omega}$, which are needed to generate random samples from the column space of $\boldsymbol{A}$. The approximation of the compressed matrix $\boldsymbol{A}_c$ proceeds as follows

$$\begin{aligned}\boldsymbol{A}_c &= \boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{Q},\\ \boldsymbol{A}_c\boldsymbol{Q}^\top\boldsymbol{\Omega} &= \boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{\Omega},\\ \boldsymbol{A}_c\boldsymbol{Q}^\top\boldsymbol{\Omega} &\approx \boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{\Omega} = \boldsymbol{Q}^\top\boldsymbol{Y}.\end{aligned}$$

The approximant $\tilde{\boldsymbol{A}}_c$ is then chosen as the symmetric matrix which minimizes the objective function $J(\boldsymbol{X}) = \|\boldsymbol{X}\boldsymbol{C}-\boldsymbol{D}\|_F$, were $\boldsymbol{C} = \boldsymbol{Q}^\top\boldsymbol{\Omega}$ and $\boldsymbol{D} = \boldsymbol{Q}^\top\boldsymbol{Y}$. From the first-order necessary optimality conditions it is determined that $\tilde{\boldsymbol{A}}_c = \frac{1}{2}(\hat{\boldsymbol{A}}_c + \hat{\boldsymbol{A}}_c^\top)$ is the symmetrization of the solution of the Sylvester equation

$$\hat{\boldsymbol{A}}_c\boldsymbol{E} + \boldsymbol{E}\hat{\boldsymbol{A}}_c = \boldsymbol{F}, \text{ where } \boldsymbol{E} = \boldsymbol{C}\boldsymbol{C}^\top, \boldsymbol{F} = \boldsymbol{D}\boldsymbol{C}^\top + \boldsymbol{C}\boldsymbol{D}^\top.$$

Algorithm 7 Single-pass randomized SVD.

Input: Symmetric $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $r \in \mathbb{N}$ desired rank and oversampling parameter $d \in \mathbb{N}$.

Output: symmetric low-rank approximation $\tilde{\boldsymbol{A}}$ of $\boldsymbol{A}$

```
$k = r + d$
$\boldsymbol{\Omega} = \texttt{randn}(N, k)$                              {Initiate random matrix}
$\boldsymbol{Y} = \boldsymbol{A\Omega}$                                   {Sample column space}
$\boldsymbol{Q_Y} = \texttt{orthog}(\boldsymbol{Y})$                      {Orthogonalize column samples}
$\boldsymbol{C} = \boldsymbol{Q}^\top\boldsymbol{\Omega}$                 {Compress random sampling matrix}
$\boldsymbol{D} = \boldsymbol{Q}^\top\boldsymbol{Y}$                      {Compress column samples}
$\boldsymbol{E} = \boldsymbol{CC}^\top$, $\boldsymbol{F} = \boldsymbol{DC}^\top + \boldsymbol{CD}^\top$    {Compute Sylvester problem data}
solve($\hat{\boldsymbol{A}}_c\boldsymbol{E} + \boldsymbol{E}\hat{\boldsymbol{A}}_c = \boldsymbol{F}$)     {Sylvester equation solve}
$\tilde{\boldsymbol{A}}_c = \frac{1}{2}(\hat{\boldsymbol{A}}_c + \hat{\boldsymbol{A}}_c^\top)$            {Enforce symmetry}
$\tilde{\boldsymbol{A}}_c = \hat{\boldsymbol{U}}\boldsymbol{\Lambda}\hat{\boldsymbol{U}}$                {SVD of approximate compressed $k \times k$ matrix}
$\boldsymbol{U} = \boldsymbol{Q}\hat{\boldsymbol{U}}$                     {Project compressed eigenvectors}
$\tilde{\boldsymbol{A}} = \boldsymbol{U\Lambda U}^\top$                    {Form low-rank approximation}
```

The single-pass algorithm like the symmetric double-pass algorithm produces a symmetric approximation but does incur additional approximation errors, namely via approximating $\boldsymbol{Q}^\top\boldsymbol{AQQ}^\top\boldsymbol{\Omega}$ by $\boldsymbol{Q}^\top\boldsymbol{Y}$. Thus, a rank k approximation generated by the single-pass algorithm will likely more poorly approximate $\boldsymbol{A}$ than a rank k approximation generated by either double-pass algorithm. However, a rank k approximation generated by either of the presented double-pass algorithms requires twice the number of $\boldsymbol{A}$-vector products.

For a direct numerical comparison of the algorithms, we utilize the double-pass, symmetric double-pass and single-pass algorithms to generate rank k approximations of a data-misfit Hessian for the deterministic inverse problem detailed in [57], which is a problem for inferring a spatially distributed diffusion coefficient for an elliptic PDE which models the steady-state temperate field on a 40 × 40 quadrilateral mesh of the unit square. This problem is small enough ($\boldsymbol{H}_{\text{misfit}} \in \mathbb{R}^{1\,681\times 1\,681}$) to allow for the explicit construction and storage of the data-misfit Hessian which, in practice, is typically not the case. For this problem, an explicit representation of data-misfit Hessian $\boldsymbol{H}_{\text{misfit}}$, is made available by applying the matrix-free data-misfit Hessian to each column of the identity matrix. We make use of the generated explicit representation of the data-mifit Hessian to compute rank k randomized singular value decomposition approximation errors as measured in the Frobenius matrix norm $\|\cdot\|_F$. The approximation error computations are displayed in Figure A.1, which demonstrates that only a small amount of additional error is incurred when using the symmetric variant of the double-pass algorithm. Rank k approximations generated by the computationally cheaper single-pass algorithm do however incur significantly more approximation error. In Figure A.1, it is seen that the single-pass randomized algorithm is more sensitive to random statistical fluctuations inherent in the algorithm, as the error is nonmonotone in the rank k. The nonmonotonicity is also due to not using a common random seed for each rank k approximation and thus each rank k approximation is a distinct random realization of the output of the randomized singular value decomposition algorithms.

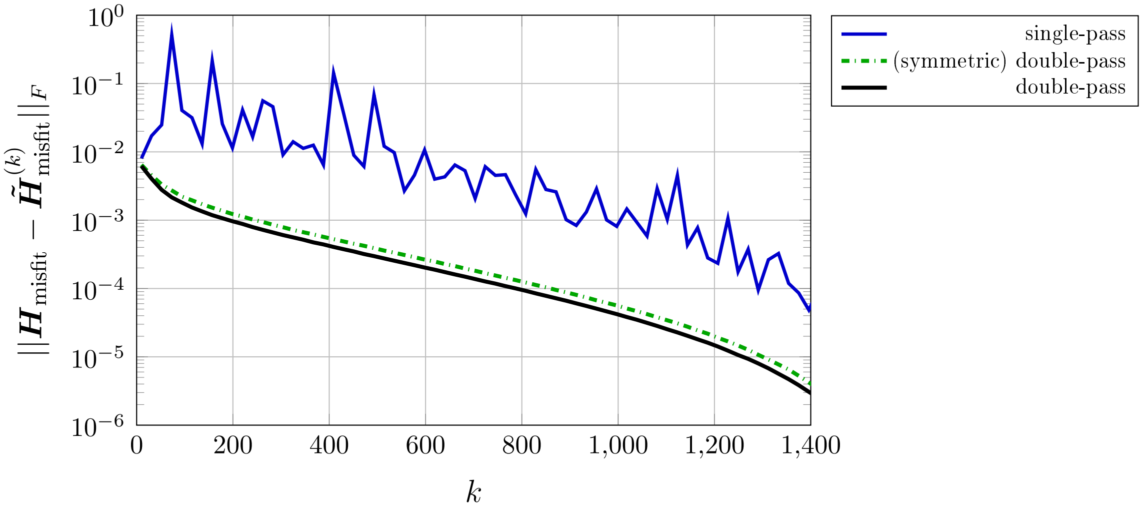


Figure A.1: $\tilde{\boldsymbol{H}}^{(k)}_{\text{misfit}}$ represents a rank k approximation of the steady state data-misfit Hessian $\boldsymbol{H}_{\text{misfit}}$. The approximation error is measured in the Frobenius norm $||\cdot||_F$.

A.2 Global HODLR approximation error from low-rank off-diagonal block approximation

Let $\boldsymbol{A}$ be a $N \times N$ matrix and consider the following partitioning

$$\boldsymbol{A}^{(1)} = \begin{bmatrix} \mathbf{0} & \boldsymbol{A}_{1,2}^{(1)} \\ \boldsymbol{A}_{2,1}^{(1)} & \mathbf{0} \end{bmatrix},$$

$$\boldsymbol{A}^{(2)} = \begin{bmatrix} \mathbf{0} & \boldsymbol{A}_{1,2}^{(2)} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{2,1}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{3,4}^{(2)} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{4,3}^{(2)} & \mathbf{0} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{A}_{1,1}^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}_{2,2}^{(2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{3,3}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{4,4}^{(2)} \end{bmatrix},$$

where $\boldsymbol{A}_{i,j}^{(\ell)}$ is the (i,j) block of a $2^\ell \times 2^\ell$ block partitioning of $\boldsymbol{A}$, where $1 \leq \ell \leq L$. $\boldsymbol{A}^{(\ell)}$ contains all blocks $\boldsymbol{A}_{i,j}^{(\ell)}$ such that $|i-j| = 1$ and $\boldsymbol{D}$ contains the diagonal blocks $\boldsymbol{A}_{i,i}^{(L)}$. Above, we show the decomposition $\boldsymbol{A} = \sum_{\ell=1}^{L} \boldsymbol{A}^{(\ell)} + \boldsymbol{D}$ for $L = 2$ hierarchical depth but in the following analysis L is arbitrary. Let $\boldsymbol{x} \in \mathbb{R}^N$, then

$$\boldsymbol{A}\boldsymbol{x} = \sum_{j=1}^{L} \boldsymbol{A}^{(j)}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{x},$$

$$\boldsymbol{A}^{(1)}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{A}_{1,2}^{(1)}\boldsymbol{x}_2^{(1)} \\ \boldsymbol{A}_{2,1}^{(1)}\boldsymbol{x}_1^{(1)} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1^{(1)} \\ \boldsymbol{x}_2^{(2)} \end{bmatrix},$$

$$\boldsymbol{A}^{(j)}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{A}_{1,2}^{(j)}\boldsymbol{x}_2^{(j)} \\ \boldsymbol{A}_{2,1}^{(j)}\boldsymbol{x}_1^{(j)} \\ \vdots \\ \boldsymbol{A}_{2^j-1,2^j}^{(j)}\boldsymbol{x}_{2^j}^{(j)} \\ \boldsymbol{A}_{2^j,2^j-1}^{(j)}\boldsymbol{x}_{2^j-1} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1^{(j)} \\ \boldsymbol{x}_2^{(j)} \\ \vdots \\ \boldsymbol{x}_{2^j-1}^{(j)} \\ \boldsymbol{x}_{2^j}^{(j)} \end{bmatrix},$$

from which we obtain the following expression

$$\|\boldsymbol{A}^{(j)}\boldsymbol{x}\|_2^2 = \sum_{k=1}^{2^{j-1}} \left(\|\boldsymbol{A}_{2k-1,2k}^{(j)}\boldsymbol{x}_{2k}^{(j)}\|_2^2 + \|\boldsymbol{A}_{2k,2k-1}^{(j)}\boldsymbol{x}_{2k-1}^{(j)}\|_2^2 \right).$$

Now assume that $\tilde{\boldsymbol{A}}$ is an HODLR approximation of $\boldsymbol{A}$, whose diagonal $\boldsymbol{D}$ is equal to the diagonal of $\boldsymbol{A}$ so that

$$\begin{aligned}\left(\boldsymbol{A}-\tilde{\boldsymbol{A}}\right) &= \sum_{j=1}^{L} \Delta\boldsymbol{A}^{(j)},\\ \Delta\boldsymbol{A}^{(j)} &:= \left(\boldsymbol{A}^{(j)}-\tilde{\boldsymbol{A}}^{(j)}\right).\end{aligned}$$

Here we assume each off-diagonal block has been approximated to some absolute tolerance $\varepsilon>0$, so that $\|\Delta\boldsymbol{A}^{(j)}_{2\,k-1,2\,k}\|_2, \|\Delta\boldsymbol{A}^{(j)}_{2\,k,2\,k-1}\| \leq \varepsilon$ for each $j=1,2,\ldots,L$ and $k=1,2,\ldots,2^{j-1}$. For $\boldsymbol{x}\in\mathbb{R}^N$ we have

$$\|\left(\boldsymbol{A}-\tilde{\boldsymbol{A}}\right)\boldsymbol{x}\|_2 \leq \sum_{j=1}^{L}\|\Delta\boldsymbol{A}^{(j)}\,\boldsymbol{x}\|_2,$$

$$\begin{aligned}\|\Delta\boldsymbol{A}^{(j)}\,\boldsymbol{x}\|_2 &= \sqrt{\sum_{k=1}^{2^{j-1}}\left(\|\Delta\boldsymbol{A}^{(j)}_{2\,k-1,2\,k}\,\boldsymbol{x}^{(j)}_{2\,k}\|_2^2+\|\Delta\boldsymbol{A}^{(j)}_{2\,k,2\,k-1}\,\boldsymbol{x}^{(j)}_{2\,k-1}\|_2^2\right)}\\ &\leq \sqrt{\sum_{k=1}^{2^{j-1}}\left(\varepsilon^2\|\boldsymbol{x}^{(j)}_{2\,k}\|_2^2+\varepsilon^2\|\boldsymbol{x}^{(j)}_{2\,k-1}\|_2^2\right)},\\ \|\Delta\boldsymbol{A}^{(j)}\,\boldsymbol{x}\|_2 &\leq \varepsilon\sqrt{\sum_{k=1}^{2^{j-1}}\left(\|\boldsymbol{x}^{(j)}_{2\,k}\|_2^2+\|\boldsymbol{x}^{(j)}_{2\,k-1}\|_2^2\right)} = \varepsilon\|\boldsymbol{x}\|_2,\\ \|\left(\boldsymbol{A}-\tilde{\boldsymbol{A}}\right)\boldsymbol{x}\|_2 &\leq \varepsilon\,L\,\|\boldsymbol{x}\|_2,\\ \|\boldsymbol{A}-\tilde{\boldsymbol{A}}\|_2 &:= \sup_{\boldsymbol{x}\neq\boldsymbol{0}}\left(\frac{\|\left(\boldsymbol{A}-\tilde{\boldsymbol{A}}\right)\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\right) \leq \varepsilon\,L.\end{aligned}$$

A.3 Analysis of posterior-covariance error due to prior-preconditioned data-misfit error

Consider a symmetric matrix $\boldsymbol{A}\in\mathbb{R}^{N\times N}$, whose eigenvalues are bounded below by a number greater than -1 and a symmetric approximant $\tilde{\boldsymbol{A}}$, with discrepancy $\Delta\boldsymbol{A}=\boldsymbol{A}-\tilde{\boldsymbol{A}}$. We signify a generic eigenvalue of $\boldsymbol{S}$ by $\lambda\left(\boldsymbol{S}\right)$ so that $s_1\leq\lambda\left(\boldsymbol{S}\right)\leq s_2$ indicates that all eigenvalues of $\boldsymbol{S}$ are bounded below by s_1 and above by s_2. Now we provide a bound for the error of $\left(\boldsymbol{I}+\boldsymbol{A}\right)^{-1}-\left(\boldsymbol{I}+\tilde{\boldsymbol{A}}\right)^{-1}$, given that $\|\Delta\boldsymbol{A}\|_2=\varepsilon$, so that one may assess the accuracy of an HODLR Gaussianized posterior covariance. When, as in Section 2.3.2, $\boldsymbol{A}$ is the prior-preconditioned Hessian misfit, $\| (\boldsymbol{I}+\boldsymbol{A})^{-1} - \left(\boldsymbol{I}+\tilde{\boldsymbol{A}}\right)^{-1} \|_2$ quantifies the discrepancy between an HODLR approximate Gaussianized posterior covariance and the true Gaussianized posterior covariance.

$$\begin{aligned}(\boldsymbol{I}+\boldsymbol{A})^{-1} - \left(\boldsymbol{I}+\tilde{\boldsymbol{A}}\right)^{-1} = (\boldsymbol{I}+\boldsymbol{A})^{-1} - (\boldsymbol{I}+\boldsymbol{A}-\Delta\boldsymbol{A})^{-1} &= \\ (\boldsymbol{I}+\boldsymbol{A})^{-1} - \left((\boldsymbol{I}+\boldsymbol{A})\left(\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right)\right)^{-1} &= \\ (\boldsymbol{I}+\boldsymbol{A})^{-1} - \left(\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right)^{-1}(\boldsymbol{I}+\boldsymbol{A})^{-1} &= \\ \left(\boldsymbol{I}-\left(\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right)^{-1}\right)(\boldsymbol{I}+\boldsymbol{A})^{-1}&.\end{aligned}$$

Given that $\|\Delta\boldsymbol{A}\|_2 = \varepsilon$, we have

$$\begin{aligned}-\varepsilon \leq \lambda(\Delta\boldsymbol{A}) &\leq \varepsilon, \\ -\varepsilon^* \leq \lambda\left((\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right) &\leq \varepsilon^*, \\ \varepsilon^* := \varepsilon(1+\lambda_{\min}(\boldsymbol{A}))^{-1}&, \\ 1+\varepsilon^* \geq \lambda\left(\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right) &\geq 1-\varepsilon^*,\end{aligned}$$

we next assume $\varepsilon^* < 1$, so that the eigenvalues of $\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}$ are necessarily positive and

$$(1+\varepsilon^*)^{-1} \leq \lambda\left(\left(\boldsymbol{I}-(\boldsymbol{I}+\boldsymbol{A})^{-1}\Delta\boldsymbol{A}\right)^{-1}\right) \leq (1-\varepsilon^*)^{-1}.$$

With this it follows that

$$\begin{aligned}\| (\boldsymbol{I}+\boldsymbol{A})^{-1} - \left(\boldsymbol{I}+\tilde{\boldsymbol{A}}\right)^{-1} \|_2 / \| (\boldsymbol{I}+\boldsymbol{A})^{-1} \|_2 &\leq \left(1-(1+\varepsilon^*)^{-1}\right) \\ \| (\boldsymbol{I}+\boldsymbol{A})^{-1} - \left(\boldsymbol{I}+\tilde{\boldsymbol{A}}\right)^{-1} \|_2 / \| (\boldsymbol{I}+\boldsymbol{A})^{-1} \|_2 &\leq \frac{\varepsilon^*}{1+\varepsilon^*},\end{aligned}$$

where, as before $\varepsilon^* = \|\Delta\boldsymbol{A}\|_2 / (1+\lambda_{\min}(\boldsymbol{A}))$.

A.4 Generalized eigenvalue ordering

Here we detail a result, that we do not believe is novel but nonetheless include it here for completeness as it is explicitly used in Section 4.3.1.

Theorem A.4.1. *Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be symmetric semidefinite and $\boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{N\times N}$ be symmetric positive definite with generalized eigenvectors $\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(i)}$ and eigenvalues $\beta_1 \geq \beta_2 \cdots \geq \beta_N \geq 0$, $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_N \geq 0$*

$$\begin{aligned}\boldsymbol{A}\boldsymbol{u}^{(i)} &= \beta_i \boldsymbol{B}\boldsymbol{u}^{(i)}, \quad 1 \leq i \leq N, \\ \boldsymbol{A}\boldsymbol{v}^{(i)} &= \xi_i \boldsymbol{C}\boldsymbol{v}^{(i)}, \quad 1 \leq i \leq N.\end{aligned}$$

Assume that $\boldsymbol{x} \in \mathbb{R}^N \implies \|\boldsymbol{x}\|_{\boldsymbol{B}}^2 = \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \geq \boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{x} = \|\boldsymbol{x}\|_{\boldsymbol{C}}^2$*, then* $\beta_k \leq \xi_k$ *for* $k = 1, 2, \ldots, N$.

Proof. For an arbitrary k, with $1 \leq k \leq N$ define E_k^u =Span$\{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{u}^{(k)}\}$ and E_k^v =Span$\{\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(k)}\}$. By utilizing the $\boldsymbol{B}$-orthogonality of the eigenvectors $\boldsymbol{u}^{(i)}$, $\left(\boldsymbol{u}^{(i)}\right)^\top \boldsymbol{B}\boldsymbol{u}^{(j)} = 0$ for $i \neq j$ it can be shown that

$$\boldsymbol{x} \in E_k^u \implies \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq \beta_k \boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{x}.$$

We now define $E_j^{v,\perp} = \{\boldsymbol{x} \in \mathbb{R}^N$ such that $\boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{v}^{(1)} = \boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{v}^{(2)} = \cdots = \boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{v}^{(j)} = 0\}$, $E_j^{u,\perp} = \{\boldsymbol{x} \in \mathbb{R}^N$ such that $\boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{u}^{(1)} = \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{u}^{(2)} = \cdots = \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{u}^{(j)} = 0\}$. Utilizing such subspaces and the $\boldsymbol{C}$-orthogonality structure of the generalized eigenvectors $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots \boldsymbol{v}^{(N)}$ we have

$$\xi_k = \sup_{\boldsymbol{x} \in E_{k-1}^{v,\perp} \setminus \{\boldsymbol{0}\}} \frac{\|\boldsymbol{x}\|_{\boldsymbol{A}}^2}{\|\boldsymbol{x}\|_{\boldsymbol{C}}^2} \geq \sup_{\boldsymbol{x} \in (E_k^u \cap E_{k-1}^{v,\perp}) \setminus \{\boldsymbol{0}\}} \frac{\|\boldsymbol{x}\|_{\boldsymbol{A}}^2}{\|\boldsymbol{x}\|_{\boldsymbol{C}}^2} \geq \beta_k,$$

where it is guaranteed that $(E_k^u \cap E_{k-1}^{v,\perp}) \setminus \{\boldsymbol{0}\} \neq \emptyset$, as E_k^u is a k-dimensional subspace of $\mathbb{R}^N$ and $E_{k-1}^{v,\perp}$ is an $(N+1-k)$-dimensional subspace of $\mathbb{R}^N$. □

Bibliography

[1] R. Courant and D. Hilbert, *Methods of mathematical physics: partial differential equations.* John Wiley & Sons, 2008.

[2] A. Borzì and V. Schulz, *Computational optimization of systems governed by partial differential equations.* SIAM, 2011.

[3] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas, "An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model," *Journal of Glaciology*, vol. 58, no. 211, pp. 889–903, 2012.

[4] T. Isaac, N. Petra, G. Stadler, and O. Ghattas, "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," *Journal of Computational Physics*, vol. 296, pp. 348–368, 2015.

[5] M. Perego, "Large-scale PDE-constrained optimization for ice sheet model initialization," *SIAM News Online*, 2022.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization.* Berlin, Heidelberg, New York: Springer Verlag, second ed., 2006.

[7] Y. Saad, *Iterative methods for sparse linear systems.* SIAM, 2003.

[8] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta numerica*, vol. 14, pp. 1–137, 2005.

[9] M. D. Gunzburger, *Perspectives in flow control and optimization.* SIAM, 2002.

[10] N. Petra and E. W. Sachs, "Second order adjoints in optimization," in *Numerical Analysis and Optimization* (M. Al-Baali, A. Purnama, and L. Grandinetti, eds.), (Cham), pp. 209–230, Springer International Publishing, 2021.

[11] I. M. Gelfand and S. Fomin, *Calculus of variations.* Dover, 1963.

[12] N. Petra and G. Stadler, "Model variational inverse problems governed by partial differential equations ices report 11-05," 2011.

[13] J. Kaipio and E. Somersalo, *Statistical and computational inverse problems*, vol. 160. Springer Science & Business Media, 2006.

[14] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler, “A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion,” *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2494–A2523, 2013.

[15] N. Petra, J. Martin, G. Stadler, and O. Ghattas, “A computational framework for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet flow inverse problems,” *SIAM Journal on Scientific Computing*, vol. 36, no. 4, pp. A1525–A1555, 2014.

[16] H. A. Van der Vorst, *Iterative Krylov methods for large linear systems.* No. 13, Cambridge University Press, 2003.

[17] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk, “Optimal low-rank approximations of Bayesian linear inverse problems,” *SIAM Journal on Scientific Computing*, vol. 37, no. 6, pp. A2451–A2487, 2015.

[18] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, “Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 407–432, 2011.

[19] A. K. Saibaba and P. K. Kitanidis, “Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems,” *Advances in Water Resources*, vol. 82, pp. 124–138, 2015.

[20] O. Ghattas and K. Willcox, “Learning physics-based models from data: perspectives from inverse problems and model reduction,” *Acta Numerica*, vol. 30, pp. 445–554, 2021.

[21] N. Alger, *Data-scalable Hessian preconditioning for distributed parameter PDE-constrained inverse problems.* PhD thesis, The University of Texas at Austin, 2019.

[22] J. Levitt, *Randomized Algorithms for Revealing Hidden Structure in Data-Sparse Matrices.* PhD thesis, The University of Texas at Austin, 2022.

[23] P.-G. Martinsson, “A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix,” *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1251–1274, 2011.

[24] N. Alger, V. Rao, A. Meyers, T. Bui-Thanh, and O. Ghattas, "Scalable matrix-free adaptive product-convolution approximation for locally translation-invariant operators," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2296–A2328, 2019.

[25] N. Alger, T. Hartland, N. Petra, and O. Ghattas, "Efficient matrix-free point spread function approximation of operators with locally supported non-negative integral kernels, with application to Hessians in PDE constrained inverse problems." To be submitted.

[26] H. Zhu, S. Li, S. Fomel, G. Stadler, and O. Ghattas, "A Bayesian approach to estimate uncertainty for full waveform inversion with a priori information from depth migration," *Geophysics*, vol. 81, no. 5, pp. R307–R323, 2016.

[27] L. Lin, J. Lu, and L. Ying, "Fast construction of hierarchical matrix representation from matrix–vector multiplication," *Journal of Computational Physics*, vol. 230, no. 10, pp. 4071–4087, 2011.

[28] P.-G. Martinsson, "Compressing rank-structured matrices via randomized sampling," *SIAM Journal on Scientific Computing*, vol. 38, no. 4, pp. A1959–A1986, 2016.

[29] C. J. Geoga, M. Anitescu, and M. L. Stein, "Scalable Gaussian process computations using hierarchical matrices," *Journal of Computational and Graphical Statistics*, vol. 29, no. 2, pp. 227–237, 2020.

[30] A. Litvinenko, Y. Sun, M. G. Genton, and D. E. Keyes, "Likelihood approximation with hierarchical matrices for large spatial datasets," *Computational Statistics & Data Analysis*, vol. 137, pp. 115–132, 2019.

[31] I. Ambartsumyan, W. Boukaram, T. Bui-Thanh, O. Ghattas, D. Keyes, G. Stadler, G. Turkiyyah, and S. Zampini, "Hierarchical matrix approximations of Hessians arising in inverse problems governed by PDEs," *SIAM Journal on Scientific Computing*, vol. 42, no. 5, pp. A3397–A3426, 2020.

[32] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[33] W. Hackbusch and S. Börm, "Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices," *Computing*, vol. 69, no. 1, pp. 1–35, 2002.

[34] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[35] A. M. Stuart, "Inverse problems: A Bayesian perspective," *Acta Numerica*, vol. 19, pp. 451–559, 2010.

[36] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[37] C. P. Robert, G. Casella, and G. Casella, *Monte Carlo statistical methods*, vol. 2. Springer, 1999.

[38] D. Rudolf and B. Sprungk, "On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm," *Foundations of Computational Mathematics*, vol. 18, no. 2, pp. 309–343, 2018.

[39] F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber, "Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions," *SIAM Journal on Scientific Computing*, vol. 37, no. 6, pp. A2733–A2757, 2015.

[40] K.-T. Kim, U. Villa, M. Parno, Y. Marzouk, O. Ghattas, and N. Petra, "hIPPYlib-MUQ: A Bayesian inference software framework for integration of data with complex predictive models under uncertainty," *arXiv preprint arXiv:2112.00713*, 2021.

[41] S. Ambikasaran and E. Darve, "An $\mathcal{O}(n \log n)$ fast direct solver for partial hierarchically semi-separable matrices," *Journal of Scientific Computing*, vol. 57, no. 3, pp. 477–501, 2013.

[42] S. Ambikasaran, M. O'Neil, and K. R. Singh, "Fast symmetric factorization of hierarchical matrices with applications," *arXiv preprint arXiv:1405.0223*, 2014.

[43] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[44] Y. Xi, J. Xia, and R. Chan, "A fast randomized eigensolver with structured LDL factorization update," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 974–996, 2014.

[45] W. Boukaram, G. Turkiyyah, and D. Keyes, "Randomized GPU algorithms for the construction of hierarchical matrices from matrix-vector operations," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. C339–C366, 2019.

[46] K. M. Cuffey and W. S. B. Paterson, *The physics of glaciers*. Academic Press, 2010.

[47] J. K. Dukowicz, S. F. Price, and W. H. Lipscomb, "Consistent approximations and boundary conditions for ice-sheet dynamics from a principle of least action," *Journal of Glaciology*, vol. 56, no. 197, pp. 480–496, 2010.

[48] E. Larour, H. Seroussi, M. Morlighem, and E. Rignot, "Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM)," *Journal of Geophysical Research: Earth Surface*, vol. 117, no. F1, 2012.

[49] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry, "Spatial patterns of basal drag inferred using control methods from a full-Stokes and simpler models for Pine Island Glacier, West Antarctica," *Geophysical Research Letters*, vol. 37, no. 14, p. L14502, 2010.

[50] M. Perego, S. Price, and G. Stadler, "Optimal initial conditions for coupling ice sheet models to Earth system models," *Journal of Geophysical Research: Earth Surface*, vol. 119, no. 9, pp. 1894–1917, 2014.

[51] J. W. Glen, "The creep of polycrystalline ice," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 228, no. 1175, pp. 519–538, 1955.

[52] F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. de Smedt, O. Gagliardini, G. H. Gudmundsson, R. C. A. Hindmarsh, A. Hubbard, J. V. Johnson, T. Kleiner, Y. Konovalov, C. Martin, A. J. Payne, D. Pollard, S. Price, M. Ruckamp, F. Saito, O. Soucek, S. Sugiya ma, and T. Zwinger, "Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP-HOM)," *The Cryosphere*, vol. 2, no. 2, pp. 95–108, 2008.

[53] Y. Daon and G. Stadler, "Mitigating the influence of boundary conditions on covariance operators derived from elliptic PDEs," *Inverse Problems and Imaging*, vol. 12, no. 5, pp. 1083–1102, 2018.

[54] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 4, pp. 423–498, 2011.

[55] M. J. Hoffman, M. Perego, S. F. Price, W. H. Lipscomb, T. Zhang, D. Jacobsen, I. Tezaur, A. G. Salinger, R. Tuminaro, and L. Bertagna, "MPAS-Albany land ice (MALI): a variable-resolution ice sheet model for Earth system modeling using Voronoi grids," *Geoscientific Model Development*, vol. 11, no. 9, pp. 3747–3780, 2018.

[56] I. K. Tezaur, M. Perego, A. G. Salinger, R. S. Tuminaro, and S. F. Price, "Albany/FELIX: a parallel, scalable and robust, finite element, first-order Stokes approximation ice sheet solver built for advanced analysis," *Geoscientific Model Development*, vol. 8, no. 4, pp. 1197–1220, 2015.

[57] K. Liegeois, M. Perego, and T. Hartland, “PyAlbany: a Python interface to the C++ multiphysics solver Albany.” Submitted.

[58] T. Trilinos Project Team, *The Trilinos Project Website*, 2020.

[59] I. Joughin, B. Smith, I. Howat, and T. Scambos, “MEaSUREs Greenland ice sheet velocity map from InSAR data, version 2,” 2015.

[60] T. R. Hillebrand, M. J. Hoffman, M. Perego, S. F. Price, and I. M. Howat, “The contribution of Humboldt Glacier, North Greenland, to sea-level rise through 2100 constrained by recent observations of speedup and retreat,” *The Cryosphere Discussions*, vol. 2022, pp. 1–33, 2022.

[61] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler, “A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion,” *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2494–A2523, 2013.

[62] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini, “Likelihood-informed dimension reduction for nonlinear inverse problems,” *Inverse Problems*, vol. 30, no. 11, p. 114015, 2014.

[63] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, “Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 407–432, 2011.

[64] N. Petra, J. Martin, G. Stadler, and O. Ghattas, “A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems,” *SIAM Journal on Scientific Computing*, vol. 36, no. 4, pp. A1525–A1555, 2014.

[65] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk, “Optimal low-rank approximations of Bayesian linear inverse problems,” *SIAM Journal on Scientific Computing*, vol. 37, no. 6, pp. A2451–A2487, 2015.

[66] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin, “On the compression of low rank matrices,” *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1389–1404, 2005.

[67] I. Ambartsumyan, W. Boukaram, T. Bui-Thanh, O. Ghattas, D. Keyes, G. Stadler, G. Turkiyyah, and S. Zampini, “Hierarchical matrix approximations of Hessians arising in inverse problems governed by PDEs,” *SIAM Journal on Scientific Computing*, vol. 42, no. 5, pp. A3397–A3426, 2020.

[68] T. Hartland, N. Petra, and P. Mauro, "Towards Inversion of the Basal Sliding Coefficient for the Humboldt Glacier in an Uncertain Ice Sheet Model," in *CSRI Summer Proceedings 2021*, pp. 347–359, 2021.

[69] A. Alexanderian, P. J. Gloor, and O. Ghattas, "On Bayesian A-and D-optimal experimental designs in infinite dimensions," *Bayesian Analysis*, vol. 11, no. 3, pp. 671–695, 2016.

[70] N. Alger, U. Villa, T. Bui-Thanh, and O. Ghattas, "A data scalable augmented Lagrangian KKT preconditioner for large-scale inverse problems," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, pp. A2365–A2393, 2017.

[71] S. Börm, L. Grasedyck, and W. Hackbusch, "Introduction to hierarchical matrices with applications," *Engineering analysis with boundary elements*, vol. 27, no. 5, pp. 405–422, 2003.

[72] W. Hackbusch, "A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[73] L. Lin, J. Lu, and L. Ying, "Fast construction of hierarchical matrix representation from matrix–vector multiplication," *Journal of Computational Physics*, vol. 230, no. 10, pp. 4071–4087, 2011.

[74] P.-G. Martinsson, "A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1251–1274, 2011.

[75] P.-G. Martinsson, "Compressing rank-structured matrices via randomized sampling," *SIAM Journal on Scientific Computing*, vol. 38, no. 4, pp. A1959–A1986, 2016.

[76] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica*, vol. 29, pp. 403–572, 2020. Section 20.

[77] N. Alger, V. Rao, A. Myers, T. Bui-Thanh, and O. Ghattas, "Scalable matrix-free adaptive product-convolution approximation for locally translation-invariant operators," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2296–A2328, 2019.

[78] T. Isaac, N. Petra, G. Stadler, and O. Ghattas, "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," *Journal of Computational Physics*, vol. 296, pp. 348–368, 2015.

[79] P. Le Tallec and A. Patra, "Non-overlapping domain decomposition methods for adaptive hp approximations of the Stokes problem with discontinuous pressure fields," *Computer Methods in Applied Mechanics and Engineering*, vol. 145, no. 3-4, pp. 361–379, 1997.

[80] Y. Saad and M. Sosonkina, "Distributed Schur complement techniques for general sparse linear systems," *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1337–1356, 1999.

[81] J. Chen and M. L. Stein, "Linear-cost covariance functions for Gaussian random fields," *Journal of the American Statistical Association*, pp. 1–18, 2021.

[82] C. J. Geoga, M. Anitescu, and M. L. Stein, "Scalable gaussian process computations using hierarchical matrices," *Journal of Computational and Graphical Statistics*, vol. 29, no. 2, pp. 227–237, 2020.

[83] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 4, pp. 423–498, 2011.

[84] L. Denis, E. Thiébaut, and F. Soulez, "Fast model of space-variant blurring and its application to deconvolution in astronomy," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2817–2820, IEEE, 2011.

[85] J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM Journal on Scientific Computing*, vol. 19, no. 4, pp. 1063–1082, 1998.

[86] A. Fichtner and T. v. Leeuwen, "Resolution analysis by random probing," *Journal of Geophysical Research: Solid Earth*, vol. 120, no. 8, pp. 5549–5573, 2015.

[87] J. Trampert, A. Fichtner, and J. Ritsema, "Resolution tests revisited: the power of random numbers," *Geophysical Journal International*, vol. 192, no. 2, pp. 676–680, 2013.

[88] H.-M. Adorf, "Towards HST restoration with a space-variant PSF, cosmic rays and other missing data," in *The Restoration of HST Images and Spectra-II*, p. 72, 1994.

[89] J. Bigot, P. Escande, and P. Weiss, "Estimation of linear operators from scattered impulse responses," *Applied and Computational Harmonic Analysis*, vol. 47, no. 3, pp. 730–758, 2019.

[90] P. Escande and P. Weiss, "Sparse wavelet representations of spatially varying blurring operators," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2976–3014, 2015.

[91] P. Escande and P. Weiss, "Fast wavelet decomposition of linear operators through product-convolution expansions," *IMA Journal of Numerical Analysis*, vol. 42, no. 1, pp. 569–596, 2022.

[92] P. Escande, P. Weiss, and F. Malgouyres, "Spatially varying blur recovery. Diagonal approximations in the wavelet domain," tech. rep., 2012.

[93] D. Fish, J. Grochmalicki, and E. Pike, "Scanning singular-value-decomposition method for restoration of images with space-variant blur," *JOSA A*, vol. 13, no. 3, pp. 464–469, 1996.

[94] L. Denis, E. Thiébaut, F. Soulez, J.-M. Becker, and R. Mourya, "Fast approximations of shift-variant blur," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 253–278, 2015.

[95] P. Escande and P. Weiss, "Approximation of integral operators using product-convolution expansions," *Journal of Mathematical Imaging and Vision*, vol. 58, no. 3, pp. 333–348, 2017.

[96] M. Gentile, F. Courbin, and G. Meylan, "Interpolating point spread function anisotropy," *Astronomy & Astrophysics*, vol. 549, 2013.

[97] L. Grasedyck and W. Hackbusch, "Construction and arithmetics of H-matrices," *Computing*, vol. 70, no. 4, pp. 295–334, 2003.

[98] T. Hartland, G. Stadler, M. Perego, K. Liegeois, and N. Petra, "Hierarchical off-diagonal low-rank approximation of Hessians in inverse problems." Submitted.

[99] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis.* Courier Corporation, 2012.

[100] N. Petra, H. Zhu, G. Stadler, T. Hughes, and O. Ghattas, "An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model," *Journal of Glaciology*, vol. 58, no. 211, pp. 889–903, 2012.

[101] L. Roininen, J. M. Huttunen, and S. Lasanen, "Whittle-Matérn priors for Bayesian statistical inversion with applications in electrical impedance tomography," *Inverse Problems & Imaging*, vol. 8, no. 2, p. 561, 2014.

[102] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer, 1999.

[103] D. H. Jacobson, “Second-order and second-variation methods for determining optimal control: A comparative study using differential dynamic programming,” *International Journal of Control*, vol. 7, no. 2, pp. 175–196, 1968.

[104] R. D. Falgout, “An introduction to algebraic multigrid,” tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2006.

[105] J. W. Ruge and K. Stüben, “Algebraic multigrid,” in *Multigrid methods*, pp. 73–130, SIAM, 1987.

[106] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[107] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: Motivation and global convergence,” *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 1–31, 2005.

[108] A. Wächter and L. T. Biegler, “Line search filter methods for nonlinear programming: Local convergence,” *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 32–48, 2005.

[109] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical analysis*. Cengage learning, 2015.

[110] R. J. LeVeque, *Numerical methods for conservation laws*, vol. 214. Springer, 1992.

[111] C. G. Petra, M. S. D. Troya, N. Petra, Y. Choi, G. M. Oxberry, and D. Tortorelli, “On the implementation of a quasi-Newton interior-point method for PDE-constrained optimization using finite element discretizations,” *Optimization Methods and Software*, vol. tbd, no. tbd, p. tbd, 2022.

[112] A. Ghannad, D. Orban, and M. A. Saunders, “Linear systems arising in interior methods for convex optimization: a symmetric formulation with bounded condition number,” *Optimization Methods and Software*, vol. 0, no. 0, pp. 1–26, 2021.

[113] C. Greif, E. Moulding, and D. Orban, “Bounds on eigenvalues of matrices arising from interior-point methods,” *SIAM Journal on Optimization*, vol. 24, no. 1, pp. 49–83, 2014.

[114] R. Fletcher and S. Leyffer, “Nonlinear programming without a penalty function,” *Mathematical programming*, vol. 91, no. 2, pp. 239–269, 2002.

[115] N.-Y. Chiang and V. M. Zavala, “An inertia-free filter line-search algorithm for large-scale nonlinear programming,” *Computational Optimization and Applications*, vol. 64, no. 2, pp. 327–354, 2016.

[116] I. S. Duff, “MA57—a code for the solution of sparse symmetric definite and indefinite systems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 2, pp. 118–144, 2004.

[117] R. E. Bank, B. D. Welfert, and H. Yserentant, “A class of iterative methods for solving saddle point problems,” *Numerische Mathematik*, vol. 56, no. 7, pp. 645–666, 1989.

[118] W. Zulehner, “A class of smoothers for saddle point problems,” *Computing*, vol. 65, no. 3, pp. 227–246, 2000.

[119] D. Drzisga, L. John, U. Rude, B. Wohlmuth, and W. Zulehner, “On the analysis of block smoothers for saddle point problems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 2, pp. 932–960, 2018.

[120] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[121] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU press, 2013.

[122] A. Logg, K.-A. Mardal, and G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84. Springer Science & Business Media, 2012.

[123] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, “Scipy 1.0: fundamental algorithms for scientific computing in python,” *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[124] U. Villa, N. Petra, and O. Ghattas, “hIPPYlib: an extensible software framework for large-scale inverse problems governed by PDEs: part i: deterministic inversion and linearized Bayesian inference,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 47, no. 2, pp. 1–34, 2021.

[125] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, *et al.*, “MFEM: A modular finite element methods library,” *Computers & Mathematics with Applications*, vol. 81, pp. 42–74, 2021.

[126] R. D. Falgout and U. M. Yang, "hypre: A library of high performance preconditioners," in *International Conference on computational science*, pp. 632–641, Springer, 2002.

[127] J. Levitt and P.-G. Martinsson, "Linear-complexity black-box randomized compression of hierarchically block separable matrices," *arXiv preprint arXiv:2205.02990*, 2022.

[128] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications.* Springer Science & Business Media, 2013.