HIGH PERFORMANCE COMPUTING METHODS FOR EARTHQUAKE CYCLE SIMULATIONS

by

ALEXANDRE CHEN

A DISSERTATION

Presented to the Department of Computer Science
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

June 2024

DISSERTATION APPROVAL PAGE

Student: Alexandre Chen

Title: High Performance Computing Methods for Earthquake Cycle Simulations

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer Science by:

Brittany A. Erickson	Chair
Boyana Norris	Core Member
Jee Choi	Core Member
Richard Tran Mills	Core Member
Leif Karlstrom	Institutional Representative

and

Krista M. Chronister	Interim Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded June 2024

DISSERTATION ABSTRACT

Alexandre Chen

Doctor of Philosophy

Department of Computer Science

June 2024

Title: High Performance Computing Methods for Earthquake Cycle Simulations

Earthquakes often occur on complex faults of multiscale physical features, with different time scales between seismic slips and interseismic periods for multiple events. Single event, dynamic rupture simulations have been extensively studied to explore earthquake behaviors on complex faults, however, these simulations are limited by artificial prestress conditions and earthquake nucleations. Over the past decade, significant progress has been made in studying and modeling multiple cycles of earthquakes through collaborations in code comparison and verification. Numerical simulations for such earthquakes lead to large-scale linear systems that are difficult to solve using traditional methods in this field of study. These challenges include increased computation and memory demands. In addition, numerical stability for simulations over multiple earthquake cycles requires new numerical methods. Developments in High performance computing (HPC) provide tools to tackle some of these challenges. HPC is nothing new in geophysics since it has been applied in earthquake-related research including seismic imaging and dynamic rupture simulations for decades in both research and industry. However, there's little work in applying HPC to earthquake cycle modeling. This dissertation presents a novel approach to applying the latest advancements in HPC

and numerical methods to solving computational challenges in earthquake cycle simulations.

CURRICULUM VITAE

NAME OF AUTHOR: Alexandre Chen

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
Fudan University, Shanghai, China

DEGREES AWARDED:

Bachelor of Science, Physics, 2018, Fudan University

AREAS OF SPECIAL INTEREST:

Itereative Algorithms
Preconditioner
Numerical PDEs
SBP-SAT finite difference methods
High performance computing

PROFESSIONAL EXPERIENCE:

Teaching Assistant, Department of Computer Science, University of
2019S, 2019F, 2020S, 2020F, 2021W, 2022W, 2023W, 2023F
Researcher, Frontier Development Lab, June - August, 2022

GRANTS, AWARDS AND HONORS:

Graduate Teaching/Research Fellowship, University of Oregon

PUBLICATIONS:

Thomas, D. R. K., & Smith, A. B. (2009). System models for the study of procrastination in high school students. *American Journal of Everything*, 100, 7–20.

ACKNOWLEDGEMENTS

Here is an acknowledgment

To so-and-so…

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 14

1.1. Chapter One Section One 14

1.1.1. Chapter one section one sub-section one 14

1.1.1.1. Chapter one section one sub-section one sub-sub-section one 14

II. METHODOLOGY 15

2.1. Chapter Two Section One 15

III. SCIENTIFIC COMPUTING LIBRARIES AND LANGUAGES 18

3.1. PETSc 18

3.2. AmgX 20

3.3. HYPRE 21

3.4. Julia 21

IV. PROBLEMS AND FORMULATIONS 22

V. RESULTS 23

5.1. Chapter Three Section One 23

5.1.1. Chapter three section one sub-section one 23

5.1.1.1. Chapter three section one sub-section one sub-sub-section one 23

VI. CONCLUSION 24

6.1. Chapter Four Section One 24

6.1.1. Chapter four section one sub-section one 24

6.1.1.1. Chapter four section one sub-section one sub-sub-section one 24

Chapter Page

REFERENCES CITED 25

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

1.1 Chapter One Section One

1.1.1 Chapter one section one sub-section one.

Test: including section 1.1 into the file

$$y = a * x + b\nabla \tag{1.1}$$

This is a sample citation: Schwartz (2012).

1.1.1.1 Chapter one section one sub-section one sub-sub-section one.

CHAPTER II

METHODOLOGY

2.1 Chapter Two Section One

Computational modeling of the natural world involves pervasive material and geometric complexities that are hard to understand, incorporate, and analyze. The partial differential equations (PDEs) governing many of these systems are subject to boundary and interface conditions, and all numerical methods share the fundamental challenge of how to enforce these conditions in a stable manner. Additionally, applications involving elliptic PDEs or implicit time-stepping require efficient solution strategies for linear systems of equations.

Most applications in the natural sciences are characterized by multiscale features in both space and time which can lead to huge linear systems of equations after discretization. Our work is motivated by large-scale ($\sim$hundreds of kilometers) earthquake cycle simulations where frictional faults are idealized as geometrically complex interfaces within a 3D material volume and are characterized by much smaller-scale features ($\sim$microns) Erickson and Dunham (2014); Kozdon, Dunham, and Nordström (2012). In contrast to the single-event simulations, e.g. Roten et al. (2016), where the computational work at each time step is a single matrix-vector product, earthquake cycle simulations must integrate with adaptive time-steps through the slow periods between earthquakes, and are tasked with a much more costly linear solve. For example, even with upscaled parameters so that larger grid spacing can be used, the 2D simulations in Erickson and Dunham (2014) generated matrices of size $\sim 10^6$, and improved resolution and 3D domains would increase the system size to $\sim 10^9$ or greater. Because iterative schemes are most often implemented for the linear solve (since direct methods require a matrix factorization that is often too large to store in memory), it is no surprise that the sparse matrix-vector product (SpMV) arises as the main computational workhorse. The matrix sparsity and condition number depend on several physical and numerical factors including the material heterogeneity of the Earth's material properties, order of accuracy, the coordinate transformation (for irregular grids), and the mesh size. For large-scale problems, matrix-free (on-the-fly) techniques for the SpMV are fundamental when the matrix cannot be stored explicitly.

In this work, we use summation-by-parts (SBP) finite difference methods Kreiss and Scherer (1974); Mattsson and Nordström (2004); Strand (1994); Svärd and Nordström (2014), which are distinct from traditional finite difference methods in their use of specific one-sided approximations at domain boundaries that enable the highly valuable proof of stability, a necessity for numerical convergence. Weak enforcement of boundary conditions has additional superior properties over traditional methods, for example, the simultaneous-approximation-term (SAT) technique, which relaxes continuity requirements (of the grid and the solution) across physical or geometrical interfaces, with low communication overhead for efficient parallel algorithms Del Rey Fernández, Hicken, and Zingg (2014).

For these reasons SBP-SAT methods are widely used in many areas of scientific computing, from the flow over airplane wings to biological membranes to earthquakes and tsunamigenesis Erickson and Day (2016); Lotto and Dunham (2015); Nordström and Eriksson (2010); Petersson and Sjögreen (2012); Swim et al. (2011); Ying and Henriquez (2007); these studies, however, have not been developed for linear solves or were limited to small-scale simulations.

With this work, we contribute a novel iterative scheme for linear systems based on SBP-SAT discretizations where nontrivial computations arise due to boundary treatment. These methods are integrated into our existing, public software for simulations of earthquake sequences. Specifically, we make the following contributions:

– Since preconditioning of iterative methods is a hugely consequential step towards improving convergence rates, we develop a custom geometric multigrid preconditioned conjugate gradient (MGCG) algorithm which shows a near-constant number of iterations with increasing system size. The required iterations (and time-to-solution) are much lower compared to several off-the-shelf preconditioners offered by the PETSc library Balay et al. (2023), a state-of-the-art library for scientific computing.

– We develop custom, matrix-free GPU kernels (specifically for SBP-SAT methods) for computations in the volume and boundaries, which show improved performance as compared to the native, matrix-explicit implementation while requiring only a fraction of memory.

– GPU-acceleration of our resulting matrix-free, preconditioned iterative scheme shows superior performance compared to state-of-the-art methods offered by NVIDIA.

Furthermore, the ubiquity of SBP-SAT methods in modern scientific computing applications means our work has the propensity to advance scientific studies currently limited to small-scale problems.

CHAPTER III

SCIENTIFIC COMPUTING LIBRARIES AND LANGUAGES

3.1 PETSc

PETSc, which stands for Portable, Extensible Toolkit for Scientific Computation, is a software library developed primarily by Argonne National Library to facilitate the development of high-performance parallel numerical code written in C/C++, Fortran and Python. It provides a wide range of functionality for solving linear and nonlinear algebraic equations, ordinary and partial differential equations, and also optimization problems (provided by TAO) on parallel computing architectures. In addition, PETSc includes support for managing parallel PDE discretizations including parallel matrix and vector assembly routines.

Key features of PETSc includes:

- Parallelism: PETSc is designed for parallel computing, especially distributed-memory parallel computing architectures. It is intended to run efficiently on parallel computing systems where multiple processors or nodes communicate over the network via a message passing interface (MPI). These architectures include clusters, supercomputers, and other HPC platforms.

- Modularity and Extensibility: PETSc is highly modular and extensible, allowing users to combine different numerical techniques and algorithms to solve complex problems efficiently. It provides a flexible framework for implementing new algorithms and incorporating external libraries. It mainly contains the following objects

    * Algebraic objects

        - Vectors (Vec) containers for simulation solutions, right-hand sides of linear systems
        - Matrices (Mat) containers for Jacobians and operators that define linear systems
    - Solvers
        - Linear solvers based on preconditioners (PC) and Krylov subspace methods (KSP)
        - Nonlinear solvers (SNES) that use data-structure-neutral implementations of Newton-like methods
        - Time integrators (TS) for ODE/PDE, explicit, implicit, IMEX
        - Optimization (TAO) with equality and inequality constraints, first and second order Newton methods
        - Eigenvalue/Eigenvectors (SLEPc) and related algorithms
- Efficiency and Performance: PETSc is optimized for performance, with algorithms and data structures designed to minimize memory usage and maximize computational efficiency. It supports parallel matrix and vector operations as well as efficient iterative solvers and preconditioners via the objects mentioned previously
- Flexibility: PETSc supports a wide range of numerical methods and algorithms and has built-in discretization tools. It provides interfaces for solving problems in various scientific and engineering disciplines, including computational fluid dynamics (CFD), solid mechanics, etc with documented examples and tutorials for researchers.

- PETSc is portable across different computing platforms and operating systems, including UNIX/Linux, macOS, and Windows. It provides a consistent interface and functionalities across different architectures, making it easy to develop and deploy simulation code across multiple platforms.

3.2 AmgX

AmgX is a proprietary software library developed by NVIDIA to accelerate the solution of large-scale linear systems arising from finite element and finite volume discretizations typically found in computational fluid dynamics (CFD) and computational mechanics simulations on NVIDIA GPUs. AmgX stands for Algebraic Multigrid Accelerated. It provides wrappers to work with other libraries like PETSc and programming languages like Julia.

Key features of AMGX include:

- Preconditioning: AmgX offers a variety of advanced preconditioning techniques, including algebraic multigrid (AMG), smoothed aggregation and hybrid methods to accelerate the convergence of iterative solvers for sparse linear systems. These preconditioners are designed for and tested in real-world engineering problems in collaboration with companies like ANSYS, a provider of leading CFD software Fluent.

- Parallelism: AmgX is optimized for NVIDIA GPUs and provides support for OpenMP to allow acceleration via heterogeneous computing and MPI to run large simulations across multiple GPUs and clusters.

- Flexibility and Customization: AmgX offers a flexible and extensible framework for configuring and customizing the solver algorithms via JSON files.

The limitation of AmgX is due to its link with NVIDIA. It can not run on GPUs from other vendors, such as AMD and Intel. Some of the latest exascale supercomputers are built with CPUs and GPUs from AMD and Intel.

3.3 HYPRE

HYPRE is a software library of high performance numerical algorithms including preconditioners and solvers for large, sparse linear systems of equations on massively parallel computers Falgout, Jones, and Yang (2006b).

Falgout, Jones, and Yang (2006a)

3.4 Julia

CHAPTER IV

PROBLEMS AND FORMULATIONS

CHAPTER V

RESULTS

5.1 Chapter Three Section One

5.1.1 Chapter three section one sub-section one.

5.1.1.1 Chapter three section one sub-section one sub-sub-section one.

CHAPTER VI

CONCLUSION

6.1 Chapter Four Section One

6.1.1 Chapter four section one sub-section one.

6.1.1.1 Chapter four section one sub-section one sub-sub-section one. This is a sample citation: Schwartz (2012).

REFERENCES CITED

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., ... Zhang, J. (2023). *PETSc Web page.* `https://petsc.org/`. Retrieved from `https://petsc.org/`

Del Rey Fernández, D. C., Hicken, J. E., & Zingg, D. W. (2014). Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids*, *95*, 171-196.

Erickson, B. A., & Day, S. M. (2016). Bimaterial effects in an earthquake cycle model using rate-and-state friction. *Journal of Geophysical Research: Solid Earth*, *121*, 2480–2506. doi: 10.1002/2015JB012470

Erickson, B. A., & Dunham, E. M. (2014). An efficient numerical method for earthquake cycles in heterogeneous media: Alternating subbasin and surface-rupturing events on faults crossing a sedimentary basin. *Journal of Geophysical Research: Solid Earth*, *119*(4), 3290–3316. doi: 10.1002/2013JB010614

Falgout, R. D., Jones, J. E., & Yang, U. M. (2006a, jan). Conceptual interfaces in hypre. *Future Gener. Comput. Syst.*, *22*(1), 239–251.

Falgout, R. D., Jones, J. E., & Yang, U. M. (2006b). The design and implementation of hypre, a library of parallel high performance preconditioners.. Retrieved from `https://api.semanticscholar.org/CorpusID:3237430`

Kozdon, J. E., Dunham, E. M., & Nordström, J. (2012). Interaction of waves with frictional interfaces using summation-by-parts difference operators: Weak enforcement of nonlinear boundary conditions. *Journal of Scientific Computing*, *50*, 341-367. doi: 10.1007/s10915-011-9485-3

Kreiss, H., & Scherer, G. (1974). Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical aspects of finite elements in partial differential equations; proceedings of the symposium* (pp. 195–212). Madison, WI. doi: 10.1016/b978-0-12-208350-1.50012-1

Lotto, G. C., & Dunham, E. M. (2015). High-order finite difference modeling of tsunami generation in a compressible ocean from offshore earthquakes. *Computational Geosciences*, *19*(2), 327–340. doi: 10.1007/s10596-015-9472-0

Mattsson, K., & Nordström, J. (2004). Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics*, *199*(2), 503–540. doi: 10.1016/j.jcp.2004.03.001

Nordström, J., & Eriksson, S. (2010). Fluid structure interaction problems: The necessity of a well posed, stable and accurate formulation. *Communications in Computational Physics*, *8*(5), 1111–1138. doi: https://doi.org/10.4208/cicp.260409.120210a

Petersson, N. A., & Sjögreen, B. (2012). Stable and efficient modeling of anelastic attenuation in seismic wave propagation. *Communications in Computational Physics*, *12*(1), 193–225. doi: 10.4208/cicp.201010.090611a

Roten, D., Cui, Y., Olsen, K. B., Day, S. M., Withers, K., Savran, W. H., . . . Mu, D. (2016). High-frequency nonlinear earthquake simulations on petascale heterogeneous supercomputers. In *Sc '16: Proceedings of the international conference for high performance computing, networking, storage and analysis* (p. 957-968). doi: 10.1109/SC.2016.81

Schwartz, S. H. (2012). An overview of the Schwartz theory of basic values. *Online Readings in Psychology and Culture*, *2*(1), 1–20.

Strand, B. (1994). Summation by parts for finite difference approximations for d/dx. *Journal of Computational Physics*, *110*(1), 47–67. doi: 10.1006/jcph.1994.1005

Svärd, M., & Nordström, J. (2014). Review of summation-by-parts schemes for initial-boundary-value problems. *Journal of Computational Physics*, *268*, 17-38. doi: https://doi.org/10.1016/j.jcp.2014.02.031

Swim, E., Benra, F.-K., Dohmen, H. J., Pei, J., Schuster, S., & Wan, B. (2011). A comparison of one-way and two-way coupling methods for numerical analysis of fluid-structure interactions. *Journal of Applied Mathematics*, *2011*, 1–16. doi: 10.1155/2011/853560

Ying, W., & Henriquez, C. (2007). Hybrid finite element method for describing the electrical response of biological cells to applied fields. *IEEE Transactions on Bio-medical Engineering*, *54*(4), 611–620. doi: 10.1109/TBME.2006.889172