SAND2022-1400R

Randomized Cholesky Preconditioning for Graph Partitioning Applications

ORISE NNSA-MSIIP Report

Name: Heliezer J.D. Espinoza
Hosting Site: Sandia National Laboratories
Mentors: Dr. Erik Boman & Dr. Jennifer Loe

Mentor's Signatures: ______________________ ______________________
Erik G. Boman, Ph.D. Jennifer Loe, Ph.D.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Abstract

Graph partitioning has emerged as an area of interest due to its use in various applications in computational research. One way to partition a graph is to solve for the eigenvectors of the corresponding graph Laplacian matrix. This project focuses on the eigensolver LOBPCG and the evaluation of a new preconditioner: Randomized Cholesky Factorization (rchol). This proconditioner was tested for its speed and accuracy against other well-known preconditioners for the method. After experiments were run on several known test matrices, rchol appears to be a better preconditioner for structured matrices. This research was sponsored by National Nuclear Security Administration Minority Serving Institutions Internship Program (NNSA-MSIIP) and completed at host facility Sandia National Laboratories [1]. As such, after discussion of the research project itself, this report contains a brief reflection on experience gained as a result of participating in the NNSA-MSIIP.

[1]Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

1 Introduction/Motivation

A graph is a mathematical representation of a network; it consists of a set of vertices, which are connected by edges. Graphs have numerous applications in various fields, as they can model all sorts of connections, processes, or relations. For example, graphs can model intricate transit systems or the human nervous system. However, graphs that are large become difficult to analyze. This is why there is an increased interest in the area of graph partitioning, or reducing the size of the graph into multiple partitions. For example, partitions of a graph representing a social network might help identify clusters of friends.[1]

2 Project Description

2.1 Mathematical Background

There are different ways to solve graph partitioning problems. For this work, we focus on a spectral partitioning method which forms a partition on the graph Laplacian L, which is an $n \times n$ matrix representation of a graph. By construction, L is symmetric and positive semi-definite, which means it has n non-negative, real-valued eigenvalues $(0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n)$. Recall, for a square matrix W, the equation $Wx = \lambda x$ is known as the the eigenvalue-eigenvector problem, where λ is the eigenvalue and x is the eigenvector.

This research project utilizes the Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG) to compute eigenvalues and eigenvectors of graph Laplacians.[2][3] To accelerate the convergence of LOBPCG, we apply a preconditioner to the internal solve of a linear system, which we denote as $Wx = b$. A *preconditioner* allows us to replace the original problem $Wx = b$ with an equivalent system that is easier for the solver to tackle. We study three different preconditioners for LOBPCG. Jacobi, a standard preconditioner, is a diagonal matrix whose entries are found by taking the reciprocals of the diagonal of W. The second preconditioner of interest, incomplete Cholesky factorization (ichol), is found through Cholesky factorization of W such that $W \approx U^T U$, where U is an upper triangular matrix with real and positive diagonal entries, and U^T denotes the conjugate transpose of U. However, ichol differs from standard Cholesky factorization by having "no fill," meaning that if an entry $w_{ij} = 0$ on the original matrix W, then the corresponding entry in U will also be zero ($u_{ij} = 0$). The final preconditioner of interest is a newly developed preconditioner: randomized Cholesky factorization (rchol). rchol approximates the Cholesky factorization for a graph Laplacian, L using a randomized algorithm.[4]

2.2 Results

As mentioned, three different types of preconditioners were applied to LOBPCG to solve the eigenvalue-eigenvector problem on different test matrices W. The test matrices are a subset of those tested in Acer, et. al, which readers may consult for further details.[5] These test matrices fall into two categories: structured matrices and unstructured matrices. The entries of structured sparse matrices have a pattern-like/formulaic relationship. For example, a finite difference matrix is a structured sparse matrix. By contrast, an unstructured matrix, perhaps from a social network graph, has no such relationship between its entries. As such, there can be a lot of unpredictable clustering for these matrices.

The results of the LOBPCG runs on the test matrices are listed in Table 1 below. The variables for the table are as follows:

- *tf*: time it takes to compute the factorization of the graph Laplacian for the preconditioner.
- *tt*: total time, including *tf*, LOBPCG took to complete a run, either by reaching the max number of iterations (2000), or by converging beyond the preset tolerance 1e−3.
- *nit*: number of iterations LOBPCG took to converge.
- *relres*: final relative residual norm of λ_5, a measure of the accuracy of the computed eigenvalue-eigenvector solutions. A smaller value indicates more accuracy.

The bold entries for each matrix highlight the preconditioning option with the fastest LOBPCG solve time. Reported solve times and corresponding iteration counts are taken from the median solve time of three runs.

Test Matrix	Jacobi				ichol				rchol			
Name	tf	tt	nit	relres	tf	tt	nit	relres	tf	tt	nit	relres
ecology1	0.03	282	313	9.79E-04	0.03	94	99	9.30E-04	**1.23**	**17**	**27**	**9.98E-04**
dielFilterV2real	0.19	241	293	9.61E-04	1.47	154	144	9.67E-04	**15**	**30**	**11**	**6.88E-04**
thermal2	0.05	351	400	9.62E-04	0.16	172	174	9.45E-04	**2.11**	**39**	**52**	**1.00E-03**
Bump_2911	0.28	1,213	396	9.93E-04	1.86	467	122	9.84E-04	**74**	**164**	**13**	**6.84E-04**
100ˆ3	0.07	187	276	9.31E-04	0.24	74	83	9.98E-04	**12**	**28**	**9**	**4.06E-04**
200ˆ3	0.55	3,153	590	9.10E-04	1.90	1,197	176	8.61E-04	**289**	**489**	**9**	**7.80E-04**
hollywood	0.14	219	192	9.17E-04	**28.4**	**85**	**26**	**5.00E-04**	58	261	54	7.44E-04
cit-Patents	0.16	620	156	7.46E-04	**3.77**	**156**	**23**	**8.10E-04**	33	441	121	7.17E-04
wb-edu	0.57	4,945	655	7.00E-04	**4.74**	**444**	**66**	**7.63E-04**	23	7258	785	1.00E-03

Table 1: LOBPCG results for varying test matrices with Jacobi, ichol, and rchol preconditioning. Observed parameters are: *tf*- the time of the preconditioner computation; *tt*- the total LOBPCG solve time; *nit*- the number of iterations LOBPCG took to converge; and *relres*- the final relative residual norm of λ_5. Note that the maximum number of iterations for all LOBPCG runs was 2000 and the relative residual tolerance given for the system convergence was 1e$-$3.

For many of our test matrices, rchol preconditioning gives the fastest solve time. However, rchol is not the best preconditioner for every matrix. Interestingly, ichol is the fastest LOBPCG preconditioner for the last 3 test matrices (hollywood, cit-Patents, and wb-edu). While this result is seemingly arbitrarily, the fastest LOBPCG applied preconditioner seems to correspond to whether a matrix is structured or not. The first set of test matrices are all structured matrices, while the final three are unstructured matrices. As such, these results seem to suggest that rchol is an outstanding LOBPCG preconditioner for structured matrices, while ichol remains a better LOBPCG preconditioner for unstructured matrices.

3 Contributions Made to the Project

Much of my work done in this project was done independently. While my Sandia mentors both proposed main goals of the project and introduced background to be able to understand the topic better, the results were computed on my own. Once I learned the prerequisites, the experimentation was my main contribution to the project. Performing the experiments involved writing test code in MATLAB, selecting appropriate test problems and parameters, running experiments on a remote computing workstation, and tabulating and analyzing results. Furthermore, I have proposed potential ways to continue this project either for my own research or for a future ORISE student. One proposal includes an idea to experiment with a better direct comparison between ichol and rchol. As presented in Chen, et.al, it is difficult (and perhaps a bit unfair) to compare the ichol and rchol preconditioned systems as was done in this project. The issue here is that standard ichol has zero fill while rchol has a randomized fill. Table 4 from Chen suggests that one way to better compare the two preconditioned systems is to perform incomplete Cholesky (ichol) with threshold dropping. In threshold dropping, there is a drop tolerance that can be adjusted to increase the fill of ichol. Thus, the drop tolerances in ichol can be altered so that the fill ratio matches that of rchol, giving a fair comparison.[4] Future work may also include testing the rchol preconditioner within the larger Trilinos[6] graph partitioning package, Sphynx[5].

4 Skills & Knowledge Gained

As MATLAB was the main software tool used for this project, I gained knowledge and understanding of MATLAB documentation for various processes and debugging techniques needed over the course of this internship program. To complicate matters, I needed to use MATLAB through remote command-line-only access of Sandia workstations for many components of this project, such as for accessing the installed rchol code. Because of this, I gained Linux command line knowledge to use Sandia remote workstations. Lastly, I learned Git, both through the Linux command line and through the GitHub/GitLab websites, to properly collaborate with my Sandia program mentors. Learning to use these software tools comprised a significant portion of the work of this internship.

5 Experience Impact on Academic Path/Career

Much of my journey in mathematics has been geared towards a career in education. Hence, when embarking on a master's degree, I had originally thought I would continue with this path and most likely teach at the community college level. However, this experience through the NNSA-MSIIP has exposed me to real-world mathematical research, especially in a national lab environment, and cultivated my interest in doing more research. Prior to this internship, I was scheduled to complete comprehensive examinations as my degree culminating experience since I was not able to work with a research/thesis advisor through my campus due to the limited availability. Through this opportunity, I will now be using this project and future research as my master's thesis.

This opportunity has allowed me to grow tremendously in mathematics, and I now see that pursuing a career in research is a great, viable option for me! I hope to continue my work with Sandia National Laboratories as I continue to make progress towards completing my master's thesis. I am also now planning on continuing in my academic journey to complete a Ph.D. While working towards a doctorate will be incredibly demanding, working through this internship experience has shown me that I am ready for the challenges ahead.

6 Relevance to the Mission of Sandia/DOE

Graph partitioning is a widely used approach to load balancing in parallel computing. This aspect of graph partitioning directly ties into work being done at Sandia National Laboratories. Scientists at Sandia frequently perform simulations for applications such as climate modeling, electromagnetics, thermodynamics, and various defense applications. The ability to decompose meshes for large-scale simulations for Sandia application problems will allow for simulations of larger problems on extreme-scale parallel computers. This again gives analysts greater understanding via more in-depth analysis of said simulations. This fact ties into both Sandia and the Department of Energy's, missions of addressing scientific challenges through innovative research and technological solutions.

7 Acknowledgments

Special thanks to my project mentors, Dr. Boman & Dr. Loe; this project would not have been possible without their immense support, guidance, and patience. Additionally, I'd like to thank my graduate school professors from California Polytechnic State University, Pomona - Dr. Von Bremen, Dr. Szypowski, Dr. Rock, and Dr. Caine - for helping to push me professionally and foster my academic abilities. It was through their constant encouragement and guidance that I even decided to apply for this opportunity.

References

[1] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. *Recent Advances in Graph Partitioning*, volume 9220, chapter 4. Springer International Publishing, 11 2016.

[2] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics (SIAM), January 2003.

[3] Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.

[4] Chao Chen, Tianyu Liang, and George Biros. Rchol: Randomized cholesky factorization for solving sdd linear systems. *arXiv:2011.07769v4*, 11 2020.

[5] Seher Acer, Erik G. Boman, Christian A. Glusa, and Sivasankaran Rajamanickam. Sphynx: A parallel multi-gpu graph partitioner for distributed-memory systems. *Parallel Computing*, 106:102769, 2021.

[6] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, September 2005.