Research Statement

Tyler Chen

Broadly, I intend to incorporate parallelism and randomness into existing algorithms for difficult linear algebra problems. Approximating hard problems with a simpler linear problems is a standard approach in the applied sciences. As a result, there is a large demand for algorithms to solve linear algebra problems. Moreover, as computing power increases so does the need for algorithms to which are well suited to take advantage of high performance hardware.

I came to UW due to the strong computing community here as well as the close proximity to tech industry. Over the past year I have I have been building the foundation necessary for my research goals. This has been primarily through a reading with Anne Greenbaum and through my coursework. In the near future I will continue my work with Anne, take relevant courses, and begin reading with Ioana Dumitriu.

Current Research: For the last two quarters I have been working with Anne Greenbaum on understanding the behavior of variants of the conjugate gradient. Conjugate gradient is a standard method for solving Hermitian positive definite linear systems, and is frequently used for problems involving discretizations of physical systems. In exact arithmetic, like most Krylov space methods, the conjugate gradient algorithm will solve a $n \times n$ system at most n steps. However, when implemented in floating point arithmetic, this is no longer the case.

Recently there has been growing interest in parallelizeable algorithms for conjugate gradient. In a single iteration of conjugate gradient, two inner products, and one (often sparse) matrix vector multiplication must be computed. Modern algorithms such as [1, 2] aim to overlap these computations in order to reduce the runtime of a single iteration. While all such variants are equivalent in exact arithmetic, they often behave very differently in floating point arithmetic. Of particular interest are how many iterations are required to reach a given level of accuracy and what level of accuracy can eventually be attained.

When applied to many systems, the parallel variants have significantly worse convergence properties than the standard (Hestenes and Stiefel) implementation. This means that even if a single iteration can be made faster, the total time needed to compute the solution, and the eventual solution itself may not be improved. The problems for which this is the case are not limited to pathological examples, but include systems commonly occurring in engineering and the physical sciences.

It is of obvious interest to understand why different variants behave differently, and our eventual hope is to develop a parallelizeable algorithm with has better numerical properties than current parallel algorithms. At the moment we are approaching the problem from a number of angles. First, we are implementing algorithms in reduced precision arithmetic, to exaggerate the effect of rounding errors, and to attempt to isolate which parts of various algorithms causes instability. Additionally, we are at-

tempting to reframe these problems in a way so that a previous result [3] of Greenbaum can be applied.

There are a variety of practical reasons why understanding how algorithms behave in reduced precision arithmetic is useful. First, it can help highlight the sources of instabilities in algorithms. Second, it is faster to compute in half precision arithmetic. Moreover, GPUs traditionally use half precision arithmetic, so if GPUs are to be used for parallelization, it is important to understand how the algorithms will behave in half precision. To help understand the effect of reduced precision arithmetic I have written a python library which allows for the simulation BLAS functions on matrices of floating point numbers with any number of mantissa and exponent digits. By implementing conjugate gradients on top of this library, we will be able to explore how convergence scales with machine precision.

Another approach to understanding the convergence of these methods In [3], Greenbaum showed that under the right conditions, finite precision Lanczos iteration applied to a given matrix could be viewed as exact Lanczos applied to a larger matrix whose eigenvalues are clustered near those of the original smaller matrix. By writing the terms of the three term recurrence in the Lanczos algorithm using quantities from a conjugate gradient algorithm the bounds from Anne's previous work can be applied.

Coursework in relation to research: In order to help build the foundational knowledge required to work on such problems, I have taken the AMATH 584/585/586, and the AMATH 561/562/563 sequences. In addition I have taken AMATH 567 and MATH 514. In these courses I have explicitly made an effort to improve my programming as well as to improve my intuition as to when rigor is and is not necessary. My background in mathematics meant that I had the tendency to get lost in the details of a problem, rather than focusing on the big picture. Over the past year I have become significantly better at starting with big picture and working out the details as necessary. This has allowed me to improve my intuition and spend time thinking the important parts of a problem.

Over the past few years I have made a point to expose myself to a variety of software libraries including GPU libraries such CUDA and Tensorflow, scientific computing libraries and languages such as C/C++, Numpy/Scipy, and Julia, symbolic manipulation software such as Sympy and Mathematica, and visualization software such as Matplotlib and TikZ. In every project and assignment I make an effort to ensure that my code is efficient and scalable, even if the problems no not require this. These practices now will carry over well when working much larger problems/data in the future.

In addition to broadening my programming skillset, I have attempted to use open source software whenever possible. Academia has historically been an ivory tower, accessible only to those with external funding and wealth. While this has improved, there are still large institutional barriers keeping many people out of academia. It is my belief that promoting open source software can help make academia more accessible to everyone. I have made hundreds of pages of my personal course notes, homework assignments, and code openly available on Github. In particular, I have ported every MATLAB program which has been part of an assignment to an analogous python program so

that future students will not need to pay for MATLAB. I intend to make all of my teaching materials and research output during my time at UW freely available.

Upcoming plans: There are near future plans in place to continue building the groundwork for my future research. First, I will be continuing my work with Anne on conjugate gradient methods. Much of this project aligns very well with my research interests. Anne has organized a small symposium on this topic at the SIAM CSE conference in the spring which I plan to attend. I will also be applying to the NSF GRFP, using this project as the topic of my application. Second, I will be doing a reading with Ioana Dumitriu during the fall quarter. During this reading I will study random linear algebra in order to start to understand how non-determinism can be used in algorithms. Finally, I will continue coursework. I plant to take the rest of the optimization sequence (AMATH 515/516) and relevant MATH courses on linear algebra.

During the last few months I have had to put aside research work in order to prepare for the qualifying exams. I look forward to being able to return my mental focus to these projects in the near future. By winter I will choose an advisor, likely Anne Greenbaum, which will allow me to start building a more detailed timeline for the rest of my PhD.

References

[1] A. T. Chronopoulos and C. W. Gear, *On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy.*, Parallel Computing, 11 (1989), pp. 37–53.

[2] P. Ghysels and W. Vanroose, *Hiding global synchronization latency in the preconditioned conjugate gradient algorithm*, Parallel Computing, (2013).

[3] A. Greenbaum, *Behavior of slightly perturbed lanczos and conjugate-gradient recurrences*, 113 (1989), pp. 7–63.