

Research Statement

Tyler Chen

Broadly, I intend to incorporate parallelism and randomness into existing algorithms for difficult linear algebra problems. Since arriving at UW I have been primarily focused on building the foundation necessary for research in the above mentioned areas. This has been primarily through coursework and a reading with Anne Greenbaum on the conjugate gradient method. In the coming months I will do another reading in order to improve my understanding about how randomness can be used in linear algebra algorithms. I will also continue to take coursework in order to build my computing toolset, and theoretical background.

For the last two quarters I have 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 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 aim to overlap these computations as much as possible in order to reduce the time it takes to compute a single iteration. While all such variants are equivalent in exact arithmetic, they behave 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. It turns out, that when applied to certain (real world) problems, each of these properties may be significantly worse than the standard (Hestenes and Stiefel) implementation. This means that even if a single iteration can be made faster through parallelization, the total time needed to compute the solution, and the eventual solution itself may not be improved.

It is of obvious interest to understand why different variants behave differently, and our eventual hope is to develop a parallelizeable algorithm which is not as susceptible to numerical instabilities as the current algorithms.

At the moment most of our work centers around trying to understand the algorithms in terms of the properties above.

In [1] Greenbaum showed, that under the right conditions, finite precision Lanczos iteration applied to a matrix A could be viewed as exact Lanczos applied to a larger matrix T , whose eigenvalues are clustered near those of A .

Of interest are 1. how orthogonal successive residual vectors are, 2. , and 3. **fill out**.

talk about what can be parallelized

Talk about what skills are required for this type of problem.

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 as 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 do not require this. These practices now will carry over well when working much larger problems/data in the future.

In addition to broadening my tool set, I have attempted to use open source software whenever possible. While not directly related to my research, I believe that access to a quality education and quality educational resources should not be dependent on socioeconomic status or geography, and forcing students to pay for software and textbooks places an unfair and unnecessary burden on students from underprivileged backgrounds.

something about academia for rich

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. It is my hope that these actions will help reduce the reliance on closed source software among future students in our department.

Finally, there are near future plans in place to continue building the groundwork for my research. Most notably, I will be doing a reading with Ioana Dumitriu during the fall quarter.

I also intend to take AMATH 515 in the winter.

During this reading I will study random linear algebra in order to start to understand how

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.

talk about advisor

References

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