

Research Statement

Tyler Chen

Broadly, I intend to incorporate parallelism and randomness into existing algorithms for high-dimensional linear algebra problems.

Up to this point my primary focus has been on foundational coursework. To this end, I have taken the AMATH 584/585/586, and the AMATH 561/562/563 sequences. In addition I have taken

What have done

Coursework

Provide groundwork for linear algebra

Provide groundwork for stochastics

Improved programming skills and tools

GPU computing

Become familiar with CUDA and packages such as tensorflow

GPU uses half precision

Need to understand the effects of this

Use and develop open source software

Talk about actively choosing python/numpy over matlab despite large department momentum towards matlab

Convert all given code and upload for future students

Reading with anne

Hands on experience with modern CG algorithms

Start to build understanding of underlying concepts

Field of values

Spectrum

Lanczos

Results of varying precision

Important to understand this because 1. Low precision is faster 2. GPUs use low precision

I have been working with Anne Greenbaum on conjugate gradient variants for the last two quarters. 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 con-

jugate gradient algorithm will solve a $n \times n$ system at most n steps. However, when implemented in floating point arithmetic, some of the properties which allowed for the algorithm to terminate are no longer satisfied. Of interest are 1. how orthogonal successive residual vectors are, 2. , and 3. **fill out**.

Under certain conditions about these properties, the convergence of **WHAT EX-ACTLY** has been characterized by Grenbaum. **CITE**

While the standard implementation of conjugate gradient is relatively well understood, recently there has been interest in parallelizing the algorithm

talk about what can be parallelized

While all such variants are equivalent in exact arithmetic, they behave differently in floating point arithmetic. In particular, the rate of convergence (in terms of the iteration count) and the eventual attained accuracy may be significantly worse than HS conjugate gradient on certain problems. If more iterations are needed to reach a given level of accuracy, this means that even if a single iteration can be made faster, the total time needed to compute the solution may not be improved.

It is of obvious interest to understand why different variance behave differently. Our hope is to be able to develop a parallelize able 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.

What I will do in the near future

Set up reading with ioana

Understand intersection of probability theory and linear algebra

Apply to NSF

Get extra funding

More collaboration