

Faster Matrices

Aidan J. Nyquist
CERN-HSF, Google Summer of Code
(Dated: March 2018)

Proposal for "Faster matrix algebra for ATLAS" Google Summer of Code, CERN-HSF project.
The purpose of this project is to increase the performance of ATLAS by optimizing its symmetric matrix operations. This will decrease computing and storage demands, and increase speed.

INTRODUCTION

The Atlas Project uses Eigen extensively. However, Eigen handles symmetric matrices inefficiently. Since a large portion of Atlas' algorithms are made up of symmetric matrices, this project is able to optimize a large niche in code. By implementing a symmetric matrix class, the storage space, and the complexity of certain algorithms like addition of a dense square matrix ($n \times n$) originally growing at n^2 becomes $\frac{n(n+1)}{2}$. This approaches a 50% complexity and memory decrease for larger and larger matrices. Of course, not all algorithms become as efficient. For example symmetric matrices are not generally closed under matrix multiplication, so it would be unreasonable to expect a dramatic speed increase. However, the memory decrease alone should be more than enough to justify a symmetric matrix class. The algorithm efficiency is just the frosting on the cake.

PROJECT GOALS

"A working implementation of symmetric matrices in Eigen, ready to be submitted as a patch for Eigen." - Proposal Page

High Priority

- Symmetric Matrix Class
- Basic matrix operations
- Seamless integration with Eigen Matrices ¹
- Use Eigen conventions and style in code
- Decrease in memory cost
- Tests
- Well documented

¹ There are *a lot* of operations Eigen can perform on matrices. By integrating with Eigen there will be no need to rewrite any algorithms that won't have a direct speedup from a Symmetric Matrix Class.

Medium Priority

- Increase in performance
 - Benchmarks to show improvement
- Eigenvalue algorithms
- Reusable code

Low Priority

- Various matrix decompositions
- Complex symmetric matrices
- Skew-symmetric matrices
- Sparse symmetric matrices

TIMELINE

The following is a rough outline. There is still some research and dialogue to be done before and during the summer. Many tasks might bleed back and forth through the timeline as the project progresses and new insights are found whilst the project progresses.

Pre-Summer

The best time to get a head start in the project is right now. One of the most important things I could do is to continue interacting with Eigen library and community. That way, I'll have practiced swimming before I have dive into Eigen.

I also will reboot my blog. This will give me a way of introducing myself and my project. I will also use it as communication platform for the rest of the summer and beyond. Meaning at minimum, there will be a blog post for every section outlined in this timeline.

There are many ways of extending Eigen, picking the right one first would be more than ideal. To make sure I pick the right one first, I will study current Eigen Matrix classes, prototype different implementations of classes and methods, and engage with the community and my mentors.

First Third of Summer

- Finish prototyping integration with Eigen
- Extend best prototype
- Implement most optimal data structure
- Implement as many functions as possible
 - Accessors
 - Arithmetic (+, +=, -, ·)
 - Comparison (==, !=)

Second Third of Summer

- Continue implementing functions ²
 - Inverse
 - REF / RREF
 - Eigenvalues & Eigenvectors
 - * Research
 - * QR decomposition
 - * Bidiagonalization
- Organize tests and write benchmarks
- Refactor any poor code
- If time allows, begin a low priority task or two

Last Third of Summer

- If time allows, finish a low priority task or two
- Finish any lingering tasks
- Prepare and submit patch
 - If the patch won't get merged, I will republish the code as a 3rd party plugin.

Post-Summer

If the patch does not initially get accepted, or the project runs into roadblocks, I plan to see the project through—whatever continuation of dialogue or programming that might involve.

Bugs and questions regarding my code are things I plan to watch for a few years. I also hope to be able to answer any questions that might arise on as many communication platforms as I can watch. Any bugs that reveal themselves later in my code, I also plan to fix.

² Some of these functions will need to be reviewed before being implemented, as there may be no advantage rewriting them.

RELATED WORK

Current support for symmetric matrices in Eigen are in the form of views. While these views support optimized algorithms, they do not optimize memory in any way. Studying the code for these views will allow insight on how to write optimal matrix algorithms. They might also be able to be exploited so my class can use their code. A few days might be used to consider/implement that possibility.

There are other popular libraries that have somewhat implemented symmetric matrices: the GNU Scientific Library, and Boost uBLAS. Both of these could have code that would prove useful to study. Time might be taken during the summer to study this code.

There is lots of academic work related to symmetric matrices. While I plan to find all relevant information before the summer, some time might be spent during the summer reviewing these resources.

BIOGRAPHICAL INFORMATION

I currently pursuing a B.S. in Mathematics and a B.A. in Physics. This project sounds more interesting than anything I could dream to do this summer. I have a strong passion for programming, mathematics, and physics. This project is a beautiful mixture of indirectly contributing to the physics community, by directly contributing to the programming community using my mathematics abilities. I've taken a couple of couple of linear algebra courses, and have used my programming skills, for recreation, open-source contribution, and paid work for a few years now. I have prototyped some code for this project already: github.com/ajjony/symmat/ I feel very qualified to take up this project, not only because of my background and hard-skills, but as well as because of my relentless passion and drive for projects like these.

CONTACT INFORMATION

- Email: contact@aidannyquist.com
- University Email: ajnyquist@northpark.edu
- Website (WIP): aidannyquist.com
- IRC/GitHub: [ajjony](#)
- Phone: 971-237-5505
- Address: 3580 NW Hill Rd, McMinnville, OR 97128