Research Statement

Abhijit Chowdhary

February 1, 2019

1 Introduction

I'm largely interested in Computational Mathematics, specificially regarding the algorithms and analysis of them. Therefore, some fields that I find interesting are numerical linear algebra, computational group theory, fast PDE/ODE solvers, etc. In addition, I have a substantial interest in the methods used to accomplish such tasks, such as compilers and high performance computing. Most have my study has gone to the pursuit of such topics, and most of my projects are directly related.

I've had a breadth type of study, and have hit most of the fields that can be modeled by a computer. Ideally, I would like to work on projects similar to FLINT [1], LinBox [2], or SageMath [3], but am also very interested in the related pure fields, though I will likely take a computational bent to the project.

2 Projects

To give you a better idea of what I've done so far, here's a few applications.

2.1 Algebraic Point Set Surfaces

During the Spring 2018 semester, my sophomore year, I took this course called Geometric Modeling. This was a masters/PhD level course, intended for those interested in computer graphics research or cutting edge level industry work, and I attended on reccomendation from a professor who thought the coursework was wild. While the whole class was a beautiful exercise in Numerical Optimization and Numerical Linear Algebra, my term project is currently a main motivation for me to pursue research in this field.

My final project was to implement the paper *Algebraic Point Set Surfaces* by Gunnebaud and Gross. Roughly speaking, in geometric modeling we have the pipeline: Physical object $\rightarrow$ spatial point cloud representing external surface $\rightarrow$ triangularized mesh representing object. The paper focused on a fast method to take the point cloud to a triangularized mesh. The prevailing method was to consider an open neighborhood of a point in the cloud, and fit a

[1] Fast Library for Number Theory: http://www.flintlib.org/
[2] Exact computational linear algebra: http://www.linalg.org/
[3] Symbolic Everything: http://www.sagemath.org/

plane to it that minimized the distance to the points in that neighborhood. Such a plane would become part of the later mesh. The paper suggested fitting a sphere to these points instead, which was hugely advantageous in regions of high curvature. Such a fit would result in a non-linear optimization problem, but the key idea of using the algebraic definition of a sphere, results in a nicer computation. Specifically, given normals we can have a linear system solve, and we can estimate them with a generalized eigenproblem.

The truly difficult parts of this project to me wasn't the actual graphics theory. What really plagued me was the implementation; I ran into serious runtime issues in the actual construction of the matrices involved in the eigenproblem, and later the linear system solve. I spent a while making various vectorization arguments to speed up the construction, and then later making sparsity arguments in the actual system solve.

Despite the difficulty of my first research level project, I truly enjoyed the construction process, nothing to say of the relief when I finally managed to triangularize a bunny [4]. In addition, it verified to me that I not only enjoyed the study of numerical methods, but liked and was capable of research and development in them.

2.2 FFPoly

This is a new project I've begun, and have taken into my semester as a project. In my graduate algebra course I was exposed to some algebraic number theory, and upon request, my professor showed me a few papers regarding the computational aspects of it. Intrigued, I decided to write a module in C++ implementing a element of the field $\mathbb{F}_p[x]$, for small $p \in \mathbb{Z}$.

This polynomial field is interesting, since the splitting behavior of minimal polynomials of number fields gives information of the splitting behavior of the corresponding primes in the algebraic number field. Furthermore, using Minkowski's bound, we could said information to verify the class groups of certain small quadratic number fields, which is interesting. SageMath has such a capability, but I had a couple of special ideas that I wanted to try; specifically I wanted to investigate the performance of parallelization on these problems. I'm taking a PhD course in High Performance Computing now, and plan on using this project as my capstone research piece. There is lots of potential in the field operations and factoring steps, and some interesting active research going on in the field [5].

This project is in it's infantile stages, and I've just barely began learning OpenMP, but since parallel computation is a future step in numerical methods, I have high hopes for its results, and hopefully can beat out similar serial libraries in speed.

[4] A classic point cloud to construct is the Stanford Bunny model.

[5] See the paper *Parallel Integer Polynomail Multiplication* by Chen, Covanov, Mansourim Maza, Xie and Xie