Analysis of Code Optimization Using Voltage Maps

Jeremiah Simmons

December 2, 2016

1 Abstract

Voltage maps visually represent the drop of voltage over distance, and can be used to explore the linear relationship of an electric field. Modeling an electric field is done by transforming physical space into a matrix allows these systems to be solved via Gaussian elimination. These system of equations become increasingly repetitive when solved by hand, therefore calling for computational methods. While computational methods are a useful resource, scientists have found many ways to optimize their code. Lapack, for instance, has been developed and optimized for years. The result is a very fast, efficient library of functions for solving linear algebra in code. In this paper we will see how different programs for solving the same scenarios can perform very differently. Lapack solves the system in 1.81 seconds, compared to our program that clocks in at 45.54 seconds to complete.

2 Introduction

In today's society almost everything interacts with the electrical grid. Power lines and wiring surrounds our neighborhoods and cities; power lines carrying enormous amounts of electricity. In order to keep the public safe these power lines require a barrier or layer insulation in order to ensure the current flows safely to power our electronics. But, if a layer of artificial insulation was added to the wiring all the power lines would collapse from the great weight. Therefore, some else is needed. Luckily, air, already installed, is both lightweight and cheap; air is an incredible insulator for electricity. This is explained by looking at the electrical field. From Coulomb's Law we can use point charges to quantify the strength of the field at a specific distant from the source. This is shown by the equation:

$E = DV/DL$

where E is the strength a point charge in the field and dv is the infinitesimally small change in voltage, where voltage is potential difference, and DL is a small change displacement.

Electric field extends outward in all directions away from a source of current. Electricity flows naturally from one node to another node of opposite charge. Positive flows naturally to negative nodes, and vice versa. When discussing a power line, the ground is, well, a ground and is neutrally charged. As electric field is observed farther away from a source, the field lines are attracted to the ground. As shown by the following diagram:

insert electric charge

3 Methods and Modeling

Construction of the mathematical space came from reducing the dimensional space our physical world, which is 3-dimensional space, to 2-dimensional. Allowing our wire to be infinitely long and at the same height from the ground, we can observe the physical world in slices of equal characteristics. Therefore we can create a 2-d array to represent the slice. Additionally we know that the ground is neutral, and at infinite distance away from the wire the electric field approaches zero. So therefore we can "pad" the array with zeros.

Using properties of electricity and magnetism, a system of linear equations was built. From here we created our own non-proprietarily owned linear algebra solver using Gaussian elimination to achieve a reduced row echelon form (rref). In the other program, for comparison of optimization, netlib's lapack, a linear algebra library, was also used to solve the same system.

4 Results

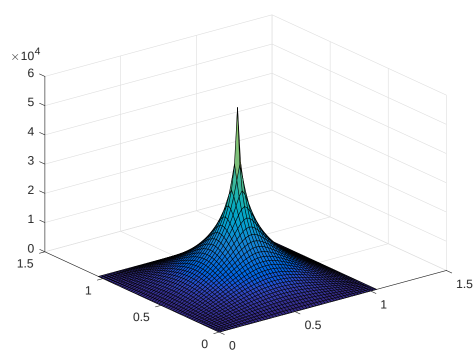

Figure 1:
Voltage map built from the our computational method.

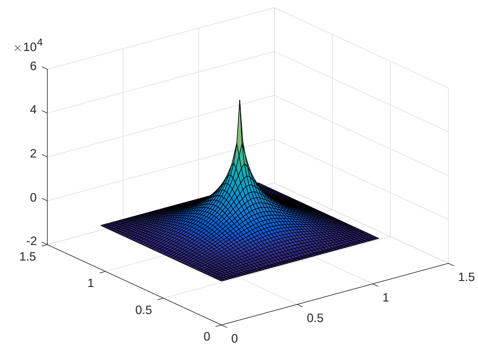

Figure 2:
Voltage map built using Lapack library.

As shown by Figure 1 and Figure 2, the results of both methods are identical. The largest difference between the two methods is optimization. Lapack has been constructed, rebuilt, and develop over many years. The execution between method one and method two is forty-four seconds. Lapack executes, for a fifty-one by fifty-one physical space , in 1.81 seconds. Our method executes, for the same size space, in 45.54 seconds.

5 Conclusion

We achieved two similar models using different computational methods and saw the how well code can be optimized. Our method was significantly slower than when using the Lapack library. While the same results were achieved, optimized code leads to faster results and ultimately allows scientists to make new discoveries much more quickly.

References

[1] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, D. Sorensen *LAPACK: a portable linear algebra library for high-performance computers* 1990. University of Tennessee, New York University, NAG Ltd., University of California, Berkeley, Argonne National Laboratory, Rice University.

[2] Justin Oelgeotz, Dr. *Physics 4000 - Computational Lecture* 2016. Austin Peay State University, Department of Physics and Astronomy.

[3] n.a. *Electric Field* 29 November 2016. Wikimedia Foundation, Inc.