Computational Study of the N-Hedron

Andrew McNutt amcnutt@reed.edu Reed College, Portland Oregon

Abstract

In this project, we employ several computational methods to approximate the locations of the collection of N points on a sphere that defines the three-dimensional platonic solid called the N-hedron. In doing so, we generate an approximate set of solutions to the famously unsolved problem of even distribution on a 2-sphere.

Introduction

The goal of this project is to approximate the shape known as N-hedron. This shape is the collection of polygons which can be described by arranging an integer N number of points on a sphere and requiring them to be maximally far apart. This has a number of interesting applications in computer graphics, geometry, and chemistry. In chemistry, this problem is formulated by considering an integer number of electrons and allowing them to find a state of minimum energy. These states of minimum energy tend to represent electronic configurations for different sorts of atoms. This formulation is known as the Thomson problem.

We will focus on a similar formulation of the problem in which we attempt to maximize the sum of the distance between each of the points in the shape. This is done by maximizing the function:

$$F = \sum_{j=1}^{N}\sum_{k=1}^{N} ||x_i - x_j|| \qquad (1)$$

where x_i is the vector representing the position of each of the points. This is equivalent to the Thomson formulation, which uses the inverse of the norm. One of the large challenges associated with this problem is ensuring that the points remain on the sphere. We solve this by persistently mapping back and forth between Cartesian three-space (x,y,z) and angular two-space (ϕ, θ). While this method does build a certain level of error into the computation because we have to continually make calls to the transcendental *sin* and *cos* functions, it does so at such a negligible level that it is not noticeable with our measurement schemes.

We will follow the three traditional measurements for the quality of optimization: Robustness, Timing, and Accuracy.

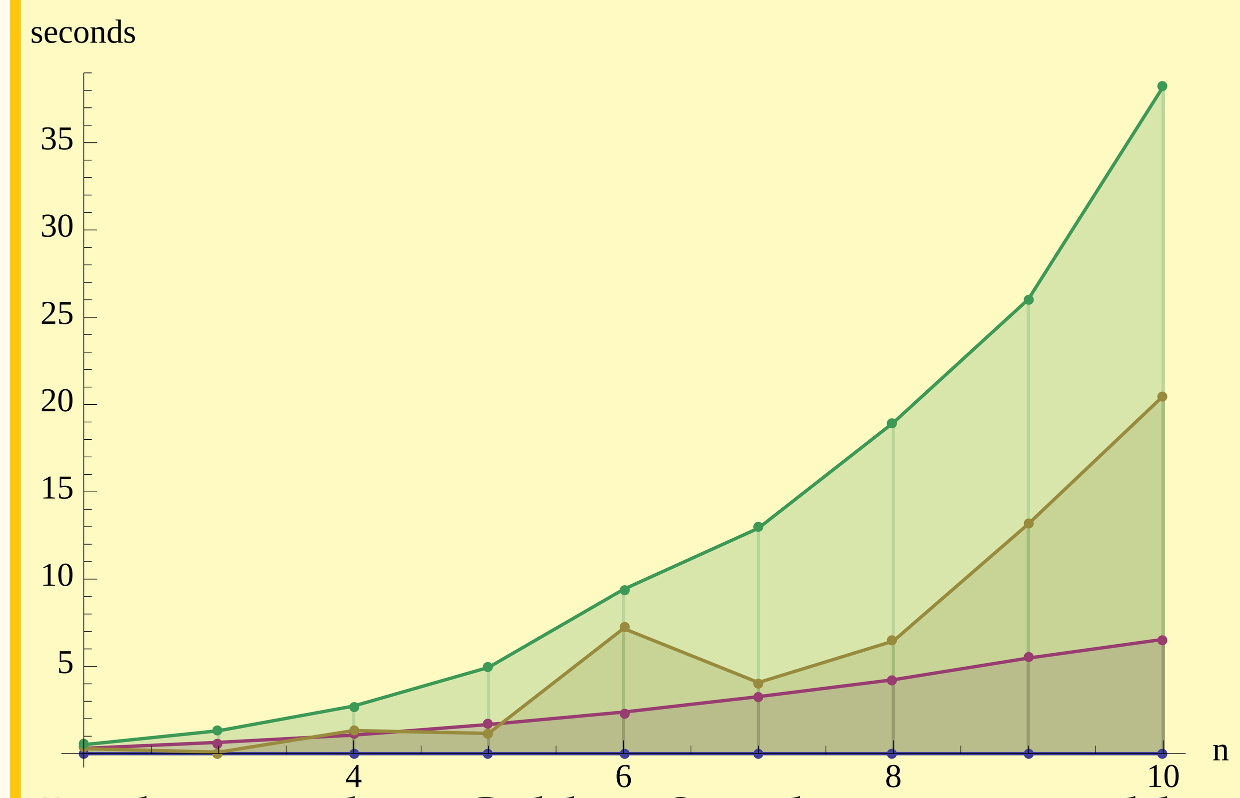
Methods

We use three different computational methods to gain traction on this problem:

1. **Golden Spiral** This method creates a logarithmic spiral, divides it evenly for an integer N, and wraps it around the sphere. This is a standard algorithm in computer graphics for creating approximations of the sphere.
2. **Monte Carlo** From an arbitrary start position, this method randomly generates perturbations about that point and then checks to see if (1) has increased. If so, it moves to that position, and if not, it tries again. In our construction it repeats this a fixed number of times.
3. **Steepest Descent** From a well-chosen start position (in our case, the Golden Spiral) this method iteratively computes the numerical derivative of (1), and then moves the position along the path of the highest derivative. We implement it both for adaptive accuracy, and for a fixed number of iterations.

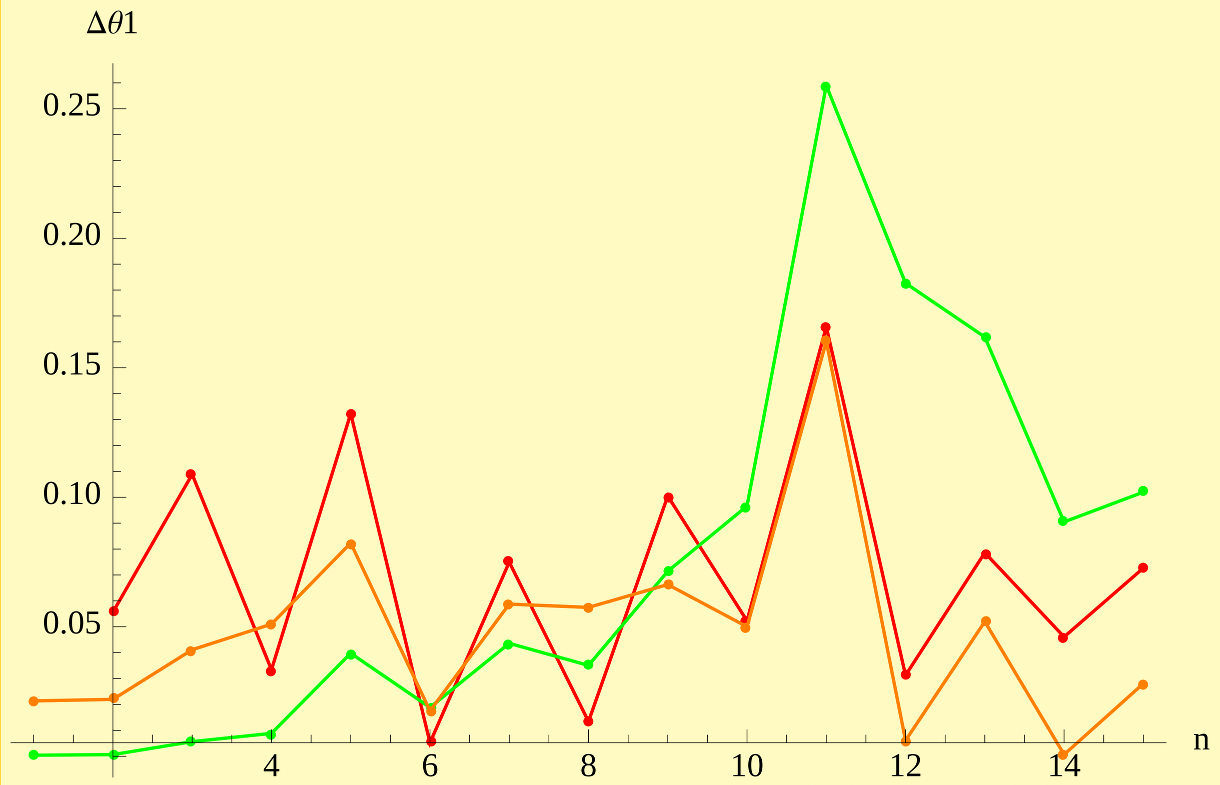
Timing

We wish to determine which is the fastest of our methods.



In this graphic, Golden Spiral appears as blue, Monte Carlo as red, Adaptive Step Steepest Descent in orange, and Fixed Step Number Steepest Descent in green.

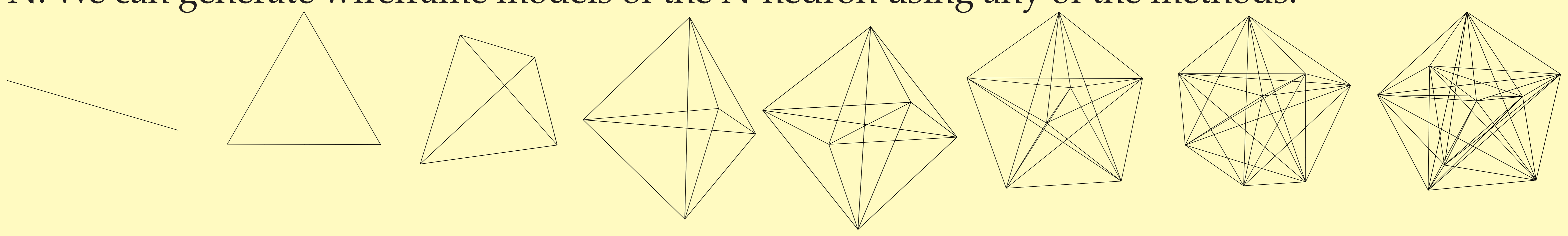
Accuracy



In a common parameterization of this problem, we consider the smallest angle involved in the polygon. In the above graphic, we have found the smallest angle constructed by each of our methods and computed the residual with the known smallest angle (based on a list from Wikipedia). Red is Golden Spiral, green is Monte Carlo, and orange is Steepest Descent with Adaptive Stepping.

Results & Robustness

We found that, in general, with these methods we were able to successfully construct the N-hedron for any N. We can generate wireframe models of the N-hedron using any of the methods:



N=2 N=3 N=4 N=5 N=6 N=7 N=8 N=9

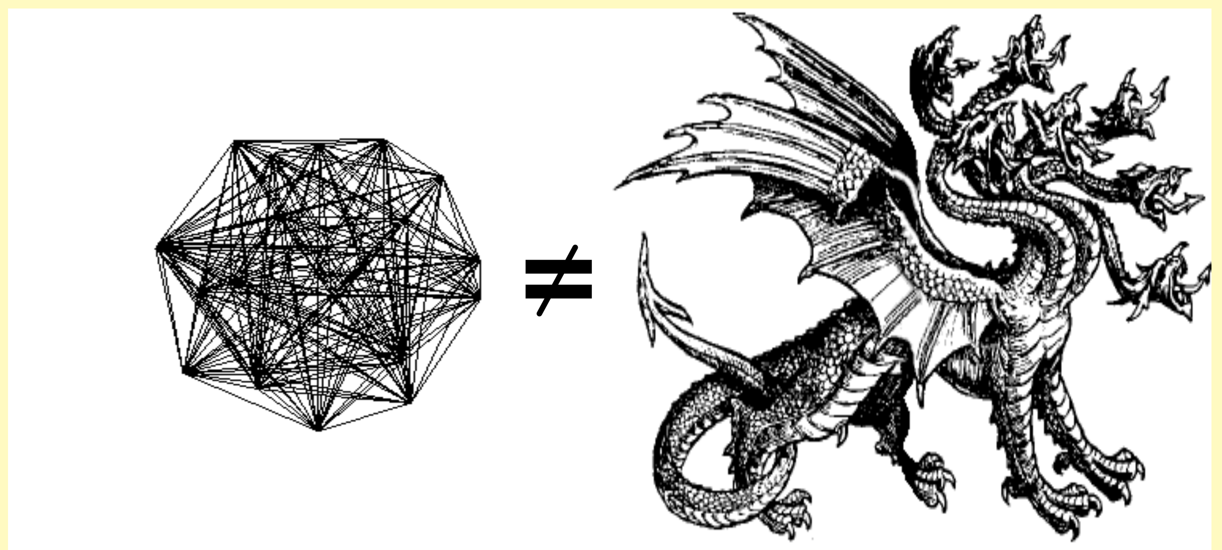
These particular approximations of the true shape were generated using Monte Carlo at 3000 steps. In traditional algorithmic analysis, it is good to check that the method is Robust (i.e., in optimization, whether or not always it converges to a minima). Golden Spiral gives a good approximation but never actually converges to anything. Monte Carlo is not Robust simply because it cannot be proven that it converges to the answer. Fortunately it has been analytically proven that Steepest Descent will converge.

Conclusions

Fascinatingly, we have three types of measurement, and one of our three methods is the most successful in each of them. Golden Spiral wins by a long margin in Timing. Steepest Descent wins in Robustness. There is a somewhat subtle point that Monte Carlo does in fact win Accuracy. In the Accuracy section we see that for low N, MC clearly does the best, but then falls off. This is actually showing that for higher step sizes an increased number of steps are required.

Whimsy

If one considers the N-hedron as a model for the N-Headed-Hydra we can approximate this beast as simply a piece of terrifying geometry. However we can clearly see that this a terrible approximation:



References

[1] J. Nocedal, S. Wright, Numerical Optimization Spring, 1999

[2] Wikipedia contributors "Thomson problem." Wikipedia, The Free Encyclopedia.