Project 1: AIM

Jack Hamel

March 8, 2018

1 Code Overview

My AIM code, written in Python, is divided into three segments based on the functionality of each piece. The `grid_mapping.py` file contains all of the functions used to map sources to the grid and off the grid as well as computing the near field potentials on the grid and their corresponding direct potentials. Most of the variables in this file correspond to the same variables in Bleszynski's AIM algorithm paper.

The `proj1.py` file computes the potentials between N points via AIM and directly. It tracks the time elapsed during both methods as well as the average error between the directly calculated potentials and the AIM potentials. The functions here are well named to describe their functionality, besides `run`. The `run` function expects the number of particles N, the dimensions of the 2D space and the step size between grid points as arguments. When called, this will compute the potentials between the sources using AIM and direct calculations and return the time to run each calculation as well as the error.

In the `aim_test.py` file, the `run` function is called here with various input specifications to investigate how the error behaves and the timings of AIM versus direct calculations.

2 Cost Analysis

Figure 1 shows the time elapsed to compute the potentials using AIM and direct calculations across the range of $N = \{250, 300, ..., 2900, 2950\}$. The break-even point lies between $N = 1400$ and $N = 1800$ with $D_x D_y \approx \frac{N}{4}$, where D_x and D_y are the dimensions of the 2D space in the x and y directions, respectively. What is not shown in the figure is the effect of particle density on the break-even point. With a density of half or less than prescribed for Figure 1, the break-even point is difficult to obtain on my machine. The step size between grid points was 0.05 for all runs, which resulted in an average error from all runs of 3.3791×10^{-4}.

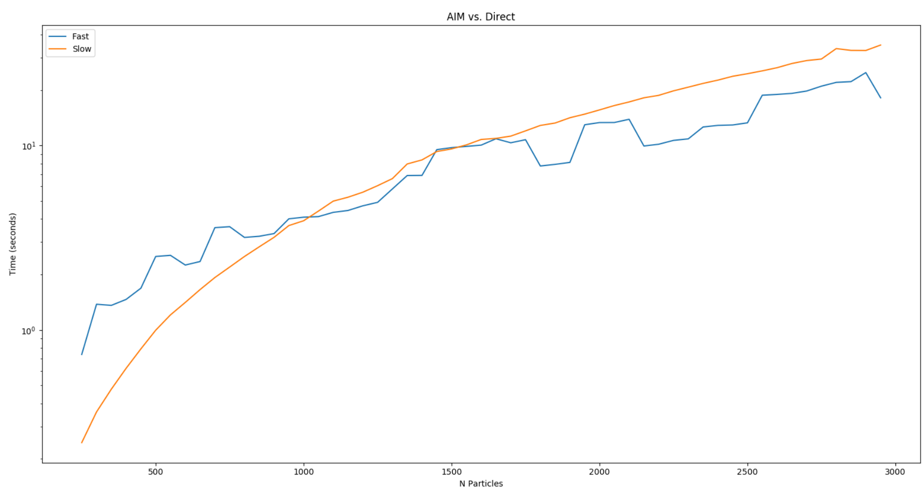

Figure 1: Cost of Calculating Potentials with AIM

3 Error Control

To investigate the effects of the grid spacing on error control, the `run` function was called with varying step sizes. The grid size was held constant while varying the step size between grid points; resulting in more grid points for low step sizes and fewer grid points for high step sizes. In Figure 2, the average error is plotted for $step = \{0.005, 0.01, ..., 0.445, 0.45\}$. For each run, $N = 100$ and $D_x = D_y = 4$. In Figure 3, the effect of source density on error is shown. $N = 100$ and $step = 0.05$ for each run, while $D_x = D_y = \{1, 1.25, ..., 14.75, 15\}$, thus decreasing the density of the source particles as the dimensions of the grid increase.

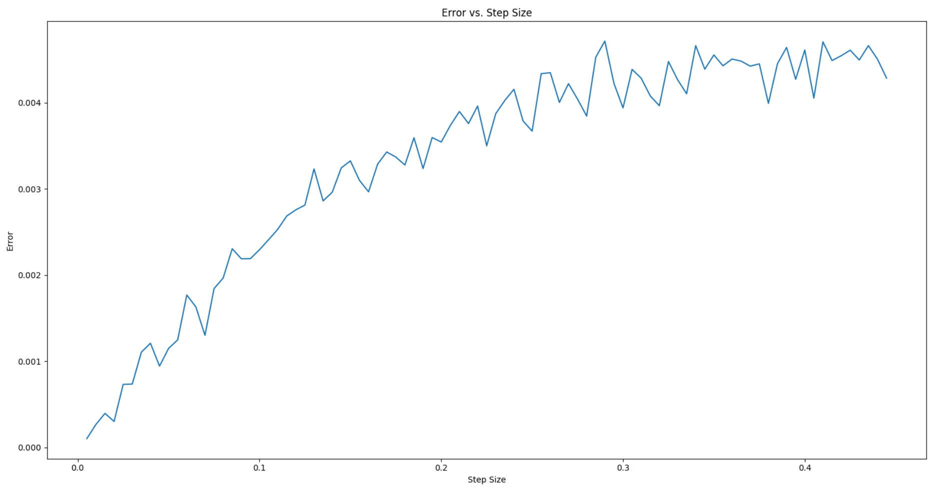

Figure 2: AIM Error Control using Grid Spacing

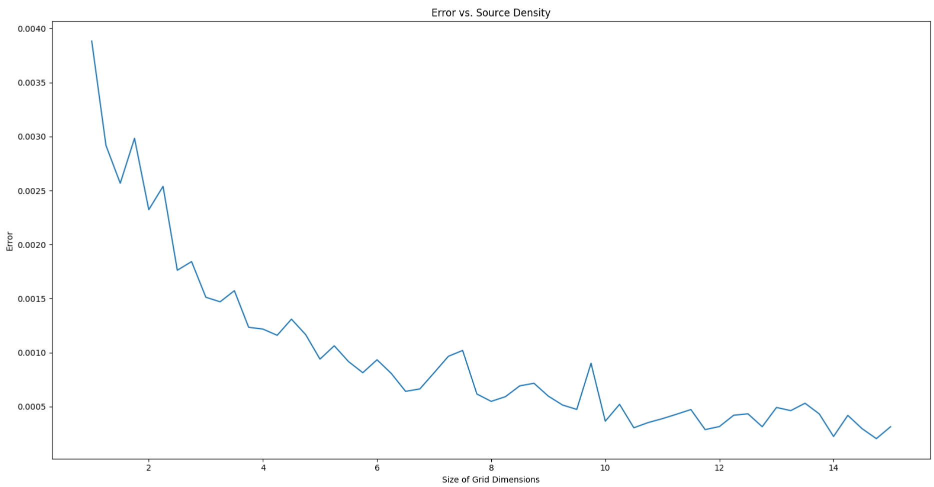

Figure 3: AIM Error Control using Source Density