Homework 5: Linear Algebra

Caden Gobat

March 26, 2021

1 Introduction

In this assignment, we employ the linear algebra technique of matrix-based Gaussian elimination to solve very large systems of equations that would be wildly unrealistic to undertake by hand.

2 Results

Question 1

The code-based numerical solution for the equivalent resistance between $(0,0)$ and $(1,1)$ on a 2×2 grid is 0.5 Ω, if we take each individual resistor to have a value of 1 Ω. To determine an analytical solution to this simple case, we recall the rules for summing resistors in series and in parallel:

$$R_\mathrm{s} = \sum_n R_n$$
$$\frac{1}{R_\mathrm{p}} = \sum_n \frac{1}{R_n}$$

Where R_n is the value of each individual resistor. (Here, $R_n = 1\ \Omega\ \forall\ n$.) Because the resistors at the end of each column and row wrap back to the beginning, this grid simplifies to a circuit with two branches which are each made up of two pairs of parallel resistors in series with one another. Thus,

$$R_\mathrm{eq} = \frac{1}{2\left(\frac{1}{1+1} + \frac{1}{1+1}\right)} = \frac{1}{2}\ \Omega$$

This agrees with the code's result.

Question 2

The equivalent resistance between the points $(1,1)$ and $(3,2)$ on a 5×5 grid is 0.68 Ω. We get the exact same result for the resistance between the points $(0,0)$ and $(2,1)$ again due to the fact that the "endpoint" resistors wrap around and are thus not really endpoints at all, meaning that absolute positions are arbitrary—relative separations are all that matters in these problems.

Question 3

N	R_eq
10	0.748564 Ω
20	0.767009 Ω
30	0.770466 Ω
40	0.771678 Ω

Table 1: Equivalent resistance between the nodes $(0,0)$ and $(1,2)$ on an $N \times N$ grid.

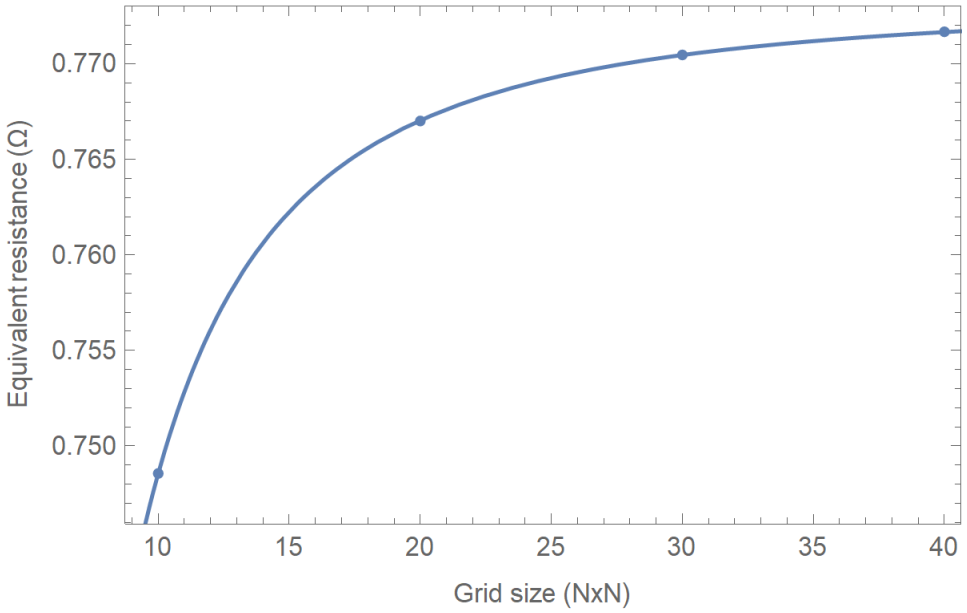

Figure 1: Equivalent resistances as a function of N using a polynomial of the form $R_{\text{eq}} = a + b/N^2$. The optimized parameters are $a = 0.7732$ and $b = -2.4641$. This model tells us that an infinite grid should have a resistance of 0.7732 Ω between these two points.

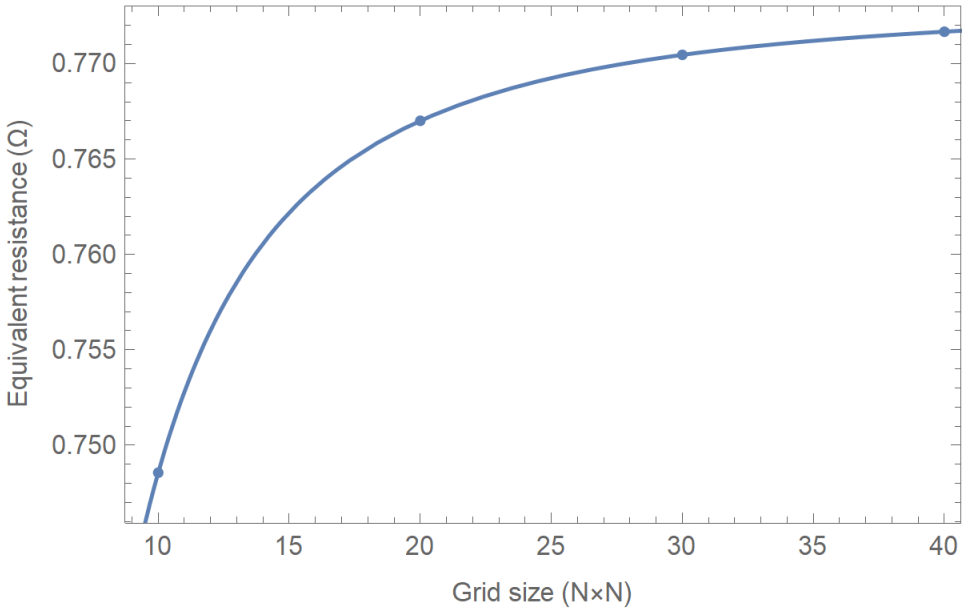

Figure 2: Equivalent resistances as a function of N using a polynomial of the form $R_{\text{eq}} = a + b/N^2 + c/N^4$. The optimized parameters are $a = 0.7732$, $b = -2.5005$, and $c = 3.2933$. This model tells us that an infinite grid should again have a resistance of 0.7732 Ω between these two points.

The fits are both visually very close, and result in very similar results for $\lim_{N\to\infty} R_{\text{eq}}$. The second (fourth-order) function nominally allows for a closer fit, although theis raises the question as to whether this extra precision is warranted or whether it is an example of overfitting, given the already good fit granted by a function of order N^{-2}.

Question 4

N	CPU time
10	0.030 s
20	1.608 s
30	18.108 s
40	127.640 s

Table 2: Computational times to determine the equivalent resistance between the nodes $(0, 0)$ and $(1, 2)$ with an $N \times N$ grid for various values of N.

The logarithmic slope (i.e., index of the power law) between the minimum and maximum points is 6.027. This tells us that the computational time scales as approximately $\mathcal{O}(N^6)$. A plot of the timing points and the fit through them using a function of the form aN^6 is shown in Fig. 3. The best-fit value of a here is approximately $3.1 \cdot 10^{-8}$.

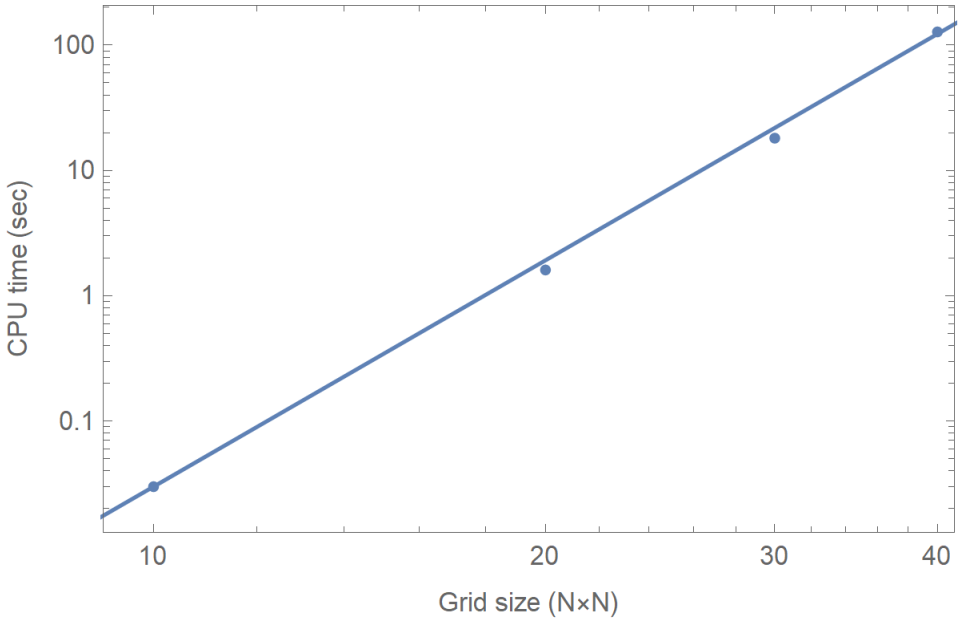

Figure 3: CPU time used to calculate the equivalent resistances between the given points on a grid of dimension N. This relationship shows close adherence to a power law, as evidenced by the fact that it looks linear in log-space.

3 Conclusions

The main challenge for me in this assignment was wrapping my head around the enumeration system for indexing the resistors and vertices on the grid. However, I eventually realized that because of the wrap-around nature of the setup, as long as every resistor and pair of nodes is accounted for, it actually does not matter how they are ordered in the backend of the code. Once the matrix is constructed, the Gaussian elimination algorithm does not care about the order of the equations.

With this conceptual barrier out of the way, I was able to fairly easily execute the rest of the tasks to solve the problems presented here, and having to code our own linear algebra routines certainly gave me a greater appreciation for the field as a whole.