Student Name: David Wang
Student ID: 99030033

PHYS 410: Computational Physics - Project 1 Fall 2022

---

1 Introduction

In this project, we seek to analyze the equilibrium distributions of N charges on the surface of a sphere. Using Newtons laws and constraining the motion of the particles to the surface of a sphere, finite differencing approximation is used to perform numerical calculations for the motions of each particle over time.

2 Problem Description

The force acting on each individual particle in a collection of N particles can be described by

$$m_i \cdot \mathbf{a}_i = -k_e \sum_{j=1, j\neq i}^{N} \frac{\mathbf{r}_{ij}}{{r_{ij}}^3} - \gamma \mathbf{v}_i$$

Without loss of generality, the masses and charges can be set to zero, and acceleration and velocity can be written into differential form to give the equation of motion for each particle:

$$\frac{d^2\mathbf{r}_i}{dt^2} = -\sum_{j=1, j\neq i}^{N} \frac{\mathbf{r}_{ij}}{{r_{ij}}^3} - \gamma \frac{d\mathbf{r}_i}{dt} \tag{1}$$

The voltage of the configuration can be determined by

$$V(t) = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \frac{1}{r_{ij}} \tag{2}$$

where $r_{ij} = |\mathbf{r}_j - \mathbf{r}_i|$ is the magnitude of the displacement vector from particle i to particle j. With these equations, we can numerically simulate the spatial propagation of the charges through time, by solving the equations using Finite Difference Approximations (FDAs). Specifically, $O(\Delta t^2)$ accurate approximations of the first and second derivative give

$$\left.\frac{d^2\mathbf{r}_i}{dt^2}\right|_{t=t^n} = \frac{\mathbf{r}_i^{n+1} - 2\mathbf{r}_i^n + \mathbf{r}_i^{n+1}}{\Delta t^2} \tag{3}$$

and

$$\left.\frac{d\mathbf{r}_i}{dt}\right|_{t=t^n} = \frac{\mathbf{r}_i^{n+1} - \mathbf{r}_i^{n-1}}{2\Delta t} \tag{4}$$

3 Numerical solution of problem

Solving for the equations of motion (1) by substituting FDA equations (2) and (3) for the first and second derivatives yield

$$\frac{\mathbf{r}_i^{n+1} - 2\mathbf{r}_i^n + \mathbf{r}_i^{n+1}}{\Delta t^2} = -\sum_{j=1, j\neq i}^{N} \frac{\mathbf{r}_{ij}}{{r_{ij}}^3} - \gamma \frac{\mathbf{r}_i^{n+1} - \mathbf{r}_i^{n-1}}{2\Delta t}$$

Solving this equation for $\mathbf{r}_i^{n+1}$ gives

$$\mathbf{r}_i^{n+1} = \frac{-1}{\gamma + \frac{2}{\Delta t}} \left[\sum_{j=1, j\neq i}^{N} \frac{\mathbf{r}_{ij}}{{r_{ij}}^3} + (\frac{2}{\Delta t} - \gamma)\mathbf{r}_i^{n-1} - \frac{4}{\Delta t}\mathbf{r}_i^n \right] \tag{5}$$

Thus, from (5) all positions of charges for time-steps greater than $n+1$ can be solved numerically given that we know the positions at time steps n and n-1. We solve the problem using a finite difference grid of $n_t = 2^\ell + 1$ timesteps, with Δt defined as $\frac{t_{max}}{n_t - 1}$.

3.1 Helper functions

In order to normalize the position vectors during charge calculation, the script normalize.m was implemented:

```
function rnorm = normalize(r)
    rnorm = r./sqrt(sum(r.^2,2));
end
```

which normalizes a n x d matrix of n d-dimensional vectors where each vector is normalized according to

$$\mathbf{r}_i \rightarrow \frac{\mathbf{r}_i}{|\mathbf{r}_i|}$$

3.2 FDA updates in code

Equation (5) was implemented in charges.m to update a (nc, 3, nt) matrix of x,y,z coordinates for nc charges across nt timesteps. Utilizing a 'for' loop to make each calculation of the summation term significantly slowed down the computation, and a more efficient vectorized method was developed as below.

```
for time_step = 2:nt-1
    for charge_number = 1:nc

        % take matrix differences r_ij and find sum of all r_ijs
        all_rijs = r([1:charge_number-1,charge_number+1:end],:,time_step
   )-r(charge_number,:, time_step);
        magnitudes = sqrt(sum(all_rijs.^2,2));
        sum_rijs = sum(all_rijs ./ magnitudes.^3);

        % terms in FDA
        summation_term = sum_rijs * 2 * dt;
        r_nminus1_term = (2/dt - gamma) * r(charge_number, :, time_step
   -1);
        r_n_term = (-4/dt)*r(charge_number, :, time_step);

        % update the r at time step n+1
```

```
        r(charge_number, :, time_step+1) = 1/(-gamma - 2/dt) * (
   summation_term + r_nminus1_term + r_n_term);
    end
    % normalize all r's
    r(:, :, time_step+1) = normalize(r(:, :, time_step+1));
end
```

3.3 Voltage calculations

After the entire grid of r values are calculated for each time step, V(t) is calculated for the entire distribution at each time step using equation (2). A vectorized implementation was utilized in charges.m:

```
v = zeros(1,nt);
for i = 1:nc-1
    differences = r(i,:,:)-r(1+i:nc,:,:);
    norms = sqrt(sum(differences.^2,2));
    v = v + reshape(sum(1./norms,1), size(v));
end
```

3.4 Equivalence Classes

Each equivalence class for the final charge equilibrium distributions is defined as the number of charges whose distances from every other charge are indistinguishable by a tolerance of ϵ_{ec}. The following accomplishes this sorting procedure by first initializing the difference matrix and then removing rows which share the same equivalence class.

```
v_ec = []; d = zeros(nc, nc);
end_rs = r(:, :, end);

for charge_num = 1:nc
    ri = end_rs(charge_num, :);
    d(charge_num,:) = sum((ri-end_rs).^2,2);
end
D = sort(d,2);

while size(D,1) ~= 0
    difs = D(1,:) - D;
    mags = sqrt(sum(difs.^2, 2));
    indices = find(mags<=epsec);
    v_ec = [v_ec length(indices)];
    D(indices,:) = [];
end

v_ec = sort(v_ec, 'descend');
```

With these implementations in charges.m, running the simulation for

```
nc = 60; tmax = 500; level = 12; gamma = 1; epsec = 1.0e-5;
```

takes roughly 4s, as profiled by the MATLAB profiler:

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
52	`sum_rijs = sum(all_rijs ./ magnitudes.^3);`	245700	1.368	35.8%	
50	`all_rijs = r([1:charge_number-1,charge_number+1:end],:,time_…`	245700	1.175	30.7%	
51	`magnitudes = sqrt(sum(all_rijs.^2,2));`	245700	0.318	8.3%	
56	`r_nminus1_term = (2/dt - gamma) * r(charge_number, :, time_s…`	245700	0.292	7.6%	
57	`r_n_term = (-4/dt)*r(charge_number, :, time_step);`	245700	0.239	6.2%	
All other lines			0.433	11.3%	
Totals			3.824	100%	

4 Experiments and results

4.1 4-level convergence test

Utilizing convtest.m, calculations are performed with

```
nc=4; tmax = 10; gamma = 1; epsec = 1.0e-5;

r0 = [ [1, 0, 0]; [0, 1, 0]; [0, 0, 1]; (sqrt(3)/3) * [1, 1, 1] ];
```

at each of 4 discretization levels 10, 11, 12, 13. The level-to-level differences are defined

$$\delta x_{10} = x_{11} - x_{10}$$

$$\delta x_{11} = x_{12} - x_{11}$$

$$\delta x_{12} = x_{13} - x_{12}$$

where x_ℓ is the x-coordinate of the first charge across all time steps at a discretization level ℓ. The level-to-level differences scaled by a factor of ρ are plotted in Figure 1.

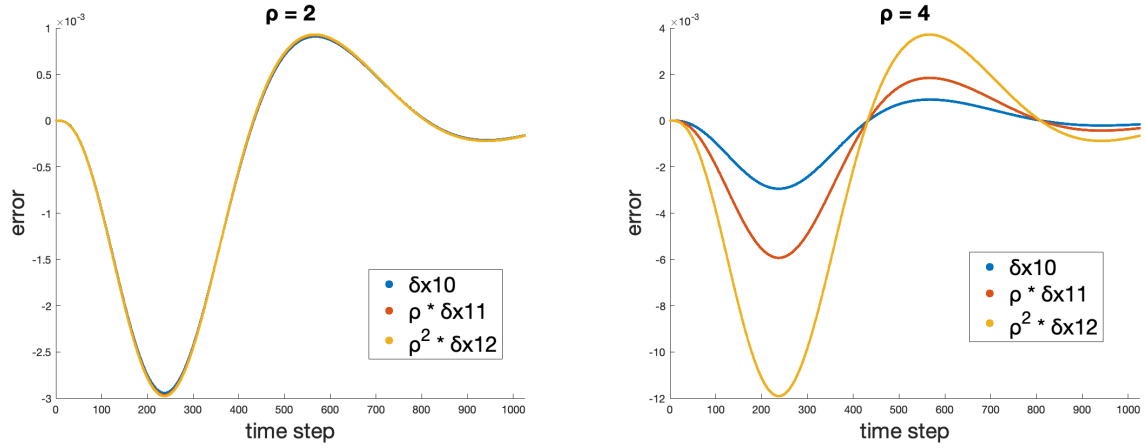

Figure 1: Plot of level-to-level differences $\delta x_{10}, \rho\delta x_{11}$and $\rho^2\delta x_{12}$ for $\rho = 2$ (left) and $\rho = 4$ (right)

From the plots, we can see that $\rho = 2$ gives near coincidence of the curves. Since the grid level is finer by a factor of 2 each time we increase the level, from $x_{10} \to x_{11} \to x_{12} \to x_{13}$ will see a change of $\Delta t \to \frac{\Delta t}{2} \to \frac{\Delta t}{4} \to \frac{\Delta t}{8}$. This will result in a change of level-to-level error $e \to \frac{e}{2} \to \frac{e}{4}$ if the FDA implementation is $O(\Delta t)$ accurate, or a change of level-to-level error $e \to \frac{e}{2^2} \to \frac{e}{4^2}$ if the FDA implementation is $O(\Delta t^2)$ accurate. From the plots, we observe the first case where $e = \rho\frac{e}{2} \to \rho^2\frac{e}{4}$ when $\rho = 2$ and thus conclude that the FDA is $O(\Delta t)$.

4.2 Time evolution of potential for 12 charge distribution

Using the following parameters, the numerical analysis is performed for N = 12 in plotv.m.

```
nc = 12; tmax = 10; level = 12; gamma = 1; epsec = 1.0e-5;

r0 = normalize(2*rand(nc,3) - 1);

[t, r, v, v_ec] = charges(r0, tmax, level, gamma, epsec);
clf; plot(t,v,'.');
title("V(t) vs t for 12 charges"); xlabel("t"); ylabel("V(t)");
```

The resulting plots is shown in Figure 2

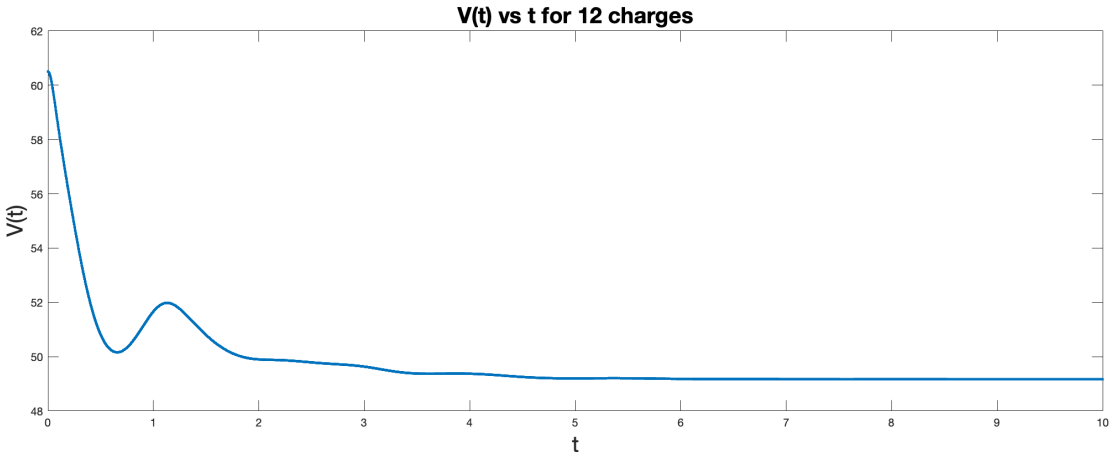

Figure 2: plot of V versus t for a distribution of 12 charges. The voltage begins at around 60 and eventually converges to a value of 49.1655

The equilibrium charge distribution on the sphere is shown in Figure 3.

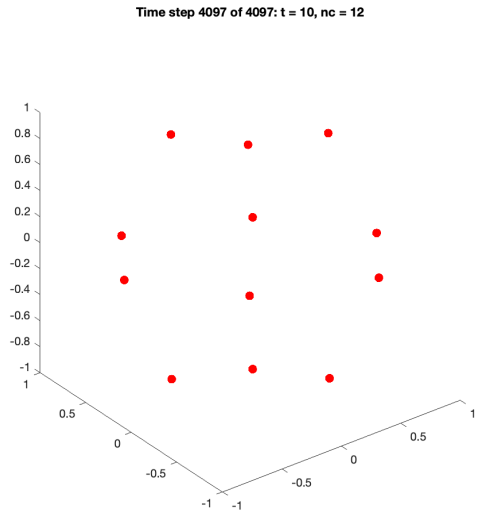

Figure 3: Equilibrium spatial distribution for 12 charges

4.3 Survey of $V(t_{max}; N)$and $v_{ec}(N)$

Using the script survey.m, a survey for N from 2 to 60 with parameters

```
tmax = 500; level = 12; gamma = 1; epsec = 1.0e-5;
```

is performed. The final values of V and equivalence class counts for each N is saved to vsurvey.dat and ecsurvey.dat. The results are also summarized in Table 1.

Table 1: Survey results for the equilibrium voltage and equivalence classes for N = 1 to 60.

N	V	Equivalence Classes
2	0.5	2
3	1.732050808	3
4	3.674234614	4
5	6.474691495	3 2
6	9.985281374	6
7	14.45297749	1 1 1 1 1 1 1
8	19.67528786	8
9	25.75998653	6 3
10	32.71694946	8 2
11	40.59645051	4 2 2 2 1
12	49.16525306	12
13	58.85323061	4 2 2 2 2 1
14	69.3063633	12 2
15	80.67024411	6 6 3
16	92.9116553	12 4
17	106.0504048	10 5 2
18	120.0844674	8 8 2
19	135.0894684	2 2 2 2 2 2 2 2 2 1
20	150.8815683	6 6 6 2
21	167.6416224	2 2 2 2 2 2 2 1 1 1 1 1 1 1
22	185.2875361	12 6 4
23	203.9301907	6 6 6 3 2
24	223.3470741	24
25	243.8127604	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
26	265.1333263	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
27	287.302615	10 10 5 2
28	310.4915424	12 12 4
29	334.6344399	2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
30	359.6039459	2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
31	385.5308381	6 6 3 3 3 3 3 3 1
32	412.2612747	20 12
33	440.2040578	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
34	468.9048533	4 4 4 4 4 4 4 4 2
35	498.5698725	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
36	529.1224091	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
37	560.6279731	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
38	593.0489435	4 4 4 4 4 4 2 2 2 2 2 2 2

39	626.389009	12 6 6 6 6 3
40	660.6752788	12 12 12 4
41	695.9167443	12 6 6 6 6 3 2
42	732.0781075	10 10 10 10 2
43	769.1908479	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1
44	807.1742631	8 8 8 8 8 4
45	846.1884038	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
46	886.1671136	12 12 12 6 4
47	927.072256	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
48	968.7134553	24 24
49	1011.557183	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1
50	1055.182315	12 12 12 12 2
51	1099.81929	6 6 6 6 6 6 6 6 3
52	1145.418964	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1
53	1191.922291	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
54	1239.361475	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
55	1287.772721	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
56	1337.094945	4 4 4 4 4 4 4 4 4 4 4 4 4 4
57	1387.383229	6 6 6 6 6 6 6 6 6 3
58	1438.638105	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
59	1490.774386	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
60	1543.83040	6 6 6 6 6 6 6 6 6 6

4.3.1 Equilibrium Energy

Using the results in vsurvey.dat, the equilibrium voltages are plotted versus N in Figure 4.

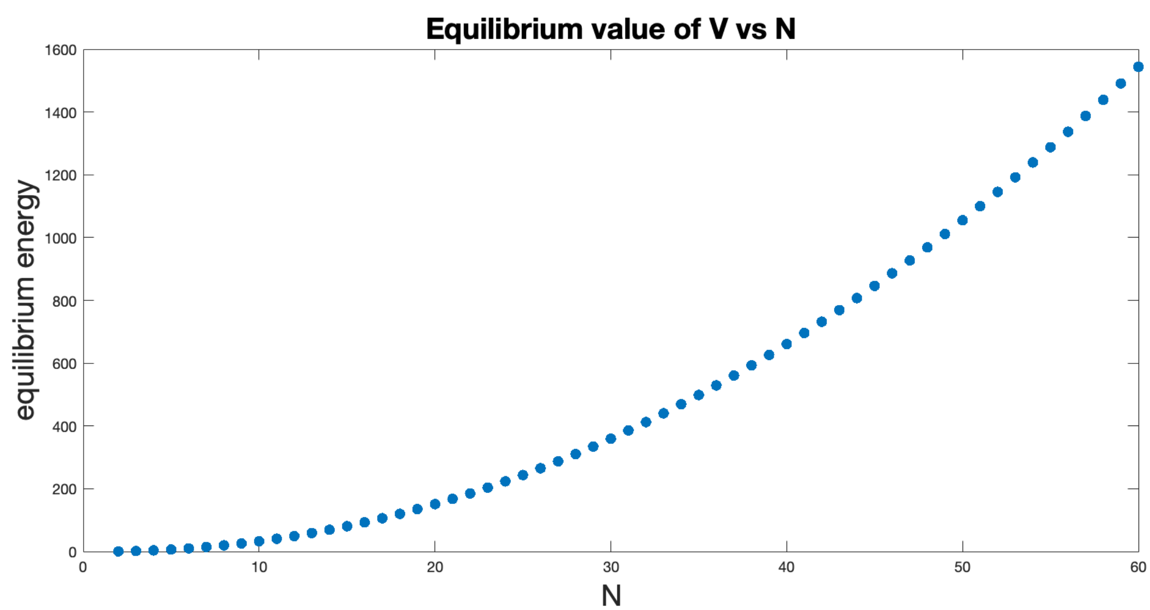

Figure 4: Equilibrium V

From the plot, we can see that there is an approximately quadratic relation between N and V.

4.3.2 Fitting the data using least squares

Seeing a roughly quadratic relationship between N and V, we attempt to fit the equation of the curve using least squares. That is want to solve a second order equation

$$cN_i^2 + bN_i + aN_i = V_i \quad \text{for } i = 2, 3, ...60$$

Writing this in matrix notation we have

$$Zw = V$$

where N is the vector of the number of charges and V is the vector for the corresponding equilibrium voltages, Z is the transformed matrix of N for second order fitting, and w is the vector of the coefficients

$$N = \begin{bmatrix} 2 \\ 3 \\ 4 \\ \vdots \\ 60 \end{bmatrix}, \quad V = \begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ \vdots \\ V_{60} \end{bmatrix}, \quad Z = \begin{bmatrix} | & | & | \\ N^0 & N^1 & N^2 \\ | & | & | \end{bmatrix}, \quad w = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

We attempt to solve this using least squares by minimizing the sum of squared errors

$$||Zw - V||^2$$

which is equivalent to minimizing

$$\frac{1}{2}||Zw - V||^2$$

taking the derivative and setting to zero leads to

$$Zw - V = 0$$

Now we can use least squares to solve for the weights

$$Z^T Zw = Z^T V$$

solving :

$$w = (Z^T Z) \setminus (Z^T V)$$

This is performed in plotfinalv.m and the curve fit is shown in Figure 5

We can read off the values of the weight matrix w to obtain the fitted equation between N and V as

$$V = 0.4603N^2 - 2.0212N + 6.0755$$

The validity of the solution is examined by plotting the absolute and relative errors in Figure 6 and 7. We note that the relative error decreases for large N but has very high relative error for low values of N.

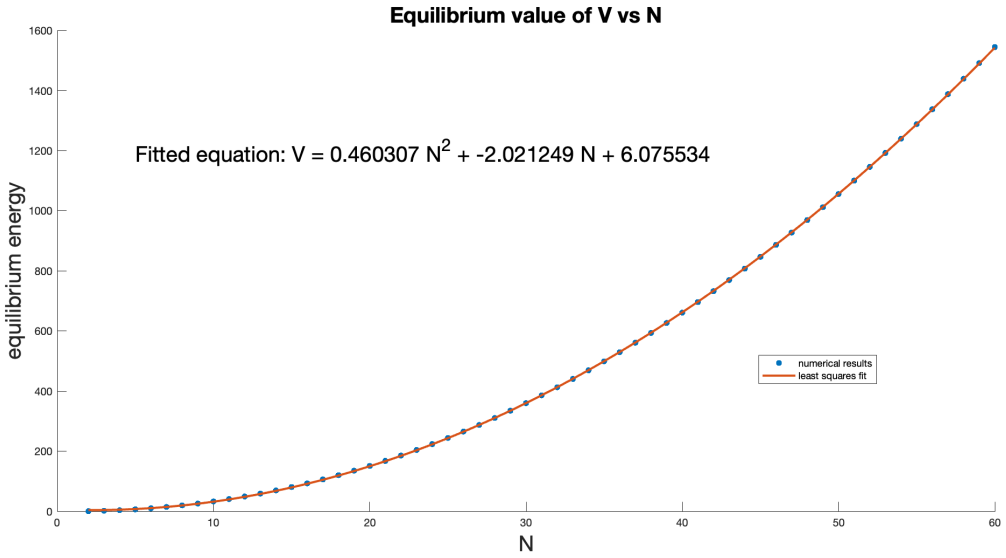

Figure 5: Curve fit of N vs V using least squares.

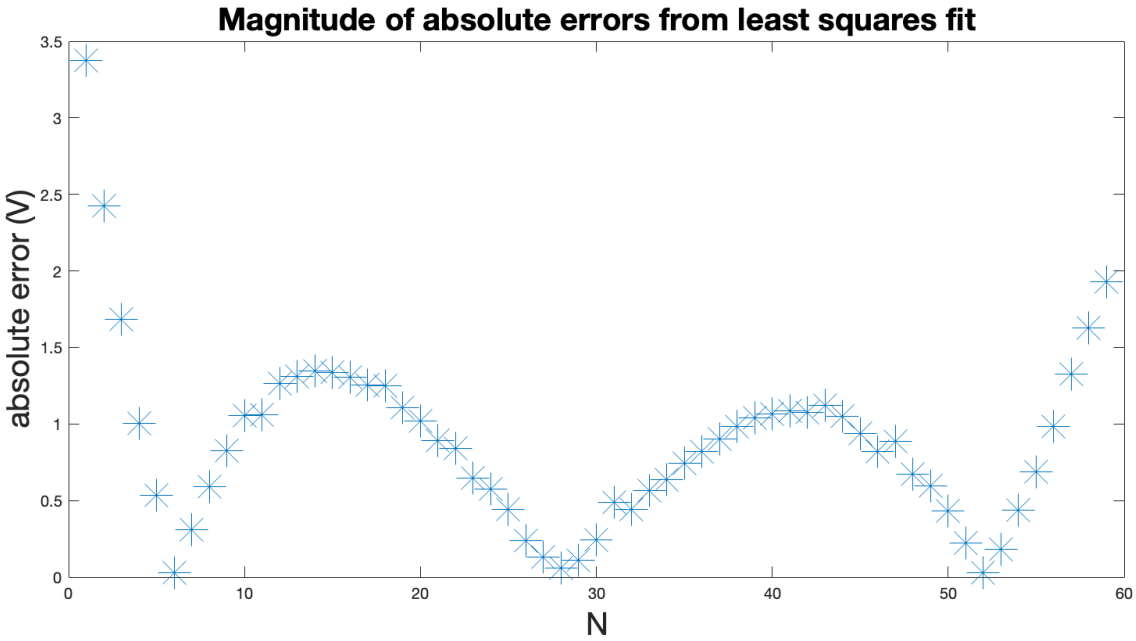

Figure 6: Absolute error from least squares.

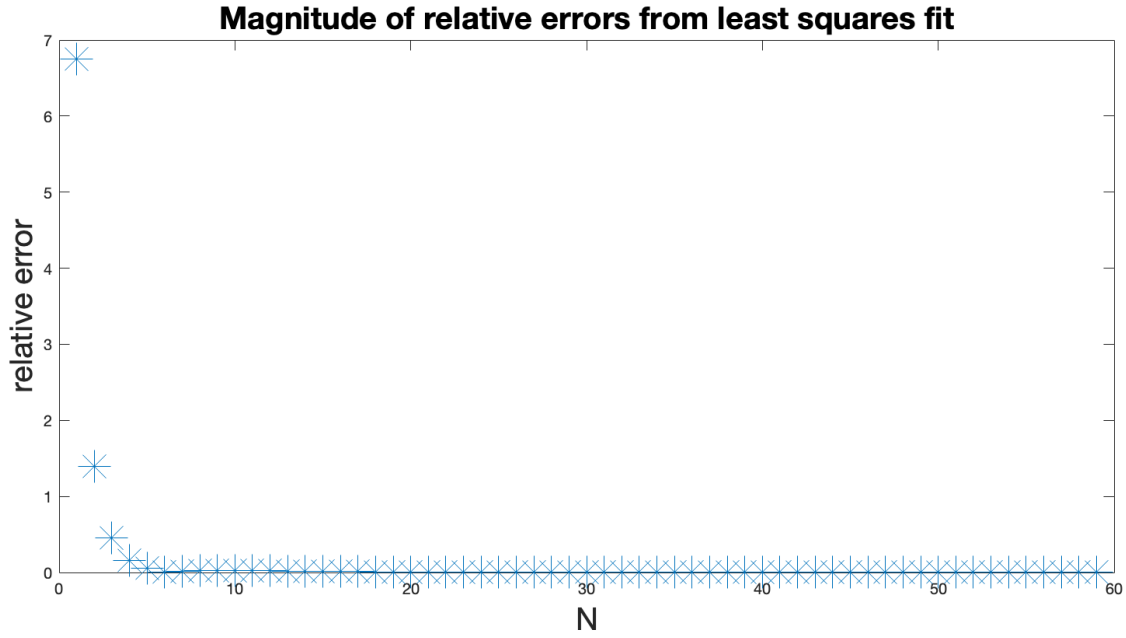

Figure 7: Relative error from least squares

4.4 Fitting the data using Barycentric Interpolation

Utilizing the barycentric.m function from tutorial 2, we also attempted to fit the equation using the barycentric interpolation method. 3 points were selected for a second order fit, corresponding to $(N_i, V_i) = (2, V_2), (31, V_{31}),$ and $(60, V_{60})$. barycentric.m was run with these as input parameters and 3 arbitrary points x_1, x_2, x_3 were chosen to be evaluated. With the values y_1, y_2, y_3 that are returned from barycentric.m fit, a deterministic equation can be determined. We set up a system of equations to solve the fitted polynomial $ax^2 + bx + c = y$

$$\begin{bmatrix} {x_1}^2 & x_1 & 1 \\ {x_2}^2 & x_2 & 1 \\ {x_3}^2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

and solve for a, b, c in polyfitv.m. The final equation is determined as

$$V = 0.4597N^2 - 1.894N + 2.4495$$

The curve fit and absolute and relative errors are plotted in Figures 8-10. We note that the relative and absolute errors follow the same trend as those observed from when using least squares. However, the magnitude of relative errors using barycentric interpolation is more than an order of 10 less than those from least squares, while the magnitudes of the absolute error is roughly the same. Thus we can say that the equation obtained by barycentric interpolation is a better representation of the actual relationship between N and V.

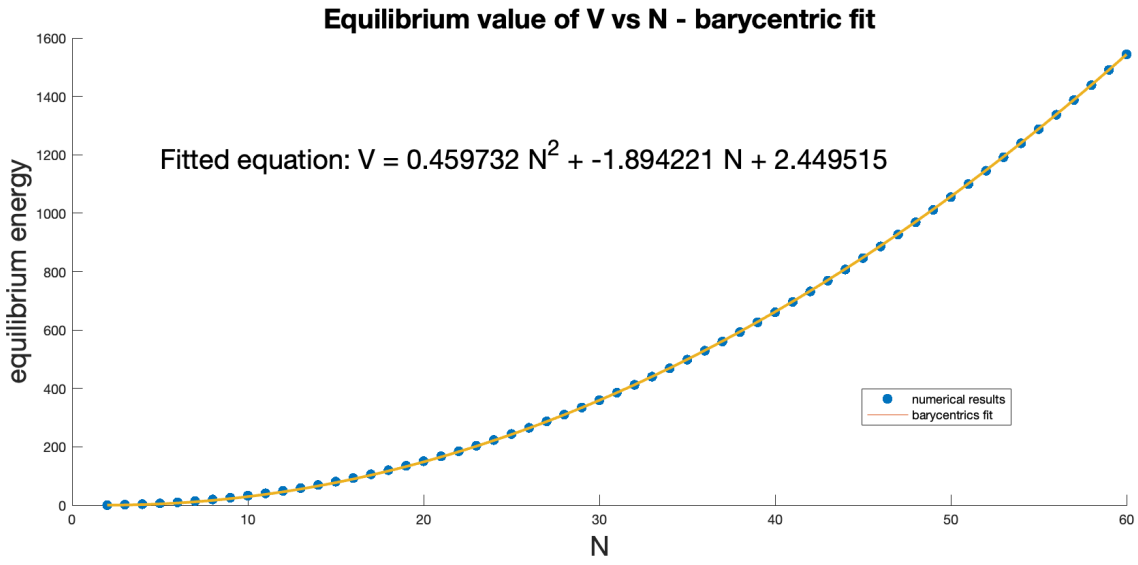

Figure 8: Fit to data using Barycentric Interpolation

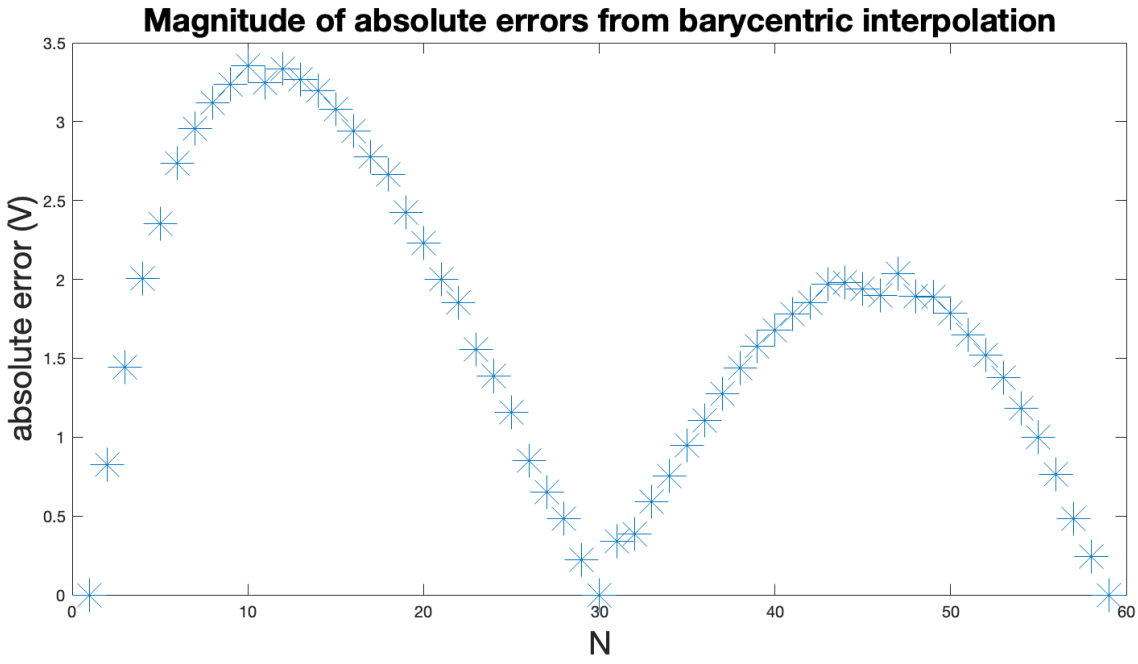

Figure 9: Absolute error from Barycentric Interpolation.

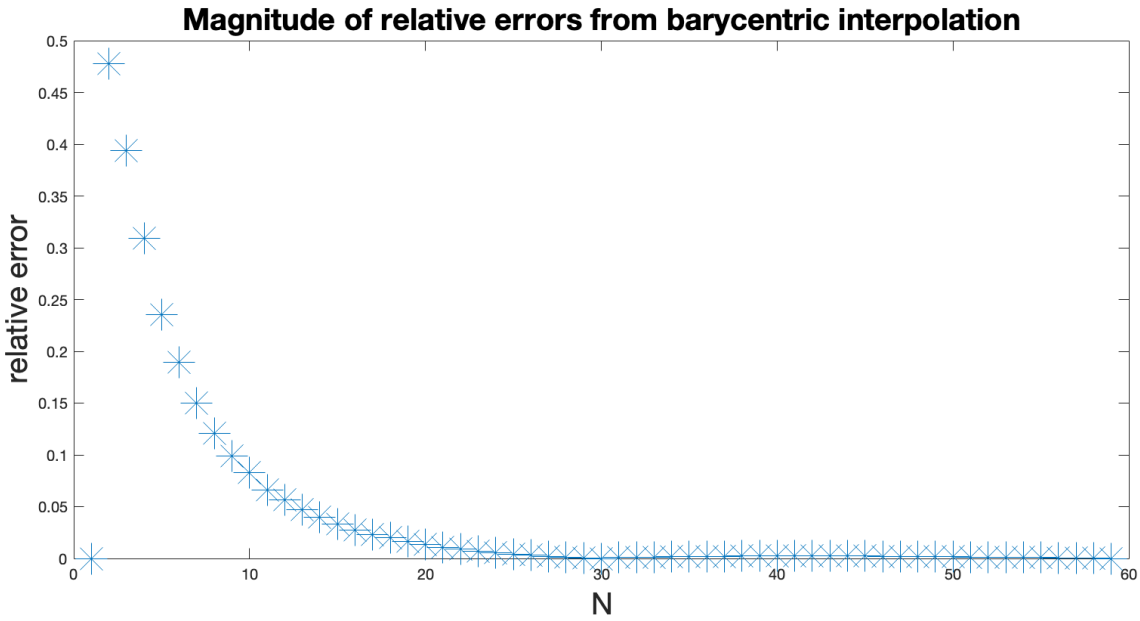

Figure 10: Relative error from Barycentric Interpolation

4.4.1 Errors in final V

The known values for V from wikipedia (https://en.wikipedia.org/wiki/Thomson_problem) were downloaded and loaded into MATLAB for comparison to the numerical solution in compare_vs.m. The magnitude of absolute error was determined using

$$e_{abs} = |V_{numerical} - V_{wikipedia}|$$

The errors in comparisons for the numerical solutions for the first 60 charges are plotted in Figure 11.

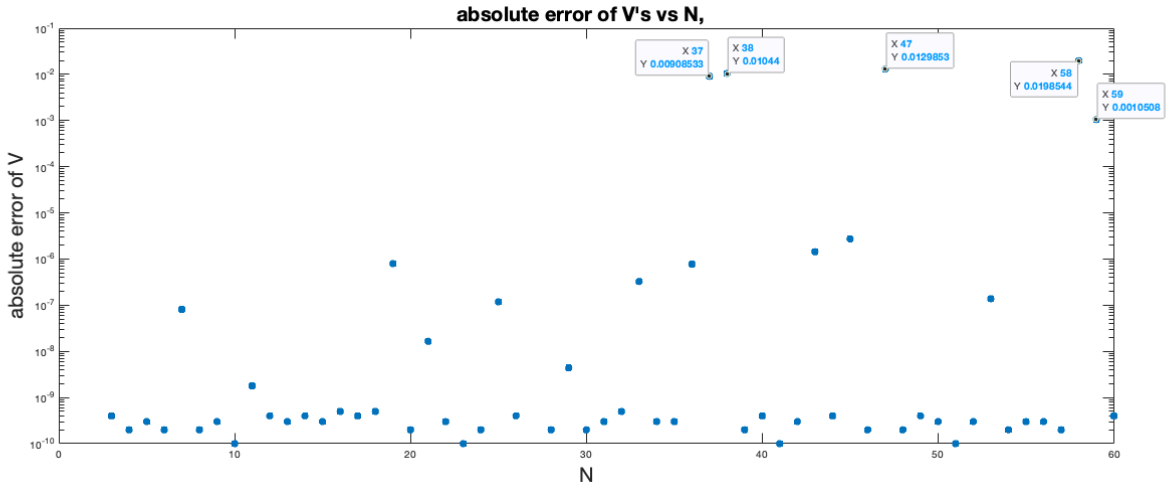

Figure 11: Plot of absolute errors for final values of V computed compared to the given values from wikipedia for each N.

It is notable that the method is accurate for most values of V up to a absolute error magnitude of around 10^{-5} while for a few troublesome cases, the magnitude of the absolute error is in the order of 10^{-2} (namely the cases of N = 37, 38, 47, 58 and 59)

4.4.2 Video of sample evaluation

The time evolution for 47 charges was plotted using in create_charges_video.m

```
tmax = 50; level = 12; gamma = 1; epsec = 1.0e-5;
r0 = normalize(2*rand(nc,3) - 1);

[t, r, v, v_ec] = charges(r0, tmax, level, gamma, epsec);
charges_video(t,r);
```

After around 10s, the charges converge to an equilibrium solution as shown in Figure 12. The video file is saved as charges_47.avi

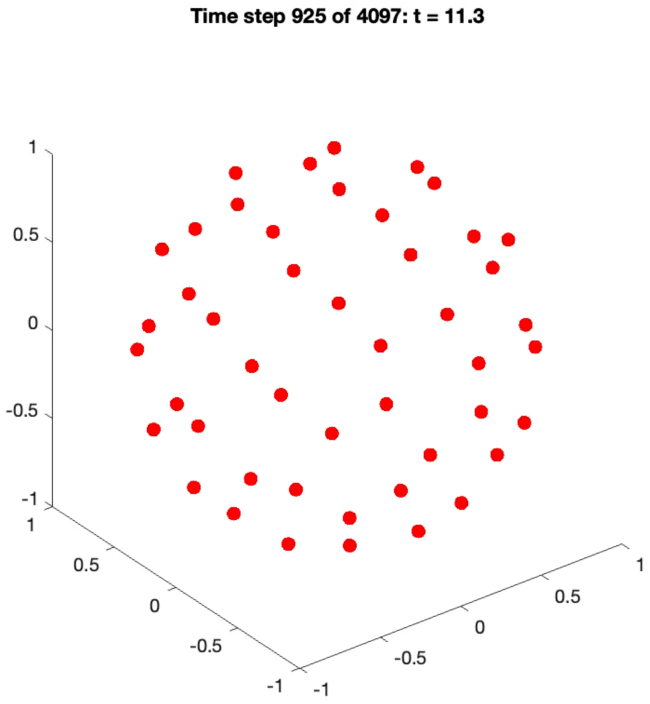

Figure 12: Distribution of 47 charges after ∼10s. No more motion is observed in the video after this point.

5 Conclusions

In this project, the spatial arrangements of N particles distributed around a sphere were analyzed using finite difference approximations. The results of the implementation appeared to be reasonable and also agreed with the known solutions in literature. It was noted that the calculations converged with high error tolerance around a few particular cases of N's. It is interesting to note that for each of these cases, they exhibited a high number of equivalence classes and the final arrangement of the particles was complex in geometry.

Improvements can be made to the analysis methods by utilizing FDAs with higher order accuracy. The methods in this project were $O(\Delta t)$ and making use of methods which are $O(\Delta t^2)$ or higher could result in higher accuracy of the particle configurations.

In addition, two curve fitting methods were examined to fit the relationship of equilibrium voltage to number of charges. it was determined that using barycentric interpolation resulted in lower relative errors and likely gave a better representation of the equation. However, it was still not very accurate for small values of N. A more thorough survey should be done fit equations that minimize the relative error, and can also be validated by checking the equation against the results of higher N values, in order to determine a more accurate equation for V vs N.