Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Christopher Tre Presley April 1, 2021

Determination of Repulsive Force Law of Jammed Emulsion Droplets through Minimization

by

Christopher Tre Presley

Dr. Eric R. Weeks

Adviser

Department of Physics

Dr. Eric R. Weeks

Adviser

Dr. Stefan Boettcher

Committee Member

Dr. Jacobus De Roode

Committee Member

2021

Determination of Repulsive Force Law of Jammed Emulsion Droplets through Minimization

by

Christopher Tre Presley

Dr. Eric R. Weeks

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Physics

2021

Abstract

Determination of Repulsive Force Law of Jammed Emulsion Droplets through Minimization
By Christopher Tre Presley

The repulsive force due to surface tension, acting between jammed emulsion droplets, is equivalent to $f_{ij} = F_0\left(\delta_{r_{ij}}/d_{ij}\right)^{\beta_f}$ where the overlap between particles *i* and *j*, $\delta_{r_{ij}}$, and the center to center distance between particles *i* and *j*, d_{ij}, can be easily found using experimental data. However, the force scaling factor F_0—a force scaling factor—and the power law β_f are not well known. Using experimental data where emulsion droplets underwent a jamming transition and experienced both gravitational and repulsive forces, a created minimization program was used to determine that $F_0 = 3.3 \pm 0.1\ \mu N$ and $\beta_f = 1.20 \pm 0.05$ are the scaling factor and power law, respectively, that are most consistent with the experimental data. The power law is consistent with previous literature, but the scaling factor is not. This could be due to the use of a subset, rather than the full set, of experimental data points.

Determination of Repulsive Force Law of Jammed Emulsion Droplets through Minimization

By

Christopher Tre Presley

Dr. Eric R. Weeks

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Physics

2021

Table of Contents

Determination of Repulsive Force Law of Jammed Emulsion Droplets through Minimization

Christopher T. Presley and Eric R. Weeks

Abstract:

The repulsive force due to surface tension, acting between jammed emulsion droplets, is equivalent to $f_{ij} = F_0 \left(\delta_{r_{ij}}/d_{ij}\right)^{\beta_f}$ where the overlap between particles *i* and *j*, $\delta_{r_{ij}}$, and the center to center distance between particles *i* and *j*, d_{ij}, can be easily found using experimental data. However, the force scaling factor F_0—a force scaling factor—and the power law β_f are not well known. Using experimental data where emulsion droplets underwent a jamming transition and experienced both gravitational and repulsive forces, a created minimization program was used to determine that $F_0 = 3.3 \pm 0.1\ \mu N$ and $\beta_f = 1.20 \pm 0.05$ are the scaling factor and power law, respectively, that are most consistent with the experimental data. The power law is consistent with previous literature, but the scaling factor is not. This could be due to the use of a subset, rather than the full set, of experimental data points.

Introduction:

The saying that two things mix "like oil and water" refers to the tendency of a mixture of oil and water to separate into two distinct layers. However, adding dish soap to such a mixture results in a mixture where the oil droplets do not coalesce and form what is known as an emulsion. An emulsion is a combination of two or more, though normally just two, immiscible liquids that creates a liquid-to-liquid phase separation (i.e. a fluid system in which liquid droplets are distributed within another liquid).[1] Within an emulsion, one liquid is in the dispersed phase while the other is in the continuous phase; the boundary between the two phases is known as the

"interface".[2] The oil-in-water emulsion described above is probably the most well-known type of emulsion. Milk, vinaigrettes, mayonnaise, and butter are also examples of emulsions. When dispersed in a continuous phase liquid, the oil molecules at the surface of the oil droplet do not have other dispersed phase oil molecules on all sides. These surface molecules, therefore, cohere strongly to other surface molecules, creating a surface tension. The surface tension results in the oil droplets (hereafter, referred to as "emulsion droplets") having a spherical shape as it lowers the internal pressure of the droplet. Surface tension, γ, can also be thought of as the surface energy, or the "energy cost" for the dispersed phase to have an interface with the continuous phase. Emulsifiers are used to lower the surface energy between the interfaces of the dispersed and continuous phases. In the oil-in-water example, the dish soap was used as an emulsifier, lowering the surface energy of the emulsion droplets so they do not coalesce.

The experimental data analyzed within this paper is from an experiment performed by Janna Lowensohn and Eric Weeks using an oil-in-water emulsion. This experimental set-up is a new model system for studying droplet physics as it allowed for forces inside the system to be measured and allowed. In the experiment, 882 emulsion droplets were produced using a standard co-flow micro-fluidic technique wherein the continuous phase was a mixture of water and "Fairy" dish soap. These droplets were placed into a 135 μm thick and roughly 2200 to 2300 μm wide chamber that was tilted 28˚ from horizontal. Figure 1 shows a schematic of both the side and top view of the oil droplets. As can be seen from the top view, the data can be viewed as quasi-two-dimensional. Because the density of the oil droplets $\left(0.83\ {}^{g}/_{cm^3}\right)$ is less than the density of water $\left(1.00\ {}^{g}/_{cm^3}\right)$, the oil droplets were buoyant, and, because the chamber was tilted, the droplets floated to the upper end of the chamber, pushing against one another. As the emulsion droplets pushed against each other, the packing fraction (i.e. the number of emulsion droplets within a given space) increased, causing a jamming transition.[3] Jamming gives rise to a system of emulsion droplets that behave more like a solid than a liquid (e.g. the system can no longer undergo microscopic reconfiguration in the presence of an external force).[3, 4] While in this jamming transition, the internal pressure of the droplets increased, so the emulsion droplets formed quasi-

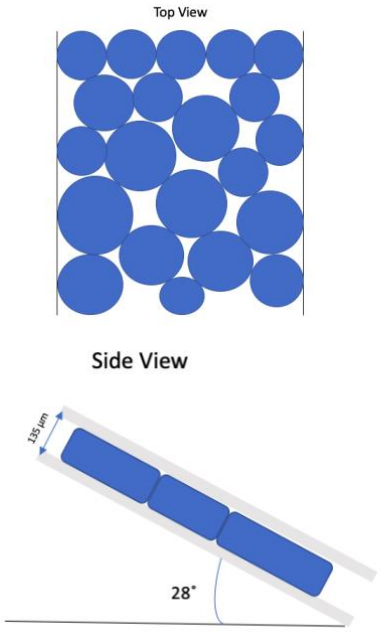

Figure 1. This figure shows both a top and side view of the emulsion droplets from the experimental data where the black lines are the walls of chamber. The gray slides on the side view depict the glass sides on the top and bottom of the chamber.

two-dimensional circles in order to lower this internal pressure. But, to reduce the gravitational potential, the droplets deformed slightly, resulting in a repulsive force between the particles. Therefore, the droplets experienced both gravitational forces as well as forces from other droplets (hereafter, referred to as the "repulsive force"). When the net force experienced by each droplet was zero, the emulsion droplets reached an equilibrium point where the droplets were motionless. It was at this point that a low magnification microscope lens was used to take many pictures of the droplets which were stitched together to create an image of all the emulsion droplets within the chamber. This image was used to gather both x-position and y-position data for the centers of each droplet as well as the radius of each droplet. The gravitational force on each emulsion droplet can be easily determined; however, the repulsive force between the droplets due to jamming is not well known. Desmond et al. determined a repulsive force law that is dependent upon the contact length of the interface of the two droplets and the modified radius of curvature. [4] However, it is much easier to think about a force law that is just dependent upon the distance between the particles, or a central force. Therefore, in order to determine an approximate force law for the quasi-two-dimension system of emulsion droplets, Desmond et al. found the average force the emulsion droplets experienced and plotted it as a function of the separation of the droplets. This allowed for an approximate repulsive force law to be determined, and it was found to be $f_{ij} = F_0 \left(\delta_{r_{ij}}/d_{ij}\right)^{\beta_f}$ where F_0 is the force scaling factor, $\delta_{r_{ij}}$ is the overlap between particles *i* and *j*, d_{ij} is the sum of the radii of the particles *i* and *j*, and β_f is the power law for the force between the particles. The overlap between particle *i* and particle *j* can be found using $\delta r_{ij} = r_{ij} - d_{ij}$ where r_{ij} is the center-to-center distance of between particle *i* and particle *j*. Therefore, because the data contains position data and radii data for each particle, both $\delta_{r_{ij}}$ and d_{ij} can be easily calculated. However, the scaling

factor F_0 and power law β_f are not well known and have not been extensively studied. Desmond et al. have previously studied the approximate force law. However, they studied this by finding the average repulsive forces two emulsion droplets were feeling and plotted it against there separation. This allowed them to determine an approximate force law from the empirical force law. The object of this paper is to determine the values for both the scaling factor and the power law for the repulsive force law using minimization techniques in Python.

Methods and Results:

In order to determine the scaling factor and power law for the repulsive force expression, the motions of the emulsion droplets were simulated based on differing force laws (i.e. force laws with different scaling factors and power laws). If the droplets were allowed to move based on the forces present, they would move to a new equilibrium point based on the force law. The force law that resulted in the least movement, characterized by the lowest error value, is the force law that is most consistent with the repulsive force present within the experimental data since the data was taken when the emulsion droplets were motionless. Because the system was at equilibrium when the data was recorded, the potential energy was minimized. So, in order to find the most consistent force law, I knew I could minimize the potential energy of the emulsion data experiencing different force laws. Because the data is quasi-two-dimensional, I only needed to minimize the potential energy over just the x-position and y-position.

In order to minimize the potential energy of the experimental data, I first needed a minimization routine that worked. I decided to use Python for my project, since the open-source Python library SciPy contains a minimization function, scipy.optimize.minimize, that allows the user to input a pre-defined function to minimize. This built-in minimization program is equipped with many different minimization techniques, including techniques based on the Simplex

algorithm and the nonlinear conjugate gradient algorithm. I arbitrarily decided to use the Nelder-Mead technique, a technique based on the Simplex algorithm, as I was familiar with the Simplex algorithm. In using this built-in minimization program, I had no clue whether the function would be reliable in giving a minimum geometry for a system of particles. So, to determine whether or not scipy.optimize.minimize could be trusted to use for the minimization of the experimental data, I used this function to minimize a known force law, the Coulomb force law, as a test.

I. Testing Built-in Minimization Function using the Coulomb Force

The Coulomb force, also known as the electrostatic force, is the attractive or repulsive force on particles based on their electric charge. So, if two positively-charged particles are subjected to electrostatic forces and allowed to move, the charges will move as far away from each other as possible. This movement, in turn, minimizes the electrostatic potential energy of the two particles. The electrostatic potential energy between particles is equal to $U_E = k\frac{q_i q_j}{r_{ij}}$, where k is the Coulomb constant, q_i and q_j are the charges of particle *i* and particle *j* respectively, and r_{ij} is the distance between particles *i* and *j*. In order to simulate the electrostatic potential energy, boundary conditions were needed. However, there was a problem with the Coulomb force: it required boundary conditions. If boundary conditions were not present, the particles would have moved to infinity. I, also, thought about creating a "box" out of which the particles could not escape. However, the edges of the box would have been crowded by the particles when the potential energy was minimized, telling me nothing about whether

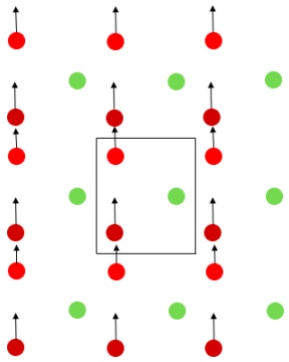

Figure 2. This figure shows a schematic of the periodic boundary conditions. The square outlined in black is the unit cell, and the red particle leaving the top of the unit cell reappears at the bottom of the cell.

the built-in function was reliable. Therefore, periodic boundary conditions, illustrated in figure 2, were used. Periodic boundary conditions are used to approximate a large system by using a "unit cell". When an object passes through one side of the unit cell, it re-appears on the opposite side of the unit cell. Using periodic boundary conditions allowed me to predict the minimum geometry of the system, based on geometry, before actually minimizing the system in order to visually understand the results. Using the electrostatic potential energy with periodic boundary conditions, the built-in scipy.optimize.minimize was used to simulate the motion of n-randomly placed particles on a l x l lattice. I began with a system of two positively, equally charged particles. Because of the periodic boundary conditions, the furthest the two particles could get from each other occured along a diagonal line with a distance of $\frac{l}{\sqrt{2}}$. This geometry is depicted in figure 3. The minimization routine resulted in just such a geometry. After minimizing this system approximately ten times and getting the predicted geometry, I decided to move to a system of three positively, equally charged particles. I predicted that, because of the periodic boundary conditions, the resulting minimum geometry should contain two particles on the diagonal a distance $\frac{l}{\sqrt{2}}$ apart with the remaining particle a distance $\frac{l}{2}$ from both of the diagonal particles. This geometry is shown in figure 4 and is the result I received after most minimizations. However, it was in doing this simulation that the built-in

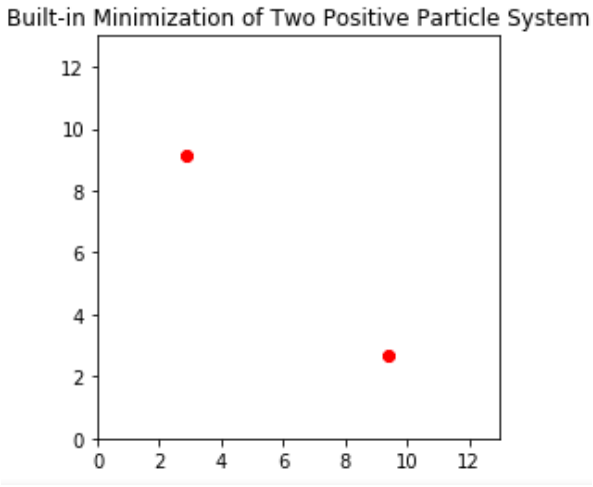

Figure 3. This figure shows the resulting geometry of a simulation of a system of two equally, positively charged particles using the built-in minimization routine, scipy.optimize.minimize. The lattice is thirteen unit-lengths wide and thirteen unit-lengths long. The distance between the particles is 9.1924 unit-lengths. Because $\frac{13}{\sqrt{2}} = 9.1924$, this is the expected geometry for a system of two equally, positively charged particles..

minimization routine didn't reliably give my predicted minimum geometry. Approximately 20% to 25% of the minimizations resulted in the geometry depicted in figure 5. The resulting potential energy of the geometry in figure 5 is greater than the potential energy of geometry in figure 4. The horizontal distance of the far-right particle is 6.50 unit-lengths. Because the particles in the vertical line are a distance of 6.50 unit-lengths long, the distance between each of the particles in the horizontal line to the vertical line is $\sqrt{6.50^2 + 3.25^2} = 7.267$. Because the particle in the far right minimized to a distance of 6.50 away from the other two particles, I believed the geometry in figure 5 was a local minimum that was a sticking point for the built-in minimization program. Therefore, I reasoned that the minimization program was minimizing the system, either to a local minimum or a global minimum, and was confident that the minimization

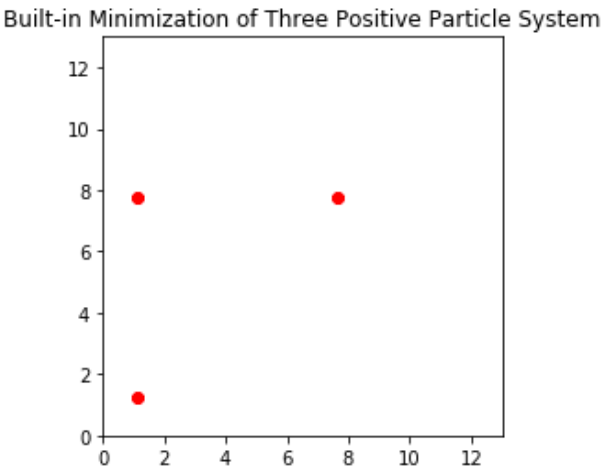

Figure 4. This figure shows the resulting geometry of a simulation of a system of three equally, positively charged particles using the built-in minimization routine, scipy.optimize.minimize. The lattice is thirteen unit-lengths wide and thirteen unit-lengths long. The distance between the particles along the diagonal is 9.1924 unit-lengths while the distance of the upper-left particle to the other two particles is 6.50 unit-lengths. Because $\frac{13}{\sqrt{2}} = 9.1924$ and $\frac{13}{2} = 6.50$, this is the expected geometry for a system of three equally, positively charged particles.

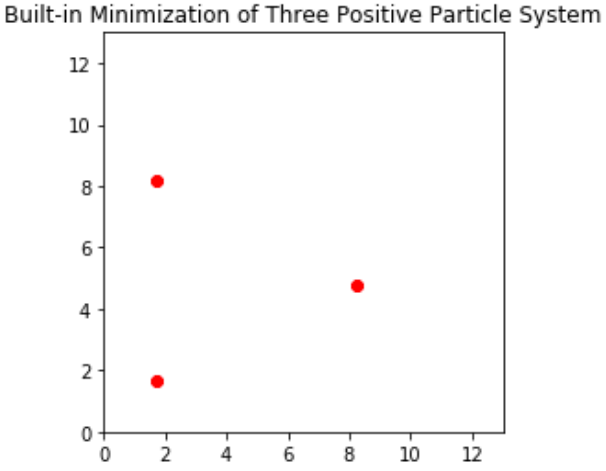

Figure 5. This figure shows the resulting geometry of a simulation of a system of three equally, positively charged particles using the built-in minimization routine, scipy.optimize.minimize. The lattice is thirteen unit-lengths wide and thirteen unit-lengths long. The distance between the particles along the straight line is 6.50 unit-lengths while the distance of the right most particle to the other two particles is approximately 7.267 unit lengths.

program was, in fact, doing what I expected it to. Therefore, I advanced to a system containing 10 positively, equally charged particles. For this system, I expected the geometry to be hexagonally packed, (i.e. the particles should rearrange themselves in a geometry that resulted in a hexagon). Unfortunately, this was not the result. Every time I minimized the system, the minimization routine gave a different resulting geometry with a different potential energy. The built-in minimization function didn't seem to be moving the particles to a geometry that resulted in one particular value for the potential energy. Three of many different final geometries of the ten particle system using the scipy.optimize.minimize function are shown in figure 6. These results are obviously not minima. Looking specifically at the geometry on the far right of figure 6, it is easy to see that the potential energy of the system could be lowered by spreading out the particles. Therefore, I decided I could not rely on the scipy.optimize.minimize function to minimize the experimental data.

II. Creating a Minimization Program and Testing it using the Coulomb Force

Because I couldn't rely on the SciPy minimization function, I created my own minimization routine. The basic principle behind my minimization routine was to find the potential energy of the original geometry of the system, take a random particle in the system and move it in any direction around a circle of radius *r* (creating a new geometry), calculate the potential energy of the new geometry, and keep the geometry—either the original geometry or the new geometry—

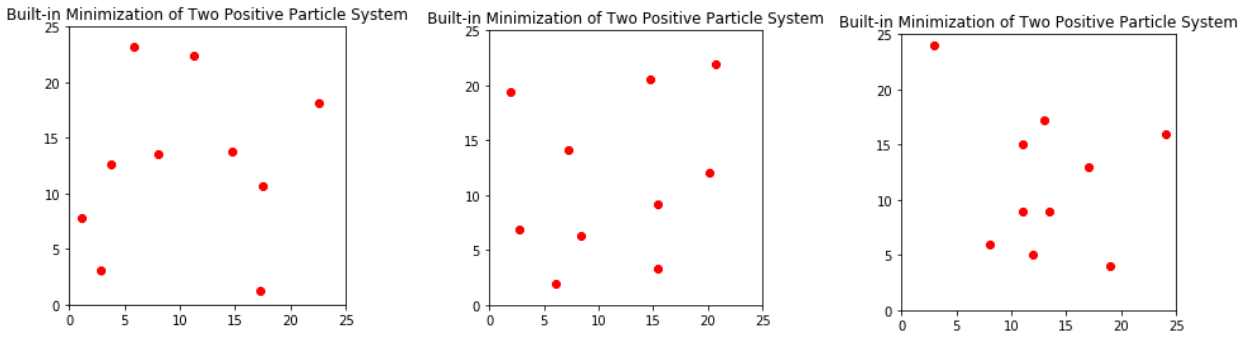

Figure 6. This figure shows three different resulting geometry of a simulation of a system of ten equally, positively charged particles using the built-in minimization routine, scipy.optimize.minimize.

that resulted in the lower potential energy. Repeating these actions until a certain difference between the potential energy was reached would theoretically result in the minimum geometry being found. To determine if my created minimization program (CMP) was reliable, I decided to test it, again using periodic boundary conditions and the electrostatic potential energy with systems containing varying numbers of particles. Starting with a system of two particles, my CMP consistently gave the expected minimum geometry shown in figure 3. After running the simulations and receiving the same results approximately ten to fifteen times, I began simulations using a three-particle system. Instead of resulting in two different geometries, my CMP resulted in the minimum geometry depicted in figure 4 every time. This allowed me to confidently advance to a system of ten particles. The minimization of the ten-particle system using my CMP resulted in either the geometry depicted in or in a geometry very similar to figure 7. The potential energies were also all very similar. After running simulations on this system over ten times and reliably obtaining the geometry in figure 7, I decided to try a system of fifty positively, equally charged particles. Every minimization of a system of fifty equally, positively charged particles using my CMP resulted in both a minimum geometry and a minimum potential energy similar to that depicted in figure 8. The last test for my CMP was on a system where one of the *n* positively charged particles had a charge that was much larger than the charges of the

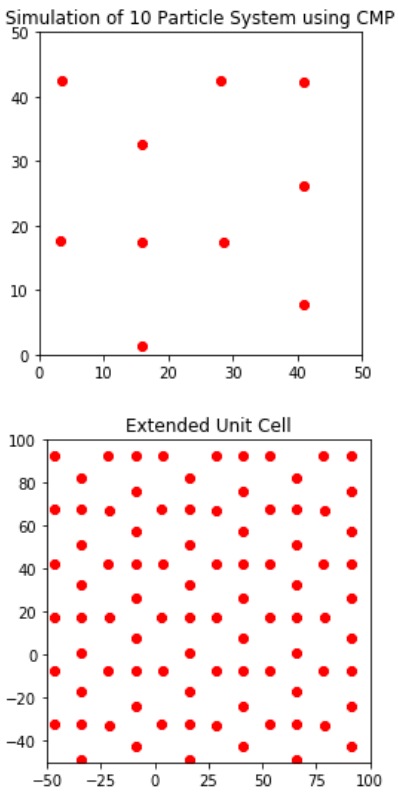

Figure 7. This figure shows the minimum geometry, along with the extended unit cell, of a simulation of a system of ten equally, positively charged particles using my created minimization program (CMP).

other $n-1$ equally, positively charged particles. I knew the $n-1$ particles would want to be as far away from the "large charge" as possible, so the "small charge" particles would be willing to sacrifice being closer to each other to be away from the large charge particle. After running simulations of ten-particle and twenty-particle systems for which the large charge particle had $q = 50$ and the remaining particles had $q = 1$, the minimum geometries matched my expectations, and the minimizations consistently resulted in minima that had similar potential energy values and geometries as those depicted in figure 9.

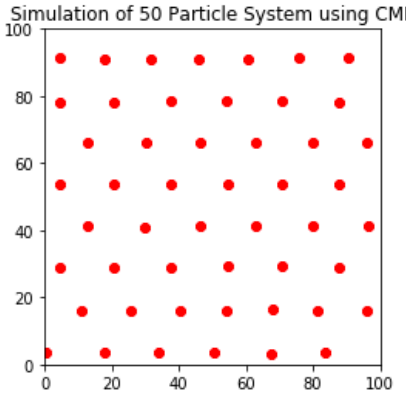

Figure 8. This figure shows the minimum geometry of a simulation of a system of fifty equally, positively charged particles using my created minimization program (CMP).

III. Minimizing the Experimental Data

After testing against the Coulomb force and gaining confidence that my CMP was reliable, I decided to begin running minimizations on the experimental emulsion droplet data. But, I first needed an expression for the potential energy of the system of emulsion droplets before I could do this. It is known that the potential energy is related to force by the expression $U = -\int_{ref}^{r} \vec{F} \cdot d\vec{r}$. So, using this expression and integrating over the overlap variable $\delta_{r_{ij}}$, the potential energy based on the repulsive force was determined to be $U_r = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{F_0}{\beta_f + 1} \left(\frac{{\delta_{r_{ij}}}^{\beta_f + 1}}{{d_{ij}}^{\beta_f}} \right)$ with units μJ. This repulsive potential energy is summed only over particles i and j if those particles touch, as the repulsive force between particles that do not touch is zero. The emulsion droplets also experienced gravitational potential energy. The gravitational force on each emulsion droplet i is equal to $F_{g,i} = \Delta\rho V_i g sin(28^\circ)$ where $\Delta\rho$ is the difference of the density of oil from the density of water, g is the gravitational acceleration constant, and V_i is the volume of emulsion droplet i. The $sin(28^\circ)$ term

is to account for the tilt of the chamber. Shown in figure 1, the emulsion droplets can be approximated as quasi-cylinders for which $V_i = \pi {d_i}^2 h$ where h is the thickness of the chamber (135 μm) and d_i is the radius of the emulsion droplets. Allowing the potential energy of original y-position for each particle to be zero, any deviation where the y-value decreases should cause a more negative potential energy (lowering the potential energy) and any deviation where the y-value increases should cause a more positive potential energy (increasing the potential energy). Therefore, the potential energy due to gravity is $U_g = \sum_{i=1}^{n} \Delta\rho g V_i (y_i - y_{0,i})\, sin(28°)$ in units μJ, where y_i is the y-position of emulsion droplet i when the potential energy is calculated and $y_{0,i}$ is the initial y-position of emulsion droplet i. Based on how I defined the gravitational potential energy, the total potential energy of the system of emulsion droplets is $U_T = U_p + U_g$. I, also, need to set new boundary conditions as periodic boundary conditions were not present within the experimental data. So, I stopped using periodic boundary conditions and "froze" the particles surrounding the edge of the subset of particles. This

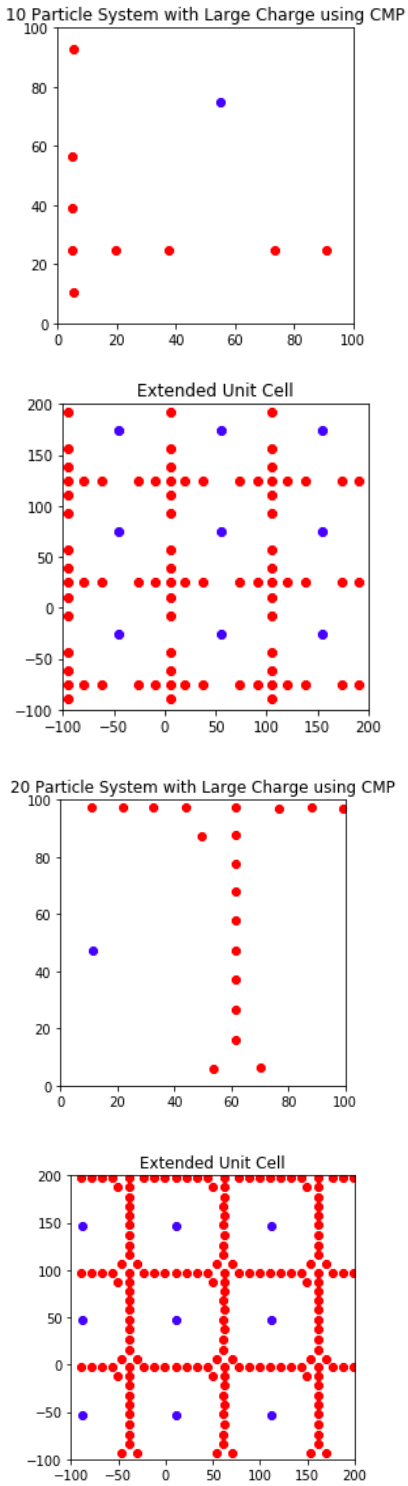

Figure 9. This figure shows the minimum geometry, along with the extended unit cell, of a simulation of a system of ten particles and twenty particles using my created minimization program (CMP). In these systems, the blue particle has a charge $q = 50$ and each of the red particles has charge $q = 1$.

essentially created a square box on which the simulation was run that the particles could not escape.

Now that the potential energy expression for the system is known and a new set of boundary conditions are set, the experimental emulsion droplet data can be minimized. Because of the long run time for the minimization of the full emulsion droplet data set, I decided to take a subset of the overall set of experimental data, containing 32 emulsion droplets with x-positions between 0 to 1500 μm and y-positions between 10,000 μm to 11,500 μm. I chose this set of data as the bounds of the data are square which allowed me to more easily analyze the results of the minimization through plotting. A subset of the data can be minimized to find the repulsive force law because all emulsion droplets in the data set experience the same repulsive force law. Desmond et al., with the same experimental design, found that values between 2.0 μN to 2.4 μN for the scaling factor F_0 and 1.15 to 1.30 for the power law β_f were the most consistent for their data set. Therefore, I decided to set the scaling factor equal to 2 μN and vary the power law from 0.5 to 2 in increments of 0.5, allowing me to find the error for a larger range of power laws. With each resulting minimum geometry, I calculated the error in distance of the minimized data points from the original data points. To do so, I calculated the distance that each data point moved, added each of these distances, and divided by the number of power laws I used. Finding the error this way "punishes" the power laws that may result in one or a few particles moving a great distance rather than all of the particles moving by an average distance. The graph of the error versus power-law is depicted in figure 10. As can be seen, the error is quadratic with respect to the power law, and the minimum error seems to occur somewhere between the values

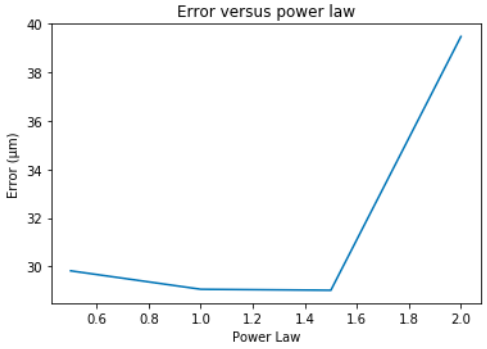

Figure 10. This figure shows the error, in μm, versus the power law used for the minimization.

1.0 and 1.5. So, the original subset of data was minimized again with the scaling factor value again equal to 2.0 μN but with power-law values varying between 1.1 and 1.5 increasing by 0.1. The graph of the error versus power-law for this set of simulations is depicted in figure 11 and shows that the minimum value is somewhere between 1.1 and 1.3. So, I repeated the minimization on the original data subset with power law values between 1.1 and 1.3 increasing by 0.025. The results of multiple simulations are shown in figure 12. When running simulations with power laws this precise, noise becomes a factor. The quadratic behavior of the error as a function of the power law exponent is lost, and it is harder to determine which value is the true value of the power law. The minimum values for the graphs in figure 12 range between 1.15 and 1.25 with 1.20 being the value that was resulted in the minimum error the majority of the simulations run. A power law value $\beta_f = 1.20 \pm 0.05$ gives the lowest error value and is, therefore, the power-law most consistent with the experimental data.

Figure 11. This figure shows the error, in μm, versus the power law graph of the minimum geometry from the original geometry of the subset of data versus the power law used for the simulation.

Using $\beta_f = 1.20$ as the power law, the subset of experimental data was minimized varying

Figure 12. This figure shows the error versus power law graph— where the error is the error in distance with units μm—after running the simulation on each of the varying power laws three times. The minimum values range between 1.15 and 1.25. Therefore, the minimum power law is $\beta_f = 1.20 \pm 0.05$.

the scaling factor. Again, Desmond et al. found scaling factors between 2.0 μN and 2.4 μN to be

most consistent with their data with the same experimental design. Therefore, minimizations were run with scaling factors between 1.0 μN and 4.0 μN, increasing the value by 0.5 μN each minimization. The error versus scaling factor graph is shown in figure 13. As can be seen, the minimum error occurred with scaling factors between 3.0 μN and 4.0 μN. So, simulations were repeated with scaling factors between 3.0 μN and 4.0 μN, increasing the value by 0.1 μN each simulation. This was repeated in excess of twenty times. Two of the resulting error versus scaling factor graphs are depicted in figure 14. Again, noise is present in the graph. In each of the times the minimization program was looped using different scaling factors between 3.0 μN and 4.0 μN, the results were almost exactly as depicted in this figure. From figure 14, it can be inferred that a value of $F_0 = 3.3 \pm 0.1$ μN is the scaling factor most consistent with the experimental data since the scaling factor that results in the minimum error changes from 3.2 μN to 3.4 μN.

Discussion and Conclusion:

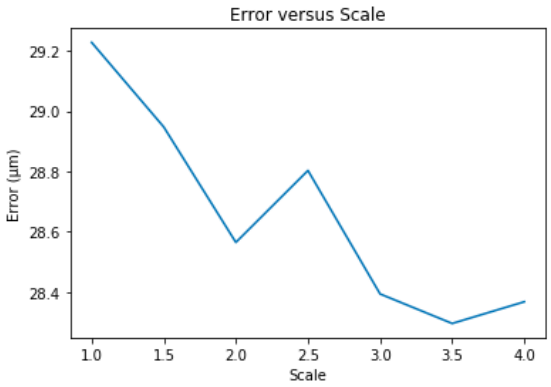

Figure 13. This figure shows the error in μm of the minimum geometry from the original geometry of the subset of data versus the scaling factor used for the simulation.

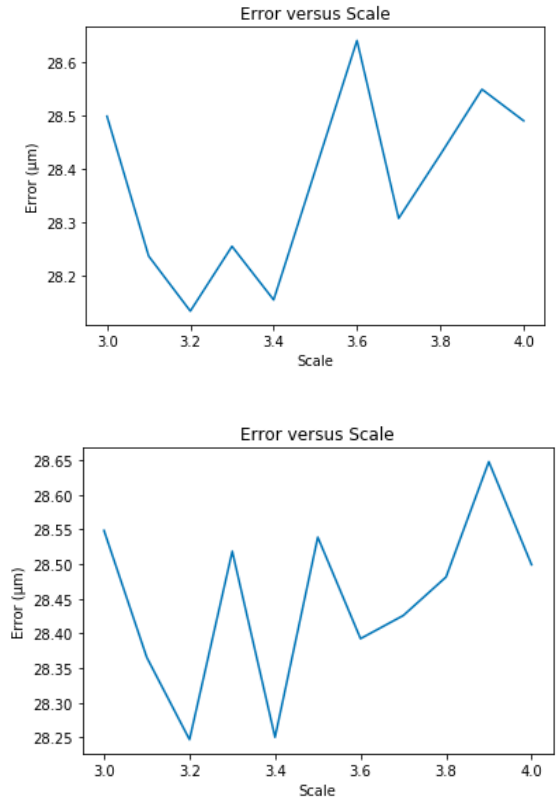

Figure 14. This figure shows the error versus scaling factor graph— where the error is the error in distance with units μm—after running the simulation on each of the varying scaling factor two times. The minimum values range between 3.2 μN and 3.4 μN. Therefore, the minimum scaling factor is equal to $F_0 = 3.3 \pm 0.1$ μN.

In summary, the experimental data from Janna Lowensohn and Eric Weeks was analyzed to determine the scaling factor and power law of the repulsive force expression for a quasi-two-dimensional system of jammed emulsions. A built-in SciPy minimization function was tested using the Coulomb force to determine whether it could be reliably used on the experimental data set. When it was determined that it was not reliable, I created a minimization program, and it was determined, after testing my created minimization program, that it was reliable enough to use on the experimental data. A subset of 32 out of 882 emulsion droplet data sets were used in the minimization process.

As reported above, $F_0 = 3.3 \pm 0.1$ and $\beta_f = 1.20 \pm 0.05$ are the scaling factor and power law found to be most consistent with the subset of experimental data. Figure 15 shows the original geometry and the minimized geometry of a simulation using $\beta_f = 1.20$ and $F_0 = 3.3\ \mu N$. In Janna Lowensohn's experiment, the droplets had a size ratio of 1.42. In determining the scaling factor and power law for a data set with the same size ratio, Desmond et al. found $F_0 = 2.4 \pm 0.1\ \mu N$ and $\beta_f = 1.19 \pm 0.02$ to be the scaling factor and power law most consistent with their data set. Therefore, the power law found in my project is consistent with the power law found by Desmond et al. The scaling force factor determined in my project is roughly 40% larger than what was determined by Desmond et al. If the full data set were used in my analysis, I do think that the scaling factor would also be consistent with literature. However, it is also important to remember that the data set used in my

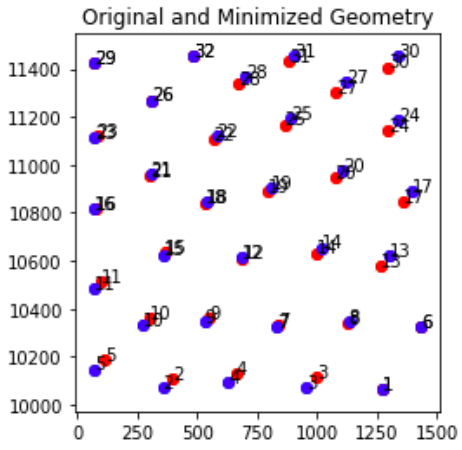

Figure 15. This figure shows the original subset of data (red particles) and the minimized subset of data (blue particles) using $\beta_f = 1.20$ and $F_0 = 3.3\ \mu N$. The error for the resulting geometry is 28.583 μm.

analysis was not the data set from Desmond et. al. Therefore, the different data sets could have slightly different parameters, such as the ratio of dish soap to water, that could've affected the values for the approximate repulsive force law. Looking at the error graphs, the error is around 28.0 μm to 30.0 μm. On average, the radii of each of the emulsion droplets are anywhere between around 90.0 μm to 125 μm. So, the average distance moved for each of the emulsion droplets is less than one-third to one-fourth of the average emulsion droplet radius. Therefore, I believe that my scaling factor and power law are good approximations for the overall scaling factor and power law of a quasi-two-dimensional system of jammed emulsion droplets.

References:

1. Goodarzi, F., & Zendehboudi, S. (2018). A comprehensive review on emulsions and emulsion stability in chemical and energy industries. *The Canadian Journal of Chemical Engineering, 97*(1), 281-309. doi:10.1002/cjce.23336
2. Loi, C. C., Eyres, G. T., & Birch, E. J. (2019). Protein-stabilised emulsions. *Encyclopedia of Food Chemistry*, 404-409.
3. Biroli, G. (2007). A new kind of phase transition? *Nature Physics, 3*(4), 222-223.
4. Desmond, K. W., Young, P. J., Chen, D., & Weeks, E. R. (2013). Experimental study of forces between quasi-two-dimensional emulsion droplets near jamming. *Soft Matter, 9*(12), 3424.