Proteins Under Pressure:

Simulating Membrane Proteins with Repulsive Interactions

Introduction/Motivation

For my Honor's Thesis, I chose to model diffusion and interactions of membrane proteins. This is a topic I've been interested in researching for many years, going back to my first undergraduate degree (Biology, UCSB) in 2011. I started working with Prof. Thomas a couple years back on this topic to get some research experience in Physics. My initial work modeled: a spherical surface on which particles were free to diffuse and interact. This was thought to be a more faithful representation of a cell membrane, but it turned out to be too computationally intensive. Thus, I changed to a flat, two dimensional model with periodic boundaries.

Cell membrane protein distribution and dynamics are critically important for cell signaling: many chemical signals are transmitted into cells through binding to cell surface receptors, followed by a redistribution of those receptors into clusters (either through direct crosslinking, i.e. one ligand (signaling chemical) binding to multiple receptors, or through the conformational change of receptors that lead to aggregation [1]. For nearly four decades, it has been known that repulsive interactions among cell membrane proteins can affect their distribution [2]. My modeling work focused on exploring the effects of a hypothetical repulsive force on the distribution of proteins in the membrane. The code I developed can easily be adapted for attractive interactions by simply changing the sign of the interaction.

My model includes random motion, which models diffusion, and an exponentially decaying repulsive force. Such a repulsive force is a good model for screened electrostatic interactions among like-charged particles. Membrane proteins are quite commonly glycosylated and thus carry negative charge. In addition, most animal cells live in a salt water environment. The dissociated ions in a salt solution screen the electric field, causing an exponential falloff in force.

The principal question I aimed to address is this: do proteins "crystallize" as the temperature is lowered (or equivalently, as the force is made stronger), and is there any evidence of a sudden phase transition? 2D phase transitions do occur with hard disks (which have purely repulsive forces), for example from fluid phase to hexatic phase at a

critical packing ratio. Additionally, we wished to determine if this system obeys the Ornstein-Zernicke (OZ) density relation [3][4], allowing for decomposition into direct and indirect correlations between particles from the radial distribution, g(r). We will see that the answer to the first of these inquiries was found with fair confidence, while the OZ density relation appears not to hold, but needs more work. (see Appendix A and B).

Proteins in the membrane tend to be negatively charged under physiological conditions due to glycosylation, the addition of carbohydrates to a molecule [5]. Since proteins are often added to the membrane by vesicles, inside of which are much higher concentrations of the protein, they often find themselves in crowded environments, experiencing mutual repulsion within this crowded environment [2]. Membrane protein distribution and diffusion dynamics are important to transmembrane signaling, especially when that signaling is mediated by protein clustering or aggregation.

Apart from academic interest in the topic, I wish to motivate my project by discussing what this type of research is used for. Proteins in cell membranes are involved in vast numbers of cellular processes, including trans-membrane transport of ions, vesicle formation and attachment, cytoskeleton structure, among many others [1]. Errors in these biochemical pathways can have severe consequences such as lethal diseases for an organism, including humans. These errors cause unimaginable pain for many people around the world, see Fig 1 [6]. Thus, it is vital for us to understand the details of the underlying behavior of membrane proteins to prevent and cure diseases which plague society at large.

<u>Model Specifics</u>

Proteins were modelled as point particles which undergo lateral, as implied by the 2-dimensional space, diffusion. Anything with a temperature must undergo some diffusive motion due to its non-zero thermal energy [7]. It is in addition to this diffusive process that we chose to add the interaction of repulsion due to presence of electrical charges on the proteins.

The electrical repulsion experienced by like-charged proteins in the membrane differs from repulsion experienced by like-charged particles in free space due to screening [8], see Fig 2. This screening is brought about by ions and aligned, polarized water molecules near the cell membrane. As a result, the force experienced has an exponential decay instead of an inverse square decay with increasing distance:

$[F_{repulsion} = F_0 \cdot e^{-\frac{r}{r_0}}]$ instead of $[F_{repulsion} = k \cdot \frac{q^2}{r^2}]$.

We chose to use a "soft" repulsion, without a hard disk repulsion (infinite at the particle's radius) or a "1/r hard" repulsion (F=$F_{repulsion}$*(1/r)), for a couple reasons. First, the membrane is roughly 20-30% proteins by area [1], which means that the likelihood of proteins getting close enough for the hard part of any repulsion come into effect. So physiologically, the soft repulsion is a reasonable approximation. Second, we wanted to investigate a specific aspect of behavior, namely long-ranged interactions, in order to elucidate information about that aspect specifically. If we mixed both the hard short-ranged repulsion and the soft long-ranged repulsion, we wouldn't know if the phase transition happened or did not happen due to the soft repulsion. In using the soft repulsion only, we know that whatever results we obtain are due exclusively to the soft repulsion.

Combining the motions due to repulsive electrical forces and random thermal energy gives the total motion of the protein (which will now be referred to only as the particle) to be simulated. All simulations were implemented on Matlab versions 2022a and b [9], and all data was gathered from UNM CARC. All functions referred to below correspond to the functions in Matlab.

Simulation/Code

The simulation was a long series of iterations, each corresponding to the passage of 1 unit of time. During each iteration, a diffusive component (hop) was added via vector addition to a repulsive hop in 2 dimensions. Each simulation had constant parameters set for the number of particles, temperature, repulsive force range, and number of iterations. Thus, each iteration updated a matrix of x- and a matrix of y-coordinates the size of the number of particles to include the distance contribution from repulsive and diffusive motions. A simulation ran until equilibrium was reached indicating the "completion" of the simulation. A typical number of iterations per simulation was 15,000,000, depending on the parameters chosen.

In order to allow for an indefinite extension of the simulation, the "save" and "load" functions were used to transfer initial x- and y-coordinates along with each parameter from the initialization code to the main code, and from the main code to itself in the outer loop (discussed below) of the program. Initial coordinates needed to be random, so 2 random number matrices were generated with the dimensions of

number of particles by number of particles (square matrices). With the circular boundary we were forced to use in lieu of the square boundary (see Boundary Conditions section), the coordinates' position had to be within a radius of 1 from the origin. Thus, a while-loop was imposed to select only x-and y-coordinates whose square root of the sum of their squares (distance from origin) was less than 1. Since the maximum number of particles needed was 400, the while-loop was not time consuming. This was of concern since while-loops can often lead to excess run times.

The circular boundary created an additional challenge in computing the normalized correlation function g(r), This correlation function represents the probability, per unit area, of finding a second particle at a distance between r and r + dr from any given particle. The correlation function is calculated from a histogram of particle-particle distances. With a circular boundary and only quasiperiodic (rather than truly periodic) boundary conditions, the area in which particles may be found varies with r. Thus, part of the initialization process was to compute the allowed area at every particle separation r, so that the correlation function could be properly normalized.

To calculate this area-normalization matrix, the law of cosines was invoked to find the angle which defines the area for every distance from each possible particle location. Thus, if a particle is located near the boundary, and the distance to another particle is greater than its distance from the origin, the area this particle could be in is reduced from the full circumference of 2π to a smaller angle θ . This area normalization matrix, once calculated, was passed to the main code through a "save" and a "load" command, where it was used repeatedly, requiring no recalculation.

The main code was organized into 3 nested loops: Inner, Middle, and Outer. The inner loop calculated each diffusive and repulsive hop, then added them to the current x- and y- coordinates. The diffusive hop was calculated simply from 2 random numbers, one for the x-coordinate hop and one for the y-coordinate hop. The repulsive hop required calculation of distance from every other particle, then calculation of force using those distances, then summation of the forces in each component, resulting in a net x and a net y displacement just as in the diffusive hop. The original coordinates then had each hop added to them, and the inner loop passed the new coordinates back out. New coordinates were only passed out to the middle loop once every thousand inner loop iterations.

The middle loop took these updated coordinates every 1000 iterations and used the "histcounts" function to count each distance and record them in a histogram. It is worth noting that this built-in function saved a lot of run time for the program overall. The cumulative histogram was created by adding distances from each new coordinate to the existing histogram. 3000 iterations of the middle loop completed this cumulative histogram, which was then passed to the outer loop.

The outer loop normalized the cumulative histogram to a radial distribution, g(r), by dividing by the total number of histograms added into the cumulative histogram, or the number of middle loop iterations. From comparison of consecutive g(r)'s, it was determined whether the simulation was complete, i.e. had reached equilibrium. Though seemingly complicated, the program was remarkably simple once complete, and simulation run time was not a limiting factor for the project.

<u>Boundary Conditions</u>

We initially set out with periodic square boundary conditions, defining the space for our particles to move in as the unit square. To be periodic, the particles were made to reappear on the opposite side of the square instead of leaving the area. In this way, a particle that meanders too far along the y-axis, appears at the bottom of the square if its y-coordinate goes over 1, and similarly for the x-coordinate going over 1. Also, if the particle's coordinate becomes less than 0, an amount of 1 is added to it and the particles reappear at the top or right side of the unit square. This keeps the average density constant throughout the simulation and prevents the need for a source of particles as particles leave the area. These boundaries appeared to work fine, and accordingly we worked with them for the whole fall semester. That is, until an error in lattice formation became apparent.

Lattices were forming at lower temperatures, as expected, however, every lattice was aligned with the boundary edges. This could not have been physically correct, as any real formation of a lattice must occur in random directions if the simulation is correct. After about a week of thinking, I was able to discover the reason for this alignment, and that it required the establishment of new boundary conditions were we to move forward with project.

In using a radially directed force, as is required for electrostatic repulsion, the range of the force is infinite. Since infinity extends beyond our boundaries (it doesn't loop around on itself as it would on a sphere or other closed surfaces), and the

boundaries are not radially symmetric, there are directions from the center of the square which contain more particles than other directions. Specifically, the direction to a corner has $\sqrt{2}$ more distance to the edge than the direction to the perpendicular edge of the unit square, see Fig 3. This means that more particles on average will exert a repulsion on a particle at the center of the square. Due to our use of image particles in force calculations (see Image Particles section), every repulsion was calculated with the particle undergoing repulsion located at the center of a square of real and image particles. In other words, each iteration encountered this imbalance in force from the corners and perpendicular edges. The imbalance creates a larger net repulsion from the corners than from the perpendicular edges, canceling out forces from the corners, leaving greater net forces in the directions of the perpendicular edges. To see this alignment occur at zero temperature with 100 particles, see attached video.

The issue boiled down to this: our boundary conditions had different symmetry than our force. Our force was radial, and our square was 2 perpendicular edges. This mismatch led to an anisotropic force for each particle on average, which led inevitably to alignment of lattices with the boundary. One way to ensure this was indeed the problem, and the next step we took, was to truncate the force so that it did not extend beyond the shortest distance to the edge of the boundary, the perpendicular edge. This ensured an isotropic force distribution on each particle. As soon as the range was reduced to be entirely within the square boundary, a range of exactly 0.5, any and all lattice formation immediately ceased. Even when the range of force was slightly greater than 0.5, lattice formation occurred. Even a slight imbalance between the boundaries and force led immediately to lattice formation. Due to this sensitivity, I suspect the lattice formation itself was initiated by this boundary/force mismatch.

To try to avoid this artifact, we designed a new model system. The new boundary was set to be the unit circle, with a different image particle scheme (see Image Particle section) with quasi-periodicity. Particles which hopped outside of the unit circle boundary reappeared immediately on the opposite side of the circle. Each particle had an image outside the unit circle: images are necessary to prevent particles near the periphery from feeling strong (artifactual) repulsion away from the center.

Of course, with a circular geometry, no alignment of lattices was possible. With this new model we were able to successfully gather the data we needed in order to test for a phase transition and for compliance with the OZ density relation.

An interesting occurrence arose with the usage of image particles in this way. Because the real particles did not apply a force back on the image particles, Newton's 3rd Law did not hold for our system. This resulted, under some conditions, in an overall rotation of the system which seemed to come from nowhere. It is of course a consequence of Newton's 3rd Law that angular momentum is conserved. The system therefore showed this lack of conservation of angular momentum. This did not affect the validity of the system, as rotation does not affect the distances between particles. Net force in the radial directions cancelled out on average, meaning that linear momentum was still conserved.

<div align="center">Image Particles</div>

In order to calculate the repulsive force correctly in a periodic or quasi-periodic system, one must use image particles in addition to, or in place of, the real particles. For the square boundary, our condition to use an image particle in place of its real particle for force calculation was whether its distance in either the x- or y-direction was greater than 0.5. If this was the case, an image particle with coordinates of plus or minus 1 was used. This placed all image particles outside the boundary and had the effect of placing the particle for which repulsion was calculated at the center of a square of real and image particles. This maintains the boundary around each particle for every iteration.

The circular boundary had a different usage of image particles for force calculation. Instead of only using image particles when a particle's distance is greater than 0.5 in either the x- or y-direction, image particles of every other particle were used without modification of coordinates for every force calculation. Image particles were defined to be a distance of 2 units away from their real particles and located directly across the origin or center of the circle. Every image particle except its own image was used to calculate force in every iteration. Since these conditions are not fully periodic, we called these boundary conditions quasi-periodic. It is worth mentioning that the square boundaries we initially used were not fully periodic either, because that would require infinitely many image particles over every square patch out to infinity in x and y. However, we did not know how to calculate these conditions, so we used "partially periodic" conditions on the unit square.

<div align="center">Dimensionless Temperature</div>

A dimensionless temperature was derived as follows: Using 2 equations for the diffusion coefficient

(1) $D = \frac{\delta^2}{2\tau}$ [7], and the 2-dimensional form (2) $D = \frac{kT}{4\pi\eta R}$,

we see that temperature (T) is proportional to hop size squared (δ^2): $T \alpha \delta^2$. Now by taking the ratio of our 2 different hop types (thermal over forced), we arrive at a dimensionless temperature: $\frac{T}{T_0} = \frac{\delta^2_{thermal}}{\delta^2_{forced}}$. It is this temperature which we plotted our data with.

Notice for a constant force hop size (same F_0 value in the force equation in the Background section above), there is a maximum temperature we can simulate. This is due to the average interparticle separation limiting how large diffusive hops can be. Average interparticle separation (IPS) is given by the square root of the area divided by the square root of the number of particles: $IPS = \sqrt{\left(\frac{Bounding\ Area}{Number\ of\ Particles}\right)}$. Diffusive motion is seen as hops only when the time increment is large enough between observations of the particle [7]. Otherwise continuous random paths are traced out as the particles "roll" past one another. Thus, using the hopping modelling of thermal motion limits us to not allow particles to hop over one another. If your model does this, it is non-physical. This requires the maximum diffusive hop size to be one half of the IPS so that diffusive motion cannot cause particles to hop over one another. As a consequence of this limitation, our dimensionless temperature has a maximum value (with constant F_0) of 4. This dimensionless temperature could go to infinity so long as we make the force magnitude smaller and smaller. However, we were investigating a constant force magnitude to be consistent across varying densities.

Long and Short Ranged Forces

In order to ensure there was no error in the simulations, it was important to test 2 different sizes of a parameter to compare with each other. If they showed different behavior, beyond what would be expected by the change, then we know there is a systematic error somewhere in the code. However, if they behave approximately similarly, then we have verified our simulation runs without systematic errors.

For this purpose, we tested 2 values for r_0 in the repulsive force equation (see Background section), with one being double the value of the other. The larger value of r_0 gives a steeper decay rate and thus a short range of repulsion, while the smaller value

corresponds to a long-range of repulsion. This is important to explain for interpretation of the results below.

<u>Radial Distribution, g(r)/Equilibrium Parameters:</u>

As mentioned above, the correlation function g(r) (or radial distribution function) represents the normalized probability of finding a particle between a distance r and r + dr away from any given particle. "Normalized" means per unit area, (which is of course larger for larger r) and approaching 1 at large distances, where particle positions are uncorrelated, If particles acted completely independently, g(r)=1, as we see when we remove the repulsive force in our simulations, leaving only thermal diffusive particles to trace out the area of the boundary, acting completely independently of each other. In this sense, g(r) accounts for correlations between particles, hence the name "correlation function" [10].

At high temperatures, particles tend toward randomness/non-interaction at larger distances from the particle. This can be seen as the flat region in Fig 4. When temperature is lowered, more ordering between particles occurs as mentioned in the introduction above. This alters g(r) by introducing density oscillations to replace the flat region seen at high temperatures, see Fig 5. It is insightful to match the g(r)'s with their particle maps from their final iteration, see Fig 6. The trough at low distance values, which we will call the depletion increases in size as temperature decreases. Height of the first oscillation peak increases as temperature decreases. It is for this reason that we chose these 2 parameters, "Depletion" and "First Peak Height", as quantities which we use to determine equilibrium, and as measurements to graph with varying temperature to determine presence of a phase transition. Depletion was defined as the horizontal distance until a value 5% of the first peak height was reached on g(r). Approximately this is just the distance to where the first peak begins. First Peak Height was defined as the maximum height reached by the first (shortest distance) oscillation on g(r), see Fig 7. Equilibrium was defined to be a lack of significant change in either of these parameters after each outer loop iteration, or each g(r).

An interesting problem arises when low enough temperatures are reached. In particular at T=0, the thermal hop size is 0, thus only repulsive, non-random hops occur in the simulation. Without the random motion of diffusion, particles tend to stick to an exact location once a suitable lattice location is reached. In addition, lattice formation occurs very quickly without random motion to interfere with a particle's path toward its

lattice location.   Because of the lack of random motion and exact locations of particles, the peaks in g(r) approach delta functions as they transition from smooth, symmetric peaks to spikes with little symmetry. This results in the first peak resolving into several component peaks with different heights than the smooth peak would have, see Fig 8. This means the first peak height loses its meaning as a parameter at temperatures approaching zero. Therefore, we left the zero-temperature data point off the graph, since it did not fit well with the rest of the data. This freed us to use a log temperature plot which makes it much more readable.

Yet another issue with our parameters arises at the high temperature limit. In this limit, depletion becomes zero for any concentration if the temperature is high enough. This is due to the lack of detailed information about the g(r) curve obtained at zero depletion. For example, since the first distance measurement needs only be above 5% of the maximum first peak height, it could be at 10% of the height or at 50% and both would simply give zero depletion. We will see this limitation occur in the short-range force depletion plots below.

Results

With these definitions in mind, we can discuss the results, see Figs 9 and 10. We see that for the long-range force, depletion increases with $\log(T/T_0)$ at a decreasing rate. Apart from some minor wiggles in the curves, no large vertical displacements nor sudden changes in slope are seen. This means there is no evidence for either a 1st or 2nd order phase transition, respectively, from the depletion parameter.

Next, the first peak height parameter increases with an increasing rate as temperature decreases logarithmically. The plot shows less wiggles than the depletion parameter, however, there is a crossing over the other 3 curves by the 400-particle curve, and the 200-particle curve crosses over 1 other curve. This must mean that there is an increasing amount of bunching with lower temperature as the concentration increases. As the depletion does not exhibit this type of behavior, the separation between each particle stays the same across concentrations, but the bunching of particles at this distance increases with increasing concentration. This could be explained by less spread over the first density oscillation corresponding to the particles approaching a specific distance from each other more closely than at lower concentrations. In other words, particles at higher concentrations approach a circle

around the central particle more closely. This could be explained by the presence of more repulsive forces directing the particles to a more precise location.

At short-range of repulsion, the curves follow the same general form for each parameter as its long-range counterpart. Depletion curves all intersect at zero at the highest temperature. This is an expected behavior, because a depletion of zero can have differing g(r)'s, as described above. While this makes the system seem more well behaved due to the gentle appearance of the concentration curves, it in fact reveals a lack of information at higher temperatures, revealing a flaw of depletion as a parameter.

The first peak height plot for short-range force shows a similar trend as the long-range counterpart. You can see the 400 and 200 curves approach crossing over the other curves, however no crossing of curves is observed. This means that the bunching effect of more precise locations is more apparent with long-range repulsion, giving evidence for the larger amount of repulsive forces leading to this increase in bunching at the first peak.

For discussion on the results of the analysis on the OZ density relation, see Appendix A and B.

<div align="center">Conclusions</div>

To conclude, I've learned a great deal from this year's Honor's project. The importance of optimizing code to reduce run time and how run time can be a limiting factor on any project using simulations. The usefulness of seemingly over-simplified systems for the understanding of the underlying nature of your system. And the importance of having a good plan to set out with, combined with the ability to adapt and problem solve as we had to do with our boundary conditions.

With no evidence of a phase transition under decreasing temperature, we can say with fair confidence that no phase transition occurs for point-like particles under purely repulsive forces on 2-dimensional spaces with quasi-periodic boundary conditions.

Thank you for your time. I hope you enjoyed reading this and learned something useful or interesting.

Appendix A: OZ Derivation

Ornstein-Zernicke Density Relation Derivation [4]:

(1) Assume there is a particle at position r_1

(2) The direct correlation effect modifies the density at r_2 by an amount ρc_{12}

(3) The density at r_3 is modified as well, by the amount ρc_{13}

(4) *This* fluctuation at r_3 has an additional effect on the density at r_2, with magnitude $\rho c_{13} \rho c_{32}$

(5) Particle 3 can be anywhere, and after integrating over r_3 and cancelling one power of ρ, one obtains

$$h_{12} \approx c_{12} + \rho \int dr_3\, c_{13} c_{32}$$

(6) The entire argument can be repeated with more intermediate particles, giving the infinite series

$$h_{12} = c_{12} + \rho \int dr_3\, c_{13} c_{32} + \rho^2 \int dr_3\, c_{13} \int dr_4\, c_{34} c_{42} + \cdots$$

(7) And finally, this series can be summed by a simple substitution to give the OZ equation.

$$h_{12} = c_{12} + \rho \int dr_3\, c_{13} h_{32}$$

OZ Equation

Appendix B: OZ Analysis

As stated in the Introduction section, we attempted to see if our g(r)'s indicated our system of particles follows the OZ density relation. We used code from Chouinard and Baddour to attempt to perform a Discrete Hankel Transform [11] [12]. We were unable to find a suitable direct correlation function (c_{12} in Appendix A) likely due to the potential from our force monotonically decreasing. Since there is no local minimum in the potential, as there are in other potentials such as the Leonard-Jones 6-12 potential, there is no way to get the direct correlation function, as no pairing between particles can occur without this minimum for a neighboring particle to "sit in". Therefore, we decided to include this in an appendix instead of the main body of the paper, as it was inconclusive and uninformative.

Figures

Figure 1:

The CFTR protein is a channel protein that controls the flow of H_2O and Cl^- ions in and out of cells inside the lungs. When the CFTR protein is working correctly, ions freely flow in and out of the cells. However, when the CFTR protein is malfunctioning, these ions cannot flow out of the cell due to a blocked channel. This causes cystic fibrosis, characterized by the buildup of thick mucus in the lungs.

[6]: Shown is a cartoon diagram of an error in a trans-membrane protein which leads directly to Cystic Fibrosis, a disease which causes suffering for thousands worldwide.

Figure 2:
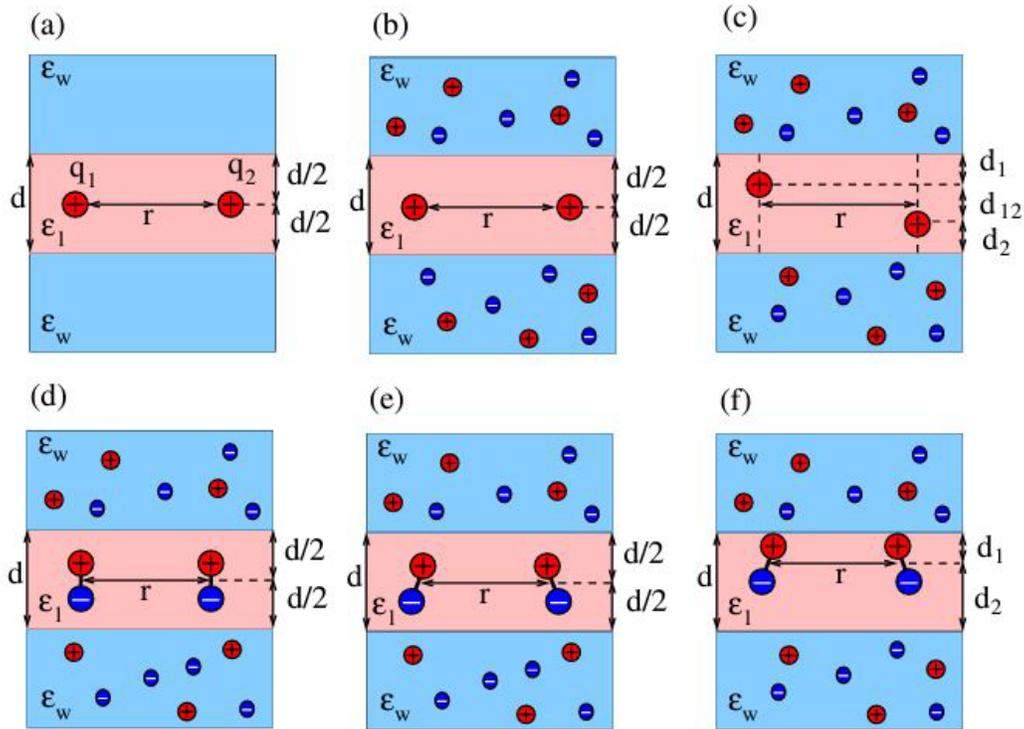
Figure 1. (a) Two interacting point charges, q_1 and q_2, separated by a distance r in the middle of a dielectric slab of dielectric constant ϵ_l and thickness d. The dielectric constant of the sandwiching media is ϵ_w. (b) Salt ions with bulk concentration n_0 are present in the two sandwiching media. (c) The two interacting charges are moved up or down so that their mutual distance is $\sqrt{d_{12}^2 + r^2}$ and their distances to the dielectric interfaces are d_1 and d_2, with $d_1 + d_2 + d_{12} = d$. (d) The two charges in diagram b are replaced by two dipoles, both located at the middle of and oriented normal to the dielectric slab, either parallel (as shown) or anti-parallel (not shown). (e) Two dipoles as in diagram d, yet with arbitrary orientations. (f) The two interacting dipoles shown in diagram e are jointly moved up or down so that their distances to the dielectric interfaces are d_1 and d_2, with $d_1 + d_2 = d$.

[8]: Shown are varying arrangements of charges within the membrane both in the presence of ionic salt nearby, and without ions.

Figure 3:
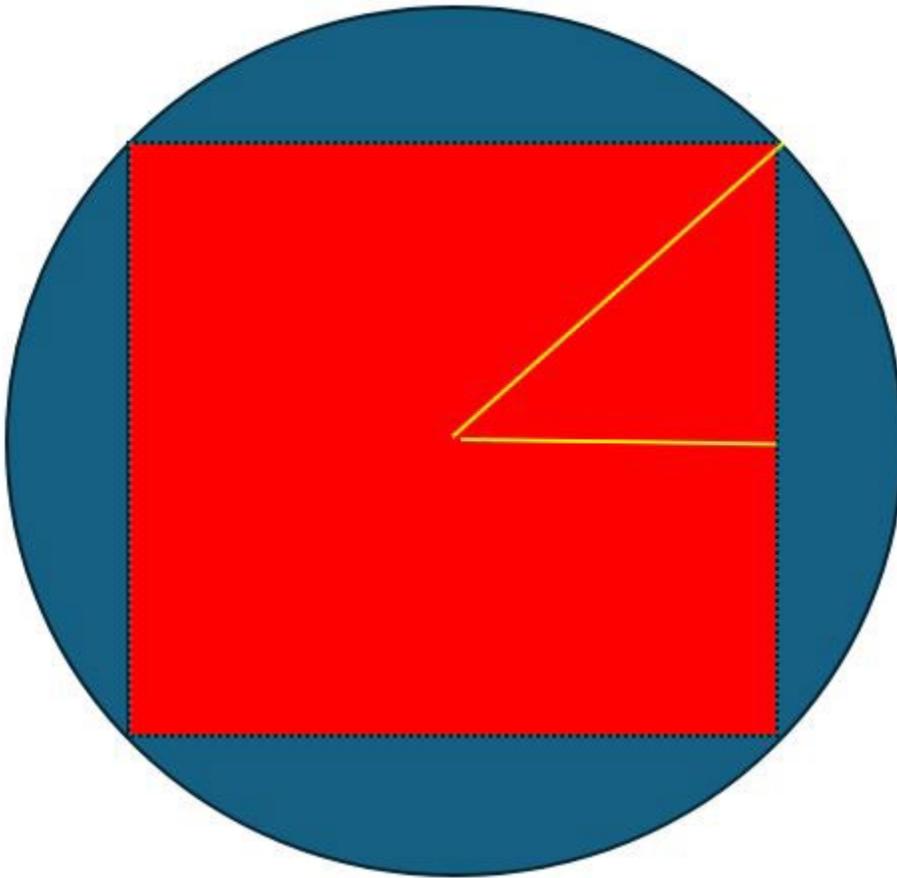
Square Boundary in red, Radial force in blue. Notice the difference in length to the corner compared to the perpendicular edge. It is easy to see from this why more repulsion comes from the corners than from perpendicular edges.

Figure 4:
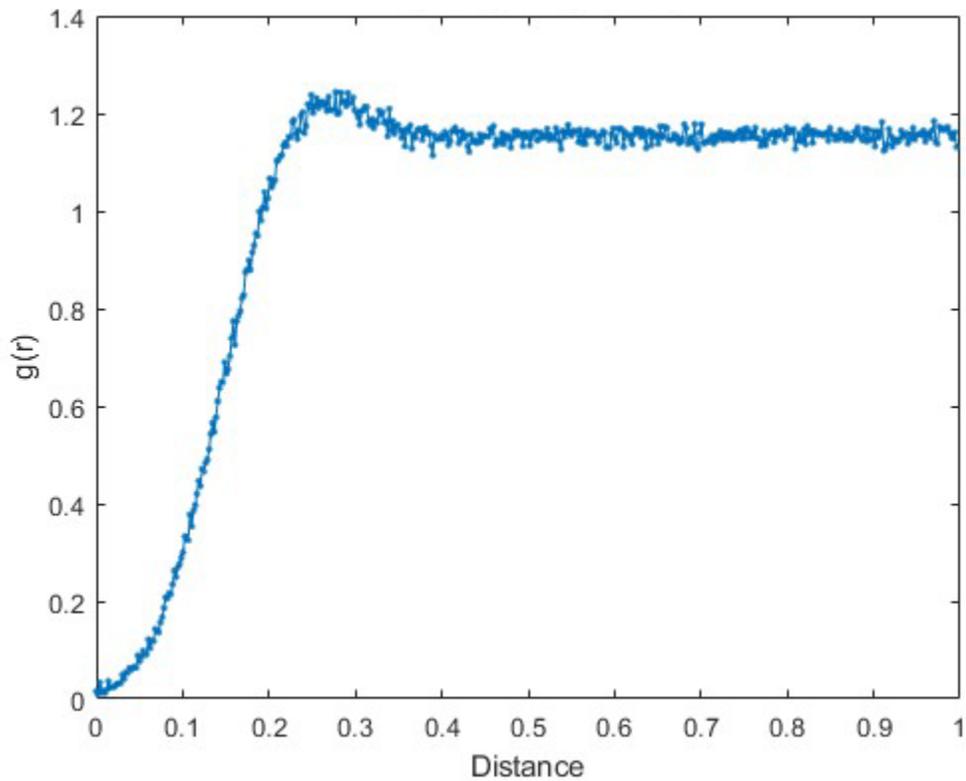
Notice the flat portion of the curve at large distances from the particle. This can be interpreted as independent particle behavior and random chance of finding a particle past the short peak at about 0.2. This g(r) comes from a 50 particle simulation with the maximum dimensionless temperature of 4.

Figure 5:
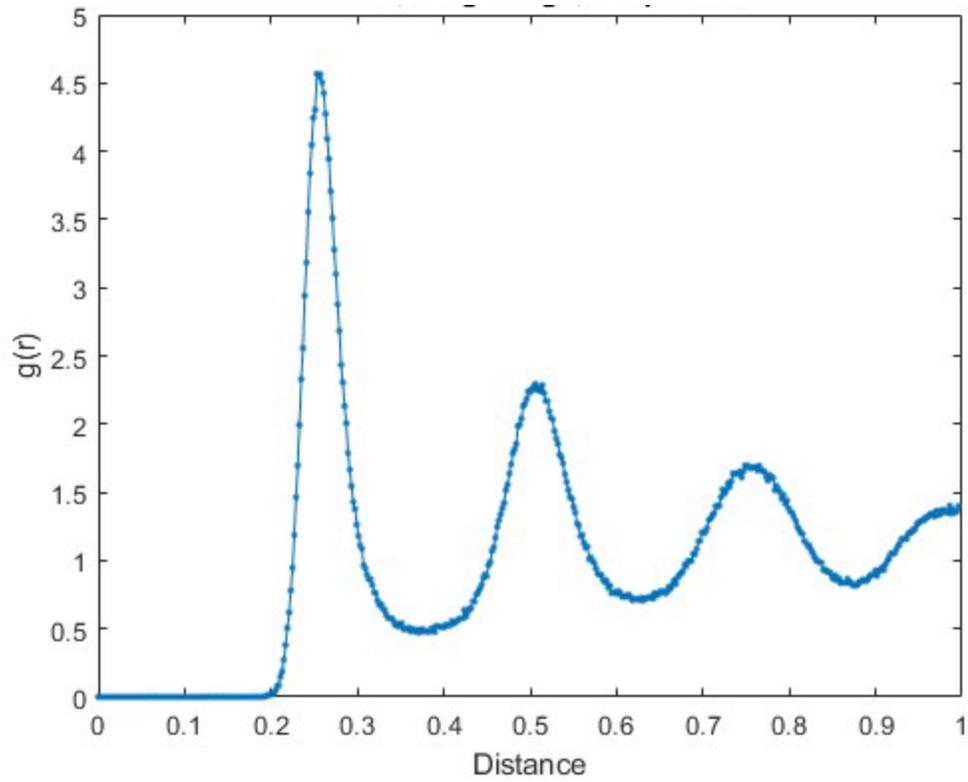
Notice the smooth oscillations which oscillate roughly around 1. The increase in order is due to the decrease in temperature. This g(r) has the same parameters as the previous figure, but at a dimensionless temperature of ~0.05, instead of 4.

Figure 6:
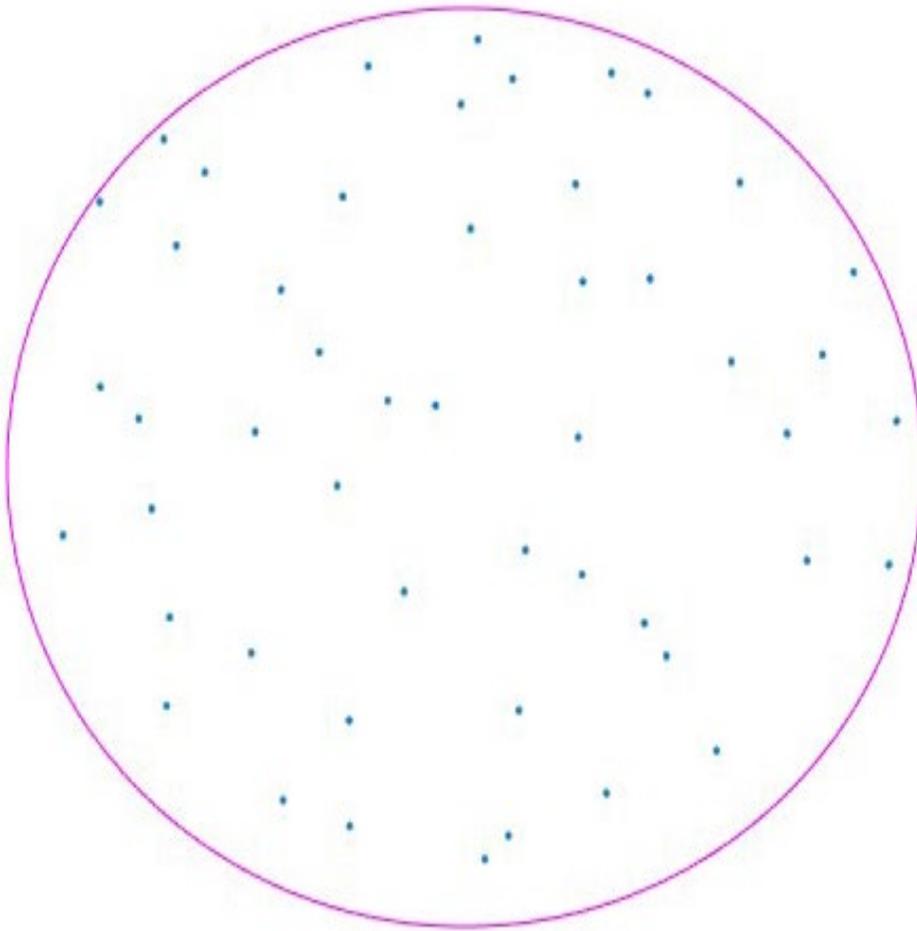
Figure 6a: Particle map corresponding to figure 4. Notice particles are not on top of each other, but there is not much order overall. Compare with figure 4 to see what the small peak and flat region translate into.
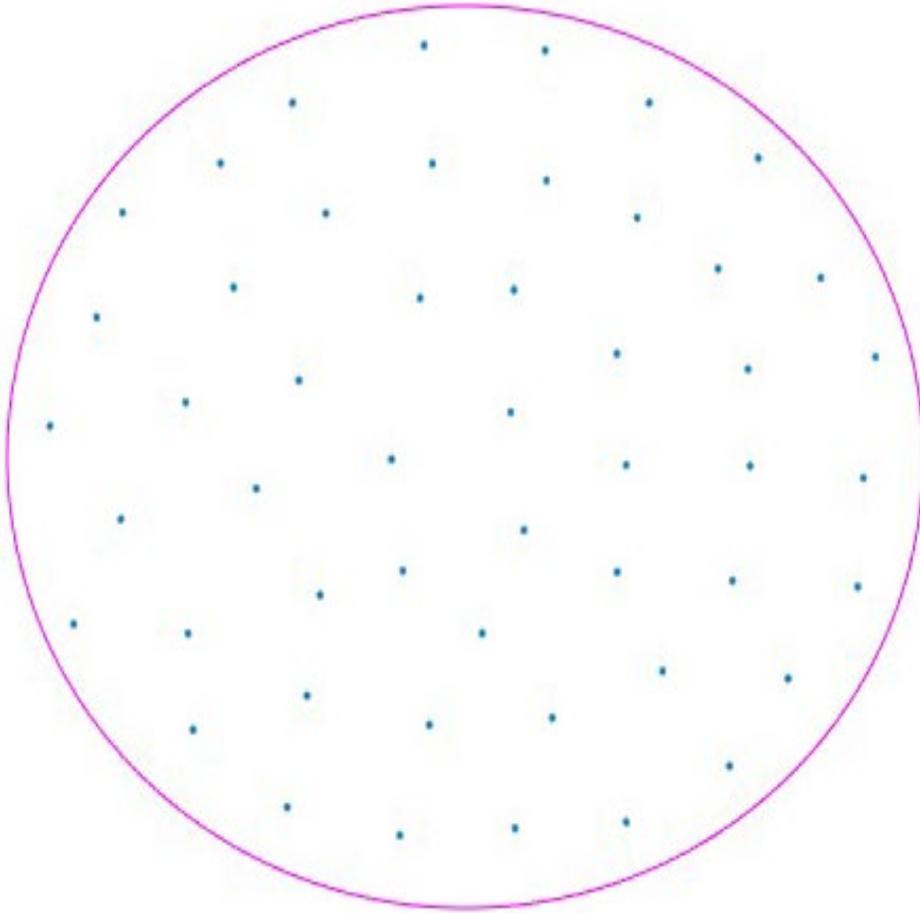
Figure 6b: Particle map corresponding to figure 5. Notice all particles have a certain amount of space around themselves, and there is clearly global ordering. Compare with figure 5 to see what the large depletion and smooth oscillations translate into.

Figure 7:

First Peak Height (vertical extent) tells us
how "crowded" each particle is

NP=50,Range=High,Temp=0.25
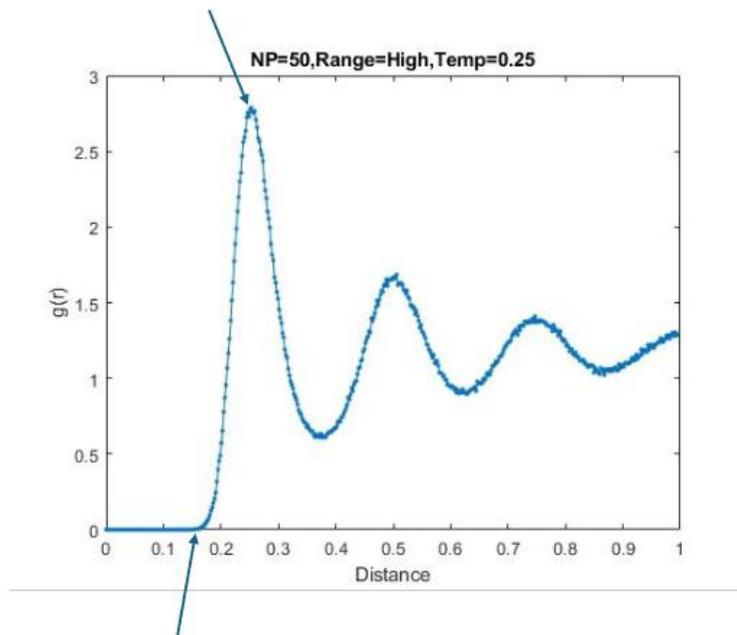
Depletion (horizontal extent) tells us how
much space a particle has around it until it
starts getting "crowded"

Explanation of 2 parameters used to determine equilibrium and analyze presence of a phase change.

Figure 8:
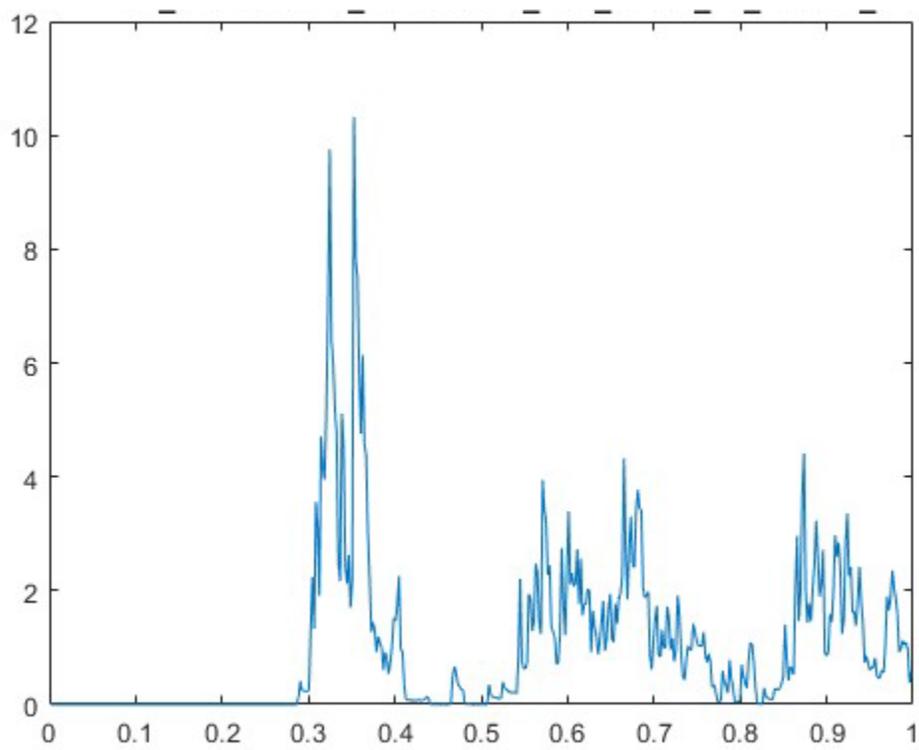
Zero Temperature g(r): Using 30 particles and zero thermal energy, we see smooth oscillations replaced by spikey component peaks which cannot be used for the First Peak Height parameter.

Figure 9:
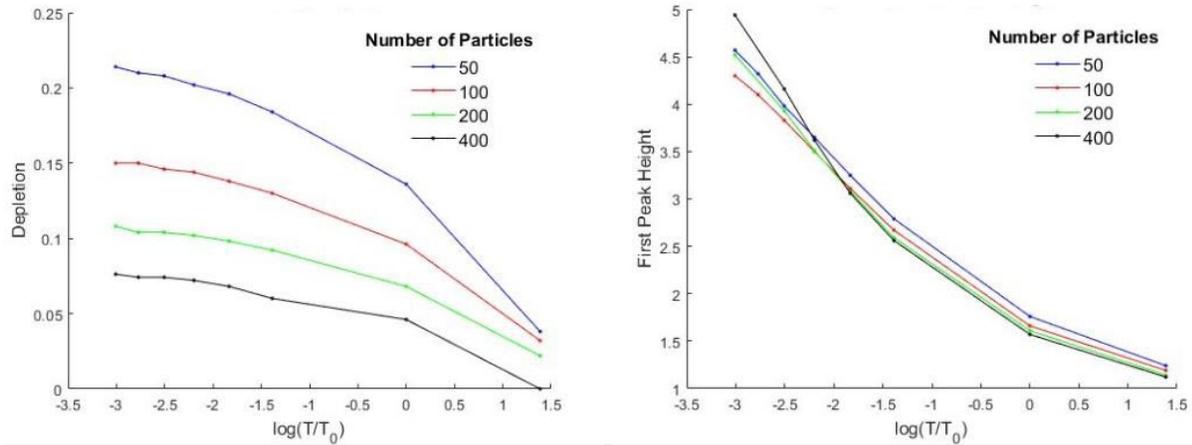
Long-Range: Depletion and First Peak Height parameters plotted over a range of temperatures. Horizontal axis is log(T/T$_0$). Individual curves represent differing concentrations. Notice the similar form across concentrations for both parameters. Smooth functions, including no sudden change in slope of vertical jumps indicate no presence of a phase change as temperature is lowered.

Figure 10:
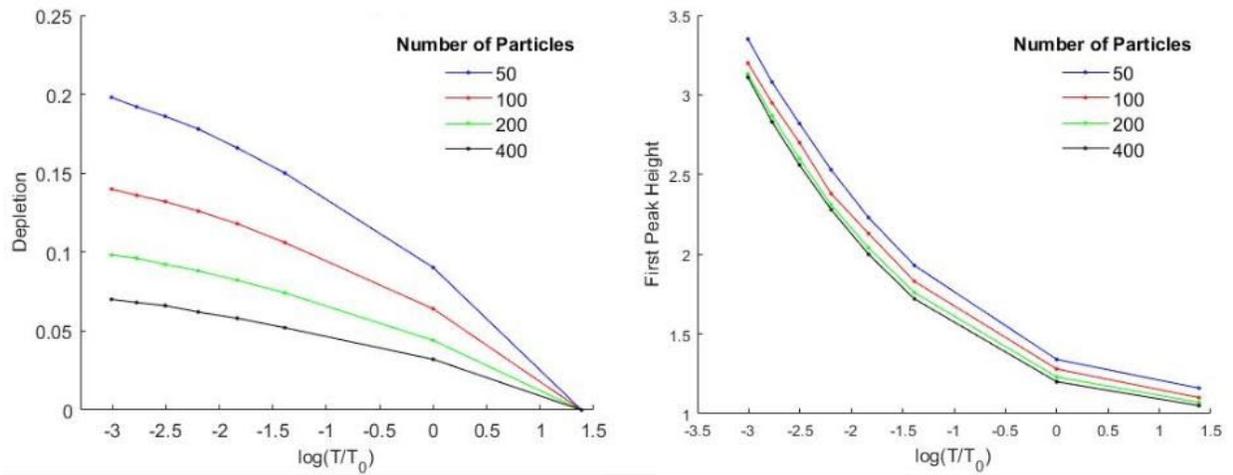
Short-Range: Notice the similar form across concentrations for both parameters. Smooth functions, including no sudden change in slope of vertical jumps indicate no presence of a phase change as temperature is lowered. For the short-ranged force, no curves cross each other, and the curves vary almost identically across differing concentrations. Again, no evidence of phase change is present in either parameter.

<u>References</u>

[1] Nelson, David L., et al. *Lehninger Principles of Biochemistry*. W.H. Freeman, 2009.

[2] Ryan, T. A., et al. "Molecular crowding on the cell surface." Science, vol. 239, no. 4835, Jan. 1988, pp. 61–64, https://doi.org/10.1126/science.2962287.

[3] Ornstein, L.S.; Zernike, F. (1914). "Accidental deviations of density and opalescence at the critical point of a single substance" (PDF). Proceedings of the Royal Netherlands Academy of Arts and Sciences. 17: 793–806. Bibcode:1914KNAB...17..793. Archived from the original(PDF) on 2021-02-06. – Archived 24 Sep 2010 at the 'Digital Library' of the Dutch History of Science Web Center.

[4] Case, David, and Frederick R. Manby. "The Ornstein Zernike equation in molecular electronic structure theory." Molecular Physics, vol. 108, no. 3–4, 10 Feb. 2010, pp. 307–314, https://doi.org/10.1080/00268970903446772.

[5] Scullion, B.F., Effects Of Mutations in 3 Domains Of The Vesicular Stomatitis Viral Glycoprotein On Its Lateral Diffusion In The Plasma-Membrane, Journal Of Cell Biology 105: 69 (1987).

[6] "Cystic Fibrosis." *Wikipedia*, Wikimedia Foundation, 15 Apr. 2024, en.wikipedia.org/wiki/Cystic_fibrosis.

[7] Bakunin, Oleg G. Turbulence and Diffusion: Scaling versus Equations. Springer Berlin Heidelberg, 2008.

[8] Bossa, Guilherme Volpe, and Sylvio May. "Integral representation of electrostatic interactions inside a lipid membrane." Molecules, vol. 25, no. 17, 22 Aug. 2020, p. 3824, https://doi.org/10.3390/molecules25173824.

[9] 'Matlab' (2022b). Natick, MA: MathWorks.

Supplemental: MATLAB Version: 9.13.0.2049777 (R2022b), MATLAB License Number: 707564, Operating System: Microsoft Windows 10 Home Version 10.0 (Build 19045), Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode

[10] McQuarrie, Donald A. *Statistical Mechanics*. Viva Books Private Limited, 2013.

[11] Baddour, Natalie, and Ugo Chouinard. "Theory and operational rules for the discrete hankel transform." Journal of the Optical Society of America A, vol. 32, no. 4, 20 Mar. 2015, p. 611, https://doi.org/10.1364/josaa.32.000611.

[12] Chouinard, Ugo, and Natalie Baddour. MATLAB Code for the Discrete Hankel Transform, 2 July 2016, https://doi.org/10.7287/peerj.preprints.2216.

[13] Bernard, Etienne P., and Werner Krauth. "Two-step melting in two dimensions: First-order liquid-hexatic transition." Physical Review Letters, vol. 107, no. 15, 7 Oct. 2011, https://doi.org/10.1103/physrevlett.107.155704.

[14] E. P. Bernard and W. Krauth "Two-Step Melting in Two Dimensions: First-Order Liquid-Hexatic Transition", Physical Review Letters 107 155704 (2011)

[15] "Hard Disk Model." Hard Disk Model Page on SklogWiki – a Wiki for Statistical Mechanics and Thermodynamics, www.sklogwiki.org/SklogWiki/index.php/Hard_disk_model. Accessed 8 Apr. 2024.