

Numerical Simulations of Dust Particles in an Ambient Magnetic Field

LUIS LAZCANO, MASARU NAKANOTANI, and GARY P. ZANK

Dust particles have a ubiquitous presence in space, being found in interplanetary and interstellar space. Dust is found in nebulae which are giant clouds of dust and plasma. Dust particles in the plasma can couple with the plasma. This coupling of dust modifies the linear dispersion relations of waves within the plasma. Furthermore, charged dust particles are subjected to the Lorentz force from the ambient magnetic field. In addition, they experience self-gravitation due to the mass of the particles within them. This gravitation leads to instabilities such as Jeans instability [Pandey et al. 1994]. Jeans instability is caused when the dust pressure is unable to counteract the gravitational forces and the particles start to self-contract. In the project, we show the linear to non-linear evolution of the dust particles over time with a background magnetic field using the Particle-In-Cell (PIC) simulation method. The gravitational potential is solved by the Poisson equation. The interaction with the background magnetic field is taken into account in the calculation of the Lorentz force. We compare the simulation results with the linear dispersion relation of Jeans instability and discuss how the background magnetic field affects the linear and non-linear evolution of dust particles.

1 INTRODUCTION

1.1 Dust

Dust is a substance that is rarely in short supply. In space dust particles are found in interplanetary to interstellar space. This dust can create varying types of structures. These structures can range in scale from the tails of comets which occur from the interactions of the solar wind and the comet. Dust is in planetary rings such as Saturn's rings containing dust particles ranging from sub-micron to micron-sized particles. On the supermassive scale, dust is present in nebulae, massive objects composed of dust and plasma light-years in scale.

1.2 Plasma

Just like dust particles, plasma is of great abundance in space. Plasma is a fundamental state of matter where gas is heated to the point that its electrons are ripped from it creating a quasi-neutral sea of free electrons and ions. Due to the nature of plasma and its free electrons plasma can be manipulated by a magnetic or electric field. Due to plasma's varying properties, it is found in a variety of conditions, plasma is found naturally in many forms such as lightning and the aurora. In addition, it is found in man-made forms and used in things such as neon lights. In space, it is incredibly abundant, being part of stars and nebulae. For example, the sun is hot enough to produce plasma and expels it as solar winds.

1.3 Dusty Plasma

Within objects such as nebulae, plasma is often found with dust. Due to plasma having free electrons, those electrons can bind themselves to the dust particles. Binding electrons to the dust particles cause the dust to gain a negative charge allowing it to interact with the plasma. This coupling of dust and plasma (dusty plasma) modifies the linear dispersion of waves within the plasma. Furthermore, these

charged dust particles are subjected to the ambient magnetic field. In addition, Dust particles due to their mass experience self-gravitation.

1.4 Instability

The dust particle's ability to experience self-gravitation leads to the appearance of instabilities. One of these such instabilities is that of Jeans instability. Jeans instability occurs when the pressure of the dust is incapable of balancing the dust's gravity. When the system becomes unstable the dust particles begin to experience self-contraction. This self-contraction from the gravitational force of the dust particles is what is believed to be the driving force behind the collapse of astrophysical objects.[Pandey et al. 1994]

2 SIMULATION

2.1 Particles in Cells

To study Jeans instability, we wrote a C++ simulation¹. Since the simulation has to compute thousands of particles' movements, an more efficient approach is used. Calculating the gravitational force of every particle on each other would have a run time in the order of $O(N^2)$ with N referencing the number of particles. With the simulation having thousands of particles the run time would become large quickly. To mitigate this issue, the Particle-in-Cell (PIC) [Birdsall and Langdon 2018] simulation method is used. In addition to using a PIC method, the simulation is normalized to prevent data loss and simplify the outputs. The PIC simulation method works by having super-particles within a grid. Super-particles are used to represent a clump of particles as opposed to an individual particle. Within the code, this is done by creating a particle object with necessary private variables and public functions to interact with. To reduce the run time the method that a PIC simulation uses is by placing particles within a periodic grid. The use of the grid is to create "cells" with which the particles interact which significantly reduces run time. The grid is used to define the dust density and gravitational potential. Using this process the run time for the most time-consuming section has been decreased to $O(N + G)$ with N being the number of particles and G being the grid points and $G \ll N$.

2.2 Initialization

To get the simulation working, a few things have to get set up within the code. First important global variables need to be set. Since making a variable global is bad practice #define will be used to declare them. The values set were the number of particles per cell (PPC), the size of the grid, the number of grid divisions, and the number of loops. After initializing the electric field, the magnetic field, and files to write data to, the grid is initialized. The grid is initialized as a 2D double vector. After the particles are initialized with a random velocity with a Maxwell distribution to represent their thermal velocity setting the thermal velocity of the whole to 1. Along with a random thermal velocity the particles are placed randomly on the grid and pushed onto a vector of "Particle" objects.

¹available at: <https://github.com/Actual-Heathen/ALPIP-Dusty-Plasma>

2.3 Gravitational Potential

Once inside the simulation's main loop, the calculation for the gravitational potential can start. First other data tracking variables are set along with important Fourier transform functions. To calculate fast Fourier transforms (FFT) the FFTW Library² will be used. After resetting the density grid the masses will be distributed. To distribute the masses of particles, the four nearest grid points are found. the points are found by using

$$x_{lower} = \left\lceil \frac{x_{dust} * divisions}{size} \right\rceil \quad (1)$$

$$x_{upper} = \left\lceil \frac{x_{dust} * divisions}{size} \right\rceil \quad (2)$$

With *divisions* being the number of divisions on the grid as defined atop the code and *size* the size of the grid as defined with *divisions*. If the ceiling values results are greater than the divisions it is changed to 0 as the grid loops periodically. The same process works for the *y* coordinate with an appropriate adjustment of *x* to *y*. In addition to getting the grid points, the weights for each point are calculated with

$$xw_{lower} = 1 - \left| \frac{x_{dust} * divisions}{size} - x_{lower} \right| \quad (3)$$

$$xw_{upper} = 1 - \left| \frac{x_{dust} * divisions}{size} - x_{upper} \right| \quad (4)$$

The mass is then distributed by adding the product of the appropriate weights onto a point with the following format

$$density_{xa,yb} += \frac{xw_a * yw_b}{ppc} \quad (5)$$

with *a* and *b* being replaced with *lower* and *upper* as necessary. Once the densities of the system have been distributed there is still more calculation to be done to get the acceleration of the particles. To get the potential the Poisson equation

$$\nabla^2 \Psi = -4\pi G \rho \quad (6)$$

must be solved. With the simulation being normalized the gravitational constant $4\pi G$ is set to 1. To isolate Ψ the densities of the grid points must be Fourier transformed. To transform the grid the real values were set in the complex 1D input in row-major order to FFTW's input. The imaginary values were then set as a value of 0. In the Fourier space, the equation becomes

$$-\mathcal{K}^2 \tilde{\Psi} = -4\pi G \tilde{\rho} \quad (7)$$

Then Ψ is isolated. To isolate Ψ first a kx and a ky of length *divisions*. The values inside the vectors are defined by

$$kx_i = \frac{2i\pi}{size} \quad (8)$$

$$kx_{division-1} = \frac{-(i+1)2\pi}{size} \quad (9)$$

Due to the symmetric nature of the Fourier transform the ends can be calculated approaching the inside requiring a loop half the size of vectors. Then to get

$$\tilde{\Psi} = \frac{4\pi G \tilde{\rho}}{\mathcal{K}^2} \quad (10)$$

²<https://www.fftw.org/>

we will use our vectors in

$$\tilde{\Psi}_{i,j} = \frac{4\pi G \tilde{\rho}}{kx_i^2 + ky_j^2} \quad (11)$$

with the same being done for the imaginary values. Finally, we inverse Fourier transform the equation giving us Ψ . Getting the acceleration from here is then a simple task. Similar to the distribution of density on a grid the now calculated Ψ values are used to get $\Delta\Psi$ in 2D to get the gradient of each point. The values are gotten in two separate 2D vectors for *x* and *y*. they are calculated by

$$\Delta\Psi x_{i,j} = \frac{\Psi_{i-1,j} - \Psi_{i+1,j}}{2spacing} \quad (12)$$

$$\Delta\Psi y_{i,j} = \frac{\Psi_{i,j-1} - \Psi_{i,j+1}}{2spacing} \quad (13)$$

2.4 Particle Acceleration

With this section being the most computationally intensive we will take advantage of the fact that all the particles are independent. To take advantage of the independence of this step the particle loop will be done in parallel with the use of the OpenMP library. With the gotten $\Delta\Psi$ values each particle gets passed the $\Delta\Psi$ vectors with the electric and magnetic fields. Within the particle, its nearest four points are found with the use of equations 1 & 2. Then weights are gotten for each with the use of equations 3 & 4. To calculate the acceleration due to gravity we add the sum of the following for all 4 points.

$$accel_x += xw_a * yw_b * \Delta\Psi x_{a,b} \quad (14)$$

The same is done for $accel_y$ by replacing $\Delta\Psi x_{a,b}$ with $\Delta\Psi y_{a,b}$. which would result in

$$F_G = \nabla \Psi \quad (15)$$

Once that is done we can move on to the application of the Lorentz force.

2.5 Lorentz Force

Since the dust particles have a charge they are also affected by the Lorentz force

$$\vec{F} = q\vec{E} + qv \times \vec{B} \quad (16)$$

To get the Lorentz force we use an array of size 3 to represent 3D space. we first start with

$$vm = \begin{cases} (accel_x + \frac{q}{m} E_0) \frac{time}{2} + velocity_x \\ (accel_y + \frac{q}{m} E_1) \frac{time}{2} + velocity_y \\ (accel_z + \frac{q}{m} E_2) \frac{time}{2} + velocity_z \end{cases} \quad (17)$$

then

$$t = \begin{cases} \frac{q}{m} B_0 \frac{time}{2} \\ \frac{q}{m} B_1 \frac{time}{2} \\ \frac{q}{m} B_2 \frac{time}{2} \end{cases} \quad (18)$$

next,

$$v_s = \begin{cases} vm_0 + vm_1 t_2 - vm_2 t_1 \\ vm_1 + vm_2 t_0 - vm_0 t_2 \\ vm_2 + vm_0 t_1 - vm_1 t_0 \end{cases} \quad (19)$$

then,

$$s = \begin{cases} \frac{2t_0}{1+t_0^2+t_1^2+t_2^2} \\ \frac{2t_1}{1+t_0^2+t_1^2+t_2^2} \\ \frac{2t_2}{1+t_0^2+t_1^2+t_2^2} \end{cases} \quad (20)$$

and finally,

$$v_p = \begin{cases} vm_0 + v_{s1}s_2 - v_{s2}s_1 \\ vm_1 + v_{s2}s_0 - v_{s0}s_2 \\ vm_2 + v_{s0}s_1 - v_{s1}s_0 \end{cases} \quad (21)$$

now with v_p the velocity of the particle can be calculated by

$$\begin{cases} v_x = vp_0 + \frac{q}{m}(E_0 + accel_x) \frac{time}{2} \\ v_y = vp_1 + \frac{q}{m}(E_1 + accel_y) \frac{time}{2} \\ v_z = vp_2 + \frac{q}{m}(E_2 + accel_z) \frac{time}{2} \end{cases} \quad (22)$$

afterward, the speed of the particle is gotten by

$$speed = \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (23)$$

Once this is all calculated the particle is pushed with the use of the velocity with

$$pos_x += vel_x * time \quad (24)$$

with the same done for the y position. When the particle is moved the position is checked if it is outside the boundaries. If the particle is moved beyond the bounds it will be looped around periodically. after moving the acceleration is reset to 0 to prepare for the next loop.

2.6 Compiling and More

2.6.1 Notes. After the main loop is through all variables are cleaned up and files are closed to prevent errors or leaks. In addition, various sections that are independent like resetting vectors are all done in parallel to maximize speed

2.6.2 Compiling. The code was compiled with

```
clang++ -g -Wall -O3 -fopenmp -march=native
clang, -O3, and -march=native were used as they will provide the
most efficient compilation output possible.
```

3 RESULTS

3.1 overview

We analyzed four different magnetic field configurations with varying field strengths. To get a reasonable compute time and quality balance all the simulations were run with 100 divisions on the grid, a grid size of 100, 100 particles per cell were used totaling 100000 particles, with 1000 loops. The cases we observed were $B(0, 0, 0)$ as a base without a field which would serve as a sort of control. We then had a case with the field going out of the page of $B(0, 0, 1)$. We had a case for the field in-plane of $B(0, 1, 0)$. In addition to these, we also had an oblique case with the field going out of the page at an angle with $B(0, 1, 1)$. For all these, we had field strengths of 0.1, 1, and 10. There was an exception in the oblique case where the values were changed to $1/\sqrt{2}$ for a total magnitude of 1 and the 10 case was not studied as the time steps were greater than the cyclotron frequency which would cause a positive energy leakage. The case was omitted

as reducing the time steps to be able to properly compare it required more computational time than was available.

3.2 Plotting

To be able to analyze the simulation data was written out into files. The data was written as simple text simply using the new line character as the delimiter for single data streams and spaces for data with time. We chose to output the densities upon the grid to be able to see how the particles are situated. Every time step was stored in the same file for simple organization and to allow the script to make an animated GIF of the data. Along with the density the real space of the Fourier transform was saved in the same way. The adjusted Fourier transform and the values of Ψ were also written onto their separate files. Having all the major stages of the gravitational potential allows for placing all four grids in a 2×2 grid for the comparison of the values over time. In addition to this, the energy values were saved to their files. The Kinetic energy was simply solved by using

$$\sum_{i=0}^{particles} \frac{1}{2} m_i v_i^2 \quad (25)$$

[Yoshikawa et al. 2012] Then the gravitational potential was calculated with the use of

$$\sum_{i=0}^{divisions} \sum_{j=0}^{divisions} -\frac{1}{2} \rho_{i,j} \Psi_{i,j} \quad (26)$$

[Yoshikawa et al. 2012] Then to get the total energy of the system the difference between the two was used. In addition to this the squared mean of the densities, to observe the dispersion is calculated with

$$\frac{1}{divisions^2} \sum_{i=0}^{divisions} \sum_{j=0}^{divisions} \rho_{i,j}^2 \quad (27)$$

To plot the data python scripts were used. While python can be a slow language it is easy to write in, letting more time be allocated to writing the C++ code.

3.3 Control

As a control, a base case with no Magnetic field was considered. Plotting the energy of the system we can see the evolution of the system's energy with time. Some fluctuations are visible in the energy. These fluctuations are however of only about 5% of the energy within the particles. These fluctuations are in addition brought upon by the sizes of the time, PPC, and grid divisions. Along with this, we have the dispersion of the system over time. A notable detail is the as squared mean goes up so does the kinetic energy. This however is no longer the case at the 6.8 time step, where the density begins to oscillate and dampen where the energy keeps climbing until time step 12. In addition to this discrepancy, the formations of the particles are vastly different. as seen in figures 3 & 4. Afterward, the data values settle down, with the particles in one mass with a small accretion disk, with the whole object rotating clockwise, as shown in figure 5.

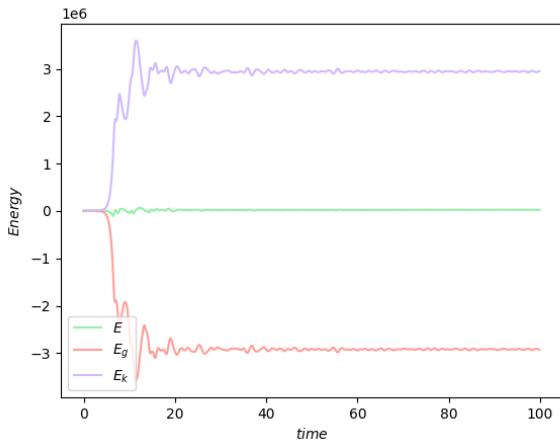


Fig. 1. Control energy values over time

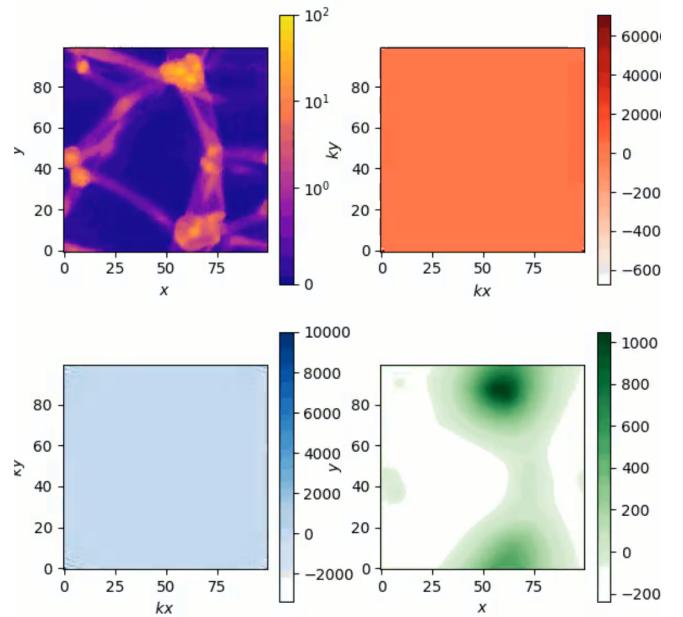


Fig. 3. Control mean density maximum

Top left: ρ on log scale, Top right: Fourier transform of densities,
Bottom left: adjusted Fourier transform, Bottom right: Ψ

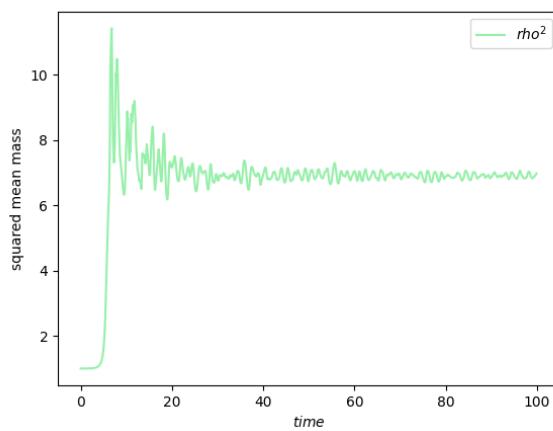
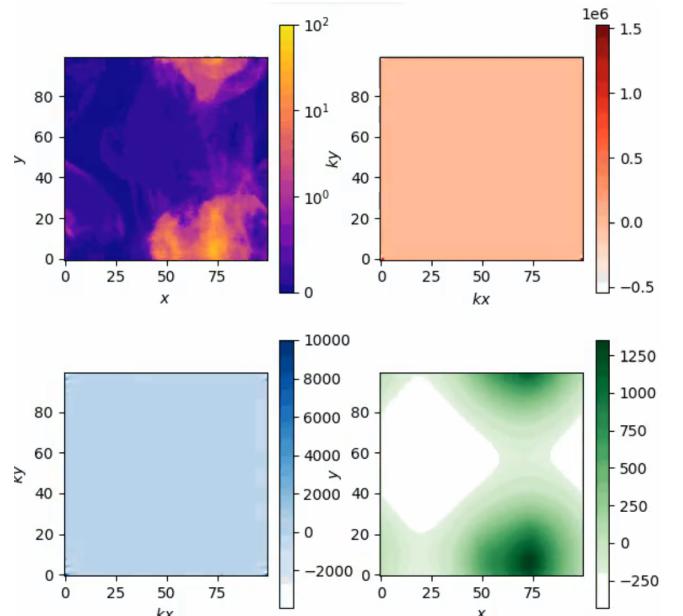
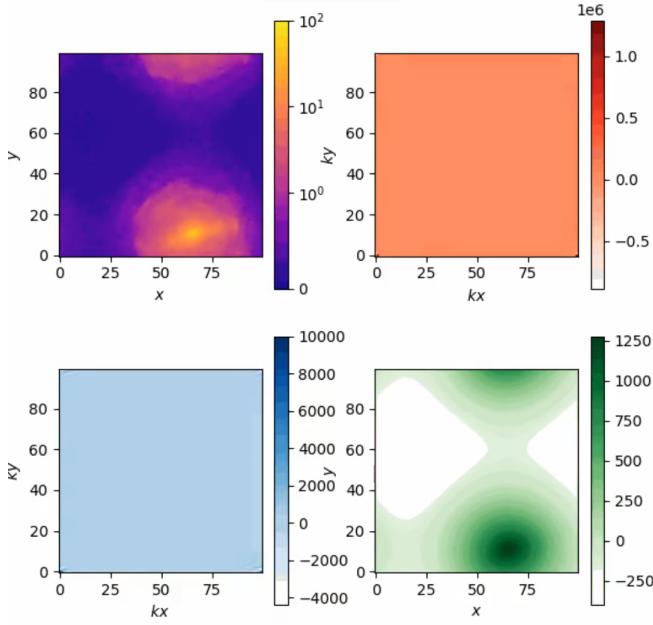


Fig. 2. Control mean density squared over time

Fig. 4. $B(0, 0, 0)$ Energy Maximum, see fig. 3 for legend.

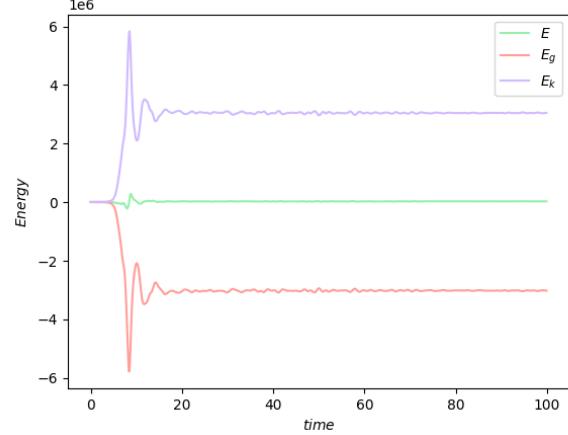
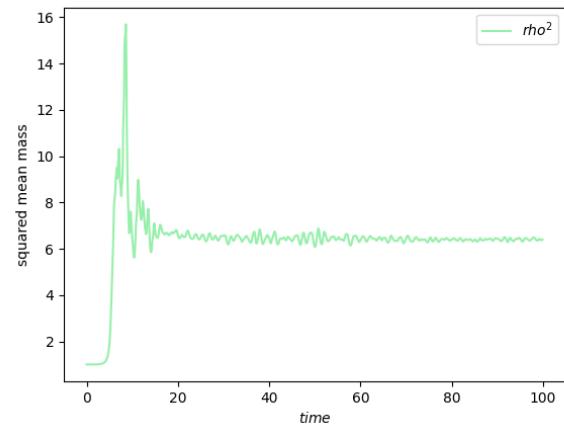
Fig. 5. $B(0, 0, 0)$ Stable, see fig. 3 for legend.

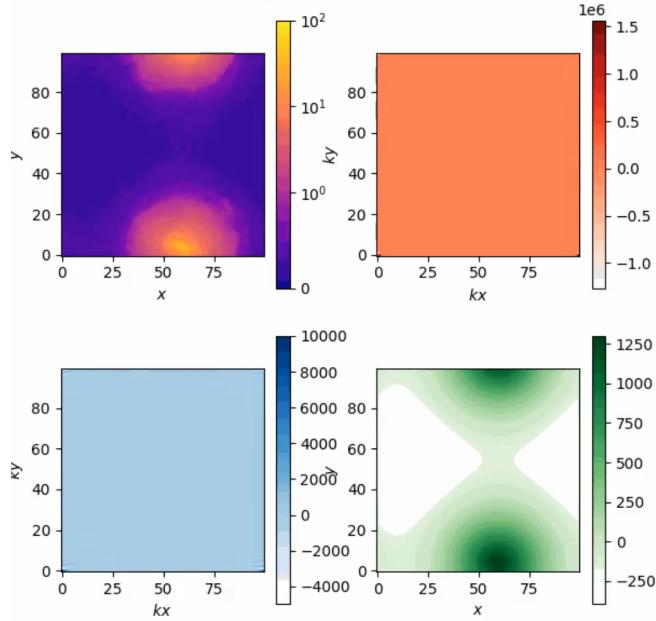
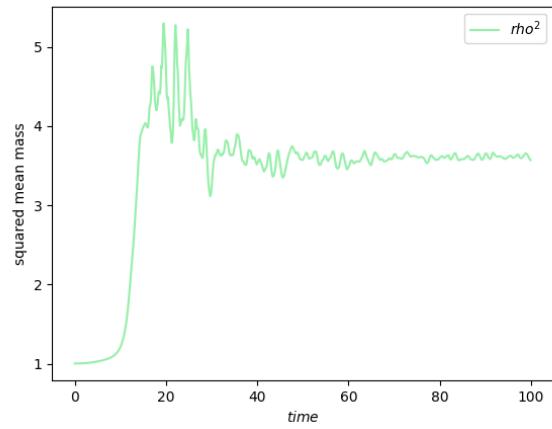
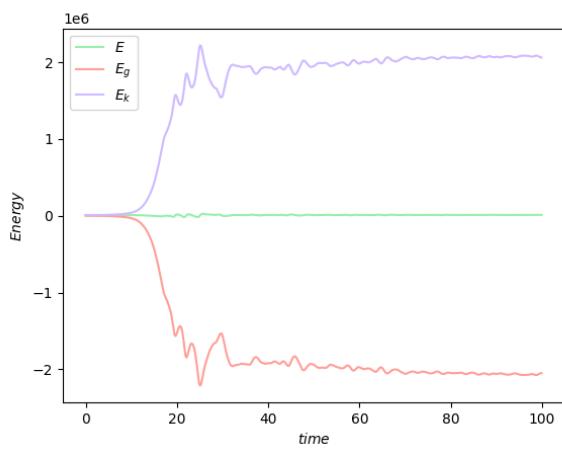
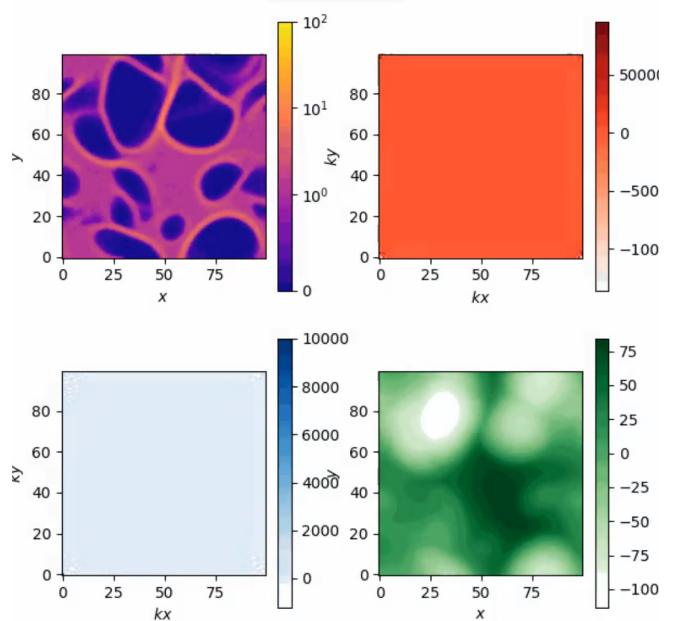
3.4 Out of Plane

3.4.1 $|B| = 0.1$. With the magnetic field perpendicular to the particles at a magnitude of 0.1, the effects on the energy and dispersion are visible. The effects, however, are not significant on the formation of the particles. One of the major changes with the field is the greater delay in the collapse of the particles due to the field only letting free movement on the z-axis. This slowing also seems to cause the system to have a greater maximum mean density and kinetic energy maximum. Along with this the maximums also occur at about the same time. Since the formations are visually the same they have been omitted with one exception. When stable, the system behaves differently having a much smaller accretion dist. in addition to this, the system starts rotating clockwise but soon starts going counterclockwise and seems to possibly begin to oscillate.

3.4.2 $|B| = 1$. At a magnitude of 1, the effects of the .1 simulation become exaggerated. One noticeable change is that the square mean has a much lower peak than .1 and the control. Furthermore, the field stretches the energy graph and lowers the peak with a smaller discrepancy between the stable values and the peak. This is also the first instance of a large visible change in the system's collapse. During collapse instead of forming straight into nodes like in figure 3 the system "rips" into holes like in cloth, which rotate while expanding. The expansion of the holes goes on to create four major holes with the particles in a cross shape with a central node on which the system then collapses. When collapsing the system doesn't go straight into a central node but collapses vertically creating a large horizontal cluster that ends as a singular object stretched horizontally.

3.4.3 $|B| = 10$. At a magnitude of 10, the system becomes "locked" with no real noticeable movement. the only visible change was in the squared mean moving up but many orders of magnitude less than any other previous graph. All the graphs except for the squared mean have been omitted as they will not be able to provide any aid.

Fig. 6. Out of Plane $|B| = 0.1$ energy plotFig. 7. Out of Plane $|B| = 0.1$ mean density plot

Fig. 8. Out of Plane $|B| = 0.1$ stable, see fig. 3 for legend.Fig. 10. Out of Plane $|B| = 1$ mean density plotFig. 9. Out of Plane $|B| = 1$ energy plotFig. 11. Out of Plane $|B| = 1$ "holes"

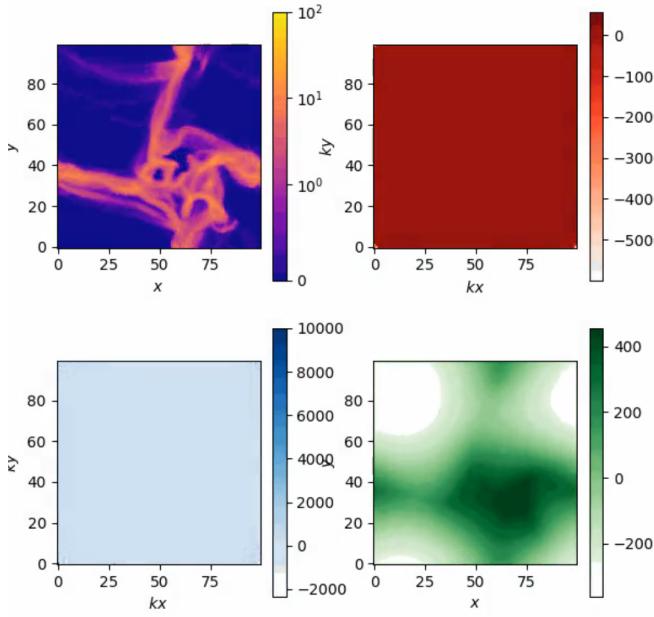
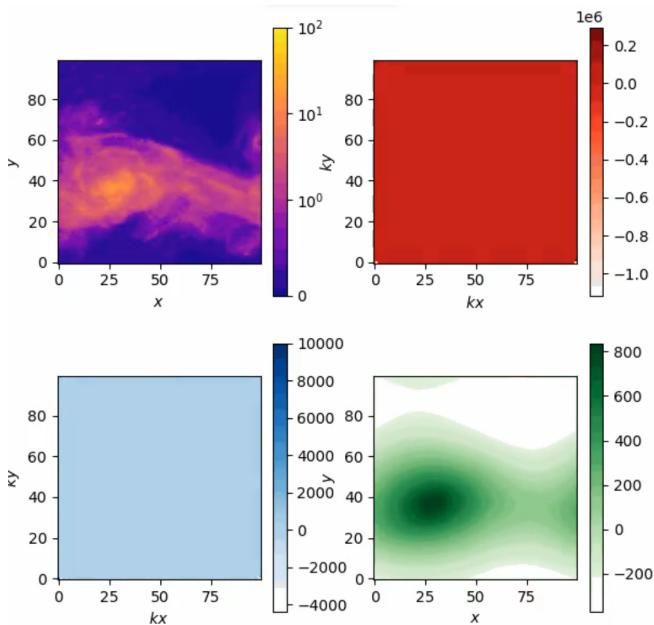
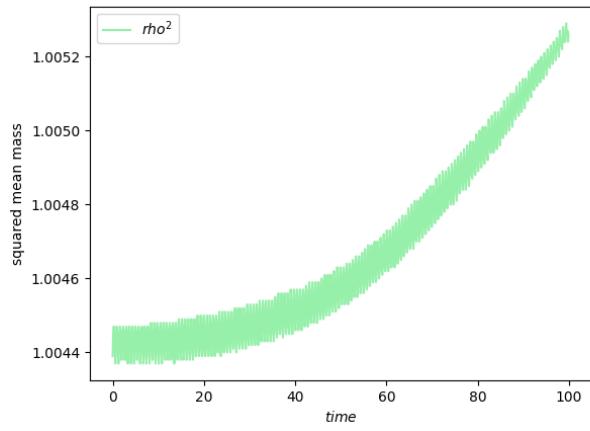


Fig. 12. Out of Plane "cross"

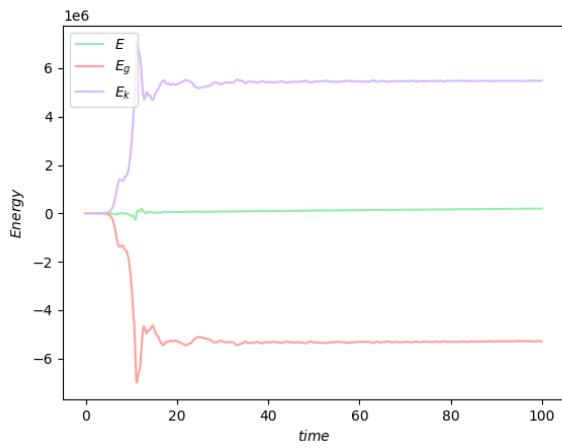
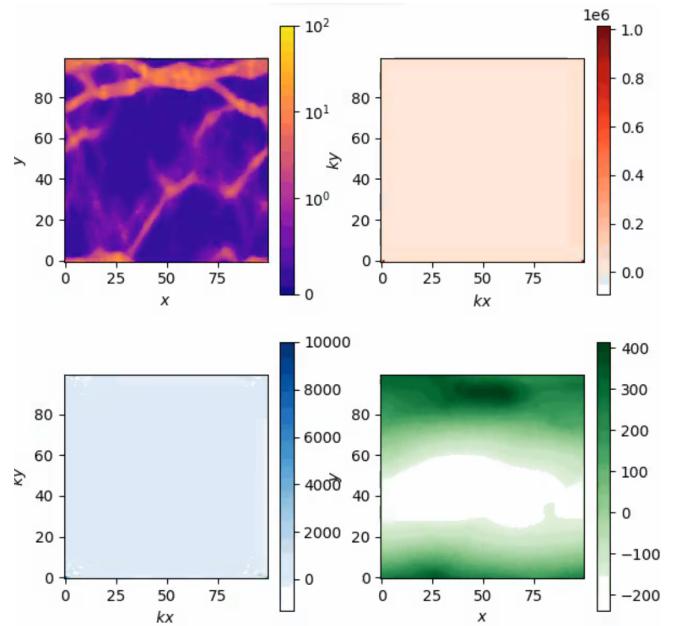
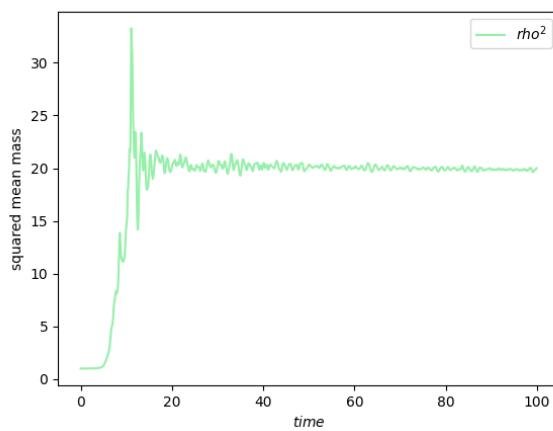
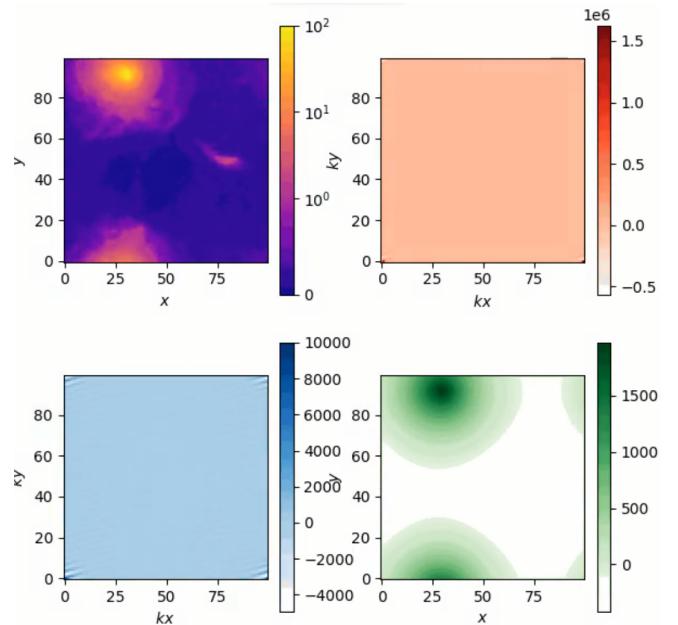
Fig. 13. Out of Plane $|B| = 1$ horizontal formation before being stableFig. 14. Out of Plane $|B| = 10$ mean squared plot

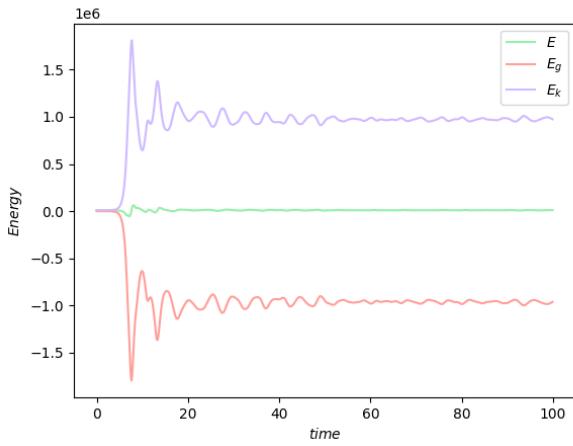
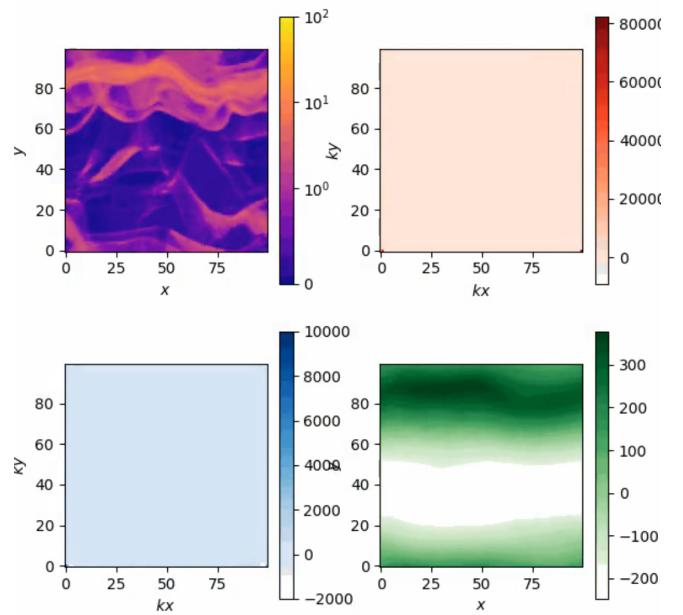
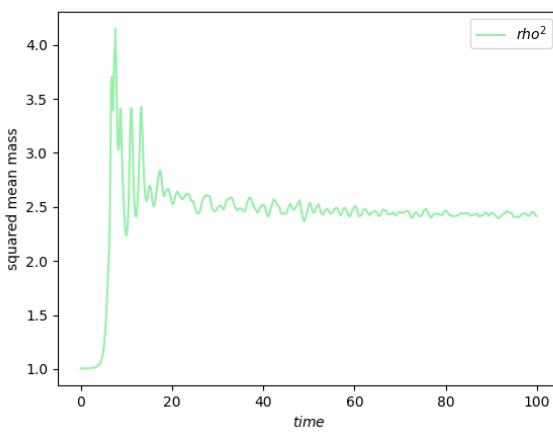
3.5 In Plane

3.5.1 $|B| = 0.1$. At a magnitude of 0.1, the effects are once more fairly muted. Many of the changes it brings along are very similar to the effects of the $B(0, 0, 1)$ at 0.1. Just like before the visual effects in the formations are like that of the control. The mean squared is different from the control but it looks like that of figure 7 but peaking at a value of 14 instead. The energy plot is shaped much like figure 6 but it with a smaller peak at 4 instead.

3.5.2 $|B| = 1$. At a magnitude of 1, there are some interesting changes in the system. First, in the energy plot it looks like that of figure 6 but it instead settles at 5. A major change comes in the way of the squared mean with the mean values being much greater than any other plot. The formation is also different in that the particles collapse vertically first followed by quickly forming into a tight object.

3.5.3 $|B| = 10$. At a magnitude of 10, unlike $B(0, 0, 1)$ there is movement in the particles. The changes to the energy plot come in the form of looking like the control but being highly damped. The squared mean is also like the control in shape but, again, it is highly damped. With the shapes being so similar to the control it would be assumed that the formation of the particles in this state would be like that of the control but looser. This however is not the case as the particles do not settle into a large central mass but instead collapse vertically into wave-shaped objects oscillating.

Fig. 15. In Plane $|B| = 1$ energy PlotFig. 17. In plane $|B| = 1$ vertical compressionFig. 16. In plane $|B| = 1$ squared mean plotFig. 18. In Plane $|B| = 1$ stable

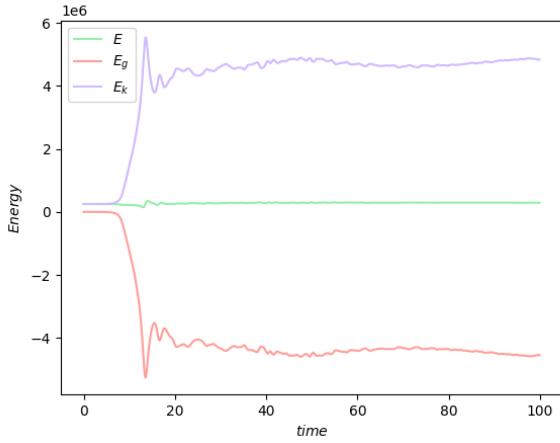
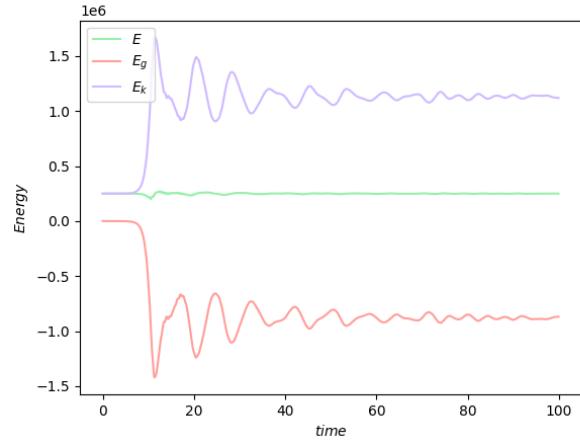
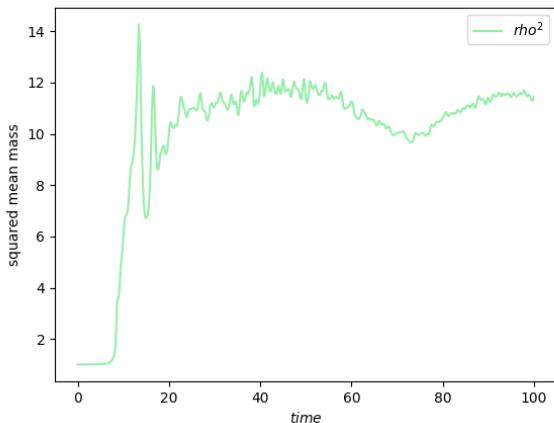
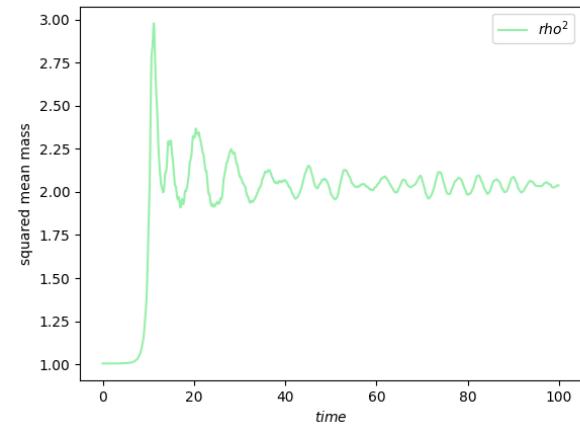
Fig. 19. In Plane $|B| = 10$ Energy plotFig. 21. In Plane $|B| = 10$ waveFig. 20. In Plane $|B| = 10$ squared mean

3.6 Oblique

3.6.1 $|B| = 0.1$. For the 0.1 magnitude of the oblique angle, the magnitude represented the magnitude of the vector and not of the components. At this magnitude, the strength of the field is small enough that the effects on it are once again incredibly minor it behaves like the control.

3.6.2 $|B| = 1$. once at a magnitude of 1, there are changes visible in the plots. Most notably that both the plots in addition to their high-frequency oscillation have a low-frequency oscillation within. For the formations, the system collapses like in figure 17 but it doesn't settle into a tight object like in figure 18, it settles like in figure 5.

3.6.3 $|B| = 10$. At a magnitude of 10, the effects from magnitude 1 are more prevalent with a few changes. At this magnitude, the additional lower frequency wave that was brought in case 1 is visible but at a higher frequency than the magnitude 1 case. In the physical formation of the particles, they behave like the in plane case in figure 21 collapsing vertically and as a long horizontal object.

Fig. 22. Oblique $|B| = 1$ energy plotFig. 24. Oblique $|B| = 10$ energy plotFig. 23. Oblique $|B| = 1$ squared densityFig. 25. Oblique $|B| = 10$ squared density

4 IMPACT OF INTERNSHIP ON MY CAREER

The internship has had an impact on my career comes in many ways. One of the major ways is by giving me experience in research. Having experience in a research setting is extremely valuable. Much of the aid comes due to my goal of pursuing a graduate degree. It also aided in that experience in researching helped by further developing my breadth of knowledge in the field. An important skill that came from the internship is also a direct tie to the gained experience. The Internship helped me with the skills used to work in a research setting. In addition to education plans, it will help me in my career plans. The experience and skills gained from the internship helped in my career. It will help in that the internship gave me research experience and skills which help my career plans as my plans directly involve research.

ACKNOWLEDGMENTS

This work is supported by the NSF EPSCoR RII-Track-1 Cooperative Agreement OIA-1655280

REFERENCES

- Charles K Birdsall and A Bruce Langdon. 2018. *Plasma physics via computer simulation*. CRC press.
- BP Pandey, K Avinash, and CB Dwivedi. 1994. Jeans instability of a dusty plasma. *Physical Review E* 49, 6 (1994), 5599.
- Kohji Yoshikawa, Naoki Yoshida, and Masayuki Umemura. 2012. Direct integration of the collisionless Boltzmann equation in six-dimensional phase space: Self-gravitating systems. *The Astrophysical Journal* 762, 2 (2012), 116.