The Effects of Turbulence on Dust Particle Coagulation in Planetary Formation

Peter J. Fager, Lorin S. Matthews, and William C. Anderson

***Abstract*— Planetary formation is caused by the coagulation of dust grains as they orbit around a star. There are several factors that affect this process, such as dust charge, drag, and random motions induced by turbulence, which is the focus of this research. The effects of turbulence have been studied on a larger scale, but a detailed investigation of the effects on coagulation of micron-sized dust has not been carried out. In order to do this, numerical data of a turbulent flow was collected and verified before being input into a numerical simulation program called BoxTree. This program simulates the motion and interaction of dust particles in space, given different user-specified parameters. This program was modified to read in the turbulence data and calculate the force on the particles due to the turbulence. Once these modifications were made, simulations were run for the simplest case of a single particle at rest, with no forces other than the turbulence. The work done for this paper shows that turbulent gas flow drives the motion of a dust grain; it also lays the foundation for more in-depth research into this topic.**

***Index Terms*— coagulation, planetary formation, turbulence, dust particle simulations**

I. Introduction

THE formation of planets is, at the smallest scale, driven by the growth of dust grains into planetesimals in the proto-planetary disc [1]. These micron-sized dust grains orbit around a new star, and are surrounded by ionized gases, which are also orbiting the star. As these dust grains are moving around the star, they collide with, and stick to, each other, forming what are called "fractal aggregates" [2].

There are several factors that influence the coagulation of these dust particles, such as dust charging due to the ionized gas. This negative charging of the particles causes an electrostatic barrier against growth, because the particles must have enough velocity to overcome this repelling force.

Another factor that affects this formation of aggregates is the random motion of the particles induced by turbulence. As the gas flows around the star, there is turbulence induced in the flow, which causes a change in the velocity and motion of the particles. This change can cause an increase in the sticking of the aggregates, an increase in the fragmentation of the aggregates, or a decrease in the overall coagulation rate of the particles [3].

As the aggregates increase in size, the collisions have enough energy to cause the aggregates to compact together [4]. As they become more compact, the drag forces decrease on the aggregates, causing their velocities to increase. It is unknown whether or not the increased velocity increases the coagulation or the fragmentation of the aggregates [5].

However, on the smaller scale, the effects of the turbulence are less known. One possibility is that the turbulence will increase the rate of coagulation of the particles. With the increased velocity of the particles due to the turbulence, the possibility of a particle traveling fast enough to overcome the electrostatic barrier is higher. If the coagulation rate increase, the next question raised is how does that coagulation happen? Are the aggregates "fluffier" or will they be denser?

The turbulence does not necessarily mean that the particles will coagulate more quickly, because turbulence is efficient at diffusing dust particles [6]. This would effectively slow down the rate of dust particle coagulation and, as a result, slow down the planetary formation rate.

II. Methods

In order to see which of the three possible effects the turbulence actually has on the dust particles, a numerical simulation of the dust particle motion was needed. A numerical model, called BoxTree [7], simulates the motion and coagulation of dust particles in a proto-planetary disc and has been used by Dr. Matthews for the last ten years to do simulations on fractal dust grain growth [8], and the coagulation of fluffy aggregates [9]. Therefore, this program was chosen to simulate the dust particle motion.

Before beginning simulations, the data of simulations of an actual turbulent flow were collected. The turbulence data was tested to ensure that it met the conditions of the underlying assumptions. Finally, code was written to read the data into BoxTree and allow it to be accessed by the program during its simulations.

A. Turbulence Data Collection

In order to be sure the data added into BoxTree was of a turbulent flow, a database containing this information was required. Johns Hopkins University (JHU) has such a database

Manuscript received August 9, 2013. This work was supported by the National Science Foundation by a grant to the CASPER program at Baylor University. P. Fager is a Senior Mechanical Engineering Student at Baylor University, Waco, TX 76798 USA (corresponding author to provide phone: 832-623-3888; e-mail: peter_fager@baylor.edu).

L. Matthews is with the Department of Physics, Baylor University, Waco, TX 76798 USA (e-mail: lorin_matthews@baylor.edu).

W. Anderson is with the Department of Mechanical Engineering, Baylor University, Waco, TX 76798 (e-mail: W.Anderson@baylor.edu).

which contains a direct numerical simulation of isotropic turbulent flow in incompressible fluid [8].

This database contains a 1024^4 space-time history of this flow field on servers that JHU maintains and provides resources and script files to collect information on flow properties for any range of points inside the field, such as the velocity, velocity gradient, force, pressure, pressure gradient, among other properties.

B. Data Verification

After collecting the data, it was necessary to verify that the data followed the assumptions used by JHU to create the data. Because of the statement that the simulation was of an incompressible fluid, the divergence of the fluid should then equal to zero, or:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

or, equivalently,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2}$$

with $\boldsymbol{u}$ being the velocity vector of the flow, and u, v and w being the x-, y-, and z-component velocities, respectively. Proof of this property was needed to verify the validity of the data. This was attempted in two different ways. First, a cube of a single time step of points with sides of 64 grid points was collected from the database. Each velocity component vector contained 64^3 points of data corresponding to the respective velocity at each point within the grid selected.

Because the data collected was a vector of values and not an equation, the differential can be approximated with finite differences by

$$\frac{\partial u}{\partial x} = \frac{\Delta u}{\Delta x}. \tag{3}$$

When performed on each of the three components of the velocity and added together, the divergence at each point in the data is calculated. This difference was calculated with the following equation:

$$\frac{\Delta u}{\Delta x} = \frac{u_{i+1} - u_i}{\Delta x} \tag{4}$$

with u_i being the *i*th term in the *u*-velocity vector. This same equation was applied to the *v* and *w* velocity vectors and the results were added together.

The divergence of the fluid was also calculated using a function provided by the JHU database called GetVelocityGradient, which called $\partial u/\partial x$, $\partial v/\partial y$, and $\partial w/\partial z$ values for all of the points desired. These values were then added together to calculate the divergence at every point in the cube of data.

The last method used to verify that the data was of an incompressible fluid was the method of Fast Fourier Transforms (fft). The method of Fourier Transforms states that

$$\hat{u} = F\{\boldsymbol{u}\} \tag{5}$$

$$\nabla \cdot \boldsymbol{u} = F^{-1}\{i\boldsymbol{k} \cdot \hat{u}\} \tag{6}$$

with $F\{\boldsymbol{u}\}$ being the Fourier transform of the vector $\boldsymbol{u}$, and $F^{-1}\{i\boldsymbol{k} \cdot \hat{u}\}$ being the inverse Fourier Transform. In order to use this method, three lines of data were selected from the database, centered on a single point. Each side of the total cube of data in the database has a length of 2π, therefore the point $p = (\pi, \pi, \pi)$ was chosen as the intersection point of the three lines. Then, a line of data extending from zero to 2π, parallel to the x-axis and intersecting *p* was chosen to collect the *u*-velocity data. The same process was used to collect the *y*- and *z*-velocity data, but with lines parallel to the *y*- and *z*-axes, respectively. Once the data was collected, the above method was applied and the divergence was then calculated.

C. Integration into BoxTree

Once the data was collected from the turbulence database, it was necessary to incorporate it into the BoxTree functionality. BoxTree was created in a Unix system using C code by Derek Richardson in order to simulate dust particle motion in space for various conditions, such as Saturn's rings or a proto-planetary disc. There are various conditions that can be turned on or off to tailor the simulation to the user's desires.

One of those conditions is an option to include a gas drag force on the particles, and since the turbulence would cause a viscous drag on the particles, the function that added the gas drag was chose to be modified to read in the turbulence data and add the force to the particles.

Because of the size of the particles, the typical equation for aerodynamic drag is insufficient in describing the actual force exerted on the particles from the turbulence. Therefore, Stokes equation for drag was used to find the drag force on the particles,

$$F_D = 6\pi\mu\boldsymbol{u}r \tag{7}$$

where F_D is the drag force, μ is the absolute viscosity, $\boldsymbol{u}$ is the velocity of the flow, and r is the radius of the particle. Once the data was read into the function in BoxTree, this equation was used to calculate the drag force on the particles, which was then added to the total force acting on the particles.

Before making modifications to the function GasDrag, a few simulations were run with the original function in order to better understand how the function worked. The program was run with drag coefficient values ranging from 0.002 to 0.01 in increments of 0.002. All three drag coefficients were equal for each run of the simulation. Once the simulations finished, the data was transferred to Matlab for analysis.

After the modifications were made and debugged, the program was run for the simplest case of one particle starting at rest with only the turbulence acting upon it. The data was then retrieved and analyzed in Matlab in the same manner as the other simulations.

III. Results

A. Turbulence Data Collection

As stated above, the first thing needed was the data on a turbulent flow. JHU provided functions with their database that allowed for the collection of the data, and a function that calls the velocity data for a defined grid of points was selected to collect the u, v, and w velocity data for the above-mentioned cube of data.

B. Turbulence Data verification

After successfully collecting the data, the first method used to verify the data that was mentioned above, using finite differences, was implemented. The results were not what was expected, and are shown in Table I.

TABLE I
DIVERGENCE VALUES (FINITE DIFFERENCES)

Method	Low Value	High Value
Matlab Function	-0.7514	0.6391
JHU Function	-0.6233	0.8447

TABLE I contains the results of the divergence calculations using finite differences on the turbulence data collected from the JHU database.

This is due to the limited scope of using finite differences – only the next point was taken into account regarding the change in velocity, as opposed to the entire data set. It is because of this property that the results from using finite differences were not what was expected. Next, the method of Fourier Transforms was applied to three different sets of three lines of data collected from the database, with the results presented in Table II.

TABLE II
DIVERGENCE VALUES (FOURIER TRANSFORMS)

Center Point	Low Value	High Value
$\left(\frac{\pi}{2},\frac{\pi}{2},\frac{\pi}{2}\right)$	-0.03581	0.03899
(π,π,π)	-0.02728	0.0508
$\left(\frac{3\pi}{2},\frac{3\pi}{2},\frac{3\pi}{2}\right)$	-0.03092	0.03952

TABLE II contains the results of the divergence calculations using the method of Fourier Transforms on three sets of three lines of velocity data.

After performing the Fourier Transforms on the three sets of data, histograms were made of the divergence values at every point for each of the three sets of data which provided significant insight into the actual value of the divergence throughout the data. Figures 1, 2 and 3 show the range of values for the divergence calculations.

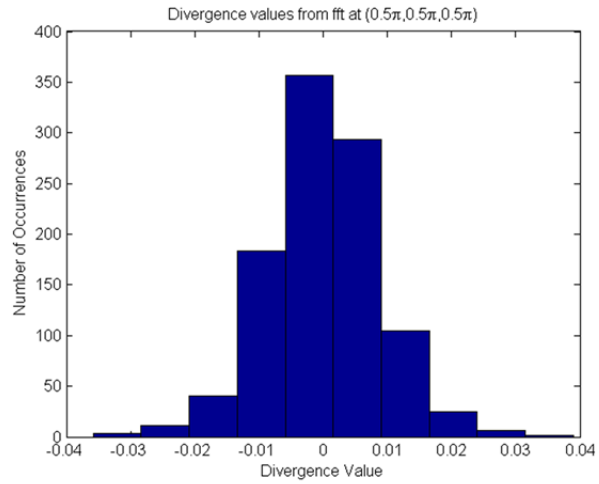

Fig. 1. Divergence Values from fft at $\left(\frac{\pi}{2},\frac{\pi}{2},\frac{\pi}{2}\right)$. This figure shows the number of occurrences of specific ranges of values of the divergence calculated for the three lines centered at $\left(\frac{\pi}{2},\frac{\pi}{2},\frac{\pi}{2}\right)$.

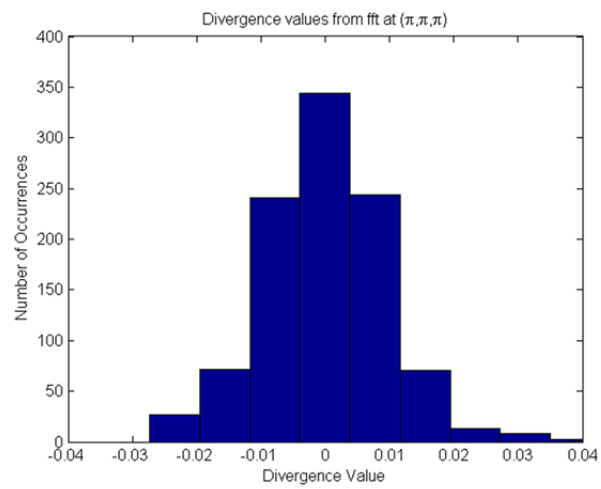

Fig. 2. Divergence Values from fft at (π,π,π). This figure shows the number of occurrences of specific ranges of values of the divergence calculated for the three lines centered in the middle of the box. Note that nearly 80% of the values lie between [-0.01,0.01].

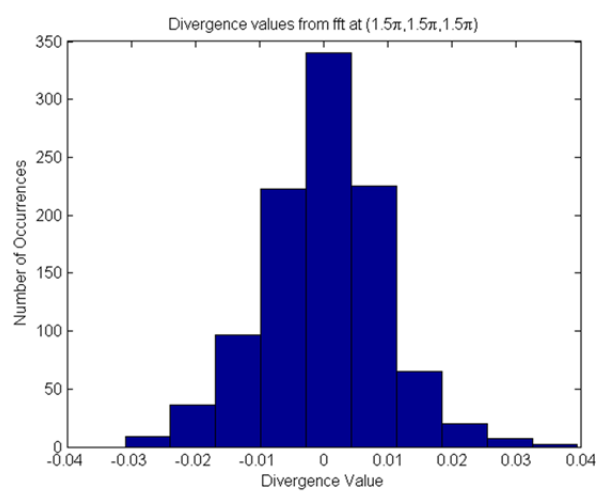

Fig. 3. Divergence Values from fft at $\left({}^{3\pi}/_{2},{}^{3\pi}/_{2},{}^{3\pi}/_{2}\right)$. This figure shows the number of occurrences of specific ranges of values of the divergence calculated for the three lines centered at $\left({}^{3\pi}/_{2},{}^{3\pi}/_{2},{}^{3\pi}/_{2}\right)$.

C. Integration into BoxTree

Before making modifications to BoxTree, a few simulations were run with the original version of the GasDrag function in order to ascertain its functionality. In the user-defined parameter file, beneath the option for "Include gas drag?" there are three properties that can be set, and they are the coefficient of drag in *x*, *y*, and *z*. These are the values that were edited as described in the previous section. A random particle was selected and the magnitude of its velocity was plotted for each value of the drag coefficients. Figure 4 below shows the results of these simulations, as well as a comparison between the magnitudes of velocity for the different drag coefficient values.

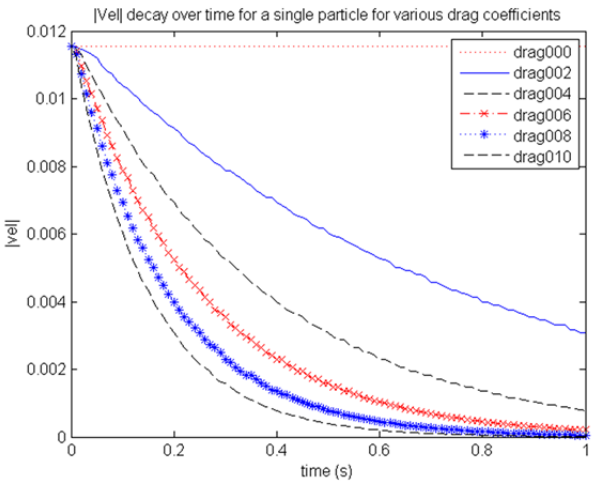

Fig. 4. Velocity Magnitude Decay Over Time. This figure shows how the magnitude of velocity of a single particle decays over time for a specified value of drag coefficient.

The next step in trying to simulate the effect of the turbulence on the dust particles was to integrate the data into BoxTree and use Stokes Law to calculate the force on the particle and add that to the forces acting on the particles.

There were a few ways that this was attempted in an effort to maximize the speed of the simulations. It would be necessary to read in the data from a text file and then, once the data was in the program, convert it into a known configuration that could then be accessed when needed.

Originally, the functionality to read in the text files and then use the values in the force calculations was placed directly inside the GasDrag function itself. However, this would cause the program to call in the text files and read them into the program every time that the function was called, which could be several thousand times for a single simulation, depending on how many particles were being simulated and for how long.

Next, an attempt was made to create another function that would call in the data into a structure of arrays and pointers to the arrays would be passed into the GasDrag function. This method would cut down on processing time and would only require one instance of calling in the data, eliminating unnecessary I/O operations. However, this method also did not work because the value needed from the turbulence arrays would not be known until GasDrag was called, which meant that it would be necessary to create an indexed pointer or create a pointer to every single value in the arrays for a total of 786,432 pointers.

Since this would not be feasible, static arrays for the turbulence were created inside the function that would be initialized the first time the function executed only, and would still be accessible by the function every other time it was called.

Once the modifications were made, simulations were run using just the drag force from the turbulence on the particles.

However, the particles were unaffected by the turbulent gas drag, and the simulations would also produce error files stating that the particles had left the boundaries of the box and could not be found.

Further examination of the code and the components of Eq. (1) revealed two errors, one being that the three variables being used for the velocity input into the force equation were defined as the wrong class. Before calculating the force, a few intermediate steps were needed in order to get the correct value from the turbulence data, based on the particle position. The final variables used to hold the velocity values from the turbulence were defined as integers, and not doubles, which would cause any value less than 1 to be stored as zero. This explains the particles not being affected by the turbulence data, despite it being read in and stored in the array correctly.

The error discovered from the examination of the components of Eq. (1) was a result of using the wrong value for viscosity. The viscosity of the fluid was calculated using and equation found in Okuzumi *et al.* [11]

$$\nu = \sqrt{\frac{2}{\pi}} * \frac{c_s m_g}{\rho_g \sigma_{coll}} \tag{8}$$

where ν is the viscosity of the turbulence, c_s is the speed of sound in the gas, m_g is the mass of the atomic mass of the gas, ρ_g is the density of the gas, and σ_{coll} is the collisional cross section of the particle. At first, the value calculated using (8) was being used in the calculation of Stokes Drag, which was incorrect because using (8) calculates the dynamic viscosity, not the absolute viscosity, which is needed for Stokes Drag. The viscosity calculated from (8) was then converted into absolute viscosity by

$$\mu = \rho_g \nu \tag{9}$$

which, when inserted into (8), becomes

$$\mu = \sqrt{\frac{2}{\pi}} * \frac{c_s m_g}{\sigma_{coll}} \tag{10}$$

where μ is the absolute viscosity of the fluid. The values of the two viscosities are listed in Table III below.

The large difference in orders of magnitude of the two values explains why the particles were traveling outside the boundaries of the simulation – the force acting upon them from the turbulence data was nine orders of magnitude too large.

TABLE III
CALCULATED VISCOSITY VALUES

	Value (cm^2/s)
Kinematic Viscosity	1.858×10^{5}
Absolute Viscosity	1.858×10^{-4}

TABLE I contains the results of the divergence calculations using finite differences on the turbulence data collected from the JHU database.

Once these two errors were discovered and corrected, a simulation was run with one particle at rest initially and with only the turbulence acting on it. The magnitude of the velocity was then plotted over time and is shown in Fig. 5.

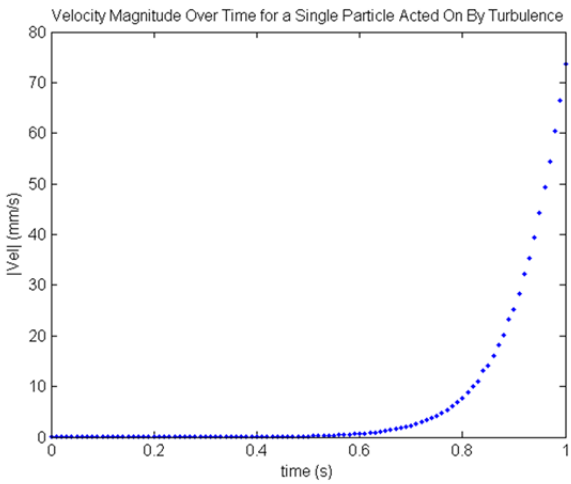

Fig. 5. Velocity Magnitude Over Time for a Single Particle. This figure shows how the magnitude of velocity of a single particle changes over time when started at rest and acted upon by turbulence.

The simulation time was one second, with data files containing position and velocity output every hundredth of a second. The data was then read Matlab and output into the graph in Fig. 5. As evidenced in the graph, the velocity of the particle started at zero, and slowly increased in total velocity until it accelerated quickly from 10 mm/s to 80 mm/s in a span of 0.2s. From the graph, it can be seen that the turbulence does have an effect on the motion of the particle.

IV. DISCUSSION

The two different results for calculating the value of divergence of the turbulence data are initially confusing that, while both finite differences and Fourier Transforms should theoretically work, only the method of Fourier Transforms produced results confirming the incompressibility of the fluid.

One possible reason for this is the scope of the data analysis. The program created in Matlab only accounted for the change between the current point and the next point in the data and no more. Due to this fact, the precision of using a first order finite difference is severely limited.

The JHU function experienced similar results, and for the same reason as the function created in Matlab. While the JHU function uses 8th order differentiation to calculate the gradient of the velocity, the nature of numerical simulations does not allow for finite differences to be as accurate in calculating the divergence of a flow.

The reason that the other two methods were ineffective for calculating the divergence of the flow is the reason why using Fourier Transforms was effective. By taking the Fourier Transform of the data and calculating the divergence through (6), the entire set of data was taken into account. The divergence at each point was calculated with respect to every point in the vector, not just the next point or the next set of points. While the first two methods would indicate that the divergence is not equal to zero and therefore the fluid is not incompressible, the fact that the Fourier Transform method indicated that the fluid is indeed incompressible, shows that using finite differences is not accurate enough to calculate the divergence. There was nothing wrong with the actual code used in calculating the divergence in the first two methods, they just were not accurate enough to show that the divergence equals zero.

While the values in Table II show minimum and maximum values for the divergence being on the order of 10^{-2}, which seems to indicate that the divergence might not be zero, Figures 1-3 reveal that these numbers are outliers in the data, and are likely the values at the boundaries of the data and can be ignored. In all three figures, nearly 75% of the values of the divergence fall within [-0.01, 0.01], which shows that the divergence can safely be considered to be equal to zero.

Verification of the data was an important step in the process of this work – it was necessary to be sure that the data incorporated into the simulations was truly data of a turbulent flow and not just a collection of arbitrary numbers. It would have been possible to perform the simulations with a random set of numbers, but those results would have been meaningless to studying the effects desired.

The tests of the original GasDrag function in BoxTree were used in order to gain a better understanding of how BoxTree worked and the effect that the unedited GasDrag function had on the particles. BoxTree was written in 1993 and the copy used for this research has been edited and modified by multiple other users, resulting in certain unknown functionalities that needed to be checked. Unfortunately, not all were able to be discovered by the initial tests. However, the initial tests of the program revealed that the original GasDrag function acted as a drag force and simply slowed the particles down and did not have an effect on the specific motion of the particles – it simply slowed them down, it did not change the direction of their motion.

Once simulations were finally run using the turbulence data to move the particles around, it showed, on the small scale, that the particles will react to the turbulence and their motion will be affected by the turbulence. The simulation validated the belief that turbulence has an effect on the coagulation of dust particles by showing the scale of the effect on the particles. In just one second, the velocity of the particle accelerated from zero to 80 mm/s.

While no simulations have been run to see what the exact

effect the turbulence has on the dust particle coagulation, the work outlined in this paper proves that there is an effect and has set up the basis needed for further research into this effect on planetary formation.

V. Conclusion

The dust particles in the simulation that were initially placed at rest were indeed moved around by the turbulence data, and the direction of the particle motion was influenced by the force due to the turbulence. This confirms that the turbulence does have an effect on the particle motion and serves to cause the motion of the particles to be even more randomized and chaotic.

In addition, the modification to the program BoxTree are functioning and can be easily modified to perform temporal interpolation to better simulate a turbulent flow.

While the exact relationship between turbulence and dust coagulation was not determined from this research, the foundations have been laid for future work on this topic.

VI. Future Work

Again, while no simulations were run to study what happens to a larger number of particles over a long time, the work done for this paper show that there is an effect to the motion of the particles and lays the foundation for future work in this research with BoxTree and the JHU database.

In the future, more data from the JHU database will need to be collected, in particular more time-steps of data will be needed for more accurate simulations. At a minimum, two time-steps of data will need to be collected in order to perform temporal interpolation of the data. The database is limited in the total number of time-steps that can be extracted from its servers – 1024 – but the researcher performing the simulations will need to specify that in the GasDrag function in order for the interpolation to be carried out correctly. How the interpolation would be calculated would be dependent upon the maximum and minimum time-steps specified in the parameter file for the function. as well as how the data is stored in the text files before being read into the program.

A more in-depth analysis of the effect of turbulence can be observed with longer simulations involving more particles. Another interesting property to examine would be the ratio of the turbulent force to the electrostatic force or the force of gravity, to see which of those three forces dominate the motion of the particles during planet formation.

Acknowledgment

This work was supported by NSF grant PHY-0847127. P. Fager would like to thank Baylor University for offering the opportunity to perform this work in the CASPER office, Dr. Truell Hyde of Baylor University, Dr. Lorin Matthews, Dr. William Anderson, and Johns Hopkins University for the use of their turbulence database.

References

[1] H. Mizuno. (1980, March). Formation of the Giant Planets. *Progress of Theoretical Physics* [Online]. 64(2). Available: http://ptp.oxfordjournals.org/content/64/2/544.abstract

[2] G. Wurm, J. Blum. (1998 March). Experiments on Preplanetary Dust Aggregation. *Icarus* [Online]. 132(1). Available: http://www.sciencedirect.com/science/article/pii/S0019103598958912

[3] S. Okuzumi, H. Tanaka, T. Takeuchi, M. Sakagami. (2011 March). Electrostatic Barrier Against Dust Growth In Protoplanetary Disks. II. Measuring the Size of the "Frozen" Zone. *The Astrophysical Journal* [Online]. 731(96). Available: http://iopscience.iop.org/0004-637X/731/2/96/fulltext/apj_731_2_96.text.html

[4] J. Blum, "Grain Growth and Coagulation," in *Astrophysics of Dust 2003 ASP Conference Series*, Colorado, 2003, pp.369

[5] J. Blum, G. Wurm. (2008, September) Tbe Growth Mechanism of Macroscopic Bodies in Protoplanetary Disks. *Annual Review of Astronomy and Astrophysics* [Online]. 46(21-56). Available: http://www.annualreviews.org/doi/abs/10.1146/annurev.astro.46.060407.145152

[6] S. J. Weidenschilling, J. N. Cuzzi. Formation of Planetesimals in the Solar Nebula. *Protostars and Planets III* [Online]. Available: http://adsabs.harvard.edu/abs/1993prpl.conf.1031W

[7] D. Richardson, "," Ph. D dissertation, Dept. Physics, Oxford Univ., Oxford, UK, 1993

[8] L. Matthews, T. Hyde. (2008, February). Charging and Growth of Fractal Dust Grains. *IEEE Transactions on Plasma Science* [Online]. Available: http://arxiv.org/abs/0711.0196

[9] L. Matthews, R. Hayes, M Freed, T. Hyde. (2007, April). Formation of Cosmic Dust Bunnies. *IEEE Transactions on Plasma Science* [Online]. Available: http://arxiv.org/abs/astro-ph/0701385

[10] E. Perlman, R. Burns, Y. Li, and C. Meneveau. "Data Exploration of Turbulence Simulations using a Database Cluster". Supercomputing SC07, *ACM, IEEE*, 2007.

[11] S. Okuzumi, H. Tanaka, T. Takeuchi, M. Sakagami. (2011 March). Electrostatic Barrier Against Dust Growth In Protoplanetary Disks. II. Measuring the Size of the "Frozen" Zone. *The Astrophysical Journal* [Online]. 731(96). Available: http://iopscience.iop.org/0004-637X/731/2/96/fulltext/apj_731_2_96.text.html