Abstract

Large scale cosmological simulations such as galaxy and star formation are of great interest to cosmologists. These simulations are done by consideration of relativistic, compressible, discontinuous fluids. Many techniques from particle based to adaptive meshing have been explored, each with advantages and disadvantages. Here, the limitations of centroidal Voronoi tessellations with these astrophysical fluids are investigated in great detail, showing that centroidal Voronoi tessellations are suitable for use with these fluids. Furthermore, the introduction of an ad hoc time dependence technique for these centroidal Voronoi tessellations is analyzed and shown to be not useful for large scale simulation. Lastly, a new meshing technique is introduced called the Euler-Lagrange centroidal Voronoi tessellations, resembling standard centroidal Voronoi tessellations but enforcing a centroidal Voronoi tessellations that advects with the fluid.

1 Background

1.1 Motivation

Knowing how matter has evolved through time is a major part of the field of cosmology. Many interesting phenomena occur in the early universe, such as star or galaxy formation. [8] Cosmological parameters of the universe can be discovered by comparison of the model to observation. Therefore, accurate cosmological simulations of matter through long periods of time must be conducted. [1] Due to the complexity of these cosmological systems, often numerical methods techniques are an indispensable tool in correctly modelling their evolution. [7] Many of the dynamic systems of these simulations are astrophysical gas systems involving the motion of compressible fluid with relativistic velocity or energy densities. [6]

Therefore, it is a relevant topic to consider how to simulate relativistic, highly compressible, discontinuous fluids in astrophysics. However, the majority of the work in fluid simulation relates to incompressible fluids. An exploration of past techniques in fluid simulations and their application to the fluids in consideration is done here, along with the specification of some new techniques for the simulation of relativistic, highly compressible, discontinuous fluids.

1.2 Lagrangian Simulations - Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is one of the most commonly used techniques for simulating astrophysical fluids. This method works by subdividing the fluid density distribution into discrete pieces, called particles. These particles have a spatial distance of influence, created by smoothing the fluid using some interpolation function, and referred to as the kernel. This means that the physical properties of any SPH particle can be determined by using the kernel function on the relevant properties of all the particles which lie within the range of the kernel. [7]
However, one of SPH's shortcomings is evident when considering it with turbulence and shock, one of the major processes in star formation. When shocks are introduced, the nature of the use of a smoothing function to determine the physics ends up often smoothing out the shocks and discontinuities actively being considered. [3] To fix this, the kernel of influence must be decreased however this adds increasingly more complexity to the system. To keep the simulations computationally viable and to keep discontinuities in the fluid sharp is an issue that SPH has issues addressing.

Furthermore, SPH has problems handling boundaries, there is no natural way to include boundary conditions directly on SPH formalism. Due to the kernel of influence, boundary conditions become complex to specify. Even simple straight line boundaries under the SPH model become rough, resulting in unphysical boundary jittering and introducing unnatural coarseness. This problem is exacerbated when mediums begin mixing, since the kernel of influence cannot accurately simulate fine structure in the fluid. This introduces smoothing of fluid structure and shocks. This makes SPH not an ideal choice for modelling the compressible astrophysical fluids in question.

1.3 Eulerian Simulations

When particle based simulations do not solve the problem, another approach to consider is a grid-based solution. The simplest of these is the Eulerian grid based fluid simulation which is a regular square grid on the space of the simulation. The position and velocity of the fluid is stored at each corner of the grid. The advantages of the Eulerian grid-based fluid simulation is its simple and fast. The grid is easy to compute and fluid is sampled at regular intervals, allowing the physics to be stated more straightforwardly.
1 BACKGROUND

On the left, a smoke simulation on a regular Eulerian grid. Sampling only on these points can result in loss of small artifacts. On the right, the same smoke simulation on an adaptive mesh. Here, areas of complexity are simulated at higher resolution. (Joe Stam, 1999)

However, it still presents many issues. One of these issues is non-Galilean invariance. Due to the sampling of a regular square grid, fluid mechanics is more accurately simulated when the fluid is moving parallel to the grid, compared to when the fluid moves diagonally. The sampling points along the diagonals of the cells are farther apart, introducing slightly more error the simulation. Secondly, a stationary grid becomes incredibly inaccurate when position updates for the fluid density between time steps moves farther than one cell. This means high velocity fluids in high resolution simulations will be prone to non-physical behavior using stationary grids. Lastly, for use on a cosmological scale, low resolution grids must be used to consider the vast amount of space being considered. These low resolution grids often cannot keep up with shocks and sharp discontinuities in the fluid density distributions and result in smoothing.

To resolve some of the issues of the regular Eulerian grid, adaptive meshing techniques are used. Adaptive meshing techniques will identify the areas of higher density fluids and subdivide these spaces further, allowing for closer modelling of regions with more fluid or physics. This solves many low resolution problems that exist from simulations on a cosmological scale, by allowing low resolution grids for areas of sparse fluids and high resolution grids on areas of dense fluids and complex fluid mechanics. Adaptive meshing still does not fully address the issues of non-Galilean invariance because it still exists on a grid that often favors a shorter sampling direction. Additionally, adaptive meshing techniques are still a static grid, introducing many of the inaccuracies of a static Eulerian grid for high velocity fluids. To solve this issue, a meshing technique that advects with the fluid must be considered. [6]
1.4 Moving Meshes

Moving meshes are a good solution to the issues brought about from use of static meshes. Moving meshes use the current velocity measurements of the fluid on the mesh to update the location of the mesh. The result is a mesh that tends to advect, where the fluid within a cell tends to stay within that cell as it travels through space.

![Example of a good moving mesh update](image1.png)

Example of a good moving mesh update, with all points carrying velocities of similar magnitude. The left is the initial mesh cell, and the right is the cell after the position update.

Using the current velocity of the fluid the new positions of the mesh are updated. This can lead to badly shaped cells if there is turbulence, rotation, or areas of high velocity near areas of low velocity. For example, the following cell can be considered, with the velocities stored at two of the vertices having low velocity and one vertex with high velocity. At the next timestep, when the cell positions are updated, the triangle become obtuse, making it a bad approximation of the continuous space. When physics acts on these cells, unphysical error is introduced to the simulation resulting in bad physics.

![Example of a bad moving mesh update](image2.png)

Example of a bad moving mesh update. Here one of the vertices on the left carries a velocity that is much higher than the other vertices on the cell. After the update, on the right, this point is stretched, creating an obtuse, badly shaped triangle.

One way to fix this issue is to remesh when bad cell shape is identified. A simple remeshing technique is the edge-flipping algorithm. This algorithm works by identifying
parts of the mesh that have badly shaped, or badly acute triangles. A quadrilateral is created by combining this badly shaped triangle with the triangle on the shared long edge. This long edge is then flipped to connect the two shortest points across the quadrilateral. This algorithm is a simple way to turn badly shaped cells into two nicely shaped cells. However, the data stored in these cells when the new cells are produced are an average of the previous badly shaped cells. This introduces a smoothing effect on the mesh that over time introduces a smoothing effect on the simulation. This removes shocks and turbulent effects from the simulations. These averaging effects smooth discontinuities and the fluid density distribution combined with bad cell shapes arising make moving meshes a bad technique for dealing with the compressible astrophysical fluids.

Examples of the edge flipping technique. Cells in the top row contain obtuse triangles that badly approximate continuous space. A common edge is flipped creating two well shaped triangles, but smoothing information between the two new cells.

One way to solve the issue of badly shaped cells is to prevent them from occurring. To investigate solutions that enforce well shaped cells at every timestep, a bit of computational geometry must be considered.

1.5 Voronoi Diagram
A Voronoi diagram is a subdivision of space into neighborhoods based on proximity to generators. These generators are specified beforehand, and for each generator there is a corresponding region of points that are closest to a certain generator. These regions become the Voronoi cells. Qualitatively, given a set of initial sample points, cells are created by including all the points that closest to that sample point. These Voronoi cells are arbitrary polyhedra in 3D space and arbitrary convex polygons in 2D space.
Voronoi diagrams have some use and interesting properties. Firstly, each Voronoi diagram has a dual graph that is a Delaunay triangulation for the same points. A Delaunay triangulation is the triangulation created by a given set of points that maximizes the minimum internal angle of all the triangles. This is done by drawing a circumcircle for each of the triangles in the triangulation such that no points are inside the circumcircle of another triangle. Qualitatively, a Delaunay triangulation is one that avoids slivers, or skinny (or badly shaped) triangles.

Non Delaunay triangulation (left) compared to a Delaunay triangulation (right) using the circumcircle technique.

However, Voronoi diagrams are static, making them unsuitable for the same reasons other static grids are unsuitable. A technique utilizing Voronoi diagrams has to be dynamic.

1.6 Moving Voronoi Meshes
An improvement to the Voronoi tessellation; a unique decomposition of space into polyhedra based on a set of mesh-generating points, is to apply the same moving mesh techniques
to this now well shaped mesh. When moving, the Voronoi tessellation adjusts its shape gradually and continuously so that the mesh does not become distorted as it evolves. [6, 9]

While the Voronoi tessellation does keep continuous changes to prevent badly shaped cells, their tendency is to produce small cells. Areas of shearing or compression will lead to areas of rapidly decreasing cell sizes. When the cells become too small, floating point error introduces non-physical artifacts into the simulation and sometimes even lead to exceptions. This makes moving Voronoi meshes unsuitable for the compressible astrophysical fluids under consideration.

1.7 Centroidal Voronoi Tessellations

A centroidal Voronoi tessellation (CVT) is a weighted Voronoi mesh that will enforce cell shape depending on the fluid density distribution. A common way of creating these CVTs is by use of Lloyd’s algorithm, described in detail in the next section. Nonetheless, CVTs are used to keep nicely shaped cells and keep them from shrinking too small. This is succinctly stated as Gersho’s conjecture, which states asymptotically speaking, all cells of the optimal CVT, while forming a tessellation, are congruent to a basic cell which depends on the dimension. [5] This is proven in two dimensions to be regular hexagons and in one dimension these are reasonably sized line segments.

CVTs where the centroids are the center of mass of the cells, uphold the variational condition:

\[
\min \sum_{\text{all cells}} \int_{\text{single cell}} \rho(x)(x - \tilde{x})^2 \, dx
\]

where \( \tilde{x} \) is the cell centroid. This implies that as the mass of a cell increases, the size of the cell must decrease by inverse square. In 1D, this minimum translates to an equipartition of this energy or:

\[
\int_a^b \rho(x)(x - \tilde{x})^2 \, dx = \text{constant}
\]

where \( a \) and \( b \) are the endpoints of a one dimensional cell and \( \tilde{x} \) is the centroid of the cell. When this condition is satisfied, the minimum energy has been met, which is the point of convergence and stability for a CVT. This means the quality of convergence can be determined by looking at the relationship between the size of the cells and the mass contained within each of the cells. When considered on a log-log plot, the relationship between mass contained in a cell and the size of a cell should yield a slope of \(-2\).
1.8 Lloyd’s Algorithm

CVTs seem to be a viable candidate for simulation of these astrophysical fluids, however their creation has yet to be discussed. A common way to create a CVT is by convergence of Lloyd’s algorithm. Lloyd’s algorithm as an iterative algorithm for finding an evenly-spaced set of weighted points.

Lloyd’s algorithm is initialized with a set of points, or generators. From these generators, an initial set of boundaries are created. At each iteration the centroid of the cell is determined, and the new Voronoi boundaries are recomputed. Each time this relaxation step is performed, the generators are slightly more evenly spaced. Areas of high mass pull the generators in, while areas of low mass push the generators out. The algorithm converges slowly and convergence is guaranteed in one dimension. Furthermore, these iterations are completed once the termination criterion is reached. One common termination criterion is to stop when the maximum distance moved by any site in an iteration falls below a preset threshold. [4]

As stated, the variational condition for CVTs creates a relationship between size of cell and mass enclosed in the cell. This is used to mathematically determine if Lloyd’s algorithm has reached complete convergence. The expected behavior for these cells as the mesh approaches convergences to CVT, the mass enclosed and size of cell should approach inverse square. Lloyd’s algorithm was implemented on a one dimensional CVT and the relationship of the masses and sizes of cells monitored at different points before completion of Lloyd’s algorithms. The following log-log plots were taken, the left before convergence and the right after convergence.

On the left the log-log plot of mass vs length of a Voronoi mesh that has yet to reach convergence. On the right a log-log plot of mass vs length of a mesh that has converged. The red line is the $-2$ slope for comparison.
Using this relationship and Lloyd’s algorithm, CVTs can be created by an iterative update of the centroids of cells and be judged for accuracy by this log mass vs log length relationship.

2 Research Questions

This is where the work addressed in the thesis begins. CVTs are a good candidate for a viable meshing technique for high velocity fluids with many discontinuities, making CVTs a good start for exploring how to model the relativistic, highly compressible astrophysical fluids in consideration on a cosmological scale.

2.1 Can CVTs guarantee accuracy on compressible, high velocity fluids?

Not much work has been done thoroughly researching the limitations of CVTs with these types of fluids. This thesis will address under what physical regimes CVTs can be guaranteed to be accurate. This means the CVTs will have to give well-shaped and sized cells, even for sparse or highly discontinuous fluid density distributions. This includes analysis of many techniques for discretizing space to form the CVTs across many different types of fluids.

2.2 How can CVTs be made to advect with the fluid?

Lastly, CVTs alone are static meshes, meaning they are prone to the errors introduced from using a static meshing technique. In this thesis, a moving mesh approach and an Eulerian-Lagrangian hybrid approach are used to allow CVTs to advect with the fluid, reducing the error brought about by the static mesh.

3 Simulation

To address these research questions, a fluid simulation was implemented. The simulation is defined on a one dimensional closed space, or a circle. The Voronoi cells are represented as line segments on this circle, with the boundaries corresponding to points. Initially, the Voronoi generators are evenly sampled across the simulation space. The cell boundaries are then computed, place at the midpoint between each of the already placed Voronoi generators.

The update of the Voronoi generator positions is done by finding the centroid of the cell. The exact centroid of a cell in this context is simply its center of mass. However, as it has been discussed, the calculation of the exact centroid for a Voronoi mesh can be difficult or unfeasible for some density distributions. Therefore, discretization and interpolation schemes must be used to calculate the center of mass of the cell.
3 SIMULATION

The simulation renders the Voronoi cells as line segments on this circle, with black points representing the Voronoi generators, and pink points representing the Voronoi cell boundaries. The generator points are rendered with a certain radius corresponding to the mass contained within the cell. When the density distribution is moving in time, it cannot be seen directly, but is visualized through the changing masses and sizes in the Voronoi cells.

Simulation screenshot, displaying the 1D, circular space from -1 to 1, where 0 is at the top of the circle and 1 is at the bottom. Black points are Voronoi generators and pink points are Voronoi boundaries. This is taken on a density distribution corresponding to $\rho(x) = 5x^2$ and with 50 Voronoi cells.

The distance between the pink points, Voronoi boundaries represents how small or large the cells are with the black points, generators, representing the mass enclosed in the cell.
3.1 Fluid Density Distributions

The full extent of fluid mechanics is not implemented in this simulation. Analytic density functions are imposed on the space with an artificial time dependence to further analyze the effects of the density distributions on the Voronoi meshes. This space in the simulation is defined from $-1$ to 1, where $-1/1$ are mapped to the same point, creating a circle of circumference 2. Example fluid density distribution functions used in this simulation are:

- $\rho(x) = \sin 2\pi(x + t) + 2$;
- $\rho(x) = 5x^2$;
- $\rho(x) = x + 1$; and
- $\rho(x) = (x + 1 + t) \mod (2)$.

The $\rho(x) = \sin 2\pi(x + t) + 2$ distribution is chosen because it is continuous and positive across the entire space. Also, this fluid distribution moves at a constant velocity. This function is an excellent candidate for showing if the proposed experiment works in the ideal fluid scenario.

The $\rho(x) = 5x^2$ is chosen because it is also continuous on the whole space, and it is static. However, this fluid distribution has a point of 0 fluid density. This density function is used to test how the simulation handles areas of sparse fluid, where there could be points of 0 density.

The $\rho(x) = x + 1$ fluid distribution is used because it is discontinuous. Because the simulation space is described from $-1$ to 1 where both $-1$ and 1 are mapped to the same point, this point has a discontinuity. This density function is useful for determining how the simulation handles sharp discontinuities in the fluid, which is a common characteristic of the fluids in astrophysical simulation. Lastly, the $\rho(x) = (x + 1 + t) \mod (2)$ fluid density distribution is used because it has a moving discontinuity. This sharp discontinuity moves at a constant velocity and is useful for representing how the simulation would handle shocks and dynamic points of 0 density.

\begin{align*}
\rho(x) &= 5x^2 \\
\rho(x) &= \sin(2\pi x) + 2 \\
\rho(x) &= x + 1
\end{align*}

Representation of the density distributions on the simulation space where the larger points represent higher fluid density.
4 Experiment and Results

4.1 Discretization

For Lloyd’s algorithm to converge to a well-shaped CVT, a centroid has to be chosen to update the Voronoi tessellation. Because of the discrete nature of a computer simulation, this requires a discretization of the fluid density distribution across the Voronoi cells. How to discretize this space to determine the centroid can be done in several ways. To ensure the accuracy of the simulation some requirements are placed on the discretization: when the CVT is created the cells should be of equal mass, preserve discontinuous structure, and be quick to compute. It can be difficult to see how cells could simultaneously obey the variational condition and have cells of equal mass. In fact, they cannot. However, if all the cells in the CVT obey the variational condition, then a mapping from the actual fluid density distributions to one that would yield a CVT with equal mass cells is simple to create. Therefore, qualitatively, the resulting CVT should resemble the fluid density distribution, and discontinuous effects in the distribution should still be seen in the CVT. These fit onto the simulation space as follows:

4.2 Exact Center of Mass

The exact center of mass discretization is impossible to ascertain for realistic fluid mechanics. When the fluid density distributions are defined analytically, the exact center of mass is simple to calculate. Despite this being an unrealistic discretization method for real fluid simulations, examining the behavior of this exact solution is crucial to determining the accuracy and realism of other discretization methods.

The calculation of the exact center of mass of the Voronoi cell can be calculated given the continuous fluid density distribution function. For a prescribed $\rho$ in one dimension, the center of mass can be calculated directly:

$$\tilde{x}_{new} = \frac{\int_a^b x \rho(x) dx}{\int_a^b \rho(x) dx}$$

where $\tilde{x}_{new}$ is the center of mass, $a$ and $b$ are the boundaries of the cell, and $\rho$ is the continuous density function. The Voronoi generators were updated to this center of mass.
The first of the density distributions considered was $\rho(x) = 5x^2$. After Lloyd’s algorithm has stabilized (more iterations do not give any changes to the cell boundary placements) the resulting CVT is:

![CVT created by the center of mass discretization for the 5x^2 distribution.](image)

Notice the larger cells are near the top of the circle, where $\rho(x)$ is close to 0 and the smaller cells are toward the bottom of the circle, where $\rho(x)$ is at its maximum. This shows that the CVT is resembling the fluid density distribution. Also notice there is a Voronoi boundary lying at 0 that does not introduce any artifacts into the CVT.
Next the fluid density distribution of \( \rho(x) = \sin(2\pi x) + 2 \) was considered.

Again, the CVT nicely resembles the fluid density distributions with large cells in the negative regimes of \( \sin(2\pi x) \), corresponding the minimum \( \rho(x) \) and smaller cells in the area near \( x = 0.5 \), the maximum \( \rho(x) \).
Lastly, the fluid density distribution of $\rho(x) = x + 1$.

The discontinuity at $x = 1$ corresponding to $\rho(x) = 0$ and $\rho(x) = 2$, does not seem to affect the CVTs conformity to the fluid density distribution. The use of CVTs on fluid density distributions that are not $C_1$ continuous has not been thoroughly researched, so Lloyd’s algorithm not converging is revolutionary fact. This allows the further pursuit of CVTs with more realistic discretization schemes.

To map these to a density distribution where all the cells have equal mass, the dependence on mass of cells versus size of cell needs to be considered.
The log mass versus log length plot for the exact center of mass discretization using 100 cells and allowing Lloyd’s algorithm to run for 10000 iterations. The red dashed line marks a slope of $-2$.

By the previous plot it can be seen that nearly all of the points lay on the slope of $-2$. This shows that the Lloyd’s algorithm is creating a CVT obeying the variational condition for CVT. Using this variational condition, each of the fluid density distributions are mapped to:

\[
\tilde{\rho}(x) = \rho(x)(b - a)^2
\]

where $\tilde{\rho}(x)$ is the density function described on the CVT after it has been created, $\rho(x)$ is the real fluid density, and $b$ and $a$ are the boundaries of the cell. After doing this mapping, the new distributions look like:
The CVTs for each of the distributions when normalized to the $\tilde{\rho}$ distribution where each cell encloses the same mass.

Here the CVTs show nearly all cells of equal mass, exactly as desired for the various fluid density distributions. By recognition of the outlier in the log-log plot, the point of lowest mass and largest cell size, the CVT may not produce cells of equal mass in areas of sparse fluid densities. This needs to be carefully considered when dealing with sparse fluids, as this could introduce some unwanted error into the simulation.

4.3 Constant

A constant discretization is determined by sampling the density distribution at the Voronoi generators. This results in a piecewise constant discretization, with discontinuities existing on each of the Voronoi cell boundaries.

Discretization of the constant scheme over a cell and its neighbors.
By considering the center of mass of a cell where the mass is constant across the cell and the density is sample at the midpoint, $m = \frac{1}{2}(a + b)$, the centroid can be determined by:

$$\bar{x}_{\text{new}} = \frac{\int_{a}^{b} x \rho(m) dx}{\int_{a}^{b} \rho(m) dx} = \frac{\rho(m) \frac{1}{2}(b^2 - a^2)}{\rho(m)(b - a)} = \frac{a + b}{2}$$

By this short derivation, it is evident that the new generator will always be updated to the midpoint of the cell. Therefore, on the entire execution of the Lloyd’s algorithm, it stabilizes with all of the cells being the same size, despite the initial sample distribution, all with size:

$$l = \frac{L}{n}$$

where $L$ is the entire length of the region, $n$ is the number of initial points, and $l$ is the stabilized size of each Voronoi cell upon the completion of Lloyd’s algorithm.
distribution at all, making this discretization unsuitable for these fluid simulations. While this is a fast discretization scheme, this does not offer any of the benefits of using CVTs.

4.4 First Order Finite Volumes

The first order finite volumes discretization utilizes sampling at the boundaries of the Voronoi cell. The first order finite volumes method creates a finite length surrounding the cell boundaries at the same value sampled. These samples are assumed to be constant from the boundary to the Voronoi center.

To consider this in the simulation, note that the boundary $a$ and $b$ correspond to density samples of $\rho(a)$ and $\rho(b)$. The current generator location in a cell is denoted as $\tilde{x}$, the Voronoi centroid between $a$ and $b$. Under this discretization, the segment from $a$ to $\tilde{x}$ has density equal to $\rho(a)$ and from $\tilde{x}$ to $b$ equal to $\rho(b)$.

Discretization of the first order finite volumes scheme over a cell and its neighbors.

Using this interpolation, the updated centroid of a cell is defined as:

$$\tilde{x}_{\text{new}} = \frac{\rho(a)(a + \frac{l_1}{2}) + \rho(b)(b + \frac{l_2}{2})}{\rho(a)l_1 + \rho(b)l_2}$$

where $l_1$ is the distance from $a$ to $\tilde{x}$ and $l_2$ is the distance from $\tilde{x}$ to $b$.

Notice this is equivalent to finding the center of mass of two point masses each located halfway between $\tilde{x}$ and the Voronoi boundary, as seen in the following figure.
The two body problem with a solution identical to first order finite volumes.

Solving the center of mass of this system where each red circle has mass defined as the density sampled at the boundary times the length between the boundary to the Voronoi center gives the same center of mass as the first order finite volumes method.

Using this with the \( \rho(x) = \sin(2\pi x) + 2 \) gave an interesting distribution.

Unlike the constant discretization, it is evident that the Voronoi cells are roughly conforming to the fluid density distribution. However, by comparison of this CVT to the exact discretization, it is clear that this is not the same CVT.
The $\rho(x) = x + 1$ fluid density distribution gave a similar result. 

CVT created by the first order finite volumes discretization for the $x + 1$ distribution. 

The Voronoi cells are conforming to the fluid density distribution, and it is different from the exact discretization. However, the discontinuity at $x = 1$ is very well pronounced and smoothed over a minor amount compared to the exact case.
Lastly, $\rho(x) = 5x^2$ distribution presented a troubling result.

The unstable CVT created by the first order finite volumes discretization for the $5x^2$ distribution contained forever growing cells.

This fluid density distribution never reaches convergence on a CVT. Because of the Voronoi boundary at $x = 0$ with a $\rho = 0$ the two large cells continue to increase in size until the cells contain the entire space and are no longer defined on the simulation space. This discretization function cannot be used for any fluid distribution that can become sparse because one Voronoi boundary falling into a region of 0 mass will disrupt the entire simulation.
To investigate the differences between this discretization and the exact one, the log mass vs log length plots were considered.

\[ \rho = x + 1 \quad n = 100 \quad i = 100000 \]

The log mass versus log length plot for \( \rho(x) = x + 1 \) and 100 cells and 10000 Lloyd iterations. The red dashed line shows a slope of \(-1\).

The cells sizes and masses considered on a \( \log_{10} \) length vs \( \log_{10} \) mass plot, does not yield a slope of \(-2\) but rather \(-1\) implying an inverse relationship between length and mass. The piecewise constant regions of the mesh are between the cell centers \( \tilde{x}_i \) to \( \tilde{x}_{i+1} \) where \( i \) and \( i + 1 \) denotes two adjacent cells. The minimization of the energy between two adjacent cells becomes:

\[
(4.6) \quad \rho(a)(\tilde{x}_i - \tilde{x}_{i-1})^2 = \rho(b)(\tilde{x}_{i+1} - \tilde{x}_i)^2
\]

Notice that this predicts a different power law relating mass and length of Voronoi cells than the other discretization methods. This shows that mass (length x \( \rho \)) by length is a constant. Therefore, the \(-1\) slope seen in the log-log plots is theoretically predicted. This implies that the resulting stabilized mesh is not the same as the stabilized mesh for the exact centroid calculations. In this example, which is representative of the other fluid density distribution functions, the \(-1\) slope implies a mapping from the real fluid density
distribution to the one that where this CVT contains cells of equal mass and can be given by:

\begin{equation}
\tilde{\rho}(x) = \rho(x)(b - a)
\end{equation}

where \( \tilde{\rho}(x) \) is the density function described on the CVT after it has been created, \( \rho(x) \) is the real fluid density, and \( b \) and \( a \) are the boundaries of the cell. After applying this mapping, the distributions have cells containing equal mass.

Nonetheless, this interpolation still yields convergence as long as no Voronoi boundaries land on a point of zero density. If one of the boundaries falls onto a region of zero density, this creates an instability because the centroid keeps being moved by \( \frac{a}{l^2} + \frac{1}{2} \) yielding a bad mesh.

### 4.5 Piecewise Linear Interpolation

The next simplest discretization scheme to consider is the piecewise linear interpolation. This discretization is done by sampling the density at the boundary of the Voronoi cell and linearly interpolated across the cell.

In one dimension, the boundary of the Voronoi cell are two points \( a \) and \( b \). The density can be described by the function \( p(y) \):

\begin{equation}
p(y) = y\rho(a) + (1 - y)\rho(b)
\end{equation}
where $y$ exists between $a$ and $b$ and is 1 at $x = a$ and is 0 at $x = b$. Using this new interpolated density function $p$, the center of mass of the cell can calculated:

$$\bar{x}_{\text{new}} = \frac{\int_{a}^{b} yp(y)dy}{\int_{a}^{b} p(y)dy}$$

where the resulting $\bar{x}_{\text{new}}$ is the centroid of the Voronoi cell. The $p$ function here does not depend on the shape of $\rho$, only the values at the boundary of the Voronoi cell, therefore this can be further simplified to:

$$\bar{x}_{\text{new}} = \frac{-2a^3(\rho(a) - \rho(b)) - 3a^2\rho(b) + b^2(2b(\rho(a) - \rho(b)) + 3\rho(b))}{3(a - b)(a(\rho(a) - \rho(b)) + b(\rho(a) - \rho(b)) + 2\rho(b))}$$

While this equation is not necessarily elegant, it does only depend on the Voronoi boundaries, making it quick to compute. Since this interpolation does not depend on the analytic form of the density functions, this is a viable interpolation scheme for realistic fluid mechanics simulations. Each iteration of Lloyd’s algorithm moved the center toward the more massive side of the cell, distributed linearly. Stabilization of Lloyd’s algorithm resulted in a distribution of larger Voronoi cells in areas of low mass and smaller cells in the region of high mass, closely reflecting the density distribution.
The $\rho(x) = \sin(2\pi x) + 2$ distribution displayed expected behavior with this discretization scheme.

The Voronoi cells are conforming to the given fluid density distribution, as desired. This Voronoi mesh more closely resembled the exact solution than the first order finite volumes method.
Unlike the first order finite volumes method, the $\rho(x) = 5x^2$ distribution was convergent. CVT created by the piecewise linear interpolation for the $5x^2$ distribution. Here, the regions of 0 density did not adversely affect the simulation, because these 0 density areas corresponded to only one point rather than to an entire area. It is worth saying here, however, that this method will break down if both of the boundary points to a cell sample at 0 density. This would create a cell that encloses 0 mass which is not defined in a CVT at all, and would cause an exception in this discretization scheme.
Lastly, the $\rho(x) = x + 1$ fluid density distribution was considered.

CVT created by the piecewise linear interpolation for the $x + 1$ distribution. Using the discontinuous fluid density distribution this method converged. Due to the sampling on the Voronoi boundary, subtle discontinuous effects within the cell were lost. However, larger discontinuous effects are seen, and treated as a cell of high mass next to a cell of low mass, smoothing out the discontinuity. If the cell boundary falls directly on the point of discontinuity, the smoothing effect is minimal, and convergence is still reached.

To consider quantitatively of how this compares to the exact discretization, the log mass versus log length plots are again analyzed.
The log mass versus log length plot for $\rho(x) = 5x^2$ with 100 cells and 10000 Lloyd iterations. The red dashed line marks a slope of $-2$.

This shows a linear relationship of $-2$ between log mass and log length of cells. This is the same relationship upheld by the variational condition. This means CVTs generated by Lloyd’s algorithm using this discretization scheme has many of the same properties as the exact solution. The same mapping from $\tilde{\rho}$ as the exact discretization works for this discretization as well. Furthermore, the discontinuity has minimal smoothing. This makes the linear interpolation discretization scheme a fast and reliable method for generating CVTs for the fluids under consideration.

At this point the limitations of the static CVTs have been explored for the astrophysical fluids. Here three different discretizations schemes have been investigated for determining the centroids of CVTs for Lloyd’s algorithm. The piecewise linear interpolation is the lowest order discretization that displays the characteristics of the exact solution. Other discretizations schemes displayed behavior that would make them ineligible for use with discontinuous fluid distributions. The linear interpolation discretization preserves discontinuities, obeys the same variational condition as the exact solution, and conforms to the fluid density distribution. Despite this correctness, these static CVTs when simulating a dynamic fluid are prone to generating diffusive error, implying the need for dynamic meshing techniques to be considered.
5 Moving CVTs

So far, the fluid density distributions under consideration have been static. This is not representative of a real fluid simulation because fluids are not stationary; the density distribution depends on time. The first way to attempt to get CVTs to converge, conform to the fluid distribution, and to preserve discontinuities is to simply run Lloyd’s algorithm to convergence between each timestep.

The $\rho(x) = (x + 1 + t) \mod (2)$ distribution after a few timesteps have been taken using stabilized time dependence.

This solution is guaranteed to work, since it was previously shown that the CVTs do converge and uphold all the desired properties, and here time dependence is being treated as just a repetition of the static case. This does work well, however some smoothing is introduced on the boundary for long time scales, as seen in the previous figure. The main issue with this approach is its time complexity. This is a computationally demanding flow because a high resolution mesh with a large number of Voronoi cells will take many iterations of Lloyd’s algorithm to reach convergence, and this will need to be completed before each timestep in the fluid simulation is taken. This time dependence method was referred to as stabilized.
5.1 Adhoc Time Dependence

To allow the mesh to perfectly stabilize between each time step of the density means to allow a full convergence of Lloyd’s algorithm. Realistically, the stabilized time dependence method is too expensive for use in real simulation. Another time dependence method is to only partially run Lloyd’s algorithm before incrementing the time step. This is called adhoc time dependence.

Adhoc time dependence choses the number of iterations of Lloyd’s algorithm to run between each timestep. This was explored on the $\rho(x) = (x + 1 + t) \mod 2$ distribution. Different values of $i$, representing the number of Lloyd iterations before incrementing the timestep. These simulations were all run for 100 timesteps then the log-log plots were captured.
Dependence of the CVT correctness to the number of Lloyd iterations taken between timesteps for $\rho(x) = x + 1$ and 1000 cells.

Using this method, the minimum number of iterations Lloyd’s algorithm takes to achieve minimum convergence is different for each density distribution, the resolution of the Voronoi mesh, and the velocity of the fluid. Minimum convergence is qualitatively defined as the least accurate a Voronoi mesh can be without inducing error or artifacts into the simulation. As seen in these diagrams, a low number of iterations lead to CVTs with no real dependence on fluid density distribution. To achieve CVTs which qualitatively conform to the fluid density distribution for 1000 cells, 150000 iterations of Lloyd’s algorithm had to be used. At this point, large differences in the number of iterations used minimally affected the accuracy of the CVT. For example, the differences between 1 million iterations and 2 million iterations on the approach to convergence is almost negligible.

This creates a tradeoff between execution time and Voronoi mesh accuracy. The more Voronoi cells in the mesh, the more iterations required to reach minimum convergence. Furthermore, distributions that are not monotonic typically take more iterations to reach convergence. These must be considered when determining the minimum number of iterations to use with the unstabilized time dependence. This was an incredibly ad hoc method.
with no real formulation, making it unsuitable for considering dynamic fluid distributions.

6 Euler-Lagrange CVTs

The moving mesh approach includes a rather ad hoc method of choosing the number iterations for convergence, introducing a direct tradeoff between computation time and accuracy. An idea that is somewhat unexplored is to introduce a variational condition on the speed of the cells, analogous to the existing variational condition on the size of the cells. By using the calculus of variations and treating this variational formula as an energy, Hamilton’s principle of least action can be invoked. Hamilton’s principle of least action states that the minimum of the action of a system will yield the equations of motion for the system. Where action is defined as:

\[ S = \int_{t_1}^{t_2} L(q(t), \dot{q}(t), t) dt \]

(6.1)

where \( L \) is the lagrangian, \( q \) is position, dot denotes a time derivative, and \( t \) is time. The minimum of this translates to:

\[ \delta S = \frac{d}{ds} \int_{t_1}^{t_2} L(q(t) + s\delta q, \dot{q}(t) + s\delta \dot{q}, t) dt = 0 \]

(6.2)

Solving this equation using the variational condition as the lagrangian gives a set of forces acting on the mesh at each timestep that will enforce cells of equal mass and cells with velocity similar to the fluids.

6.1 Continuous

This variational condition is an energy. It has a potential-like term, depending on position, and a kinetic-like term, depending on velocity. The potential-like term of this energy is an energy that encourages well shaped cells, and penalizes major deformations to the cell. The kinetic-like term of this energy is an energy that encourages the Voronoi cells to move with the fluid. This is important to avoid fluid flowing through the cells and to enforce moving mesh to resemble the physics of the fluid. The notation for the Euler-Lagrange energy is expressed as follows:

\[ E(t) = \sum_{i=1}^{n} \alpha \int_{B_i(\tilde{x})} \rho(x, t)(x - \tilde{x}_i)^2 dx + \beta \int_{B_i(\tilde{x})} \rho(x, t)(v - \tilde{v}_i)^2 dx \]

(6.3)

Each \( B_i(\tilde{x}) \) is the \( i^{th} \) Voronoi cell. The \( \tilde{x} \) and \( \tilde{v} \) are the Voronoi generator position and velocity, respectively. The \( x \) and \( v \) terms are the fluid position and velocity. \( \rho(x, t) \) is the fluid density at a position \( x \) and a time \( t \).
To derive equations of motion from this energy, Hamilton’s principle of least action is invoked, stating that the minimum of the energy over time will correspond to the optimal physical path.

\[
\text{min} \int_0^T E(\ddot{x}, \dot{\bar{v}}, t) \, dt
\]

(6.4) The minimum is taken by finding the critical points of this action quantity. To take the derivative of such a quantity, the total change in energy caused by a small perturbation some \( \delta(\ddot{x}) \) off of the \( \ddot{x} \) is considered.

\[
\left. \frac{d}{ds} \int_0^T E(\ddot{x} + s\delta \ddot{x}, \dot{\bar{v}} + s\delta \dot{\bar{v}}, t) \, dt \right|_{s=0} = 0
\]

(6.5) where \( T \) is some ending time far in the future.

This equation is evaluated at \( s = 0 \) to find the \( \ddot{x} \) which corresponds to the critical point.

\[
\int_0^T \left[ \frac{\partial}{\partial \ddot{x}} E(\ddot{x} + s\delta \ddot{x}, \dot{\bar{v}}, t) \cdot \delta \ddot{x} + \frac{\partial}{\partial \dot{\bar{v}}} E(\ddot{x}, \dot{\bar{v}} + s\delta \dot{\bar{v}}, t) \cdot \delta \dot{\bar{v}} \right] \, dt \bigg|_{s=0} = 0
\]

(6.6) For concise notation, \( D_1 = \frac{\partial}{\partial \ddot{x}} \) and \( D_2 = \frac{\partial}{\partial \dot{\bar{v}}} \).

The \( \ddot{x} \) that corresponds to the critical point will solve \( D_1 E(\ddot{x} + s\delta \ddot{x}, \dot{\bar{v}}, t) \delta \ddot{x} = 0 \), so we can set the \( D_1 E(\ddot{x} + s\delta \ddot{x}, \dot{\bar{v}}, t) \) = 0 for all \( t \). This is an issue due to the \( \delta \dot{\bar{v}} \) term that needs to be an \( \delta \ddot{x} \). To solve this, integration by parts is used on this term.

\[
\int_0^T \left[ D_1 E(\ddot{x} + s\delta \ddot{x}, \dot{\bar{v}}, t) \cdot \delta \ddot{x} - \frac{d}{dt} D_2 E(\ddot{x}, \dot{\bar{v}} + s\delta \dot{\bar{v}}, t) \cdot \delta \ddot{x} \right] \, dt + D_2 E(\ddot{x}, \dot{\bar{v}} + s\delta \dot{\bar{v}}, t) \cdot \delta \ddot{x} \bigg|_0^T = 0
\]

(6.7) When this is evaluated at \( s = 0 \) the \( \delta \ddot{x} \) is lost and \( \delta \dot{\bar{v}} \) terms. The last term outside of the integral also goes to 0 because at 0 and \( T \) the \( \delta \ddot{x} \) is required to be 0 at these points since the end points cannot move, as part of Hamilton’s principle of least action.

\[
\int_0^T \left[ D_1 E(\ddot{x}, \dot{\bar{v}}, t) - \frac{d}{dt} D_2 E(\ddot{x}, \dot{\bar{v}}, t) \right] \cdot \delta \ddot{x} \, dt = 0
\]

(6.8)
To satisfy the above equation, \( D_1 E(\tilde{x}, \tilde{v}, t) - \frac{d}{dt} D_2 E(\tilde{x}, \tilde{v}, t) \) must be 0, which can be rearranged into the form:

\[
(6.9) \quad D_1 E(\tilde{x}, \tilde{v}, t) = \frac{d}{dt} D_2 E(\tilde{x}, \tilde{v}, t)
\]

Now the evaluation of this is tricky since both internal integral and the boundary of integration depends on \( \tilde{x} \).

To deal with this, the Reynold’s transport theorem is used, as stated below:

\[
(6.10) \quad \frac{d}{dx} \int_{\Omega(x')} f(x, y) dy = \int_{\Omega(x')} \frac{\partial}{\partial x} f(x, y) dy + \oint_{\partial \Omega} \frac{\partial}{\partial x} \partial \Omega \cdot \hat{n} f(\partial \Omega)
\]

The Reynold’s transport theorem states the the derivative of an integral where the derivative depends on the boundary of integration can be considered as the sum of change of the function within the area of integration plus however much the function changes on the boundary of the area of integration.

The remaining terms are evaluated one at a time for conciseness. At this juncture, the one dimensionality of the simulation is considered.

\[
(6.11) \quad D_2 E(\tilde{x}, \tilde{v}, t) = \beta \int_{B_i(\tilde{x})} 2\tilde{v}_i \rho(x, t) - 2v \rho(x, t) dx
\]

Assuming a one-dimensional space, the cell boundaries are points in space notated by \( B_{i-\frac{1}{2}} \) and \( B_{i+\frac{1}{2}} \) and the generators are \( \tilde{x}_{i-1}, \tilde{x}_i, \tilde{x}_{i+1} \), where \( \tilde{x}_i \) is the Voronoi generator of the \( i \)th cell and the \( \tilde{x}_{i-1} \) is the generator sharing the \( B_{i-\frac{1}{2}} \) boundary and similarly on the positive side. For convention-sake it is taken that \( \hat{n} \) points away from boundary. In this convention, a movement in \( \tilde{x}_i \) will move \( B_{i-\frac{1}{2}} \) change by \( \frac{1}{2} \) of this distance, since remeshing places the generators directly between the new boundaries. So, \( B_{i+\frac{1}{2}} = \frac{x_{i+1} + x_i}{2} \) and \( \frac{d}{\tilde{x}_i} B_{i+\frac{1}{2}} = \frac{1}{2} \).

The next term invokes the Reynolds transport theorem, evaluating points on the boundary. Since the boundary in this 1D space are just points, the boundary integral from the Reynold’s transport theorem becomes simply difference of functions evaluated on these points.

\[
(6.12) \quad D_1 E(\tilde{x}, \tilde{v}, t) = \beta \int_{B_i(\tilde{x})} 2\tilde{v}_i \rho(x, t) - 2v \rho(x, t) dx \quad + \left[ \frac{1}{2} \rho(B_{i+\frac{1}{2}}, t) \cdot (B_{i+\frac{1}{2}} - \tilde{x}_i)^2 - \frac{1}{2} \rho(B_{i-\frac{1}{2}}, t) \cdot (B_{i-\frac{1}{2}} - \tilde{x}_i)^2 \right. \\
- \left. \frac{1}{2} \rho(B_{i+\frac{1}{2}}, t) \cdot (B_{i+\frac{1}{2}} - \tilde{x}_{i+1})^2 + \frac{1}{2} \rho(B_{i-\frac{1}{2}}, t) \cdot (B_{i-\frac{1}{2}} - \tilde{x}_{i-1})^2 \right]
\]
Physically, the first two of the bracketed terms are interpreted as the effect on the boundary when moving the \( \tilde{x}_i \) generator. The second two terms in the bracket come about because the movement of the \( \tilde{x}_{i-1} \) and \( \tilde{x}_{i+1} \) causes the boundary shared with \( \tilde{x}_i \) to move. These all cancel because the \( \tilde{x}_{i-1} \) and \( \tilde{x}_{i+1} \) cause this change in the boundary to be exactly the opposite of the movement of the \( \tilde{x}_i \) generator, due to the generators being equidistant from the boundary. This same argument can be made for the right boundary as well.

The only remaining term to be expanded is the time derivative of the \( D_2 \) term.

\[
\frac{d}{dt} D_2 E(\tilde{x}, \tilde{v}, t) = 2\beta \frac{d}{dt} \int_{B_i(\tilde{x})} \tilde{v}_i \rho(x, t) - v \rho(x, t) dx \\
= 2\beta \int_{B_i(\tilde{x})} \tilde{v}_i \rho(x, t) + \tilde{v}_i \dot{\rho}(x, t) - \dot{v} \rho(x, t) - v \frac{d}{dt} \rho(x, t) dx \\
+ \left[ \frac{1}{2}(\tilde{v}_i + \tilde{v}_{i+1}) \cdot (\tilde{v}_i \rho(B_{i+\frac{1}{2}}, t) - v \rho(B_{i+\frac{1}{2}}, t)) \\
- \frac{1}{2}(\tilde{v}_i + \tilde{v}_{i-1}) \cdot (\tilde{v}_i \rho(B_{i-\frac{1}{2}}, t) - v \rho(B_{i+\frac{1}{2}}, t)) \right]
\]

(6.13)

The average of \( \tilde{v}_{i+1} \) and \( \tilde{v}_i \) comes from \( \frac{d}{dt} B_{i+\frac{1}{2}} \).

Once familiarized with the physical interpretation of the continuous Euler-Lagrange derivation, this is discretized to determine update rules for \( \tilde{x}, \tilde{v}, \) and \( \tilde{\dot{v}} \).

### 6.2 Discrete

This continuous derivation gives the forces that act on the mesh to enforce a well shaped and well moving mesh. These forces can be directly translated into time integrators by repeating the derivation on a discrete lagrangian formulation.

\[
E_{\text{tot}} = \sum_i \alpha \int_{B_i(\tilde{x})} \rho(x, t)(x - \tilde{x}_i)^2 dx + \beta \int_{B_i(\tilde{x})} \rho(x, t)(v - \tilde{v}_i)^2 dx
\]

In the discretized Euler-Lagrange energy, with each generator having a position and velocity, the time integration of \( E \) becomes a sum over all timesteps. Furthermore, using \( v^j = \frac{x^j+1-x^j}{h} \), the action can be expressed as:

\[
S = h \sum_j \sum_i^n \alpha \int_{B_i(\tilde{x})} \rho(x, t)(x - \tilde{x}_i^j)^2 dx + \beta \int_{B_i(\tilde{x})} \rho(x, t^j)(v - \frac{\tilde{x}_i^{j+1} - \tilde{x}_i^j}{h})^2 dx
\]

(6.15)
where \( \tau \) is the final timestep and \( j \) represents the \( j^{th} \) timestep. Using the Hamilton’s principle of least action, the minimum \( \tilde{x} \) and \( \tilde{v} \) can be determined analogously to the previous continuous derivation.

\[
(6.16) \quad \frac{d}{ds} h \sum_{j}^{\tau} E(\tilde{x}^{j+1}, \tilde{x}^{j}, t^{j}) = 0
\]

By the chain rule, this \( \frac{d}{ds} \) derivative on the action becomes a \( \frac{\partial}{\partial \tilde{x}^{j}} \) acting on the energy. Notice that only two terms from the \( \sum_{j}^{\tau} \) have a dependence on \( \tilde{x}^{j} \), therefore the terms from the other timesteps can be ignored.

\[
(6.17) \quad h \frac{\partial}{\partial \tilde{x}^{j}} (E(\tilde{x}^{j+1}, \tilde{x}^{j}, t^{j}) + E(\tilde{x}^{j}, \tilde{x}^{j-1}, t^{j-1}) = 0
\]

The previous equation is expanded using the lagrangian definition of \( E \):

\[
(6.18) \quad h \sum_{i}^{n} \frac{\partial}{\partial \tilde{x}^{j}} \left[ \alpha \int_{B_{i}(\tilde{x}^{j})} \rho(x,t^{j})(x - \tilde{x}^{j}_{i})^{2} dx + \beta \int_{B_{i}(\tilde{x}^{j})} \rho(x,t)(v - \frac{\tilde{x}^{j+1}_{i} - \tilde{x}^{j}_{i}}{h})^{2} dx \right]
\]

\[
+ \alpha \int_{B_{i}(\tilde{x}^{j-1})} \rho(x,t^{j-1})(x - \tilde{x}^{j-1}_{i})^{2} dx + \beta \int_{B_{i}(\tilde{x}^{j-1})} \rho(x,t)(v - \frac{\tilde{x}^{j}_{i} - \tilde{x}^{j-1}_{i}}{h})^{2} dx = 0
\]

Again there are two integrals here that have an area of integration that is dependant on the variable the derivative is taken with respect to. This means the Reynolds transport theorem is used to determine the derivative of the first two integrals in this expansion. Furthermore, the third integral here has no dependence on \( \tilde{x}^{j} \) and thus is zero. The fourth term in this expansion can be separated into two terms: \( \int_{\partial B_{i}(\tilde{x}^{j-1})} \rho(x,t^{j-1}) v dx \) and \( \int_{\partial B_{i}(\tilde{x}^{j-1})} \rho(x,t^{j-1})(x - \tilde{x}^{j-1}_{i}) dx \). From this point the \( \frac{\partial}{\partial \tilde{x}^{j}} \) can be calculated.

\[
(6.19) \quad h \sum_{i}^{n} \left[ \alpha \int_{B_{i}(\tilde{x}^{j})} \rho(x,t^{j})(2x - 2\tilde{x}^{j}) dx + \int_{\partial B_{i}} (...) \right] + \beta \int_{B_{i}(\tilde{x}^{j})} \rho(x,t^{j}) \frac{2}{h} (v - \frac{\tilde{x}^{j+1}_{i} - \tilde{x}^{j}_{i}}{h}) dx + \int_{\partial B_{i}} (...) \right] = 0
\]
To evaluate the boundary terms, the continuous argument is considered for the first \( \int (...) \) term to be 0 due to the cancellations on the boundaries. The second of the boundary \( \partial B_i \) terms is expanded to yield:

\[
\int_{\partial B_i(\bar{x}^j)} (\partial B_i(\bar{x}^j)) \cdot (v - \frac{\bar{x}^{j+1}_i - \bar{x}^j_i}{h})^2 \bigg|_{\partial B_i(\bar{x}^j)} = \frac{1}{2} \rho(B_i, t^j) (v(B_i + \frac{1}{2}) - \bar{x}^{j+1}_i - \bar{x}^j_i)^2
\]

(6.20)

\[
\frac{1}{2} \rho(B_i, t^j) (v(B_i + \frac{1}{2}) - \bar{x}^{j+1}_i - \bar{x}^j_i)^2 - \frac{1}{2} \rho(B_{i+\frac{1}{2}}, t^j) (v(B_{i+\frac{1}{2}}) - \bar{x}^{j+1}_{i+1} - \bar{x}^{j+1}_i)^2 + \frac{1}{2} \rho(B_{i-\frac{1}{2}}, t^j) (v(B_{i-\frac{1}{2}}) - \bar{x}^{j+1}_{i-1} - \bar{x}^{j+1}_i)^2
\]

Using to two above equations equal to 0, this can be solved for \( \bar{v}^{j+1} \) via a Newton’s Method.

### 6.3 Time Integrator Euler-Lagrange

Solving for \( \bar{v}^{j+1} \) to determine the time integrators requires the use of an iterative Newton’s method.

\[
F = h \sum_i^n \left[ \alpha \int_{B_i(\bar{x}^j)} \rho(x, t^j) (2v - 2\bar{v}^j) dx + \beta \int_{B_i(\bar{x}^j)} \rho(x, t^j) \frac{2}{h} (v - \bar{v}^{j+1}) dx \right.
\]

(6.21)

\[
\left. + \beta \int_{B_i(\bar{x}^{j-1})} \rho(x, t^{j-1}) \frac{2}{h} (v - \bar{v}^j) dx + \frac{1}{2} \rho(B_i, t^j) (v(B_i + \frac{1}{2}) - \bar{v}^{j+1})^2 - \frac{1}{2} \rho(B_{i+\frac{1}{2}}, t^j) (v(B_{i+\frac{1}{2}}) - \bar{v}^{j+1})^2 \right.
\]

\[
\left. + \frac{1}{2} \rho(B_{i-\frac{1}{2}}, t^j) (v(B_{i-\frac{1}{2}}) - \bar{v}^{j+1})^2 = 0 \right]
\]

At this point in the derivation the current generator position is known and determined by:

(6.22) \[ \bar{x}^{j+1} = \bar{x}^j + h\bar{v}^j \]

Therefore, the only unknown is \( \bar{v}^{j+1} \). A first order Taylor expansion is done of \( F \), under the assumption that \( \delta \bar{v}^{j+1} \) is small, so the velocity difference between timesteps cannot be too large.

(6.23) \[ F(\bar{v}^{j+1} + \delta \bar{v}^{j+1}, \bar{v}^j, \bar{x}^{j+1}, \bar{x}^j) \approx F(\bar{v}^{j+1}, \bar{v}^j, \bar{x}^{j+1}, \bar{x}^j) + DF(\bar{v}^{j+1}, \bar{v}^j, \bar{x}^{j+1}, \bar{x}^j) \delta \bar{v}^{j+1} \]
where $DF$ is the derivative matrix of $F$ and $\hat{v}$ is the guess for $\tilde{v}$ in the Newton’s method. This gives a $\delta \tilde{v}^{j+1}$ of the form:

$$\delta \tilde{v}^{j+1} = (DF)^{-1}(-F)$$

The derivative matrix, $DF$, is sparse. The only nonzero terms are of the form:

$$(6.24) \quad \delta \tilde{v}^{j+1}_j = DF^{-1}(-F)$$

The derivative matrix, $DF$, is sparse. The only nonzero terms are of the form:

$$\frac{\partial F}{\partial \tilde{v}^{j+1}_i} = h\beta \left[ \int_{B_i(\tilde{x}^j)} \rho(x,t^j)dx - \rho(B_{i+j^1/2},t^j)(v(B_{i+j^1/2}) - \tilde{v}^{j+1}_i) + \rho(B_{i-j^1/2},t^j)(v(B_{i-j^1/2}) - \tilde{v}^{j+1}_i) \right]$$

$$\frac{\partial F}{\partial \tilde{v}^{j+1}_{i-1}} = h\beta \left[ -\rho(B_{i-j^1/2},t^j)(v(B_{i-j^1/2}) - \tilde{v}^{j+1}_{i-1}) \right]$$

$$\frac{\partial F}{\partial \tilde{v}^{j+1}_{i+1}} = h\beta \left[ \rho(B_{i+j^1/2},t^j)(v(B_{i+j^1/2}) - \tilde{v}^{j+1}_{i+1}) \right]$$

The derivative matrix that is tridiagonal, with symmetric corner values, yielding 3 non-zero values per row. Using a Newton’s method the $\delta \tilde{v}^{j+1}$ can be determined, and thus the velocity time integrator is:

$$\tilde{v}^{j+1} = \tilde{v}^j + \delta \tilde{v}^{j+1}$$

giving a complete set of time integrators to enforce advecting CVT-like behavior in a Voronoi mesh.
These time integrators were implemented into the simulation. This results in the following CVT:

![Render of the Euler-Lagrange CVT on the sin(2πx) + 2 distribution.](image)

As the fluid flows at a constant velocity, the CVT cells advect, containing the same parts of the fluid density distribution. This is expected behavior, and exhibits Euler-Lagrange CVTs to be a good solution to cosmological fluid simulations.

In these early stages, strange behavior is still seen. This displayed some interesting behavior when some of the cells contain more mass than others:
Artifacts arising on the Euler-Lagrange CVT for cells enclosing different masses.

This results in an interesting bulge artifact. This shows that it is important to try use this technique for cells holding similar enclosed masses.

Nonetheless, this demonstrates this technique is possible, implementable, and viable. The current status of this method is exhibiting some unstable behavior for large timesteps and has divergent, unphysical behavior when initial conditions are not predicted exactly.

7 Conclusion

This paper explores use of these fluid simulation techniques to compressible, discontinuous fluids used in cosmological simulations. Centroidal Voronoi tessellations are used and proven viable for certain discretization schemes, while other discretization schemes are proven to be unsuitable. Adhoc time dependence methods are used with CVTs to analyze the limitations of this model. Error in the simulation is reduced by adding an advection term to CVT, called the Euler-Lagrange CVT. These are shown to be theoretically possible and are implemented to show viability. While these Euler-Lagrange CVTs are still in their beginning stages, the desired behavior of a CVT that advects with the fluid has shown great promise for use in future astrophysical simulation.
8 Future Work

The work done in this project is restrained to one dimension. Much of the analysis on the limitations of CVTs can readily be abstracted to higher dimensions, however, many new restrictions could arise when explored in 2D and 3D. The Euler-Lagrange CVTs have unstable behavior when simulated over more than a few timesteps. To make these truly applicable, much work is to be done narrowing down the conditions for convergence and divergence of this method. In addition, the Euler-Lagrange derivation was derived completely in one dimension. Since astrophysics does not exist in purely a 1D world, new forces and time integrators will need to be derived for higher dimensional simulations.

References


