**SPH Simulation of Astrophysical Gas Dynamics**

## Visual Goals

The visual goal of this project was to simulate the collapse of a gas cloud into a rotating disk-shaped protostar in grayscale using smoothed particle hydrodynamics. As the project progressed, however, the simulator could be adapted to simulate a wider variety of astrophysical phenomena by tuning the initial conditions and the relative strengths of gravity, pressure, and viscosity. These additional phenomena included galaxy formation and binary star formation from colliding gas clouds.

## Implementation Details

The details of the project implementation can be broadly divided into four parts: gravitational force calculation, hydrodynamical force calculation, time integration, and initialization of particle attributes. Each particle was rendered as solid sphere, and its color was scaled from red to yellow depending on the density. This was an ad-hoc way to enable visualization of denser regions, as it would have been harder to distinguish subtle structure formation if all particles had the same brightness.

At the beginning of each time step, all particles in the system were inserted into a quadtree such that each leaf contained no more than one particle. The total mass of child particles and the position of the center-of-mass was stored at each internal node. Once the quadtree was constructed, the Barnes-Hut algorithm was used to approximately calculate interparticle gravitational forces.[1] Given that most journal papers that use the Barnes-Hut algorithm to calculate gravitational forces deal with extremely large particle numbers (usually > $10^6$), I relied on another source[2] dealing with $O(10^3)$ particle numbers that approximated the error of Barnes-Hut for different values of $\theta$ (which determines the minimum distance between the center-of-mass of an internal node and particle *i* such that we can approximate the force on particle *i* due to all particles contained in the internal node by the force due to the center of mass "particle") and I concluded that $\theta$=1.8 was the optimal tradeoff between computational efficiency and force accuracy for particle systems of around 1,000-10,000 particles.

Next, the density at each particle was calculated using the cubic spline interpolation kernel introduced by Gingold & Monaghan in their seminal 1977 paper on smoothed particle hydrodynamics.[3] The smoothing length – the radius of the particle beyond which the smoothing kernel goes to 0 – was set such that on average 20 particles were located within the smoothing length of any given particle in the initial configuration. Although Gingold & Monaghan used variable smoothing lengths such that the number of particles within the smoothing length of any given particle was almost exactly constant, I did not have the time to implement this feature. The

---

[1] https://www.cs.princeton.edu/courses/archive/fall03/cs126/assignments/barnes-hut.html
[2] https://jheer.github.io/barnes-hut/
[3] Gingold, R.H; Monaghan, J.J. "Smoothed particle hydrodynamics: theory and application to non-spherical stars".

pressure at each particle was then calculated as a function of density using a polytropic equation of state. There are a variety of equations of state that have been used to approximate the pressure of a gas as a function of density, depending on relevant physical scenario (e.g., isothermal gas, convective star gas, etc.). The polytropic equation of state, $P = A\rho^{\gamma}$, was chosen due to its tunability; different choices of $\gamma$ can model a vast range of physical systems, from neutron stars to isothermal spheres of gas and more.

Having obtained the density and pressure per particle, the hydrodynamical forces were calculated. This was done for each particle by finding its neighboring particles (i.e., particles within the smoothing length) and calculating the pressure-induced force using the SPH formulation of the Lagrangian Navier-Stokes expression for force. An additional fictitious pressure called "artificial viscosity" was included, as is commonplace in astrophysical gas simulations since its introduction by Monaghan & Pongracic in 1985.[4] The exact expression proposed by Monaghan was used, albeit with the consolidation of physical constants to make parameter-tuning easier. This artificial viscosity partly removed some strange oscillations that occurred, presumably to the interplay of pressure and gravitational forces.

The nearest neighbor search required for efficient density and force calculation can be implemented in a variety of ways, but I wanted to find a way to reuse the Barnes-Hut tree. The algorithm I ultimately devised was as follows.

- Starting from the root of the tree, if the circle of radius *h* (corresponding to the smoothing length) about the particle of interest *p* intersects the circle that circumscribes the given node's quadrant, then recursively check all child nodes of that quadrant.
- Repeat this until you reach a leaf, at which point check if the particle contained in said leaf lies within the smoothing radius

While this algorithm is not as efficient as possible, it considerably simplified the overall project, as I did not need to implement the different data structures used in other neighbor search algorithms.

With all relevant forces calculated per particle, leapfrog integration was used to update particle positions and velocities. Leapfrog integration was chosen because it is symplectic (it conserves the underlying Hamiltonian) and straightforward to implement, as well as a common choice of SPH integration scheme.[5]

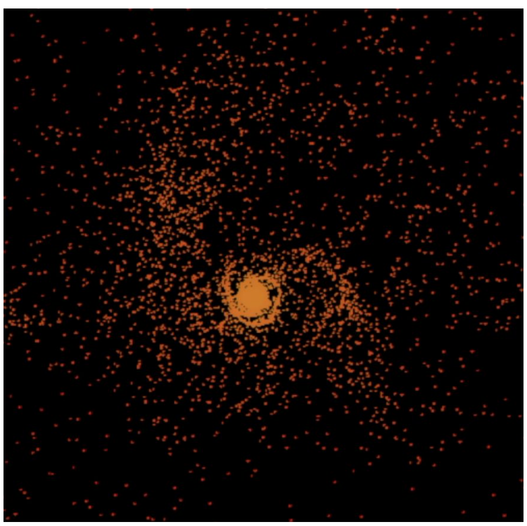
**Analysis of Visual Results**

My visual goals were roughly achieved, although not to the resolution that I had initially desired. Throughout the duration of the project, I discovered that there was an obstacle of scale, both in time and length. More precisely, starting with a spherical gas cloud (with or without rotation), a large amount of time must elapse before the system reaches the accretion disk quasi-equilibrium. Since each time step took as long as 15 seconds to render, particularly when

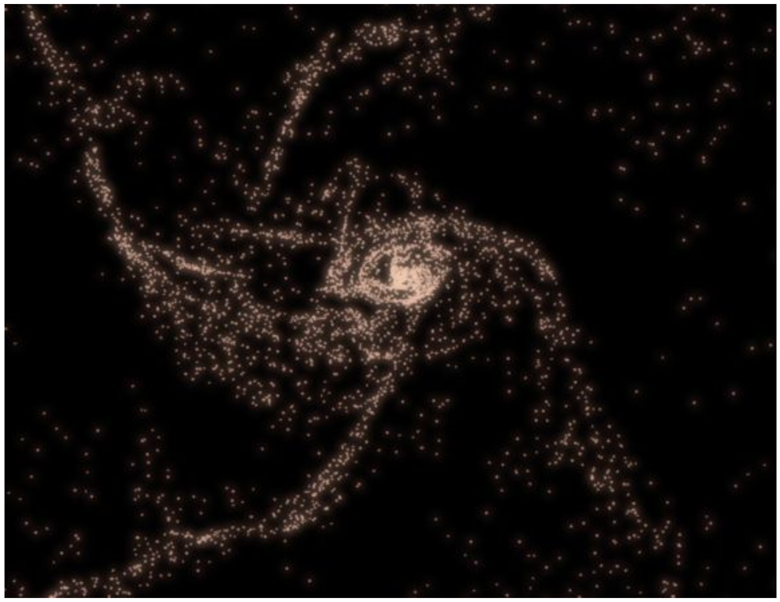
[4] Monaghan, J.J.; Pongracic, H. "Artificial Viscosity for Particle Methods".
[5] https://www.cs.cornell.edu/~bindel/class/cs5220-f11/code/sph.pdf

particles accumulated into regions of high density, it was simply infeasible to perform a simulation this long. Moreover, in reality the initial size of the gas cloud is orders of magnitude larger than the protostar systems that form within it during gravitational collapse, so the simulation must be able to simulate both large and small scales with relatively high resolution to achieve visual fidelity. Aside from the computational cost of doing so, the ability to magnify regions of interest would be necessary. This subtle issue was the most difficult roadblock to overcome.

Nonetheless, by fine-tuning the parameters governing the relative strengths of the gravitational and hydrodynamical forces, compelling animations can be produced. I went about tuning the parameters by first finding the extremum at which pressure overcame gravity and caused the cloud to expand like a hot gas, as well as the extremum at which gravity was so strong that pressure could not halt gravitational collapse to a near singularity. The critical density at which pressure balanced gravity could then be adjusted by tuning the relative force strengths within these bounds. Such force tuning was the most impactful (and time-consuming) component in achieving the final visual result. It was particularly important in circumventing problems stemming from high particle densities, as the pressure could be adjusted such that arbitrarily high particle densities were highly improbable Moreover, it enabled the simulation of different types of astrophysical gas clouds: for larger clouds, gravitational collapse must be the dominant effect, so its relative strength must be increased (assuming particle number is fixed).



*Above: Small accretion disk forming around protostar,*
*10k particles, medium pressure*



*Above: Gas spiraling into an early-stage galaxy,*
*15k particles, low pressure*

**Future Work**

The project could be improved significantly with the addition of parallelization and sink particles. Almost all stages in the simulation, with the exception of quad-tree construction, can be made parallel across all particles, which would lead to a massive decrease in computation time and enable longer simulations exploring the equilibrium behavior in greater depth. Furthermore, dynamic sink particle creation would significantly reduce the depth of the quad-tree by consolidating gravitationally-bound areas with large particle density into single particles representing the entire mass and radius of the region. In conjunction with a hybrid quad-tree/uniform-grid method (where the uniform grid would enable efficient force calculation and neighbor search in high density regions), problems due to high particle density would all but disappear.