Using FLASH to Build and Analyze a Protoplanetary Disk

Sean C. Lewis[1]

[1] *Drexel University Department of Physics*
Philadelphia, Pennsylvania

(Received May 29, 2019)

Submitted to Drexel University Department of Physics

ABSTRACT

Protoplanetary disks are the consequence of angular momentum conservation: remnants of collapsed interstellar nebulae around newborn stars. The disks are thought to be the precursors of planetary systems, including ours. The observation of short-lived radioisotopes embedded within objects in our Solar System suggests that at a close proximity supernova injected material into the gaseous environment that would eventually become our home. Furthermore, the observed presence of protoplanetary disks in and around star birthing regions (O'dell & Wen 1994) lends credence to the hypothesis that a massive star gone supernova may have injected enriched matter directly into our protoplanetary disk (Ouellette et al. 2007; Bastian et al. 2013b; Wijnen et al. 2017). An important limiting factor is the survivability of a protoplanetary disk after it is impacted by a supernova blast wave, and it is my goal to create a simulation framework on which to test this. The problem is inherently hydrodynamical. As such, I employ the Eulerian grid-based magnetohydrodynamics code FLASH (Fryxell et al. 2000) to construct a protoplanetary disk in two ways. First, I unsuccessfully attempt to initialize a disk in hydrostatic equilibrium. Density discontinuities at the surface of the disk require a smooth buffer region to be present, diminishing the ability for the user to define and maintain a predetermined density profile. In my second attempt, I repurpose a test problem to initialize a gravitationally unstable, rotating, spherical cloud. I find that, if enough time is allowed for the cloud to collapse and relax, it forms a disk structure with the radial density profile tending towards the distribution predicted by disk models (Lynden-Bell & Pringle 1974; Hayashi 1981; Desch 2007) at distances beyond 100AU from the disk's central star. Such regions of a protoplanetary disk are most likely to be disrupted by a supernova blast via momentum stripping or ram pressure stripping. My efforts have enabled users to initialize a gas cloud in a star-forming environment with control over the initial dimensions, density, pressure, and equation of state. The cloud will then collapse and relax into a disk in any chosen inclination relative to the yet-to-be simulated supernova blast.

1. INTRODUCTION

Protoplanetary disks can be characterized as the left-over gaseous material orbiting a newly formed star, sometimes stretching hundreds or thousands of Astronomical Units (AU) across. With recent observations of planet formation in protoplanetary disks, it is reasonable to suggest that protoplanetary disks evolve over time, progressively evaporate away due to the host star's radiation, become disrupted by external systems, or condense into planetary systems over the course of 10 Myr (Williams & Cieza 2011). The study of the formation and evolution of protoplanetary disks is the study of the history of our own solar system. The formation environment of protoplanetary disks has become an active and important area of astronomy especially in the last forty years. Even more recently, increasingly accurate and robust computer models has allowed computational astrophysicists to better model the complex hydrodynamical environments in and around protoplanetary disks. A recent area of investigation made possible by new computational capabilities is the source of short-lived radionuclei (SLRs) found in asteroids of our solar system.

It is well known that hydrogen and helium are naturally abundant in all star-forming regions. However, heavier elements must be formed via nucleosynthesis. Therefore, any presence of elements heavier than helium in a Solar System-like environment requires the injection of such elements from either the winds of the host star around which the system is bound, or from an external source. In our own Solar System, the presence of elements heavier than helium is abundantly clear, meaning the injection of processed elements into the Sun's formation environment certainly is not a novel idea. However, two elemental isotopes are uniquely suggestive of a large deposition of material specifically at or around the time of its formation. The SLRs Al-26 and Fe-60 have observed abundances in meteorites that places their creation and deposition to be at the time just before the Sun's formation (the collapse of the solar nebula) or while the protoplanetary disk is still in the process of condensing.

Bastian et al. (2013b) notes that supernovae may deposit their contents into the diffuse solar nebula as well as trigger a collapse into a protoplanetary disk and central star. Klein et al. (1994) determined that a supernova blast in the proximity of an interstellar cloud is able to completely destroy the cloud disallowing any collapse of the potentially enriched nebula into a protoplanetary system. Alternatively, Ouellette et al. (2007) and Wijnen et al. (2017) explore the possibility that a solar nebula collapse into a protoplanetary disk occurs prior to a supernova interaction, allowing the disk to act as a gaseous net of sorts, sweeping up material synthesized within the exploded star and find that a proximity supernova will not destroy the gaseous disk. The constraining factors of this interaction are two-fold: the supernova must be close enough to the target disk to deposit enough material based on ejection efficiency and geometric arguments Adams, F. C. (2010) while also remaining far enough away to survive ram pressure stripping and momentum stripping. Further exploration of these factors are described in § 4.2. Although the survivability of large, interstellar, gaseous structures around a supernova has been disputed (Mac Low et al. 1994) and so to the possible enrichment scenarios, the focus in this paper will follow the interaction of an enriched supernova blast with a protoplanetary disk.

Ultimately, the survivability of the protoplanetary disk is an important limiting factor for the supernova matter injection problem. As evident by the investigational approach of Ouellette et al. (2007), Wijnen et al. (2017), Bastian et al. (2013b), and others, hydrodynamical simulations are necessary to accurately describe and probe systems of this scale. In this paper, I employ the Eulerian Grid code, FLASH, to properly model the environment and formation of a protoplanetary disk to create a simple model to which more advanced physics can be included and tested. There are additional questions as to the likelihood of such an event occurring, Protoplanetary disks, with a lifetime on the order of Megayears, could be present in a cluster-like environment where a massive star detonates and so provide a reasonable scenario worth investigating. The specific timing and probability of these events is beyond the scope of this paper.

2. NUMERICAL METHODS: FLASH

2.1. *Overview*

FLASH (Fryxell et al. 2000) is a multi-physics Eulerian hydrodynamics code with a large collection of physics modules/units capable of tackling an enormous range of physical problems including astrophysical systems. FLASH is highly modular code suite allowing for the implementation of new physics beyond hydrodynamics and gravity including non-ideal equations of state, nuclear reaction chains, MHD, star formation and radiation feedback among others. That being said, FLASH manages to greatly reduce compile times by only compiling module codes that are necessary for an individual problem. FLASH also employs adaptive mesh refinement (AMR), which automatically increases the levels of refinement in user-defined areas of interest allowing the code to minimize the number of grid cells in a simulation while also maintaining the greatest possible accuracy. Figure 1 depicts AMR in action during one of my simulations. FLASH, even with its modular and well-organized code structure is extremely complex so this paper will overview one of the more rudimentary aspects of the code: how and where simulation data is stored and computed.

2.2. *The Grid*

The Eulerian grid-based structure of FLASH, with the ability to automatically refine on user-defined physical parameters, is known as PARAMESH (MacNeice et al. 2000) (Olson & MacNeice 2005) and is depicted in Figure 2 from Fryxell et al. (2000). PARAMESH is a FORTRAN 90 subroutine package meant to give users an easy avenue to parallelize an adaptive 'block-structured' mesh that, beyond initial parameters, requires little user adjustment during a simulation. The 'blocks' around which PARAMESH is designed are, by default, cubic sections of the simulation space containing 8x8x8 cells where where field values like gas density, pressure, velocity, and internal energy are defined at their centers. Such fields are allowed to change and travel through the faces of the cells/blocks in accordance to the hydrodynamic equation solver used. The fluxes of the model solutions through cell/block faces is necessarily conserved as seen in Figure 2.

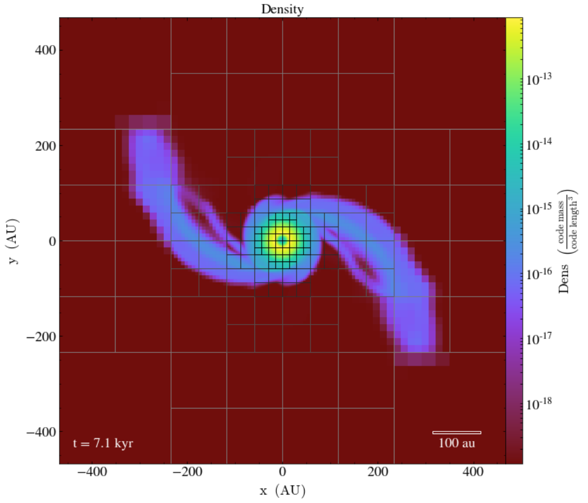

Figure 1. A sliceplot of one of my simulation runs, looking down the z-axis towards a face-on actively forming disk. The outlined squares are the boundaries of each block each containing 8x8 cells that are visible from this perspective. The highest level of refinement occurs around the sink particle, as per the sink particle creation criteria and is refinement level 7. The minimum refinement can be seen at the corners of the simulation space since there is little to know material in these regions and are at refinement level 3.

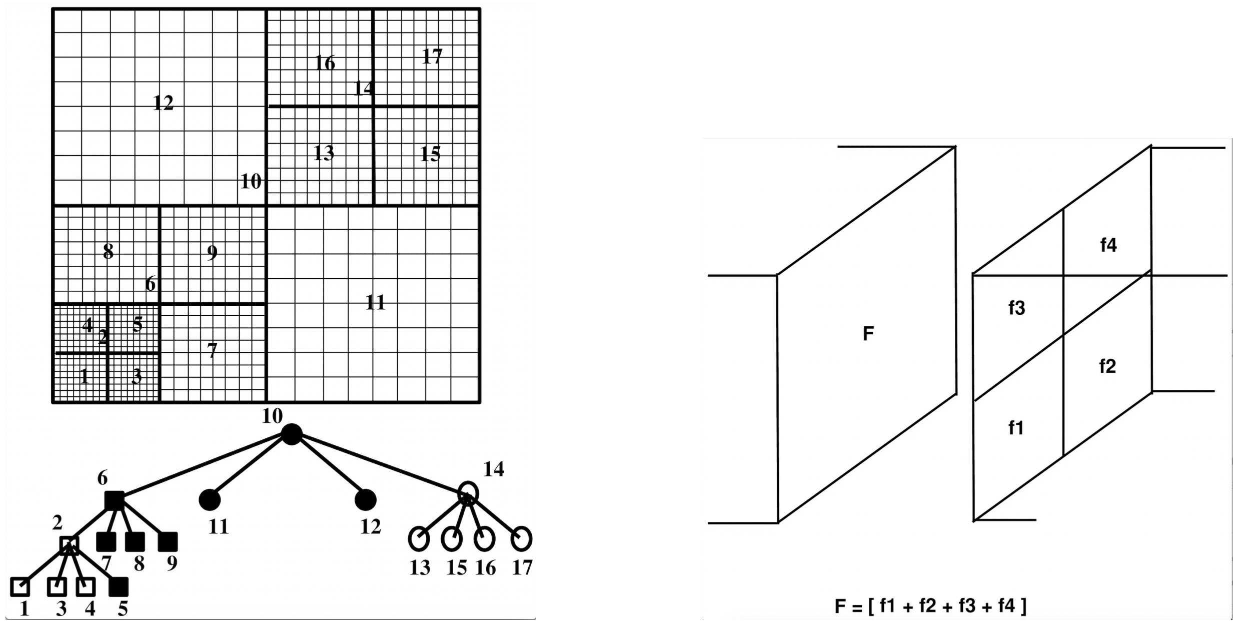

Figure 2. (Left) A simple representation of FLASH's adaptive mesh refinement and it's resulting nested block structure. Notice that each block, labeled by a number, has 8x8 total cells (given the 2D geometry) regardless of the block's size. Below the grid is the corresponding tree structure for the block arrangement. Each unique shape represents a block's location in 1 of 4 processors. (Right) A depiction of the field flux conservation requirements imposed by FLASH for each cell. The types of field flux include density, energy, radiation, magnetic fields, etc. As per the refinement criteria, only one level of refinement may separate two adjacent blocks.

In conservative form, the Euler equations for compressible gas dynamics which are solved in 3D for each cell face are:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{v} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} + \frac{\nabla p}{\rho} = \boldsymbol{g} \tag{2}$$

$$\frac{\partial e}{\partial t} + \boldsymbol{v} \cdot \nabla e + \frac{p}{\rho} \boldsymbol{\nabla v} = 0 \tag{3}$$

If the physical parameters within a block satisfy user-defined refinement criteria, the block will bisect itself along each Cartesian axis such that the resulting 8 blocks (assuming a 3D geometry) each have the same 8x8x8 cell structure and in total have the same volume as the parent block. In addition, in order to accurately conserve the physical quantities contained in a block upon refinement, PARAMESH extrapolates information contained within guard cells. If a block is already at least one level of refinement higher than any of its neighbors, further refinement is prevented from occurring.

To aid in accurately refining a block and determining field flows, FLASH employs guard cells: a buffer region 4 cells wide around each block. The guard cells are filled independently to the cells they surround, and when refinement is increased or decreased, the guard cells provide the necessary information to be interpolated and placed in the new block. Sometimes, errors occur during the guard cell filling process and can be addressed accordingly, see §2.5.

The block and the cells provide a method of identification and location information for the data stored inside them and their uniform shape and hierarchal structure allows for the usage of a data tree structure that identifies block's position and processor allocation, see Figure 2. Thus, with a message passing interface such as OpenMPI, huge amounts of simulation data can be processed on multiple cores across many nodes.

2.3. *Sink Particles*

FLASH supports the inclusion of active particles, also known as sink particles, a data structure containing field values such as position, velocity, momentum, etc. and physical attributes such as mass. Sink particles with user defined field and attribute values can be placed during the initialization of the simulation as I do in §3.2. Sink particles can also form if specific conditions of the gas are met during the simulation. The conditions are described in detail in Federrath et al. (2010), but the condition and properties in direct control for the user is the “sink density threshold” and the “sink accretion radius”. Once the gas density threshold is achieved by 8 cells sharing a corner (in 3D), the hydrodynamical solver is prevented access to the gas and all the field values contained within the data structure of the cells is then attributed to a sink particle such that mass, energy, and momentum is conserved during the conversion. Once a sink is formed, it is able to take on more gas as long as the gas meets the density threshold as well as the other requirements as mentioned before. In my simulations, sink particles act as the central star of the protoplanetary disk, providing the gravitational pressure keeping the system bound.

2.4. *Segmentation Fault Errors*

Occasionally, Segmentation Fault errors are detected by FLASH or OpenMPI and the simulation is halted. Such errors, also known as segfaults, have a tendency to occur within the first dozen time steps of a simulation, or after a many thousands of time steps, for differing reasons. A segfault occurring very early on is almost certainly due to an extreme discontinuity in a calculated variable such as density, pressure, internal energy. If the discontinuity is spread over a space less than a cell's width, the equation of state solver may record —inf internal energy, 0.0 pressure or another non-physical property. This type of segfault can be seen in action in Figure 4 and its fix is described in §3.2. The latter segfault type is much less predictable and difficult to combat without knowledge of the internal optimizations defined within FLASH's cell-filling routine. FLASH users other than myself recently determined that the error is most likely produced during the Morton filling method for the guardcells. Specifically, an optimization routine for the filling method defined by the variable `PM_OPTIMIZE_MORTONBND_FETCHLIST` produces disagreements between the sending and receiving end for guardcell data, resulting in the segfault. Leaving this variable undefined seems to fix the erroneous behavior in some cases. Though, often the user must force the simulation to dump checkpoint files right before the error to use as a restart file. A more permanent fix is being explored by the FLASH user community at the moment.

2.5. *Considerations for the use of FLASH*

Eulerian grid-based codes Smooth particle hydrodynamic (SPH) simulations are the other prominently used tool to study astrophysical systems. SPH codes are unaffected by the presence of near vacuums or field discontinuities unlike grid codes, and so are often superior for modeling systems such as galaxies or interstellar media. However, SPH codes have a near or complete inability to promote and resolve Kevin-Helmholtz and Rayleigh-Taylor instabilities as well as gas mixing events, while grid-based codes are much more capable as described in Agertz, O. et al. (2007). For this reason, FLASH is a logical choice for investigating a problem involving a supernova blast wave.

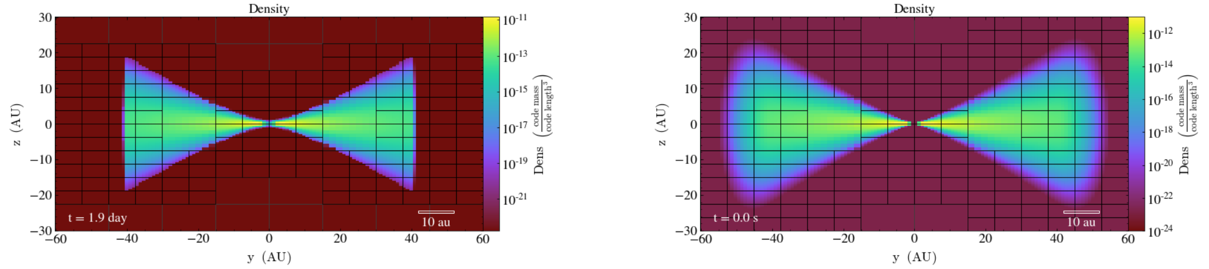

Figure 3. Together, the above images depict the profile perspective of two simulations and my problem solving process of including buffer regions around areas of density discontinuity. The density is measured in g cm^{-3}. The buffer region was calculated and placed by hand at the radial edge of the disk. In the image on the Right, I have also lowered the density floor for the image to show the already present vertical buffer region, best described as the "fuzziness" of the top and bottom of the disk. Both of these simulations are dysfunctional for similar errors but due to different regions producing those errors. On the left, the simulation aborts due to segfaults, memory errors, occurring in cells at the vertical outer edge of the disk. On the right, there are no longer segfaults at the outermost radial regions, but similar errors occur at the innermost surface of the disk. This is explored in later section and can be seen in detail in Figure 4.

An important consideration, however, is grid codes naturally bleed angular momentum due to the physical grid structure itself. Material and field flux is only allowed to pass through the faces of a cubic grid cell. This introduces numerical error in macro scale flows that is exasperated when the field's flow vectors are directed towards the edge or corner of a cell. Therefore, especially with rotating systems, angular momentum is expected to be inaccurately reported, with error compounding as the simulation time increases. For the purposes of the investigation of a protoplanetary disk, this is certainly a concern and will be explored further, see §4.3 for a working example of extreme angular momentum loss due to numerical effects.

3. MODELING METHODS OF THE DISK AND BLAST WAVE

3.1. *Overview*

As mentioned in the previous section, FLASH has been set up such that the density and pressure of each grid cell can be specified at the initialization of a simulation and by calling a user defined equation of state solver, the internal energy is determined at the start of each evolution time step after the cell information is determined. This allows for the user to approach the construction of a protoplanetary disk via two methods. First, disk in equilibrium. Second, collapsing cloud.

3.2. *A Disk in Hydrostatic Equilibrium*

Protoplanetary disks have been modeled extensively and their hydrostatic equilibrium state is well described (Lynden-Bell & Pringle 1974; Hayashi 1981; Desch 2007). The earliest models attempted to match the present day radial density distribution of planets and planetesimals with the initial conditions of the gaseous disk, resulting in a very centrally concentrated disk structure. As evidence for planetary migration and movement of the gas giants from the inner solar system outwards became more prevalent, models such as the Hayashi model Hayashi (1981) fell out of favor. In its wake, other, even more centrally concentrated disk models were proposed such as the Nice model (Desch 2007). The legitimacy and usefulness of any individual model is not explored in detail in this paper. I attempted to initialize a disk with the radial density profile matching that of the Hayashi model around an actively accreting sink particle (Federrath et al. 2010):

$$\rho(r, z) = \rho_0 r^{-2.75} exp\{-z^2/z_0^2(r)\} \tag{4}$$

With $\rho_0 = 1.4 \times 10^{-9}$ g cm^{-3} and $z_0 = 0.0472 r^{5/4}$. r, z, and z_0 are all given in AU. The density description is specifically for a cool gaseous disk around a 1 solar mass star.

Ideally, the radial density as well as other disk initial behavior knobs can be easily turned and tweaked based on the user's preference in the Simulation Block Initialization file. However, being able to place a rotating hydrodynamic disk such that it achieves hydrostatic equilibrium quickly enough to preserve user-defined behavior proves difficult. All disk models mentioned here as well as the equation explicitly provided describe the gaseous disk as having a gaussian vertical density profile, derived from the hydrostatic equilibrium state that the disk is assumed to be in. So,

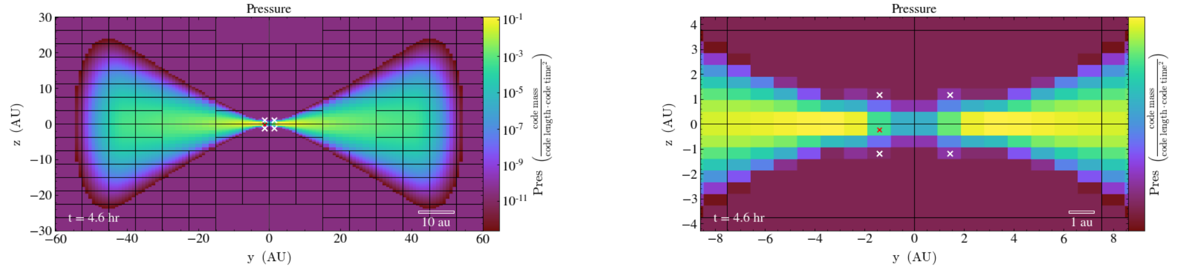

Figure 4. Pressure plots, measured in g cm^{-1} s^{-2}, of the buffered disk seen in Figure 3. Left: Slice through the center of the disk, viewed from the side. In the center of the image, 4 white “x’s” can be seen. Each x is placed at the point where 0.0 pressure is recorded, and corresponds to the cells causing the segmentation fault. Right: simply a zoomed in view of the region causing the error. The red “x” marks the highest pressure value recorded out of all of the cells. The segfault is occurring in a region where the highest pressure environment is separated from the much lower pressure of the halo gas by only 1 or 2 cells. This is a very large change in pressure over a short distance as seen by the hydrodynamical Reimann solver. The error can therefore be resolved if either the refinement is drastically increased or another buffer region is added to encase the innermost parts of the disk.

I specifically initialize the disk to have a vertical gaussian profile, and expect such a feature to remain throughout the simulation. The derivation for the vertical density profile is straightforward.

Even with the vertical disk structure defined properly and gas rotational velocity initialized to be the standard Keplerian velocity, additional fundamental complications are present in this method of disk modeling. As mentioned in the previous section, grid-based hydrodynamics codes such as FLASH have difficulty modeling vacuums and the flow and equation of state solvers do not play well with extreme discontinuities in pressure, density, or energy. Since the modeled disk is necessarily placed in a diffuse halo gas to mimic the low density surroundings solar nebulae find themselves, drastic discontinuities in density are present at the disk-halo boundary. Such an issue is inherently never present in a smooth-particle hydrodynamics (SPH) code. The discontinuity issue is further exasperated when the fluids at the boundary have differing flow velocities. The result is cells in and around the discontinuity will be subject to equation of state solver errors, causing individual cells or guard cells to be filled with huge amounts of impossibly negative energy or zero pressure. Therefore, I added a simple exponential decay in the radial density profile of the disk to buffer the sharp outer limit of the disk, see Figure 3. The presence of a smoothed outer edge of the disk not only creates a necessary buffer for the appropriate grid solvers, but it also mirrors the exponential decay in radial density as predicted by Lynden-Bell & Pringle (1974) and as inferred though observations of Orion proplyd disks (McCaughrean & O‘dell 1996). However, even with the added radial buffer, initializing a stable disk and allowing it to evolve for more than a few time steps proves difficult.

Figure 4, shows the simulation time step in which FLASH is no longer able to progress due to multiple cells existing on or around a significant discontinuity resulting in 0 pressure and/or infinite internal energy. The problem cells, marked with a white ‘x’, are present at the innermost region of the disk which understandably corresponds to the densest and highest pressure region of the disk. In order to eliminate the discontinuities at the disk’s inner regions, additional spatially decaying buffers would need to be added, at which point, the disk structure has necessarily diverged from any proposed hydrostatic equilibrium case. The divergence was such that the user essentially lost all control over defining a preferred disk structure, since any density profile had to be buffered in a such a strong-handed way. Therefore, I decided to abandon this approach to creating a disk, and instead augmented a working test problem predefined in FLASH.

3.3. *A Disk from a Collapsing Cloud*

Rather than fight with the difficult-to-quantify requirements for property discontinuities, I am able to use a test problem within FLASH to initialize a diffuse spherical cloud that, after collapsing and relaxing, forms a disk of gas orbiting a sink particle. The Sink Rotating Cloud Core (SRCC) test problem is originally meant to test the formation of a sink particle from a collapsing cloud, mimicking the formation of a star and protoplanetary disk from an unstable solar nebula (Federrath et al. 2010), (Boss & Bodenheimer 1979). Out of the box, the SRCC simulation initializes 1 solar mass sphere of gas with radius of 5.0e16 cm, about 3342 AU. Each cell of gas is given a velocity vector that is tangential to the axis of rotation, z-axis, and a magnitude of $r\Omega$ with $\Omega = 7.2 \times 10^{-13}$ rad s^{-1}. The value of Ω is chosen such that the ratio of the gas rotational energy to gravitational energy is $\beta = 0.16$. The gas density is described as $\rho = \rho_0[1.0 + 0.1cos(2\phi)]$ with $\rho_0 = 3.82 \times 10^{-18}$ g cm^{-3} and ϕ is the azimuthal angle. The cloud is initialized to be Jeans unstable (Jeans, J. H. 1902), see equation (5). The gas surrounding the cloud (which I will refer to as the halo gas), is defined to be at rest, in pressure equilibrium with the nebula gas, and 100 times less dense. The halo gas is a necessary component of the simulation as it acts as a stand in for the relative vacuum around the dense collapsing nebula and provides a buffer over which the differential equation solvers can step through without encountering too high of a discontinuity. After about 150kyr of simulation time, 5 times the free fall time of 34 kyr, the gas relaxes into a disk 2000 AU across.

In order to analyze the innermost regions of the protoplanetary structure, I decided to scale the SRCC test problem down to a spatial scale 10 times smaller. The most important factor to consider is I must maintain the appropriate scaling of the Jean's Length (Jeans, J. H. 1902). The original SRCC problem is Jeans unstable, as evident by its immediate collapse, thereby must be satisfying the gravitational collapse condition that the free-fall time is less than the compression travel time. Where $t_{sound} = \frac{R}{c_s}$ and $t_{ff} = \frac{1}{\sqrt{G\rho}}$ such that:

$$\lambda_J = \frac{c_s}{\sqrt{G\rho}} \tag{5}$$

Therefore, if the spatial regime of the problem is reduced by a factor of 10, the gas density must increase by a factor of 100. By maintaining the Jeans condition, the behavior of the cloud's collapse should appear visually to be identical, though will initialize as a 0.1 M$_\odot$ cloud and will result in a disk 200 AU across with a free fall time of 3.4 kyr: $\tau_{ff} \propto (\rho)^{-1/2}$. However, even while scaling all physical parameters properly, the now smaller spherical nebula will not fully collapse though a sink particle does form.

The equation of state for the system is described with a polytropic exponent. A polytropic equation of state is necessary only if a robust heating and cooling module is not being used. Although I did attempt to design and implement my own custom heating and cooling module, I was never successful and ultimately accepted that the discrete continuous polytrope captures the behavior of the gas well enough and its relative simplicity and user control are undeniable advantages. The equation of state is:

$$P = c_s^2 \rho^\Gamma \tag{6}$$

With gamma defined such that discrete gas density regimes are prescribed a value to best match their dynamical behavior during the collapse of the gas cloud, and throughout the simulation:

$$\Gamma = \begin{cases} 1 & \text{for} \rho/(10^{-13}\text{g cm}^{-3}) \leq 0.25 \\ 1.1 & \text{for} \rho/(10^{-13}\text{g cm}^{-3}) \leq 5.0 \\ 4/3 & \text{for} \rho/(10^{-13}\text{g cm}^{-3}) > 5.0 \end{cases} \tag{7}$$

Ultimately, the polytrope acts as a simplistic heating and cooling module. A gamma value of 1.0 describes an isothermal gas, one that can be compressed and perfectly radiate away the increase in energy. $\Gamma = 4/3$ corresponds to an adiabatic gas, one that, if compressed will increase in temperature and resist further collapse. $\Gamma = 1.1$ is essentially an intermediate regime that allows the gas to contract while providing some outward resistive thermal pressure. After scaling the density of the SRCC test problem, all of the cloud gas is then initialized to be in the adiabatic regime, essentially disallowing any significant collapse of the cloud. So, the density regimes, defined in the simulation parameter file, must also be scaled, mirroring the new density of the cloud and halo gas. The post-scaled Γ is defined above. The lowest value of Γ describes only the halo gas upon initialization. The entirety of the cloud gas lies in the $\Gamma = 1.1$ regime, promoting the collapse of the cloud with enough outward thermal pressure to prevent all of the gas from falling unimpeded to the cloud's center of mass at (0, 0, 0)AU. The adiabatic regime is not achieved until after the central, most dense, cells of gas convert to a sink particle, onto and around which further material collapses. It is only within the innermost 30AU that the adiabatic behavior, and resulting increase in temperature, is seen. The definitions of the three polytropic regimes, with further consideration, can be made more precise though I will not explore the possibilities of a more robust equation of state, I mention the implications in §4.5.

At this point, the now scaled down SRCC problem behaves nearly identically to the original test problem but on a timescale 10 times shorter. A new problem emerges, however, when considering the now much denser halo gas, which has been increased in density from the order of 10^{-20} g cm^{-3} to 10^{-18} g cm^{-3}. A typical H[II] region is now 1000 times less dense than the halo gas, and so the gas must be initialized as such. A too dense halo will artificially impede material injected into the simulation space and potentially prevent a shock wave from traveling from the simulation boundary to the disk. As a further complication, reducing the halo gas density 1000-fold introduces an extreme density discontinuity at the outer edge of the gas sphere. The introduction of an steep, exponentially decaying buffer zone at the outer edge of the initialized cloud, a similar combative measure as in §3.2, allows the simulation to proceed without encountering segmentation fault errors within the first few time steps. In Figure 5, the in-progress result of this method can be seen. Firstly, note the low resolution of the run. A lack of resolution will allow for the system to collapse, maintain enough of its angular momentum to form a disk, but will refrain from ramping computation time to an unreasonable level.

3.4. *A Simplified Supernova Blast Wave*

A supernova blast wave can be introduced to the simulation space containing the now relaxed protoplanetary disk, propagating downwards from the upper z-boundary. The blast wave material can have its density and pressure parameters adjusted by the user given the chosen proximity of the supernova detonation. A physically accurate supernova blast wave may require the implementation of time-dependent inhomogeneities in the wave structure which is beyond the scope of this paper. I take the ideal, spherical, smooth, and scalable ejection as described by Matzner & McKee (1999).
The initial velocity of the ejected material:

$$v_* = \sqrt{\frac{E_{in}}{M_{ej}}} = 2240\Big(\frac{E_{in}}{10^{51} erg \cdot s}\Big)^{0.5} \times \Big(\frac{M_{ej}}{10 M_\odot}\Big)^{-0.5} km \cdot s^{-1} \tag{8}$$

Where E_{in} is the initial energy of the supernova and M_{ej} is the total mass of the ejected material. The resulting velocity of the material is taken to be the radial velocity of the outwardly expanding sphere and is constant throughout the journey of the material from its star to the target. The elapsed time for material to escape the star:

$$t_* = R_* \sqrt{\frac{M_{ej}}{E_{in}}} = 1.552 \times 10^4 \Big(\frac{R_*}{50 R_\odot}\Big) \times \Big(\frac{E_{in}}{10^{51} erg \cdot s}\Big)^{-0.5} \times \Big(\frac{M_{ej}}{10 M_\odot}\Big)^{0.5} s \tag{9}$$

The initial density and pressure of of the ejecta:

$$\rho_* = \frac{M_{ej}}{R_*^3} = 4.72 \times 10^{-4} \Big(\frac{M_{ej}}{10 M_\odot}\Big) \times \Big(\frac{R_*}{50 R_\odot}\Big)^{-3} g \cdot cm^{-3} \tag{10}$$

$$P_* = \frac{E_{ej}}{R_*^3} = 2.278 \times 10^{13} \Big(\frac{E_{in}}{10^{51} erg \cdot s}\Big)^{0.75} \times \Big(\frac{R_*}{50 R_\odot}\Big)^{0.75} \times \Big(\frac{M_{ej}}{10 M_\odot}\Big)^{-1} ergs \cdot cm^3 \tag{11}$$

Where R_* is the radius of the exploding star.
The ratio of the escape time t_* to the travel time between the explosion and target, $t_{trav} = d/v_*$, scales the blast wave's density and pressure. In addition, as the blast wave passes over the disk at t = 0, the density, pressure, and wave velocity decrease as time elapsed increases:

$$\rho_{ej}(t) = \rho_* \Big(\frac{t_*}{t_{trav}}\Big)^3 \Big(\frac{t_{trav}}{t + t_{trav}}\Big)^3 \tag{12}$$

$$P_{ej} = P_* \Big(\frac{t_*}{t_{trav}}\Big)^4 \Big(\frac{t_{trav}}{t + t_{trav}}\Big)^4 \tag{13}$$

$$v(t) = v_* \Big(\frac{t_{trav}}{t + t_{trav}}\Big) \tag{14}$$

As of yet, the supernova blast has not been simulated here. By defining the +z boundary as an inflow boundary, I will be able to control the velocity, pressure, and density behavior of gas as it begins to move into the simulation-space.

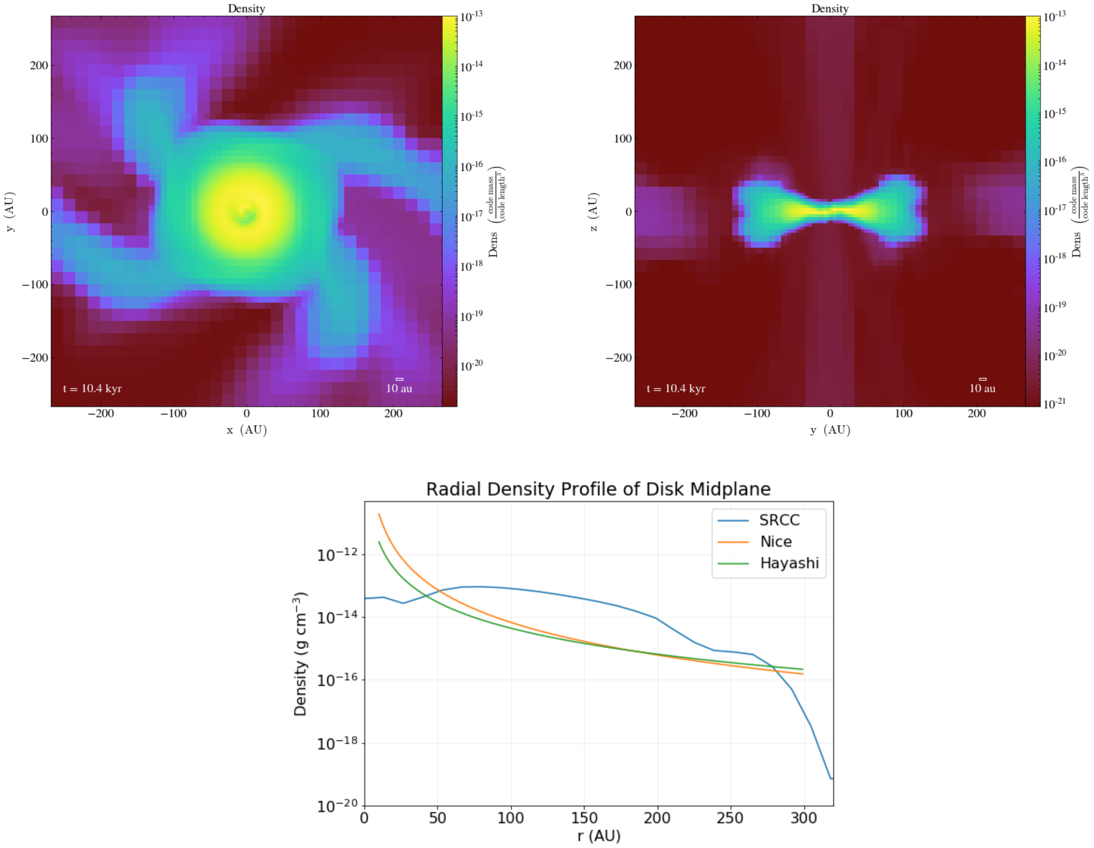

Figure 5. The most accurate simulation I have created (in terms of initialization parameters) with density slice plots allowing for face-on (Top Left) and edge-on (Top Right) examination of the disk with density measured in g cm^{-3}. The images are taken at t = $3\tau_{ff}$, hence the disk is still in the process of rebounding and re-contracting as it oscillates around equilibrium. The line plot is produced by recording the density values present along a ray data object drawn from the simulation center (0, 0, 0)AU to the edge of the simulation space (300, 0, 0)AU these density value are shown with the blue line. In addition, the orange and green lines show the theoretical radial density predicted by the Hayashi and Nice protoplanetary disk models.

4. RESULTS AND ANALYSIS

4.1. *Comparing Density Profiles, Angular Momentum Behavior*

Higher resolution runs are currently underway on the Draco computer cluster. However, even with low resolution or incomplete runs, the behavior of the gaseous disk's radial density profile and angular momentum. The simulation shown in Figure 5 is the result of the most accurate cloud collapse simulation I have run as of now in terms of initial conditions. At the time the images were pulled, the total time elapsed is about $3\tau_{ff}$. The cloud is obviously still in the process of relaxing, and the still extended spiraled arms are indicative of this. Upon first glance, the disk's density profile seems to be in the same ballpark as the minimum mass solar nebula as described by Hayashi (1981) and Lynden-Bell & Pringle (1974), shown in §3.2. The disk is obviously flared and the density at the innermost 10s of AU reach 10^{-}13 g cm^{-}3. Obviously, the innermost 10 AU are not resolved very well, and so the density recorded there seem to diverge greatly from the minimum mass solar nebula description.

To analyze the radial density profile in greater detail the python based analysis package **yt** (Turk et al. 2011) can be used. A ray data object can be drawn through a FLASH plotfile output that acquires the field values of each cell it passes through. As seen in 5, the density profile of the simulation cloud is indeed much less dense than both the Hayashi and Nice models, though this is expected. In addition, the simulation cloud is as much as an order of magnitude more dense than either model for the majority of the disk structure. This is also expected, since the disk used is a few free-fall times away from being nearly relaxed, so matter is still sloshing around the equilibrium point.

Figure 6 depicts the ending density profile of a less accurately initialized simulation run, but a run that was allowed to progress much further in time than all others that I have available. As can be seen, the simulation still fails to

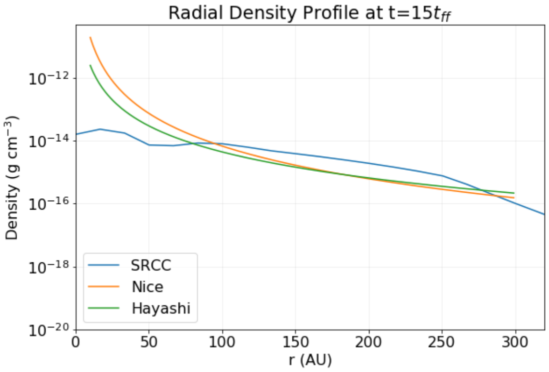

Figure 6. Repeated analysis of the radial density profile after the disk has been allowed to evolve and relax for a total of $15\tau_{ff}$. The resulting density distribution has an expected flattened characteristic, indicative of relaxation of the gas, but the innermost regions of the disk diverge from the power law distribution predicted by the theoretical models.

capture the power-law density increases at small radial distances as predicted by Hayashi and Nice. However, the discrepancy at radii in the 100s of AU more closely match the models, implying that the disk is tending towards the hydrostatic equilibrium state predicted for this region.

4.2. *Disk Stripping: Momentum and Pressure*

Once the protoplanetary disk is being impacted by the supernova blast wave, it is expected that some level of destruction of the disk will occur. Two basic mechanisms will contribute to the stripping away of the outer parts of the disk. Firstly, ram pressure stripping: where the pressure imparted onto the disk gas from the blast wave exceeds gravitational pressure due to the disk's central star (Adams, F. C. 2010).

$$\rho_{ej} v_*^2 \approx \frac{GM_\odot}{r_d^2} \Sigma(r_d) \tag{15}$$

Secondly, momentum stripping: where the momentum transferred to the disk gas exceeds the gas momentum as any particular radius (Adams, F. C. 2010).

$$\frac{M_{ej} v_*}{4\pi r^2} = \left(\frac{2GM_\odot}{r_d}\right)^{1/2} \Sigma(r_d) \tag{16}$$

Where $\Sigma = 1700(r/1AU)^{-1.5}$ (Hayashi 1981) and $\Sigma = 343(r/10AU)^{-2.168}$ (Desch 2007). A back of the envelope approximation can be made as to the radius of the disk at which both mechanisms of stripping will become significant. Given the surface density of the disk is inversely proportional to the radial distance away from the disk's central star, outermost regions of the disk are expected to experience the highest levels of destruction.

4.3. *Inclining the Disk*

It is likely that a face-on impact scenario will result in the largest amount of disk destruction and/or deposition of supernova material into the disk since it is the scenario with the largest disk surface area able to be directly impacted by the blast wave. However, being able to analyze the interaction between the blast wave and an inclined protoplanetary disk would certainly be interesting, and if anything can test the notion that the face-on case is optimum for destruction. Such an interaction is a complex problem, and at high resolution scales, unanticipated results may arise from a non ideal interaction. To incline the protoplanetary disk, the most straightforward is to simply rotate all position and velocity vectors within the simulation initialization file. I chose to rotate the initial gas sphere about the x-axis by applying the appropriate Euler rotation matrix. The rotation transformation of the velocity vectors can be seen here:

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) \\ 0 & -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{17}$$

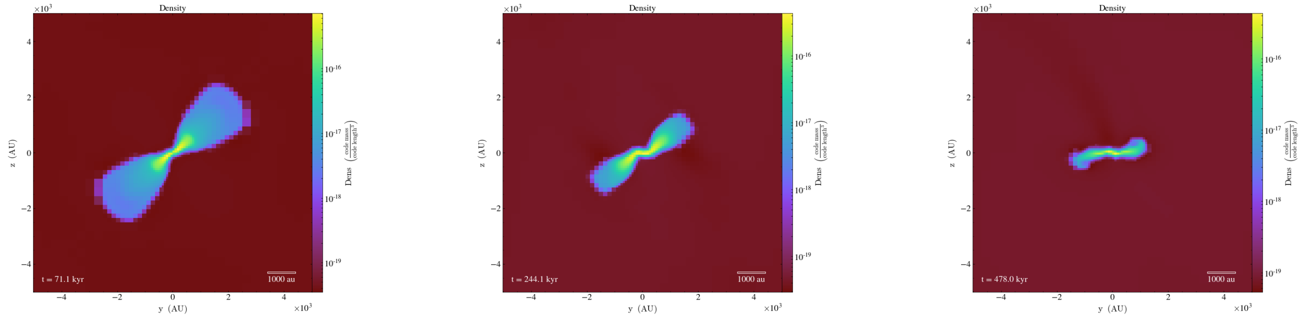

Figure 7. Density slice plots with the positive x-axis pointing towards the viewer, allowing for an edge-on view of the disk. Depicted is the out-of-the-box Sink Rotating Cloud Core test problem with the coordinate axes rotated 45 degrees counterclockwise from our perspective. From Left to Right, the images are taken at t = $2\tau_{ff}$, $7\tau_{ff}$, and $14\tau_{ff}$. This orientation of the disk is expected to produce the highest rate of numerical error due to angular momentum loss from field fluxes being unable to pass through the corners of cells. The error should scale with the number of dynamical times that have passed for any region of the disk, so numerical anomalies in the disk structure should be apparent first at the innermost regions, eventually effecting the entire disk. Indeed, at $7\tau_{ff}$ the disk is beginning to kink and degrade, eventually completely loosing its initialized inclination by $14\tau_{ff}$.

Where v_x, v_y, and v_z are defined in the simulation initialization file. Ultimately, any angle can be chosen and defined in the simulation parameter file, and the initial rotating gas sphere will be rotated to a new set of axes with its initial rotation behavior defined identically as the standard case but now for the new set of axes. Since the gas cloud is simply initialized as inclined, the user must wait an appropriate amount of time for the system to collapse and relax. This is the main trade-off for this method of inclination. But, given that the collapsing phase of the simulation can be run at a lower resolution (which is then ramped up later) the computation time is generally minimal: on the order of an hour wall-clock-time.

A second method for disk rotation is possible, though untried at this point. Checkpoint cell data from a simulation containing a non-inclined, relaxed disk can be deconstructed and stored in an array using **yt** and then each cell can be rotated to a new location with its corresponding velocity vectors accompanying the same rotation. The data can then be reconstructed into the checkpoint file data format, read into FLASH, and restarted from said augmented checkpoint file. Once this task is completed for the first time, the resulting python script to perform the data extraction, rotation, and reconstitution can be used for all other rotations for the particular problem.

The most prominent draw back to analyzing an inclined rotating disk is that angular momentum is lost extremely quickly due to computational error. In FLASH, material can only flow through the faces of each cell, and inclining the disk causes a larger component of flow to be directed towards the cell corners. Therefore, the central regions of the disk, where the dynamical time is small, will experience a large cumulative angular momentum computation error before the outer regions. As a result, the disk twists itself back to a horizontal orientation within a few hundred dynamical times.

This is a catastrophic result for any analysis that is needed to be done on an inclined disk system. The deviation in flow can be countered by increasing refinement, but ultimately a significant amount of planning on the user's part is necessary to chose a time to inject the supernova blast wave into the simulation: late enough so the disk has relaxed, and early enough to avoid the ever-increasing level of computational error.

5. CONCLUSION AND DISCUSSION

My goal was to create a framework to provide for the production and analysis of a protoplanetary disk in a hydrodynamics code. FLASH was chosen on the basis that an AMR grid-based code would be well suited to resolve the shock front at the interface between a supernova blast wave and the likely instabilities resulting thereafter. On the framework I created, users are now able to scale the size of the rotating cloud collapse problem, control the initial density and pressure distribution of the cloud and surrounding halo gas, as well as the initial orientation of the gas cloud's velocity vectors, allowing for simulating an inclined disk. The disk's radial density distribution seems to tend towards predicted hydrostatic equilibrium disk models at 100s of AU from the central star. Although the inner most regions of the disk may not be accurately represented, most likely due to low resolution, the outer regions are more likely to experience destructive forces due to ram pressure and momentum stripping. Beyond using the model I created to test the survivability of a protoplanetary disk, the user may also control the polytropic equation of state governing the compressibility of the gas: both the adiabatic constant values and the density threshold at which those value are applied are knobs for the user to turn at will. Ultimately, the user can now produce a rotating, gaseous, relaxed disk that sits in an environment similar to that found in star-forming regions and can be initialized to be inclined relative to the x-y plane at will.

At low resolutions, any of these simulations mentioned can be run on a robust laptop such as a MacBook Pro quad-core: 2.8GHz Intel Core i7, and can be completed in a few hours. Higher resolution runs can of course be completed on a computer cluster. In both cases, users will most likely have to build the message passing interface OpenMPI by hand, as I did, and point FLASH to it and to the appropriate compiler by customizing their own FLASH Makefile and .bash_profile.

5.1. *Future Work*

In its current state, my code structure is able to set up the scenario described in Ouellette et al. (2007): a gaseous disk in hydrostatic equilibrium. Next, a steady plane wave can be introduced to the simulation space by defining in inflow boundary condition at the +z axis. The total mass of the disk can be tracked through **yt** before and after the introduction of the blast wave in addition to the amount of material accreted onto the sink particle. In total, the total remaining mass in the simulation space after the blast has passed through will reveal if the disk has had matter stripped away or added as will visual inspection. A more complicated analysis is possible, by adding in modules available through FLASH the user can place tracer particles into the blast wave material. The location of the tracer particles can be recorded, and allow for the confirmation of mass injection into the disk. Other complex physics modules can be added. Some include: a custom heating and cooling module in place of the polytropic equation of state or a radiation subroutine to accurately describe the center star's influence on the disk. As an extremely ambitious goal, `Torch`, the result of coupling FLASH and the n-body dynamics code AMUSE (Zwart & McMillan 2018) would allow users to investigate protoplanetary disk behavior in an n-body situation such as a star cluster or multiple close proximity stellar systems with numerous planetesimals embedded within the disk gas.

REFERENCES

Adams FC. 2010. Annu. Rev. Astron. Astrophys. 48:47

Agertz O., et al., 2007, Mon. Not. R. Astron. Soc, 380, 963

Andrews S. M. et al. (2013) Astrophys. J., 771, 129.

Bastian N., Lamers H. J. G. L. M., de Mink S. E., Longmore S. N., Goodwin S. P., Gieles M., 2013, MNRAS, 436, 2398

Boss, A. P., & Bodenheimer, P. 1979, ApJ, 234, 289

Choudhuri, A. R. (1998). *The Physics of Fluids and Plasmas: An Introduction for Astrophysicists.* Cambridge University Publishing

Desch, S. J. Astrophys. J. 671,878—893 (2007).

Federrath C., Banerjee R., Clark P. C., Klessen R. S., 2010, ApJ, 713, 269

Fryxell B., et al., 2000, ApJS, 131, 273

Hartmann, L. (1998). *Accretion Processes in Star Formation.* Cambridge Astrophysics Series: 32

Hayashi, C. 1981, Prog. Theor. Phys. Suppl., 70, 35

Jeans, J. H. (1902). "The Stability of a Spherical Nebula". Philosophical Transactions of the Royal Society A. 199: 1—53.

Klein R. I., McKee C. F., Colella P., 1994, ApJ, 420, 213

Lynden-Bell D, Pringle J. 1974. MNRAS 168:603—37

Mac Low M.-M., McKee C. F., Klein R. I., Stone J. M., Norman M. L., 1994, ApJ, 433, 757

MacNeice, P., Olson, K. M., Mobarry, C., de Fainchtein, R., & Packer, C. 2000, CoPhC, 126, 330

Matzner, C. D., & McKee, C. F. 1999, ApJ, 510, 379

McCaughrean, M. J., & O'dell, C. R. 1996, AJ, 111, 1977

O'dell C. R., Wen Z., 1994, ApJ, 436, 194

Olson, K. M., & MacNeice, P. 2005, in Adaptive Mesh Refinement—Theory and Applications, ed. T. Plewa, T. Linde, & V. G. Weirs (New York: Springer), 315

Ouellette, N., Desch, S. J., & Hester, J. J. 2007, ApJ, 662, 1268

Turk M. J., Smith B. D., Oishi J. S., Skory S., Skillman S. W., Abel T., Norman M. L., 2011, ApJS, 192, 9

Wijnen, T. P. G., Pelupessy, F. I., Pols, O. R., & Portegies Zwart, S. 2017, A&A, 604, A88

Williams, J. P., & Cieza, L. A. 2011, ARA&A, 49, 67

Zwart, S. P., McMillan, S. (2018). *Astrophysical Recipes The Art of AMUSE.* Institute of Physics Publishing