Cassiopeia A:

Seeing a Supernova Remnant in a New Light



Travis Court

Department of Physics

Allegheny College

Cassiopeia A:
Seeing a Supernova Remnant in a New Light

Travis Court
Department of Physics
Allegheny College

April 20, 2018

Submitted in fulfillment of the senior thesis requirements of the Department of Physics at Allegheny College and approved by the senior thesis committee.

Dr. James C. Lombardi Jr.	Date	Dr. David Statman	Date

Pledge:

Travis A. Court

Acknowledgements

Dr. Lombardi - Thank you for always believing in me and pushing me to do my absolute best. You have led me to fully realize that I want to be an astronomer.

Dr. Statman - Thank you for all of your valuable questions and comments, and for grilling me on the fundamental *physics*, not astronomy, underlying my project.

Dr. Petasis - You were the first professor I had at Allegheny. You taught me the true value of a liberal arts education and re-ignited my love of books.

To all of the professors - Your different teaching styles and undying support are what brought me here and why I love it so much. I am proud to have had all of you.

Liam & **Sunshine** - Cheers to my first friends in the department and some of my best friends from college! You helped me open up, relax, and bring the ruckus.

Francesca - Thank you for all of the stupid and inapprorpiate jokes, the support, and for reminding me that learning material is more important than grades at the end of the day.

Matt - Cheers! From the awkward Eli interactions, to the "there's no way that is right" case study in Physics 110, to the late nights in Carr we have been through it all

Anna - You'll always be a part of the physics fam even if you aren't a physics major. Thank you for your infinite support, hugs, and cat pictures.

Trevor, Valerie & **Sam** - An uncountable number of my favorite college stories come from hanging out with you guys. Let's play some darts and roll around on the floor!

Other seniors, Sean, Carlos, Kylee, Celena, Brandon - We did it!!

Mom, Dad, Josh - I couldn't have gotten this far without you; I would not be the person I am without you all as role models. You are nothing but encouraging of my ideas and my career. I love yinz more than I can write on paper. Thank you.

Abstract

Cassiopeia A (Cas A) is a supernova remnant in the Milky Way galaxy. The supernova itself has been identified as a Type IIb supernova based on the spectra from the remnant. The spatial asymmetry of the spectra is the inspiration for multiple research projects and models. We model Cas A pre-supernova as a binary merger product of a 16 $M_{\odot}$ red giant star. For the secondary star, we vary its mass, radius, and separation from the primary. We investigate the role of these variables on the radial velocity and distribution of ejected mass. Our asymmetric outflow model will then be used as a starting point for a supernova explosion simulation to ultimately test if Cas A could have been a merger of two stars before going supernova.

Contents

1 Introduction **1**

2 Background **4**

2.1 Common Envelope 4

2.2 Star Models 4

2.3 Existence of Jets 7

3 Methodology **7**

3.1 Hydrodynamics 7

3.2 Relaxation of a Single Star 10

3.3 Binary Scan 11

3.4 Dynamical Simulation 12

4 Results and Analysis **15**

4.1 Varying Secondary Mass 17

4.2 Varying Radius 21

4.3 Varying Separation 22

4.4 Varying Resolution 25

4.5 Mass Lost 26

5 Conclusion **28**

5.1 Future Work 29

Appendices **31**

A Example of *splot* Routine **31**

A.1 pplot.f 31

A.2 grids.f 33

B Example of Super Mongo (SM) Macro **42**

Figures

1 Supernova Flow Chart 1

2 Asymmetry of the Spectra 3

3 A pictorial representation of Cas A evolution 3

4 Equipotentials and the Common Envelope 5

5 Hertzsprung-Russel Diagram 6

6 Column Density of Jets 8

7 Relaxation Particle Projection (Single Star) 11

8 Binary Scan of Two Stars 12

9 Energy and Time of Binary Scan 13

10 Column Density Plot of Merger 14

11 Velocity Distribution of Unbound Mass 17

12 Histogram of Ejecta Velocities Before and After Merge 18

13 Velocity and Distribution of Ejecta of Small $0.5M_{\odot}$ Secondary 19

14 Velocity and Distribution of Ejecta of Small $1M_{\odot}$ secondary 20

15 Velocity and Distribution of Ejecta of Small $2M_{\odot}$ Secondary 21

16 Velocity and Distribution of Ejecta of Large $1M_{\odot}$ Secondary 22

17 Comparison of Density for Differing Secondary Radii 23

18 Comparison of Velocity for Differing Separations 24

19 Comparison of Velocity Histograms in 8k, 24k, and 80k Resolutions 26

20 Comparison of Velocity Distributions in 8k, 24k, and 80k Resolutions 26

21 Mass Lost as a Function of Time 27

22 Time Series of 24k Primary Ejecta 28

1 Introduction

There are many brilliant stages within a massive star's life, and one of the most exciting is when it dies in a gigantic explosion called a supernova (SN). This is not the only way that a star can create such an outburst, however. Sometimes there are two stars in a binary system that merge into a new star which then goes SN. There are two main types of supernovae (SNe) classifications, with many subtypes, as shown in the flowchart in Figure 1. The differences between them involve their evolution and what elements are present within their post-outburst spectra. All of the SNe are considered to be core-collapse events when a massive star dies and the iron core inside of it collapses and rebounds. There is one exception to the rule, the Type Ia supernova, which involves a binary system with a white dwarf star.

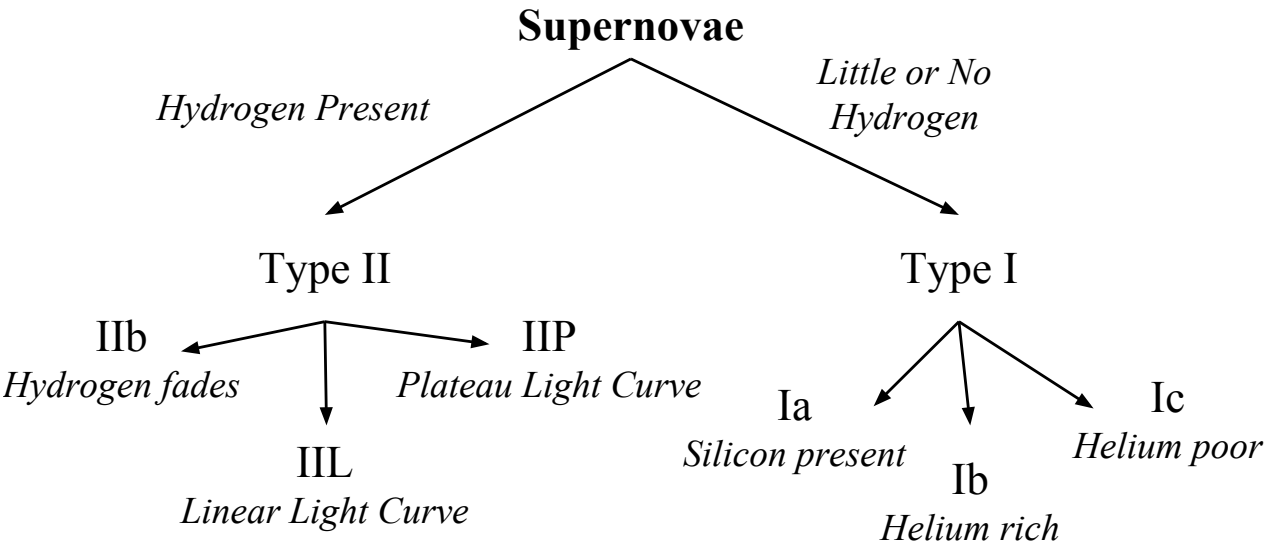


Figure 1: Flowchart of SNe Types. This chart outlines the different types of SNe and their fundamental differences in spectra. Every classification except the Ia is a core-collapse supernova, meaning one star dying in a large explosion (Ryden & Peterson, 2010).

Some of the most interesting SNe are ones that we have not actually observed explod-

ing. These are observed many years after their explosions and are referred to as supernova remnants (SNRs). The reasons these are so highly studied is the presence of shockwaves in the ejecta and the fact that SNe are the primary producer of cosmic rays. One particularly interesting SNR is Cassiopeia A (Cas A), a remnant in the Cassiopeia constellation. There are multiple reasons why Cas A is so fascinating: its young age, proximity to the Earth, and the mystery surrounding the system before the explosion. While there are SNe constantly being discovered and classified, Cas A is unique because of its age and proximity. It is classified to have been a Type IIb supernova, and it was a core-collapse SN that occurred in the Milky Way galaxy. It is the most recent example of a core collapse so close to Earth at only 3.4 kpc, about 11,000 lightyears away. Mystery surrounds the progenitor system, i.e. before it went supernova, because multiple models, e.g. a merged binary or a neutrino outburst, can explain Cas A. A review of this information is given in Krause et al. (2008) and references therein. The asymmetry of the SNR have lead to many studies and allow an exploration of SN physics as a whole. Figure 2 shows the asymmetry in the spectra of different wavelengths. To try and re-create an asymmetric product, we model the progenitor of Cas A as a binary system that merges into a single star, the star and unbound mass around it being asymmetric as well. That resulting star is what will go SN and produce the remnant we know as Cas A. This is represented in the drawing of Figure 3.

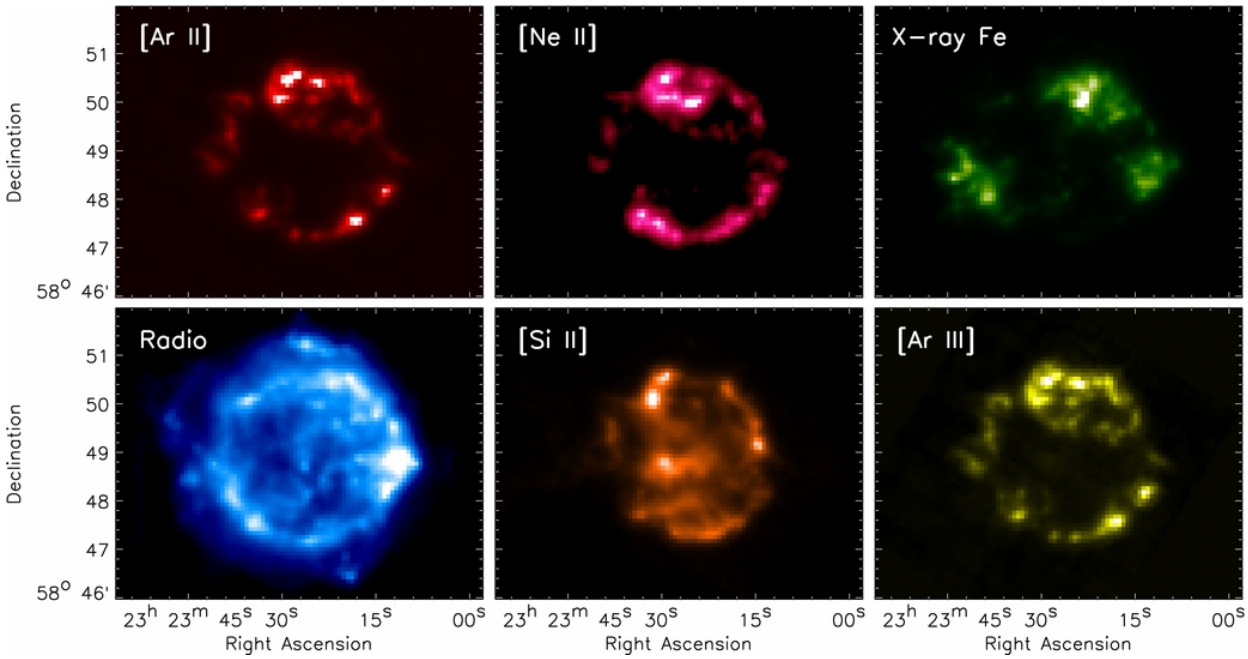


Figure 2: Asymmetry of Spectra. Different wavelength filters are applied to Cas A and are displayed and labeled. The most relevant aspect of these is the lack of symmetry in each filter. For example, the upper-left image shows a concentration of Argon II around the pole, but upper-right image shows a concentration of X-Ray Iron (Fe) in different areas.

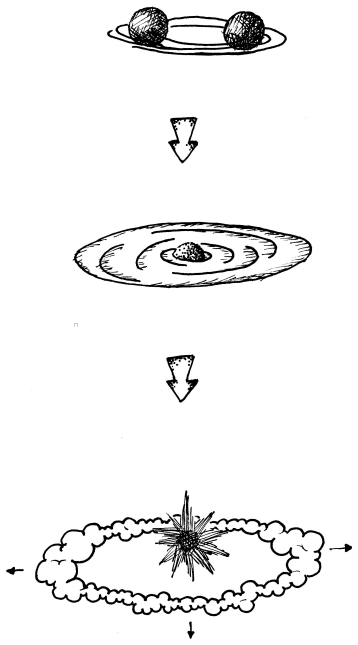


Figure 3: A pictorial representation of Cas A evolution. The first frame is our initial condition of two stars in binary orbit. The stars them merge to create a new object. We are able to use *Starsmasher* to simulate the first and second frames of the image. The goal is to set up sufficient initial conditions for a supernova simulation to be continued, thus completing the third frame of the image. (Seitanakis, 2017).

2 Background

2.1 Common Envelope

We make the assumption that the progenitor for Cas A was a binary system that had a SN occur from the product of a stellar merger. We use that basis to explore the common envelope problem as well as the physical processes within the simulation code that we use. The common envelope is a name for a phenomena in a system of two stellar bodies. If one draws effective gravitational equipotential lines about the two bodies, such as in Figure 4, one finds a figure eight shape in the effective equipotential line, mathematically derived in, e.g. Rasio & Shapiro (1991). Every point on this line has the same effective gravitational potential as the L_1 point as shown in Figure 4. Each teardrop shape is called a Roche Lobe, or inner Lagrangian surface, for the object inside of it. When the Roche lobe is filled, the mass of that body is unstable and starts to transfer to the other object. This is called a contact binary (Figure 4, left). Once mass is outside of the Roche lobes, it is not bound to a single star, but rather to the entire system until it reaches the second equipotential line. The mass bound to both bodies is said to be within the common envelope (Figure 4, right). There is still a lot to learn about the nature of common envelope systems and their ejecta, and it is the primary focus of the project.

2.2 Star Models

As previously mentioned, stars have several stages of life that can be displayed in a Hertzsprung-Russel Diagram (Figure 5). The life stages that we care the most about are the main sequence and the red giant branch. A main sequence star is classified as fusing hydrogen to helium

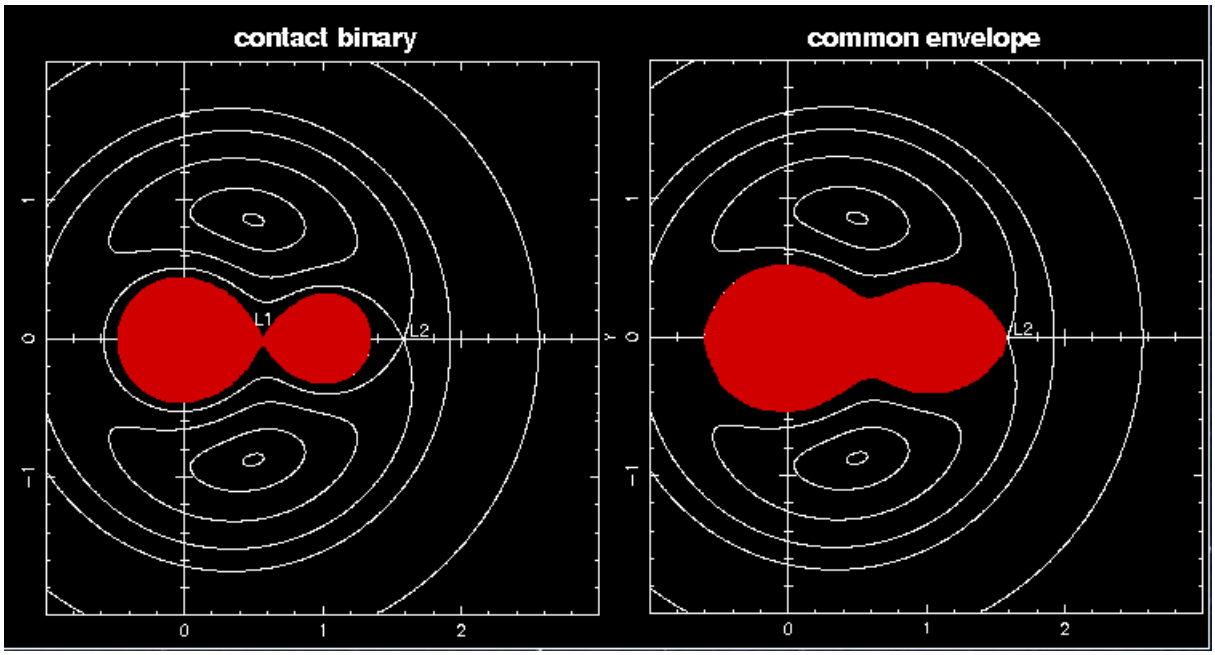


Figure 4: Equipotentials and the Common Envelope. The concept of a Roche lobe and the common envelope in a binary system are shown by effective equipotential lines around two bodies. On the left there is an image of two stars with their respective Roche lobes filled, where mass transfers through the center L_1 point. On the right there is an image of the common envelope being filled. As the stars merge, mass tends to be expelled through the L_2 point (Dhillon, 2012).

in its core. After depleting the hydrogen supply, the star expands to be a red giant as the nuclear reactions move to a shell surrounding an inert helium core. Any sun-like star will go from the main sequence stage to the red giant branch in its lifetime after the inner core stops fusing hydrogen into helium (Ryden & Peterson, 2010). A look at Figure 5 shows that the red giant stars are larger and typically have a lower temperature than a main sequence star.

We use a primary star that is approx 16 $M_\odot$, 300 $R_\odot$ (solar mass and solar radius, respectively) at an age of 10.3 Myr. The primary star is taken to be on the red giant branch and the secondary star is taken to be either a gas cloud approaching the main sequence, or a zero-age main sequence stars. One aspect to explore is seeing if it is possible for the gas

cloud to collapse to the main sequence in the time it takes the primary star to evolve to the red giant phase. From Kepler's third law, the time that it takes for a gas cloud to collapse into a main sequence star, the free-fall time, is given by

$$t_{\rm ff} \approx 4 \times 10^4 {\rm yr} \left(\frac{3 \times 10^{-15} {\rm kg \cdot m^{-3}}}{\rho_0} \right)^{1/2}, \tag{1}$$

where ρ_0 is the initial density of the gas cloud, and the other values were calculated as in Ryden & Peterson (2010). The free-fall time of approximately 10^4 yr is much less than the 10.3×10^6 yr age of the primary, so it is physically reasonable that a gas cloud could collapse to the main sequence in that amount of time.

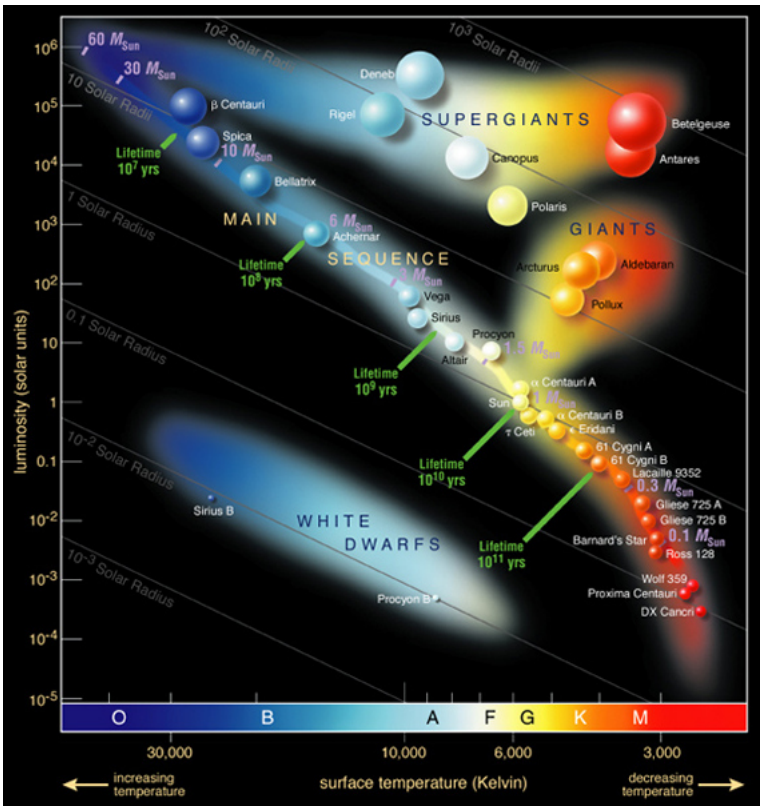


Figure 5: Hertzsprung-Russel Diagram. Stars of different stages in their lives are plotted with their luminosity and temperature along the vertical and horizontal axes, respectively. Diagonal lines of constant radius are also plotted. The spectral types are also displayed across the horizontal axis.

2.3 Existence of Jets

When you're a jet you're a jet all the way! Figure 6 shows the jets in a plot of the column density of the outflow. The outflow is primarily equatorial before the merger. After the stars merge, a distinct jet of matter comes from each of the poles and keeps expanding outward. The existence of jets in core collapse supernovae has been pointed to by many groups, e.g. Moreno Méndez et al. (2017), Pejcha et al. (2016), and Piran et al. (2017). One of the possible triggers of the jets is when there is a fast increase in radius of one of the stars, such as the onset of a red-giant phase like our primary star undergoes (Moreno Méndez et al., 2017). As the smaller star plunges into the red giant, the common envelope can be removed via the jet mechanism (Schreier & Soker, 2016). Jets have also been pointed to specifically in Type IIb supernovae, and as a way to explain the lack of hydrogen in the spectra of Cas A (Schure et al., 2008). The jets in our merger product can remain through the supernova simulation to be carried out at a later time.

3 Methodology

3.1 Hydrodynamics

We use a smoothed particle hydrodynamics (SPH) code, *Starsmasher*, to model interactions of gases in stars, specifically merging stars. Any form of hydrodynamics uses Euler's equation for non-viscous flow

$$\frac{d\vec{v}}{dt} = -\frac{\vec{\nabla}P}{\rho} + \vec{g}, \tag{2}$$

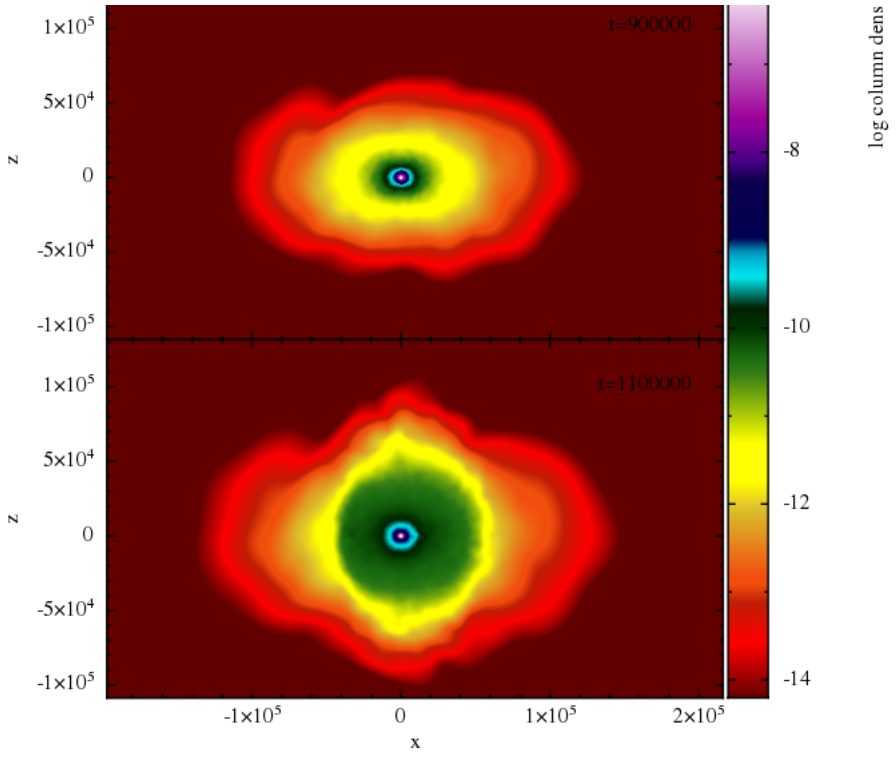


Figure 6: Column density plot showing polar regions before (top) and after (bottom) merger. The z and x axes are scaled with solar radii and represent position. The right hand bar is the log of the column density of the system, in terms of $M_\odot/R_\odot{}^2$ with the red being a less dense area, and white/pink being less dense. This illustrates the outflow of gas from the system as the mass at the poles shoots through the other ejecta.

where ∇P is the pressure gradient, ρ is the density and g is the acceleration due to gravity. The right hand side of Equation 2 has two contributions to acceleration, the first term is hydrodynamic and the second is gravitational. The pressure gradient accelerates the material from high pressure toward low pressure, hence the minus sign. The pressure is from a tabulated equation of state based on the environment. The system has a low physical viscosity, and since it is negligible we use Equation 2 rather than the Navier-Stokes equation. The code also has a numerical viscosity which we try to minimize.

To utilize hydrodynamics, *Starsmasher* models individual particles of mass to simulate the motion of a fluid (Monaghan, 1992). The positions of the SPH particles, as well

as their velocities, local pressures, and masses are tracked in the code and can be thought of as tracers of the fluid flow. An advantage of Lagrangian codes like SPH is they do not waste computational resources on empty parts of space like the large distances between stars before merging. A taste of the computational methods are be given below; for the sake of brevity and clarity, however, many details are being brushed over. For a more comprehensive exposition of the *Starsmasher* code, see Appendix A of Gaburov et al. (2010). There they write down a Lagrangian for the system and apply the Euler-Lagrange equations to derive the SPH equations of motion.

The individual SPH particles have a finite range of where they are hydrodynamically influenced by other particles, to be less computationally expensive. If every SPH particle had to interact with every single other one, calculations for a single time step would take an extremely long time. To minimize the amount of calculations necessary, a numerical "smoothing length," h, is calculated (Rasio & Shapiro, 1991), and the radius of interaction for a single particle is $2h$. The smoothing length is related to the amount of neighbors that a particle has. This means that there are only so many interaction calculations that have to be done per SPH particle. An SPH particle in a numerically dense region has a smaller smoothing length than one in a low density region; the result of this is that the smoothing length varies with time and position. It should be noted that gravitational forces act between all SPH particles, and only the hydrodynamic interactions are limited.

The density of an SPH particle i for a given h is given by

$$\rho_i = \sum_j m_i W(|\vec{r_i} - \vec{r_j}|, h_i), \tag{3}$$

where W is the kernel function that depends on the distance $r \equiv |\vec{r_i} - \vec{r_j}|$ between particles

i and j and h_i is the smoothing length of particle i. There is then a summation over all j particles within the kernel. In the case where the smoothing length is constant, we can then calculate the acceleration on particle i due to all particles j as

$$a_{\mathrm{h},i} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} \vec{\nabla}_i W_{ij}(h_i) + \frac{P_j}{\rho_j^2} \vec{\nabla}_i W_{ij}(h_j) \right), \tag{4}$$

then using Newton's second law, we can find the force on particle i from a single particle j as

$$F_{ij} = -m_i m_j \left(\frac{P_i}{\rho_i^2} \vec{\nabla}_i W_{ij}(h_i) + \frac{P_j}{\rho_j^2} \vec{\nabla}_i W_{ij}(h_j) \right). \tag{5}$$

3.2 Relaxation of a Single Star

Each star starts as a system of mass particles in a hexagonal close packed lattice. This stage is where we set parameters such as the star's mass, radius, and age. The system is allowed to relax, or reach an equilibrium. The difference between the initial placement of particles and the relaxed star is shown in Figure 7. The equilibrium is reached after the system is evolved allowing particles to interact with each other. It should be noted that we typically have two different stars with different parameters in a collision simulation. A star typically has on the order of $N = 10^5$ particles and that number can be adjusted depending on the resolution we want. More particles means we get a more accurate depiction of what is physically happening, but computation takes a longer amount of time. The time per each iteration scales like N^2, which occurs when gravity is the bottleneck computation. Maintaining accuracy while keeping the time of a simulation reasonable is a large challenge of computational physics. One could write a code that is physically accurate but takes months or years to run, and then even more months to adjust parameters, which is less useful for

the progression of science.

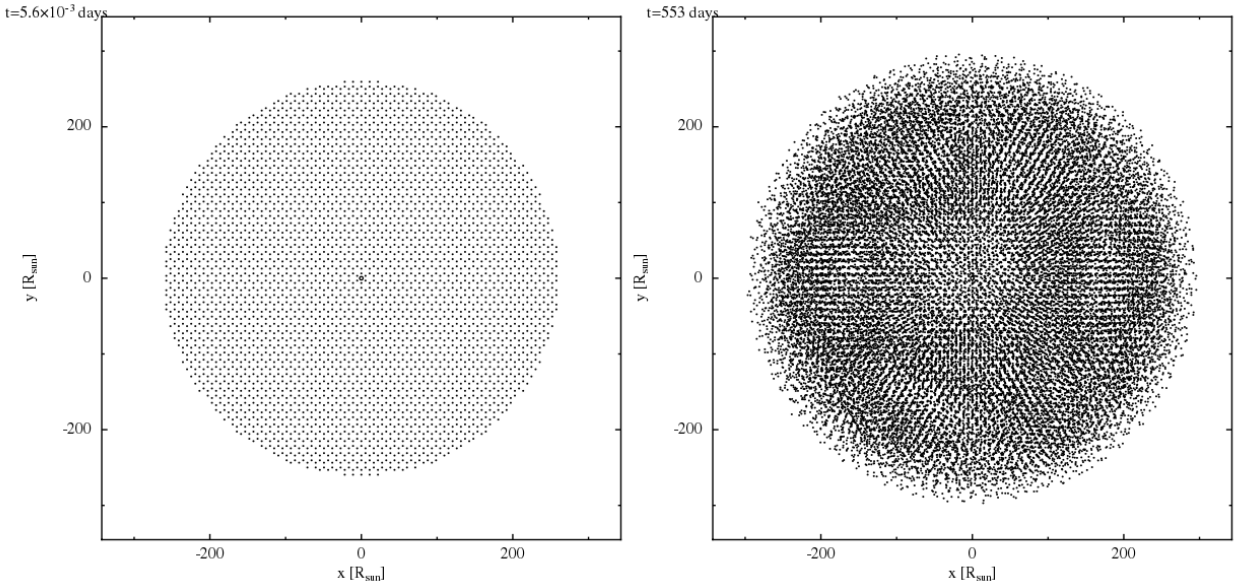


Figure 7: Particle projection of relaxation of a single star. On the left side we have the initial star before the relaxation and on the right side we have the star post-relaxation. The x and y axes are scaled with solar radii and represent position. The open circle in the center is a point mass to aid in calculation as it would be too dense to be represented by multiple SPH particles without computational strain.

3.3 Binary Scan

After the relaxation, the stars are placed near each other and are slowly marched closer to one another in what we call a binary scan run: "binary" because it is two stars, and "scan" because we are scanning the relative positions of the stars, shown in Figure 8. The speed of the stars throughout the scan is adjusted so a circular orbit is maintained. The binary scan brings the two stars together along a central axis until mass begins to transfer through the L_1 Lagrangian point and the system of stars is considered a contact binary (see Introduction). We bring the two stars together until the system reaches a minimum total energy as shown in Figure 9.

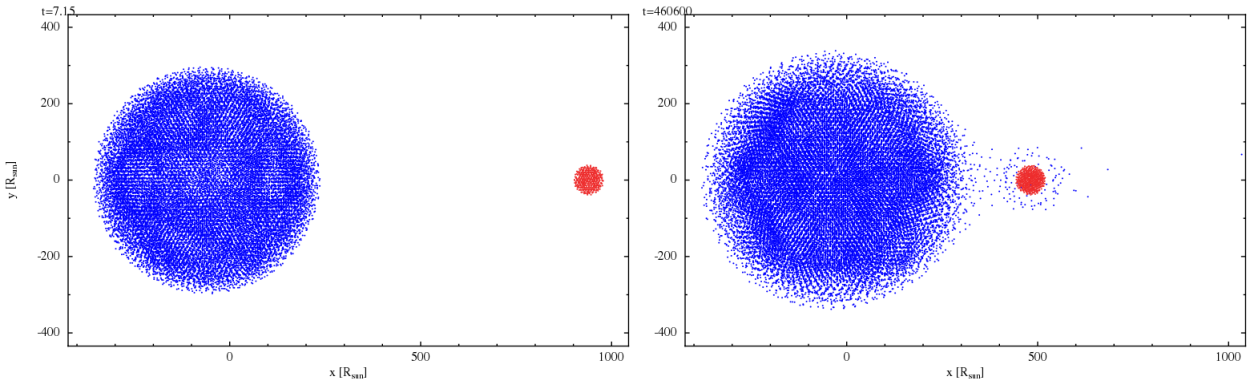


Figure 8: Binary Scan of Two Stars. An example of a primary and secondary star being scanned together in a binary scanning run. The stars are brought together until the mass starts to transfer, meaning one star has reached its Roche Lobe limit.

3.4 Dynamical Simulation

Following the set up, the particles are allowed to interact freely and the stars allowed to collide as we monitor the results. As the two stars orbit the center of mass, they get closer together and orbit faster. The second star begins to plunge into the first one, and a trail of mass is expelled. During the merge process we use a software called *splash* to observe particle interactions, follow the mass of individual stars, and look at the density of the gas flow (Price, 2007). We used *splash* to create Figures 7, 8, and 10, for example. *Splash* is a useful tool for when the results do not go as expected because we can track individual particles in it. If a certain iteration is taking a long time, we can look at the energy output and determine which particle is causing the hold-up and find that particles position in *splash* to see what else is going on physically around it. In Figure 10, we have a column density plot of the two stars merging. The left panel is the same as the right panel of Figure 8, and the right panel shows them after some time about to merge together.

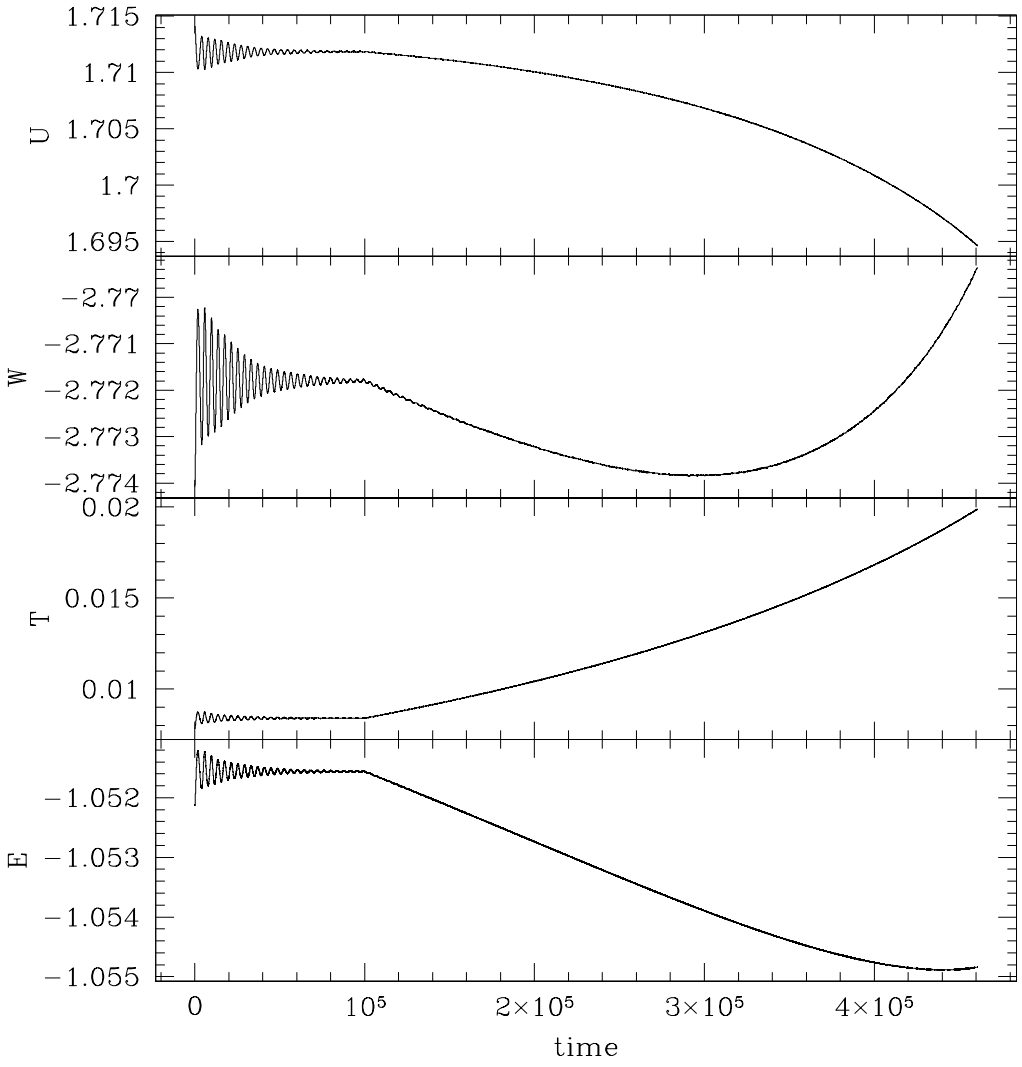


Figure 9: Graph of energy versus time for a binary scan of a 16 $M_\odot$ and 1 $M_\odot$. The plots go from top to bottom: internal energy, U, gravitational potential, W, kinetic energy, T, and total energy, E, all in units of $GM_\odot{}^2/R_\odot$. The end of the simulation is triggered when a sufficient amount of matter passes through the L2 point. In this case, the time is approximately 4.6×10^5 time units $[\sqrt{R_\odot^3/GM_\odot}]$. The oscillations occur when the stars are held at a fixed separation (before time 10^5 time units). We use snapshots from this simulation to initiate the dynamical simulation.

Once the modeling of the interactions is complete, the data are extracted to model the radial velocities and distribution of outflow. This process is done using a routine called splot, which can be customized to analyze an assortment of parameters. One of the common

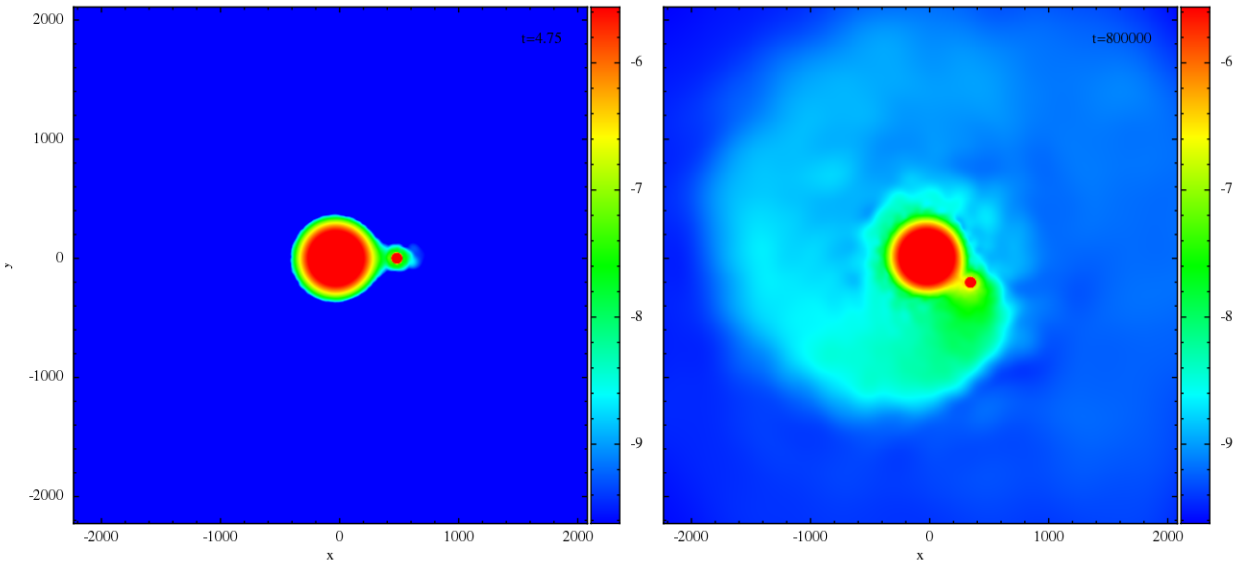


Figure 10: Column Density Plot of Merger. After the stars are scanned close to one another, we set them on their orbits and allow them to merge. The x and y axes are scaled with solar radii and represent position. The right hand bar is the log of the column density of the system, in terms of $M_{\odot}/{R_{\odot}}^2$ with the red being a more dense area, and blue being less dense. This illustrates the outflow of gas from the system.

uses for splot is to create light curves of the data and compare to observational light curves. However, for Cas A simulations, we did not create light curves as there is no observational data before the supernova. We are more interested in the way the object evolves and the outflow and density around the surface is evolving with it. The splot we use works by looking at all of the particles and the velocity near the surface of the star to determine the local density around the surface. A splot analysis is quicker than a typical hydrodynamic simulation as it can be done for a single time step (or multiple) and it can be made more efficient as not every interaction is necessary depending on what information is needed, e.g. since we are only interested in the density in the ejecta, we can neglect particles in the core of the system.

Shortly after analysis, new simulations are underway to use different parameters to

refine models. Running simulations is a valuable research method as it allows us to adjust different parameters, such as mass of stars, radius, and periastron separation to test their effects and best match the data after the objects collide. These details tell us about the progenitor systems and can assist future astronomers in looking for similar objects. The specific models used are outlined in Table 1 and 2. We hope to evolve a system accurately to be used to model the SN physics of Cas A as further research elsewhere.

Mass [$M_\odot$]	Radius [$R_\odot$]	Particle Number
16	300	80,000
16	290	24,000
16	270	8,000

Table 1: Primary stars used in collisions

Mass [$M_\odot$]	Radius [$R_\odot$]	Particle Number
0.25	0.2	50
0.5	0.3	50
1	0.4	50
1	40	1,000
2	0.55	50

Table 2: Secondary stars used in collisions

4 Results and Analysis

When looking at the current results, we find an interesting physical feature which is shown in Figures 11 and 12. Shortly following the merger, a wave of mass begins to be pushed through at the poles with high velocity, faster than the equator or any intermediate surface. This phenomenon was also hinted at in Figure 6, but the radial velocity is a more concrete

and quantitative example. We investigate the effect that parameter space has on these jets of matter and mass loss as a whole. The parameter space we look at involves changing different aspects with respect to the secondary star. We change the mass of the secondary, the radius (i.e. main sequence or pre-main sequence), the separation of the two stars, and keep an eye on mass outflow.

The two main ways we present the data are in histogram form and a radial velocity distribution curve. The histogram is shown in Figure 12. The advantage to displaying the data like this is that there is a full range of positions shown in terms of being polar, equatorial, or in between. It also shows the percentage of the mass that moves at that radial velocity or at a lower velocity, thus we know if all of the mass is moving out in a jet or just a small percentage of it with a high velocity. A disadvantage of the histogram is that it does not portray any form of waves of matter moving outward. In other words, we do not know the radial distance of the unbound mass, merely that it is moving radially at a certain angle from the pole. The radial velocity distribution curve does allow us to look at waves of matter moving outward radially. However, it is not as detailed since there are only 5 discrete values of the angle that are used for the simplicity of legibility. Using these graphs in conjunction gives a fairly clear picture of what is going on in terms of mass ejection from the merging system.

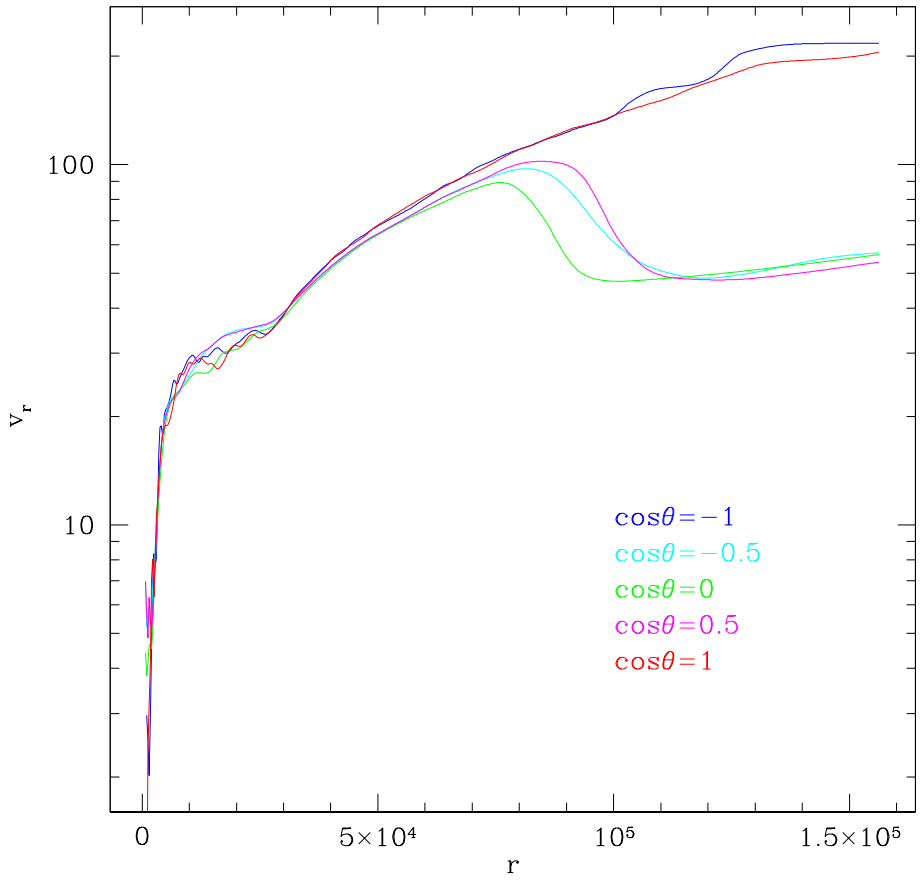


Figure 11: Velocity of unbound mass in different radial zones. The radial velocity, v_r is given in km/s and the radius, r, is given in $R_\odot$. The lines are color coded to represent different regions of the outflow based on the cosine of the angle from the z-axis. We see that the mass is moving faster in the polar directions past a distance of 8×10^4 $R_\odot$ as the curves diverge. This points to the existence of outflow jets.

4.1 Varying Secondary Mass

The secondary masses tested are listed in Table 2. The 0.25 $M_\odot$ star did not merge with the primary over the time evolved. To be consistent with comparison, we use the same amount of time after the merge to collect the data for the histogram and velocity distributions. The time post-merge is 330 output files or 33,000 time units $[\sqrt{R_\odot^3/GM_\odot}]$ of the code. This was arbitrarily determined from the $1M_\odot$ simulation's runtime. The masses varied are stars on

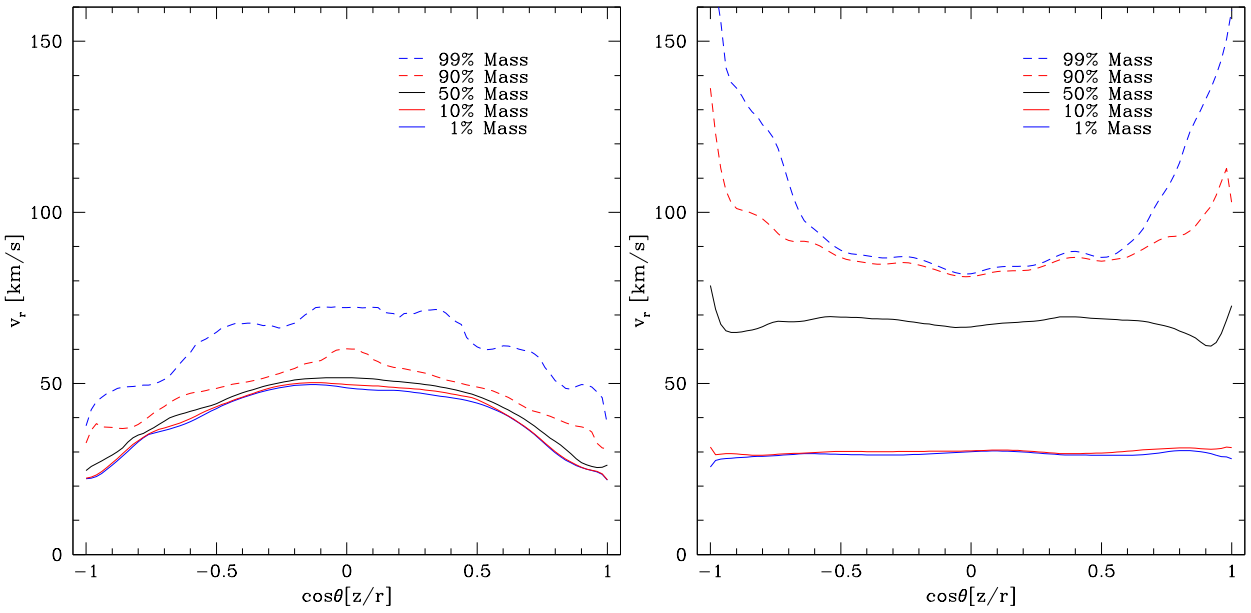


Figure 12: Histogram of ejecta velocities before (right) and after (left) merge. The radial velocity as a function of the cosine of the polar angle is plotted. The x-axis corresponds to the polar angle. $\cos\theta = \pm 1$ are the poles of the merger product, and $\cos\theta = 0$ is the equatorial plane. Each line represents an amount of mass that is contained within the bin. The black line is the median line, meaning 50% of the ejected mass is below it. The blue lines contain the central 99% of mass and the red lines contain the central 10% of mass. Before the merge, the velocity is primarily equatorial and the bulk of the mass has a similar speed. After the merge, there is a clear indication that all of the mass at the poles has a higher velocity than the equatorial mass.

the main sequence. These runs are all 80k particles for the primary star and approximately 50 particles for the secondary. Having such a numerically dense secondary made these simulations extremely time intensive, approximately 2-4 weeks from relaxation to dynamical simulation.

For the $0.5 M_\odot$ secondary, we see distinct jets in both sides of Figure 13. The peak velocities of the jets are 150 km/s and the jet is contained within 2×10^4 $R_\odot$. The jet is illustrated in all curves except the 1% curve, this means the slowest mass is still expanding on the equatorial plane rather than through the poles. In the radial distribution shows a

similar story. Since the peak of the red and blue lines is above the distant mass radii, the new mass ejected has to travel through the already present mass (think back to Figure 6 and how the gas was before the merger).

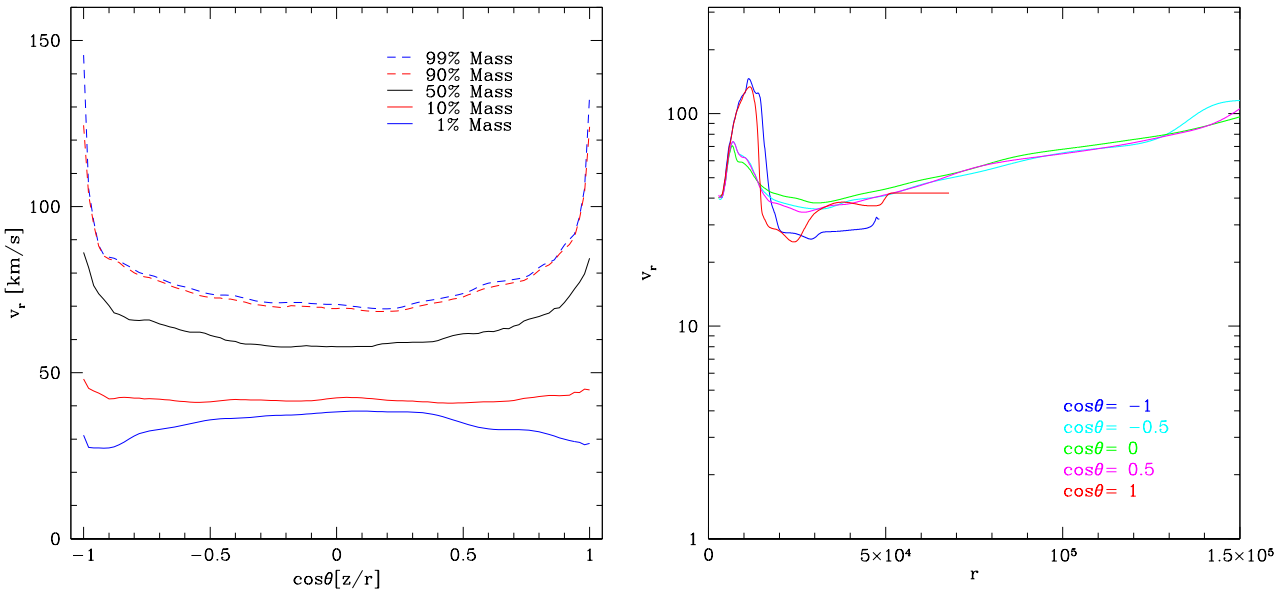


Figure 13: Radial velocity and its polar and radial distribution of $0.5M_{\odot}$ Secondary. The left histogram shows peak jet velocities at 150 km/s. The right plot shows that this first wave of ejected mass peaks at 1.2×10^4 $R_{\odot}$.

For the $1M_{\odot}$ secondary, we also see jets in both both poles of Figure 14. The peak velocities of the jets are again 150 km/s and the jet is again contained within 2×10^4 $R_{\odot}$, peaking around 1×10^4 $R_{\odot}$. We do not see jet behavior in the red and blue lines of the histogram, while we do see it in all the lines above the median. This simulation has more drastic jet behavior across a wider range of polar angles. The mass at the poles (red and blue on the right side) reaches out to a farther radius than in the 0.5 $M_{\odot}$ case. This is because the larger mass is more disruptive to primary the star than the smaller mass and creates a more outflow before the merge occurs.

For the $2M_{\odot}$ secondary, we see distinct jets in both sides of Figure 15. The peak

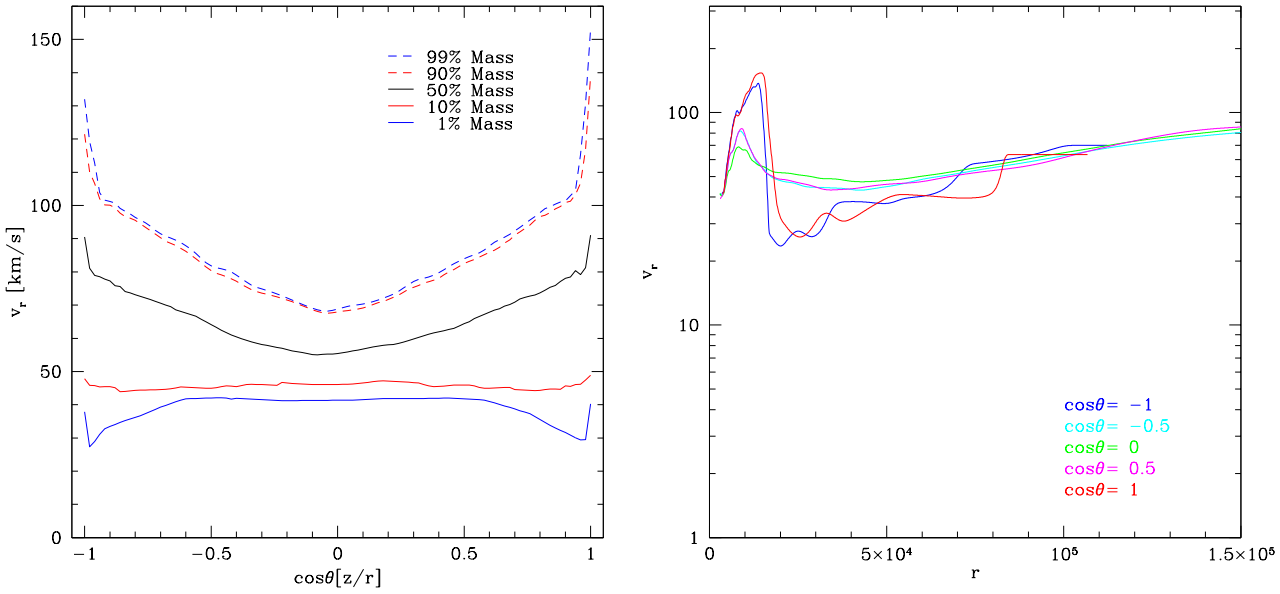


Figure 14: Radial velocity and its polar and radial distribution of $1M_{\odot}$ Secondary. The left histogram shows peak jet velocities at ∼150 km/s. Even the median of the unbound mass follows the jet pattern. The right plot shows that this first wave of ejected mass peaks at 1.5×10^4 $R_{\odot}$ is contained within 2×10^4 $R_{\odot}$. The red curve has a noise artifact in the histogram plot that is not present in any other plots.

velocities of the jets are 140 km/s and the jet is contained within 2×10^4 $R_{\odot}$. The histogram shows a peculiar behavior contrary to the other masses. The median line still has remnants of equatorial ejection and the 1 and 10 % lines have a low radial velocity compared to Figure 13. In the radial distribution does not bring up anything seemingly suspicious to explain this behavior. All of the polar tails of the velocity curves are at or above the peaks of the non-polar velocities. It could be that the larger mass secondary causes a longer time between the merge occurring and the jets being visible. This is also suggested by the tails of the velocity curves, as the equatorial mass closest to the origin is not moving faster than at the farthest radial distance.

We see that as we increase the mass of the secondary, there is ejected mass farther from the center in all directions. The jet behavior also gradually decreases in the 1 and 10%

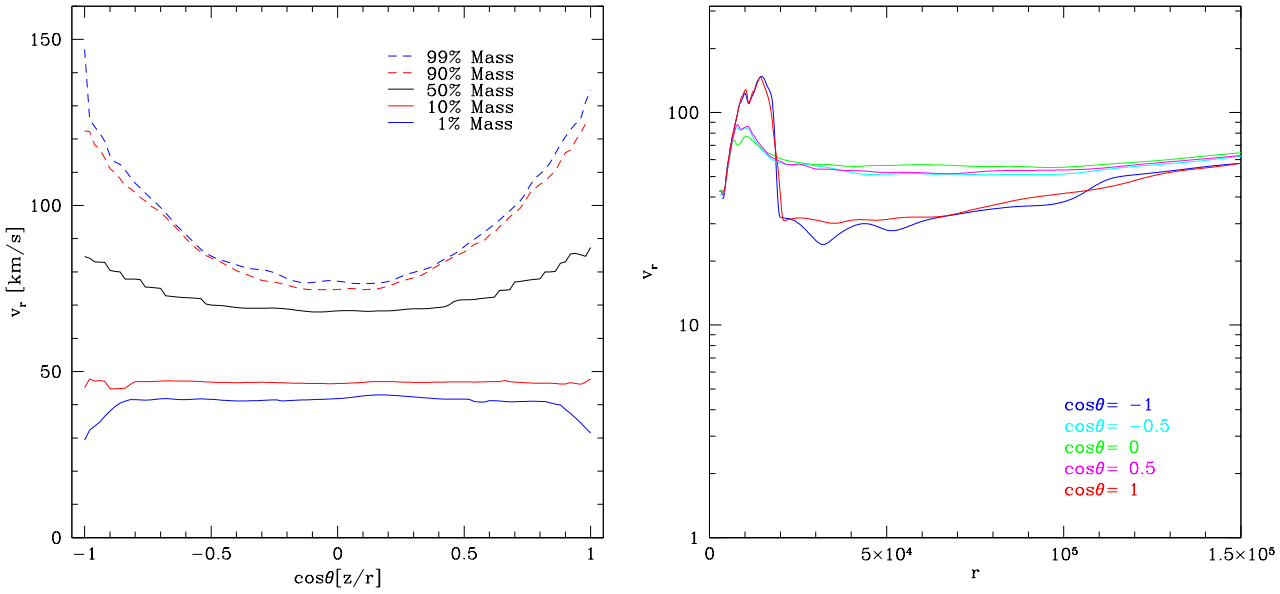


Figure 15: Radial velocity and its polar and radial distribution of $2M_{\odot}$ secondary. The left histogram shows peak jet velocities at 150 km/s. There is not jet behavior in the lower lines of the histogram.

lines with more mass. The peak pole velocity for all simulations is about 150 km/s and the radial distance of the jet's peak velocity is about 1×10^4 $R_{\odot}$.

4.2 Varying Radius

When comparing the differences between Figures 14 and 16, we see that the radial plot on the right hand side are nearly identical in terms of where the peak velocities are. However, the histogram shows that for the smaller secondary radius, more of the mass has started flowing in the jet pattern. This is indicated by the 1 and 10% lines still following an equatorial behavior with a peak around 0 on the x axis. Even when looking at the radial density distribution of the ejected mass, in Figure 17, we see little difference. The differences in the curves occur at such low densities and high distances from the jet that they are negligible. It is worth nothing that the peak in the jet velocity occurs where the density curves start to

diverge. This makes sense physically as the jet of more dense material propagates through less dense material.

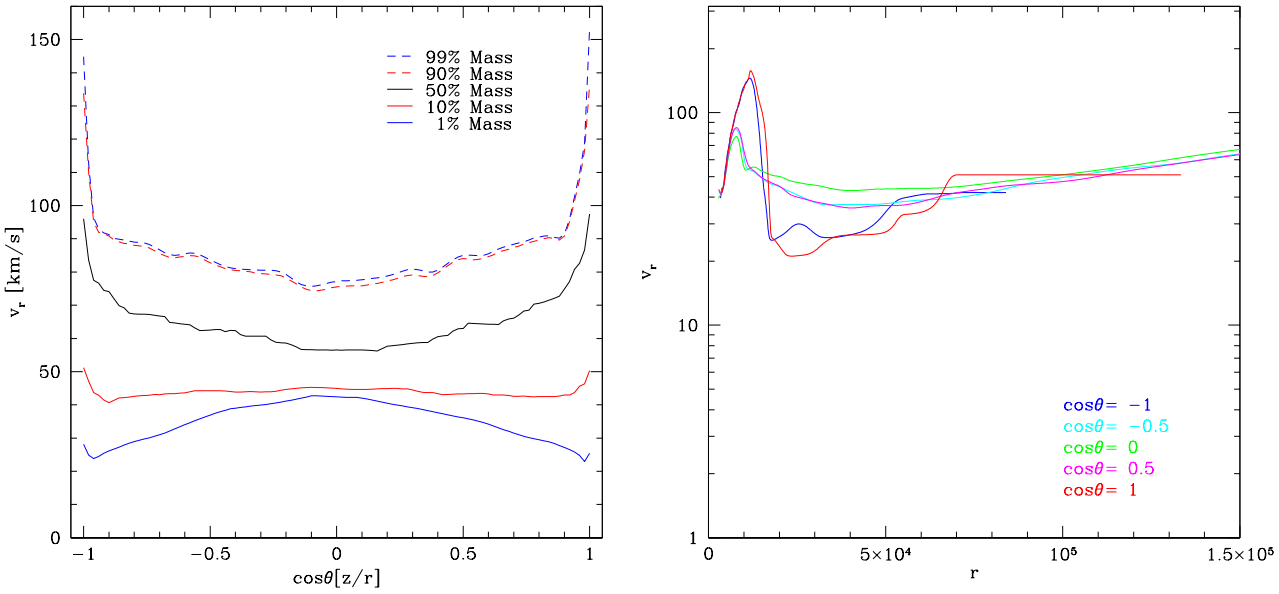


Figure 16: Radial velocity and its polar and radial distribution of large $1M_{\odot}$ Secondary. The left histogram shows peak jet velocities at 160 km/s. All of the unbound mass follows the jet pattern. The right plot shows that this first wave of ejected mass peaks at 1.2×10^4 $R_{\odot}$ is contained within 2×10^4 $R_{\odot}$.

4.3 Varying Separation

We compare the separation between the two low resolution simulations of the $1M_{\odot}$ secondary. The separations used were 512 and 531 $R_{\odot}$. The histograms of the systems are compared in Figure 18, with the larger separation being on the left hand side. Both the histograms and velocity distributions show drastic differences. There is no distinctly equatorial outflow behavior in the farther separation (left) of Figure 18. We see by looking at the velocity distribution that the jet at the poles is moving radially past the underlying matter that was there before the merge. On the right hand side, we see familiar behavior, with some

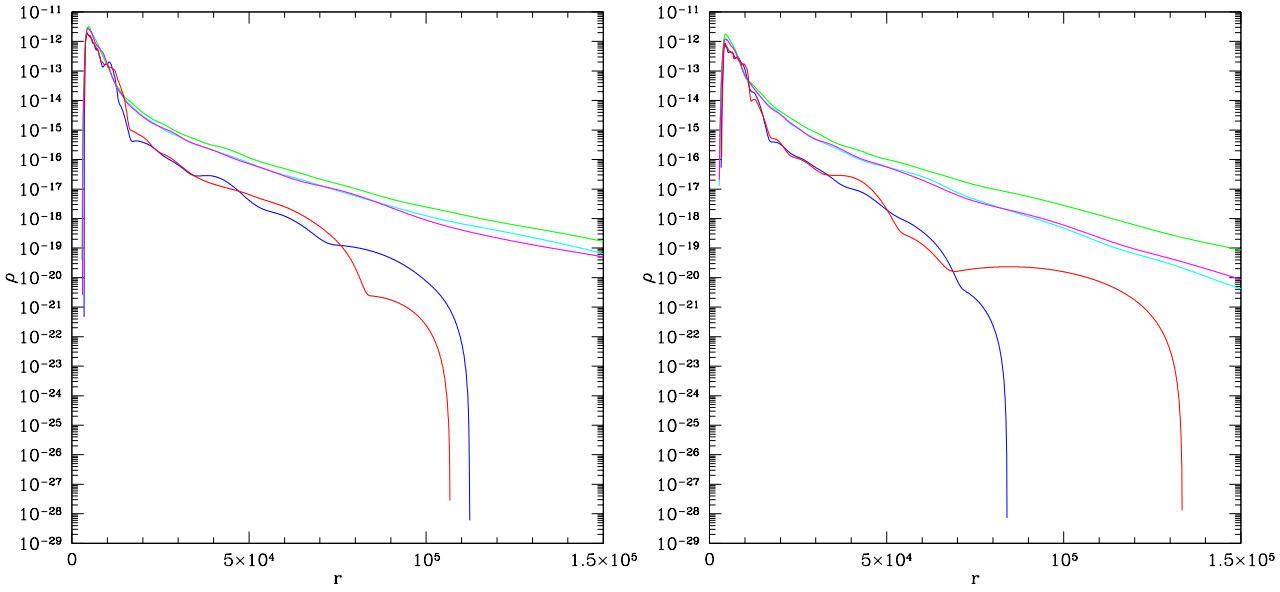


Figure 17: Comparison of Density for Differing Secondary Radii. Each plot shows the density of ejected mass with respect to the radius. The plots show similar peaks in density at 5×10^{-24} $M_\odot/R_\odot$ with a radius of 0.5×10^4 $R_\odot$). The red and blue curves between the two are different. However, they occur at such a low mass that the effect is negligible on the outflow

equatorial outflow remaining and some jet behavior in the faster mass.

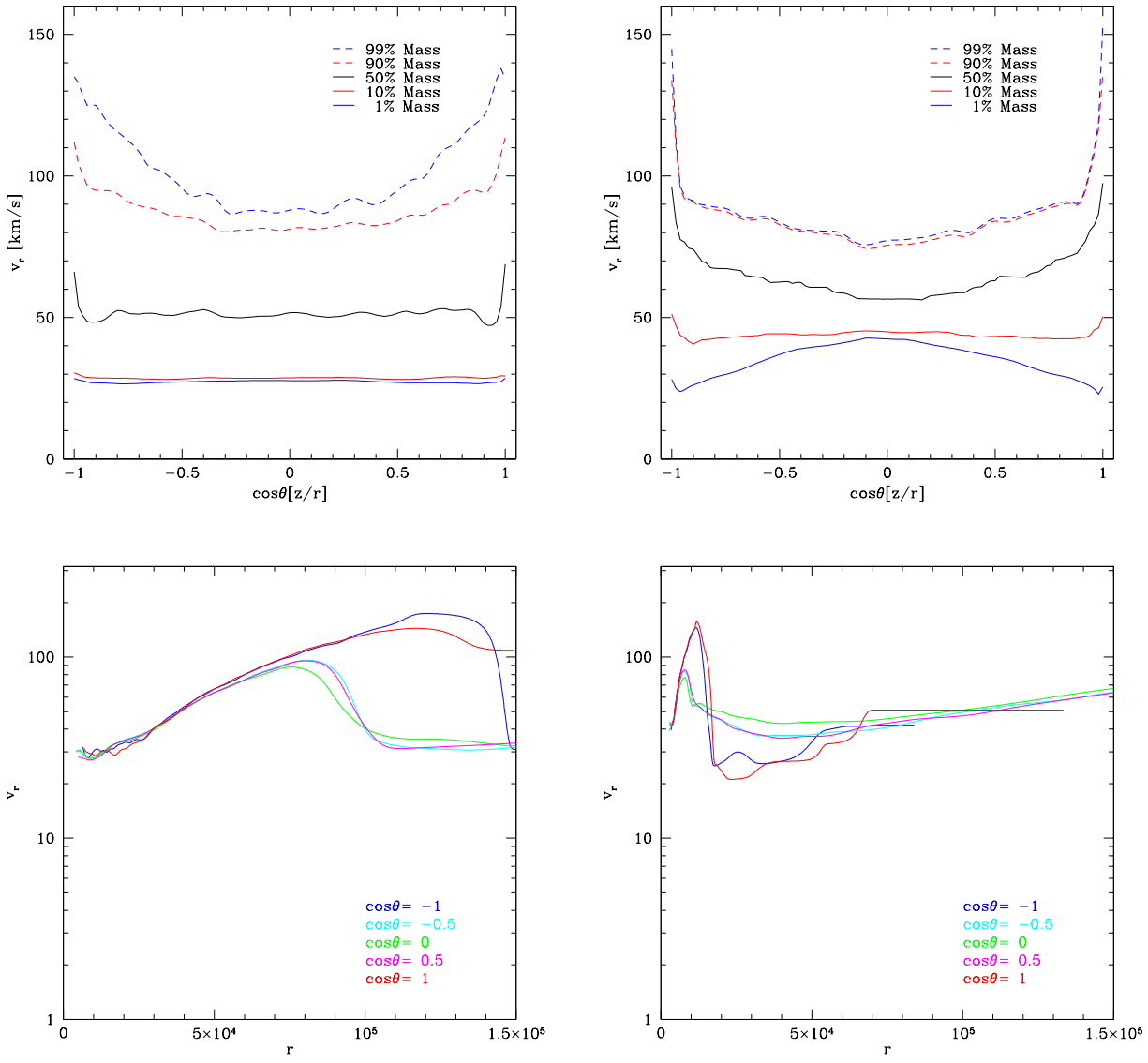


Figure 18: Comparison of Velocity Histograms and Distributions for Differing Separations. The 531 $R_\odot$ (left) and 512 $R_\odot$ (right) show large discrepancies between both the histograms and velocity distribution curves. Using a different separation has a drastic effect on the jet behavior and distribution.

4.4 Varying Resolution

When running simulations, there is a constant struggle between a number of particles and the time it takes to run the simulations. Frequently we explore parameter space in smaller resolution studies then translate them to higher numbers of particles. We used 8k, 24k, and 80k particles for a comparison of results. The only thing changing between the simulations was the primary star's number of particles. Each system is the $16M_{\odot}$ primary relaxed with the appropriate number of particles and the $1M_{\odot}$, $16R_{\odot}$ secondary with 1000 particles. This secondary was used because relaxing such a large star with so few particles becomes problematic in terms of getting it to the right size. The histogram comparisons are shown in Figure 19. The 8k simulation is the left, the 24k in the middle, and the 80k on the right. As we use more particles, we are becoming more similar to what we assume is accurate. The 8k particle simulations are different in the velocity histograms in terms of peak velocities and the presence/distinction of the jets. A possible explanation for this is that the relaxation of the primary star at 8k particles is $\sim 30R_{\odot}$ less than it should be. This could lead to a star that is less likely to be perturbed in its outer layers thus jets would not be able to form as easily. Similarly, we see the less active jets in Figure 20. The 8k particle run has no significant difference between the curves at the poles and those at the equators. However, there is a strong similarity between the curves of the 24k and 80k simulations, meaning that hopefully adding more particles would also lend to more accurate results.

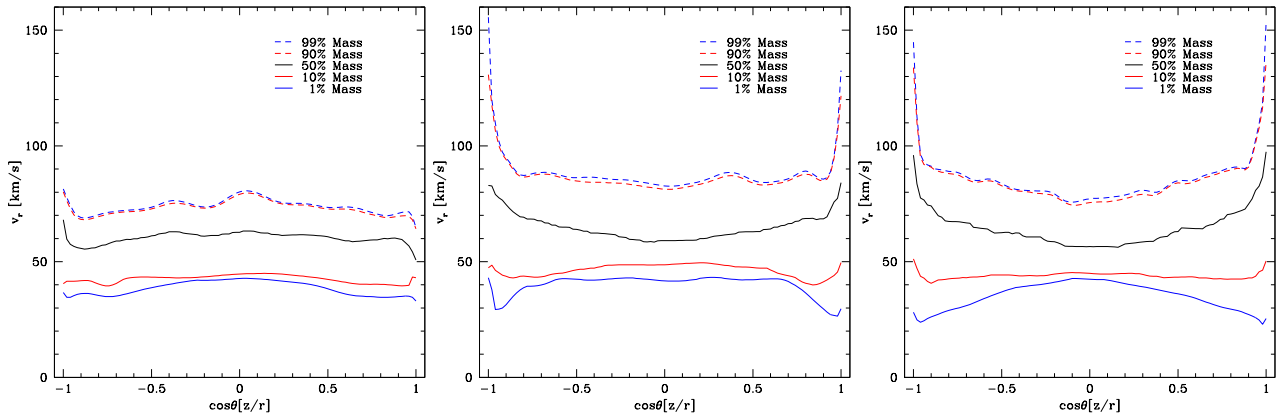


Figure 19: Comparison of velocity histograms in 8k, 24k, and 80k resolutions. Each simulation used the same $1M_{\odot}$ secondary, while the primary star's number of particles was altered. The separation criteria was the same for all simulations. The features between the 24k and 80k simulations are similar in shape and qualitatively.

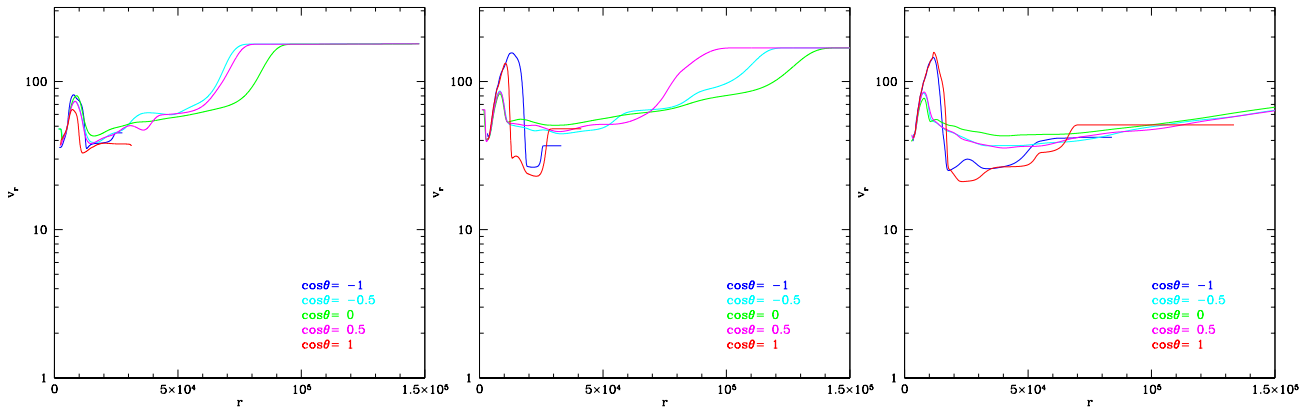


Figure 20: Comparison of velocity distributions in 8k, 24k, and 80k resolutions. The curves become more detailed and extend to farther radii when the number of particles is increased

4.5 Mass Lost

As we consider what is happening during the merger process it is important to remember that we are focused on the unbound mass. Thus we must also take mass lost from the system into consideration. The mass loss curve of Figure 21 shows that the ejected mass increases before eventually reaching a plateau. The merge of the system occurs in between the two vertical lines which causes the large increase in the amount of mass lost from the system.

Snapshots for data analysis are taken at the three points indicated in the figure. The first point is right before the merge, the middle point is following the merge, and the last is after the mass loss has sufficiently plateaued. When it reaches the plateau, the jet behavior continues! This is exciting because it indicates the effect of the jets is real and long-lasting, as shown in Figure 22. Even after the lines have converges as in the velocity distribution on the far right side, we see jet behavior in the 90 and 99% lines, which is a strong indication that jets are real. We see the impact that the jets have on mass loss in Table 3, this also acts as a useful reminder of the simulations presented.

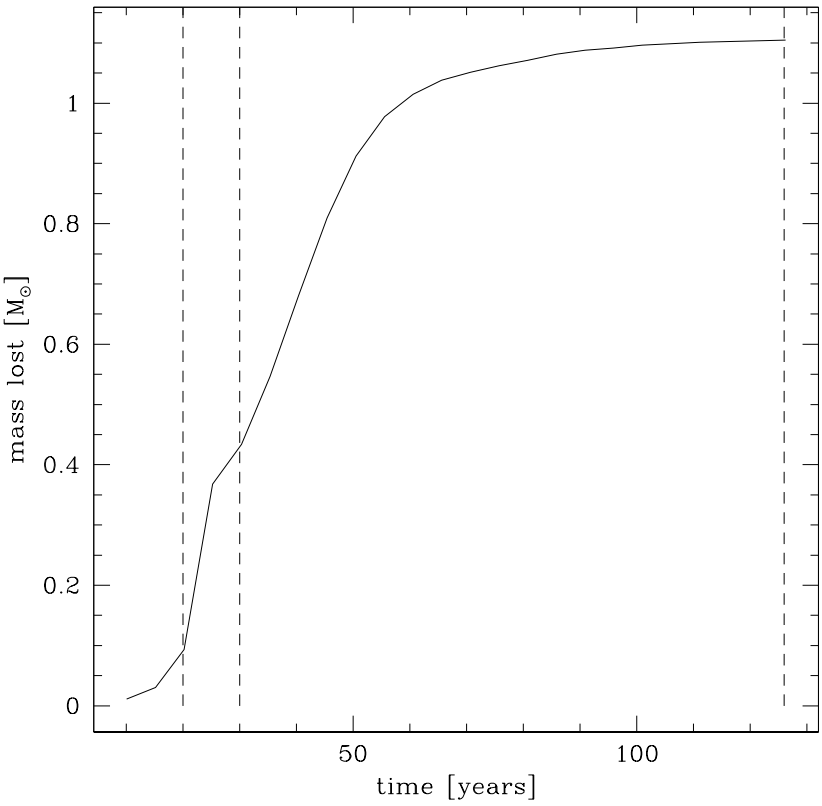


Figure 21: The unbound mass as a function of time. The vertical dotted lines are picked to represent before the merge, right after, and significantly after. These lines represent points in the simulation that are further discussed in the text and Figure 22.

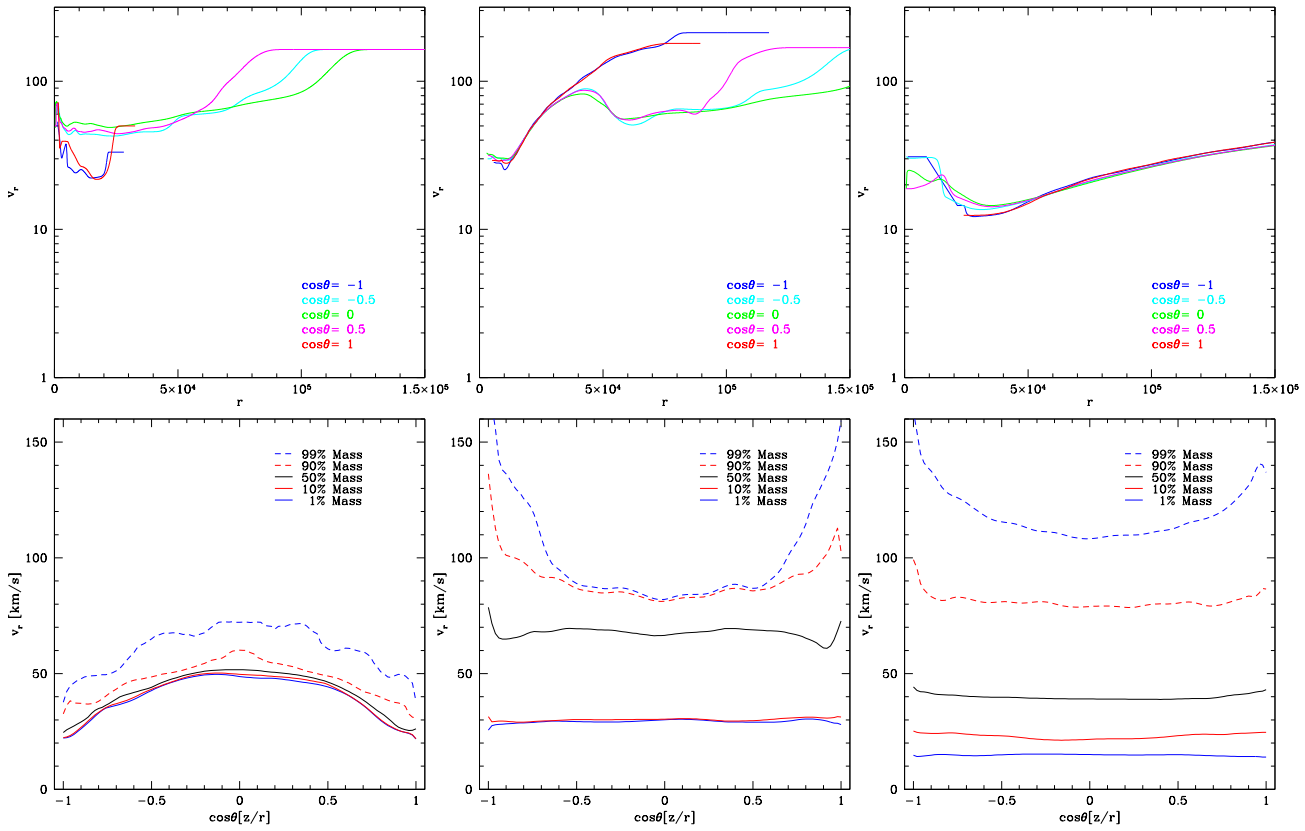


Figure 22: A time series of mass ejected from 24k primary merger. Histograms and velocity distribution plots at 20, 30 and 126 years from start of simulation. Before the merge we see the familiar equatorial ejecta, then the jet behavior right after. However, long after the merge we see the histogram lines still indicate jets even though the velocity curves have converged.

5 Conclusion

Overall, there are many things that can impact the jets in a binary merger simulation. The mass of the secondary star impacts how quickly the jets form and how they propagate through the matter. This makes sense as the more massive stars are better able to disrupt the outer layers of the large star. The radius of the secondary has surprisingly little effect on the jets themselves, but does impact the amount of unbound mass. The separation of the stars also had a drastic effect on the outflow. The mass loss from the system is impacted based on all of these areas of parameter space. Since the jet behavior of the outflow is present

Primary Radius [$R_\odot$]	Primary Particles	Secondary Mass [$M_\odot$]	Secondary Radius [$R_\odot$]	Separation [$R_\odot$]	Mass Lost [$R_\odot$]
300	80k	0.5	0.3	485	0.25
300	80k	1	0.4	513	0.63
300	80k	2	0.6	560	1.72
300	80k	1	0.4	513	0.63
300	80k	1	40	512	0.35
300	80k	1	40	512	0.35
300	80k	1	40	531	0.60
300	80k	1	40	513	0.63
290	24k	1	40	492	0.36
270	8k	1	40	445	0.26

Table 3: A summary of the results is given. Each sub-section represents a subsection of the results chapter, varying mass, radius, separation, and resolution respectively. The mass lost and radial extension are taken at the same time 33000 time units after the merge.

after mass transfer is complete, it is reasonable to assume that the jets were present when Cas A went supernova.

5.1 Future Work

Another goal is to better understand the modeling from *MESA* for the primary star. A substantial amount of time went into trying to model the large star at different stages of the red giant branch, a major advantage of the *MESA* code. Ultimately, the research was left to be continued by another student in the lab, as the stars were not relaxing in a satisfactory manner. The primary goal for this work was to create profiles to run supernova simulations through at Los Alamos National Laboratory under Dr. Chris Fryer. These data will be sent to the laboratory and continued through a summer project this year. Hopefully more light will be shed on the progenitor of Cas A and we will be able to determine definitively if there was a binary merger before the supernova explosion 500 years ago.

References

Dhillon, V. 2012, Lives of Binary Stars: Mass Transfer

Gaburov, E., Lombardi, Jr., J. C., & Portegies Zwart, S. 2010, MNRAS, 402, 105

Krause, O., Birkmann, S. M., Usuda, T., Hattori, T., Goto, M., Rieke, G. H., & Misselt, K. A. 2008, Science, 320, 1195

Monaghan, J. J. 1992, ARA&A, 30, 543

Moreno Méndez, E., López-Cámara, D., & De Colle, F. 2017, MNRAS, 470, 2929

Pejcha, O., Metzger, B. D., & Tomida, K. 2016, MNRAS, 461, 2527

Piran, T., Nakar, E., Mazzali, P., & Pian, E. 2017, ArXiv e-prints

Price, D. J. 2007, PASA, 24, 159

Rasio, F. A., & Shapiro, S. L. 1991, ApJ, 377, 559

Ryden, B., & Peterson, B. 2010

Schreier, R., & Soker, N. 2016, Research in Astronomy and Astrophysics, 16, 70

Schure, K. M., Vink, J., García-Segura, G., & Achterberg, A. 2008, ApJ, 686, 399

Appendices

A Example of *splot* Routine

A.1 pplot.f

When making a splot, you should create a file that asks you what scripts you want to use and what files you want to analyze. The following file is pplot.f which prompts the user for this information and calls the appropriate other routines. This is written in Fortran

```
      INCLUDE 'sphu.h'
      integer iout,ni1,ni2,ni3,nnit

C Type of output desired:

      WRITE (6,*) 'Select output type:  0= no output'
      WRITE (6,*) '                     1= ASCII out'
      WRITE (6,*) '                     2= grids for Cas A'
      WRITE (6,*) '                     71= trajectories&distributions'
      READ (5,*) IOUT

      WRITE (6,*) 'Enter Nbegin, Nfinal, dN:'
      READ (5,*) NI1,NI2,NI3
      IF(NI3.EQ.0) NI3=NI2-NI1
      IF(NI3.EQ.0) NI3=1

      if(iout.eq.2) call tabulinit

C     Main loop over binary output files:
      DO nnit=NI1,NI2,NI3
         CALL READIT(nnit)
         WRITE (6,*)
         WRITE (6,*) 'NOUT=',NOUT,' NIT=',NIT,' T=',T

         IF (IOUT.EQ.1) CALL ASCIIOUT
```

```
   IF (IOUT.EQ.2) CALL grids
   IF (IOUT.EQ.71) CALL trajANDdist

enddo
end
```

A.2 grids.f

The following is a file called grids.f. The purpose of the file is to integrate the ejected mass from the poles to create our velocity distribution plots and our histograms. This is written in Fortran

```
      subroutine grids
      include 'sphu.h'
      integer ir, icostheta, iphi, irmax, icosthetamax, iphimax,ip,box
      parameter(box = 101)
      parameter (irmax = 32*box, icosthetamax = box, iphimax = box)
      real rhortp(irmax,icosthetamax,iphimax)
      real vrrtp(irmax,icosthetamax,iphimax)
      real*8 zpos, ypos, xpos, r, r2, theta, phi,rmax,wpc,rip
      integer index,iripmin,iripmax,icosthetaipmin,icosthetaipmax
      integer iphiipmin,iphiipmax,iphistore
      real*8 thetaipmin,thetaipmax,deltatheta,thetaip,phiip,deltaphi
      real*8 costhetaipmin, costhetaipmax,rcip,phiipmin,phiipmax,mass
      real*8 stepcostheta,stepr,stepphi,vol

      REAL*8 am4
      real*8 X1,Y1,Z1,VX1,VY1,VZ1,am1
      real*8 X2,Y2,Z2,VX2,VY2,VZ2,am2
      real*8 X3,Y3,Z3,VX3,VY3,VZ3,am3
      integer icomp(nmax)
      integer ncomp1,ncomp2,ncomp3
      real*8 rhoavg, vravg
      real*8 speedunit,rhounit
      character*16 FNAME
      integer k01,k10,k50,k90,k99,nbini,k,nbins
      parameter(nbins=10001)
      real*8 v01,v10,v50,v90,v99,vmax,amv(nbins)


      COMMON/COMPBETTERCOM3/AM1,X1,Y1,Z1,VX1,VY1,VZ1,
     $     AM2,X2,Y2,Z2,VX2,VY2,VZ2,
     $     AM3,X3,Y3,Z3,VX3,VY3,VZ3,am4,
```

```
      $     ICOMP,ncomp1,ncomp2,ncomp3

       call COMPBEST3

       do ip=1,N
          x(ip)=x(ip) - x1
          y(ip)=y(ip) - y1
          z(ip)=z(ip) - z1
          vx(ip)=vx(ip) - vx1
          vy(ip)=vy(ip) - vy1
          vz(ip)=vz(ip) - vz1
       enddo

c      initializing the r theta phi matrix
       do iphi = 1,iphimax
          do icostheta = 1,icosthetamax
             do ir = 1,irmax
                rhortp(ir,icostheta,iphi)=0
                vrrtp(ir,icostheta,iphi)=0
             enddo
          enddo
       enddo
c      find rmax by looping over all particles
       rmax=0d0
       r2=0d0
       do ip=1,N
          r2=r2 + x(ip)**2+y(ip)**2+z(ip)**2
          rmax = max(rmax,sqrt(x(ip)**2+y(ip)**2+z(ip)**2))
       enddo

       print *, 'rmax = ', rmax,' so spacing is',rmax/irmax
       stepcostheta = 2/(icosthetamax-1d0)

       stepr = rmax/irmax

       stepphi = 2*pi/iphimax
```

```
      do ip=1,N
         if (u(ip).gt.0.d0 .and. icomp(ip).eq.4) then
            rip = sqrt(x(ip)**2 + y(ip)**2 + z(ip)**2)
            iripmin = max(int((rip-2*hp(ip))/stepr +1),1)
            iripmax = min(int((rip+2*hp(ip))/stepr),irmax)

            rcip = sqrt(x(ip)**2 + y(ip)**2) ! cylindrical radius or something like that


            phiip = atan2(y(ip),x(ip))

            thetaip = acos(z(ip)/rip)

************************** THETA *********************************
c     make the theta spanning more efficient
            if(rip.ge.2*hp(ip)) then
               deltatheta=asin(2*hp(ip)/rip)
               thetaipmax = thetaip + deltatheta
               thetaipmin = thetaip - deltatheta

!     Note: thetaipmax is a larger number than thetaipmin
!     Note: cos(thetaipmax) is a smaller number than cos(thetaipmin)

c     Set a value of icosthetaipmin (index corresponding to largest theta to consider):
               if(thetaipmax.ge.pi)then
                  costhetaipmax = -1
                  icosthetaipmin= 1
               else
                  costhetaipmax = cos(thetaipmax)
                  icosthetaipmin  = int(
     $                 (costhetaipmax+1)/stepcostheta+2d0)
               endif

c     Set a value of icosthetaipmax (index corresponding to smallest theta to consider):
               if(thetaipmin.le.0)then
                  costhetaipmin = 1
                  icosthetaipmax= icosthetamax
```

```
               else
                  costhetaipmin = cos(thetaipmin)
                  icosthetaipmax  = int(
     $                 (costhetaipmin+1)/stepcostheta+1d0)
               endif

            else
               icosthetaipmin = 1
               icosthetaipmax = icosthetamax
            endif

************************ PHI *********************************
c     this is to make spanning more efficient
            if(rcip.ge.2*hp(ip)) then
               deltaphi=asin(2*hp(ip)/rcip)
               phiipmax = phiip + deltaphi
               phiipmin = phiip - deltaphi
!     Note: phiipmax is a larger number than phiipmin
               iphiipmin  = int(phiipmin/stepphi+2+iphimax)
     $              -iphimax
               iphiipmax  = int(phiipmax/stepphi+1+iphimax)
     $              -iphimax
            else
               iphiipmin = 1
               iphiipmax = iphimax
            endif

            do iphi = iphiipmin,iphiipmax
               phi = (iphi-1)*stepphi

               do icostheta = icosthetaipmin,icosthetaipmax  ! icostheta=1 means theta=p
                  theta = acos((icostheta-1d0)*stepcostheta - 1d0)
                  do ir = iripmin,iripmax
                     r = ir*stepr
                     zPOS = r*cos(theta)
                     yPOS = r*sin(theta)*sin(phi)
                     xPOS = r*sin(theta)*cos(phi)
```

```
                r2 = (x(ip)-xpos)**2+(y(ip)-ypos)**2+
$                    (z(ip)-zpos)**2
                if(r2.lt.4*hp(ip)**2) then
                   index = int(ctab*r2/hp(ip)**2)+1
                   wpc = wtab(index)/hp(ip)**3
                   if(iphi.le.0)then
                      iphistore=iphi+iphimax
                   elseif(iphi.gt.iphimax)then
                      iphistore=iphi-iphimax
                   else
                      iphistore = iphi
                   endif

                   rhortp(ir,icostheta,iphistore) = rhortp(ir,
$                       icostheta,iphistore) + am(ip)*wpc
                   vrrtp(ir,icostheta,iphistore) =
$                       vrrtp(ir,icostheta,iphistore)
$                       + (am(ip)*wpc*(vx(ip)*x(ip)+vy(ip)*y(ip)
$                       + vz(ip)*z(ip)) / rip )
                endif
             enddo
          enddo
       enddo
    endif
 enddo

 PRINT *, 'EVERYTHING IS OKAY'
 do iphi = 1,iphimax
    do icostheta = 1, icosthetamax
       do ir = 1, irmax
          if (rhortp(ir,icostheta,iphi).gt.0) then
             vrrtp(ir,icostheta,iphi) = vrrtp(ir,icostheta,iphi)
$              / rhortp(ir,icostheta,iphi)
          endif
       enddo
    enddo
 enddo
```

```
      mass=0
      speedunit=(gravconst*munit/runit)**0.5d0*1.d-5
      rhounit=munit/runit**3d0

      print *,'fyi, speed and density units (in km/s and g/cm^3)=',
     $     speedunit,rhounit

      IF(NOUT.LE.99999) THEN
         WRITE (FNAME,1102) NOUT
      else
         WRITE (FNAME,1103) NOUT
      ENDIF
 1102 FORMAT ('rcrv',I5.5,'.sph')
 1103 FORMAT ('rcrv',I6.6,'.sph')
      OPEN(123, FILE=FNAME)
      write(123,'(2a16,9a17)')'Radius(Rsun)','CosTheta',
     $     'RhoAvg(g/cm^3)','VradAvg(km/s)'
      do ir=1,irmax
         r = ir*stepr
         do icostheta = 1,icosthetamax
            rhoavg = 0
            vravg = 0
            do iphi = 1,iphimax
               rhoavg = rhoavg + rhortp(ir,icostheta,iphi)
               vravg = vravg +
     $              rhortp(ir,icostheta,iphi)*vrrtp(ir,icostheta,iphi)
            enddo
            mass = mass + rhoavg*r**2
            if(rhoavg.ne.0) vravg=vravg/rhoavg
            rhoavg=rhoavg/iphimax
            write(123,'(g16.9,f16.9,9g17.9)') r,
     $           (icostheta-1d0)*stepcostheta - 1d0,
     $           rhoavg*rhounit, vravg*speedunit
         enddo
      enddo
      close(123)
```

```
      mass=mass*stepcostheta*stepphi*stepr

      print *,'mass = ', mass,am4
      write(1122,*)am4,t

c     WRITING A NEW OUTFILE FOR VELOCITY HISTOGRAM
      IF(NOUT.LE.99999) THEN
         WRITE (FNAME,1104) NOUT
      else
         WRITE (FNAME,1105) NOUT
      ENDIF
 1104 FORMAT ('vhist',I5.5,'.sph')
 1105 FORMAT ('vhist',I6.6,'.sph')
      OPEN(124, FILE=FNAME)

      vMAX=400d0/speedunit
      do icostheta = 1,icosthetamax
C     Initialize elements of array Amv.  The Kth element of this
C     array will be the amount, in the final configuration,
C     of mass enclosed by the sphere whose radius maps to Bin# K
c     according to
c     v = vMAX*(Bin# -1)/(NBINS-1)

         DO K=1,NBINS
            AMV(K)=0.d0 ! amv  amount of velocity in cosine bin
         ENDDO

         do ir=1,irmax
            r = ir*stepr
            rhoavg = 0
            vravg = 0
            do iphi = 1,iphimax
               rhoavg = rhoavg + rhortp(ir,icostheta,iphi)
               vravg = vravg +
     $              rhortp(ir,icostheta,iphi)*vrrtp(ir,icostheta,iphi)
            enddo
            vol = r**2*stepcostheta*stepphi*stepr
```

```
            if(rhoavg.ne.0) vravg=vravg/rhoavg
            rhoavg=rhoavg/iphimax

c     At this point all particle masses and positions have been assigned.
c     So we will construct an array to help keep track of enclosed mass
c     vs. radius r

            NBINI=nint(vravg*(NBINS-1.d0)/vMAX+1.d0)
            IF(NBINI.GT.NBINS .OR. NBINI.LT.1)then
               WRITE(69,*)'Choose a larger vmax'
               stop
            endif
            AMV(NBINI)=AMV(NBINI)+0.5d0*rhoavg*vol*iphimax
            DO K=NBINI+1,NBINS
               AMV(K)=AMV(K)+rhoavg*vol*iphimax
            ENDDO
         enddo
C     The array AMV is now defined.  The Kth element of this
C     array is the amount, in the final configuration,
C     of mass enclosed by the sphere of radius
C     v = vMAX*(K -1)/(NBINS-1)

c     Let's use the AMV array to get the median velocity:
         do k =1,nbins
            if(amv(k).lt.amv(nbins)*0.5d0) k50=k
            if(amv(k).lt.amv(nbins)*0.1d0) k10=k
            if(amv(k).lt.amv(nbins)*0.9d0) k90=k
            if(amv(k).lt.amv(nbins)*0.01d0) k01=k
            if(amv(k).lt.amv(nbins)*0.99d0) k99=k
         enddo
         v01 = vMAX*(K01 -1)/(NBINS-1)*speedunit
         v10 = vMAX*(K10 -1)/(NBINS-1)*speedunit
         v50 = vMAX*(K50 -1)/(NBINS-1)*speedunit
         v90 = vMAX*(K90 -1)/(NBINS-1)*speedunit
         v99 = vMAX*(K99 -1)/(NBINS-1)*speedunit
```

```
    write(124,*) (icostheta-1d0)*stepcostheta - 1d0, v01,v10,v50,
$        v90,v99

    write(69,*)
$        '       bin          velocity      mass below velocity'
    do k=1,nbins,nbins/10
       write(69,*) k, vMAX*(k -1)/(NBINS-1), amv(k)
    enddo

 enddo
 close(124)

 end
```

B Example of Super Mongo (SM) Macro

SuperMongo is a useful graphing package that is used to display all of the data above except the plots created in *splash*. It has its own language but is fairly straightforward in terms of how it works. One can type these directly into the SM window or create a macro with extension ".sm" and plot that way. At any time, one can type "help *command*" to learn the proper arguments for said command.

```
rcrv 1 #
era
expand 1.3
data "rcrv"$1".sph"
lines 2 0
read{r 1 costheta 2 rho 3 vr 4}
        ticksize 0 0 -1 10

set rtmp = r if(costheta==-1 && rho>0)
set rhotmp = lg(rho) if(costheta==-1 && rho>0)
limits rtmp rhotmp
box
ctype blue
connect rtmp rhotmp

set rtmp = r if(costheta==-0.5 && rho>0)
set rhotmp = lg(rho) if(costheta==-0.5 && rho>0)
ctype cyan
connect rtmp rhotmp

set rtmp = r if(costheta==0 && rho>0)
set rhotmp = lg(rho) if(costheta==0 && rho>0)
ctype green
connect rtmp rhotmp

set rtmp = r if(costheta==0.5 && rho>0)
set rhotmp = lg(rho) if(costheta==0.5 && rho>0)
```

```
ctype magenta
connect rtmp rhotmp

set rtmp = r if(costheta==1 && rho>0)
set rhotmp = lg(rho) if(costheta==1 && rho>0)
ctype red
connect rtmp rhotmp

ctype default
xlabel r
ylabel \rho
vrcrv 1 #
era
expand 1.3
data "rcrv"$1".sph"
lines 2 0
read{r 1 costheta 2 rho 3 vr 4}
        ticksize 0 0 -1 10

set rtmp = r if(costheta==-1 && vr>0)
set vrtmp = lg(vr) if(costheta==-1 && vr>0)
#limits rtmp vrtmp
limits 0 250000 0 2.5
#limits rtmp 0 2.5
box
ctype blue
connect rtmp vrtmp

set rtmp = r if(costheta==-0.5 && vr>0)
set vrtmp = lg(vr) if(costheta==-0.5 && vr>0)
ctype cyan
connect rtmp vrtmp

set rtmp = r if(costheta==0 && vr>0)
set vrtmp = lg(vr) if(costheta==0 && vr>0)
ctype green
connect rtmp vrtmp
```

```
set rtmp = r if(costheta==0.5 && vr>0)
set vrtmp = lg(vr) if(costheta==0.5 && vr>0)
ctype magenta
connect rtmp vrtmp

set rtmp = r if(costheta==1 && vr>0)
set vrtmp = lg(vr) if(costheta==1 && vr>0)
ctype red
connect rtmp vrtmp

relocate 150000 0.6
ctype blue
label cos\theta = -1
ctype cyan
relocate 150000 0.5
label cos\theta = -0.5
ctype green
relocate 150000 0.4
label cos\theta = 0
ctype magenta
relocate 150000 0.3
label cos\theta = 0.5
ctype red
relocate 150000 0.2
label cos\theta = 1
ctype default


ctype default
xlabel r
ylabel v_r

mt 0 #
    era
#location 0 1500 0 0
expand 1.3
```

```
data "fort.1122"
lines 2 0
read{mass 1 time 2} #The axes are read backwards
set time = (time*0.018445)/365.25
limits time mass
box
ltype 2
relocate 20 0
draw 20 1.5
relocate 30 0
draw 30 1.5
relocate 126 0
draw 126 1.5
ltype 0
connect time mass
ylabel mass lost [M_\odot]
xlabel time [years]

hist 1  #
      era
expand 1.3
data "vhist"$1".sph"
lines 1 0
read{costheta 1 v01 2 v10 3 v50 4 v90 5 v99 6}
limits -1.05 1.05 0 160
box
ltype 0
ctype blue
connect costheta v01
relocate 0.2 125
draw 0.3 125
relocate 0.39 123
ctype black
label 1% Mass

ltype 2
ctype blue
```

```
connect costheta v99
relocate 0.2 145
draw 0.3 145
relocate 0.35 143
ctype black
label 99% Mass

ltype 0
ctype red
connect costheta v10
relocate 0.2 130
draw 0.3 130
relocate 0.35 128
ctype black
label 10% Mass

ltype 2
ctype red
connect costheta v90
relocate 0.2 140
draw 0.3 140
relocate 0.35 138
ctype black
label 90% Mass

ltype 0
ctype black
connect costheta v50
relocate 0.2 135
draw 0.3 135
relocate 0.35 133
ctype black
label 50% Mass
ctype default
ylabel v_r [km/s]
xlabel cos\theta [z/r]
```

```
iteratev #
 do outnum=3000,10000,100 {
   dev postencap vrcrv_0$outnum.eps
   vrcrv $outnum
   expand 1.4
   toplabel t=$(int($outnum*100*1.8445e-2+0.5)) days
   hc
}
iteratehist #
    do outnum=10000,35000,1000 {
       dev postencap hist_0$outnum.eps
       hist $outnum
       expand 1.4
       toplabel t=$(int($outnum*100*1.8445e-2+0.5)) days
       hc
     }
```