Senior Thesis:

Astrophysical Blastwaves

Phillip Zukin – Caltech 2006

Mentors:

Dr. Marc Kamionkowski

Dr. Steve Furlanetto

Table of Contents

Introduction

Shock waves are known to significantly alter the medium through with they propagate. Shocks in the interstellar medium have been known to affect star formation. Likewise, shocks in the intergalactic medium are expected to affect galaxy formation and the confinement of intergalactic clouds. Because of their significant influence, it is important to have a sound understanding of the different solutions used to study shocks and see how they differ. In this paper, I will focus on the self similar solution, thoroughly developed by Ostriker and McKee, and the explosion model, developed by Tegmark, Silk, and Evrard.

Background

A shock resulting from an explosion undergoes four phases. In the first phase, the external medium can be neglected, as the ejected mass undergoes free expansion. Then, when the shock absorbs a mass equal to the initially ejected mass, the shock begins its adiabatic phase, where it can be assumed that radiative losses are negligible, and all of the shock's energy is conserved. At this point, the well known Taylor Sedov solution is applicable and the shock is said to be self–similar. Later, when radiative losses come to total the amount of the initially injected energy, the shock enters the snow plow phase, where momentum is conserved. Last, as the pressure in the interior of the shock drops to the pressure of the external medium, the shock merges with the outside medium.

This paper focuses mainly on the second phase of the supernova remnant. I will analyze the models in both a static and expanding universe. The static case applies when the shock remains in its galaxy. The expanding universe case applies when the shock reaches the IGM. In order for the shock to reach the IGM, it must either be infused with energy from multiple supernova bursts, with a total energy input estimated at around 10^{53} ergs, or the galaxy must be small, which was the case for early galaxies. In addition, I will use the models to analyze upper limits on how much energy can be inputted into the system so that the shock becomes weak on time scales smaller than the Hubble time. Last, I will give the solution to a simplified model that takes into account the first correction to a blastwave during its free expansion phase.

Self Similar Solution

A self similar flow means that the flow at any point and time looks the same as it did at a prior point and a prior time. In other words, while the flow evolves in real space, it is invariant once suitably scaled. This implies that any non-dimensional quantity constructed out of parameters describing the system must remain constant for all time. Hence, every quantity must have power law dependences on a specific parameter, the most convenient of which is the normalized shock radius.

Solution in a static universe:

The parameters used to describe the system in this case are the initial injected energy (E), the outside density (ρ_1), the radius of the shock (r), and time (t). Given these quantities, we can make a non-dimensional constant ξ as shown below, and then solve for the radius of the shock.

$$\xi = r\left(\frac{\rho_1}{Et^2}\right)^{\frac{1}{5}} \qquad 1.1.1$$

$$r_{sh}(t) = \xi_o\left(\frac{Et^2}{\rho_1}\right)^{\frac{1}{5}} \quad \text{where } \xi_0 = \text{numerically computed constant} \qquad 1.1.2$$

For this case, a single supernova is injecting all of the energy. However, if multiple supernovas are firing, the energy is additive, and when looking at the solution at time scales when the self similar solution is valid, it is as if a single supernova injected the sum total of all the multiple energy inputs. Note that ξ_0 is a specific value chosen to enforce energy conservation (see equation 1.1.12) Given this power law solution for the radius of the shock, it is important to get a feel for the magnitude of quantities we are dealing with and approximately when the solution is valid. Look to the table below for reference. Clearly, the velocity of the shock cannot exceed that of the initial ejected mass. For an initial energy input of 10^{51} ergs and an external density of $2 * 10^{-24}$ g cm^{-3}, assuming an ejection of 1 M$\odot$, we will have an initial velocity of 10^4 km s^{-1}. So we cannot expect the solution to be valid for times much smaller than 100yr. Now consider how much energy is radiated away. The cooling function has a value of $10^{-21} - 10^{-22}$ erg

cm^{-3} s^{-1} for a temperature of $2 * 10^5$ K. With these values, you'd expect by a time of 10^5 yr for about 10^{51} ergs to be radiated away, which implies that the solution will not be valid for times much larger than 10^5 yr. (For a derivation of the post shock temperature, shown in the table, look to appendix B. To see how the radiated energy was found, look to the Energy Analysis section)

t (yr)	1	10	100	1,000	10,000	100,0000
r_{sh} (pc)	.315	.791	1.99	4.99	12.5	31.5
U_{sh} (km s^{-1})	124,000	31,000	7,820	1,970	494	124
T_2 (kelvin)	2.08E11	1.3E10	8.3E8	5.3E7	3.3E6	2.1E5

Aside from the shock radius, it is also important to analyze the interior structure of the shock. In order to know how the pressure, velocity, and density in the interior of the shock act, we first need to know the values of these quantities directly after the shock. These quantities, labeled with a subscript 2, are called the post-shock values. They are determined by the Rankine – Hugoniot jump conditions, which come about by integrating the continuity equations over the shock. The post shock values, in terms of the external values, in the limit of a strong shock, are shown below.

$$\rho_2 = \left(\frac{\gamma+1}{\gamma-1}\right)\rho_1 \qquad u_2 = \frac{2}{\gamma+1}U_{sh} \qquad P_2 = \frac{2}{\gamma+1}\rho_1 U_{sh}^2 \tag{1.1.3}$$

Once these quantities are known, we can begin to determine the internal structure, by using the fluid conservation equations shown below (mass, momentum, and energy).

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho u) = 0 \tag{1.1.4}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} = -\frac{1}{\rho}\frac{\partial P}{\partial r} \tag{1.1.5}$$

$$\frac{\partial}{\partial t}\left[\rho\left(E + \frac{1}{2}u^2\right)\right] + \frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2 \rho u\left(E + \frac{P}{\rho} + \frac{1}{2}u^2\right)\right] = 0 \tag{1.1.6}$$

We first assume the hydrodynamic variables have the following dependences on ξ:

$$\rho(r,t) = \rho_2 \alpha(\xi) \qquad u(r,t) = u_2 \frac{\xi}{\xi_o} v(\xi) \qquad P(r,t) = P_2 \left(\frac{\xi}{\xi_o}\right)^2 p(\xi) \tag{1.1.7}$$

Plugging in these dependences into the fluid conservation equations, we are left with 3 coupled ordinary differential equations.

$$-\xi \frac{d\alpha}{d\xi} + \frac{2}{\gamma+1}\left[3\alpha v + \xi \frac{d}{d\xi}(\alpha v)\right] = 0 \tag{1.1.8}$$

$$-v - \frac{2}{5}\xi \frac{dv}{d\xi} + \frac{4}{5(\gamma+1)}\left(v^2 + v\xi \frac{dv}{d\xi}\right) = -\frac{2}{5}\left(\frac{\gamma-1}{\gamma+1}\right)\frac{1}{\alpha}\left(2p + \xi \frac{dp}{d\xi}\right) \tag{1.1.9}$$

$$-2(p+\alpha v^2) - \frac{2}{5}\xi \frac{d}{d\xi}(p+\alpha v^2) + \frac{4}{5(\gamma+1)}\left\{5v(\gamma p + \alpha v^2) + \xi \frac{d}{d\xi}[v(\gamma p + \alpha v^2)]\right\} = 0 \tag{1.1.10}$$

Note though, as mentioned before, that only one particular value of ξ_o is appropriate. In order to determine this value, we have a normalizing condition, that forces energy conservation.

$$\int_0^{r_{sh}(t)} \left[\frac{P}{\gamma-1} + \frac{1}{2}\rho u^2\right] 4\pi r^2 dr = E \tag{1.1.11}$$

The first term represents the thermal energy in the shock while the second represents the kinetic energy. Note that radiative losses such as ionizing the external medium and Compton cooling off the CMB are not taken into account. This equation, once the above dependences are substituted in, yields the following integral.

$$\frac{32\pi}{25(\gamma^2-1)} \int_0^{\xi_0} [p(\xi) + \alpha(\xi) v^2(\xi)] \xi^4 d\xi = 1 \tag{1.1.12}$$

Only for an appropriate value of ξ_0 will the integral be satisfied. In our case the value is approximately 1.17. It is important to notice that self similarity forces the thermal and kinetic energy to always remain in the same ratio. Numerically integrating the above

equations (1.1.8 – 1.1.10), we get the following graphs for the internal structure of a blastwave in a static medium.

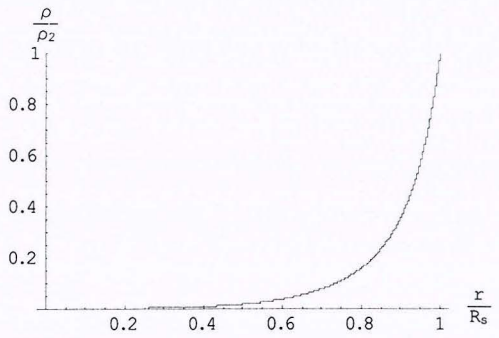

Figure 1.1.1

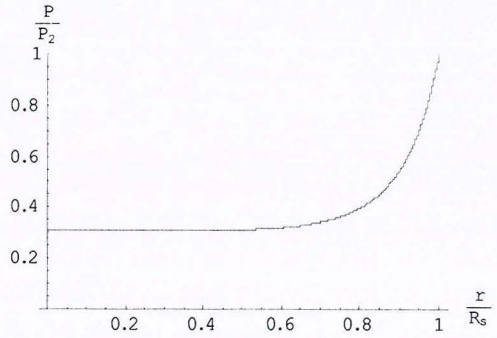

Figure 1.1.2

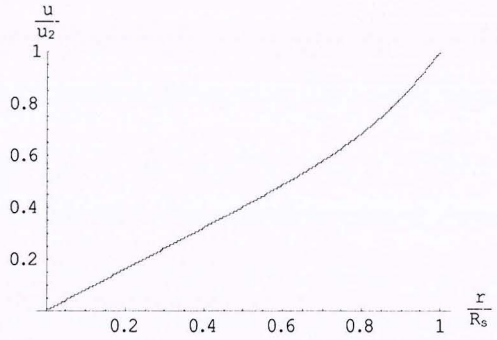

Figure 1.1.3

Figures: 1.1.1 normalized density profile, 1.1.2 normalized pressure profile, 1.1.3 normalized velocity profile

Notice that the density profile is biased towards the shock radius. Since radiative cooling has an n^2 dependence, it is clear that radiative effects will be most prevalent towards the shock radius. As a result the particles at a smaller normalized radius will catch up with the mass in front of them, and as time progresses, a shell structure will form. Also note that both the velocity and density go to zero at the shock center, while the pressure remains nonzero, signifying a very hot inner core. A plot of the temperature of the blastwave, as a function of normalized shock radius, is shown below.

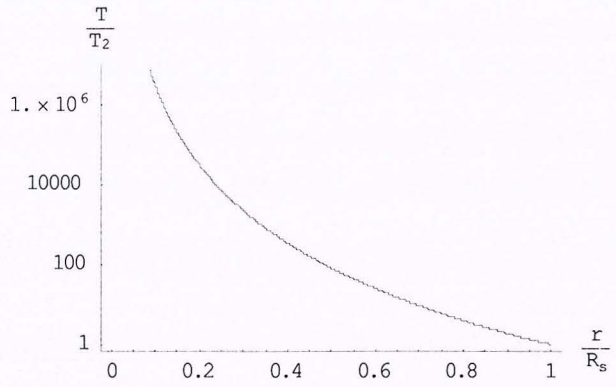

Figure 1.1.4 – normalized internal temperature.

Solution in an expanding universe

In an expanding universe, the density is time dependent. As a result, the time dependence of the shock wave changes. We are interested in analyzing a matter dominated universe with $\Omega = 1$, where Ω is the ratio of the energy density to the critical density today. Given this type of universe, the scaling of the density is shown below.

$$\rho_0 = \rho_u \tau^{-3h} \quad \text{where } h = \frac{2}{3} \text{ for } \Omega = 1, \ \tau \text{ is the cosmic time}$$
$$\text{and } \rho_u \text{ is the density of the external medium when the shock begins.} \qquad 1.2.1$$

In addition, it is possible to make the solution more general, given a new formalism for expressing different parameters in terms of power law dependences on a normalized shock radius. The formalism is shown below.

$$x = x(1) R^{-k_x} \qquad 1.2.2$$

Where x is any hydrodynamic variable, x(1) represents the post shock value of the variable, and R is a normalized shock radius. This new formalism allows for the possibility of introducing energy decay and other complications. The new time dependence of the shock wave, assuming an expanding universe and no energy decay, is expressed below.

$$R_S = \xi \left(\frac{E}{\rho_u} \right)^{\frac{1}{5}} \tau^{(2+3h)/5} \qquad 1.2.3$$

Since for $\Omega = 1$, $h = 2/3$, the exponent of time is 4/5. It is intuitive that the exponent for the shock should be greater in a moving medium. Primarily, since the density of the outside medium is decreasing with time, there is less frictional force stopping the shock from propagating. In addition, the shock picks up kinetic energy from the Hubble flow.

Note that because of self similarity, the ratio of the shock velocity to the Hubble velocity must be constant for all time. This ratio, which depends on the cosmology, is shown below.

for $\Omega = 1$

$$\nu_H = \frac{v_H}{v_S} = \frac{5 * h}{2 + 3h} = \frac{5}{6} \qquad 1.2.4$$

Now given these expressions for the shock radius, we can look for the internal structure of the blastwave. Once again, we look to the Rankine – Hugoniot jump conditions to establish a post shock value. Because of the velocity of the external medium, the post shock values have a different form. In the limit of a strong shock, they are shown below.

$$\rho_2 = \left(\frac{\gamma + 1}{\gamma - 1}\right) \rho_1 \qquad u_2 = \frac{2 + (\gamma - 1)\,\nu_H}{\gamma + 1} U_{sh} \qquad P_2 = \frac{2}{\gamma + 1} (1 - \nu_H)^2 \rho_1 U_{sh}^2 \qquad 1.2.5$$

Given our new formalism, it is more convenient to 1) use the entropy equation instead of the energy equation and 2) express the coupled differential equations in terms of logarithmic derivatives. The definitions used in the differential equations are shown below.

$$x = x_1(t)\,\tilde{x}(\lambda) \qquad \text{where } \lambda = \frac{r}{R_S} \qquad 1.2.6$$

$$x^* = \frac{\partial \ln x}{\partial \ln r} = \frac{d \ln \tilde{x}}{d \ln \lambda} \qquad 1.2.7$$

Note that x once again represents any hydrodynamic variable, and x_1 represents the post shock value. The differential equations, expressed in terms of the above definitions then become:

$$\nu v^* = \rho^* (1-\nu) + k_\rho - 2\nu \qquad 1.2.8$$

$$(1-\nu)\, v^* = \frac{\theta P^*}{\nu} + \frac{1}{2}(k_\rho - k_P) + \frac{\Omega\, \nu_h{}^2\, \tilde{g}}{2\nu^2 \lambda^2} \qquad 1.2.9$$

$$(1-\nu)\, P^* = \gamma (1-\nu)\, \rho^* + \gamma k_\rho - k_P \qquad 1.2.10$$

$$\text{where } \theta = \frac{P}{\rho\, v_\lambda{}^2} \qquad \nu = \frac{v}{v_\lambda} \qquad v_\lambda = \lambda v_s \qquad \nu_H = \frac{v_H}{v_s}$$

Note that the coupled differential equations all depend on gravity. But the gravity term is dependent on the density distribution inside the shock. Hence, it is best to assume a certain density profile, compute the internal distributions, and then recompute the profiles based on a gravity term that coincides with the previously computed density distribution. I iterated this approach until I reached a steady state solution. Plots of the internal distributions for an expanding universe are shown below.

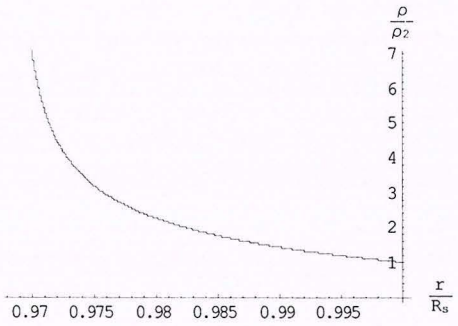

Figure 1.2.1

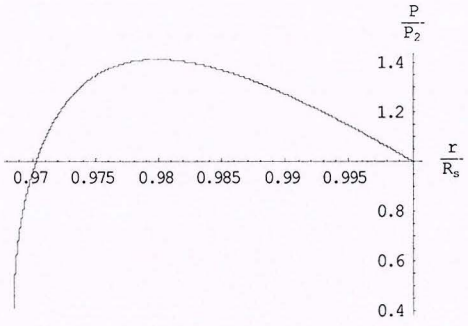

Figure 1.2.2

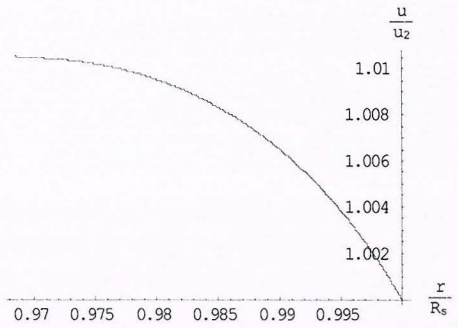

Figure 1.2.3

Figures: 1.2.1 normalized density profile, 1.2.2 normalized pressure profile, 1.2.3 normalized velocity profile

Notice that all the material is concentrated between $\lambda = .965$ and $\lambda = 1$. In addition, notice that the pressure approaches zero at $\lambda = .965$. If this were not the case, then the nonzero pressure would push the material in all directions, and the shell structure would not exist. Moreover, the steadily decreasing velocity profile implies that the velocity of objects at smaller λ will catch up the mass in front of it, further contributing to the shell structure.

Explosion Model

Since it takes a while for the self similar solution to become valid, it seems unnatural to use the energy of the initial injected energy in the formula, because of losses during the free expansion phase. As a result, Tegmark, Evrard, and Silk developed a different solution that models a blastwave from the time of the explosion to the time the shock merges into the IGM.

This model has 3 assumptions about the blastwave. 1) It assumes that most of the mass in the shock is concentrated in a thin spherical shell centered at the shock radius. 2) It assumes a neutral, uniform, pressure-less IGM. 3) It assumes that the internal plasma is thin and has a constant pressure and temperature. Given these assumptions, they derive a force formula which characterizes the system.

$$\frac{d^2 R}{dt^2} = \frac{8 \pi p G}{\Omega_b H^2 R} - \frac{3}{R}\left(\frac{dR}{dt} - HR\right)^2 - \left(\Omega_d + \frac{1}{2}\Omega_b\right)\frac{H^2 R}{2} \qquad 2.1$$

where p is pressure, H is the Hubble constant, Ω_b is the density of baryonic matter today divided by the critical density, and Ω_d is the ratio of dark matter today divided by the critical density. The first term in the equation represents the pressure of the internal plasma driving the shock outward. The second term is a braking force, which describes the necessary force needed to increase the velocity of the outside material from the Hubble velocity to the shock velocity, as it is picked up. The last term is the gravitational force. Note that the gravitational term resulting from the baryonic matter is half that of the dark matter. The ½ factor comes from the self-gravity of a thin shell. Also, note that the dark matter does not interact with the shock. This model assumes an $\Omega = 1$ matter dominated universe, only composed of baryonic and dark matter. This is a valid

assumption for redshifts smaller than 3000, when the matter and radiation energy density were equal, till recent redshifts, when the cosmological constant has taken over.

In addition to an equation for the radius, they also use an equation that describes how the internal thermal energy evolves with time.

$$E_t = \frac{3}{2} pV = 2\pi pR^3 \qquad 2.2$$

$$\frac{dE_t}{dt} = L - p\frac{dV}{dt} = L - 4\pi pR^2 \frac{dR}{dt} \qquad 2.3$$

$$L = L_{sn} - L_{comp} - L_{brems} - L_{ion} + L_{diss} \qquad 2.4$$

$$L_{sn} = 1.2\, L_\odot \frac{M_b}{M_\odot}\, \theta\,(t_{burn} - t) \qquad L_{diss} = f_d \frac{3m}{2R}\left(\frac{dR}{dt} - HR\right)^3 \qquad 2.5$$

where L is the sum of all radiative losses. Since the internal plasma is assumed to have constant pressure, the internal energy is straight forward to compute. A change in internal energy occurs because of radiative losses and the work necessary to expand the shock. The only terms we focused on for the radiative effects were L_{sn} and L_{diss}. L_{sn} represents a sourcing term that occurs while multiple supernovas are going off. Notice that is has the form of a constant energy injection into the shock for times smaller than t_{burn} ($\sim 5 * 10^7$ years). L_{diss} occurs because we assume the collision of the shock with the external medium is inelastic. As a result, some of the excess energy, after interacting with the shell, leaks inside to reheat the interior plasma. However, the amount of energy that does this is tough to gauge so f_d is introduced to account for the complicated microphysics. The parameter f_d is very important, and we will vary its value in our analysis.

These two equations give us two coupled differential equations. Moreover, normalizing the variables in the context of a matter dominated $\Omega = 1$ universe, we get the following equations, which can be numerically integrated.

$$r''(\tau) = \frac{18\pi}{\Omega_b}\eta(\tau)^{-2}\frac{q(\tau)}{r(\tau)} - 3\left[1 - \frac{2}{3}\frac{\eta(\tau)\, r(\tau)}{r'(\tau)}\right]^2 \frac{r'(\tau)^2}{r(\tau)}$$

$$-\left(\frac{2}{9}\Omega_d + \frac{1}{9}\Omega_b\right)\eta(\tau)^2 r(\tau) \qquad 2.6$$

$$q'(\tau) = \frac{l(\tau)}{2\pi r(\tau)^3} - 5\frac{r'(\tau)}{r(\tau)} q(\tau) \qquad 2.7$$

$$\text{where } \tau = \frac{t}{t_*} - 1 \text{ and } t_* = \text{hubble time.}$$

q, l, and r are all the normalized versions of p, L, and R respectively. η is the ratio of the Hubble constant at τ to the Hubble constant when the shock begins expanding. Note that the Hubble time t_* is a function of redshift, and is the time at which the shock begins expanding. Just to get a feel for the order of magnitudes we are dealing with, below shows a plot of the shock radius as a function of time. The upper line is for $f_d = 1$, while the lower is for $f_d = 0$.

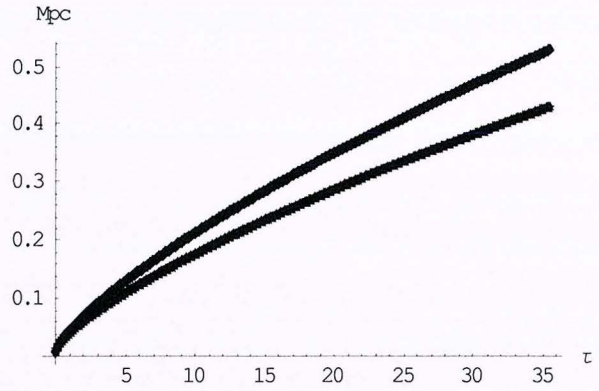

Figure 2.1 – radius of the shock as a function of normalized time.

Comparing the Models

We have chosen several possible ways to compare the two models. First is the time dependence of the solution. Given the clear time dependence in the self similar solution, we can look at how the explosion model scales with time at late times in the solution. We have also decided to test the accuracy of the explosion model's assumptions by comparing its internal structure to the internal structure of the self similar solution. Then, given these distributions, we want to compare the energy distributions of both solutions.

Comparisons Static Universe

It would be best to first compare the models, in a simpler context, by neglecting the Hubble flow and gravity. In this case, the explosion model's equations of motion reduce dramatically and are shown below.

$$r''(\tau) = 4\pi \frac{q(\tau)}{r(\tau)} - 3\frac{r'(\tau)^2}{r(\tau)} \qquad 3.1.1$$

$$q'(\tau) = \frac{l(\tau)}{2\pi r(\tau)^3} - 5\frac{r'(\tau)}{r(\tau)} q(\tau) \qquad 3.1.2$$

Note that the explosion model, even neglecting radiative losses, does not conserve energy because of the losses due to the work in expanding the shock. Since the self similar solution conserves energy, it is most natural to first force energy conservation in the explosion model, and hope the time dependence matches at late times. The new equation for the loss in thermal energy is shown below.

$$E = \frac{3}{2} pV \sim qr^3 = \text{const} \rightarrow q' r^3 + 3qr^2 r' = 0 \qquad 3.1.3$$

$$\rightarrow q' = -3\frac{r'}{r} q \qquad 3.1.4$$

With these new equations, numerically we found that the solution scales as 2/5, which matches exactly with the self similar solution.

Next we looked at the scaling of the explosion model using its original equations of motion. This time however, we were curious to find for what value of f_d the solution looked like the self similar solution. Below is a chart of the exponent of time for various values of f_d at late times in the solution. Note that for this simplified case, there is no appropriate time to check the time dependence of the solution, but it was clear that the time dependence of all solutions were asymptotic. As a result, we checked the time exponent at late times.

f_d	1	.8	.6	.4	.2	0
Exp of time	.4004	.3812	.3616	.3410	.3184	.2901

We can see from the chart that $f_d = 1$ best approximates the self similar solution. Analytically, I was also able to verify the exponents for $f_d = 0$ and $f_d = 1$. This was a good check that the program was working appropriately. The analysis is shown below.

$$f_d = 0 \rightarrow q' = -5\frac{r'}{r}q \rightarrow q \sim r^{-5}$$

$$\rightarrow r'' \sim t^{n-2} \sim \frac{q}{r} \sim r^{-6} \sim t^{-6n} \rightarrow n = \frac{2}{7} \qquad 3.1.5$$

$$f_d = 1 \rightarrow q' = -3\frac{r'}{r}q \rightarrow q \sim r^{-3}$$

$$\rightarrow r'' \sim t^{n-2} \sim \frac{q}{r} \sim r^{-4} \sim t^{-4} \rightarrow n = \frac{2}{5} \qquad 3.1.6$$

It was surprising that, in the case of $f_d = 1$, a dissipative system acts like a system that conserves energy. In order to get a better understanding of how the system is behaving, we looked at the thermal and kinetic energy distributions as a function of time. They are shown below.

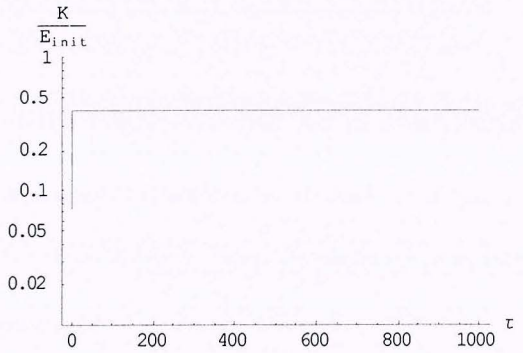

Figure 3.1.1 – Kinetic energy vs time.

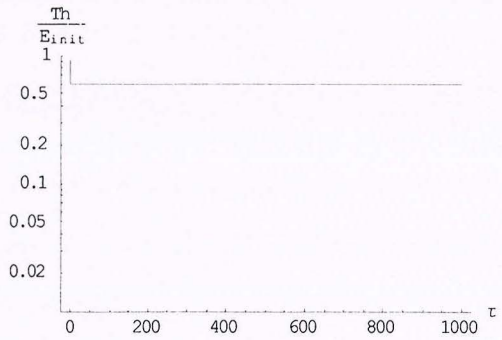

Figure 3.1.2 – Thermal energy vs time.

Notice that after the first bump, the system conserves its energy. Looking at equation (3.1.6), which is valid at late times, this should not be considered a coincidence, since for this case $q \sim r^{-3}$. Hence, the total thermal energy, which goes as qr^3, should be constant.

Another way of understanding the conservation of energy is that the only energy lost by the system, which results from the inelastic collisions, is regained by the system as thermal energy. In addition, note that since the self similar solution conserves energy, the time it takes for this particular case to stabilize may be a good indication for how long it takes an explosion to become self similar.

Next in the analysis is verifying the assumptions the explosion model made about the internal structure of the shock. Look to figures 1.1.1 – 1.1.3, for reference of the internal structure of the shock in the self similar solution. A plot of the explosion model's internal structure is not shown because it is not instructive. It consists of all the mass concentrated in a thin shell at the shock radius, with uniform pressure inside.

Note that the self similar solution does not include certain microphysics taken into account by the explosion model. In the explosion model, it is assumed that a small amount of the ionized medium leaks from the shell into the interior, providing the internal pressure that drives the shock. Moreover, they do not consider the thermal energy of the shell itself. The self similar solution, on the other hand, does not take into account the ionized medium produces by the shock, but does compute the pressure associated with the mass inside the shock. Hence, the only characteristic we can compare between the two internal structures is the density distribution. Looking at the density profile, for the self similar case, we see that the mass is concentrated towards the shock radius. But, it is clear that all of the mass is not concentrated in a thin spherical shell. Moreover, assuming the pressure for the self similar case roughly corresponds to the pressure of the ionized medium, we can see that it is approximately constant for r/R smaller than .6. Hence, their seems to be poor agreement between the internal structure of the Explosion Model and the internal structure of the self similar solution in the case of a shock in a static medium.

Aside from comparing internal structures, we can also compare energy distributions. We could ask again for what value of f_d will the energy distributions in the explosion model be equal to that of the self similar solution. Note that for a self similar solution, the shock contains 70% thermal energy and 30% kinetic energy. The table below shows the

computation done for several values of f_d, where the first pair of energies are scaled by the initial energy inputted into the shock, and the last pair is scaled by the final energy.

f_d	1	.8	.6	.4	.2	0
T / E_{init}	.6	.26	.110	.0433	.015	.0035
K / E_{init}	.4	.19	.088	.0407	.0176	.0067
T / E_{fin}	.6	.577	.556	.515	.460	.343
K / E_{fin}	.4	.423	.444	.485	.540	.657

We can see that no value of f_d will match the energy distributions in the self similar solution. Moreover, the energy in the system is only conserved for $f_d = 1$. No matter how high f_d is, the model will never be able to convert enough collisional energy into thermal energy that reheats the interior of the shock to the self similar value. This implies that there may be another mechanism for transferring heat into the interior that the explosion model has not taken into account. Or it's possible, as we saw before, that the thin shell approximation is not valid. It may be the case that the majority of the mass is not moving at the shock velocity, but some velocity slightly lower. As a result, the kinetic energy would be smaller and the energy distributions could match the self similar case.

Comparisons Expanding Universe

Next, we moved on to a more general problem, taking into account the expansion of the universe.

As before, we first compare time scalings for the modified explosion model, which conserves energy (see equation 3.1.4). In this case, just as before, we get a match of .8. Next, we analyzed how the time scales for the original explosion model. Below is a table showing the computed time dependences for various values of f_d. Note though that for the case of the expanding universe, there is a specific time, where the solution's time exponent can be tested. Since we know, from before, that for a cosmological blatwave, the ratio of the shock velocity to the Hubble velocity is a specified value, we can check the exponent of time at a point in the evolution of the blastwave where its velocity is 6/5

of the Hubble velocity. Since it's expected that all solutions to the explosion model will eventually match this criteria, given that the velocity of the shock begins at a velocity higher than this value and decays to the Hubble velocity monotonically, I included the time at which this criteria is met in the table below.

f_d	1	.8	.6	.4	.2	0
Exp of t	.7999	.7999	.7999	.7999	.7999	.7999
Time	7.87	7.27	6.71	6.18	5.67	5.18

First we can see that the exponent of time for all values of f_d matches the self similar solution exactly. This however, should be expected, because we are measuring this value at a time when the explosion model is approximately self similar. Note that the solution does not stay at this exponent, and hence does not remain self similar, for all times after this period. Plots of the scaling of the shock radius as a function of time are shown below, for multiple values of f_d. We can see that the solutions begin to asymptote to the same value at late times, which is expected as the shock begins to merge with the IGM. Also, we can gauge how long the solution is approximately self similar by measuring when the scaling is between some ϵ of .8.

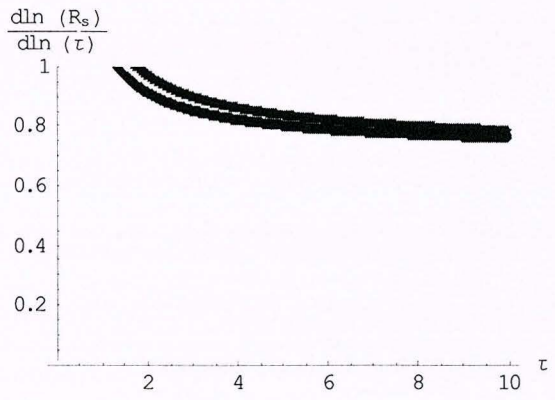

Figure 3.2.1 – dln(R) / dlnτ (the exponent of time). Line above is for $f_d = 1$. Line below is for $f_d = 0$.

What's more interesting is the time it takes for the explosion model to become approximately self similar, as a function of f_d. We can see, as we expect, that it takes longer for the solution to become self similar for higher values of f_d because, there is more internal pressure driving the shock at speeds too high to be self similar. Moreover,

the relationship between this time and f_d seems linear, as the following plot shows. This is surprising, in that the physics of is f_d complicated, as discussed before, and you would not expect it to have a simple linear effect.

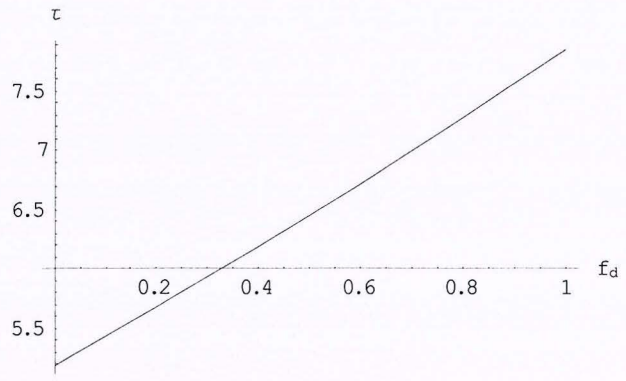

Figure 3.2.2 - τ vs f_d.

Next, we compared the internal structure of the blastwave computed in the self similar solution to the assumptions of the explosion model. Look to figures 1.2.1 – 1.2.3 for the computed internal structure of the self similar solution.

As discusses before, both models take into account thermal pressure from different sources. The explosion model has its thermal energy sourced by the ionized plasma, while the self similar solution's thermal energy comes from all material inside the shock. Because of this discrepancy, it is tough to gauge the difference in the internal structures of both models. However, we can compare density distributions, and we see that the shell approximation is valid in an expanding universe. Notice however that the shell is not uniformly distributed, and does not travel at the shock velocity, as assumed in the explosion model.

Now we look to the energy distributions of each model. Below is a table listing the energy distributions for different values of f_d. The first set is normalized by the initial energy input, while the second is normalized by the energy at the time of measurement. Note that once again I chose to compare the distributions when the shock velocity was 6/5 of the Hubble velocity. Also note that a self similar blastwave has 68.5% kinetic energy and 31.5 % thermal energy.

f_d	1	.8	.6	.4	.2	0
K / E_{init}	.9329	.7753	.6474	.5430	.4571	.3861
T / E_{init}	.0338	.0257	.0192	.0140	.0097	.0062
K / E_{fin}	.9650	.9679	.9712	.9749	.9792	.9842
T / E_{fin}	.0350	.0321	.0282	.0251	.0208	.0158

We can see that no matter what value of f_d, the energy distributions do no match the self similar case. Note that in an expanding medium, for the self similar solution, the kinetic energy dominates while in a static medium, the thermal energy dominates. This is to be expected because the shock absorbs the kinetic energy of the Hubble flow. Also, it's expected that the energy distributions above will not match that of the self similar solution because, just as in the static case, the model cannot account for enough thermal energy.

Energy Analysis

Next we wanted to find out on what time scales the shock becomes weak. A shock will be considered weak when it merges with an already ionized IGM, which is predicted to have a temperature of about 10000K. Setting the mean thermal velocity equal to that of a gas of protons at this temperature, we get an approximate velocity for when the shock becomes weak.

$$v_f = \sqrt{\frac{3\,kT}{m}} \sim 1.9 * 10^6 \frac{cm}{s} \qquad 4.1$$

Setting this velocity equal to the velocity of a self similar shock in a static medium, we get a relation for the time at which the shock becomes negligible.

$$v_s = \frac{2}{5}\,\xi_o \left(\frac{E}{\rho_0}\right)^{\frac{1}{5}} t^{-\frac{3}{5}} = v_f \rightarrow t_{weak} = \left(\frac{2}{5}\,\frac{\xi_o}{v_f}\left(\frac{E}{\rho_o}\right)^{\frac{1}{5}}\right)^{\frac{5}{3}} \qquad 4.2$$

Note that this time is a function of redshift because the external density is a function of redshift. In this approximation though, we are not taking into account the time dependence of the external density, as the shock propagates. We are assuming that the external density at the time of the explosion is approximately correct throughout the shock's propagation on the time scales we are interested in. Adjusting for the redshift dependence, we get the following.

$$t_{weak} = \left(\frac{2}{5} \frac{\xi_o}{v_f} \left(\frac{E}{\Omega_b \rho_{cr}} \right)^{\frac{1}{5}} \right)^{\frac{5}{3}} \frac{1}{1+z_*} \sim 1.34 * 10^{16}\, s \left(\frac{E}{E_{51}} \right)^{\frac{1}{3}} \frac{1}{1+z_*} \qquad 4.3$$

where z_* = redshift when shock began expanding

ρ_{cr} = critical density today

Ω_b = ratio of baryonic density to critical density

Next, we want to compare this time to the hubble time when the shock began. This can be easily solved for, assuming a matter dominated universe with $\Omega = 1$.

$$t_H = \frac{2}{3 H_0} \left(\frac{1}{1+z_*} \right)^{\frac{3}{2}} \sim 2.90 * 10^{17}\, s \left(\frac{1}{1+z_*} \right)^{\frac{3}{2}} \qquad 4.4$$

where H_0 = hubble constant today

Setting these two expressions equal to each other, we get an upper limit on how much energy can by initially injected such that the hubble flow can be neglected.

$$E < 10^{55}\, ergs \left(\frac{1}{1+z_*} \right)^{\frac{3}{2}} \qquad 4.5$$

Note that most supernova explosions are on the order of 10^{51} ergs. Only when multiple supernovas are firing at the same time, as is assumed in the explosion model, will the injected energy reach a value of 10^{53} ergs. Therefore we can see from the above limit that for redshifts of about 15, when shocks become important for their influence in ionizing the IGM, we know that the shock will become weak before a Hubble time elapses, hence the Hubble flow can be neglected, as is shown in the plot below.

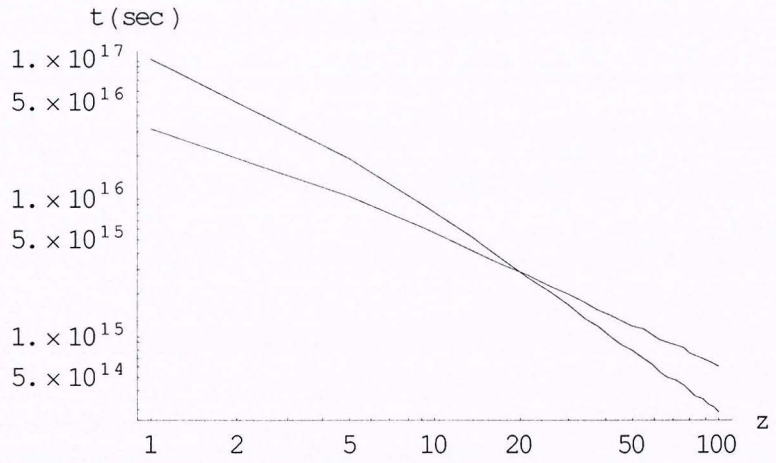

Figure 4.1: Plot of t_H and t_{weak} as a function of redshift for $E = 10^{53}$ ergs. (t_H is plot above for low z). We can see that for low redshifts, t_H is large compared to t_{weak}. For high redshift past the crossing point, the Hubble flow cannot be neglected, while for lower redshifts before the crossing point, it can.

Given the upper limit, it is important to verify the limit's validity. First we can ask whether the solution is approximately self similar on these time scales. To address these issues, we need to determine approximately when the solution becomes self similar, and when, because of radiative losses, the solution is not self similar.

The solution becomes self similar on time scales when the shock absorbs an amount equal to the initially ejected mass.

$$\frac{4}{3}\pi\rho_o R_s^3 = M_0 \rightarrow \frac{4}{3}\pi\rho_o\left(\xi_o\left(\frac{E}{\rho_0}\right)^{\frac{1}{5}} t^{\frac{2}{5}}\right)^3 = M_o$$

$$\rightarrow t = \left(\frac{3M_o}{4\pi\rho_o^{\frac{2}{5}}\xi_0^3 E^{\frac{3}{5}}}\right)^{\frac{5}{6}} \sim 2.83 * 10^{11}\, s\left(\frac{1}{1+z_*}\right)^{\frac{1}{3}}\left(\frac{M}{M_\odot}\right)^{\frac{5}{6}}\left(\frac{E_{51}}{E}\right)^{\frac{1}{2}} \quad 4.6$$

Given this computation, and looking to the figure above, we can see that the time scales we are interested in are well past the first phase of a SNR.

Now we look at radiative losses. The energy dissipated goes as:

$$E \sim Ln^2 Vt \quad 4.7$$

$$\text{where } n \sim \frac{8\,\Omega_b \rho_{cr}}{m}\,(1+z_*)^3 \qquad 4.8$$

where L is the cooling function, n is the number density, V is the volume of the shock, and t is time. The expression for n is the external baryonic density times a factor of 8. The extra factor comes from the Rankine Hugoniot jump conditions which states that, for a gas with γ= 5/3, the post shock density is 8 times the external density. Moreover, since most of the cooling occurs near the shock, it is appropriate to use this value. At a temperature of 10000K, the cooling function is at approximately 10^{-24} erg cm^3 s^{-1}. Hence we can show that radiative losses become severe at a time shown below.

$$L * n^2 * t_{cool} * \frac{4}{3}\,\pi \left(\xi_o \left(\frac{E}{\rho_0}\right)^{\frac{1}{5}} t^{\frac{2}{5}}\right)^3 \sim E$$

$$\rightarrow t_{cool} \sim 7.87 * 10^{22} \left(\frac{E}{E_{51}}\right)^{\frac{2}{11}} (1+z_*)^{\frac{-63}{11}} \qquad 4.9$$

For a redshift of 20, with an energy of 10^{51} ergs, $t_{cool} \sim 2.1 * 10^{15}$ s. Looking to equation 4.3, we see $t_{weak} \sim 6.4 * 10^{14}$ s. Therefore, for times that we are interested in, the shock remains self similar, and the above limit is valid.

We can also form a lower limit on the energy necessary to produce a shock wave, as a function of redshift. In order to produce a shock, the ejected material must have a speed greater than the sound speed. Note that the sound speed is dependent on the temperature of the medium, which approximately scales with redshift as (1+z).

$$v_s = \sqrt{\frac{\gamma kT}{m}} \rightarrow E \sim \frac{1}{2} M_{eject} v^2 \rightarrow E > 7.5 * 10^{41} (1+z_*) \text{ ergs } \frac{M}{M_\odot} \qquad 4.10$$

This limit is easily met by most explosions at a redshift of 20.

First Phase of the Remnant

I was also hoping to have a solution for the blastwave during its initial free expansion phase. However, instead of having a simple free expansion, I wanted to take into account a first order correction term. It seemed logical to assume a frictional term dependent on the cross section of the particles of the outside medium, as well as the density of the external medium. By dimensional analysis, we get the following force equation.

$$mx'' = -\rho\sigma v^2 \tag{5.1}$$

Note however that the first phase of the supernova remnant occurs on time scales small relative to the expanding universe. As a result, I will neglect the time dependence of the external density. Integrating the above equation, we get:

$$v(t) = v_0\left(1 + \frac{\sigma\rho v_0}{m}\,t\right)^{-1} \tag{5.2}$$

The coefficient in front of t is particularly interesting: ρ/m is the number density of the outside medium, which implies $m/\sigma\rho$ is the mean free path length, which makes $m/\sigma\rho v_0$ the time it takes to travel the mean free path length. The above equation can be integrated again to get the radius of the shock.

$$x(t) = \frac{m}{\sigma\rho}\ln\left(1 + \frac{\sigma\rho v_0}{m}\,t\right) \tag{5.3}$$

Notice that in the limit of $\sigma\rho v_0 t/m << 1$, the solution reduces to free expansion, as you'd expect. Previously I mentioned that the free expansion phase breaks down when the shock accumulates a mass equal to the ejected mass. I'd like to verify this claim based on the above model.

$$\frac{4}{3}\pi\rho x^3 = M_{ab} \rightarrow x = \left(\frac{3M_{ab}}{4\pi\rho}\right)^{\frac{1}{3}} \rightarrow t_* = \frac{m}{\sigma\rho v_0}\left\{\exp\left(\frac{\sigma\rho}{m}\left(\frac{3M_{ab}}{4\pi\rho}\right)^{\frac{1}{3}}\right) - 1\right\} \tag{5.4}$$

where M_{ab} is the absorbed mass at t_*. Now, plugging in this time to the velocity, we get:

$$v(t_*) = v_0 \exp\left\{-\frac{\sigma\rho}{m}\left(\frac{3M_{ab}}{4\pi\rho}\right)^{\frac{1}{3}}\right\} \qquad 5.5$$

Clearly, the free expansion assumption is not valid anymore when the exponent is of order 1, at which time the velocity has decreased by more than a factor of 2. Therefore we have:

$$\frac{\sigma\rho}{m}\left(\frac{3M_{ab}}{4\pi\rho}\right)^{\frac{1}{3}} \sim 1 \rightarrow M_{ab} \sim \frac{4\pi}{3}\left(\frac{m}{\sigma\rho}\right)^3 \rho = \frac{4\pi}{3} l^3 \rho \qquad 5.6$$

where l is the mean free path length previously discussed. Note that in a mean free path length, the shock on average has no time to interact, let alone, absorb any mass. Hence, the quantity we see above is approximately all of the initially ejected mass. The above model therefore, correctly predicts when the solution transitions out of the free expansion phase.

Conclusion

The above analysis compared the self similar solution to the explosion model in both a static and expanding medium. Doing this allowed us to understand the benefits of both models. While the self similar solution is approximately exact during the second phase of a supernova remnant, it does not take into account more complicated processes, like ionizing the IGM. The explosion model, on the other hand, which accounts for different cooling effects, is useful in that it models a blastwave throughout its entire evolution. However, it is not able to accurately account for the energy distributions during the second phase of the remnant. The energy analysis was useful because it revealed that for most supernova at a redshift of 15, the time at which shocks are important for ionizing the IGM, the Hubble flow can be neglected, which greatly simplifies the problem. Last, using a simplistic model that describes the first phase of a remnant, we were able to verify the claims made that the free expansion solution breaks down when the shock absorbs a mass equal to the initially ejected mass. The above analysis is fundamental because explosions, and their resulting shocks, greatly alter the surrounding medium. Since the surrounding medium is important to most astrophysical questions, knowledge of shocks is necessary.

Appendix A: Accuracy of Numerical Analysis

The coding was all straightforward for most of the cases described above. I typically used a second order Runge-Kutta algorithm. However, numerically finding the internal structure of the cosmological blastwave was more difficult because there were singularities involved. As a result, to show the accuracy of my code, I will analyze the solutions of the coupled differential equations at the singularity, using the Ostriker and McKee formalism, and plot the difference between my numerical results and the actual solutions near the singularity.

For a hollow blastwave, which was the case for the cosmological solution, the internal radius of the shock (behind which no mass exists), is defined as λ_i. Since the structure of the blastwave is self similar, λ_i remains constant, which implies the velocity at λ_i is $\lambda_i v_s$.

$$\rightarrow \nu_i = \frac{v_i}{\lambda v_s} = 1 \qquad \text{A.1}$$

Looking at the entropy equation (1.2.10), we see that at λ_i, ρ^* and P^* must diverge. Ostricker and McKee, in order to solve this problem, then define the following:

$$\rho^+ = (1 - \nu)\, \rho^* \qquad P^+ = (1 - \nu)\, P^* \qquad \text{A.2}$$

And noting that as $\lambda \rightarrow \lambda_i$

$$\Theta = \frac{(\gamma - 1)\,(1 - \nu)\,\nu^2}{2\,(\gamma\nu - 1)} \qquad \text{A.3}$$

The continuity equations then reduce to:

$$\nu v^* = \rho^+ + k_\rho - 2\nu \qquad \text{A.4}$$

$$(1 - \nu)\, v^* = \frac{(\gamma - 1)\,\nu P^+}{2\,(\gamma\nu - 1)} + \frac{1}{2}\,(k_\rho - k_P) \qquad \text{A.5}$$

$$P^+ = \gamma\rho^+ + \gamma k_\rho - k_P \qquad \text{A.6}$$

Solving these equations at $\lambda = \lambda_i$, we get:

$$P_i^+ = 3 + k_E - k_\rho \qquad \text{A.7}$$

$$\rho_i^{+} = \frac{6 + 2\,k_E - (\gamma + 1)\,k_\rho}{\gamma} \qquad \text{A.8}$$

$$v_i^{*} = \frac{6 + 2\,k_E - k_\rho - 2\,\gamma}{\gamma} \qquad \text{A.9}$$

Assuming the following solution for a hydrodynamic variable x:

$$x \sim (\lambda - \lambda_i)^{l_{xi}} \qquad \text{A.10}$$

We get the following relation for the exponent:

$$l_{xi} = -\frac{x_i^{+}}{v_i^{*}} \qquad \text{A.11}$$

which implies:

$$l_{\rho i} = \frac{2\,(3 + k_E) - (\gamma + 1)\,k_\rho}{k_\rho + 3\,\gamma - 2\,(3 + k_E)} \qquad \text{A.12}$$

$$l_{Pi} = \frac{\gamma\,(3 + k_E - k_\rho)}{k_\rho + 3\,\gamma - 2\,(3 + k_E)} \qquad \text{A.13}$$

Given these power laws, the following plots show how close my numerically computed solutions are to the actual solutions near the singularity.

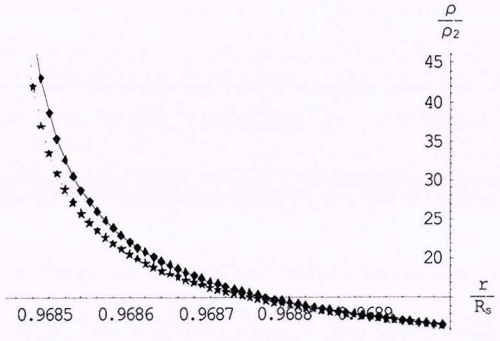

Figure A.1 – Actual solution is to the left

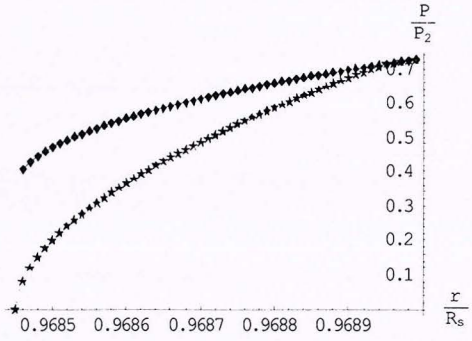

Figure A.2 – Actual Solution goes to zero

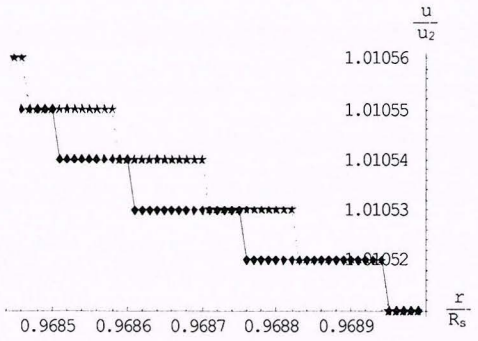

Figure A.3 – Actual Solution is above

For figure A.1, we see that the actual solution has a stronger singularity than my numerically computed solution. However, both profiles go to infinity at approximately the same normalized radius. For figure A.2, the actual solution reaches zero identically, while mine stops at .4. I believe this is just a result of the numerical integrator not being able to handle the coupled singularity in the density. For figure A.3, the difference in the actual verse numerical solution always stays within .00001, which is a trivial difference. In addition, to make sure difference would not affect the data, I computed the energy distributions for two sets of internal structures. The first set was my numerical solutions. The second set was my numerical solutions connected to the correct analytical solutions near the singularity. The difference in energy distributions between these two sets was 1 part in a 100.

Appendix B: Post-Shock Temperature

The specific internal energy for a gas particle is:

$$\epsilon = \frac{E}{m} = \frac{P}{(\gamma - 1)\,\rho} \qquad \text{where } \gamma = \text{ratio of specific heats} \tag{B.1}$$

Setting E equal to the energy in terms of T, we get:

$$\frac{P\,m}{(\gamma - 1)\,\rho} = \frac{3}{2}\,kT \rightarrow T = \frac{P}{(\gamma - 1)\,\rho}\,\frac{2\,m}{3\,k} \tag{B.2}$$

Then, plugging in the values for the post shock values of P and ρ given in equation 1.1.3, we get:

$$T = \frac{4\,m}{3\,k}\,\frac{\gamma - 1}{(\gamma + 1)^2}\,U_{sh}^{\,2} = \frac{3\,m}{16\,k}\,U_{sh}^{\,2} \qquad \text{B.3}$$

Where in the last step I took $\gamma = 5/3$. Note that using the post shock temperature to determine the cooling function's value, even though the temperature increases dramatically inside the shock, is valid because most of the cooling occurs near the shock radius, where the density is highest.

Appendix C: Code

Code used for Self Similar Solution

```
// Omega = 1 solution for expanding universe
// including gravity

#include <iostream>
#include <vector>
#include <cmath>
#include <fstream>

using namespace std;

vector <double> den;
vector <double> vel;
vector <double> pres;
vector <double> grav;
double den_1, vel_1, pres_1;

// important parameters (fix)

double g = 6.672E-8;
double t = 365E2 * 24 * 60 * 60; // seconds
//double t = 10.;
double r_s = 1.898 * pow(1E51 * g * pow(t, 4), 1/5.); // cm
//double r_s = pow(1E51 / 2E-24 * pow(t, 2), 1/5.) * 1.17;
double out_den = 1 / 6. / M_PI / g / pow(t, 2); // gm / cm3
//double out_den = 2E-30;
double v_s;

double step;
int accuracy; // # of steps away from singularity to recompute
double lambda_0 = 1.;
double gam = 5 / 3.;
double eta = 4 / 5.;
int length;
int temp = 0;
```

```
double k_p = 3.;
double k_d = 2.5;
double v_h = 5 / 6.;
double omega = 1.;

double lam_i;
double l_p = gam * (3. - k_d) / (k_d + 3. * gam - 6.);
double l_d = (6. - (gam + 1) * k_d) / (k_d + 3. * gam - 6.);
double c_p, c_d, c_v;

void Init_funcs() {

  length = (int)(lambda_0 / step) + 1;
  den.resize(length);
  vel.resize(length);
  pres.resize(length);
  grav.resize(length);
  lambda_0 = step * (length - 1);
  v_s = eta * r_s / t;

  // Initial values set according to jump conditions

  den_1 = (gam + 1) / (gam - 1) * out_den;
  vel_1 = (2 + (gam - 1) * v_h) / (gam + 1) * v_s;
  pres_1 = 2 / (gam + 1) * out_den * pow(v_s * (1 - v_h), 2);

  // normalized values

  den[length - 1] = 1;
  vel[length - 1] = 1;
  pres[length - 1] = 1;

  // intially define gravity to have a lambda distribution

  for (int a = 0; a < length; a++) {

    grav[a] = step * a;

  }

}

vector <double> Compute_der(double l, double d, double v, double p,
double g) {

  temp++;
  vector <double> ret(3);
  double v_log, d_log, p_log;
  double v_der, d_der, p_der;
  double theta = p * pres_1 / d / den_1 / pow(l * v_s, 2);
  double v_n = v * vel_1 / l / v_s;

  double v_log_num, v_log_den, d_log_num, d_log_den, p_log_num,
p_log_den;

  double g_mine = omega * pow(v_h / l, 2) * g / v_n;
  //double g_theirs = omega * pow(v_h, 2) / pow(l, 3) * g;
```

```
  v_log_num = 2. * theta * (k_p - 2. * gam * v_n) + (1 - v_n) * ((k_p -
k_d) * v_n - g_mine);
  v_log_den = 2. * v_n * (gam * theta - pow(1 - v_n, 2));
  v_log = v_log_num / v_log_den;

  d_log_num = 2. * theta * (k_p - gam * k_d) / (1 - v_n) - (3. * v_n -
2.) * k_d + k_p * v_n - 4. * v_n * (1 - v_n) - g_mine;
  d_log_den = 2. * (gam * theta - pow(1 - v_n, 2));
  d_log = d_log_num / d_log_den;

  p_log_num = -gam * v_n * k_d + k_p * (gam * v_n + 2. * (1 - v_n)) -
4. * gam * v_n * (1 - v_n) - gam * g_mine;
  p_log_den = 2. * (gam * theta - pow(1 - v_n, 2));
  p_log = p_log_num / p_log_den;

  v_der = v_log * v / l;
  d_der = d_log * d / l;
  p_der = p_log * p / l;

  ret[0] = step * d_der;
  ret[1] = step * v_der;
  ret[2] = step * p_der;

  //cout << l << endl;
  //cout << "v_log " << v_log_num << " " << v_log_den << " " << v_log
<< endl;
  //cout << "d_log " << d_log_num << " " << d_log_den << " " << d_log
<< endl;
  //cout << "p_log " << p_log_num << " " << p_log_den << " " << p_log
<< endl;
  //cout << endl;

  //if (temp % 5 == 0)
  //cin >> temp;

  return ret;

}


void Compute_funcs() {

  double lambda;

  vector <double> k1(3);
  vector <double> k2(3);
  //vector <double> k3(3);
  //vector <double> k4(3);

  int a = length - 2;
  lambda = lambda_0 - step * (length - 2 - a);

  while (lambda > .90) {

    //  for (int a = length - 2; a >= 0; a--) {
```

```
    //lambda = lambda_0 - step * (length - 2 - a);
    //cout << lambda << endl;

    k1 = Compute_der(lambda, den[a + 1], vel[a + 1], pres[a + 1],
grav[a]);
    k2 = Compute_der(lambda - .5 * step, den[a + 1] - .5 * k1[0], vel[a
+ 1] - .5 * k1[1], pres[a + 1] - .5 * k1[2], grav[a + 1]);
    //k3 = Compute_der(lambda - .5 * step, den[a + 1] - .5 * k2[0],
vel[a + 1] - .5 * k2[1], pres[a + 1] - .5 * k2[2]);
    //k4 = Compute_der(lambda - step, den[a + 1] - k3[0], vel[a + 1] -
k3[1], pres[a + 1] - k3[2]);

    //cout << k2[0] << endl;
    //cout << k2[1] << endl;
    //cout << k2[2] << endl;
    //cin >> temp;

    den[a] = den[a + 1] - k2[0];
    vel[a] = vel[a + 1] - k2[1];
    pres[a] = pres[a + 1] - k2[2];

    //den[a] = den[a + 1] - 1 / 6. * (k1[0] + 2 * k2[0] + 2 * k3[0] +
k4[0]);
    //vel[a] = vel[a + 1] - 1 / 6. * (k1[1] + 2 * k2[1] + 2 * k3[1] +
k4[1]);
    //pres[a] = pres[a + 1] - 1 / 6. * (k1[2] + 2 * k2[2] + 2 * k3[2] +
k4[2]);

    a--;
    lambda = lambda_0 - step * (length - 2 - a);
  }

}

void Determine_lam() {

  double lambda;

  int a = (int)(length * .96);
  lambda = step * a;
  double dif = abs(vel[a] - lambda * 24 / 23.);

  while ((abs(vel[a] - lambda * 24 / 23.) <= dif) && (a < length)) {

    dif = abs(vel[a] - lambda * 24 / 23.);
    a++;
    lambda = step * a;;

  }

  lam_i = lambda - step;
  cout << lam_i << endl;

}

void ReCompute() {
```

```
  // fit proportionality constants;

  int a_i = (int)(lam_i / step);
  int a_f = a_i + accuracy;

  if (a_f >= length) {
    a_f = length - 1;
  }

  c_p = pres[a_f] / pow(step * (a_f - a_i), l_p);
  c_d = den[a_f] / pow(step * (a_f - a_i), l_d);
  c_v = (1 - vel[a_f] * 23 / 24. / (step * a_f)) / (step * (a_f -
a_i));

  // recompute function from a_i to a_f

  for (int a = a_i; a <= a_f; a++) {

    den[a] = c_d * pow(step * (a - a_i), l_d);
    pres[a] = c_p * pow(step * (a - a_i), l_p);
    vel[a] = (1 - c_v * step * (a - a_i)) * step * a * 24 / 23.;

  }

}

void Determine_grav() {

  double lam;
  double sum = 0;

  for (int a = 0; a < length; a++) {

    lam = step * a;

    if (a < (int)(length * lam_i)) {

      grav[a] = 0;

    }

    else {

      sum += den[a] * pow(lam, 2) * step;
      grav[a] = sum;

    }

  }

  for (int b = (int)(length * lam_i); b < length; b++) {

    lam = step * b;
    grav[b] = grav[b] / grav[length - 1] / pow(lam, 2);

  }
```

```
}

void Output_vals() {

  double lambda;

  ofstream out_A("den.dat");
  ofstream out_B("vel.dat");
  ofstream out_C("pres.dat");
  ofstream out_D("grav.dat");

  for (int a = (int)(length * lam_i); a < length; a++) {

    lambda = step * a;

    //if (abs(den[a]) < 5)
    out_A << lambda << " " << den[a] << endl;

    //if (abs(vel[a]) < 5)
    out_B << lambda << " " << vel[a] << endl;

    //if (abs(pres[a]) < 5)
    out_C << lambda << " " << pres[a] << endl;

    out_D << lambda << " " << grav[a] << endl;
    //a--;

  }

  out_A.close();
  out_B.close();
  out_C.close();
  out_D.close();

}

void Determine_Energy() {

  double therm = 0;
  double kin = 0;
  double lambda;

  for (int a = length - 1; a > (int)(length * lam_i); a--) {

    lambda = a * step;
    therm += pres[a] * pow(lambda, 2) * step;
    kin += den[a] * pow(vel[a] * lambda, 2) * step;

  }

  cout << "Thermal: " << therm / (therm + kin) << endl;
  cout << "Kinetic: " << kin / (therm + kin) << endl;

}

int main() {
```

```
  int iterate;
  //cout << "Starting value for t (after explosion): ";
  //cin >> time;
  cout << "Step size: ";
  cin >> step;
  cout << "# of steps: ";
  cin >> accuracy;
  cout << "# of iterations: ";
  cin >> iterate;
  Init_funcs();
  Compute_funcs();
  for (int a = 0; a < iterate; a++) {
    Determine_lam();
    //ReCompute();
    Determine_grav(); // concerned about my definition
    Compute_funcs();
  }
  Determine_lam();
  ReCompute();
  Determine_Energy();
  Output_vals();
  return 0;

}
```

Code used for Explosion Model

```
#include <iostream>
#include <vector>
#include <cmath>
#include <fstream>

using namespace std;

// notice f_d and M_5

vector <double> r;
vector <double> r_prime;
vector <double> q;
double omega_b = 1/7.;
double omega_d = 6/7.;
double t_fin, step;
int length, temp;
double f_d;
double f_m = .1;
int a_vel = 0;
double t_burn = 5E7;
double M_5 = omega_b * 20.;
double H_0 = 3.24E-18;
double h = 0.5;
double t_hubble;

void Init_funcs() {

  length = (int)(t_fin / step) + 1;
```

```
  r.resize(length);
  r_prime.resize(length);
  q.resize(length);
  t_fin = step * (length - 1);

  r[0] = 1.;
  r_prime[0] = 0.; // doesnt matter long run
  q[0] = 1.;

}

vector <double> Compute_der(double t, double R, double Rprime, double
P) {

  vector <double> ret(3);
  double rDPrime, qPrime, eta, grav, l_diss;

  eta = 1 / (1 + t);

  grav = -(2 / 9. * omega_d + 1 / 9. * omega_b) * pow(eta, 2) * R;

  rDPrime = 18 * M_PI / omega_b / pow(eta,2) * P / R - 3. *
(pow(Rprime, 2) - 4 / 3. * eta * R * Rprime + 4 / 9. * pow(eta * R, 2))
/ R + grav;

  l_diss = 1 / 3. * f_d * omega_b * pow(eta * R, 2) * pow(Rprime - 2 /
3. * eta * R, 3);

  //l_diss = 0;

  // changed to make NOT constant energy.

  if (t_hubble * t < t_burn) {

    l_diss += 1;

  }

  qPrime = l_diss / 2. / M_PI / pow(R, 3) - 5 * Rprime / R * P;
  //cout << l_diss / 2. / M_PI / pow(R, 3) / (5 * Rprime / R * P) <<
endl;

  ret[0] = step * rDPrime;
  ret[1] = step * Rprime;
  ret[2] = step * qPrime;

  //cout << t << " " << rDPrime << endl;
  return ret;

}

void Compute_funcs() {

  double t;

  vector <double> k1(3);
  vector <double> k2(3);
```

```
  for (int a = 1; a < length; a++) {

    t = (a - 1) * step;

    k1 = Compute_der(t, r[a-1], r_prime[a-1], q[a-1]);
    k2 = Compute_der(t + .5 * step, r[a-1] + .5 * k1[1], r_prime[a-1] +
.5 * k1[0], q[a-1] + .5 * k1[2]);

    r[a] = r[a - 1] + k2[1];
    r_prime[a] = r_prime[a - 1] + k2[0];
    q[a] = q[a - 1] + k2[2];


    //if ((q[a] > q[a-1]) || q[a-1] == 0)
    //  q[a] = 0;

    //cout << "Z: " << z << endl;
    //cout << "R Doulbe prime: " << rDPrime << endl;
    //cout << "R prime: " << r_prime[a - 1] << endl;
    //cout << "Q prime: " << qPrime << endl;
    //cin >> temp;

    //cout << z << " " << rDPrime << endl;

  }

}

void Determine_vel() {

  double eta = 1;
  double diff_min = abs(2 / 3. * eta * r[0] / r_prime[0] - 5 / 6.);
  double t;
  double t_vel;

  for (int a = 1; a < length; a++) {

    t = a * step;
    eta = 1 / (1 + t);
    if (abs(2 / 3. * eta * r[a] / r_prime[a] - 5 / 6.) < diff_min) {

      a_vel = a;
      diff_min = abs(2 / 3. * eta * r[a] / r_prime[a] - 5 / 6.);
      t_vel = t;

    }

  }

  //cout << "a" << a_vel << endl;
  cout << "t_vel " << t_vel << endl;

}

void Output_vals() {
```

```
  double t, eta, l_diss;
  double e_init = 2. * M_PI * pow(r[0], 3) * q[0] + omega_b / 9. *
pow(r[0], 3) * pow(r_prime[0], 2);
  double e_last;

  double r_factor = .13 / h / (1 + t_fin) * pow(M_5, .2);

  ofstream out_A("r.dat");
  ofstream out_B("q.dat");
  ofstream out_C("log.dat");
  ofstream out_D("slope_log.dat");
  ofstream out_E("l_vs_w.dat");
  ofstream out_F("therm.dat");
  ofstream out_G("kin.dat");
  ofstream out_H("rel_vel.dat");
  ofstream out_I("temp.dat");
  ofstream out_J("slope.dat");

  double temp;
  double temp_1 = 10000;
  int counter = 0;

  for (int a = 0; a < length; a++) {

    t = a * step;

    //if (abs(r[a]) < 1000)
    out_A << t << " " << r[a] * r_factor << endl;

    //if (abs(q[a]) < 1000)
    out_B << t << " " << q[a] << endl;

    out_C << log(t + 1) << " " << log(r[a]) << endl;

    if (a != length - 1) {
      out_D << t << " " << (log(r[a + 1]) - log(r[a])) / (log(t + 1 +
step) - log(t + 1)) << endl;

      out_J << t << " " << r_prime[a] << endl;

      if (a == length - 2)
      cout << "scaling " << (log(r[a + 1]) - log(r[a])) / (log(t + 1 +
step) - log(t + 1)) << endl;
    }

    eta = 1 / (1 + t);
    l_diss = 1 / 3. * f_d * omega_b * pow(eta * r[a], 2) *
pow(r_prime[a] - 2 / 3. * eta * r[a], 3);
    out_E << t << " " << l_diss - 4 * M_PI * q[a] * r_prime[a] *
pow(r[a], 2) << endl;

    out_F << t << " " << 2. * M_PI * pow(r[a], 3) * q[a] / e_init <<
endl;
    out_G << t << " " << omega_b / 9. * pow(r[a], 3) * pow(r_prime[a] *
eta, 2) / e_init << endl;

    if ((a == length - 1) || (a == a_vel)) {
```

```
      e_last = 2. * M_PI * pow(r[a], 3) * q[a]  + omega_b / 9. *
pow(r[a], 3) * pow(r_prime[a] * eta, 2);
      cout << "Thermal: " << 2. * M_PI * pow(r[a], 3) * q[a] / e_init
<<
endl;
      cout << "Kinetic: " << omega_b / 9. * pow(r[a], 3) *
pow(r_prime[a]
* eta, 2) / e_init << endl;
      if (a == a_vel) {
      cout << "scaling " << (log(r[a + 1]) - log(r[a])) / (log(t + 1 +
step) - log(t + 1)) << endl;
      }
      cout << endl;

    }

    out_H << t << " " << r_prime[a] / (2 / 3. * eta * r[a]) << endl;

    temp = 4.5E5 * pow(M_5, .4) * q[a] / f_m / omega_b / pow(eta, 2);
    out_I << t << " " << temp << endl;

    if ((counter == 0) & (temp < temp_1)) {

      cout <<"t of temperature: " << t << endl;
      cout << endl;
      counter++;

    }

  }

  out_A.close();
  out_B.close();
  out_C.close();
  out_D.close();
  out_E.close();
  out_F.close();
  out_G.close();
  out_H.close();
  out_I.close();
  out_J.close();

}

int main() {

  cout << "End val of t: ";
  cin >> t_fin;
  cout << "Step size: ";
  cin >> step;
  cout << "f_d: ";
  cin >> f_d;
  //cout << "T burn: ";
  //cin >> t_burn;
  cout << "Initial z: " << pow(1 + t_fin, 2/3.) - 1 << endl; //possible
wrong
```

```
  cout << endl;
  t_hubble = 2 / 3. / (H_0 * h) * pow(1 + t_fin, -1);
  cout << "t_hubble: " << t_hubble << endl;
  cout << endl;
  Init_funcs();
  Compute_funcs();
  Determine_vel();
  Output_vals();
  return 0;

}
```

References

1) Astrophysical Blastwaves, Reviews of Modern Physics, Vol. 60, No. 1, January 1988.

2) The Astrophysical Journal, 417:54-62, 1993 November 1

3) Shu, Frank H. The Physics of Astrophysics: Volume 2 Gas Dynamics. University Science Books, 1992.