Computer Lab for Astrophysical Gasdynamics Course

Garrelt Mellema

Dept. of Astronomy, Stockholm University
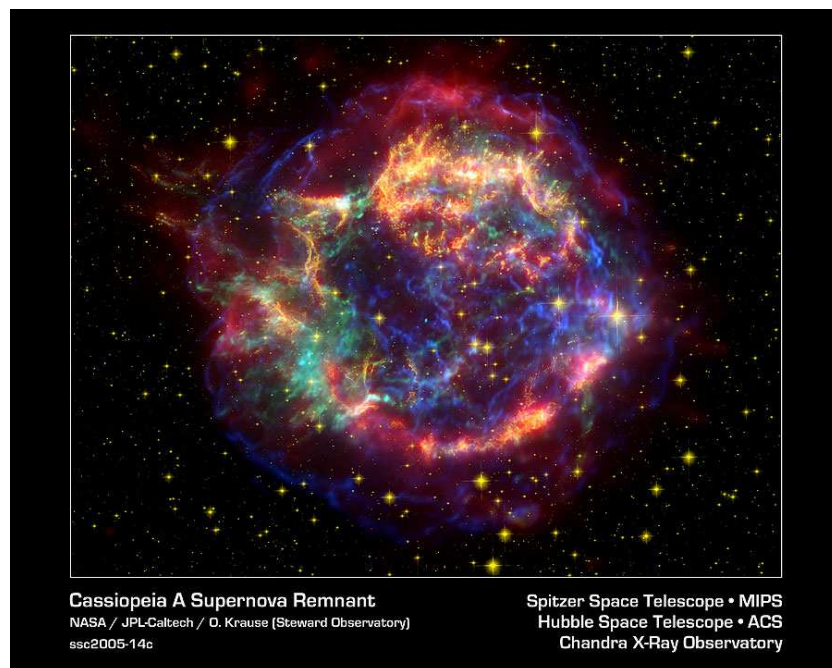
September 18, 2011

Figure 1: Composite image of the supernova remnant Cas A. Infrared data from the Spitzer Space Telescope are colored red; visible data from the Hubble Space Telescope are yellow; and X-ray data from the Chandra X-ray Observatory are green and blue

1 Introduction

In this lab you will run different (small) simulations of gasdynamic problems, the results of which you are asked to analyse. There are two main parts. In the first part we will be looking at so-called Riemann problems, in which two constant states are exposed to each other, leading to the formation of different types of discontinuities. In the second part we will study the interaction of a supernova blastwave with the supernova's environment, the so-called 'circumstellar interaction' problem for supernovae.

In this introduction we describe the general details of the simulations.

1.1 Technical Details

All of the necessary files can be found in Mondo (`http://mondo.su.se`). You should have added *AS7002 HT11* to your list of active sites. The files can be found under `Resources` and then `Computer Lab`.

1.1.1 Executables

Each of the two problems has its own executable which are called `riemann_ht11` and `supernova_ht11`. These executables have to be run on a Linux system by for example giving the command `./riemann_ht11`. Note that in order to run them you will need to turn on the executable permission for the two binary files after you downloaded them (e.g. `chmod u+x riemann_ht11`). They will ask for input parameters, which can either be given interactively, or using the Unix redirection option, from a file. If you have collected the input parameters in a file (say `sn_example.in`), you can run for example `./supernova_ht11 < sn_example.in`. An example of such an input file can be found in the Mondo resources. It is recommendable to keep the input parameters in a file so that you have a record of what went into a particular simulation. Similarly it may be good to make a separate directory for every simulation run (also because the programs overwrite the output files, see below).

1.1.2 Input

The required input is slightly different for the two programs, so it is described in detail below. Both codes need to be told the size of the computational mesh (how many computational cells should be used), the size of the domain on which the calculation is done, some parameters describing the initial conditions, the time between outputs, and the total time for which to run the simulation (time here means physical time, not wall clock or CPU time). The `riemann_ht11` program will also ask for the CFL number (which controls the length of the time steps, see Chapter 9 of the Lecture Notes).

1.1.3 Output

The codes produce two types of output, one to the terminal window, the other to files. The output to the terminal is meant to monitor the progress of the simulation. The code prints one line for every time step showing the number of the time step, the time, size

of the time step and the time when the next output file is produced. At the end of the run it will report the CPU time and real time (so-called wallclock time) used by the program.

The output files are ASCII tables containing four columns of the length of the mesh. The four columns are the position (in cm), the mass density (ρ in g cm^{-3}), the velocity (u in cm s^{-1}), and the pressure (p in dyne cm^{-2} = g cm^{-1} s^{-2}). The units are really only important for the post-processing in the supernova case.

For each output time a new file is created, and they are numbered sequentially from `output000.dat` (initial conditions) to (maximally) `output999.dat`. How many output files are generated depends on the frequency of outputs specified in the input parameters. Note that the output files will overwrite any files with the same name in the directory where you run the program, so it is best to run different simulations in different directories.

The output files can be read into your favourite plotting program for further analysis. Examples of plotting programs are `python`, `IDL`, `gnuplot`, `supermongo`, `matlab`, `etc`. Ask for help if you have problems plotting the results.

1.1.4 End of simulation

Both programs will run until one of these conditions is met

1. The specified end time is reached.

2. The generated flow patterns have reached an edge of the grid.

3. The maximum of 1000 output files have been generated.

4. An error has occured in the calculation of the evolution of the flow.

It is not strictly necessary to stop on condition 2, but since the purpose of the lab is to analyze the flow patterns, it is best if they are all contained on the grid.

If the code stops on condition 3, you should choose to generate output files less often by increasing the time between outputs (specified in the input parameters).

Condition 4 shows that the numerical method used has its draw backs. In fact most numerical methods for gasdynamics have such draw backs, not every problem can be handled by an off-the-shelf method. For the problems you are asked to study below, errors in the numerical method should not normally occur, but it is important to remember that the codes are not guaranteed to work for arbitrary initial conditions.

Both codes solve the Euler equations in one spatial direction, assuming $\rho\mathcal{E} = p/(\gamma - 1)$ with $\gamma = 5/3$.

1.1.5 Writing the report

To pass this part of the course you will need to write a report about what you did. The report should address the questions associated with each of the problems, but it should be written such that it can be read independently from these instructions. In other words, write it as if it was a report on experiments you have done, not as if you are answering questions that were posed to you.

2

2   Riemann Problems

Riemann problems are defined by the initial condition

$$\mathbf{W}(t = 0) = \left\{ \begin{array}{lll} \mathbf{W}_{\text{left}} & : & x < x_0 \\ \mathbf{W}_{\text{right}} & : & x > x_0 \end{array} \right.$$

They can be thought of two constant states seperated by a diaphragm at x_0 which is removed at $t = 0$. In the program `riemann_ht11` x_0 is always the middle of the computational grid and is defined to be 0. To define the grid you only need to specify the size of the domain. The initial condition for the Riemann problem is set by specifying the left and right values of the density, velocity and pressure.

As input, the code also needs the so-called CFL number (should be set to a number smaller than 1; for a safety margin, use a number in the range $0.4 - 0.8$, except when asked to try a different value).

The Riemann problem for the Euler equations leads to two types of waves spreading. The two types are shock and expansion waves. The possible Riemann configurations are thus,

1. one shock and one expansion wave

2. two shock waves

3. two expansion waves.

In each of the cases there will be a contact discontinuity in between the two waves (due to the fact that we started with an initial discontinuity). In this lab, we will look only at cases 1 and 2[1].

2.1   Shock tube

The first Riemann problem we will consider is the so-called shock tube.

A shock tube is a device used primarily to study gas phase combustion reactions. A simple shock tube is just a metal tube in which a gas at low pressure and a gas at high pressure are separated using a diaphragm. This diaphragm suddenly bursts open under predetermined conditions to produce a shock wave. The low-pressure gas, referred to as the driven gas is subjected to the shock wave while the high pressure gas is known as the driver gas. The bursting of the diaphragm creates a compression wave in the driven gas, which then rapidly steepens to form a shock front, known as the incident shock wave. Simultaneously, a rarefaction wave, often referred to as a fan, travels back in to the driver gas. The experimental gas and the driver gas make contact at the contact surface, which moves rapidly along the tube behind the shock front (from Wikipedia, the free encyclopedia).

We do not consider any combustion reactions here, but are instead interested in the pattern of discontinuities. In fact one particular shock tube problem, known as the Sod shock tube (Sod 1978), is one of the classical test problem for gasdynamics codes.

---

[1]If you are curious you may also study a dual expansion flow. You will notice that the code is more likely to generate errors in this case. To get a result choose velocities which are close to sonic.

We will use the initial conditions of the Sod shock tube: $\rho_L = 1$, $v_L = 0$, $p_L = 1$, $\rho_R = 0.125$, $v_R = 0$, $p_R = 0.1$ but since the code uses $\gamma = 5/3$, we are not exactly doing the classical Sod problem (which requires $\gamma = 1.4$). Use a domain size of 1.

A) To set good values for the output and end times, estimate a typical velocity from the initial conditions given above.

B) Run the simulation and inspect for a late time the graphs for ρ, v, p, as well as the total energy density E, specific internal energy density $\mathcal{E}$, the Mach number $\mathcal{M}$ (in the lab frame), and the entropy related quantity $p\rho^{-\gamma}$. Identify the three discontinuities: expansion wave, contact discontinuity and shock, and motivate your choices by identifying their characteristics.

C) In the same output, derive the shock strength (characterized by the Mach number in the shock frame) and the shock's velocity (in the lab frame) from the pre- and post-shock conditions.

D) Measure the speed of the 3 discontinuities by looking at a range of outputs. Are the values for the shock the same as derived under C)?

E) Try running with a CFL number larger than 1. If numerical errors occur before the first output is produced, try changing the output time so that at least one output is produced. What do the results look like?

2.2   Collision

A different Riemann problem is the one in which two flows collide. If the flows are supersonic this will generate two shocks seperated by a contact discontinuity. The simplest case would be a head-on collision between two flows. Since one can change the reference frame to another one which is moving with a constant velocity (Galileo transformation), there are in fact many variations on this theme. You do not need to analyse this case in as much detail as the previous one, but this case bears some similarities to the supernova problem from the next section, so you should have a look at it.

The choose as initial conditions the following set up: $\rho_L = 0.1$, $v_L = 10$, $p_L = 0.1$, $\rho_R = 1$, $v_R = 2$, $p_R = 1$ The size of the domain can again be set to 1. For the end time either experiment or make an estimate as in case of the shock tube.

A) Run the simulations and inspect for a late time the graphs for ρ, v, p, as well as the specific internal energy density $\mathcal{E}$. Identify the three discontinuities.

3   Supernova Explosion

In the second part we look at an astrophysical application. The program `supernova_ht11` calculates the evolution of the earliest phases of a supernova explosion when the stellar atmosphere ejected by the explosion interacts with the environment of the star. This process is often called "circumstellar interaction", and Claes Fransson and Peter Lundqvist at Stockholm Observatory have both worked on this problem (see e.g., Fransson et al. 1996).

We will assume that both the supernova ejecta and the environment are spherically symmetric, which allows us to solve the problem in one dimension, with as spatial

variable the radius r.

The stellar atmosphere of a star exploding as a supernova can be described by a powerlaw of index n, where n is in the range 6–12. As the envelope expands it decreases in density. Initially the expansion is completely unhindered, so the density profile is given by

$$\rho = \rho_0 \left(\frac{r_0}{r}\right)^n \left(\frac{t_0}{t}\right)^3 \ , \tag{1}$$

where ρ_0, r_0 and t_0 are reference values. In such a free expansion the velocity is given by

$$v = v_0 \left(\frac{r}{t}\right) \left(\frac{t_0}{r_0}\right) \ , \tag{2}$$

so it increases with position. At some point the whole atmosphere will have been ejected but in our simulation we will not consider that transition.

The environment of the star could be the interstellar medium (ISM) of roughly constant density, or more likely in the case of a massive star, the remnant of the star's stellar wind. In either case it can be described by a powerlaw

$$\rho = \rho_1 \left(\frac{r_0}{r}\right)^s \ , \tag{3}$$

with a velocity v_1 (which could be zero in the constant density ISM case). We will assume a supersonic stellar wind, in which case $s = 2$, and the wind can be characterized by a constant mass loss rate $\dot{M} = 4\pi r^2 \rho v_1$.

This configuration has an analytical solution found by Chevalier (1982). This solution is self-similar which means that apart from scale factors, the solution is identical for all times.

The interaction between the supernova ejecta and the environment leads to four zones:

1. Freely expanding supernova ejecta

2. Shocked supernova ejecta

3. Shocked environment

4. Undisturbed environment

separated by three discontinuities:

1. an inner shock where the stellar ejecta are shocked

2. a contact discontinuity between the shocked stellar ejecta and the shocked environment,

3. an outer shock where the environment is shocked

In this sense it resembles the second Riemann problem we looked at above. However, we are now working in the spherical coordinate r, so the problems are not identical, specifically, the flow variables are not necessarily constant within the different zones.

In astrophysical problems radiative cooling can often modify the flow. The program supernova_ht11 allows you to turn radiative cooling on and off. If you run with cooling you should have the file called corocool.tab available in your working directory. This file contains a table of the radiative cooling rate as a function of temperature. Without it the code will stop.

3.1 Adiabatic Simulation

We start without radiative cooling, so type n when asked whether to use it. We run the simulation with the following parameters:

- mesh size: 4000 (or more)
- domain size: 2.5×10^{15} cm
- $n = 9$
- $r_0 = 5 \times 10^{14}$ cm
- $\rho_0 = 1.3 \times 10^{-18}$ g cm^{-3}
- $v_0 = 5.8 \times 10^4$ km s^{-1}
- $T_0 = 10^4$ K
- $\dot{M}_{\text{wind}} = 5 \times 10^{-5}$ M$_\odot$ yr^{-1}
- $v_{\text{wind}} = 10$ km s^{-1}
- $T_{\text{wind}} = 10^4$ K

For the output times you can take 0.1 days, and for the total time 30 days. This will lead to the outer shock running off the grid, at which point the simulation will stop.

Note that the variations across the grid for many quantities are such that it is better to plot the logarithm ($\log_{10}$) of these quantities. Also remember that the units are in the cgs system (cm, g, s).

A) Look at the output for 9 days. Identify the four regions and three discontinuities (use as many flow quantities as you think is useful). Measure the shock jump conditions and derive the shock strengths (as measured by the Mach number $\mathcal{M}$ in the shock frame) in both the inner and outer shock.

B) The full set of parameters for the self-similar solutions is given in the table (Table 1). We will only use the ratios of the inner and outer shock position to the contact discontinuity (R_1/R_C, R_2/R_C) for the case we are considering: $s = 2$, $n = 9$. Check these ratios for various times in the simulation. Do the results match the self-similar solution for all times?

C) The x-ray emissivity of a hot gas can be approximated by (free-free emission from hydrogen[2]):

$$4\pi n(\text{H}^+)n(\text{e}^-)\frac{f_k}{\sqrt{T}}\exp\left(\frac{-h\nu}{k_{\text{B}}T}\right) \quad \text{erg s}^{-1}\text{ cm}^{-3}\text{ Hz}^{-1} \tag{4}$$

---

[2]assuming the so-called Gaunt factors to be 1

TABLE 1

PROPERTIES OF THE SELF-SIMILAR SOLUTIONS

s	n	R_1/R_c	R_2/R_c	A	ρ_2/ρ_1	p_2/p_1	u_2/u_1	M_2/M_1
0	6	1.256	0.906	2.4	0.75	0.39	1.203	0.28
0	7	1.181	0.935	1.2	1.3	0.47	1.253	0.50
0	8	1.154	0.950	0.71	2.1	0.52	1.263	0.71
0	9	1.140	0.960	0.47	3.1	0.55	1.263	0.93
0	10	1.131	0.966	0.33	4.3	0.57	1.260	1.1
0	12	1.121	0.974	0.19	7.2	0.60	1.255	1.6
0	14	1.116	0.979	0.12	11	0.62	1.250	2.0
2	6	1.377	0.958	0.62	3.9	0.21	1.006	0.44
2	7	1.299	0.970	0.27	7.8	0.27	1.058	0.82
2	8	1.267	0.976	0.15	13	0.31	1.079	1.2
2	9	1.250	0.981	0.096	19	0.33	1.090	1.6
2	10	1.239	0.984	0.067	27	0.35	1.096	1.9
2	12	1.226	0.987	0.038	46	0.37	1.104	2.7
2	14	1.218	0.990	0.025	70	0.38	1.108	3.4

Figure 2: Table 1 from Chevalier (1982)

with $f_k = 5.44436 \times 10^{-39}$, n the number density (of protons and electrons respectively), T the temperature, h the Planck constant and k_B the Boltzmann constant. To keep things simple you may assume that the gas consists only of hydrogen. This means that the proton and electron number densities are equal. This also means that the μ needed for calculating the pressure from the mass density is $\mu = 0.5$ (since electrons contribute to the pressure, but hardly at all to the mass density ρ). Calculate the emissivity at a photon energy of 100 keV, with a bandwidth of 1 keV at $t = 9$ days. Which region produces substantial amounts of x-ray emission? Next calculate the total x-ray luminosity (at 100 keV) of the supernova by summing over all radial points (remember that the volume of each cell is $4\pi r^2 \Delta r$). Do this for a sufficient number of times to make an x-ray light curve (luminosity against time). Convert the luminosities to solar luminosities ($L_\odot = 3.846 \times 10^{33}$ erg s^{-1}) to check that your answer makes sense. X-ray emission from supernovae can be detected even from other galaxies, so their luminosity should exceed that of a star by many orders of magnitude, but it should not be many order of magnitudes more than the luminosity of an entire galaxy.

D) Run the same simulation with only 2000 points. Compare the necessary CPU time with the 4000 points run. How can you explain the ratio? Hint: think of the CFL condition.

3.2 Cooling Simulation

Run the same simulation, but now with radiative cooling on.
A) Look at the output for 9 days, and compare to the case without radiative cooling.

What has changed, and why?

B) Look in detail at the shock structures. Is the simulation resolving the cooling region in the sense that you can see the temperature go up just behind the shock and then cooling as the material flows away from the shock? If not, is it possible to make an estimate of the *effective* adiabatic index γ for the cooling shock?

C) Redo all of the x-ray calculation, and compare to the adiabatic result you found before.

References

[1] Chevalier, R. A. 1982, ApJ 258, 790

[2] Fransson, C., Lundqvist, P., Chevalier, R. A. 1996, ApJ 461, 993

[3] Sod, G. 1978, J. Comp. Phys., 27, 1