In AstroBEAR, we solve the following Euler equation :

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \vec{u}) = 0$$

$$\rho(\frac{\partial}{\partial t} + \vec{u} \bullet \nabla)\vec{u} + \nabla p = 0$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \bullet (\vec{u}(\varepsilon + p)) = \nabla \bullet (\kappa T^n \nabla T)$$

where the internal energy per unit volume is defined as:

$$\varepsilon = \rho e + \frac{1}{2}\rho u^2$$

$$e = p / (\gamma - 1)$$

The code is also capable of solving the Euler equation with central or point gravity, self-gravity, resistive MHD et al. Those are ignored here.

Then the following Euler equation is solved using Riemann solver:

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \vec{u}) = 0$$

$$\rho(\frac{\partial}{\partial t} + \vec{u} \bullet \nabla)\vec{u} + \nabla p = 0$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \bullet (\vec{u}(\varepsilon + p)) = 0$$

The diffusion part is separated by the operator splitting method. After converting internal energy to temperature, the separated diffusion equation is:

$$\frac{3}{2}\rho \frac{\partial T}{\partial t} = \nabla \bullet (\kappa T^n \nabla T)$$

Use 1-D case (along y axis) as an example, the equation is:

$$\frac{3}{2}\rho \frac{dT}{dt} = \frac{\partial}{\partial y} \kappa T^n \frac{\partial T}{\partial y}$$

Then, using Crank-Nicholson scheme, we convert the diffusion equation

$$\frac{3}{2}\rho\frac{dT}{dt} = \frac{\partial}{\partial y}\kappa T^{n}\frac{\partial T}{\partial y}$$

into

$$\frac{3}{2}\rho\frac{dT}{dt} = \frac{1}{2}\frac{\partial^2 z^*}{\partial y^2} + \frac{1}{2}\frac{\partial^2 z}{\partial y^2}$$

where $z = \dfrac{T^{n+1}}{n+1}$, and “*” denotes the newer time step we are solving.

By differentiating the above equation, we have:

$$\frac{3}{2}\rho(T_i^* - T_i) = \frac{\kappa dt}{2dy^2}(z_{i+1}^* + z_{i-1}^* - 2z_i^* + z_i + z_{i-1} - 2z_i)$$

Using Taylor expansion,

$$z^* = z + \frac{dz}{dT}(T^* - T) = z + T^{n+1}(T^* - T)$$

We convert the diffusion equation into:

$$\frac{3}{2}\rho(T_i^* - T_i) = k\left(\frac{1-n}{1+n}T_{i+1}^{n+1} + \frac{1-n}{1+n}T_{i-1}^{n+1} - 2\frac{1-n}{1+n}T_i^{n+1} + T_{i+1}^n T_{i+1}^* + T_{i-1}^n T_{i-1}^* - 2T_i^n T_i^*\right)$$

where $k = \dfrac{\kappa dt}{2dy^2}$

This equation has a form of

$$a_{i+1}T_{i+1}^* + a_i T_i^* + a_{i-1}T_{i-1}^* = a$$

with

$$a_{i+1} = T_{i+1}^n \qquad a_{i-1} = T_{i-1}^n \qquad a_i = -2T_i^n - \frac{3}{2}\rho$$

$$a = -\frac{3}{2}\rho T_i + k\frac{n-1}{n+1}(T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1})$$

.

The linear equations with this form is solved using the Hypre, which is essentially a library of linear solvers.
The entire process is summarized below:

(1) Riemann solver solves the classical fluid equation and updates the temperature.

(2) Using the updated temperature, we get all the “a” factors on the previous page, and construct the linear equation system.

(3) The preferred time step “dt” on the previous page is found by the following equation:

$$dt = \min\left(\frac{\rho l^2}{\kappa T^n}\right)$$

where “min” is taking the minimum over the entire computation domain, “l” is the local temperature scale length:

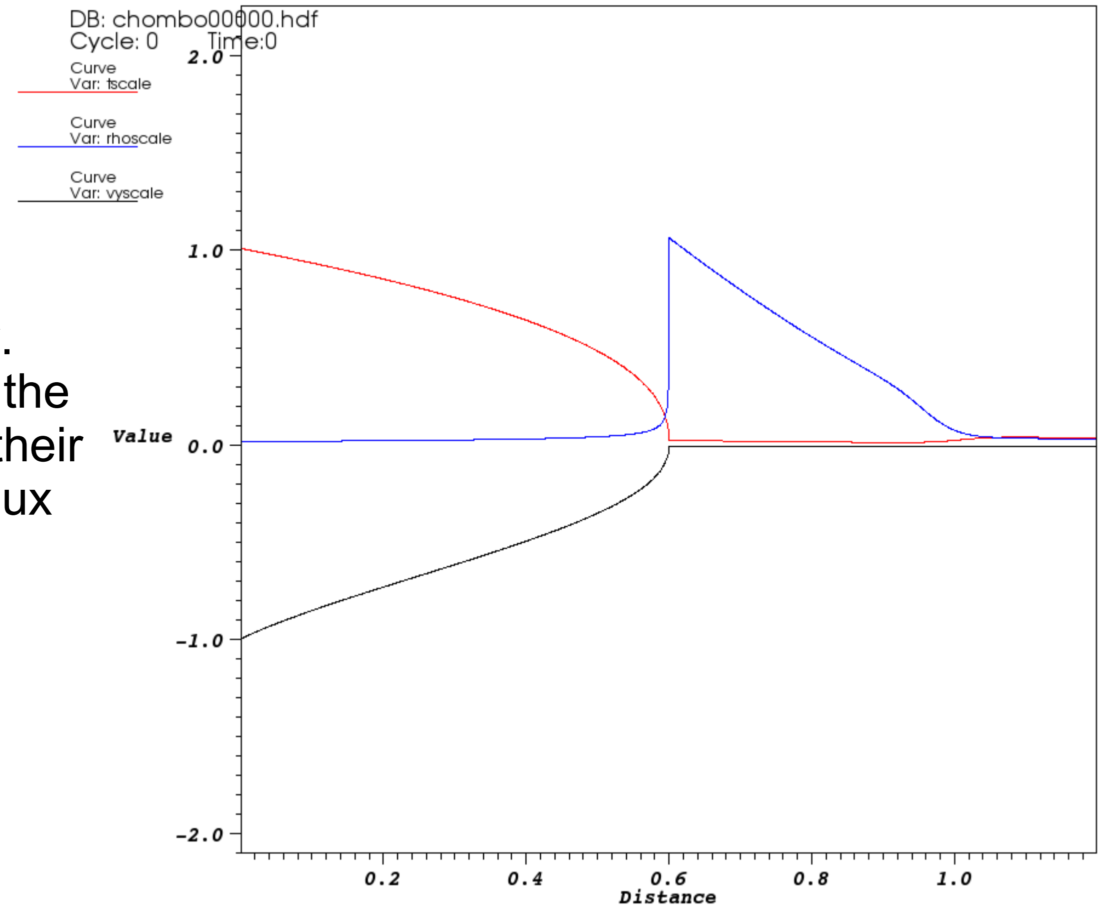
$$l = \frac{T}{|\nabla T|}$$

(4) The linear solver solves the system with time step dt, updates the temperature. It recycles till the accumulated time step equals the time step taken by the Riemann solver.

(5) Then the final updated temperature is used to calculate the internal energy, the updated internal energy goes into the next Riemann solver step.

Our problem is a RT instability with thermal diffusion and a heat flux applied at the bottom. The fluid has a speed profile along y direction. There is a uniform gravity applied on the -y direction. The density, velocity and temperature profiles are shown below:

Here, blue is density,
red is temperature, black is y velocity.
The horizontal axis is the y direction, the physics quantities are normalized to their peak values. There is a strong heat flux q applied at the bottom (y=0).



The boundary condition is treated as follows (I'll only mention the bottom boundary since that is where things are tricky):

(1) In the Riemann solver part, we have the following equations at the bottom boundary:

$$\frac{\partial(P+\rho v^2)}{\partial y} = -\frac{1}{2}(\rho_0+\rho_1)g$$

$$\kappa T^n \frac{\partial T}{\partial y} = q$$

where “0” and “1” denote boundary (ghost zone) cell and the first internal cell, respectively. “q” is the applied boundary heat flux, which is given fixed.

From the diffusion equation, we have:

$$\kappa\left(\frac{T_0+T_1}{2}\right)^n \frac{T_0-T_1}{dy} = q$$

Since T1 is known, we can solve this nonlinear equation using Newton's Method.

Then we convert the equilibrium equation

$$P_1 + \rho_1 v_1{}^2 - P_0 - \rho_0 v_0{}^2 = -\frac{1}{2}(\rho_0 + \rho_1) g dy$$

to quadratic equation about density:

$$(-\frac{1}{2} g dy + T_0)\rho_0{}^2 + (-\frac{1}{2} g dy \rho_1 - \rho_1 T_1 - \rho_1 v_1{}^2)\rho_0 + \rho_1 v_1{}^2 = 0$$

Since rho0 is the only unknown here, we solve this quadratic equation to obtain rho0.
We then obtain v0 using the mass flow conservation:

$$\rho_0 v_0 = \rho_1 v_1$$

(2) In the diffusion equation solver, we apply the boundary condition as follows:
Remember that we can convert the diffusion equation into the following linear system (page 4):

$$a_{i+1}T_{i+1}^{*} + a_i T_i^{*} + a_{i-1}T_{i-1}^{*} = a$$

with

$$a_{i+1} = T_{i+1}^{n} \qquad a_{i-1} = T_{i-1}^{n} \qquad a_i = -2T_i^{n} - \frac{3}{2}\rho$$

$$a = -\frac{3}{2}\rho T_i + k\frac{n-1}{n+1}(T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1})$$

At the bottom boundary, since we are solving the system implicitly, we should really have:

$$\kappa\left(\frac{T_0^{*} + T_1^{*}}{2}\right)^{n}\frac{T_0^{*} - T_1^{*}}{dy} = q$$

But since this equation does not give us an explicit relation of T0 and T1, I am using the following equation instead:

$$\kappa\left(\frac{T_0 + T_1}{2}\right)^{n}\frac{T_0^{*} - T_1^{*}}{dy} = q$$

This equation can be converted into the following relation:

$$T_0^* = T_1^* + dT$$

where dT is given by:

$$dT = \frac{qdy}{\kappa}(\frac{2}{T_0 + T_1})^n$$

So by substituting T0 into the first equation on the previous page, we can elliminate T0 in that equation. This is equivalent to changing the first few “a” coefficients in the linear system (see previous page) to “a' ”:

$$a_0' = 0$$

$$a_1' = a_1 + a_0$$

$$a' = a - a_0 dT$$

Since a0 now equals 0, the linear system is closed. We solve the system using the linear solver library.

The problem we encounter is that when using the solver described above, the temperature at the bottom of the domain always seems to rise after a short while (shorter than the sound crossing time or RT growth time) although it should be in equilibrium. Here are the plots for the temperature cut on y direction (see page 6 for initial profile). The numbers and scaling seem to be correct – the heat flux and the thermal diffusion at the bottom should provide the exact support for the profile to sit there for a long time. It seems that the way the boundary condition is applied is not exactly correct since the rest of the domain sits still.