ZLS/Fyris User Notes

RSS

2009

0.1 Normalisation Convention

ZLS code usually normalises the simulation variables internally to optimise the numerical dynamic range with the typically limited floating point precision available (32 bit single or 64 bit double precision are typical). The convention is that a quantity is divided by a normalisation constant when input into the simulation and multiplied by the constant upon output. Usually input occurs once, and output may occur many times so the faster multiply operation is best on the output. e.g. if Q is a physical quantity value, Q_N is the internal normalised value, and f_N is the normalising constant, the following apply:

$$\begin{aligned} \text{Input } Q &\rightarrow Q/f_0 = Q_N \text{ Internal Normalised.} \\ \text{Output } Q_N &\rightarrow Q_N f_0 = Q \text{ External Value.} \end{aligned}$$

0.2 Fiducial Values

Using three physical quantities (4 for MHD, to include B) with a range of dimensions covering the basic physical units of time, mass, and length, normalisation constants can be derived. Typically we use density (ρ_0), length (x_0), and velocity (v_0) as fiducial quantities for typical simulations. So, $\rho_N = \rho/\rho_0$ is the internal normalised density value in the code. The constants for other quantities can be derived in terms of these three quantities simply on the basis of a MLT dimensional analysis.

$$\text{Density} : \rho_0 = \rho_0 \tag{1}$$
$$\text{Length} : x_0 = x_0 \tag{2}$$
$$\text{Velocity} : v_0 = v_0 \tag{3}$$
$$\text{Time} : t_0 = x_0/v_0 \tag{4}$$

$$\text{Acceleration} : a_0 = v_0/t_0 \quad (5)$$
$$\text{Pressure} : P_0 = \rho_0 v_0^2 \quad (6)$$
$$\text{Energy} : E_0 = P_0/x_0^3 \quad (7)$$
$$\text{Mass} : m_0 = \rho_0/x_0^3. \quad (8)$$

Other scaling factors include standard numerical constants for convenience. With gaussian (c.g.s.) MHD, a convention of adopting a magnetic $\mu_B = 4\pi$ gives a scaling for B,

$$\text{Magnetic Field} : B_0 = v_0(\mu_B \rho_0)^{1/2} \quad (9)$$

And the magnetic pressure is then simply B_N^2.

0.2.1 Temperature Normalisation

Temperature is treated differently, and is never used in a normalised way (i.e. T_N is not used). If a temperature is needed to look up a cooling rate for example, it is computed in Kelvin. A scaling factor, T_0 is used for this: $T(\mathrm{K}) = T_0 P_N/\rho_N$, but is applied the ratio of pressure and density which is a common construction in the simulations. T_0 then contains some additional constants to arrive at the absolute Kelvin scale.

So, $T_0 = v_0^2 \mu m/k$ where μ is the molecular weight (0.6 ionised, 2.3 molecular etc), m is the atomic mass unit ($1.66053873 \times 10^{-24}$ g), and k is Boltzmann's constant ($1.3806503 \times 10^{-24}$ erg K^{-1}).

0.2.2 Cooling Normalisation

The cooling function, Λ, is defined so that the volume emmissivity (or volume cooling rate, or simply cooling) L, is given by:

$$L = n^2 \Lambda \text{ erg cm}^{-3} \text{ s}^{-1}, \quad (10)$$

where n is the number density of particles in cm^{-3}. Therefore Λ has units of erg cm^3 s^{-1}. (note the +3 power for cm).

When working with mass densities is is convenient to use a density cooling function Λ_ρ.

$$L = \rho^2 \Lambda_\rho \text{ erg cm}^{-3} \text{ s}^{-1}, \quad (11)$$

So Λ_ρ has units of erg cm^3 s^{-1} g^{-2}. With μ and m these can be related:

$$\Lambda_\rho = \left(\frac{1}{\mu m}\right)^2 \Lambda \, . \quad (12)$$

In the code the cooling, L, is normalised; $L_N = \rho_N^2 \Lambda_{\rho,N}$. The normalisation for $\Lambda_{\rho,N} = \Lambda_\rho / \Lambda_{\rho,0}$ is

$$\Lambda_{\rho,0} = P_0/(\rho_0^2 t_0) \; . \tag{13}$$

However this may be evaluated frequently and the inverse is also used in the code: $\Lambda_{\rho,N} = \Lambda_\rho \mathrm{inv}\Lambda_{\rho,0}$ and

$$\mathrm{inv}\Lambda_{\rho,0} = \rho_0^2 t_0 / P_0 \; . \tag{14}$$

Finally the normalised cooling L_N needs to be restored on output to give the units desired, $L = L_N L_0$. L_0 is similar to $\Lambda_{\rho,0}$ without the density dependence, so in summary:

$$\text{Density Cooling Function} : \Lambda_{\rho,0} = P_0/(\rho_0^2 t_0). \tag{15}$$

$$\text{Inverse Cooling Function} : \mathrm{inv}\Lambda_{\rho,0} = \rho_0^2 t_0 / P_0. \tag{16}$$

$$\text{Cooling Rate} : L_0 = P_0/t_0. \tag{17}$$

0.3 Scaled Systems

For an adiabatic scale free simulation, x_0, ρ_0, and v_0 can be set equal to 1.0 and the internal and external values will be the same. This is the default case and is useful for comparing with standard shock tube models and Riemann problems in the literature.

For a c.g.s. system these can be set to values that are appropriate to the simulation in question. A local ISM model may have x_0 = 3.08e18 (cm), $v_0 = 1.0 \times 10^6$(cm/s), $\rho_0 = 1.0 \times 10^{-24}$ (g). Would give a characteristic timescale, t_0, of 3.08×10^{12} s ($\sim$100 000 yr). Input values can be in c.g.s. units, and are normalised by these values by the code. The grid can be specified in normalised unit, so a range of $0 - 1.0$ would represent 1 parsec.