Disclaimer: There's quite a bit of algebra in this document, that I mostly skipped to focus on the immediately usable results. I've tried my best to keep my steps clear, and I encourage you to go through the math once. Nevertheless, if something seems to appear "out of thin air", feel free to contact me.

Introduction

Consider gas with mass density ρ, temperature T, thermal energy density $e = \rho c_\mathrm{V} T$, velocity $\boldsymbol{u}$. The gas cools by coupling to the background radiation field with radiation energy density E, and heats up due to irradiation from a central star with luminosity $L_\star$. Within the flux-limited diffusion closure (FLD), which assumes an isotropic radiation field, the thermal evolution of the gas then follows:

$$\frac{\partial e}{\partial t} + \boldsymbol{u} \cdot \nabla e = -\gamma e \nabla \cdot \boldsymbol{u} - \kappa_\mathrm{P} \rho c \left(a_\mathrm{R} T^4 - E \right) + Q_\mathrm{irr}, \tag{1}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \boldsymbol{F}_\mathrm{rad} = \kappa_\mathrm{P} \rho c \left(a_\mathrm{R} T^4 - E \right), \qquad \boldsymbol{F}_\mathrm{rad} = -\frac{\lambda c}{\kappa_\mathrm{R} \rho} \nabla E. \tag{2}$$

Here, κ_R and κ_P are the Rosseland and Planck mean opacities, c the speed of light, λ the flux limiter, and Q_irr the irradiation heating term. We ignore the gray terms and focus on the e–E coupling.

Contents

1 Flux-limited diffusion

First, we'll describe how to solve Eqs. (1) & (2) with respect to e and E. Due to the very strict timestep limitation coming from the tern $\nabla \cdot \boldsymbol{F}_{\rm rad}$, we'll approach this problem with an *implicit* method. This allows us to maintain numerical stability regardless of timestep, but is much more computationally expensive.

1.1 Discretization

In the finite-volume framework, the evolved quantity is not what is defined at a point within a cell, but is rather its volume average within the cell. In other words, if q is a quantity defined within a cell, then we are solving for

$$\tilde{q} = \frac{1}{\Delta V} \int_{V_{\rm cell}} q \, \mathrm{d}V. \tag{3}$$

We can take advantage of this in Eq. (2). Integrating over a cell volume we get:

$$\begin{gathered} \frac{\partial}{\partial t} \int_{V_{\rm cell}} E \, \mathrm{d}V + \int_{V_{\rm cell}} \nabla \cdot \boldsymbol{F}_{\rm rad} \, \mathrm{d}V = \int_{V_{\rm cell}} \kappa_{\rm P} \rho c \left(a_{\rm R} T^4 - E \right) \mathrm{d}V \Rightarrow \\ \Delta V \frac{\partial \tilde{E}_{\rm rad}}{\partial t} + \oint_{S_{\rm cell}} \boldsymbol{F}_{\rm rad} \cdot \mathrm{d}\boldsymbol{s} \approx \Delta V \tilde{\kappa}_{\rm P} \tilde{\rho} c \left(a_{\rm R} \tilde{T}^4 - \tilde{E}_{\rm rad} \right), \end{gathered} \tag{4}$$

where we've used the divergence theorem to convert the volume integral to an integral over the cell surface $S_{\rm cell}$. For simplicity, from now on we'll drop the tilde (~) from all quantities and assume they refer to their volume-averaged counterpart. Since we're solving the set of equations (1) & (2) implicitly, we also assume that T and E are defined at $t + \Delta t$, denoted with a ($'$). We then have:

$$\rho c_{\rm V} \frac{T' - T}{\Delta t} = -\kappa_{\rm P} \rho c \left[a_{\rm R} (T')^4 - E' \right] + Q_{\rm irr}, \tag{5}$$

$$\frac{E' - E}{\Delta t} + \frac{1}{\Delta V} \sum_{S_{\rm cell}} \boldsymbol{F}'_{\rm rad} \cdot \Delta \boldsymbol{s} = \kappa_{\rm P} \rho c \left[a_{\rm R} (T')^4 - E' \right]. \tag{6}$$

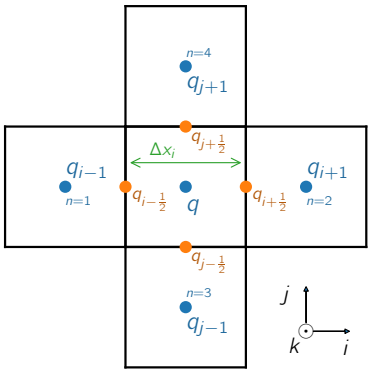

Fig. 1: Set of grid cells using our notation. The k direction extends out of the page.

We now focus on the flux term, $\sum \boldsymbol{F}'_{\rm rad} \cdot \Delta \boldsymbol{s}$. Discretizing this term is now much easier, as it's simply:

$$\sum \boldsymbol{F}'_{\rm rad} \cdot \Delta \boldsymbol{s} = -\sum_{n=1}^{N} \frac{\lambda c}{\kappa_{\rm R} \rho} \frac{\partial E'}{\partial x_n} \hat{x}_n \cdot \Delta \boldsymbol{s}_n, \tag{7}$$

summing over all $N = 6$ interfaces of the cell (see Fig. 1). All that remains is to evaluate $D = \frac{\lambda c}{\kappa_{\rm R} \rho}$ and $\frac{\partial E'}{\partial x_n}$ at their respective cell interfaces. We can do this by defining the Lagrange interpolating polynomial $\mathcal{L}$ between cells i and $i \pm 1$, and evaluating either $\mathcal{L}_{i \pm \frac{1}{2}}$ or $\frac{\partial \mathcal{L}}{\partial x_n}\Big|_{i \pm \frac{1}{2}}$. This gives us:

$$D_{i \pm \frac{1}{2}} \approx \frac{x_{i \pm 1} - x_{i \pm \frac{1}{2}}}{x_{i \pm 1} - x_i} D_i + \frac{x_{i \pm \frac{1}{2}} - x_i}{x_{i \pm 1} - x_i} D_{i \pm 1}, \quad \left. \frac{\partial E'}{\partial x_n} \right|_{i \pm \frac{1}{2}} \approx -\frac{1}{x_{i \pm 1} - x_i} E'_i + \frac{1}{x_{i \pm 1} - x_i} E'_{i \pm 1}. \tag{8}$$

We can now discretize the flux term, taking into account possible geometrical factors in different coordinate systems. For brevity, we will write that $\nabla E' \cdot \hat{x}_d = h_d \frac{\partial E'}{\partial x_d}$, where h_n is a geometrical factor in direction d (e.g., in cylindrical coordinates we have that $\nabla E' \cdot \hat{\varphi} = \frac{1}{R} \frac{\partial E'}{\partial \varphi}$, therefore $h_\phi = 1/R$). After applying Eq. (8) and repackaging everything, we find:

$$\frac{\Delta t}{\Delta V} \sum \boldsymbol{F}'_{\rm rad} \cdot \Delta \boldsymbol{s} = M E' + M_{i-1} E'_{i-1} + M_{i+1} E'_{i+1} + M_{j-1} E'_{j-1} + M_{j+1} E'_{j+1} + M_{k-1} E'_{k-1} + M_{k+1} E'_{k+1}, \tag{9}$$

where M are coefficients that depend on both physics (D) and the grid. For a direction $d \in \{i, j, k\}$:

$$M_{d\pm1} = \mp \frac{D_{d\pm\frac{1}{2}} h_d}{x_{d\pm1} - x_d} \frac{\Delta t}{\Delta V} \Delta s_{d\pm\frac{1}{2}}, \quad M = -\sum_{d\in\{i,j,k\}} (M_{d-1} + M_{d+1}) .$$

Note: This is not the final form of these coefficients! We need to include the RHS and boundaries.

We can then write Eq. (6) as:

$$E' - E + ME' + \sum_{d\in\{i,j,k\}} (M_{d-1}E'_{d-1} + M_{d+1}E'_{d+1}) = \kappa_{\mathrm{P}}\rho c\Delta t \left[a_{\mathrm{R}}(T')^4 - E'\right]. \tag{10}$$

Before compactifying this equation more, we'll need to work on that $(T')^4$ on the RHS.

1.1.1 *Bonus*: improved fluxes

Above, we used Lagrange polynomials to interpolate the radiation fluxes on cell interfaces. This works, but we can do better. We can demand that the flux exiting cell i from the right is equal to that entering cell $i + 1$ from the left. In other words:

$$F_{i+\frac{1}{2}} = F_i^{\mathrm{R}} = F_{i+1}^{\mathrm{L}} \Rightarrow -D_i \frac{E'_{i+\frac{1}{2}} - E'_i}{x_{i+\frac{1}{2}} - x_i} = -D_{i+1} \frac{E'_{i+1} - E'_{i+\frac{1}{2}}}{x_{i+1} - x_{i+\frac{1}{2}}} \Rightarrow D_i \frac{E'_{i+\frac{1}{2}} - E'_i}{\Delta x_i} = D_{i+1} \frac{E'_{i+1} - E'_{i+\frac{1}{2}}}{\Delta x_{i+1}}, \tag{11}$$

where $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is the cell width along direction i. Solving for $E'_{i+\frac{1}{2}}$ gives

$$E'_{i+\frac{1}{2}} = \frac{D_i E'_i/\Delta x_i + D_{i+1}E'_{i+1}/\Delta x_{i+1}}{D_i/\Delta x_i + D_{i+1}/\Delta x_{i+1}}. \tag{12}$$

and finally substituting $E'_{i+\frac{1}{2}}$ into $F_{i+\frac{1}{2}}$ yields

$$F_{i\pm\frac{1}{2}} = \mp \frac{2h_i}{\Delta x_i/D_i + \Delta x_{i\pm1}/D_{i\pm1}} (E'_{i\pm1} - E'_i) = \mp \tilde{D}_{i\pm1} h_i \frac{E'_{i\pm1} - E'_i}{\Delta x_i}. \tag{13}$$

The off-diagonal coefficients can then be written as

$$M_{d\pm1} = -\frac{\tilde{D}_{d\pm1} h_d}{\Delta x_i} \frac{\Delta t}{\Delta V} \Delta s_{d\pm\frac{1}{2}}, \qquad \frac{\tilde{D}_{d\pm1}}{\Delta x_i} = 2\left[\frac{\Delta x_i}{D_i} + \frac{\Delta x_{i\pm1}}{D_{i\pm1}}\right]^{-1}. \tag{14}$$

In doing so, we have achieved flux conservation *and* avoided having to interpolate any quantity onto the cell interface.

1.1.2 *Bonus*: geometrical factors

To avoid confusion, here's a list of the geometrical factors h for the three typical geometries:
Cartesian: $h_x = 1, \quad h_y = 1, \quad h_z = 1.$
Cylindrical: $h_R = 1, \quad h_\phi = \frac{1}{R}, \quad h_z = 1.$
Spherical: $h_r = 1, \quad h_\theta = \frac{1}{r}, \quad h_\phi = \frac{1}{r\sin\theta}.$

1.2 Radiation–thermal energy coupling

A common approach to expressing $(T')^4$ in terms of T and T' is to linearize it as

$$(T')^4 = (T + \Delta T)^4 \approx T^4 + 4T^3\Delta T = T^4 + 4T^3(T' - T) \Rightarrow (T')^4 \approx 4T^3T' - 3T^4. \tag{15}$$

Then, substitute Eq. (15) into Eq. (5), and obtain:

$$T' - T = -\frac{\kappa_\mathrm{P} c\Delta t}{c_\mathrm{V}}\left[a_\mathrm{R}\left(4T^3T' - 3T^4\right) - E'\right] + \frac{Q_\mathrm{irr}\Delta t}{\rho c_\mathrm{V}}. \tag{16}$$

To make our life easier we can define two dimensionless quantities:

$$Y = \kappa_\mathrm{P}\rho c\Delta t, \quad X = \frac{a_\mathrm{R}T^4}{e}Y = \frac{a_\mathrm{R}T^3\kappa_\mathrm{P}c\Delta t}{c_V}. \tag{17}$$

These quantities relate the distance traveled by a photon within Δt to its mean free path (Y), and the ratio of radiative to thermal energy density (X). Equation (16) then becomes

$$T' = \frac{1 + 3X}{1 + 4X}T + \frac{Y}{1 + 4X}\frac{E'}{\rho c_\mathrm{V}} + \frac{1}{1 + 4X}\frac{Q_\mathrm{irr}\Delta t}{\rho c_\mathrm{V}} \tag{18}$$

We will use this to update $T \to T'$ later, once we have computed E'. For now, we plug it back into Eq. (10) and get:

$$\left(1 + \frac{Y}{1 + 4X}\right)E' + ME' + \sum_{d\in\{i,j,k\}}\left(M_{d-1}E'_{d-1} + M_{d+1}E'_{d+1}\right) = E + \frac{Y}{1 + 4X}a_\mathrm{R}T^4 + \frac{4X}{1 + 4X}Q_\mathrm{irr}\Delta t. \tag{19}$$

This is starting to take shape. Next, we'll work on simplifying the equation before solving it.

1.3 Matrix form

If we express all quantities (e.g., E') as vectors spanning the entire grid, the coefficients M represent the *radiation matrix* $\overline{\mathrm{M}}$ (see Fig. 2). We can then write Eq. (19) as

$$\overline{\mathrm{M}} \cdot E' = B, \tag{20}$$

where B is the RHS, which depends on the current state:

$$B = E + \frac{Y}{1 + 4X}a_\mathrm{R}T^4 + \frac{4X}{1 + 4X}Q_\mathrm{irr}\Delta t \tag{21}$$

and the matrix elements of $\overline{\mathrm{M}}$ are given by:

$$\begin{aligned} M_{d\pm1} &= \mp\frac{D_{d\pm\frac{1}{2}}h_d}{x_{d\pm1} - x_d}\frac{\Delta t}{\Delta V}\Delta s_{d\pm\frac{1}{2}} \quad \text{or} \quad -\frac{\tilde{D}_{d\pm1}h_d}{\Delta x_i}\frac{\Delta t}{\Delta V}\Delta s_{d\pm\frac{1}{2}}, \\ M &= 1 + \frac{Y}{1 + 4X} - \sum_{d\in\{i,j,k\}}\left(M_{d-1} + M_{d+1}\right). \end{aligned} \tag{22}$$

The last step before solving the above system is taking care of boundary conditions.

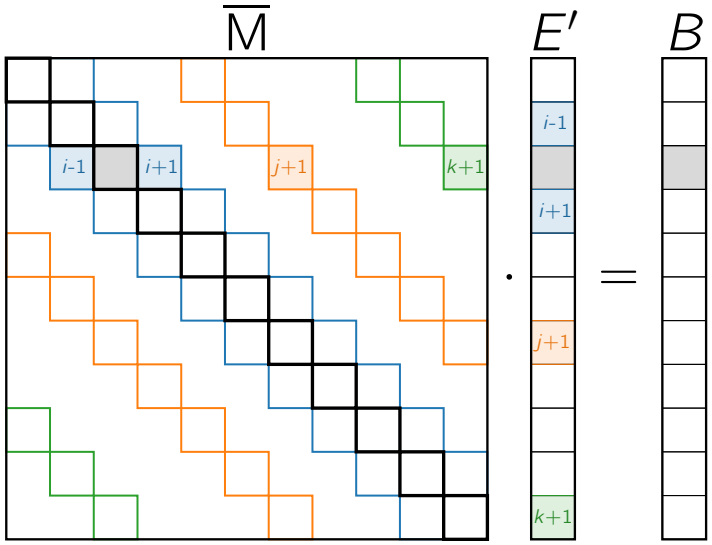

Fig. 2: Example of the linear system of equations $\overline{\mathrm{M}} \cdot E' = B$. The matrix $\overline{\mathrm{M}}$ is banded, with diagonal elements M and off-diagonals $M_{d\pm1}$. Shaded boxes denote all cells involved in computing the value of E' at a given point. Note that the off-diagonal bands are shorter than the diagonal, which will be addressed in the form of boundary conditions. In the case of the example cell shown, this is needed for cells $j-1$ and $k-1$. The gap between the blue and orange columns is $i_{\max}$ cells wide, and the orange–green gap $i_{\max} \times j_{\max}$ cells wide.

1.4 Boundary conditions

A clue to how we'll handle boundaries is in Fig. 2: for the example cell shown, the matrix element M_{k-1} is missing. In reality, it's been absorbed into the diagonal and/or the RHS.
Consider for example the case of a zero-gradient boundary. In the case of the above example, this implies $E'_{k-1} = E'$. We can then modify Eq. (19) as follows to include this information:

$$M \to M + M_{k-1}, \quad \text{then} \quad M_{k-1} = 0. \tag{23}$$

Similarly, a fixed-value boundary condition $E'_{k-1} = E_0$ would require the following modification:

$$B \to B - M_{k-1}E_0, \quad \text{then} \quad M_{k-1} = 0. \tag{24}$$

Finally, a periodic boundary would require that $E'_{k-1} = E'_{k_{\max}}$, which is handled by default as long as periodicity (or an MPI communication) is enforced before solving the equation system.

More sophisticated boundary conditions such as fixed-gradient or constant flux are of course possible, following the same logic as above. In general, if the target value/gradient is known *a priori* the information is passed to the RHS, while if it's relative to E' it is absorbed by the diagonal. The corresponding non-diagonal term is then eliminated.

Now that we have fully defined the above system, we can move on to solving it. Finally, once we've solved the system, we can update the thermal energy via Eq. (18), as $e' = \rho c_{\mathrm{V}} T'$.

1.5 Numerical solvers

Naively, one could use a typical linear algebra solver that aims to invert the matrix $\overline{\mathrm{M}}$ (e.g., Gauss–Seidel elimination). The problem with this approach is the size of the matrix, $N_{\text{cells}} \times N_{\text{cells}}$. With a typical grid containing anywhere between 10^4–10^{12} cells, this is impossible from both a time and memory standpoint.

We can instead note that the matrix is almost empty: only the 7 bands highlighted above (the diagonal, and a non-diagonal per side per orthogonal direction) are non-zero. The matrix $\overline{\mathrm{M}}$ is called *sparse*, with density $\sim 7/N_{\text{cells}} \ll 1$. This makes it a prime candidate for sparse matrix solvers. The approach here is iterative, with each iteration asymptotically approaching the exact solution. The solution is "reached" when a convergence criterion is met (e.g., once the residual $\sum_{\text{grid}} |\overline{\mathrm{M}} \cdot E' - B|$) is sufficiently small).

An entry-level (i.e., rather suboptimal but functional and easy to implement) method used to solve

sparse matrices is via *Successive Over-Relaxation* (SOR). By marking E' at a given iteration with superscript "p", $E'|^{p+1}$ is given by:

$$E'|^{p+1} = (1-\omega)\, E'|^{p} - \frac{\omega}{M}\left[\sum_{d\in\{i,j,k\}} \left(M_{d-1}\, E'_{d-1}\Big|^{p} + M_{d+1}\, E'_{d+1}\Big|^{p}\right) - B\right], \tag{25}$$

where $1 < \omega < 2$ is the relaxation parameter and $E'|^{p=0} = E$ is the initial guess. Since the method relies heavily on using an optimal value for ω, solving this system repeatedly (i.e., for many timesteps) would ideally require computing ω adaptively (e.g, adjusting ω slightly after every successful solution based on the number of iterations needed).
A "quirk" of the SOR method is that, since we update E' while sweeping through the grid, we need to be careful not to include terms $E'_n\Big|^{p+1}$ on the RHS. This can be achieved by doing two sweeps, hopping over every other grid cell in each sweep. In a 2D example, where the grid can be visualized as a checkerboard, this implies sweeping over all white cells first, then over all the black ones.

While SOR is quite easy to implement, it is far from the fastest. There exists an entire family of *gradient descent* solvers, specifically designed for solving problems with massive sparse matrices as quickly as possible. A popular example is the *BiConjugate Gradient—Stabilized* (BiCGSTAB) algorithm.

Note that some of these algorithms (e.g., the Conjugate Gradient *method) assume $\overline{M}$ to be symmetric, which is generally not the case. For the purpose of stability, it is advised to drop this assumption.*

Another note: A good initial guess can really help reduce the number of iterations needed for convergence. A typical starting point is the solution of the previous timestep, or $E = a_R T^4$ for step #1.

Another another note: The `scipy` *module for* `python` *contains a set of sparse matrix solvers that can solve Eq.* (20) *efficiently, as long as you provide $\overline{M}$ and B as discussed.*

1.5.1 *Bonus*: preconditioning

A good initial guess and an efficient solver are both very useful, but there is a third, sometimes significantly more important ingredient to quick convergence (or sometimes convergence at all): *preconditioning* the linear system. The idea is to manipulate $\overline{\mathrm{M}}$, E', and/or B in a way that favors the numerical scheme used to solve the system.

Here, we'll describe the simplest form of "Jacobi preconditioning", where we simply rescale the diagonal of $\overline{\mathrm{M}}$ to have a magnitude of 1. This is particularly useful in the optically thin limit ($Y \to 1$), where the diagonal is nearly zero and the system is prone to both roundoff errors and hundreds or thousands of iterations to convergence.

Having defined $\overline{\mathrm{M}}$ fully, we now define the diagonal matrix $\overline{\mathrm{P}} = P\bar{\mathrm{I}}$ with $P_i = \sqrt{|M_i|}$. Since it's diagonal, its inverse satisfies $P_i^{-1} = 1/P_i$. We will now apply $\overline{\mathrm{P}}$ to Eq. (20):

$$\overline{\mathrm{P}}^{-1}\,\overline{\mathrm{M}}\,\overline{\mathrm{P}}^{-1}\,\overline{\mathrm{P}}E' = \overline{\mathrm{P}}^{-1}B \Rightarrow \overline{\mathrm{m}}\cdot\varepsilon' = b, \tag{26}$$

where $\varepsilon'_i = P_i E'_i$, $b_i = B_i/P_i$, $m_i = M_i/P_i^2 = \mathrm{sgn}(M_i)$, and $m_{d\pm1} = M_{d\pm1}/(P_i P_{d\pm1})$. After solving this modified system for ε' (with the same methods as above), we can recover $E'_i = \varepsilon'_i/P_i$.

Note: it is recommended to handle the boundary conditions before applying the preconditioner to avoid complications.

2 Irradiation

We now shift our focus to a different problem: solving for the irradiation heating due to a central star. Here, we need to trace rays of starlight along the radial direction r:

$$Q_{\rm irr} = -\nabla \cdot (F_{\rm irr}\hat{r}), \qquad F_{\rm irr} = \frac{L_\star}{4\pi r^2} e^{-\tau}, \qquad \tau = \int_{R_\star}^{r} \kappa\rho \mathrm{d}r, \tag{27}$$

where τ is the optical depth and κ is the absorption opacity. The strategy here is to first compute τ throughout the domain (as this requires ray-tracing), evaluate $Q_{\rm irr}$, and finally update e.

The first step is easy (but expensive): starting from $\tau_{i=0} = 0$ (or τ_0, if the effect of a feature not included in the domain such as an inner disk must be captured), we iterate over r and set $\tau_{i+1} = \tau_i + \kappa_i\rho_i\Delta r_i$.

Note: in the case that we have multiple processors in the r direction, this operation can be parallelized! Each processor can compute its local $\delta\tau^p(r) = \int_{r^p_{min}}^{r} \kappa\rho\Delta r$*, and then all processors can communicate to trade "offsets" such that* $\tau^p(r) = \delta\tau^p(r) + \sum_{n<p} \delta\tau^n(r^n_{max})$.

2.1 Discretization

Similar to FLD, we'll use the divergence theorem and write:

$$\int_{V_{\rm cell}} Q_{\rm irr}\,\mathrm{d}V = -\int_{V_{\rm cell}} \nabla\cdot(F_{\rm irr}\hat{r})\,\mathrm{d}V \Rightarrow Q_{\rm irr} = \frac{1}{\Delta V}\left[F^{i-\frac{1}{2}}_{\rm irr}\Delta s_{i-\frac{1}{2}} - F^{i+\frac{1}{2}}_{\rm irr}\Delta s_{i+\frac{1}{2}}\right]. \tag{28}$$

Normally we would then follow an approach similar to Eq. (8), where we interpolate $F_{\rm irr}$ on the respective cell face. However, we can avoid this if we assume a spherical geometry (most commonly used for this problem, since the rays follow the grid naturally). Since the surface element is $\Delta s = r^2\Delta\Omega$, with $\Delta\Omega$ constant along a ray, we have:

$$Q_{\rm irr} = \frac{1}{\Delta V}\left[\frac{L_\star}{4\pi r^2_{i-\frac{1}{2}}} e^{-\tau_{i-\frac{1}{2}}} r^2_{i-\frac{1}{2}}\Delta\Omega - \frac{L_\star}{4\pi r^2_{i+\frac{1}{2}}} e^{-\tau_{i+\frac{1}{2}}} r^2_{i+\frac{1}{2}}\Delta\Omega\right] = \frac{L_\star}{4\pi\Delta V}\Delta\Omega\left[e^{-\tau_{i-\frac{1}{2}}} - e^{-\tau_{i+\frac{1}{2}}}\right]. \tag{29}$$

Finally, note that $\tau_{i+\frac{1}{2}} = \tau_{i-\frac{1}{2}} + \kappa\rho\Delta r$. Therefore:

$$Q_{\rm irr} = \frac{1}{\Delta V}\frac{L_\star}{4\pi r^2_{i-\frac{1}{2}}}\Delta s_{i-\frac{1}{2}} e^{-\tau_{i-\frac{1}{2}}}\left[1 - e^{-\kappa\rho\Delta r}\right] = \frac{F^{i-\frac{1}{2}}_{\rm irr}}{\Delta V}\Delta s_{i-\frac{1}{2}}\left[1 - e^{-\kappa\rho\Delta r}\right]. \tag{30}$$

Interpolating τ on the left cell interface is actually not necessary, as τ is *defined* on cell interfaces by virtue of it being an integral of $\kappa\rho$ over a cell width. In other words, the τ_i we defined above is actually $\tau_{i-\frac{1}{2}}$.

Note: when implementing the above numerically, consider using the function `expm1()` *(included in* `math.h` *in* `C`*, or equivalent). This returns* $e^x - 1$ *as a whole, rather than exposing us to numerical roundoff errors for very small* τ*, where* $e^{-\tau} \to 1$.

2.2 Updating the thermal energy

A simple update method would be a first-order accurate Euler step in an operator-split approach:

$$\frac{\partial e}{\partial t} = Q_{\rm irr} \Rightarrow e' = e + Q_{\rm irr}\Delta t, \tag{31}$$

assuming that all other terms (advection, radiation transport) have been handled already. The problem with this approach when used in conjuction with cooling (e.g., FLD) is that gas is sequentially heated/cooled depending on the order of operations. This makes computing an equilibrium state artificially harder. A solution is to couple irradiation heating and FLD into one step. As long as $Q_{\rm irr}$ does not depend on T, we can simply compute $Q_{\rm irr}$ and add it to the RHS in Eq. (21) and to the temperature update step in Eq. (18).

Note: if κ is a function of T, Q_{irr} can technically not be incorporated to the FLD problem above, as it would need to be computed "implicitly" (i.e., using T'). Nevertheless, the error due to incorporating it anyway is preferable and much smaller than the temperature difference due to successive heating and cooling.

2.2.1 *Bonus*: frequency-dependent irradiation

We can easily extend the above irradiation method to include a more sophisticated, frequency-dependent opacity component. For a frequency ν, we have that

$$F^{\nu}_{\rm irr} = \frac{L^{\nu}_{\star}}{4\pi r^2} e^{-\tau^{\nu}}, \qquad \tau^{\nu} = \kappa^{\nu} \int_{R_\star}^{r} \rho \mathrm{d}r = \kappa^{\nu} \sigma(r), \tag{32}$$

where $\sigma(r)$ is the column of material within radius r. We can then use ray-tracing to compute σ rather than τ, and use the appropriate κ^{ν} and $L^{\nu}_{\star}$ for each frequency bin. The total flux is then $F_{\rm irr} = \sum_{\nu} F^{\nu}_{\rm irr}$.

Of course, for consistency, our choices should satisfy that

$$L_{\star} = \sum_{\nu} L^{\nu}_{\star}, \qquad \kappa = \frac{\int \kappa^{\nu} B_{\nu}(T_{\star}) \mathrm{d}\nu}{\int B_{\nu}(T_{\star}) \mathrm{d}\nu} = \kappa_{\rm P}(T_{\star}), \tag{33}$$

where $B_{\nu}(T)$ is the black body radiance (i.e., the opacity can be replaced with a Planck mean opacity when integrated over all frequencies). Funny enough, frequency-dependent irradiation is sometimes more appropriate as κ_{ν} typically doesn't depend on T, *and* it's more accurate.

References

Levermore & Pomraning, 1981 • Kley, 1989 • Chiang & Goldreich, 1997 • Commercon et al., 2011 • Kuiper et al., 2013 • Kolb et al., 2013 • Robinson et al., 2024