Draft version May 7, 2024
Typeset using LaTeX default style in AASTeX62

A Coupled Hydrodynamics and Radiation Code in Python

Zachary L. Andalman,[1] Yue Pan,[1] and James Sunseri[1]

[1]*Department of Astrophysical Sciences, Princeton University, 4 Ivy Lane, Princeton, NJ, 08544, USA*

1. INTRODUCTION

Radiative transfer (RT) plays a crucial role in many astrophysical processes including but not limited to (i) the internal structure and evolution of stars, (ii) the outflows and emission produced by active galactic nuclei (AGN), (iii) the transport of energy in exoplanet atmospheres, and (iv) the formation of galaxies during the epoch of reionization. Such systems can be modeled using a post-processing technique, in which the RT equation is solved in a static snapshot of a hydrodynamics simulation. However, this approach neglects the ways in which the radiation field shapes the gas properties e.g., through radiation pressure. Alternatively, one can couple the RT equation to the equations of hydrodynamics to model the interplay between radiation and gas self-consistently. The field of solving the coupled equations are known as radiation hydrodynamics (RHD).

However, RHD is computationally expensive for several reasons. First, RT is a high-dimensional problem with space, angle, and frequency dimensions. Second, RHD requires accurate opacity models which may carry large uncertainties or not cover the required parameter space. Third, there is often a large separation between the timescales for radiative and hydrodynamical processes. Finally, the thermochemistry which couples radiation and hydrodynamics is complex, and usually only a small number of interactions can be modeled.

Approximations are required to make the problem computationally tractable. RHD methods can be divided into ray-based and moment-based methods. Ray-based methods directly model the propagation of light rays from radiation sources. Typically, they approximate the RT problem by discretizing the angular dimensions of the problem or randomly sampling the radiation field. Even with these approximations, the ray-based methods often too expensive because they scale with the number of radiation sources.

Moment-based methods ignore the angular dimensions entirely by taking angular moments of the RT equation, effectively modeling the radiation field as a fluid where individual beams are replaced with a bulk direction. Using moment-based methods, the RT equation reduces to a set of conservation laws, mirroring the form of the hydrodynamic equations. This the coupling to hydrodynamics simple to implement. However, moment-based methods make radiation diffusive, which is a poor approximation when the gas becomes optically thin. Furthermore, they cannot handle multiple beams of radiation moving in different directions in the same location, which occurs in situations where there are multiple discrete sources of radiation. Still, moment-based methods are often the only viable approach because the computational cost does not scale with the number of radiation sources.

As mentioned earlier, one of the key computational challenges is the separation between the timescales of radiative and hydrodynamic processes. Radiation travels at the speed of light, whereas hydrodynamic waves travel at most at the sonic speed. When using explicit methods in RHD, the timestep is limited by the light-crossing time of the grid cells to ensure stability, which can be prohibitively expensive. A common work-around is to reduce the speed of light to bring the timescales closer together.

In this work, we design an RHD code in `Python` which simultaneously solves the Euler equations, describing an adiabatic and inviscid flow, and the RT equation under M1 closure. We couple the hydrodynamics and radiation via thermochemistry. Our source code is available on github.

We apply our code to several problems, including the Kelvin-Helmholtz instability, the collision of radiation beams, and the ionization of a sphere by a radiation source, analogous to the ionization of the interstellar medium (ISM) by

a young star. We base our numerical approach off the `RAMSES-RT` cosmological RHD code (Rosdahl et al. 2013). In Section 2, we describe our numerical methods. In Section 3, we discuss our results. We conclude in Section 4.

2. NUMERICAL METHODS

We discretize the domain $\Omega = [0,1] \times [0,1]$ into $N \times N$ cells. In each cell, we store five hydrodynamic variables $\mathcal{U} = (\rho, p_x, p_y, E, \rho_{\rm ion})$ and three radiation variables $\mathcal{U}_{\rm rad} = (N_{\rm phot}, f_x, f_y)$. Each timestep, we first advance the Euler equations using a second-order Godunov method and the Harten-Lax-van Leer (HLL) Riemann solver with MonCen slope limiter and smooth extrema detection. Then, we advance the RT equation under M1 closure using a first-order Godunov method and the Global Lax Friedrich (GLF) Riemann solver. Finally, we couple the hydrodynamic and radiation variables via thermochemistry. Throughout this report, subscripts will denote spatial indices and superscripts will denote time indices, although we sometimes remove indices when it improves the clarity of the equations. Every time step taken in the simulation consists of 3 important substeps

1. Hydrodynamics

2. Radiative Transfer

3. Thermochemistry Coupling

Once all of these substeps have succesfully computed for a time step Δt then then next time step can be taken to evolve the system in time. In the following subsections we will explain each of these substeps in more detail.

2.1. *Hydrodynamics*

Our solver solves the set of Euler equations using a second-order finite volume scheme as described below.

2.1.1. *Euler equations*

The Euler equations are a set of coupled partial differential equations in density ρ, velocity $\mathbf{u}$, and pressure P which describe an adiabatic and inviscid flow. Using two dimensions and assuming a polytropic equation of state $P = \kappa\rho^\gamma$ with polytropic index γ, the Euler equations read

$$\frac{\partial \rho}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\rho + \rho\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \quad \text{mass conservation} \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} + \frac{\boldsymbol{\nabla}P}{\rho} = 0 \quad \text{momentum conservation} \tag{2}$$

$$\frac{\partial P}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}P + \gamma P\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \quad \text{energy conservation} \tag{3}$$

This form of the equations is a non-linear analog to a set of coupled linear partial differential equations, so it is called the quasi-linear form. By changing variables to density, momentum $\mathbf{p} = \rho\mathbf{u}$, and energy $E = P + \rho\|\mathbf{u}\|^2/2$, the Euler equations can be expressed as a set of conservation laws,

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \mathbf{p}\\ E\end{pmatrix} + \boldsymbol{\nabla}\cdot\begin{pmatrix}\mathbf{p}\\ \mathbf{p}\otimes\mathbf{u} + P\\ (E+P)\mathbf{u}\end{pmatrix} = 0 \tag{4}$$

In addition to the usual variables, we evolve the density of ionized gas $\rho_{\rm ion} = \rho x_{\rm HII}$ as a passive scalar, where x is the ionization fraction. The passive scalar is advected with the flow according to

$$\frac{\partial}{\partial t}\rho_{\rm ion} + \boldsymbol{\nabla}\cdot(\rho_{\rm ion}\mathbf{u}) = 0 \tag{5}$$

but only enters the physics in the thermochemistry calculations discussed in Section 2.3. We call $\mathcal{W} = (\rho, u_x, u_y, P, \rho_{\rm ion})$ the primative variable array and $\mathcal{U} = (\rho, p_x, p_y, E, \rho_{\rm ion})$ the conservative variable array. The argument of the divergence of the equations in the conservative form represents the flux $\mathcal{F}$ of the conserved variables.

2.1.2. *Numerical approach*

We compute the timestep according to the Courant–Friedrichs–Lewy (CFL) condition, which reads

$$\Delta t = \frac{C\Delta x}{\max_{i,j}\{|u_x|, |u_y|, c_\mathrm{s}, c\}} \tag{6}$$

where $c_\mathrm{s} = \sqrt{\gamma P/\rho}$ is the sound speed, c is the speed of light, and C is the Courant factor. Usually, the timestep is limited by the speed of light. However, in some contexts, the timestep restriction from the speed of light is too prohibitive, so we artificially lower the speed of light to reduce the computational cost. In these cases, it is necessary to account for a case that the timestep is limited by hydrodynamic waves.

The velocity and pressure solutions are typically smoother than the momentum and energy solutions due to discontinuities in the density near shocks. Our numerical methods are more accurate for smooth solutions, so we work with primative variables rather than conserved variables. Since we are using a second-order finite volume method, we approximate the solution within each cell as piecewise linear. In each dimension, we compute the slope of the piecewise linear solution as

$$(\Delta\mathcal{W}_\mathrm{L})_i = \mathcal{W}_i - \mathcal{W}_{i-1} \tag{7}$$
$$(\Delta\mathcal{W}_\mathrm{C})_i = (\mathcal{W}_{i+1} - \mathcal{W}_{i-1})/2 \tag{8}$$
$$(\Delta\mathcal{W}_\mathrm{L})_i = \mathcal{W}_{i+1} - \mathcal{W}_i \tag{9}$$

We compute the slope in each cell using the MonCen slope limiter as

$$\Delta\mathcal{W} = \min\{2\Delta\mathcal{W}_\mathrm{L}, \Delta\mathcal{W}_\mathrm{C}, 2\Delta\mathcal{W}_\mathrm{R}\} \tag{10}$$

The slope limiter prevents oscillations by ensuring that the solution at the cell interface never exceeds the mean solution in the neighboring cell. However, in some cases, the slope limiter may artificially prevent the solution from accurately capturing true extrema in the solution. Therefore, we implement a smooth extrema detector which turns off the slope limiter when an extrema is detected. In each cell, we compute

$$\alpha_\mathrm{L,R} = \begin{cases} \min\left\{1, \frac{\min\{2\Delta\mathcal{W}_\mathrm{L,R}, 0\}}{\Delta\mathcal{W}_\mathrm{C}}\right\} & \Delta\mathcal{W}_\mathrm{C} < 0 \\ 1 & \Delta\mathcal{W}_\mathrm{C} = 0 \\ \min\left\{1, \frac{\max\{2\Delta\mathcal{W}_\mathrm{L,R}, 0\}}{\Delta\mathcal{W}_\mathrm{C}}\right\} & \Delta\mathcal{W}_\mathrm{C} > 0 \end{cases} \tag{11}$$

We define a smoothness indicator in each cell as

$$\alpha = \min\{\alpha_\mathrm{L}, \alpha_\mathrm{R}\} \tag{12}$$

We apply the slope limiter in each dimension in each cell only if

$$\min\{\alpha_{i-1}, \alpha_i, \alpha_{i+1}\} < 1 \tag{13}$$

When the slope limiter is not applied, we simply use the second-order cell-centered slope.

For consistency with the second-order finite difference kernel, we use a second order time integrator. However, the second-order Runge Kutta time integrator has been shown to cause high dissipation. Instead, we use the MUSCL-Hancock scheme, which relaxes the requirement of having a conservative predictor scheme. In addition to improving accuracy, this approach improves efficiency by halving the number of calls to the Riemann solver. For the predictor

step, we solve the quasi-linear form of the Euler equations using the slope, giving

$$\rho^{n+1/2} = \rho^n - \frac{\Delta t}{2\Delta x}\left[u_x^n \Delta_x \rho^n + u_y^n \Delta_y \rho^n + \rho^n(\Delta_x u_x^n + \Delta_y u_y^n)\right] \tag{14}$$

$$u_x^{n+1/2} = u_x^n - \frac{\Delta t}{2\Delta x}\left[u_x^n \Delta_x u_x^n + u_y^n \Delta_y u_x^n + \frac{\Delta_x P^n}{\rho^n}\right] \tag{15}$$

$$u_y^{n+1/2} = u_y^n - \frac{\Delta t}{2\Delta x}\left[u_x^n \Delta_x u_y^n + u_y^n \Delta_y u_y^n + \frac{\Delta_y P^n}{\rho^n}\right] \tag{16}$$

$$P^{n+1/2} = P^n - \frac{\Delta t}{2\Delta x}\left[u_x^n \Delta_x P^n + u_y^n \Delta_y P^n + \gamma P^n(\Delta_x u_x^n + \Delta_y u_y^n)\right] \tag{17}$$

$$\rho_{\rm ion}^{n+1/2} = \rho_{\rm ion}^n - \frac{\Delta t}{2\Delta x}\left[u_x^n \Delta_x \rho_{\rm ion}^n + u_y^n \Delta_y \rho_{\rm ion}^n + \rho_{\rm ion}^n(\Delta_x u_x^n + \Delta_y u_y^n)\right] \tag{18}$$

Next, we use the slope to extrapolate the solution to the cell interfaces in each cell and each dimension, giving

$$\mathcal{W}_{i\pm1/2}^{n+1/2} = \mathcal{W}_i^{n+1/2} \pm \Delta\mathcal{W}_i^{n+1/2}/2$$

At each interface, we now have a left state $\mathcal{W}_{\rm L}$ and a right state $\mathcal{W}_{\rm R}$. We use the HLL Riemann solver the compute the resulting flux at the interface.

The HLL Riemann solver assumes that the Riemann solution is piecewise constant around the interface with three regions: the left unperturbed state $\mathcal{W}_{\rm L}$, the right unperturbed state $\mathcal{W}_{\rm R}$, and a central state $\mathcal{W}_*$. The regions are separated by left and right moving waves. The wavespeeds are computed as

$$S_{\rm L} = \min\{v_{\rm L}, v_{\rm R}\} - \max\{c_{\rm s,L}, c_{\rm s,R}\} \tag{19}$$

$$S_{\rm R} = \max\{v_{\rm L}, v_{\rm R}\} + \max\{c_{\rm s,L}, c_{\rm s,R}\} \tag{20}$$

We compute the conserved variables $\mathcal{U}_{\rm L}$ and $\mathcal{U}_{\rm R}$ at the left and right states respectively. Then, using the conservative form of the Euler equations, we compute the fluxes $\mathcal{F}_{\rm L}$ and $\mathcal{F}_{\rm R}$ of the left and right states respectively. It can be shown that the flux of the central state is

$$\mathcal{F}_* = \frac{\mathcal{F}_{\rm L} S_{\rm R} - \mathcal{F}_{\rm R} S_{\rm L}}{S_{\rm R} - S_{\rm L}} + \frac{S_{\rm R} S_{\rm L}}{S_{\rm R} - S_{\rm L}}(\mathcal{U}_{\rm R} - \mathcal{U}_{\rm L}) \tag{21}$$

The flux is selected from these three states according to

$$\mathcal{F} = \begin{cases} \mathcal{F}_L & S_{\rm L} > 0 \text{ and } S_{\rm R} > 0 \quad \text{supersonic to the right} \\ \mathcal{F}_R & S_{\rm L} < 0 \text{ and } S_{\rm R} < 0 \quad \text{supersonic to the left} \\ \mathcal{F}_* & S_{\rm L} < 0 \text{ and } S_{\rm R} > 0 \quad \text{subsonic} \end{cases} \tag{22}$$

Finally, we evolve the conserved variables using the fluxes.

$$\mathcal{U}_{i,j}^{n+1} = \mathcal{U}_{i,j}^n + \frac{\Delta t}{\Delta x}\left(\mathcal{F}_{i-1/2,j}^{n+1/2} - \mathcal{F}_{i+1/2,j}^{n+1/2} + \mathcal{F}_{i,j-1/2}^{n+1/2} - \mathcal{F}_{i,j+1/2}^{n+1/2}\right) \tag{23}$$

2.2. *Radiation transport*

In this section we describe the M1 Closure scheme used to simulate radiative transfer in our volume.

2.2.1. *Radiation transport equation*

The radiation field is described by the specific radiation intensity, defined as the energy of the radiation field per unit time per unit area per unit frequency per unit solid angle, or in mathematical notation,

$$\mathrm{d}E = I_\nu \mathrm{d}t \mathrm{d}A \mathrm{d}\nu \mathrm{d}\Omega \tag{24}$$

The photon number N_ν, photon flux $\mathbf{F}_\nu$, and radiation pressure tensor are defined via angular moments of the specific radiation intensity:

$$N_\nu = \frac{1}{h\nu}\int \mathrm{d}\Omega \frac{I_\nu}{c}, \quad \mathbf{F}_\nu = \frac{1}{h\nu}\int \mathrm{d}\Omega I_\nu \mathbf{n}, \quad \mathbb{P} = \int \mathrm{d}\Omega \frac{I_\nu}{c}\mathbf{n}\otimes\mathbf{n} \tag{25}$$

The local change in the specific intensity of radiation I_ν is given by the RT equation, which reads (Mihalas & Mihalas 1984)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}\cdot\nabla I_\nu = -\kappa_\nu I_\nu + \eta_\nu \tag{26}$$

where κ_ν is the absorption coefficient, η_ν is a source function, and $\mathbf{n}$ is the direction of propagation. Taking the zeroth and first angular moments of the RT equation, we find

$$\frac{\partial N_\nu}{\partial t} + \nabla\cdot\mathbf{F}_\nu = -\kappa_\nu c N_\nu + \frac{4\pi\eta_\nu}{h\nu} \tag{27}$$

$$\frac{\partial \mathbf{F}_\nu}{\partial t} + c^2\nabla\cdot\mathbb{P}_\nu = -\kappa_\nu c\mathbf{F}_\nu \tag{28}$$

The absorption coefficient can be divided into contributions from each photo-absorbing species, giving $\kappa_\nu = \sum_j n_j\sigma_{\nu j}$, where n_j is the number density of species j. The source term can be divided into contributions from recombination radiation from gas $\dot{N}_{\mathrm{rec},\nu}$ and injection sources $\dot{N}_{\mathrm{inj},\nu}$, such as stars are quasars in the astrophysical context. We make two major approximations in our numerical approach. First, we average the RT equation over frequency, removing all frequency dependencies. Second, we only consider a single photo-absorbing species, neutral hydrogen. Under these approximations, Eqs. 27 and 28 become

$$\frac{\partial N_{\mathrm{phot}}}{\partial t} + \nabla\cdot\mathbf{F} = -(1-x_{\mathrm{HII}})n_{\mathrm{H}}\sigma + \dot{N}_{\mathrm{inj}} + \dot{N}_{\mathrm{rec}} \tag{29}$$

$$\frac{\partial \mathbf{F}}{\partial t} + c^2\nabla\cdot\mathbb{P} = -(1-x_{\mathrm{HII}})n_{\mathrm{H}}\sigma c\mathbf{F} \tag{30}$$

To close Eq. 27 and 28, we must choose a specific form for the pressure tensor. We choose the M1 closure relation, which has the advantage of being purely local. Under the M1 closure relation, the pressure tensor is given by $\mathbb{P} = \mathbb{D}N_{\mathrm{phot}}$, where the Eddington tensor $\mathbb{D}$ is

$$\mathbb{D} = \frac{1-\chi}{2}\mathbf{I} + \frac{3\chi-1}{2}\mathbf{n}\otimes\mathbf{n} \tag{31}$$

and

$$\mathbf{n} = \frac{\mathbf{F}}{|\mathbf{F}|}, \quad \chi = \frac{3+4f^2}{5+2\sqrt{4-3f^2}}, \quad f = \frac{|\mathbf{F}|}{cN_{\mathrm{phot}}} \tag{32}$$

are the unit vector pointing in the flux direction, the Eddington factor, and the reduced flux respectively. A low reduced flux means the radiation is predominantly isotropic, and a high Eddington factor means that it is predominantly flowing in one direction, as expected from a single source in an optically thin medium. The M1 closure relation therefore interpolates between the optically thick and optically thin limits.

2.2.2. *Photon injection and transport*

We use an operator splitting approach to break the solution into three steps. In the injection step, we inject photons into the grid, solving $\partial N_{\mathrm{phot}}/\partial t = \dot{N}_{\mathrm{inj}}$. In the transport step, we propagate the photons through space neglecting the absorption and source terms, solving the RT equation with the RHS equal to zero. In the thermochemistry step, we couple the photons to the gas, solving the full RT equations. This step is described in detail in Section 2.3. In the injection step, we use the forward Euler scheme, giving

$$N_{\mathrm{phot}}^{n+1} = N_{\mathrm{phot}}^{n} + \dot{N}_{\mathrm{inj}}\Delta t \tag{33}$$

In the transport step, we use the same timestep used for the Euler equations (see Section 2.1.2). The modified RT equation can be expressed as a set of conservation laws,

$$\frac{\partial}{\partial t}\begin{pmatrix} N_{\rm phot} \\ \mathbf{F} \end{pmatrix} + \boldsymbol{\nabla}\cdot\begin{pmatrix} \mathbf{F} \\ c^2\mathbb{P} \end{pmatrix} = 0 \tag{34}$$

We call $\mathcal{U}_{\rm rad} = (N_{\rm phot}, F_x, F_y)$ the radiation variable array. The argument of the divergence of the equations in the conservative form represents the flux $\mathcal{F}_{\rm rad}$ of the conserved variables.

Since we are using a first-order finite volume method, we approximate the solution within each cell as piecewise constant. At each interface, we now have a left state $\mathcal{U}_{\rm rad,L}$ and a right state $\mathcal{U}_{\rm rad,R}$. Using the conservative form of the RT equation, we compute the fluxes $\mathcal{F}_{\rm rad,L}$ and $\mathcal{F}_{\rm rad,R}$ of the left and right states respectively. The GLF Riemann solver gives the flux at the interface as

$$\mathcal{F}_{\rm rad} = \frac{\mathcal{F}_{\rm rad,L} + \mathcal{F}_{\rm rad,R}}{2} - \frac{c}{2}(\mathcal{U}_{\rm rad,R} - \mathcal{U}_{\rm rad,L}) \tag{35}$$

Finally, we evolve the radiation variables using the fluxes, giving

$$(\mathcal{U}_{\rm rad})^{n+1}_{i,j} = (\mathcal{U}_{\rm rad})^{n}_{i,j} + \frac{\Delta t}{\Delta x}\left((\mathcal{F}_{\rm rad})^{n}_{i-1/2,j} - (\mathcal{F}_{\rm rad})^{n}_{i+1/2,j} + (\mathcal{F}_{\rm rad})^{n}_{i,j-1/2} - (\mathcal{F}_{\rm rad})^{n}_{i,j+1/2}\right) \tag{36}$$

2.3. *Thermochemistry*

Once the hydrodynamics solver has taken a time step and the radiative transfer solver has also taken the time step, then we need to couple them via thermochemistry. For the sake of simplicity, in our simulation we assume only 2 chemical species, neutral hydrogen HI and ionized hydrogen HII. The thermochemistry equations we solve are

$$\frac{\partial N_{\rm phot}}{\partial t} = -n_{\rm HI} c \sigma^N_{\rm HI} N_{\rm phot} \tag{37}$$

$$\frac{\partial \mathbf{F}}{\partial t} = -n_{\rm HI} c \sigma^N_{\rm HI} \mathbf{F} \tag{38}$$

$$\frac{\partial \epsilon}{\partial t} = \mathcal{H} + \mathcal{L} \tag{39}$$

$$n_{\rm H}\frac{\partial x_{\rm HII}}{\partial t} = n_{\rm HI}\left(\beta_{\rm HI} n_e + \sigma^N_{\rm HI} c N_{\rm phot}\right) - n_{\rm HII}\alpha^B_{\rm HII} n_e \ , \tag{40}$$

where $N_{\rm phot}$ is the photon number density, $n_{\rm HI}$ is the number density of HI, c is the speed of light, $\sigma^N_{\rm HI}$ is the cross section of HI neutral hydrogen, $\mathbf{F}$ is the photon flux, ϵ is the thermal energy density of the gas, $\mathcal{H}$ is the heating rate, $\mathcal{L}$ is the cooling rate, n_H is the number density of hydrogen, $x_{\rm HII} \equiv n_{\rm HII}/n_H$ is the hydrogen ionization fraction, $\beta_{\rm HI}(T)$ is the rate of collisional ionization, n_e is the electron number density, and $\alpha^B_{\rm HII}$ is the HII recombination coefficient.

Given these coupled differential equations, we need to solve for the new thermochemical variables $\mathcal{U}_T = (\epsilon, x_{\rm HII}, N, \mathbf{F})$ which we use to update the stored state variables $\mathcal{U} = (\rho, \rho\mathbf{u}, E, \rho_{\rm ion}, N, \mathbf{F})$. We take the same semi-implicit approach as `RAMSES-RT` where we use any updated quantities we have access to in a given time step. The order in which we solve the necessary differential equations is described in the following substep updates

1. Photon Density and Flux Update

2. Thermal Update

3. Ionization Fraction Update

The reason why we first update the photon density is this quantity proves to be the most unstable one. We describe each of these substeps below.

2.3.1. *Photon Density and Flux Update*

The photon number density and fluxes N and $\mathbf{F}$ are updated using the on-the-spot approximation (OTSA) which gives us that

$$N_i^{t+\Delta t} = \frac{N_i^t}{1+\Delta t D} \tag{41}$$

$$\mathbf{F}_i^{t+\Delta t} = \frac{\mathbf{F}_i^t}{1+\Delta t D} \ , \tag{42}$$

where D is the photon-destroying absorption coefficient given by

$$D = c_r \sigma_{\rm HI}^N n_{\rm HI} \ . \tag{43}$$

We ignore the photon-creating re-combination term C because we are using the OTSA. We make sure to follow the 10% rule by checking if

$$\frac{|N_i^{t+\Delta t} - N_i^t|}{N_i^t} > 0.1 \tag{44}$$

is true. If this condition is not met, then we restart the entire thermochemistry update with half of the time step $\Delta t_{\rm new} = \Delta t/2$.

2.3.2. *Thermal Update*

In this substep we are updating the energy of the system by evolving the temperature of our gas. More specifically, we evolve the quantity

$$T_\mu \equiv \frac{T}{\mu} \tag{45}$$

where T is the temperature and μ is

$$\mu = \left[X(1+x_{\rm HII})\right]^{-1} \ . \tag{46}$$

where X is the primordial hydrogen mass fraction which for our project will just be unity. We ignore Helium and other heavier elements for simplicity as stated before. The equation we need to solve to update the temperature is

$$\frac{\partial T_\mu}{\partial t} = K\Lambda \ , \tag{47}$$

where

$$K \equiv \frac{(\gamma-1)m_{\rm H}}{\rho k_{\rm B}} \ , \tag{48}$$

and $\Lambda \equiv \mathcal{H} + \mathcal{L}$. The heating function is given by

$$\mathcal{H} = n_{\rm HI} c_r N_{\rm phot} \left(\bar{\epsilon}_\gamma \sigma_{\rm HI}^E - \epsilon_{\rm HI}\sigma_{\rm HI}^N\right) \ , \tag{49}$$

and the cooling function is given by

$$\mathcal{L} = \left[\zeta_{\rm HI}(T) + \psi_{\rm HI}(T)\right] n_e n_{\rm HI} \tag{50}$$

$$+ \eta_{\rm HII}^B(T) n_e n_{\rm HII} \tag{51}$$

$$+ \theta(T) n_e n_{\rm HII} \tag{52}$$

$$+ \varpi(T) n_e \ . \tag{53}$$

Here we denote the various cooling process functions to be: collisional ionizations ζ, collisional excitations ψ, recombinations η, bremsstrahlung θ, and Compton cooling ϖ. All these functions are analytic and pulled from Rosdahl et al. (2013) Appendix E. Throughout this work we use OTSA for ionization to simplify the original expressions from Rosdahl et al. (2013). Using all of this, then we can write down what our semi-implicit scheme should be to compute the next iteration of the reduced temperature

$$T_\mu^{t+\Delta t} = T_\mu^t + \frac{\Lambda K \Delta t}{1 - \Lambda' K \Delta t} \ . \tag{54}$$

We define $\Lambda' \equiv \mu \frac{\partial \mathcal{L}}{\partial T}$ which is acquired by analytically computing the derivative of $\mathcal{L}$. What makes this scheme semi-implicit is that it uses the updated values of N and $\mathbf{F}$ to compute the update similar to how a full implicit scheme would do, except we are also using state variables that havent been updated yet as well. As done in the photon update, we have to also adhere to the 10% rule

$$\frac{|T_\mu^{t+\Delta t} - T_\mu^t|}{T_\mu^t} > 0.1 \ , \tag{55}$$

and as done before, we restart the entire thermochemistry step with half of the Δt if the 10% rule is not followed. We also apply a second condition to ensure stability that reduces oscillations in the temperature

$$\frac{|K\Lambda\Delta t|}{T_\mu^t} > 0.1. \tag{56}$$

If this condition is not also met, then we halve the time step as well and try restart the thermochemistry step.

2.3.3. *Ionization Fraction Update*

The equation that governs how the ionization fraction changes is given by

$$\frac{\partial x_{\rm HII}}{\partial t} = (1 - x_{\rm HII}) \left[\beta_{\rm HII} n_e + \sigma_{\rm HI}^N c_r N\right] - x_{\rm HII} \alpha_{\rm HII}^B n_e \tag{57}$$

$$= (1 - x_{\rm HII}) C - x_{\rm HII} D \tag{58}$$

$$= C - x_{\rm HII}(C + D) \ , \tag{59}$$

where C represents the HII creation and D represents the destruction. Our semi-implicit scheme then is

$$x_{\rm HII}^{t+\Delta t} = x_{\rm HII}^t + \Delta t \frac{C - x_{\rm HII}^t (C + D)}{1 - J\Delta t} \ , \tag{60}$$

where J is defined as

$$J \equiv \frac{\partial \dot{x}_{\rm HII}}{\partial x_{\rm HII}} = \frac{\partial C}{\partial x_{\rm HII}} - (C + D) - x_{\rm HII} \left(\frac{\partial C}{\partial x_{\rm HII}} + \frac{\partial D}{\partial x_{\rm HII}} \right) \ . \tag{61}$$

The creation and destruction derivatives are given by

$$\frac{\partial C}{\partial x_{\rm HII}} = n_{\rm H} \beta_{\rm HI} - n_e T_\mu \mu^2 X \frac{\partial \beta_{\rm HI}}{\partial T} \tag{62}$$

$$\frac{\partial D}{\partial x_{\rm HII}} = n_{\rm H} \alpha_{\rm HII}^B - n_e T_\mu \mu^2 X \frac{\partial \alpha_{\rm HII}^B}{\partial T} \ . \tag{63}$$

We follow the same 10% rule once again, requiring that

$$\frac{|x_{\rm HII}^{t+\Delta t} - x_{\rm HII}^t|}{x_{\rm HII}^t} > 0.1 \ , \tag{64}$$

be true. We also have a secondary first-order condition similar to the thermal update

$$\frac{C - x_{\rm HII}^t (C + D)}{x_{\rm HII}^t} \Delta t > 0.1 \ . \tag{65}$$

If either of these conditions are not met, we halve the time step and restart the entire thermochemistry update step. Depending on however many halvings $n_{\rm h}$ occurred in the process of computing the thermochemistry step, we have to take $2^{n_{\rm h}}$ thermochemistry steps to reach the same Δt size taken by the radiative transfer solver and hydro solver.

3. RESULTS

We test each part of our code individually. To test the hydrodynamics solver, we use the Kelvin-Helmholtz instability. To test the radiation transport solver, we use the collision of radiation beams. To test the thermochemistry solver, we use the ionization of a sphere by a radiation source, analogous to the ionization of the ISM by a young star. For all tests, we use a Courant factor of $C = 0.4$.

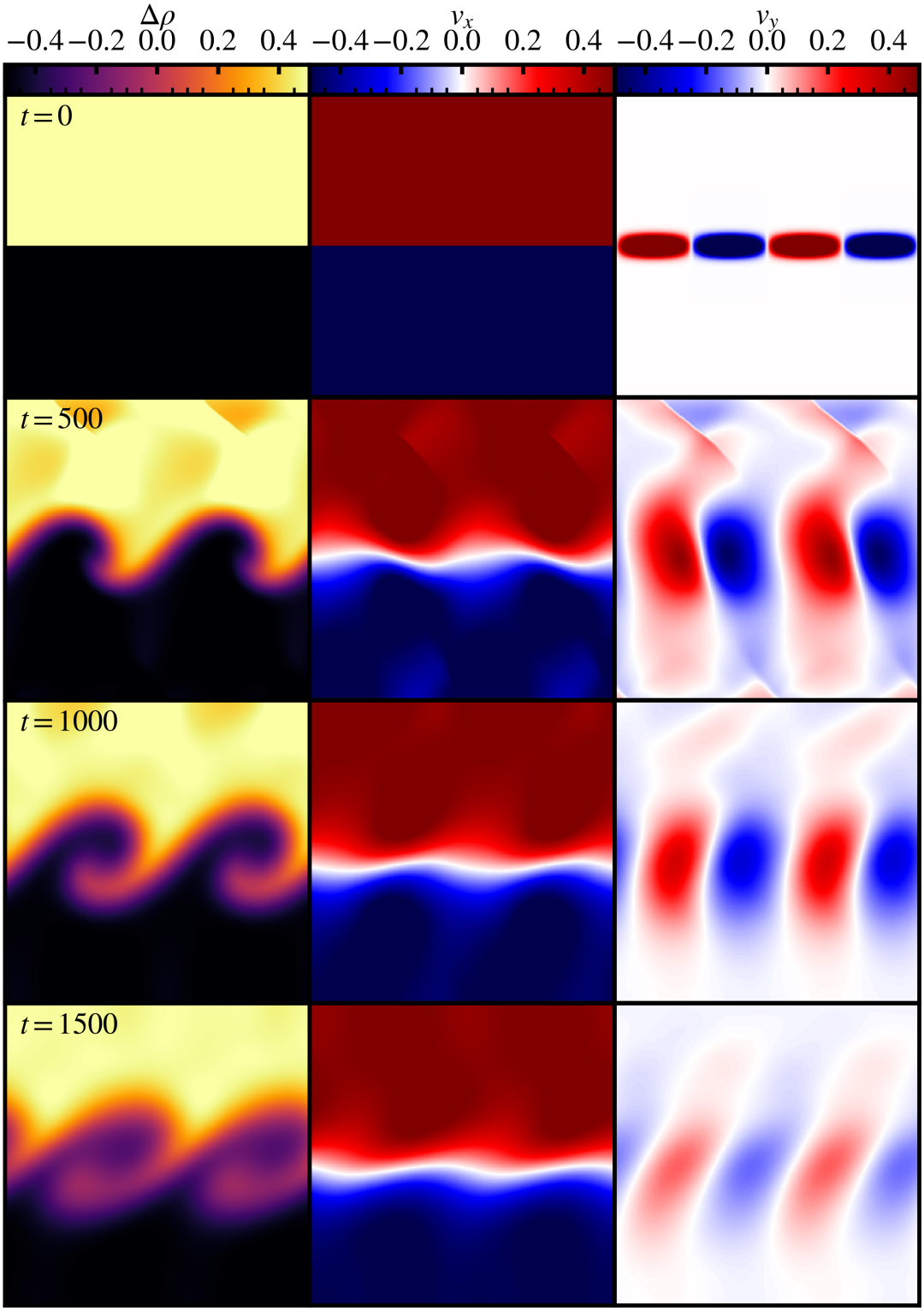

Figure 1. Density perturbation (left column), x-velocity (middle column), and y-velocity in four time slices of a simulation demonstrating the Kelvin-Helmholtz instability.

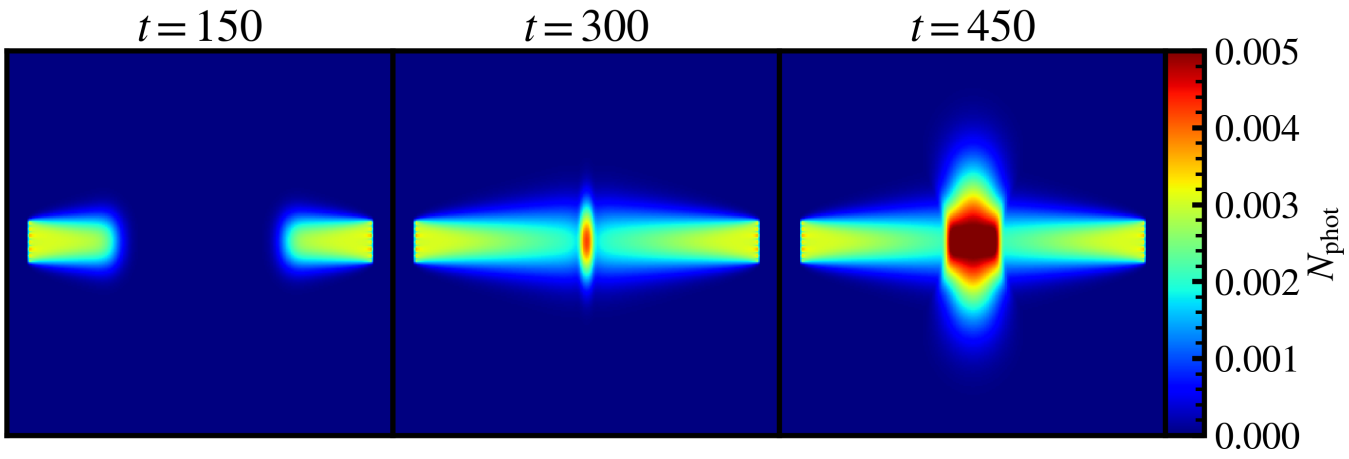

Figure 2. Photon number in three time slices of a simulation demonstrating the collision of two radiation beams in the M1 closure scheme.

The Kelvin-Helmholtz instability occurs when there is a velocity difference across the interface between two fluids. We run a simulation with two fluids of different densities and velocities, with small perturbations in the velocity of the fluid perpendicular to the interface. The initial conditions are described by

$$\begin{aligned} \rho^0 &= H(y - 0.5) + 1 \\ v_x^0 &= H(y - 0.5) - 0.5 \\ v_y^0 &= 0.1 \sin(4\pi x)\mathcal{N}(0.5, 0.02) \\ P^0 &= 3 \end{aligned} \tag{66}$$

where H is the Heaviside step function and $\mathcal{N}$ is a Gaussian. We use an $N = 256$ grid with periodic boundary conditions in the x-direction and Neumann boundary conditions in the y-direction. We show the results in Figure 1. A movie of the simulation can be found on github.

A Kelvin-Helmholtz instability indeed arises, as seen by the swirling patterns which form around the fluid interface. Additionally, a shock wave is launched in the denser fluid, where the velocity is close to the sound speed. However, our scheme suffers from numerical diffusion. Towards the end of the simulation, the details of the perturbation are diffused away. The numerical diffusion could be improved by switching to a less diffusive Riemann solver, such as HLLC, or by increasing the resolution of the simulation.

In reality, two intersecting electromagnetic waves can pass through one another. One weaknesses of the M1 closure scheme is that the radiation field within each cell can only have one flux direction. When two beams of radiation collide, the conservation laws force flux to be generated in a direction perpendicular to the collision. This makes the M1 closure scheme a poor approximation in scenarios when there are several discrete sources of radiation, but reasonable when the radiation field is dominated by a single source or by diffuse emission regions.

We run a simulation with two beams of radiation colliding in the center of the simulation box. Each timestep, we inject photons on the left side of the simulation box with a high positive x-flux and on the right side of the simulation with a high negative x-flux. We use an $N = 256$ grid with Neumann boundary conditions in both directions. We show the results in Figure 2. A movie of the simulation can be found on github.

When the radiation beams collide, flux is indeed generated in the perpendicular direction, as expected from the M1 closure scheme. Furthermore, the radiation beams remain well-collimated as they propagate through space, although some diffusion of photons can be seen on the outskirts of the beams.

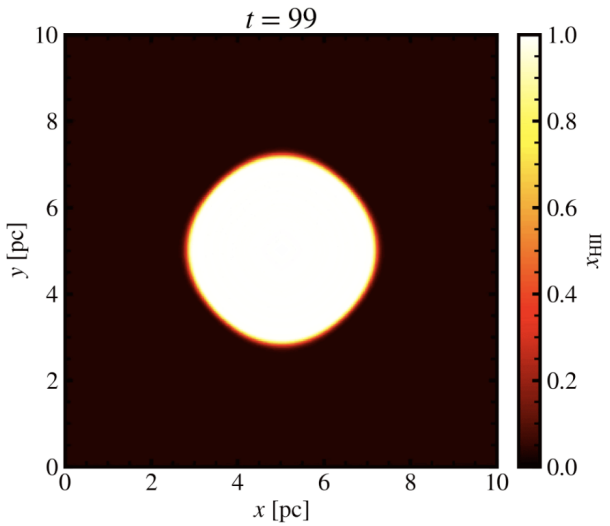

Figure 3. Hydrogen ionization fraction of an ionized region around a central source embedded in a uniform neutral medium at one snapshot of the simulation. This snapshot corresponds to the 100th time step taken. The expansion of the ionization front is clearly seen. We stop the simulation after the ionization bubble has reached $\sim r_{\rm S}$

Very massive stars emit significant amounts of ionizing radiation. Because stars are born in neutral hydrogen, the star creates a region of ionized hydrogen known as a Strömgren sphere. The radius of the sphere is set by an equilibrium between ionization and recombination, which itself depends on the density. Specifically, the Strömgren radius is

$$r_{\rm S} = \left(\frac{3}{4\pi} \frac{Q}{n_{\rm H}^2 \alpha^{\rm B}} \right)^{1/3} \tag{67}$$

where Q is the rate at which the star emits ionizing photons, $n_{\rm H}$ is the number density of hydrogen, and $\alpha^{\rm B}$ is the case-B recombination rate as a function of temperature.

We test our coupled RHD solver by simulating the formation of a Strömgren sphere. We consider typical conditions inside a molecular cloud in the Milky Way referring to Rosdahl et al. (2013). Specifically, we set $n_{\rm H} = 10^2$ cm^{-3} and $T = 100$ K. The pressure can be determined from the temperature assuming an ideal gas equation of state, i.e., $P = n_{\rm H} k_{\rm B} T$ where $k_{\rm B}$ is the Boltzmann constant. We set a typical value $Q = 2 \times 10^{48}$ s^{-1} for a OB-type star. The corresponding Strömgren radius is $r_{\rm S} \sim 2$ pc, so we have our simulation box size set to be 10 pc. We use a reduced speed of light $c_{\rm r} = 6 \times 10^{-4} c$ to avoid a prohibitive timestep restriction. Our total simulation time is 100 kyr. We start with all neutral hydrogen, i.e., $x_{\rm HII} = 0$.

To prevent numerical overflow and instabilities due to quantities being orders of magnitudes different, in the fluid and radiation steps, we have all quantities in code units, while in the thermochemisty step, all quantites are in physical (cgs) units. In code units, 1 length unit equals 10 pc, 1 mass unit equals the solar mass 1.99×10^{33} g, 1 time unit equals 1 kyr, and 1 $N_{\rm phot}$ unit equals 10^{48}. We convert all conserved variables from code units to physical units when entering the thermochemistry step, and after the thermochemistry update, we convert everything back to code units to be fed into the next iteration of hydrodynamics and radiation steps.

Figure 3 shows the expansion of an ionizing region around a central source embedded in a uniform neutral medium at one snapshot of our simulation. The radius of this sphere is ~ 2 pc. As the simulation proceeds, this radius should stay the same and the sphere should stop expanding beyond the Strömgen sphere radius $r_S = 2$ pc, since r_S represents an equilibrium point between ionization and recombination. However, we find this to be not the case, as our sphere keeps expanding and ionizing the entire box. We suspect there are several issues. We believe there might be some

issues with units and convergence, but due to the relatively short timescale we had to work on this project we did not have time to do an exhaustive set of tests to rigorously ensure every aspect of the code is correct. The purpose of this project was to learn about methods used to simulate RHD and implement them ourselves as this will serve valuable in our research endeavors in the future.

4. CONCLUSION

In this work, we design an RHD code in `Python` which simultaneously solves the Euler equations and the RT equation under M1 closure. Hydrodynamics and radiation are coupled via thermochemistry. We solve the Euler equations using a second-order Godunov method. More specifically, we use the MUSCL-Hancock scheme with MonCen slope limiter and smooth extrema detection and the HLL Riemann solver. We solve the RT equation under M1 closure using a first-order Godunov method. More specifically, we use the forward Euler scheme and GLF Riemann solver. The thermochemistry part is done by only assuming one species – hydrogen – of the gas and assuming only one group of ionizing photons with energy 13.6 eV.

We apply our code to several test problems. To test the hydro solver, we simulate the Kelvin-Helmholtz instability generated by two fluids moving at different velocities. We find that our hydro solver captures the instability, but our scheme suffers from numerical diffusion. This could be improved by switching to a less diffusive Riemann solver, such as HLLC, or by increasing the resolution of the simulation.

To test the RT solver, we simulate colliding radiation beams. When the radiation beams collide, flux is indeed generated in the perpendicular direction, as expected from the M1 closure scheme. Furthermore, the radiation beams remain well-collimated as they propagate through space, although some diffusion of photons can be seen on the outskirts of the beams.

To test the thermochemistry coupling of the gas and the radiation, we simulate the formation of a Strömgren sphere. We assume typical conditions of a molecular cloud in the Milky Way and successfully produce the expansion of an ionizing region embedded inside neutral hydrogen. However, the expansion does not halt at the Strömgren radius as theory would predict. There are likely still some bugs in our code which are creating this issue which we would rigorously explore if we had more time.

In future work, we could test an implicit time integrator instead of an explicit one. Then, the timestep would not be limited by the speed of light, making the reduced speed-of-light approximation unnecessary. This would be useful in scenarios in which one wishes to capture the evolution of multiple interacting radiation sources e.g. in a simulation of the epoch of reionization in which radiation from multiple galaxies gradually ionized the intergalactic medium. Another avenue for future work would be to simulate not only a single photon group and single chemical species, but instead simulate multiple photon groups and the ionization of helium as well.

5. AUTHOR CONTRIBUTION STATEMENT

ZA was responsible for the code which evolves the Euler equations. The RT equation under M1 closure was written by both ZA and YP. The thermochemistry module was written by both JS and YP. We often had two group members write the same code independently at first and then compare afterwards to help expedite the debugging process. ZA wrote Sections 1 and 2.1, both ZA and YP wrote Section 2.2, and JS and YP both wrote Section 2.3. Sections 3 and 4 were written by the entire team. For some movies of all the simulations we encourage the reader to view our github repository hosting the code.

REFERENCES

Mihalas, D., & Mihalas, B. W. 1984, Foundations of radiation hydrodynamics

Rosdahl, J., Blaizot, J., Aubert, D., Stranex, T., & Teyssier, R. 2013, MNRAS, 436, 2188