rt1d: Documentation

Jordan Mirocha

Last Updated: April 20, 2012

Contents

1 Introduction

Let's see what happens to the gas around stars and black holes, shall we? The primary reference for this code is arxiv:1204.1944 – much of the text has been extracted from there.

2 Methods

2.1 The Rate Equations

In general, the chemical and thermal evolution of gas surrounding a radiation source is governed by a set of differential equations describing the number densities of all ions and the temperature of the gas. Assuming a medium consisting of hydrogen and helium only, we first solve for the abundances of each ion via

$$\frac{dn_{\rm HII}}{dt} = (\Gamma_{\rm HI} + \gamma_{\rm HI} + \beta_{\rm HI} n_{\rm e}) n_{\rm HI} - \alpha_{\rm HII} n_{\rm e} n_{\rm HII} \tag{2.1}$$

$$\frac{dn_{\rm HeII}}{dt} = (\Gamma_{\rm HeI} + \gamma_{\rm HeI} + \beta_{\rm HeI} n_{\rm e}) n_{\rm HeI} + \alpha_{\rm HeIII} n_{\rm e} n_{\rm HeIII} - (\beta_{\rm HeII} + \alpha_{\rm HeII} + \xi_{\rm HeII}) n_{\rm e} n_{\rm HeII} \tag{2.2}$$

$$\frac{dn_{\rm HeIII}}{dt} = (\Gamma_{\rm HeII} + \gamma_{\rm HeII} + \beta_{\rm HeII} n_{\rm e}) n_{\rm HeII} - \alpha_{\rm HeIII} n_{\rm e} n_{\rm HeIII}. \tag{2.3}$$

Each of these equations represents the balance between ionizations of species HI, HeI, and HeII, and recombinations of HII, HeII, and HeIII. Associating the index i with absorbing species, $i = \mathrm{HI}, \mathrm{HeI}, \mathrm{HeII}$, we define Γ_i as the photo-ionization rate coefficient, γ_i as the secondary ionization rate coefficient, α_i (ξ_i) as the case-B (dielectric) recombination rate coefficients, β_i as the collisional ionization rate coefficients, and $n_{\rm e} = n_{\rm HII} + n_{\rm HeII} + 2n_{\rm HeIII}$ as the number density of electrons.

At each time-step, we also solve for the temperature evolution, dT_k/dt, which is given by

$$\frac{3}{2}\frac{d}{dt}\left(\frac{k_{\rm B} T_k n_{\rm tot}}{\mu}\right) = f^{\rm heat} \sum_i n_i \mathcal{H}_i - \sum_i \zeta_i n_e n_i - \sum_i \eta_i n_e n_i - \sum_i \psi_i n_e n_i - \omega_{\rm HeII} n_e n_{\rm HeII} \tag{2.4}$$

where $\mathcal{H}_i$ is the photo-electric heating rate coefficient (due to electrons previously bound to species i), $\omega_{\rm HeII}$ is the dielectric recombination cooling coefficient, and ζ_i, η_i, and ψ_i are the collisional ionization, recombination, and collisional excitation cooling coefficients, respectively. The constants in Equation 2.4 are the total number density of baryons, $n_{\rm tot} = n_{\rm H} + n_{\rm He} + n_e$, the mean molecular weight, μ, Boltzmann's constant, $k_{\rm B}$, and the fraction of secondary electron energy deposited as heat, $f^{\rm heat}$. We use the formulae in Appendix B of Fukugita & Kawasaki (1994) to compute the values of α_i, β_i, ξ_i, ζ_i, η_i, ψ_i, and $\omega_{\rm HeII}$.

2.2 Computing the Ionization and Heating Rates

The most critical aspect of propagating the radiation field in our 1D simulations is computing the ionization (Γ_i, γ_i) and heating ($\mathcal{H}_i$) rate coefficients accurately. In order to directly relate our

results to fully 3-dimensional radiative transfer calculations we have chosen to adopt a photon-conserving (PC) algorithm nearly identical to those employed by several widely used codes, like *C*^{2}*Ray* (Mellema et al., 2006), and *Enzo* (Wise & Abel, 2011). Our code is able to compute Γ_i, γ_i, and $\mathcal{H}_i$ in a non-photon-conserving (NPC) fashion as well, to enable comparison with previous 1D work such as Thomas & Zaroubi (2008). The two formalisms are equivalent in the limit of very optically thin cells, a condition that can be met easily in 1D calculations but is rarely computationally feasible in 3D. For NPC methods, if the optical depth of an individual cell is substantial, the number of ionizations in that cell will *not* equal the number of photons absorbed for that cell, i.e. photon number will not be conserved. This problem was remedied by Abel et al. (1999), who inferred the number of photo-ionizations of species i in a cell from the radiation incident upon it and its optical depth,

$$\Delta\tau_{i,\nu} = n_i \sigma_{i,\nu} \Delta r. \tag{2.5}$$

It is most straightforward to imagine our 1D grid as a collection of concentric spherical shells, each having thickness Δr and volume $V_{\rm sh}(r) = 4\pi[(r+\Delta r)^3 - r^3]/3$ (i.e. radii correspond to cell interfaces). The ionization and heating rates can then be related to the number of absorptions in any given shell (thus preserving photon number), as

$$\Gamma_i = A_i \int_{\nu_i}^{\infty} I_\nu e^{-\tau_\nu} \left(1 - e^{-\Delta\tau_{i,\nu}}\right) \frac{d\nu}{h\nu} \tag{2.6}$$

$$\gamma_{ij} = A_j \int_{\nu_j}^{\infty} \left(\frac{\nu - \nu_j}{\nu_i}\right) I_\nu e^{-\tau_\nu} \left(1 - e^{-\Delta\tau_{j,\nu}}\right) \frac{d\nu}{h\nu} \tag{2.7}$$

$$\mathcal{H}_i = A_i \int_{\nu_i}^{\infty} (\nu - \nu_i) I_\nu e^{-\tau_\nu} \left(1 - e^{-\Delta\tau_{i,\nu}}\right) \frac{d\nu}{\nu}, \tag{2.8}$$

where we've defined the normalization constant, $A_i \equiv L_{\rm bol}/n_i V_{\rm sh}(r)$, and denote the ionization threshold energy for species i as $h\nu_i$.

Equation 2.7 represents ionizations of species i due to fast secondary electrons from photoionizations of species j, which has number density n_j, and ionization threshold energy, $h\nu_j$. $f_i^{\rm ion}$ is the fraction of photo-electron energy deposited as ionizations of species i. In the remaining sections we only include the effects of secondary electrons when considering X-ray sources, which emit photons in the range $10^2{\rm eV} < E < 10^4{\rm eV}$. In this regime, the values of $f^{\rm heat}$ and $f_i^{\rm ion}$ computed via the formulae of Shull & van Steenberg (1985) are sufficiently accurate, but for radiation at lower energies where $f^{\rm heat}$ and $f_i^{\rm ion}$ have a stronger energy dependence, the lookup tables of Furlanetto & Stoever (2010) would be more appropriate. The total secondary ionization rate for a given species, γ_i, is the sum of ionizations due to the secondary electrons from all species, $\gamma_i = f_i^{\rm ion} \sum_j \gamma_{ij} n_j / n_i$.

The optical depth, $\tau_\nu = \tau_\nu(r)$, in the above equations is the total optical depth at frequency ν due to all absorbing species, i.e.

$$\begin{aligned} \tau_\nu(r) &= \sum_i \int_0^r \sigma_{i,\nu} n_i(r') dr' \\ &= \sum_i \sigma_{i,\nu} N_i(r) \end{aligned} \tag{2.9}$$

where N_i is the column density of species i at distance r from the source. We calculate the bound-free absorption cross-sections using the fits of Verner et al. (1996) throughout.

The values of Γ_i, γ_i, and $\mathcal{H}_i$ are completely predetermined for a given radiation source, and as a result, can be tabulated as a function of column density to avoid evaluating the integrals in these expressions numerically 'on-the-fly' as a simulation runs (e.g. Mellema et al., 2006; Thomas & Zaroubi, 2008). Isolating the frequency dependent components of Equations 2.6-2.8, we can define the integrals

$$\Phi_i(\tau_\nu) \equiv \int_{\nu_i}^{\infty} I_\nu e^{-\tau_\nu} \frac{d\nu}{h\nu} \tag{2.10}$$

$$\Psi_i(\tau_\nu) \equiv \int_{\nu_i}^{\infty} I_\nu e^{-\tau_\nu} d\nu, \tag{2.11}$$

allowing us to re-express the rate coefficients as

$$\Gamma_i = A_i \left[\Phi_i(\tau_\nu) - \Phi_i(\tau'_{i,\nu})\right] \tag{2.12}$$

$$\gamma_{ij} = \frac{A_j}{h\nu_i} \left\{\Psi_j(\tau_\nu) - \Psi_j(\tau'_{j,\nu}) - h\nu_j \left[\Phi_j(\tau_\nu) - \Phi_j(\tau'_{j,\nu})\right]\right\} \tag{2.13}$$

$$\mathcal{H}_i = A_i \left\{\Psi_i(\tau_\nu) - \Psi_i(\tau'_{i,\nu}) - h\nu_i \left[\Phi_i(\tau_\nu) - \Phi_i(\tau'_{i,\nu})\right]\right\}, \tag{2.14}$$

where $\tau'_{i,\nu} \equiv \tau_\nu + \Delta\tau_{i,\nu}$. Later references to "continuous SEDs" signify use of this technique, where the integral values Φ_i and Ψ_i are computed over a column density interval of interest a priori using a Gaussian quadrature technique, rather than on-the-fly via discrete summation.

Equations 2.12-2.14 are completely general for photon-conserving algorithms, whether the source SEDs are discrete or continuous – the only difference being for discrete SEDs, the integrals in Equations 2.10-2.11 become sums over the number of frequencies used, n_ν. In practice, computing Γ_i, γ_i, and $\mathcal{H}_i$ is more straightforward than this for sources with discrete SEDs, as we can simply count the number of ionizations caused by each individual frequency group, and convert this into the amount of excess electron kinetic energy available for further heating and ionization. When testing the accuracy of discrete solutions in later sections we employ this method, where radiation is emitted at n_ν frequencies, with each frequency ν_n carrying a fraction I_n of the bolometric luminosity. The photoionization and heating coefficients for each individual frequency group can then be expressed as

$$\Gamma_{i,n} = \frac{A_i I_n}{h\nu_n} e^{-\tau_{\nu_n}} (1 - e^{-\Delta\tau_{i,\nu_n}}) \tag{2.15}$$

$$\gamma_{ij,n} = \Gamma_{j,\nu_n} (\nu_n - \nu_j)/\nu_i \tag{2.16}$$

$$\mathcal{H}_{i,n} = \Gamma_{i,\nu_n} h(\nu_n - \nu_i). \tag{2.17}$$

The total rate coefficients can be found by summing each of these expressions over all frequencies, $n = 1, 2, 3...n_\nu$. These equations are identical to Equations 2.12-2.14 for the discrete SED case, but are perhaps more intuitive.

2.3 Additional Complexity: Energy Dependent Secondary Electron Treatment

In the previous section (and in all of Mirocha et al. 2012), we considered the asymptotic limit of Shull & van Steenberg (1985), in which the fractional energy deposition of secondary electrons as

heat, ionization, and excitation depends only on the hydrogen ionized fraction. Furlanetto & Stoever (2010) updated this work, providing lookup tables of deposition fraction as a function of both ionized fraction and electron energy. These effects are most important at $E \lesssim 10^2$ eV.

To accommodate this model, we must redefine the quantities Φ_i and Ψ_i. We will use a tilde to denote the new quantities related to secondary ionization,

$$\widetilde{\Phi}_{ij} \equiv \int_{\nu_j}^{\infty} f_{ij,(\nu-\nu_j)}^{\text{ion}} I_\nu e^{-\tau_\nu} \frac{d\nu}{h\nu} \tag{2.18}$$

$$\widetilde{\Psi}_{ij} \equiv \int_{\nu_j}^{\infty} f_{ij,(\nu-\nu_j)}^{\text{ion}} I_\nu e^{-\tau_\nu} d\nu, \tag{2.19}$$

and a hat to denote the new quantities related to photo-electric heating,

$$\widehat{\Phi}_i \equiv \int_{\nu_i}^{\infty} f_{(\nu-\nu_i)}^{\text{heat}} I_\nu e^{-\tau_\nu} \frac{d\nu}{h\nu} \tag{2.20}$$

$$\widehat{\Psi}_i \equiv \int_{\nu_i}^{\infty} f_{(\nu-\nu_i)}^{\text{heat}} I_\nu e^{-\tau_\nu} d\nu. \tag{2.21}$$

Now, our ionization and heating coefficients read

$$\Gamma_i = A_i \left[\Phi_i(\tau_\nu) - \Phi_i(\tau'_{i,\nu})\right] \tag{2.22}$$

$$\gamma_{ij} = \frac{A_j}{h\nu_i} \left\{ \widetilde{\Psi}_{ij}(\tau_\nu) - \widetilde{\Psi}_{ij}(\tau'_{j,\nu}) - h\nu_j \left[\widetilde{\Phi}_{ij}(\tau_\nu) - \widetilde{\Phi}_{ij}(\tau'_{j,\nu}) \right] \right\} \tag{2.23}$$

$$\mathcal{H}_i = A_i \left\{ \widehat{\Psi}_i(\tau_\nu) - \widehat{\Psi}_i(\tau'_{i,\nu}) - h\nu_i \left[\widehat{\Phi}_i(\tau_\nu) - \widehat{\Phi}_i(\tau'_{i,\nu}) \right] \right\}. \tag{2.24}$$

Various methods for handling secondary electrons are compared for a 10^5 K blackbody source of $\dot{Q} = 5 \times 10^{48}$ ionizing photons s^{-1} in Figure 1.

2.4 Time-stepping Techniques

2.5 Parallelism

3 Classes

In order of appearance in `RT1D.py`.

3.1 `InitializeGrid`

Initialize the temperature and ion densities for each cell in the grid.

3.2 `InitializeIntegralTables`

If `IntegralTabulation` = 1 (necessary when `DiscreteSpectrum` = 0), tabulates the quantities Φ_i and Ψ_i. If `MultiSpecies` = 1, tabulation occurs for hydrogen and helium ions, and if `SecondaryIonization` ≥ 2, also tabulate $\widetilde{\Phi}_i$, $\widetilde{\Psi}_i$, $\widehat{\Phi}_i$, and $\widehat{\Psi}_i$, adding an additional dimension for x_{HII}.

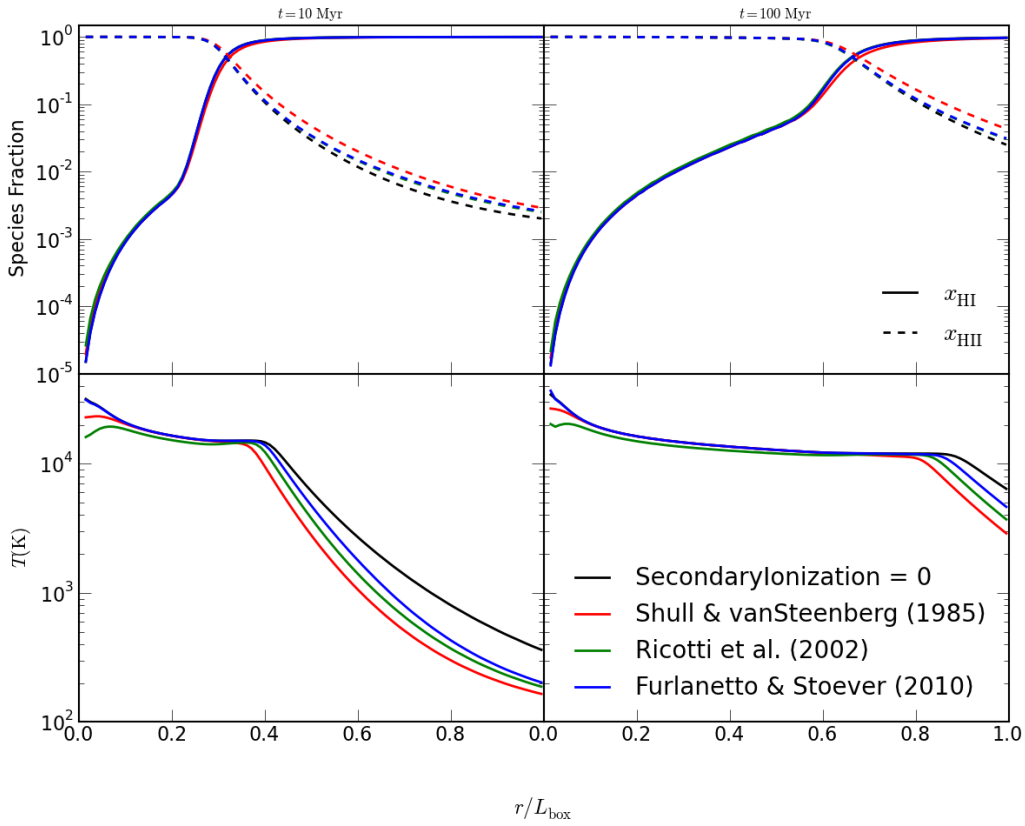

Figure 1: Comparison of different methods for secondary ionization. See §4.5 for details on setting the secondary ionization method.

3.3 `RadiationSource`

Initialize the radiation source. Most importantly the `Spectrum` function, which returns the fraction of the bolometric luminosity emitted at energy E (only argument).

3.4 `Radiate`

Calls `EvolvePhotons` routines, which loop over grid, call solver, and update ion densities and the temperature in each cell.

3.5 `SolveRateEquations`

ODE solver.

3.6 `Interpolate`

Performs linear, bi-linear, tri-linear, and eventually quad-linear interpolation on integral tables to extract ionization and heating coefficients as a function of column density.

3.7 `WriteData`

Write out data to HDF5. Would be easy to add support for ASCII write-out.

4 Parameters

4.1 Problem Types

These were added for convenience in January, 2011. If you want to make changes to these test problems you can, just make sure `ProblemType` is the first (uncommented) line in your parameter file. Additional parameters afterwards will be interpreted as usual.

`ProblemType = 0` Single-zone.

`ProblemType = 1` This will set up a standard test with known analytic solution: the propagation of an ionization front in an isothermal, hydrogen-only, static universe. The spectrum used is monochromatic at $E = 13.6$ eV, with photon luminosity $L_\nu = 5 \times 10^{48}\ s^{-1}$. The initial density, temperature, and box size are the same as the values used in Test 1 of Wise & Abel 2010.

`ProblemType = 2` This problem is almost exactly the same as `ProblemType = 1`. However, the temperature is allowed to evolve, and instead of using a monochromatic spectrum, we use a 10^5 K blackbody spectrum. Again, the initial density, temperature, box size, and energy groups are the same as the values used in Test 1 of Wise & Abel 2010.

`ProblemType = 2.1` Same as `ProblemType = 2`, except we sample the continuous blackbody spectrum with the four energy groups used by Wise & Abel (2011).

ProblemType = 2.2 Same as ProblemType = 2, except we sample the continuous blackbody spectrum with the four energy groups used by Mirocha et al. (2012).

ProblemType = 3 Test problem # 3 from Iliev et al. (2006) – I-front trapping in a dense clump and the formation of a shadow. The setup is the same as in ProblemType = 2, except a 1D clump is initialized at $x = 0.76L_{\rm box}$, overdensity $\delta = 200$, radius 0.12kpc, and temperature $T = 40$K, and the radiation field is plane-parallel.

ProblemType = 3.1 Same as ProblemType = 3, except we sample the continuous blackbody spectrum with the four energy groups used by Wise & Abel (2011).

4.2 Initial Conditions

DensityProfile

TemperatureProfile

InitialHIIFraction

4.3 Control Parameters

ODEatol Absolute tolerance of the ODE solver. If ionized fractions are zero to this tolerance, they are reset to MinimumSpeciesFraction. Combination of this parameter and ODErtol (below) determine whether or not the ODE time-step is cut in half. (Default = 10^{-5})

ODErtol Relative tolerance of the ODE solver. Combination of this parameter and ODEatol determine whether or not the ODE time-step is cut in half via the formulae $|y_2 - y_1| =$ ODEatol $+$ ODErtol $\times y_2$. (Default = 10^{-5})

4.4 Source Parameters

4.5 Physics Parameters

5 Analysis

Since our datasets are small, we can read in the entire time evolution all at once. To do this, type the following in a Python terminal:

```
import rt1d.analysis as rta
import pylab as pl

ds = rta.Analyze('./pf.dat')                # load parameter file and data
pl.loglog(ds.data[10].r, ds.data[10].x_HI)  # radius vs. neutral fraction
```

6 Test Problems

In this section we focus on test problems 0-3 of the Radiative Transfer Comparison Project (hereafter RT06; Iliev et al., 2006).

6.1 Test #0

Single-zone ionization, heating, and cooling.

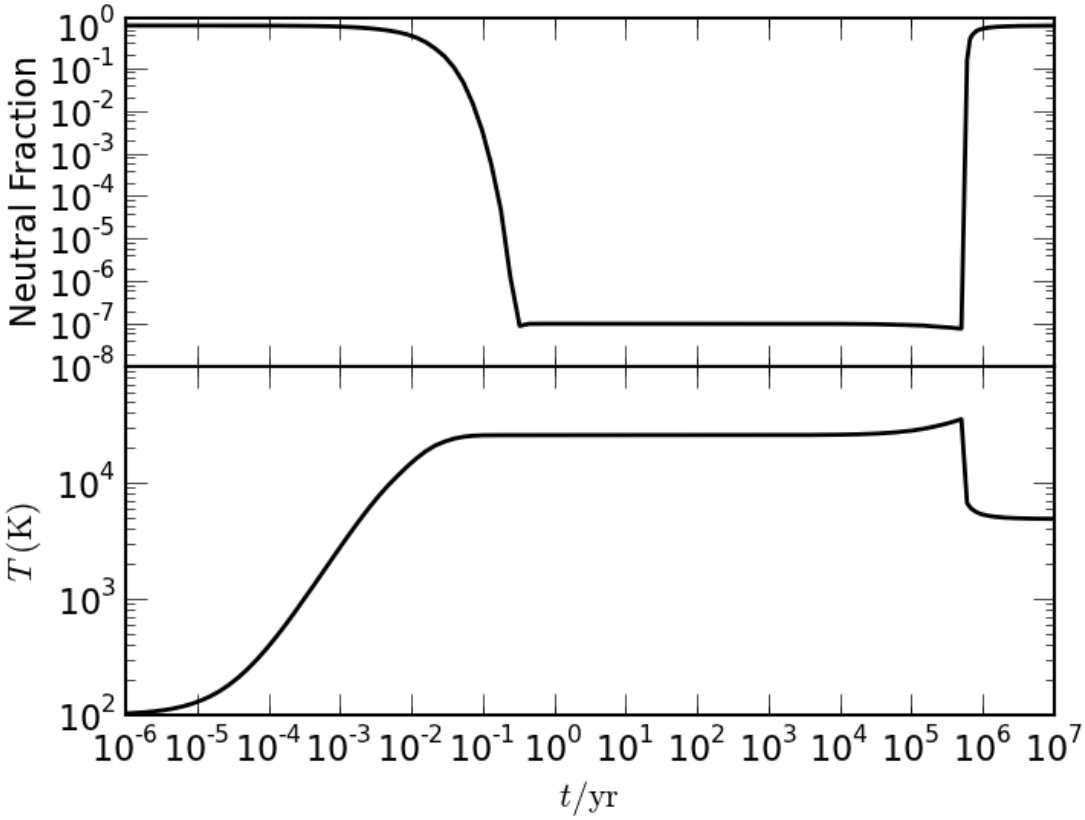

Figure 2: Single-zone ionization, heating, and cooling RT06 #0.

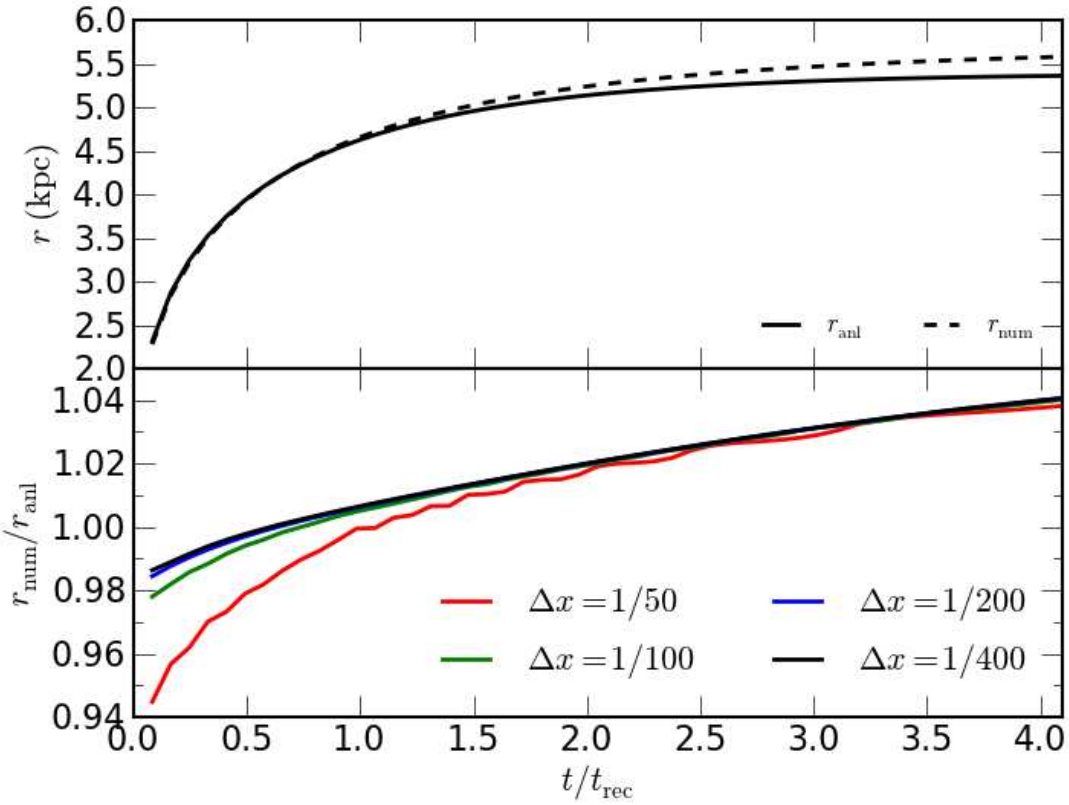

Figure 3: Convergence test for RT06 #1.

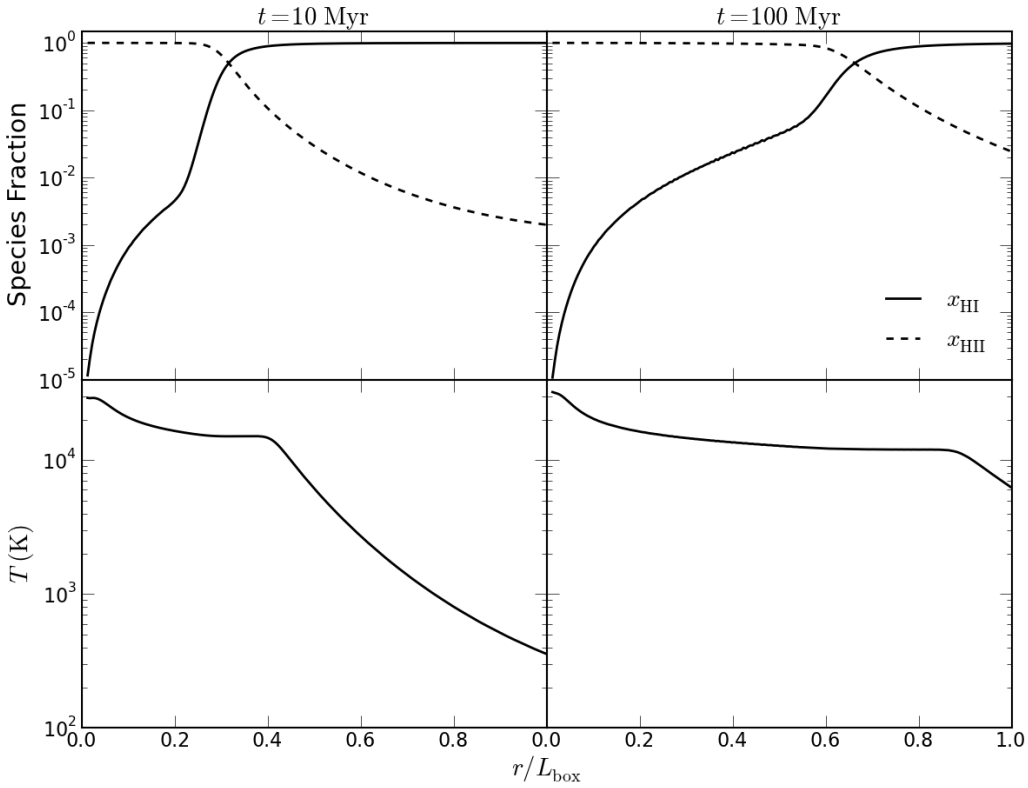

Figure 4: RT06 #3.

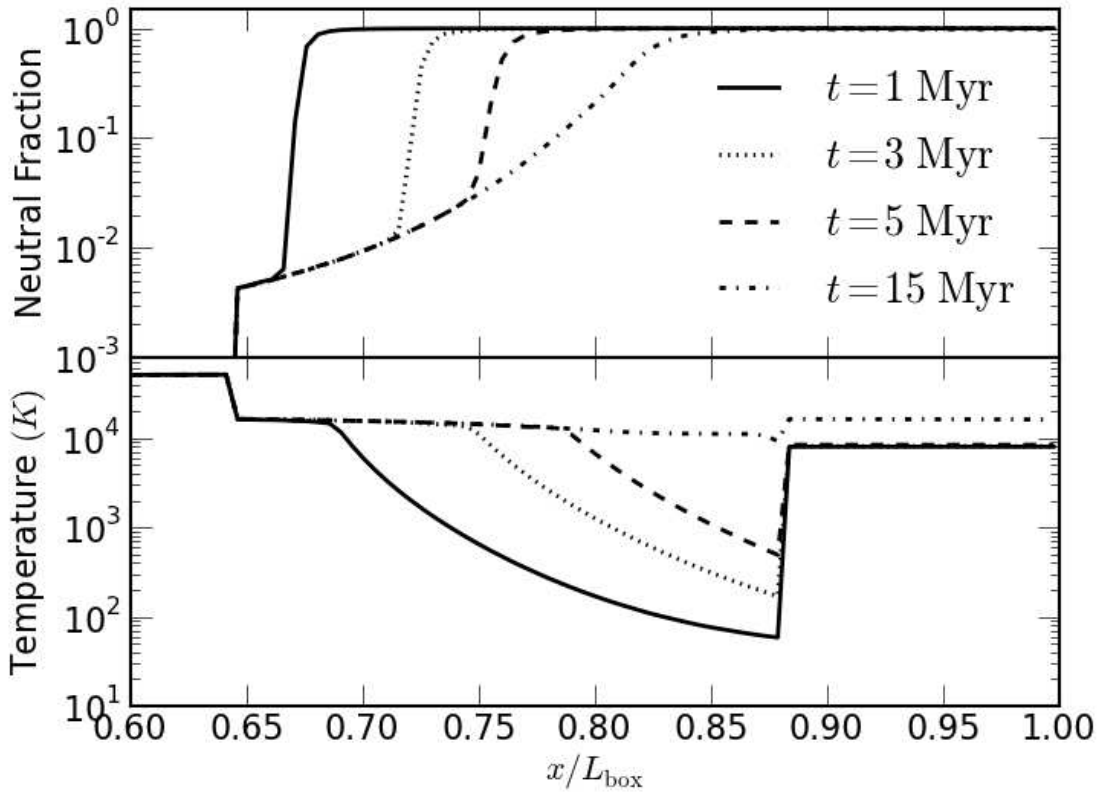

Figure 5: RT06 #3.

6.2 Test #1

6.3 Test #2

6.4 Test #3

7 Start to Finish

8 Advanced Usage

9 Convergence

In `rt1d/doc/examples` you will find directories containing convergence test suites for RT06 test problems 1 & 2 (`RT06_1_ResolutionTestSuite` and `RT06_2_ResolutionTestSuite`). Each of these directories contains the parameter files necessary to run these problems on grids containing 100×2^n cells, where $n = 0, 1, ...6$. If you want, you can run them all at once, though keep in mind the simulations with 100 cells will finish much much faster than those containing 6400 cells, so it might not be good to check out a node for this as most of the computation will be spent on 1 or 2 simulations. For instance, you could do:

```
cd rt1d/doc/examples/RT06_1_ResolutionTestSuite/c_infinite
mpirun -np 4 python RT1D.py -b batch_list.dat
```

I've included the same setup for $c \neq \infty$ in `rt1d/doc/examples/RT06_1_ResolutionTestSuite/c_finite`, so you can see the differences (if any) caused by making the $c \rightarrow \infty$ approximation.

10 Troubleshooting

If your results look horribly, horribly wrong, here are a few places where you could've gone wrong. We'll start with things that may occur with the stable version of the code.

Now, if you modify or add anything in the source code (awesome by the way!), you're far more likely to encounter troubles.

Totally Bogus Results This is will almost certainly happen at some point in the development process. But, if it happens even after you're *positive* the error was fixed, my guess is that you are using the integral tables that were generated with the previous, error-ridden version of the code. Try removing the integral tables and re-running your simulation.

11 Interactive Mode

Is it worth doing this? Would want it to look something like this:

```
import rt1d
import numpy as np
```

```
pf = rt1d.SetDefaultParameterValues()
r = np.linspace(3e19, 3e21, 100) # 100 grid cells between 0.01-1kpc. (optional)
data = rt1d.Shine(pf, r)
```

This would make it easy to:

- Allow users to write their own rt1d ‘scripts.’
- Change source properties manually with time.
- Making arbitrarily structured grids.
- Very specific initial conditions.

What parameters must be set?

- ProblemType
- GridDimensions
- MultiSpecies
- LengthUnits
- InitialDensity
- SourceParameters

References

Abel, T., Norman, M. L., & Madau, P. 1999, , 523, 66

Fukugita, M. & Kawasaki, M. 1994, , 269, 563

Furlanetto, S. R. & Stoever, S. J. 2010, , 404, 1869

Iliev, I. T., Ciardi, B., Alvarez, M. A., Maselli, A., Ferrara, A., Gnedin, N. Y., Mellema, G., Nakamoto, T., Norman, M. L., Razoumov, A. O., Rijkhorst, E.-J., Ritzerveld, J., Shapiro, P. R., Susa, H., Umemura, M., & Whalen, D. J. 2006, , 371, 1057

Mellema, G., Iliev, I. T., Alvarez, M. A., & Shapiro, P. R. 2006, , 11, 374

Shull, J. M. & van Steenberg, M. E. 1985, , 298, 268

Thomas, R. M. & Zaroubi, S. 2008, , 384, 1080

Verner, D. A., Ferland, G. J., Korista, K. T., & Yakovlev, D. G. 1996, , 465, 487

Wise, J. H. & Abel, T. 2011, , 414, 3458