Comparison of Two Distinct BBN Codes

1 Motivation

In 1993 I started working on a Big Bang Nucleosynthesis code used by Professors Tony Rothman and Richard Matzner for a 1984 paper on BBN in anisotropic models, and last updated by Professors Matzner and Hannu Kurki-Suonio. (To study neutrino degeneracy in anisotropic models.) Two years later Mr. David Thomas e-mailed me a different BBN code based on a 1992 preprint by Professor Lawrence Kawano and used in papers such as TSOF, 1993. These two different codes calculated different results for the final mass fractions X_i, particularly for $X_{^4He}$. This talk will outline the efforts I made to bring the two codes into agreement with each other. I was particularly concerned that the way each of the codes sets up the BBN problem to solve might be erroneously conceived, or that one or the other code uses approximate methods of calculating mass fractions and/or other physical quantities that improved computational accuracy have now rendered obsolete. Also, I wanted to add into the codes physical effects such as the neutrino heating effect, which changes $X_{^4He}$ an amount an order of magnitude less than the disagreement between the codes. I was hoping that from my work on both codes I could decide on the definitive way to solve the BBN problem.

2 Trace of Each Program

2.1 The Rothman Code

The Rothman code could work on anisotropic models, but I will cover only the code's sections that apply to the standard model in my talk. The code starts at a temperature $T = 100$ GK and initializes the following thermal quantities according to their familiar equations:

$$\begin{aligned}
a &= \frac{\pi^2}{15}\frac{k^4}{(c\hbar)^3} \\
E &= \frac{m_e}{kT} \\
\rho_\gamma &= aT^4 \\
n_\gamma &= \frac{30(1.202)}{k\pi^4}aT^3 \\
\rho_e &= \frac{30}{\pi^4}aT^4\int_0^\infty dX\frac{X^2\sqrt{X^2+E^2}}{e^{\sqrt{X^2+E^2}}+1} \\
P_e &= \frac{10}{\pi^4}aT^4\int_0^\infty dX\frac{X^4}{\sqrt{X^2+E^2}(e^{\sqrt{X^2+E^2}}+1)} \\
N_e &= \frac{30}{k\pi^4}aT^3\int_0^\infty dX\frac{X^2}{e^{\sqrt{X^2+E^2}}+1} \\
\rho_\nu &= \frac{7}{8}aT_\nu^4 \qquad (1) \\
\rho_b R^3 &= const. \qquad (2)
\end{aligned}$$

$$\rho_b = m_N n_\gamma \eta \tag{3}$$

though the code approximates $\rho_e = 7/4aT^4$ at first. The code also initiallizes the value of $\ln(R) = \ln(T_{\nu_0}/T)$ where R is the expansion coefficient of the universe and $T_{\nu_0} = 1.946$ K is the current neutrino temperature. Note in particular that the code uses the assumption that $\rho_b R^3$ stays constant for all time and so sets the value of ρ_b by inputing a present-day value for ρ_b. The code also initiallizes

$$\begin{aligned} Y_n &= \frac{1}{e^{\frac{m_n - m_p}{kT}} + 1} \\ Y_p &= 1 - Y_p \end{aligned}$$

based on the fact that neutrons and protons start out in thermal equilibrium with each other via neutron to proton conversion rates. But the code sets the rest of the element abundances to 0.

The Rothman code goes about evolving the physical quantities per timestep in this manner. At a step number n we have the abundances $Y_n(i)$, where i refers to the particular isotope, and the temperature and thermal quantities that I've mentioned with values corresponding to step n. As with most other codes, the Rothman code evolves the abundances using a 2nd. order Runge-Kutta scheme.

$$Y_{n+1}(i) = Y_n(i) + \frac{1}{2}\left[\frac{dY(i)}{dt}(t, Y_n) + \frac{dY(i)}{dt}(t + \Delta t, Y_{n+1})\right]\Delta t \tag{4}$$

For reactions involving $Y(i)$ of the form

$$N_i \left({}^{A_i}Z_i\right) + N_j \left({}^{A_j}Z_j\right) \leftrightarrow N_k \left({}^{A_k}Z_k\right) + N_l \left({}^{A_l}Z_l\right)$$

$dY(i)/dt$ has the following equation, from Wagoner, 1969.

$$\frac{dY(i)}{dt} = \sum_{j,k,l} N_i \left(-\frac{Y^{N_i}(i)Y^{N_j}(j)}{N_i!N_j!}[ij] + \frac{Y^{N_k}(k)Y^{N_l}(l)}{N_k!N_l!}[kl]\right) \tag{5}$$

where $[ij]$ refers to the forward rate and $[kl]$ the reverse rate. But Equation (5) has a very small difference between to very large numbers, and so one has to use the familiar linearization of the right-hand-side written out in Wagoner, 1969

$$\frac{dY(i)}{dt}(t, Y_n) = \frac{Y_{n\alpha}(i) - Y_n(i)}{\Delta t} \tag{6}$$

$$\begin{aligned} \frac{dY(i)}{dt}(t, Y_n) = \sum_{i,j,k} &-\frac{N_i[ij]}{N_i!N_j!(N_i + N_j)} \\ &[N_i Y_n^{N_i - 1}(i) Y_n^{N_j}(j) Y_{n\alpha}(i) + N_j Y_n^{N_i}(i) Y_n^{N_j - 1}(j) Y_{n\alpha}(j)] + \frac{N_i[kl]}{N_k!N_l!(N_k + N_l)} \\ &[N_k Y_n^{N_k - 1}(k) Y_n^{N_l}(l) Y_{n\alpha}(k) + N_l Y_n^{N_k}(k) Y_n^{N_l - 1}(l) Y_{n\alpha}(l)] \end{aligned} \tag{7}$$

calculated using the latest value of temperature T_n (The code also calculates the numerical integral form for the neutron-proton conversion rates.) Also note that Equation (7) is the right hand side of (4) averaged between Y_n and $Y_{n\alpha}$, as opposed to an expansion of (4). The Rothman code also evolves $\ln R$ and $\rho_{e+\gamma}$ using 2nd. order Runge-Kutta. Assuming a Friedmann-Robertson-Walker model of the universe, and the universe as a perfect fluid.:

$$\begin{aligned} ds^2 &= dt^2 - R^2(t)\frac{dr^2 + r^2\left(d\theta^2 + \sin^2\theta d\phi^2\right)}{\left(1 + \frac{1}{4}kr^2\right)^2} \\ T^{ab} &= (\rho + p)U^a U^b - pg^{ab} \end{aligned}$$

one can derive an expression for $\dot{R}/R$ from the Einstein field equations and one for $\dot{\rho}_{e+\gamma}$ from the energy conservation law $T^{0b}{}_{;b}$.

$$\begin{aligned} \frac{\dot{R}}{R} &= \sqrt{\frac{8}{3}\pi G\left(\rho_\gamma + \rho_e + \rho_\nu\right)} \\ \dot{\rho}_{e+\gamma} &= -3\frac{\dot{R}}{R}\left(p_{e+\gamma} + \rho_{e+\gamma}\right) \\ \dot{\rho}_{e+\gamma} &= -4\frac{\dot{R}}{R}\rho_\gamma - 3\frac{\dot{R}}{R}\left(p_e + \rho_e\right) \end{aligned} \tag{8}$$

also using $p_\gamma = \rho_\gamma/3$ for the massless photons. (Note: The Rothman code can also R-K evolve 14 other thermal quantities, but these quantities all apply only to the anisotropic model and so I will not discuss them now.)

¿From (6) and (7) one can construct a matrix Q_{ij} and solve the following matrix equation.:

$$Y_n(i) = Q_{ij}\left(Y_n, [ij](T_n), [kl](T_n)\right) \times Y_{n\alpha}(j) \tag{9}$$

Please note that for this first R-K time derivative, the Rothman code uses Y_n in both the left hand side and the matrix Q_{ij}. Right after calculating $dY(i)/dt(t, Y_n)$ the code calculates $\dot{R}/R(t,\rho)$ and $\dot{\rho}_{e+\gamma}(t,\rho,p)$ with ρ and p corresponding to T_n. The code then evolves $Y(i)$, $\ln R$ and $\rho_{e+\gamma}$ to interim ($\tilde{}$) values.

$$\begin{aligned} \tilde{Y}_n(i) &= Y_n(i) + \frac{dY(i)}{dt}(t, Y_n(i))\Delta t_n \qquad (10) \\ (\tilde{\ln R})_n &= (\ln R)_n + \left[\frac{\dot{R}}{R}(t,\rho)\right]\Delta t_n \\ (\tilde{\rho}_{\gamma+e})_n &= (\rho_{\gamma+e})_n + (\dot{\rho}_{\gamma+e}(t,\rho,p))\Delta t_n \end{aligned}$$

and from these new values derives a new value $\tilde{T}_n$. The Rothman code used to use an interpolation table for T between 11 GK and 0.5 GK, when electron/positron annihilation occurs, approximating m_e =0.0 for higher temperatures and $\rho_e = 0.0$ for lower temperatures. But I replaced the table with Newton's Method, where I define,

$$F(T) = (\rho_\gamma(T) + \rho_e(T)) - \tilde{\rho}_{\gamma+e}$$

using the first transparency's equations for ρ_γ and ρ_e to find a temperature $\tilde{T}$ where $F(\tilde{T})$ is closest to 0.0. The code then determines the corresponding values of p_e and n_e. The code calculates ρ_b and ρ_ν from Equations (1) and (2)

After updating these thermal quantities, the code calculates the second R-K derivatives. But it calculates the derivatives for $\ln R$ and $\rho_{e+\gamma}$ before it does $dY(i)/dt$. The code calculates the final values for $\ln R$ and $\rho_{e+\gamma}$

$$\begin{aligned}(\ln R)_{n+1} &= (\ln R)_n + \frac{1}{2}\left[\frac{\dot{R}}{R}(t, \rho(T_n)) + \frac{\dot{R}}{R}(t+\Delta t_n, \rho(\tilde{T}_n))\right]\Delta t_n \\ (\rho_{e+\gamma})_{n+1} &= (\rho_{e+\gamma})_n + \frac{1}{2}\left[\dot{\rho}_{e+\gamma}(t, \rho(T_n), p(T_n)) + \dot{\rho}_{e+\gamma}(t+\Delta t_n, \rho(\tilde{T}_n), p(\tilde{T}_n))\right]\Delta t_n\end{aligned}$$

and determines the final T_{n+1} and other thermal quantities the same way it did the intermediate $\tilde{T}_n$. It uses T_{n+1} to calculate the reaction rates for the second R-K derivative of $Y(i)$.

$$\begin{aligned}\tilde{Y}_n(i) &= Y_n(i) + \frac{dY(i)}{dt}(t, Y_n(i))\Delta t_n \\ Y_n(i) &= Q_{ij}\left(\tilde{Y}_n, [ij](T_{n+1}), [kl](T_{n+1})\right) \times Y_{n\beta}(j)\end{aligned}$$

So the second matrix is made of the intermediate abundances $\tilde{Y}_n(i)$ and the rates evaluated at $t + \Delta t_n$.

$$\begin{aligned}Y_n(i) &= Q_{ij}(\tilde{Y}_n, [ij](T_{n+1}), [kl](T_{n+1})) \times Y_{n\beta}(j) && (11)\\ \frac{dY(i)}{dt}(t+\Delta t_n, \tilde{Y}_n) &= \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t} \\ Y_{n+1}(i) &= Y_n(i) + \frac{1}{2}\left[\frac{dY(i)}{dt}(t, Y_n) + \frac{dY(i)}{dt}(t+\Delta t_n, \tilde{Y}_n)\right]\Delta t_n && (12)\end{aligned}$$

Only now does the code determines the value Δt_{n+1} for the next timestep:

$$\Delta t_{n+1} = dk \min\left[\frac{Y_{n+1}(i)}{\frac{dY(i)}{dt}}, \frac{(lnR)_{n+1}}{\frac{\dot{R}}{R}}, \frac{(\rho_{e+\gamma})_{n+1}}{\dot{\rho}_{e+\gamma}}\right] \qquad (13)$$

where the time derivatives are the averages of the two R-K time derivatives. The code follows this process for all timesteps until T falls below 0.01 GK.

2.2 The Kawano Code

The Kawano code initializes $Y(1)$ and $Y(2)$ as before, but for the initial value of the deuterium abundance, $Y(3)$ it uses an equation derived from the case Nuclear Statistical Equilibrium:

$$
\begin{aligned}
X_i &= \frac{g_i}{2}\left(2(1.202)\sqrt{\frac{2}{\pi}}\right)^{A_i-1} A_i^{5/2}\left(\frac{kT}{m_N}\right)^{3(A_i-1)/2} \eta^{A_i-1}X_p^{Z_i}X_n^{A_i-Z_i}\exp\left(\frac{B}{kT}\right) \\
B &= Z_i m_p + (A_i - Z_i)m_n - m_i
\end{aligned}
$$

derived from the fact that the small traces of light nuclei around at the earliest times are in thermal equilibrium with each other via particular reactions and the good estimate of their distribution functions as Maxwell-Boltzmann distributions. More importantly, though, it sets all other $Y(i)$ to a minimum value of $Y_{min} = 10^{-25}$. Instead of an input value for $\rho_b(2.726\text{ K})$ one inputs the electron chemical potential ϕ_e (100 GK). The Rothman codes assumes $\phi_e = 0$, and therefore $n_{e^-} = n_{e^+}$, but the Kawano uses ϕ_e to determine the difference between n_{e^-} and n_{e^+} via this Bessel expansion of the the equation.

$$
n_{e^-} - n_{e^+} = \frac{2}{15}\frac{m_e^3}{k^4}\sum_{n=1}^{\infty}\sinh(n\phi_e)L(nz)
$$

where $z = m_e/kT$ and $L(nz)$ are Bessel functions. Charge conservation requires that this difference equal the proton number density, and so

$$
\begin{aligned}
n_{e^-} - n_{e^+} &= \frac{\rho_b S}{m_N} \\
\rho_b &= h_v a\frac{15}{\pi^2}T^3
\end{aligned} \tag{14}
$$

where $S = \sum_i Z_i Y(i)$, and from ρ_b one gets η. The Kawano code defines a variable h_v such that $\rho_b = h_v(15a/\pi^2)T^3$. The code then initiallizes $T = 100$ GK and $T_\nu = T$ at this high a temperature and initiallizes ρ_γ, ρ_e, p_e, and ρ_ν using the same equations as before.

Instead of evolving $\rho_{e+\gamma}$ and $\ln R$, the code evolves T, h_v and ϕ_e. Actually, to derive dT/dt the code starts with the energy conservation equation.

$$
\begin{aligned}
\frac{d\rho_{e+\gamma}}{dT}\frac{dT}{dt} &= -3\frac{\dot{R}}{R}(p_{e+\gamma} + \rho_{e+\gamma}) \\
\frac{dT}{dt} &= -\frac{3\frac{\dot{R}}{R}(p_{e+\gamma} + \rho_{e+\gamma})}{\frac{d\rho_\gamma}{dT} + \frac{d\rho_e}{dT}}
\end{aligned} \tag{15}
$$

But in calculating $d\rho_e/dT$ the Kawano code also takes into account the time dependence of ϕ_e.

$$
\begin{aligned}
\frac{d\rho_e}{dT} &= \frac{\partial\rho_e}{\partial T} + \frac{\partial\rho_e}{\partial\phi_e}\frac{\partial\phi_e}{\partial T} \\
\frac{\partial\rho_e}{\partial\phi_e} &= \frac{2}{15}a(zT)^4\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{z}\sinh(n\phi_e)M(nz)
\end{aligned}
$$

The code uses a Bessel expansion of $\rho_{e^-} + \rho_{e^+}$ to get an expression for $\partial\rho_e/\partial\phi_e$. And for $\partial\phi_e/\partial T$ it divides Equation (14) through by T^3, anc calls the left hand side $N(T, \phi_e)$ and the right hand side $M(T, S, R)$.

$$\begin{aligned} M(T,S,R) &= \frac{\rho_b S}{m_N T^3} \\ N(T,\phi_e) &= \frac{2}{15}\frac{az^3}{k}\sum_{n=1}^{\infty}(-1)^{n+1}\sinh(n\phi_e)L(nz) \\ \frac{\partial\phi_e}{\partial T} &= \frac{\frac{\partial M}{\partial T} - \frac{\partial N}{\partial T}}{\frac{\partial N}{\partial \phi_e}} \end{aligned}$$

Both functions depend explicitly on T, but N has a ϕ_e dependence as well. So the code takes the partials of both functions with respect to T and divides the difference by $\partial N/\partial\phi_e$. dh_v/dt is easily derived from the equation $h_v = const/(RT)^3$

$$\frac{dh_v}{dt} = -3h_v\left(\frac{1}{R}\frac{dR}{dt} + \frac{1}{T}\frac{dT}{dt}\right) \tag{16}$$

The code assumes that ϕ_e depends on T, R and S and uses the M and N functions to calculate $d\phi_e/dt$.

$$\begin{aligned} \frac{d\phi_e}{dt} &= \frac{\partial\phi_e}{\partial T}\frac{dT}{dt} + \frac{\partial\phi_e}{\partial R}\frac{dR}{dt} + \frac{\partial\phi_e}{\partial S}\frac{dS}{dt} \\ \frac{d\phi_e}{dt} &= \frac{\partial\phi_e}{\partial T}\frac{dT}{dt} + \frac{\frac{3h_v S}{m_N}\frac{\dot{R}}{R} + + \frac{h_v}{m_N}\sum_i Z_i \frac{dY(i)}{dt}}{\frac{\partial N}{\partial\phi_e}} \end{aligned} \tag{17}$$

At step n, the Kawano code builds Q_{ij} out of Y_n and the rates evaluated at T_n, but on the left hand side the code uses Y_{n-1} instead of Y_n.

$$Y_{n-1}(i) = Q_{ij}\left(Y_n, [ij](T_n), [kl](T_n)\right) \times Y_{n\alpha}(j) \tag{18}$$

By this point the Rothman code will have already calculated Δt_n, at the end of step $n-1$. But the Kawano code uses Δt_{n-1} to calculate the first dY/dt in the R-K scheme.

$$\frac{dY(i)}{dt}(t, Y_n(i)) = \frac{Y_{n\alpha}(i) - Y_{n-1}(i)}{\Delta t_{n-1}}$$

At this point, unlike the Rothman code, the Kawano code calculates the new timestep Δt_n using the time derivatives for $Y(i)$ and T calculated just now.

$$\begin{aligned} \Delta t_n &= dk \min\left[\frac{Y_n(i)}{\dot{Y}(i)}dk_{fac}, \frac{T_n}{\frac{dT}{dt}}\right] \\ dk_{fac} &= \left(1 + \left(-\frac{\log Y_n(i)}{dt_{fac}}\right)^2\right) \end{aligned} \tag{19}$$

$$dt_{fac} = -\frac{\log Y_{min}}{\frac{dk_{max}}{dk} - 1}$$

The factor dk_{fac} makes the code less likely to pick a value of Δt from a $Y(i)$ close to Y_{min}. We can then evolve $Y(i)$ and the three thermal quantities to their interim values.

$$\begin{aligned}
\tilde{Y}_n(i) &= Y_n(i) + \frac{dY(i)}{dt}(t, Y_n(i))\Delta t_n \\
\tilde{T}_n &= T_n + \left[\frac{dT}{dt}(t, \rho_n)\right]\Delta t_n \\
(\tilde{h}_v)_n &= (h_v)_n + \left[\frac{dh_v}{dt}(t, R_n)\right]\Delta t_n \\
(\tilde{\phi}_e)_n &= (\phi_e)_n + \left[\frac{d\phi_d}{dt}(t, T_n, R_n, S_n)\right]\Delta t_n
\end{aligned} \tag{20}$$

With these new values of the thermal quantities the code updates the other quantities $[ij]$, ρ_γ, ρ_e, p_e, ρ_b, and ρ_ν and T_ν. But to determine T_ν the code uses the equation for the total entropy per unit volume.

$$\begin{aligned}
s &= \frac{R^3}{T}(\rho_{\gamma+e^\pm} + p_{\gamma+e^\pm}) \\
s &= \frac{4}{3}a(RT)^3 I\left(\frac{m_e}{kT}\right) \\
I(x) &= 1 + \frac{45}{2\pi^4}\int_0^\infty dy \frac{y}{e^{\sqrt{x^2+y^2}}+1}\left(\sqrt{x^2+y^2} + \frac{y^2}{3\sqrt{x^2+y^2}}\right)
\end{aligned} \tag{21}$$

The code calculates $I(m_e/kT)$ for T and for T_o =100GK. Then one can use RT_ν = a constant to get

$$\frac{T_o}{(T_\nu)_o}T_\nu = \left[\frac{I(x)}{I(x_o)}\right]^{\frac{1}{3}} T \tag{22}$$

where $(T_\nu)_o = T_o$ at 100 GK. Now the code can calculate the second R-K derivatives for $Y(i)$ and T, h_v, and ϕ_e.

$$\begin{aligned}
Y_n(i) &= Q_{ij}\left(\tilde{Y}_n, [ij](\tilde{T}_n), [kl](\tilde{T}_n)\right) \times Y_{n\beta}(j) \\
\frac{dY(i)}{dt}(t + \Delta t_n, Y_{n+1}(i)) &= \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t_n}
\end{aligned} \tag{23}$$

Note that the equation now uses Y_n on the left hand side, and the rates dependent on $\tilde{T}_n$ rather then the final T_{n+1}.

$$Y_{n+1}(i) = Y_n(i) + \frac{1}{2}\left[\frac{dY(i)}{dt}(t, Y_n) + \frac{dY(i)}{dt}(t + \Delta t_n, Y_{n+1})\right]\Delta t_n \tag{24}$$

$$
\begin{aligned}
T_{n+1} &= T_n + \frac{1}{2}\left[\frac{dT}{dt}(t,\rho_n) + \frac{dT}{dt}(t+\Delta t_n, \tilde{\rho}_n)\right]\Delta t_n \\
(h_v)_{n+1} &= (h_v)_n + \frac{1}{2}\left[\frac{dh_v}{dt}(t,R_n) + \frac{dh_v}{dt}(t+\Delta t, \tilde{R}_n)\right]\Delta t_n \\
(\phi_e)_{n+1} &= (\phi_e)_n + \frac{1}{2}\left[\frac{d\phi_e}{dt}(t,T_n,R_n,S_n) + \frac{d\phi_e}{dt}(t+\Delta t, \tilde{T}_n, \tilde{R}_n, \tilde{S}_n)\right]\Delta t_n
\end{aligned}
$$

I started working on both codes in 1995. The Rothman code originally calculated the abundances for only nine isotopes. So I added in the reaction network featured in TSOF, 1993 and TSOMMF, 1994 to raise the number to 68. I updated the values of the constants and the reverse rate parameters, and added to the Rothman code the Coulomb correction of the n-p conversion rates, which the Kawano code already had. Table (1) shows the final abundances for selected isotopes corresponding to a final value of $\eta_{10} = 3.0$. Specifically we have disagreement between a Rothman $X_{^4He} = 0.2397015878$ vs. a Kawano $X_{^4He}$ =0.2369001572, on the order 0.001. And Figure (1), which graphs $X_{^4He}$ for both codes as a function of η_{10}, shows that the codes have this disagreement all through the relavent ranges of η_{10}. I need to bring the codes into close enough agreement to bring the difference order down to at least 0.0001.

3 Convergence of the Codes

We'll first look at the convergence of the codes. During a run, the timestep order can vary from 10^{-8} to 10^6. So to test a code's convergence, I save its timestep values as a file and work in the option to read the timestep off this file. Then I could modify the code so that the code could run with those values halved, then quartered, and so forth, and I could determine to what values the mass fractions eventually converge. The Kawano code's mass fraction of ^4He was found to best obey:

$$
X_{^4He} = 0.2392602277 - (0.2474541681 \times 10^{-2})d^2
$$

where $d =$ one over the number of divisions of each timestep value in the file used (1/1, 1/2, 1/4, and so forth). This equation has the second order convergence that one expects from the second order R-K scheme. But the Rothman code data instead best obeys:

$$
X_{^4He} = 0.2394509836 + (0.3832102957 \times 10^{-3})d
$$

Only a first-order convergence. Figure (2) shows the graphs of the differences between the $X_{^4He}$'s and the constants they converge to. The data points show the first and second order convergences I've mentioned, though the Rothman code's data doesn't fit into its graph until the higher numbers of divisions.

To explain this behavior, one has to look again at how each code goes about calculating $dY(i)/dt$ and how well their methods actually fit into theory. Both codes use the approximations (6) and (7) of Equation (1) instead of directly calculating it. One derives approximation (6) from the semi-implicit Euler Method described in Wagoner, 1969, and Kawano, 1992.

$$\begin{aligned} Y_n(i) &= Y_{n+1}(i) - \frac{dY(i)}{dt}[t+\Delta t_n, Y_{n+1}(i)]\Delta t_n \\ \frac{dY(i)}{dt}[t+\Delta t_n, Y_{n+1}(i)] &= \frac{Y_{n+1}(i) - Y_n(i)}{\Delta t_n} \end{aligned} \tag{25}$$

(Semi-implicit because one linearizes the expression for the time derivative into Equation (7).) Looking back at the Kawano code, one can match its dY/dt equations to the semi-implicit method very clearly.

$$\begin{aligned} Y_{n-1}(i) &= Q_{ij}\left(Y_n, [ij](T_n), [kl](T_n)\right) \times Y_{n\alpha}(j) \\ \frac{dY(i)}{dt}(t, Y_n(i)) &= \frac{Y_{n\alpha}(i) - Y_{n-1}(i)}{\Delta t_{n-1}} \\ Y_n(i) &= Q_{ij}\left(\tilde{Y}_n, [ij](\tilde{T}_n), [kl](\tilde{T}_n)\right) \times Y_{n\beta}(j) \\ \frac{dY(i)}{dt}(t+\Delta t_n, Y_{n+1}(i)) &= \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t_n} \end{aligned}$$

So $Y_{n\alpha}$ here is a first order Euler approximation of Y_n, and $Y_{n\beta}$ is a first order approximation of Y_{n+1} Plugging into Equations (20) and (24), we get

$$\begin{aligned} \tilde{Y}_n(i) &= Y_n(i) + \frac{Y_{n\alpha}(i) - Y_{n-1}(i)}{\Delta t_{n-1}}\Delta t_n \\ Y_{n+1}(i) &= Y_n(i) + \frac{1}{2}\left[\frac{Y_{n\alpha}(i) - Y_{n-1}(i)}{\Delta t_{n-1}} + \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t_n}\right]\Delta t_n \end{aligned}$$

Note that because of the use of both Δt_{n-1} and Δt_n the intermediate $\tilde{Y}_n(i)$ is not equal to $Y_{n\alpha}(i)$. Also, the expression $[Y_{n\alpha}(i) - Y_{n-1}(i)]/\Delta t_{n-1}$ is used for $dY(i)/dt$ to determine Δt_n. This makes sense since this expression does correspond the time derivative at time t_n.

But we've noted that the first matrix equation in the Rothman code has the same matrix as in the Kawano code but uses Y_n instead of Y_{n-1}

$$\begin{aligned} Y_n(i) &= Q_{ij}\left(Y_n, [ij](T_n), [kl](T_n)\right) \times Y_{n\alpha}(j) \\ \frac{dY(i)}{dt}(t, Y_n)? &= \frac{Y_{n\alpha}(i) - Y_n(i)}{\Delta t_n} \end{aligned}$$

Matching the right-hand-side to (25), one would think that the left hand side should be $dY(i)/dt(t+\Delta t_n, Y_{n+1})$ instead of $dY(i)/dt(t, Y_n)$, except that the matrix used to construct $dY(i)/dt$ is written for $dY(i)/dt(t, Y_n)$. Similarly, the second R-K derivative

$$\begin{aligned} Y_n(i) &= Q_{ij}\left(\tilde{Y}_n, [ij](T_{n+1}), [kl](T_{n+1})\right) \times Y_{n\beta}(j) \\ \frac{dY(i)}{dt}(t+\Delta t_n, \tilde{Y}_n) &= \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t_n} \end{aligned}$$

would also seem to match (25) for $dY(i)/dt(t+\Delta t_n, Y_{n+1})$ So both $Y_{n\alpha}$ and $Y_{n\beta}$ seem to be first order Euler approximations of Y_{n+1}, while $Y_{n\alpha}$ is an approximation of Y_n in the Kawano code. And when you plug in these expressions for $dY(i)/dt$.:

$$\begin{aligned}
\tilde{Y}_n(i) &= Y_n(i) + \frac{Y_{n\alpha}(i) - Y_n(i)}{\Delta t_n}\Delta t_n \\
\tilde{Y}_n(i) &= Y_{n\alpha}(i) \\
Y_{n+1}(i) &= Y_n(i) + \frac{1}{2}\left[\frac{Y_{n\alpha}(i) - Y_n(i)}{\Delta t_n} + \frac{Y_{n\beta}(i) - Y_n(i)}{\Delta t_n}\right]\Delta t_n \\
Y_{n+1}(i) &= \frac{1}{2}(Y_{n\alpha}(i) + Y_{n\beta}(i))
\end{aligned} \tag{26}$$

. Because the Rothman code uses Δt_n throughout the timestep, the interim value $\tilde{Y}_n(i)$ is the same value $Y_{n\alpha}(i)$ calculated from the matrix equation. And the final value of $Y_{n+1}(i)$ is the average of the two first-order Euler approximations $Y_{n\alpha}(i)$ and $Y_{n\beta}(i)$ calculated from the matrix equations alone.

The terms in the Rothman code's R-K scheme seem more poorly defined than the terms in the Kawano code's scheme, and the final value of $Y_{n+1}(i)$ is constructed from the matrix equation solutions alone. So I feel the Kawano code's means of carrying out the R-K calculation of $Y_{n+1}(i)$ sounder. I have converted the Rothman code to carry out the scheme in the same way as the Kawano code, resulting in this graph shown in Figure (2), which obeys the equation

$$X_{^4He} = 0.2394508633 - (0.1509902093 \times 10^{-3})d^2$$

The Rothman code now has the second order convergence it should have, and it still converges to the same value of $X_{^4He}$ as in the first order case and approaches the convergence value more quickly than the Kawano code.

4 The Value of Y_{min}

Now we modify the Kawano code so that $Y(i)$ can have a minimum value of 0.0. Figure (3a) shows the graphs of the timesteps Δt_n for the case of $Y_{min} = 0.0$ and $Y_{min} = 10^{-25}$ in the Kawano code. When $Y_{min} = 0.0$ $X_{^4He}$ =0.2392598117. Less than $1.0E^{-6}$ from the final value of 0.2392602277. But the code takes 2791 timesteps to complete a run compared to only 323 timesteps for $Y_{min} = 10^{-25}$. Note that in this early region of the $Y_{min} = 10^{-25}$ code Δt_n is decided almost always by $T/(dT/dt)$

In Figure (3b) I vary the value of Y_{min} from 10^{-25} all the way down to 10^{-95}. As Y_{min} decreases the graphs break away from this straight line dependence at higher temperatures. These breaks correspond to when the ratio $Y(i)/(dY/dt)$ for ^{4}He is small enough to set the value of Δt. (Note: the first hitch at about 8 GK corresponds to the moment when ^{4}He falls out of NSE with the lighter isotopes, and then Δt_n depends on tritium until that falls out of NSE, yielding Δt_n to deuterium up until nucleosynthesis begins at around 0.8 GK. All these dents are illustrated in Figure (3c).) But as I lower Y_{min} by a factor of 10^{-10} the number of steps increases by only about 100 steps, reaching 1020 for the 10^{-95} case. Remember from Equation (15) that

$$\Delta t_{n+1} = dk \min \left[\frac{Y_n(i)}{\frac{dY(i)}{dt}} dk_{fac} \right]$$
$$dk_{fac} = \left(1 + \left(-\frac{\log Y_n(i)}{dt_{fac}} \right)^2 \right)$$

. In the $Y_{min} = 10^{-95}$ case, dk_{fac} can increase $Y/(dY/dt)$ for abundances close to Y_{min} by a factor of four or five. So these ratios don't get small enough to determine Δt_n until an order of magnitude higher than the $Y_{min} = 0.0$ case, which doesn't use dk_{fac}. Figure (4) compares the timesteps for the Y_{min} case with and without using dk_{fac}, the with case using 10^{-99} to calculate dt_{fac}. The run then has 1067 timesteps vs. 2791, and $X_{^4He} = 0.2391949247$. Further from $X_{^4He}$ than without the factor, but better than the non-zero Y_{min} cases.

Figure (5a) graphs the $X_{^4He}$ dependence on Y_{min}. As one decreases Y_{min} $X_{^4He}$ does seem to converge to $(X_{^4He})_0$ So I graphed $\log((X_{^4He})_0 - X_{^4He})$ vs $\log(Y_{min})$ in Figure (5b), but I didn't find as strong a linear dependence as I did in the convergence tests. Similary, in Figure (5c) I graphed $\log((X_{^4He})_0 - X_{^4He})$ vs. 1 / the # of steps, normalized to equal 1, hoping to match the convergence graphs. Some of the lower steps seem to show some linearity, but it corresponds to 3 instead of two. So the convergence of $X_{^4He}$ in terms of Y_{min} is unclear.

The changes in $X_{^4He}$ and in the # of timesteps depend on how much influence the abundances have in determining Δt_n. But I've found that if I lower the dk for $T/(dT/dt)$ to 0.01 I can stabilize $X_{^4He}$ Figure (6) shows this very vividly. I vary Y_{min} from 10^{-25} down to 10^{-95}, but $X_{^4He}$ varies only in the sixth digit all through this range. The # of timesteps varies from 1009 to 1287, comparable to the 1020 and 1067 timesteps used for the graphs in Figure (4). Figure (7) shows that in this case $T/(dT/dt)$ dominates the determination of Δt_n for nearly all of the run.

I've also modified the Rothman code to have a non-zero Y_{min}. But for the Rothman code to match the same number of timesteps as the Kawano code, I have to multiply the ratio $\rho_{\gamma+e}/\dot{\rho}_{\gamma+e}$ by a factor of 4.

$$4\frac{\rho_{\gamma+e}}{\dot{\rho}_{\gamma+e}} \approx \frac{T}{\dot{T}}$$

At the highest temperatures both ρ_γ and ρ_e have a T^4 dependence, and at the lowest ρ_e is nearly zero. But of course this equation isn't accurate for the temperatures in between, but is still a good approximation. With this modification, the Rothman code exhibits similar behavior to the Kawano code. Figure (8) is the Rothman equivalent to Figure (5a) for the Kawano code. But the corresponding values of $X_{^4He}$ are much lower than in the Kawano code. Figure (9) shows that $X_{^4He}$ in the Rothman code starts out slightly higher than in the Kawano code, but then ^{4}He falls out of NSE slightly earlier, and the Rothman $X_{^4He}$ trails the Kawano $X_{^4He}$ from then on. Figure (10) shows Δt_n for various values of Y_{min}. The graph looks very similar to Figure (3b), but ^{4}He starts to determine Δt_n at later times than in the Kawano code. And Figure (11) shows that we can greatly reduce the $X_{^4He}$ dependence on Y_{min} by adding in another factor of 0.1 to $\rho_{\gamma+e}/\dot{\rho}_{\gamma+e}$ in the Rothman code as well, though not as much as in the Kawano code.

5 Final Results

Table (3) shows some final mass fractions where both codes have $Y_{min} = 0.0$ and use dk_{fac} next to $Y/\dot{Y}$ and 0.1 next to $T/\dot{T}$. Both runs took about 1320 timesteps to complete a run. The $X_{^4He}$ now agree with each other to within an order of 0.0001, and the other abundances for the most part are in closer agreement with each other than in Table (1). And Table (4) shows the mass fractions for the case of $Y_{min} = 10^{-25}$. The $X_{^4He}$'s also agree within 0.0001 with each other and the mass fractions from the $Y_{min} = 0.0$ case.

With the convergence order and effect of Y_{min} reduced, one can compare the effects of the differences in equations between the codes. For instance, Kawano's Equation (16) is very similar to Rothman's Equation (8), except that the Kawano code uses $(\partial\rho_e/\partial T) + (\partial\rho_e/\partial\phi_e)(\partial\phi_e/\partial T)$ for $d\rho_e/dT$ instead of just $(\partial\rho_e/\partial T)$ in the Rothman code. But actually, the $(\partial\rho_e/\partial\phi_e)(\partial\phi_e/\partial T)$ term is as much as 17 orders of magnitude smaller than the other term. The two terms don't come to the same order of magnitude until around T =0.2 GK, long after most abundances have reached their final values and a temperature where $d\rho_\gamma/dT$ dominates over both terms. The Kawano code's $\rho_b = h_v(15a/\pi^2)T^3$ has both h_v and T time dependent. But since we derived dh_v/dt from $h_v = const/(RT)^3$ the code's ρ_b should still have the same time evolution as the Rothman code's ρ_b. And ϕ_e pretty much plays a significant part only in the initialization of ρ_b and its own evolution. Figure (12) shows a graph of ρ_e, p_e, ρ_ν, ρ_b, $\lambda(n \to p)$, and $\lambda(p \to n)$. All the graphs coincide for both codes. And Figures (13a) and (13b) graph $X_{^4He}$ for both codes for the cases of $Y_{min} = 0.0$ and $Y_{min} = 10^{-25}$ respectively. Once set far apart, the two graphs now nearly coincide with one another.

The convergences of $X_{^4He}$ for each code now obey the equation

$$X_{^4He} = 0.2392673182 - (0.3253724834 \times 10^{-4})d^2$$

for the Kawano code and

$$X_{^4He} = 0.2394508628 - (0.1203603394 \times 10^{-3})d^2$$

for the Rothman code. So while $X_{^4He}$ for both of the codes nearly equal each other at around 1000 timesteps, they still converge to the different values as before. This difference seems to be the main real effect of using one code over the other.

6 Future Research

I plan to further compare the differences between the Rothman and the Kawano codes to see how real the disagreement in convergent values is. The codes still have different means of calculating numerical integrals and of solving the matrix equations. Perhaps one of those differences affects the convergent values. Likewise they might explain why the Rothman code tends to be more sensitive to changes in the coding than the Kawano code. I also plan to determine whether or not the linearization of Equation (5) is still necessary. We now have packages like LSODA which could calculate $\dot{Y}(i)$ directly and solve for the first order ODE that is Equation (5) even though (5) is very stiff. Nevertheless, the codes are now in better agreement than they were before. So now I can implement small effects like the

Neutrino Heating effect and investigate anisotropic and inhomogeneous models with greater confidence in the final results.