Comparison of Two Distinct BBN Codes

by: Juan F. Lara
Center for Relativity
University of Texas at Austin

Motivation

- BBN Codes with different evolution schemes may give answers that don't agree.
- The disagreement could be comparable or larger than the influence of physical effects one wants to investigate.
- The scheme in a code may be erroneously conceived.
- Improved accuracy in computers may render approximations in code obsolete.

Trace of Each Program

Sources

Kawano, L. Fermilab-Pub-92/04-A (1992)

Rothman, T., Matzner, R. Physical Review D. **30**, 1649 (1984).

Thomas, D., Schramm. D., Olive, K.A., Fields, B.D. Astrophysical Journal **406** 569 (1993)

Wagoner, R. V. Astrophysical Journal Supplement Series **18**, 247 (1969).

The Rothman Code

The Rothman code uses following equations to initiallize abundances and thermal values.

$$
\begin{aligned}
a &= \frac{\pi^2}{15}\frac{k^4}{(c\hbar)^3} \\
E &= \frac{m_e}{kT} \\
\rho_\gamma &= aT^4 \\
n_\gamma &= \frac{30(1.202)}{k\pi^4}aT^3 \\
\rho_e &= \frac{30}{\pi^4}aT^4\int_0^\infty dX\frac{X^2\sqrt{X^2+E^2}}{e^{\sqrt{X^2+E^2}}+1} \\
P_e &= \frac{10}{\pi^4}aT^4\int_0^\infty dX\frac{X^4}{\sqrt{X^2+E^2}\left(e^{\sqrt{X^2+E^2}}+1\right)} \\
n_e &= \frac{30}{k\pi^4}aT^3\int_0^\infty dX\frac{X^2}{e^{\sqrt{X^2+E^2}}+1} \\
\rho_\nu &= \frac{7}{8}aT_\nu^4 \\
\rho_\nu R^4 &= const. \qquad (1) \\
\rho_b R^3 &= const. \qquad (2) \\
\rho_b &= m_N n_\gamma \eta \qquad (3) \\
Y_n(100GK) &= \frac{1}{e^{\frac{m_n-m_p}{kT}}+1} \\
Y_p(100GK) &= 1-Y_p \\
Y_i(100GK) &= 0.0
\end{aligned}
$$

The code applies the Second-Order Runge-Kutta method to calculating abundances $Y(i)$.

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

The equation

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

applies to reactions of the form:

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

But as of now one has to linearize Equation (5)

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

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

The program applies the R-K method to $\ln R$ and $\rho_{\gamma+e}$ as well.

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

using equations derived from the Einstein field equations and the energy conservation law.

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

The First Runge-Kutta Step (Rothman)

Equations (6) and (7) form the following matrix equation.

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

Then the code evolves to the interim $\tilde{}$ position using timestep Δt_n.

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

It uses Newton's method to determine $\tilde{T}_n$ from $\tilde{\rho}_{\gamma+e}$.

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

and then plugs in $\tilde{T}_n$ in the equations in the first transpancy.

The Second Runge-Kutta Step (Rothman)

The code then calculates $\dot{R}/R$ and $\dot{\rho}_{\gamma+e}$ and performs the second R-K step on the two thermal quantities.

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

From which the code calculates a final value of T_{n+1}. It uses T_{n+1} in the second matrix equation for $Y(i)$.

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

The Rothman code ends a step by calculating the next timestep Δt_{n+1}

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

The Kawano Code

The Kawano Code uses the Nuclear Statistical Equilibrium approximation

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

to initiallize $Y(d)$. Then it sets.

$$Y(i) = 10^{-25}$$

To determine ρ_b, the code assumes a non-zero value of ϕ_e and uses charge conservation between protons and electrons.

$$
\begin{aligned}
n_{e^-} - n_{e^+} &= \frac{\rho_b S}{m_N} && (14) \\
S &= \sum_i Z_i Y(i) \\
n_{e^-} - n_{e^+} &= \frac{2}{15}\frac{m_e^3}{k^4}\sum_{n=1}^{\infty}\sinh(n\phi_e)L(nz) \\
\rho_b &= h_v a \frac{15}{\pi^2}T^3 && (15)
\end{aligned}
$$

The code defines h_v such that it satisfies Equation (14).

The Kawano code will then use the R-K method to evolve

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

along with $Y(i)$. The code derives the expression for $\dot{T}$ from the energy conservation Equation (8), but also takes into account ρ_e's dependence on ϕ_e.

$$\begin{aligned}
\frac{d\rho_{\gamma+e}}{dT}\frac{dT}{dt} &= -3\frac{\dot{R}}{R}(p_{\gamma+e} + \rho_{\gamma+e}) \\
\frac{dT}{dt} &= -\frac{3\frac{\dot{R}}{R}(p_{\gamma+e} + \rho_{\gamma+e})}{\frac{d\rho_\gamma}{dT} + \frac{d\rho_e}{dT}} \\
\frac{d\rho_e}{dT} &= \frac{\partial\rho_e}{\partial T} + \frac{\partial\rho_e}{\partial\phi_e}\frac{\partial\phi_e}{\partial T}
\end{aligned} \tag{16}$$

It derives $\dot{h}_v$ from the equation $h_v = \mathrm{const}/(\mathrm{RT})^3$

$$\frac{dh_v}{dt} = -3h_v\left(\frac{1}{R}\frac{dR}{dt} + \frac{1}{T}\frac{dT}{dt}\right) \tag{17}$$

and for $\dot{\phi}_e$:

$$\frac{d\phi_e}{dt} = \frac{\partial\phi_e}{\partial T}\frac{dT}{dt} + \frac{\partial\phi_e}{\partial R}\frac{dR}{dt} + \frac{\partial\phi_e}{\partial S}\frac{dS}{dt}$$

the derivatives all derived from Equation (14).

The First Runge-Kutta Step (Kawano)

For the first matrix equation, the Kawano code uses Y_{n-1} instead of Y_n on the left hand side, and uses Δt_{n-1} from the previous step.

$$\begin{aligned} Y_{n-1}(i) &= Q_{ij}(Y_n, [ij](T_n), [kl](T_n)) \times Y_{n\alpha}(j) \\ \frac{dY(i)}{dt}(t, Y_n(i)) &= \frac{Y_{n\alpha}(i) - Y_{n-1}(i)}{\Delta t_{n-1}} \end{aligned} \tag{18}$$

It then calculates $\dot{T}$, $\dot{h}_v$ and $\dot{\phi}_e$. **At this point** it calculates the new Δt_n

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

which is used to get the interim $\tilde{}$ values.

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

The code then updates all thermal quantities using the interim values $\tilde{T}$, $\tilde{h}_v$, and $\tilde{\phi}_e$, using the same equations as in the Rothman code, except for ρ_b and ρ_ν. Instead it uses the entropy conservation equation

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

eliminating R by using $RT_\nu =$ const and comparing s with s_o when $T = T_\nu = 100$ GK.

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

The Second Runge-Kutta Step (Kawano)

The second matrix equation uses $Y_n(i)$ on the left-hand-side, and $\tilde{T}_n$ instead of T_{n+1} in the matrix

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

and Δt_n to determine the second $\dot{Y}(i)$. Then, after calculating $\dot{T}$, $\dot{h}_v$ and $\dot{\phi}_e$ using the interim thermal values.:

$$\begin{aligned} Y_{n+1}(i) &= Y_n(i) + \frac{1}{2}\left[\frac{dY(i)}{dt}(t, Y_n) + \frac{dY(i)}{dt}(t+\Delta t_n, Y_{n+1})\right]\Delta t_n \qquad (24) \\ T_{n+1} &= T_n + \frac{1}{2}\left[\frac{dT}{dt}(t, \rho_n) + \frac{dT}{dt}(t+\Delta t_n, \tilde{\rho}_n)\right]\Delta t_n \\ (h_v)_{n+1} &= (h_v)_n + \frac{1}{2}\left[\frac{dh_v}{dt}(t, R_n) + \frac{dh_v}{dt}(t+\Delta t, \tilde{R}_n)\right]\Delta t_n \\ (\phi_e)_{n+1} &= (\phi_e)_n + \frac{1}{2}\left[\frac{d\phi_e}{dt}(t, T_n, R_n, S_n) + \frac{d\phi_e}{dt}(t+\Delta t, \tilde{T}_n, \tilde{R}_n, \tilde{S}_n)\right]\Delta t_n \end{aligned}$$

and the code uses T_{n+1}, $(h_v)_{n+1}$ and $(\phi_e)_{n+1}$ to find the final values of $\rho_{\gamma,e,\nu,b}$ and p_e.

isotope	Kawano X	Rothman X
n	$0.4203325866 \times 10^{-15}$	$0.5898987182 \times 10^{-15}$
p	0.7629357063	0.7601374531
d	$0.1203806246 \times 10^{-3}$	$0.1239148256 \times 10^{-3}$
^{3}H	$0.5791818972 \times 10^{-6}$	$0.5993673829 \times 10^{-6}$
^{3}He	$0.3616593549 \times 10^{-4}$	$0.364445127 \times 10^{-4}$
^{4}He	0.2369001572	0.2397015878
^{6}Li	$0.1421679767 \times 10^{-12}$	$0.1485212581 \times 10^{-12}$
^{7}Li	$0.3349991579 \times 10^{-9}$	$0.3496296345 \times 10^{-9}$
^{7}Be	$0.286454318 \times 10^{-9}$	$0.2679050102 \times 10^{-9}$
^{9}Be	$0.1036417579 \times 10^{-16}$	$0.1126556133 \times 10^{-16}$
^{10}B	$0.8927007279 \times 10^{-18}$	$0.9643350569 \times 10^{-18}$
^{11}B	$0.9314884547 \times 10^{-16}$	$0.91442805 \times 10^{-16}$
^{12}C	$0.2898262833 \times 10^{-13}$	$0.3048347851 \times 10^{-13}$
^{13}C	$0.1981016226 \times 10^{-14}$	$0.206376067 \times 10^{-14}$
^{14}C	$0.6230054685 \times 10^{-13}$	$0.6588234647 \times 10^{-13}$
^{14}N	$0.1275229996 \times 10^{-12}$	$0.1326174221 \times 10^{-12}$
^{16}O	$0.7290963822 \times 10^{-17}$	$0.7730848001 \times 10^{-17}$
^{19}F	$0.3484143146 \times 10^{-21}$	$0.7549164005 \times 10^{-24}$
^{20}F	$0.5477065467 \times 10^{-21}$	$0.3759130859 \times 10^{-24}$
^{21}Ne	$0.6698206755 \times 10^{-21}$	$0.5517307027 \times 10^{-23}$

Table (1): Mass Fractions from each code.

Convergence of the Codes

$$d \equiv \#\text{ of divisions per timestep}$$

At $\eta_{10} = 3.0$ $X_{^4He}$ for the Kawano code obeys the equation

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

whereas $X_{^4He}$ for the Rothman code obeys the equation

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

The Semi-Implicit Euler Method

To ensure stability, a BBN code is supposed to calculate $\dot{Y}$ based on this expansion.

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

Equations (18) and (23) correspond to Equation (25) very clearly.

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

Plugging these expressions into Equations (20) and (24).

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

But Equations (9) and (11) in the Rothman code do not match up as clearly.

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

Y_n on the left hand side of both equations would imply that both $Y_{n\alpha}(i)$ and $Y_{n\beta}(i)$ approximate Y_{n+1}, though they don't equal each other because the equations use different matrices. Further, when one plugs into Equations (10) and (12)

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

So in the Rothman code, $Y_{n+1}(i)$ is the average of the two first order Euler values $Y_{n\alpha}(i)$ and $Y_{n\beta}(i)$.

When Equations (9), (11) are modified to resemble (18) and (23), and when the point when the Rothman code calculates the timestep is changed to match that of the Kawano code, $X_{^4He}$ obeys the equation.

$$X_{^4He} = 0.2394508633 - (0.1509902093 \times 10^{-3})d^2 \qquad (27)$$

The Value of Y_{min}

Y_{min}	Kawano	Kawano (0.1)	Rothman	Rothman (0.1)
10^{-25}	323	1009	313	1029
10^{-35}	463	1033	434	1051
10^{-45}	562	1055	542	1069
10^{-55}	621	1078	589	1088
10^{-65}	701	1110	684	1118
10^{-75}	804	1159	787	1164
10^{-85}	913	1218	897	1220
10^{-95}	1020	1287	1004	1286
0.0	2791	2882	2772	2879
0.0 (dk_{fac})	1067	1321	1050	1320

Table (2): # of timesteps

The third and fifth columns have an extra factor of 0.1 multiplied to $dk(T/\dot{T})$ or $dk(\rho_{\gamma+e}/\dot{\rho}_{\gamma+e})$.

In the last line, the factor dk_{fac} is included in $Y(i)/\dot{Y}(i)$ for $Y_{min} =$ 0.0 as well, where:

$$\begin{aligned} dk_{fac} &= \left(1 + \left(-\frac{\log Y_n(i)}{dt_{fac}}\right)^2\right) \\ dt_{fac} &= -\frac{\log 10^{-99}}{\frac{dk_{max}}{dk} - 1} \end{aligned}$$

For the Rothman code to behave like the Kawano code, one has to multiply a factor of 4 to $dk(\rho_{\gamma+e}/\dot{\rho}_{\gamma+e})$.

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

Final Results

isotope	Kawano X	Rothman X
n	$0.5127256402 \times 10^{-15}$	$0.5271994410 \times 10^{-15}$
p	0.7606071494	0.7605121399
d	$0.1214409373 \times 10^{-3}$	$0.1222110665 \times 10^{-3}$
^{3}H	$0.5869118485 \times 10^{-6}$	$0.5905659297 \times 10^{-6}$
^{3}He	$0.3615054989 \times 10^{-4}$	$0.3624197334 \times 10^{-4}$
^{4}He	0.2392346716	0.2393288161
^{6}Li	$0.1452312342 \times 10^{-12}$	$0.1461970786 \times 10^{-12}$
^{7}Li	$0.3424567551 \times 10^{-9}$	$0.3445304655 \times 10^{-9}$
^{7}Be	$0.2810372859 \times 10^{-9}$	$0.277271461 \times 10^{-9}$
^{9}Be	$0.1078010758 \times 10^{-16}$	$0.1091991775 \times 10^{-16}$
^{10}B	$0.9337145278 \times 10^{-18}$	$0.9425546233 \times 10^{-18}$
^{11}B	$0.9855540716 \times 10^{-16}$	$0.9595478082 \times 10^{-16}$
^{12}C	$0.3026820198 \times 10^{-13}$	$0.30138561 \times 10^{-13}$
^{13}C	$0.2065161640 \times 10^{-14}$	$0.2045059261 \times 10^{-14}$
^{14}C	$0.6519717631 \times 10^{-13}$	$0.6515513573 \times 10^{-13}$
^{14}N	$0.1322155965 \times 10^{-12}$	$0.1320911107 \times 10^{-12}$
^{16}O	$0.7673887093 \times 10^{-17}$	$0.7632076774 \times 10^{-17}$
^{19}F	$0.7522419741 \times 10^{-24}$	$0.7478911071 \times 10^{-24}$
^{20}F	$0.3472762215 \times 10^{-24}$	$0.3558662855 \times 10^{-24}$
^{21}Ne	$0.5388613183 \times 10^{-23}$	$0.5374132790 \times 10^{-23}$

Table (3): Mass Fractions for Modified Codes ($Y_{min} = 0.0$)

isotope	Kawano X	Rothman X
n	$0.5363013418 \times 10^{-15}$	$0.5572047404 \times 10^{-15}$
p	0.7606145704	0.7605549812
d	$0.1211888661 \times 10^{-3}$	$0.122088352 \times 10^{-3}$
^{3}H	$0.5856634466 \times 10^{-6}$	$0.5899267748 \times 10^{-6}$
^{3}He	$0.3611129967 \times 10^{-4}$	$0.3621882542 \times 10^{-4}$
^{4}He	0.2392205836	0.2392793527
^{6}Li	$0.1449150981 \times 10^{-12}$	$0.1460020872 \times 10^{-12}$
^{7}Li	$0.3418406159 \times 10^{-9}$	$0.3441658463 \times 10^{-9}$
^{7}Be	$0.2827437965 \times 10^{-9}$	$0.2783319238 \times 10^{-9}$
^{9}Be	$0.1073550777 \times 10^{-16}$	$0.1089450724 \times 10^{-16}$
^{10}B	$0.9316553017 \times 10^{-18}$	$0.9415169158 \times 10^{-18}$
^{11}B	$0.9986427123 \times 10^{-16}$	$0.9676443571 \times 10^{-16}$
^{12}C	$0.3028298366 \times 10^{-13}$	$0.3011196604 \times 10^{-13}$
^{13}C	$0.2068589367 \times 10^{-14}$	$0.2044492789 \times 10^{-14}$
^{14}C	$0.6516868023 \times 10^{-13}$	$0.6505998791 \times 10^{-13}$
^{14}N	$0.1321917464 \times 10^{-12}$	$0.1319412529 \times 10^{-12}$
^{16}O	$0.7650373838 \times 10^{-17}$	$0.759474919 \times 10^{-17}$
^{19}F	$0.1618658902 \times 10^{-20}$	$0.1617387233 \times 10^{-20}$
^{20}F	$0.2432645701 \times 10^{-20}$	$0.2432641727 \times 10^{-20}$
^{21}Ne	$0.1938939934 \times 10^{-20}$	$0.1938889787 \times 10^{-20}$

Table (4): Mass Fractions for Modified Codes ($Y_{min} = 10^{-25}$)

$$\frac{\partial \rho_e}{\partial T} \gg \frac{\partial \rho_e}{\partial \phi_e}\frac{\partial \phi_e}{\partial T}$$

all through the time when $d\rho_e/dT$ is significant. And the use of h_v and ϕ_e doesn't seem to significantly alter the evolution of thermal variables.

The convergence of $X_{^4He}$ now obeys 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.

Future Research

- Check to see if difference in convergent values between the codes is still due to some error.
- Check to see why the Rothman code seems more sensitive to change than the Kawano code.
- See if linearization of Equation (5) is still necessary.
- Add in new physical effects like the neutrino heating effects to improved codes.