Webnucleo Technical Report: Network Calculations with libnucnet

Bradley S. Meyer

April 17, 2011

This technical report describes some details of solving nuclear reaction networks with the libnucnet module.

1 The Network Equations

A nuclear reaction network is a collection of nuclear species and reactions among them. We denote the number density of species i in the network by n_i. It is convenient to define the number density as

$$n_i = \rho N_A Y_i, \tag{1}$$

where ρ is the mass density (grams/cm^3), N_A is Avogadro's number, and Y_i is the abundance of species i per nucleon.

The number density of a species changes with time both due to nuclear reactions and to a change in the volume of the material. For purposes of illustration, let us assume a simple three-element network with species 1, 2, and 3 and two reactions:

$$1 + 2 \rightleftharpoons 3 + \gamma \tag{2}$$

and

$$2 \rightarrow 1 \tag{3}$$

The former reaction is a radiative capture reaction (with its reverse disintegration). The latter reaction is a decay reaction.

The differential equation describing the rate of change of species 1 is:

$$\frac{dn_1}{dt} = -n_1 n_2 \langle \sigma v \rangle + \lambda_\gamma n_3 + \lambda n_2 - \frac{n_1}{\tau}. \tag{4}$$

Here σ is the interaction cross section for a pair of nuclei of type 1 and 2. v is the relative velocity of the interacting nuclei. The term $\langle \sigma v \rangle$ is thus the thermally averaged cross section for the interaction of a pair of nuclei of type 1 and 2. By multiplying this term by the number densities of the nuclei of type 1 and 2, we get the total number of interactions of nuclei of type 1 and 2 per unit volume per unit time. The negative sign indicates that such interactions decrease the number density of species 1. The term λ_γ is the rate of disintegration of a nucleus of type 3. This is the reverse reaction to the first and increases the number of nuclei of type 1 (and 2). The term λ is the rate for a nucleus of type 2 to decay into a nucleus of type 1. Finally, the term $-\frac{n_1}{\tau}$ represents the decrease in the number density of species i simply due to the expansion of the system. The expansion timescale τ is defined by

$$\frac{1}{\tau} \equiv -\frac{1}{\rho}\frac{d\rho}{dt}. \tag{5}$$

The differential equations for the number densities of species 2 and 3 are:

$$\frac{dn_2}{dt} = -n_1 n_2 \langle \sigma v \rangle + \lambda_\gamma n_3 - \lambda n_2 - \frac{n_2}{\tau} \tag{6}$$

and

$$\frac{dn_3}{dt} = n_1 n_2 \langle \sigma v \rangle - \lambda_\gamma n_3 - \frac{n_3}{\tau}. \tag{7}$$

It is convenient to remove the dependence on density by substituting Eqs. (1) and (5) in Eqs. (4), (6), and (7). The results are:

$$\frac{dY_1}{dt} = -N_A \langle \sigma v \rangle \rho Y_1 Y_2 + \lambda_\gamma Y_3 + \lambda Y_2 \tag{8}$$

and

$$\frac{dY_2}{dt} = -N_A \langle \sigma v \rangle \rho Y_1 Y_2 + \lambda_\gamma Y_3 - \lambda Y_2 \tag{9}$$

and

$$\frac{dY_3}{dt} = N_A \langle \sigma v \rangle \rho Y_1 Y_2 - \lambda_\gamma Y_3. \tag{10}$$

These are the appropriate forms of the network equations to solve. We note that the equations must be modified to account for identical reactants or products. libnucnet routines automatically account for these.

2 The Matrix

If we know the abundances of the species at time t, we seek their abundances at time $t + \Delta t$. This involves integrating the coupled differential equations of the type in Eqs. (8-10) over time step Δt. Because the system of equations is very stiff (i.e., the shortest timescale is much less than the longest), we need to solve the equations implicitly. To do so, we first finite difference in time and use the definition

$$\Delta Y_i = Y_i(t + \Delta t) - Y_i(t). \tag{11}$$

Implicit differentiation means that we compute derivatives at time $t + \Delta t$. In this case, Eq. (8) becomes

$$\frac{\Delta Y_1}{\Delta t} = -N_A \langle \sigma v \rangle \rho Y_1(t+\Delta t) Y_2(t+\Delta t) + \lambda_\gamma Y_3(t+\Delta t) + \lambda Y_2(t+\Delta t) \tag{12}$$

where the rates (such as $N_A\langle\sigma v\rangle\rho$) are computed at time $t+\Delta t$. We may then rearrange this equation to read

$$f_1 = \left(Y_1(t+\Delta t) - Y_1(t)\right)/\Delta t + N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t)Y_2(t+\Delta t) \\ - \lambda_\gamma Y_3(t+\Delta t) - \lambda Y_2(t+\Delta t) = 0. \tag{13}$$

Similarly, we find

$$f_2 = \left(Y_2(t+\Delta t) - Y_2(t)\right)/\Delta t + N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t)Y_2(t+\Delta t) \\ - \lambda_\gamma Y_3(t+\Delta t) + \lambda Y_2(t+\Delta t) = 0. \tag{14}$$

and

$$f_3 = \left(Y_3(t+\Delta t) - Y_3(t)\right)/\Delta t - N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t)Y_2(t+\Delta t) \\ + \lambda_\gamma Y_3(t+\Delta t) = 0. \tag{15}$$

The goal is to solve for the abundances at $t+\Delta t$ such that $f_1 = f_2 = f_3 = 0$. We may do this by the Newton-Raphson method.

We begin by considering a Taylor series expansion of the functions f_i:

$$f_i(\mathbf{Y}+\delta\mathbf{Y}) = f_i(\mathbf{Y}) + \sum_{k=1}^{N} \frac{\partial f_i}{\partial Y_j}\delta Y_j + \mathcal{O}(\delta\mathbf{Y}^2), \tag{16}$$

where N is the total number of equations (species), $\mathbf{Y}$ is the vector of all Y's, and δY_j represents the change in Y_j over one Newton-Raphson iteration. By neglecting terms higher order than linear and assuming $f_i(\mathbf{Y}+\delta\mathbf{Y}) = 0$, we may find

$$\sum_j \alpha_{ij}\delta Y_j = \beta_i \tag{17}$$

where

$$\alpha_{ij} = \frac{\partial f_i}{\partial Y_j} \tag{18}$$

and

$$\beta_i = -f_i. \tag{19}$$

The N simultaneous equations in Eq. (17) make up a matrix equation $Ax = b$. The matrix A is known as the Jacobian matrix and has elements α_{ij} given in Eq. (18). The right-hand-side vector b has elements β_i given in Eq. (19). The solution vector x contains the updates δY_i to the abundance. The matrix equation is solved by any practical matrix solution method (see below). The corrections are then added to the solution vector:

$$Y_i^{new} = Y_i^{old} + \delta Y_i \tag{20}$$

for $i = 1$ to N and the process is iterated to convergence, that is, until the δY_i's are much smaller than the Y_i's and no longer change the solution.

For our three-element problem, our matrix is

$$A = \begin{pmatrix} \frac{1}{\Delta t} + N_A\langle\sigma v\rangle\rho Y_2(t+\Delta t) & N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t) - \lambda & -\lambda_\gamma \\ N_A\langle\sigma v\rangle\rho Y_2(t+\Delta t) & \frac{1}{\Delta t} + N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t) + \lambda & -\lambda_\gamma \\ -N_A\langle\sigma v\rangle\rho Y_2(t+\Delta t) & N_A\langle\sigma v\rangle\rho Y_1(t+\Delta t) & \frac{1}{\Delta t} + \lambda_\gamma \end{pmatrix} \tag{21}$$

The matrix A changes from one Newton-Raphson iteration to the next as the values Y_i are improved. Our right-hand side vector is

$$b = \begin{pmatrix} -f_1 \\ -f_2 \\ -f_3 \end{pmatrix} \tag{22}$$

where the values of f_1, f_2, f_3 are evaluated from Eqs. (13-15) using the current values of the Y's. The solution vector x is given by

$$x = \begin{pmatrix} \delta Y_1 \\ \delta Y_2 \\ \delta Y_3 \end{pmatrix} \tag{23}$$

Our matrix equation is thus $Ax = b$. We iterate until the elements of x converge to zero.

3 The Single Zone Example

The single zone nuclear network example of the libnucnet distribution uses the above procedure to evolve the abundances. This example uses an exponential expansion of the temperature and density such that the density $\rho(t)$ behaves as

$$\rho(t) = \rho_0 \exp(-t/\tau), \tag{24}$$

where ρ_0 is the initial density and τ is the expansion timescale. The temperature $T_9 = T/10^9$ K is such that $\rho \propto T_9^3$. The example reads nuclear, reaction, and initial mass fraction data from the input XML file along with the initial density, T_9, and τ from the command line. The user may also choose XPath expressions to limit the species or reactions present in the network. A $\tau = 0$ indicates a static calculation (i.e., constant temperature and density).

The example uses libnucnet API routines to compute the rates at time $t+\Delta t$, the Jacobian matrix [Eq. (18)], and the right-hand-side vector [Eq. (19)]. It then uses the wn_matrix arrow solver or GNU Scientific Library (gsl) matrix routines to solve the resulting matrix equations [Eq. (17)]. Newton-Raphson iterations are performed until the abundances change Y converge. Convergence is usually rapid (two or three iterations). The convergence criteria are set by changing the parameters D_MIN and D_YMIN. The maximum number of iterations allowed is set by I_ITMAX.

Upon convergence of the abundances, the example then updates the timestep interval Δt with the appropriate API routine and moves on to the next timestep.

The example prints out the abundances and the abundance changes for each species with abundance greater than a certain value every m timesteps, where m is input from the user. The example also outputs the difference between the sum of the mass fractions and unity. Since the sum of all mass fractions should be unity, 1 minus this sum should always be small.

4 Multi-Zone Example

libnucnet easily handles multi-zone networks, as the multi-zone example in the libnucnet distribution demonstrates. This example uses a 1-D set of zones such that each zone mixes with its two neighbors–the previous zone and the next zone. The mixing time is the same between all zones. The nuclear data, reaction data, and abundance data are read from an input xml file while the temperature and density of the zones are read from an input text file. The mixing time τ_{mix} between zones and the duration of the calculation (in seconds) are entered from the command line. The user can also enter XPath parameters to limit the species or reactions included in the calculation.

The rate of change of the abundance of species i in zone k is given by

$$\frac{dY_i^{(k)}}{dt} = (nuclear\ reaction\ terms) + \frac{Y_i^{(k-1)}}{\tau_{mix}} - 2\frac{Y_i^{(k)}}{\tau_{mix}} + \frac{Y_i^{(k+1)}}{\tau_{mix}}, \tag{25}$$

where the *nuclear reaction terms* are the normal terms in the single-zone network. For the first zone ($k = 1$), the equation reads

$$\frac{dY_i^{(1)}}{dt} = (nuclear\ reaction\ terms) - \frac{Y_i^{(1)}}{\tau_{mix}} + \frac{Y_i^{(2)}}{\tau_{mix}} \tag{26}$$

since there is no previous zone with which to mix. For the last zone ($k = M$), the equation reads

$$\frac{dY_i^{(M)}}{dt} = (nuclear\ reaction\ terms) - \frac{Y_i^{(M)}}{\tau_{mix}} + \frac{Y_i^{(M-1)}}{\tau_{mix}} \tag{27}$$

since there is no next zone with which to mix. When all zones are incorporated into a single matrix, the result is a block diagonal matrix with a lower band giving the mixing between the given zone and the previous one and an upper band giving the mixing between the given zone and the next one.

The example loops over all the zones setting up the single-zone Jacobian for each zone and then inserting that into the matrix for all zones. Once that is done, the mixing terms are inserted into the matrix. The full multi-zone matrix equation is then solved and the results are then re-apportioned into the separate zones. The example uses Newton-Raphson iterations and, upon convergence, uses the updateTimeStep routine to find the next timestep interval. The example prints out the abundances above a threshold for all zones every timestep.

5 Matrix Solver

The single-zone and multi-zone examples provide templates for using libnucnet in more complicated codes. Users will probably want to use their own sparse matrix solvers in place of the routines used in the examples. Since the libnucnet routines return the Jacobian matrix in WnMatrix form, the user should call wn_matrix routines to transform the matrix into compressed sparse row form or Yale sparse matrix form. See the wn_matrix API documentation for a description of those routines.