Fermi National Accelerator Laboratory

FERMILAB-PUB-88/34-A

March, 1988

LET'S GO: EARLY UNIVERSE

Guide to Primordial Nucleosynthesis Programming

Lawrence Kawano

The University of Chicago
and
NASA/Fermilab Astrophysics Center

Abstract

This is the owner's manual for the primordial nucleosynthesis program *NUC123*, an updated and revised version of the code of R. V. Wagoner. *NUC123* is faster, better documented, and easier to use than the original code but retains its general computational structure. Presented here is a description of the basic structure of the program, the directions for usage and for generation of output, and some suggestions on how to make modifications to the program.

Operated by Universities Research Association Inc. under contract with the United States Department of Energy

– Program Summary –

Title of program: NUC123

Program obtainable from: Lawrence Kawano (see addresses in back)

Computers: VAX 11/780 and others with FORTRAN 77 compiler

Operating System: VAX/VMS

Programming language used: VAX FORTRAN (FORTRAN 77 except for `DO...END DO` statement)

Storage required: 527 blocks

Peripherials used: terminal for input, terminal or line printer for output

No. of lines in program: 5815

Keywords: primordial nucleosynthesis

Nature of physical problem: Solves for elemental abundances arising from the epoch of primordial nucleosynthesis in the early universe. The initial conditions are entered via menu interface which also serves to allow the selection of the mode of output.

Method of solution: Time evolution of abundances carried out by second-order Runge-Kutta driver. The abundance changes are determined by obtaining a matrix equation from implicit differencing and solving this using Gaussian elimination with back substitution.

Typical running time: Requires about 12 CPU seconds for 1 run with default parameters.

Unusual features of the program: Very detailed documentation within the program and user-friendly menu-driven interface.

– Outline –

I. Introduction.
   A. General Features of New Code.
   B. Subroutine Hierarchy.
   C. Menu Interface.

II. Input Parameters.
   A. Computational Parameters.
   B. Model Parameters.
   C. New Reaction Rates.

III. Running The Code.
   A. Menu Options.
   B. Calculational Algorithms.

IV. Output Options.
   A. Menu Options.
   B. Specific Output File.

V. Modifying The Program.
   A. Using The Adaptive Subroutine.
   B. Suggestions For Implementing Changes.

VI. Acknowledgements.

I. Introduction

A. General Features of New Code

The computer program of R. V. Wagoner (Wagoner 1969 and Wagoner 1972), has proved to be a very useful and indeed, vital tool in research dealing with primordial nucleosynthesis and has become the standard program of the trade. However, for all of its usefulness, it is saddled with a number of severe drawbacks. The program had its origins as a stack of cards in the 1960's so its operation reflects this inconvenient batch mode of operation: the input parameters are read in line by line from a data file and an output listing is produced after every run. Furthermore, some of the input parameters are awkward to use: the program requires the input of the initial chemical potential of the electron rather than the more convenient parameter, the baryon-to-photon ratio. The documentation is skimpy and efforts to decipher the program's workings are further hampered by a web of `GO TO` statements. To add to the confusion, the present version of the program retains vestiges of its days when it utilized a much larger network of reactions to investigate the production of heavy elements in addition to the light elements; loose appendages appear here and there in the program.

These problems are rectified in *NUC123*, an updated and revised version of the nucleosynthesis code. *NUC123* retains the computational structure of the old code but with its inefficiencies taken out, *NUC123* enjoys about a 30% improvement in computation time. The inner workings of the code are illuminated by a generous helping of comments and detailed documentation. And as its name implies, the code has been modified to be run interactively to allow the user greater flexibility and ease.

NUC123 is written in FORTRAN and conforms to the standards of ANSI FORTRAN 77 except for the usage of the `DO...END DO` statement. This was done to limit the number of statement labels. The program was developed on the VAX/VMS system of the Fermilab Vax Cluster.

B. Subroutine Hierarchy

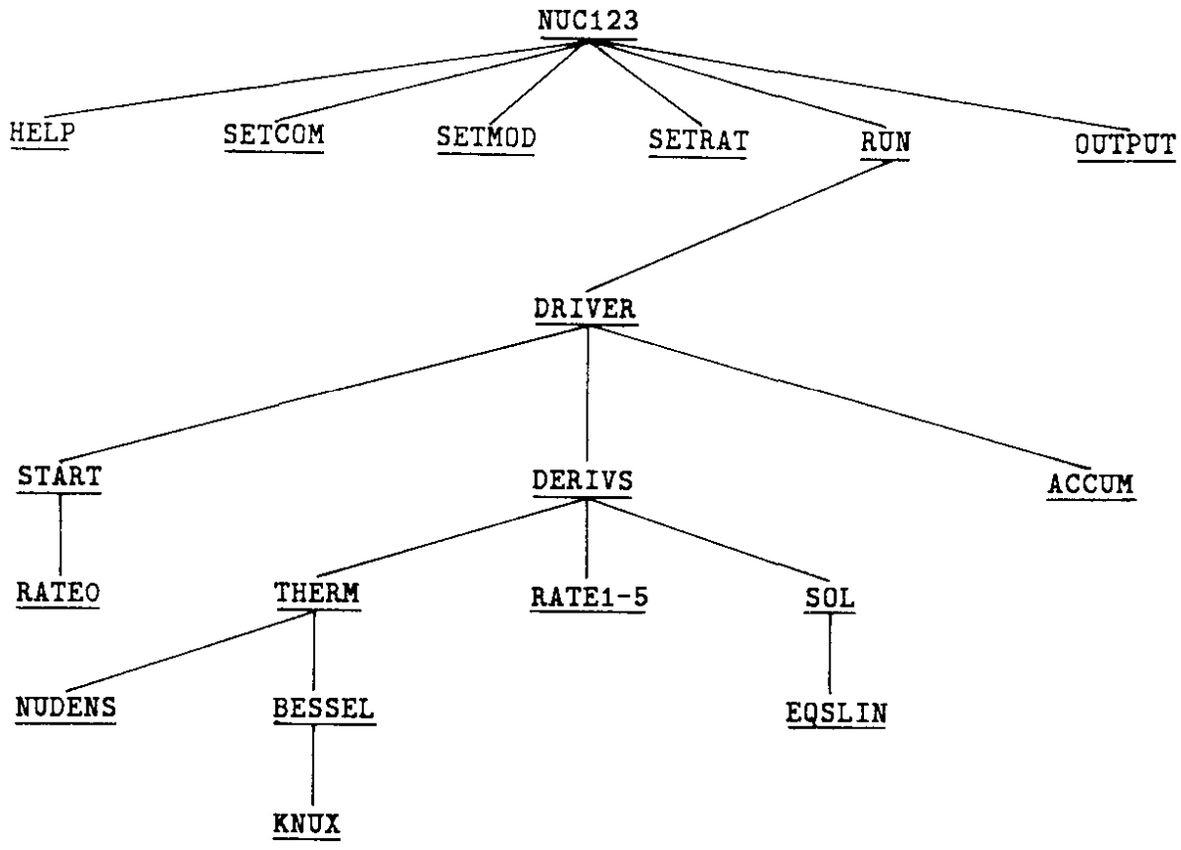


Figure I.1. Subroutine hierarchy in NUC123.

At first glance, the program strikes the observer as a long, intimidating list of one subroutine after another. To allay these fears and to illustrate the the overall structure of the program and the connections between the subroutines, a subroutine hierarchy is given in figure I.1. The picture has been simplified somewhat and not all the connections have been draw in. A complete picture can be inferred from the "linkages" information I have provided at the beginning of the listing of every subroutine.

The subroutines which appear across the top handle various options offered by the main menu in program NUC123 (the menu interface will be discussed in the next section). The Runge-Kutta routine which evolves the abundances is contained in subroutine DRIVER. Beginning values (including weak decay rates from subroutine RATEO) are initialized in START, time derivatives of the temperature and abundances are computed in DERIVS, and these quantities are accumilated after a certain number of Runge-Kutta iterations in ACCUM.

DERIVS utilizes energy densities which are computed in THERM which uses solutions of modified bessel functions from KNUX which come through functions defined in BESSEL. Neutrino energy densities are solved separately in NUDENS. To solve for abundance changes, DERIVS calls upon subroutines RATE1 through RATE5 to compute reaction rate coefficients and using these values, a matrix equation for the time derivative of the abundances is constructed in SOL and solved in EQSLIN.

The old Wagoner code does not contain any equivalent of the subroutines involved in the menu-driven user interface. The main program in the Wagoner code is essentially the same as the combination of the subroutines DRIVER and DERIVS and subroutine ACCUM in *NUC123* is called OUTPUT in the original program. There is no equivalent for subroutine NUDENS or RATE5 and statements in RATE2 through RATE4 are divided into the 2 routines RATE1 and RATE2 for Wagoner's code.

C. Menu Interface

Instead of blandly referring to "the user" each time, I propose to personlize him/her/it by using the name "Rocky." So let's now go join our pal Rocky and journey back to the early universe. To run *NUC123*, all Rocky has to do is to merely compile it (nothing fancy here) and then run it; its that simple ($\sim$ "Just add water") – no messy input files to fuss with, no output file to create. Now *NUC123*, as it is written, is directed to interact actively with Rocky so as to allow him to set input parameters, run modes, and determine output display. So as soon as Rocky starts the program and gets past the greeting, he is shown the main menu as listed in figure I.2.

- HELP section is intended as a refugee camp for terrified users. Now there really shouldn't be any reason why anyone would be terrified, particularily if he or she had consulted this particular manual you are now reading. Even if this wasn't the case, *NUC123* is user-friendly enough to allow those who wish to "wing it" to get through a basic run-through. Nevertheless, we can imagine that poor Rocky here is totally baffled and he doesn't have this manual in hand. Is he doomed, is this the end of our hero? No! He sees HELP in the main menu, he then succeeds in entering a "1," and

```
MENU SELECTION
---- ---------

1. HELP
2. SET COMPUTATION PARAMETERS
3. SET MODEL PARAMETERS
4. SET TEMPORARY REACTION RATES
5. RUN
6. OUTPUT
7. EXIT


Enter selection (1-7).
```

Figure I.2. NUC123 main menu.

```
HELP SELECTION
---- ---------

1. INTRODUCTION
2. SETTING UP A RUN
3. RUNNING THE PROGRAM
4. OUTPUT OPTIONS
5. GENERAL METHOD OF COMPUTATION
6. USING THE ADAPTIVE SUBROUTINE
7. EXIT


Enter selection (1-7).
```

Figure I.3. HELP section submenu.

he is saved for he now has available an online equivalent of this manual (although a bit scaled-down). He still does have to figure out what he is confused about for he is presented with a HELP submenu as shown in figure I.3.

- SET COMPUTATION PARAMETERS section allows Rocky to set the values of an array of parameters used in constraining the computation. These values include constants which help determine the time step size, the initial time step size, initial and final temperatures, and minimum allowable nuclide abundances. The submenu

for this section will be described in section II.A

- SET MODEL PARAMETERS section lets Rocky play "God" and lets him choose the features of the universe at the time of nucleosynthesis. He can have his pick of the baryon-to-photon ratio (η), neutron half-life, number of neutrino types, and so on. All this will be discussed in more detail in section II.B.
- SET TEMPORARY REACTION RATES section is for the real nucleosynthesis jock who wants to know what happens to the predicted abundances every time a new reaction rate measurement is made. Here, Rocky can temporarily substitute his own reaction rate numbers for the reactions of his choice during a run. Specifics will be forthcoming in section II.C.
- RUN section lets Rocky sit and stare at the screen while the code does its wisdom thing. Actually, Rocky can initially determine the size of the reaction network to be used in the computation and specify multiple loopings of runs as parameters are shifted through a range of values before he gives the go for the code to crunch numbers. More on this in section III.
- OUTPUT section lets Rocky see the results of the computation in 2 ways: as an output file which can be printed out or as a listing right there on the screen. This is elaborated upon in section IV.
- Exit – It's Miller time for Rocky.

  If a response to a menu option request is not in the range 1-7 or if Rocky simply hits $< RETURN >$, then the response is assumed to be 7, the option for Exit. This feature holds analogously for all of the submenus as well.

Although the menu interface is nice and convenient, it may be that Rocky wants to do a lot of runs and does not wish to take up a lot of terminal time. In this case, the program can be easily converted to run in batch mode through a convertion macro such as the one listed in figure I.4 (this one is written for the VAX/VMS system). The required changes involve converting the `READ/WRITE` unit numbers to correspond to data files instead of the terminal. This macro specifically substitutes unit 1 for the `READ/WRITE` statements with unit 7 for input from file `BATIN.INC` and with unit 8 for output to file `BATOUT.INC` in the `OPEN` and `CLOSE` file statements. These substitutions are also made in the `PARAMETER`

```
$ ed nuc123.for
find 'unit=1'
sub/unit=1/unit=7
sub/sys$command/batin.inc
copy . to .+
sub/unit=7/unit=8/.-
sub/batin/batout
sub/old/new
find 'close (unit=1)'
sub/unit=1/unit=7
copy . to .+
sub/unit=7/unit=8/.-
sub/iu=1/iu=7/whole
sub/ou=1/ou=8/whole
exit
$ ren nuc123.for nucbat.for
$ write sys$output "NUCBAT ready"
```

Figure I.4. VAX/VMS command file for converting NUC123 into NUCBAT, a version for batch runs.

statement of all the subroutines which contain the appropriate READ/WRITE statements.

All Rocky needs to do to run this batch version of *NUC123* (given the name NUCBAT.FOR here) is to edit into file BATIN.INC all the commands he would have put in at a terminal session. Thus, even with a batch run, all the flexibility of an interactive session is retained.

II. Input Parameters

A. Computational Parameters

Suppose that Rocky has consulted the HELP section of the menu so he has now rolled up his sleeves to do some heavy-duty nucleosynthesis computing. Because Rocky has got himself all oriented, he knows that he can first go into menu selection 2 (taken care of by subroutine `SETCOM`) to adjust the computation mechanism of the program to his liking. The submenu shown in figure II.1 gives him complete control over the adjustments of computational parameters which include 2 parameters which set a limit on the amount the nuclide abundances can change over a time step, the size of the initial time step, the starting and ending temperatures, a lower limit on nuclide abundances, a condition on when to accumilate data, and the maximum number of accumilations. Rocky is also given the opportunity to restore, in one bold stroke, all parameters to their default values (the ones listed in the figure). Each of these input parameters will now be discussed in detail.

The first three and sixth parameters involve the adaptive stepsize control for the Runge-Kutta routine used in the main program. The basic idea of this control mechanism is to allow the stepsize to be varied so that the error in quantity being evolved is limited by predetermined accuracy (for details, consult *Numerical Recipes* section 15.2).

The particular manifestation of this control feature in Wagoner's code is described in Appendix C of Wagoner's 1969 paper (equations C6 and C7 which will be discussed here but not reproduced here; however, don't get impatient as I will write down their relevant equivalents in due time). There, the temperature and nuclide abundances are required not to change by too much for each time step. Parameter `dlt90` ($= \frac{dT_9}{T_9}$) in subroutine `DRIVER` corresponds to limit K_T in equation C6 and time step limiting constant 1 (variable `reg` in the program) corresponds essentially to limit K_Y in equation C7. Time step limiting constant 2 (variable `regm` in the program) does not appear in equation C7 as the equation actually used in the program is a modification of C7. In the program, a time step is computed for each abundance change through an equation similar to C7:

$$\Delta t_{lim} = \left| \frac{Y_i(t)}{\frac{dY_i(t)}{dt}} \right| \mathtt{reg} \left(1 + \left[\left(\frac{log(Y_i)}{log(Y_m)} \right) \left(\frac{\mathtt{regm}}{\mathtt{reg}} - 1 \right) \right]^2 \right) \tag{2.1}$$

```
          SET COMPUTATION PARAMETERS SELECTION
          --- ---------- ---------- ---------

 1. CHANGE TIME STEP LIMITING CONSTANT 1  FROM 0.300
 2. CHANGE TIME STEP LIMITING CONSTANT 2  FROM 0.600
 3. CHANGE INITIAL TIME STEP              FROM 1.00E-06
 4. CHANGE INITIAL TEMPERATURE (10**9 K)  FROM 1.00E+02
 5. CHANGE FINAL TEMPERATURE (10**9 K)    FROM 1.00E-02
 6. CHANGE SMALLEST ABUNDANCES ALLOWED    FROM 1.00E-25
 7. CHANGE CONDITION FOR ACCUMILATION     FROM 3.00E+01 ITERATIONS
 8. CHANGE MAXIMUM LINES TO BE PRINTED    FROM 40
 9. RESET ALL TO DEFAULT VALUES
10. EXIT

Enter selection (1-10).
```

Figure II.1. SET COMPUTATIONAL PARAMETERS section submenu.

in which the abundances are given by $Y_i = X_i/A_i$, where X_i is the mass fraction contained in nucleus i having atomic weight A_i and the parameter which puts a lower limit on the abundances during the calculations (sixth item in the submenu) is given by Y_m. Wagoner mentions in the same appendix that "it was found helpful to not allow the abundance of any nucleus to be less than Y_m, chosen as 10^{-25} (the default value here)." The smallest value of Δt_{lim} found amongst all of the abundance changes is retained and is compared to Δt_{min} which is found from the temperature change using an equation equivalent to C6:

$$\Delta t_{min} = \left| \frac{1}{\bar{T}_9} d\bar{T}_9 \frac{1}{\frac{1}{T_9}\frac{dT_9}{dt}} \right| \tag{2.2}$$

in which the temperature is given by T_9 in units of 10^9 K. The smaller of these two is retained as the new time step unless it is more than 50% larger than the old time step in which case it is limited to being 150% of the old time step.

The initial time step (the third item in the submenu) is the first time step to start the computation off and the default value is set to 10^{-6}. The default value of **reg** is 0.3 and of **regm** is 0.6. All three of these default values were retained from the Wagoner code version that I had obtained from Edward Kolb. The default values of all of the computational and model parameters and the values of reaction rate coefficients can be found in the `BLOCK DATA` at the very end of the program listing.

The fourth and fifth parameters, the initial and final temperatures, are in units of 10^9 K and have default values of 10^{11} K and 10^7 K respectively. The program takes the initial temperature, computes an initial time and then proceeds to do the time evolution from thereon. It continues in this manner until the temperature falls below the final temperature or accumilates the maximum number of lines of data allowed by the eighth parameter in the submenu which has a default value of 40. The condition for accumilation (the seventh parameter) specifies at which time step the recording of information (about the temperature, nuclides abundances, etc.) is to take place. The default setting is to record every 30 loopings of the Runge-Kutta routine. It is also possible to record after a given linear (in units of 10^9 K) or logrithmic (in terms of $\log(10^9$ K)) drop in temperature.

B. Model Parameters

With the computation mechanism tuned just the way he wants it, Rocky can put together his very own recipe of cooking up light elements during primordial nucleosynthesis by fiddling around with the model parameter settings (note: sometimes the type of universe selected affects the kind of settings one can have for the computation parameters). Selection 3 (contained in subroutine `SETMOD`) presents the submenu show in Figure II.2. Rocky can investigate regions of the standard model by ranging through values of the second, third, and fourth parameters. The first and last four parameters let Rocky do some roaming outside of the standard scenario.

The first parameter is a multiplication factor which is applied to the accepted value of the graviational constant to produce a variant value. The default value for this parameter is set to 1. The second parameter is the neutron half-life in minutes and the default is 10.5 minutes. The neutron half-life is converted into the lifetime in seconds in subroutine `START` for use in rate calculations. The third parameter is the number of neutrino species and the default value here is 3. The fourth parameter, the baryon-to-photon ratio η, is set to a value of 2.751×10^{-10} so as to produce a present day value of 10^{-10}, going down by a factor of 11/4 due to e^+-e^- annihilation just before nucleosynthesis (see Weinberg p. 536 or Peebles p. 250).

```
                SET MODEL PARAMETERS SELECTION
                --- ----- ---------- ---------

 1. CHANGE VARIATION OF GRAV CONSTANT      FROM  1.000E+00
 2. CHANGE VARIATION OF NEUTRON HALF-LIFE  FROM  1.050E+01
 3. CHANGE NUMBER OF NEUTRINO SPECIES      FROM  3.000E+00
 4. CHANGE BARYON TO PHOTON RATIO          FROM  2.751E-10
 5. CHANGE COSMOLOGICAL CONSTANT           FROM  0.000E+00
 6. CHANGE PSI ELECTRON                    FROM  0.000E+00
 7. CHANGE PSI MUON                        FROM  0.000E+00
 8. CHANGE PSI TAUON                       FROM  0.000E+00
 9. RESET ALL TO DEFAULT VALUES
10. EXIT


Enter selection (1-10).
```

Figure II.2. SET MODEL PARAMETERS section submenu.

The fifth parameter is the value of the cosmological constant (default value is 0) in the usual units as used in the Friedmann equation

$$H^2 = \frac{8\pi}{3} G \left(\rho + \frac{\lambda}{3} \right) \tag{2.3}$$

to compute the expansion rate in subroutine DERIVS.

The last three parameters deal with neutrino degeneracy and in the the default case, the neutrinos are treated as being nondegenerate. The neutrino degeneracy parameters are defined as $\xi = \mu/T$ with μ the chemical potential in units of MeV and T the temperature in MeV and these parameters are available for up to 3 neutrino species. Details concerning the cosmology of neutrino degeneray can be found in the literature (Wagoner, Fowler, and Hoyle 1967, Beaudet and Goret 1976, David and Reeves 1980, Scherrer 1983, and Boesgaard and Steigman 1985). Basically speaking, the initial neutron-to-proton ratio is altered (in subroutine START) if ξ_e is nonzero, the neutrino energy densities (see Scherrer 1983) are generalized to

$$\rho_{\nu,\bar{\nu}} = \frac{1}{2\pi^2} T_{\nu,\bar{\nu}}^4 \int_0^\infty dx \frac{x^3}{1 + exp(x \mp \xi_\nu)} \tag{2.4}$$

which is numerically integrated in subroutine NUDENS if it is beyond the valid range of approximations, and the n $\leftrightarrow$ p reaction rates for ξ_e nonzero (see Beaudet and Goret 1976) generalize to

$$\lambda_{n\to p} = K\int_1^\infty dx \frac{(x+q)^2(x^2-1)^{1/2}}{(1+e^{-xz})[1+e^{(x+1)z_\nu+\xi_e}]} \tag{2.5}$$

$$+ K\int_1^\infty dx \frac{(x-q)^2(x^2-1)^{1/2}}{(1+e^{+xz})[1+e^{-(x-1)z_\nu+\xi_e}]}$$

for n $\rightarrow$ p with $q=(m_n-m_p), z=m_e/T_\gamma, z_\mu=m_e/T_\nu$, and K normalized using the neutron half-life and similarily for p $\rightarrow$ n. This rate is then numerically integrated (as opposed to the nongenerate case in which the rate is analytically evaluated using an approximation which is given in Wagoner 1969, Table 14B) in subroutine RATE1.

C. New Reaction Rates

Perhaps Rocky has gotten wind of a recent remeasurement of an important reaction rate used in the nucleosynthesis computation. What are the consequences? – another quick paper for Rocky of course. Seriously, its consequences for primordial nucleosynthesis predictions are easy to find out using selection 3 (subroutine SETRAT) from the main menu. With the submenu shown in figure II.3, Rocky can opt to change extant reaction rates, add new reactions involving nuclides already in the code, or restore all reaction rates to their default values. The permanent reaction rates are computed in subroutines RATE0 (weak decay rates), RATE1 (n, p rates), RATE2 (up to Be^7), RATE3 (up to N^{12}), and RATE4 (up to O^{16}). The new rates are computed in routine RATE5.

If Rocky wants to change a reaction rate, he enters an 'a' whereupon he sees on the screen a roster of changed reaction rates (which is empty the first time around). He then can add a changed reaction rate to the roster or he can either change the value of the reaction rate coefficients of, list the reaction coefficients for, or delete the change from (restore to default) a reaction already on the roster. If Rocky wants to add a reaction rate change, he is required to specify a reaction number which can be obtained from the program listing of subroutines RATE1-4 from the index of the reaction rate variable f. Once this is given, Rocky is then asked to fill in a very general formula for the reaction

```
SET NEW REACTION RATES SELECTION
--- ------------ ----- ---------

 1. CHANGE REACTION RATES
 2. ADD NEW REACTIONS
 3. RESET ALL TO DEFAULT VALUES
 4. EXIT



Enter selection (1-4).
```

Figure II.3. SET TEMPORARY REACTION RATES section submenu.

rate

$$f(T_9) = w_1 T_9^{w_2} exp(w_3/T_9^{w_4} - (w_5 T_9)^2). \qquad (2.6)$$
$$\times (1 + w_6 T_9^{w_7} + w_8 T_9^{2/3} + w_9 T_9 + w_{10} T_9^{4/3} + w_{11} T_9^{5/3})$$
$$+ w_{12} T_9^{w_{13}} exp(w_{14}/T_9)$$
$$+ w_{15} T_9^{w_{16}} exp(w_{17}/T_9)$$
$$+ w_{18} T_9^{w_{19}} exp(w_{20}/T_9)$$

which is in units of sec^{-1} (see Wagoner 1969, pp. 253-254 and Fowler, Caughlan, and Zimmerman 1967, p. 528 equation 6 and 7 for a general discussion of the reaction rates). This formula is general enough to suffice for most reactions though not for all. Rocky thus has to enter values for the w's to duplicate the reaction rate formula which he wishes to enter. The maximum number of reaction rates that can be altered in this fashion is 15. If Rocky makes a mistake here or if for some other reason he wants to change rates he's already put in, all he has to do is to type in 'c' and enter the roster number of the reaction of interest and indicate the nth coefficient with which to start inputting the change. The listing option is specified by 'l' and the delete option is enacted by a 'd.'

Rocky might also wish to add a completely new reaction (say the reaction $d(d,\gamma)He^4$ which is not part of the regular reaction network) in which case the screen shows a roster

of new reactions. The process here is analogous to the description given above for changing existing rates except that instead of a reaction number, Rocky is asked to put in a reaction label (i.e. $d(d,\gamma)He^4$) and instead of the reaction rate coefficients, Rocky puts in the reaction type (to figure this out, go to the 200 section of subroutine `SOL`), the code numbers of all 4 participating nuclides (see the `DATA nuclds` statement in `BLOCK DATA`), the reverse reaction coefficient (discussed in Fowler, Caughlan, and Zimmerman 1967, pp. 528-532 and in the form listed in FCZ 1975 and Harris et al. 1983), and heat of reaction (in units of $10^9 K$). To input the reaction rate coefficients themselves, Rocky then has to go into the change section (selection 1 on the New Reaction Rates submenu) and proceed as previously described. A maximum of 12 new reactions may be added in this way (which is why many of the variables in the program are dimensioned to 100 instead of 88).

Selection 3 in the submenu will erase both rosters, restoring the changed rates and erasing any new ones. Rates for new reactions are also erased any time the reaction network size is changed (see section III.A). This is because the new reactions are added to the end of the current reaction network size. For instance, in the smallest network, there are 34 reactions. If 3 new reactions are added their rates will go into slots `f(35)`, `f(36)`, and `f(37)` which will be needed for the original reactions if the network is enlarged.

III. Running The Code

A. Menu Options

Once our pal Rocky is through setting computation parameters, model parameters, and reaction rates, he is all set to let the code get down to its business of evolving the universe within the predetermined temperature range and with the given conditions. The running of the code can be accessed through selection 5 of the main menu (subroutine `RUN`) which offers the submenu depicted in Figure III.1. In it, we see that there are two ways to run the program and two further options which affect the running.

The "Go" option will set the program off on solving the differential equations for evolving the temperature, nuclide abundances, etc. With the default settings, this takes about 12 CPU seconds on the VAX 11/780. When the computation is complete, a message to that effect is displayed on the screen so that Rocky can now go into the Output section to look at the final results (see section IV for discussion of the Output section).

Another option is to do many runs while allowing some of the model parameters to step through a specified range. If Rocky chooses the "Do Multiple Runs" option, he can specify up to three model parameters which are to be incremented through given step sizes through predetermined ranges (the model parameter settings are thus altered and ultimately take the values for the final run). Rocky first must choose how many parameters are to be varied (1-3) with the default being 1 and then must specify which parameter is to be varied along with the starting value, the final value, and the incremental value. If parameter is not specified, the number of parameters varied automatically goes down by one. The first parameter entered is varied in the outermost `DO` loop whereas the last parameter entered is varied in the innermost loop. The starting and final values may be the same (to produce no variation for this particular parameter) but the incremental value cannot be entered as zero (as the equation to determine the number of loopings will blow up) so some other arbitrary value must be submitted. Once the value specifications are made for each parameter, Rocky is asked to confirm them (in case of a mistaken entry) and once this is done for all of the parameters, the program goes off to do the multiple

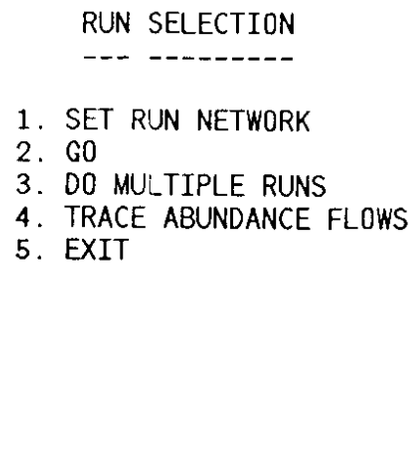


Figure III.1. RUN section submenu.

computations and when everything is done, it comes back with an appropriate message. For a large number of loopings, running in batch mode is recommended (for batch runs see section I.C).

These multiple runs can take up a long time especially if the computer is busy. Thus Rocky has the option of either dozing off in front of the terminal or utilizing the "Set Run Network" option in which he can cut the computation time by reducing the size of the reaction network. The default network consists of 26 nuclides and 88 reactions but through the use of this option, this may be cut to 18 nuclides and 64 reactions or even down to 9 nuclides and 34 reactions. For the first reduction, the time needed is about 60% of the full network and the variation (using the default parameters) in final abundances from that of the full network by about .1%. For the more drastic reduction, time is cut down to about 20% with a variation of something like .5%. The sizes of the three networks are depicted in Figure III.2. As mentioned before, if the reaction network size is changed, then any new reactions entered will be erased.

Finally, the option "Trace Abundance Flows" allows Rocky to see changes in the abundances during the accumilation times by creating a file `NUC123.FLO`. All reactions considered in the program contain at most four different nuclides and thus are of the form

$$N_i(^{A_i}Z_i) + N_j(^{A_j}Z_j) \leftrightarrow N_k(^{A_k}Z_k) + N_l(^{A_l}Z_l) \tag{3.1}$$

where N_m is the number of nuclide m and $A_i \geq A_j$ and $A_l \geq A_k$. The abundance changes

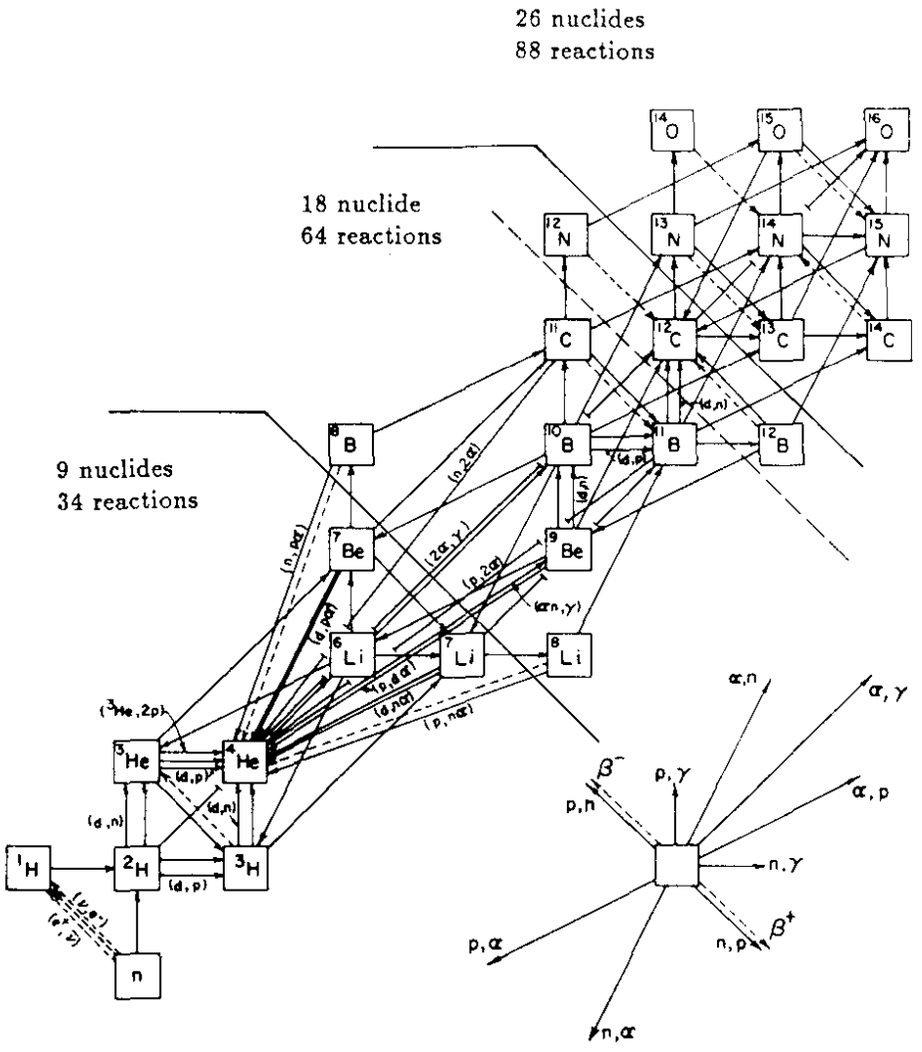


Figure III.2. Display of the nuclides involved in the abundances computation and the network of reactions between those nuclides. The standard computation involves everything shown whereas smaller computations use only the part of the network below the lines drawn. (Figure adapted from Schramm and Wagoner 1977).

are given in terms of dY_i/dt which is computed from equation 28 of Wagoner 1969

$$\frac{dY_i}{dt} = \sum_{j,k,l} N_i \left(- \frac{Y_i^{N_i} Y_j^{N_j}}{N_i! N_j!} [ij]_k + \frac{Y_l^{N_l} Y_k^{N_k}}{N_l! N_k!} [lk]_j \right) \tag{3.2}$$

in which $[ij]_k$ is the forward reaction rate. In this equation, the abundance changes are summed over all reactions involving nuclide i but in the trace request, the abundance changes are given for a particular reaction. These abundance flow traces can be requested in three ways: by specifying particular nuclides for which the abundances changes for all

reactions involving the nuclides will be given; by specifying particular reactions; and by asking for the abundance changes for all reactions. The nuclides are specified by numbers which can be deduced from the `DATA nuclds` statement in `BLOCK DATA` and the reaction numbers by the indices of the variable `f`, the forward reaction rate value which is found in subroutines `RATE0-4`. The particular times for which these changes are recorded are determined by the accumiliation criterion discussed in section II.A.

B. Calculational Algorithms

Let us suppose now that Rocky has chosen his run scheme in some fashion and has set the code off and running. While Rocky is staring blankly at the screen waiting for the program to come back with a message that it is done, let us go right into the computer program to see just what is happening and to follow along, for a bit, the algorithms by which it gets its results.

The basic purpose of the computational part of the code (which is essentially the same as that of the Wagoner code) is to time evolve three quantities, the temperature, the electron chemical potential ϕ_e, and a quantity proportional to the baryon density defined by $h \equiv M_u N_b / T_9^3$ where M_b is the unit of atomic mass (see equation 4 in Wagoner 1969) along with the nuclide abundances $Y_i = X_i/A_i$ with X_i the mass fraction in nuclide i of atomic weight A_i. This time evolution is driven by a second-order Runge-Kutta routine that solves the differential equations which give the time derivatives of these quantities. The differential equations for the abundance changes are arranged into a single matrix equation which is solved by Gaussian elimination and the solutions are used to help find the time derivatives of the other three quantities being evolved.

The first thing that happens when Rocky gives the green light is for the driving subroutine `DRIVER` to call subroutine `START` to compute some initial values. The initial temperature comes from a specification in the computational parameter section, the initial value of h comes from the baryon-to-photon ratio (see Schramm and Wagoner 1977 equation 3.11)

$$h = 3.37 \times 10^4 \eta, \tag{3.1}$$

and the electron chemical potential is calculated from h and the temperature. The initial abundances of neutrons and protons given by

$$Y_n = \frac{1}{1 + e^{q/kT}} \tag{3.2}$$

$$Y_p = \frac{1}{1 + e^{-q/kT}} \tag{3.3}$$

with $q = m_n - m_p$. Among other quantities which are computed at this time are the initial baryon energy density,

$$\rho_b = hT_9^3 \quad , \tag{3.4}$$

the time (see Wagoner, Fowler, and Hoyle 1967, equation A15),

$$t = (12\pi Gac^{-2})^{1/2} T_9^{-2} = (10.4)^2 T_9^{-2} \quad , \tag{3.5}$$

and the weak decay rates (in subroutine RATE0).

The Runge-Kutta driving routine is contained in the subroutine DRIVER. Here, the three quantities of the temperature, h, and ϕ_e and the abundances are time evolved under the label of the variable v. Details concerning the Runge-Kutta method can be found in *Numerical Recipes* section 15.1 but the basic idea of the second-order Runge-Kutta routine used here is that given data $Y(x_1)$ for a particular value of x_1, one may wish to find the quantity $Y(x_2)$ for a value of x_2. To do this, one first finds the derivative at x_1, $dY/dx(x_1)$, and linearly extrapolates, using interval $\Delta x = x_2 - x_1$, to find $\tilde{Y}(x_2)$, an initial trial value (all this is done in the first Runge-Kutta loop in the program). This alone is not very accurate nor stable so one goes further and computes the derivative at x_2, $d\tilde{Y}/dx(x_2)$ and then averages the two values of the derivatives to get

$$\frac{dY}{dx}(x_1) = \frac{1}{2}\left[\frac{dY}{dx}(x_1) + \frac{d\tilde{Y}}{dx}(x_2)\right] \tag{3.6}$$

from which one can extrapolate to get to the new value $Y(x_2)$:

$$Y(x_2) = \frac{dY}{dx}(x_1)\Delta x \tag{3.7}$$

(this part is done in the second Runge-Kutta loop). A graphic representation of the procedure is shown in figure III.3. In the program, the value of a quantity at point 1 in

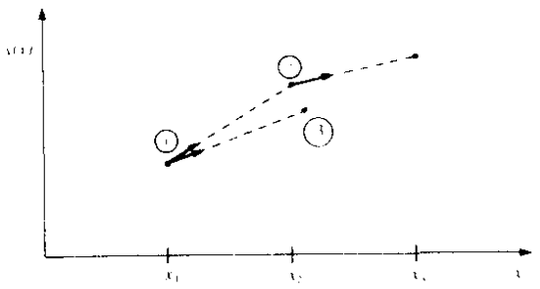


Figure III.3. Illustration of the second-order Runge Kutta procedure. Point 1 indicates the value of y at x_1 and the arrow there indicates the value of the derivative of y at x_1. This value of the derivative is used to computed a trial value of y at x_2 for which the derivative is also calculated. The actual value of y at x_2 to be used, point 3, is derived by using the average of the derivatives computed at points 1 and 2. (Figure adapted from *Numerical Recipes*).

the figure is given by v0 and v initially refers to the value at point 2 but later to that at point 3.

In the first Runge-Kutta loop, if the accumiliation criterion is met, DRIVER calls subroutine OUTPUT to accumilate the current values of the temperature, h, ϕ_e, abundances, etc. If the termination criterion is satisfied, then the current information is accumilated before the computation is terminated and if it is not, the program goes on to adjust the time step before computing the the trial value $\tilde{Y}(x_2)$.

This is the basic picture of the operation of the code. More specific details are involved in the computation of the derivatives which is taken care of in subroutine DERIVS. The abundance derivatives are computed; then, these are included in the computation of the derivatives for the temperature, h, and ϕ_e. To compute the abundance derivatives, the energy densities of baryons, photons, neutrinos, etc. are first computed in DERIVS and in THERM, utilizing functional values computed in BESSEL using modified Bessel function solutions derived in KNUX. Next, DERIVS calls routines RATE1 through RATE5 to compute the forward reaction rates. RATE1 computes the n $\rightarrow$ p and p $\rightarrow$ n rates, RATE2 through RATE4 the regular reaction rates depending on the reaction network size, and RATE5 the new reaction rates if new reactions were entered. The functional form of the reaction rates are discussed in Fowler, Caughlan, and Zimmerman 1967 and in Wagoner 1969 and

rates are given for reactions in these two papers as well as in the more recent papers of Fowler, Caughlan, and Zimmerman 1975, and Harris et al. 1983. Some basic discussion of reactions rates are also given in Clayton chapter 4. Clayton makes clear in his book the point that the distribution of the velocities of a nuclide species is taken into account not simply by taking an average velocity for a given temperature but actually dealing with an integral over the velocity distribution. References are given in the program for reaction rates whose origin in the literature I have been able to trace. A discussion of the rates for reactions number 24 and 26 can be found in Kawano et al. 1988.

Once the forward reaction rates are computed, `DERIVS` then calls subroutine `SOL` which builds the matrix equation for the abundance changes. Wagoner uses the method of implicit differencing because the right-hand side of equation (3.2) is a small difference of large numbers due to the near equality of the forward and reverse rates at high temperatures and this requires prohibitively small step sizes to maintain stability (see Wagoner 1969, appendix C). The method is discussed in section 15.6 of *Numerical Recipes* and I will present some relevant material here. Consider the equation

$$Y' = -CY \tag{3.8}$$

with constant C > 0. In the explicit differencing scheme, this is rewritten as

$$Y_{n+1} = Y_n + \Delta t Y'_n = (1 - C\Delta t)Y_n \tag{3.9}$$

which is unstable for $\Delta t > 2/C$ since in this case $Y_n \rightarrow \infty$ as $n \rightarrow \infty$. The suggested cure is implicit differencing which is to use instead

$$Y_{n+1} = Y_n + \Delta t Y'_{n+1} = \frac{Y_n}{(1 + C\Delta t)} \tag{3.10}$$

which is absolutely stable. This idea can be generalized to a system of linear equations which result in the matrix equation

$$\mathbf{Y_{n+1}} = (\mathbf{1} + \mathbf{C\Delta t})^{-1} \cdot \mathbf{Y_n} \tag{3.11}$$

Thus, in the Wagoner code, the abundance change equation (3.2) is replaced by the equation

$$\frac{dY_i}{dt}(t + \Delta t) = \frac{\bar{Y}_i(t + \Delta t) - Y_i(t)}{\Delta t} \tag{3.12}$$

$$
\begin{aligned}
&= \sum_{i,j,k} - \frac{N_i[\tilde{ij}]_k}{N_i!N_j!(N_i+N_j)} \\
&\times [N_i\tilde{Y}_i^{N_i-1}\tilde{Y}_j^{N_j}\bar{Y}_i(t+\Delta t) + N_j\tilde{Y}_i^{N_i}\tilde{Y}_j^{N_j-1}\bar{Y}_j(t+\Delta t)] \\
&+ - \frac{N_i[\tilde{lk}]_j}{N_k!N_l!(N_k+N_l)} \\
&\times [N_k\tilde{Y}_k^{N_k-1}\tilde{Y}_l^{N_l}\bar{Y}_k(t+\Delta t) + N_l\tilde{Y}_k^{N_k}\tilde{Y}_l^{N_l-1}\bar{Y}_l(t+\Delta t)]
\end{aligned}
$$

in which the barred abundances are computed solely to find the abundance change from the first equality and the quantities with the tildes are those of a previous time step point. The equation for the second equality can be written in the form of equation (3.11)

$$(A_{ij})[\bar{Y}_j(t+\Delta t)] = [Y_i(t)] \tag{3.13}$$

which is then solved by Gaussian elimination and back substition in subroutine `EQSLIN`. This solution method is thoroughly covered in section 2.2 of *Numerical Recipes* but I will put in a few words here. The matrix equation (3.13) can be triangularized to look like

$$
\begin{pmatrix}
A'_{11} & A'_{12} & A'_{13} & \dots & A'_{1n} \\
0 & A'_{22} & A'_{23} & \dots & A'_{2n} \\
0 & 0 & A'_{33} & \dots & A'_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & A'_{nn}
\end{pmatrix}
\cdot
\begin{pmatrix}
\bar{Y}_1 \\ \bar{Y}_2 \\ \bar{Y}_3 \\ \vdots \\ \bar{Y}_m
\end{pmatrix}
=
\begin{pmatrix}
Y'_1 \\ Y'_2 \\ Y'_3 \\ \vdots \\ Y'_m
\end{pmatrix}
\tag{3.14}
$$

in which the primes indicate that the values of the A's and Y's have been modified by the row operations used in the Gaussian elimination procedure. The solution for the $\bar{Y}$'s are found using back substitution as follows. For Y_n, the final line of the matrix equation gives

$$\bar{Y}_n = Y'_n/A'_{nn} \tag{3.15}$$

and this can be used to solve for $\bar{Y}_{n-1}$ and so on with the general solution being

$$\bar{Y}_i = \frac{1}{A'_{ii}}\left[Y'_i - \sum_{j=i+1}^{N} A'_{ij}\bar{Y}_j\right] \quad . \tag{3.15}$$

In the program, entries for the matrix equation (3.13) are inserted in reverse order so that the equations for nuclides with substantial abundances are solved first.

At accumilation times, the code allows for iterative improvement of these solutions. Here I am again indebted to *Numerical Recipes* (where would we be without that book!) where this topic is covered in section 2.7. Although we are solving for the equation

$$\mathbf{A} \cdot \bar{\mathbf{Y}} = \mathbf{Y} \quad , \tag{3.16}$$

the solution we do find inevitably has an error $\delta\bar{\mathbf{Y}}$ associated with it:

$$\mathbf{A} \cdot \bar{\mathbf{Y}} + \delta\bar{\mathbf{Y}} = \mathbf{Y} + \delta\mathbf{Y} \quad . \tag{3.17}$$

Subtracting equation (3.16) from (3.17) gives

$$\mathbf{A} \cdot \delta\bar{\mathbf{Y}} = \delta\mathbf{Y} \tag{3.18}$$

into which the solution for $\delta\mathbf{Y}$ from equation (3.17) can be inserted to give

$$\mathbf{A} \cdot \delta\bar{\mathbf{Y}} = \mathbf{a} \cdot (\bar{\mathbf{Y}} + \delta\bar{\mathbf{Y}}) - \mathbf{Y} \tag{3.19}$$

for which the right-hand side is know and can thus be solved for $\delta\bar{\mathbf{Y}}$.

Once the abundances are solved for, subroutine `DERIVS` then goes on to compute the derivatives for the temperature, h, and ϕ_e from

$$\frac{dT_9}{dt} = \frac{-3R^{-1}dR/dt}{\rho_b^{-1}d\rho_b/dT_9} \tag{3.20}$$

$$\frac{dh}{dt} = 3h\left[\frac{1}{R}\frac{dR}{dt} - \frac{1}{T_9}\frac{dT_9}{dt}\right] \tag{3.21}$$

$$\frac{d\phi_e}{dt} = \frac{\partial\phi_e}{\partial T_9}\frac{dT_9}{dt} + \frac{\partial\phi_e}{\partial h}\frac{dh}{dt} + \frac{\partial\phi_e}{\partial S}\frac{dS}{dt} \tag{3.22}$$

where

$$S = \sum_i Z_i Y_i \quad ,$$

and the partial derivatives are computed from

$$n_-(T_9, \phi_e) - n_+(T_9, \phi_e) = N_a h T_9^3 S \tag{3.23}$$

where N_a is Avogadro's number (see Wagoner 1969, appendix C).

IV. Output Options

A. Menu Options

So now we're done with our journey into the intricacies of the code and we can rejoin Rocky to see what has come out of all this. With the computations all complete, Rocky can go to option 6 in the main menu (handled by subroutine `OUTPUT`) to display the results in a number of ways. This leads to the submenu shown in figure IV.1 which offers Rocky the choice of creating an output file (named `NUC123.DAT`) which can be printed out or of seeing the results right there on the screen. A sample output file is shown in figure IV.3 which was produced using the default parameters. The parameters set for the run are shown listed above the nuclide abundances which are given in terms of the mass fraction for ^{4}He and relative to the H abundance for the rest of the nuclides. The abundances given for each accumilation point along with the temperature at that point with the temperature given in units of 10^9 K but which can be listed in terms of MeVs by the use of option 1 in the submenu. In the second row, the temperatures are again listed along with the time (in seconds), the energy density (in units of g·cm^{-3}) of photons, electrons, electron neutrinos, and baryons, the electron chemical potential, time step size, baryon-to-photon ratio, and the expansion rate. Through option 1 again, the energy densities can be listed as fractions of the total energy density.

If Rocky is impatient to see the results, he can choose to flash them right on the screen with option 3. The tables which appear on the screen are arranged in the same manner as with the output file but have been split up into four parts in order to accommodate them on the screen. Thus, Rocky can choose which of these four parts to view by using the submenu shown in figure IV.2. The first part contains the abundances for the nuclides usually referred to in primordial nucleosynthesis studies as well as tritium. The second part contains the abundances for other light nuclides, the third part the energy densities, and the fourth part the rest of the quantities. A sample of the first part is displayed in figure IV.4.

```
OUTPUT SELECTION
------ ---------

1. CHANGE UNITS
2. REQUEST OUTPUT FILE
3. REQUEST OUTPUT ON SCREEN
4. EXIT



Enter selection (1-4).
```

Figure IV.1. OUTPUT section submenu.

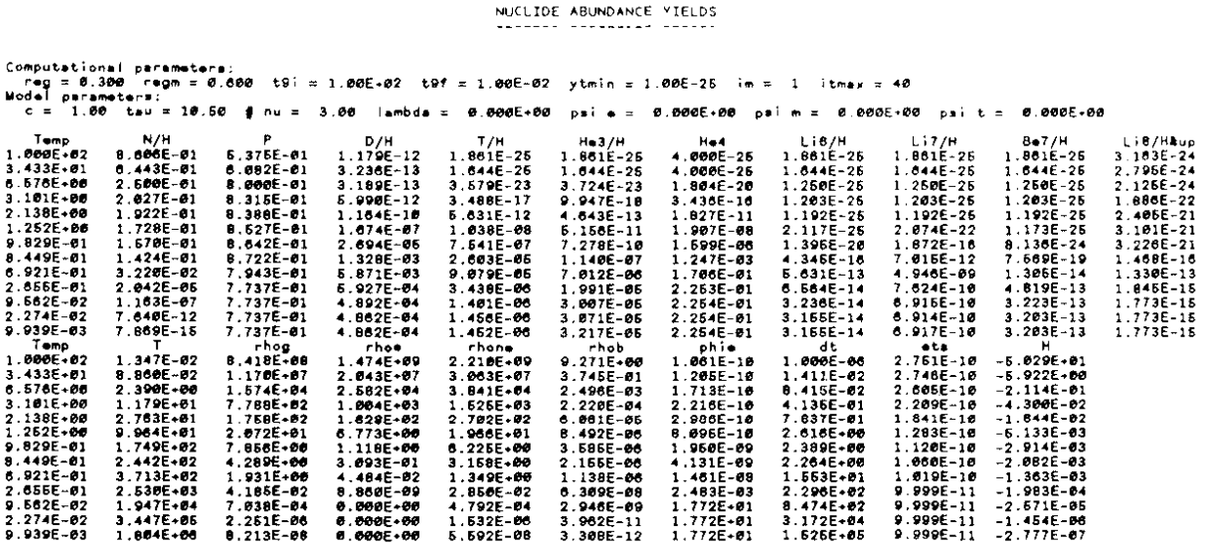
```
SCREEN OUTPUT SELECTION
------ ------ ---------

1. DISPLAY D,T,HE3,HE4,LI7
2. DISPLAY N,P,LI6,BE7,LI8&UP
3. DISPLAY RHOG,RHOE,RHONE,RHOB
4. DISPLAY T,DT,PHIE,ETA,H
5. EXIT



Enter selection (1-5).
```

Figure IV.2. Screen output sub-submenu.



NUCLIDE ABUNDANCE YIELDS

Computational parameters:
reg = 0.300 regm = 0.600 t9i = 1.00E+02 t9f = 1.00E-02 ytmin = 1.00E-25 im = 1 itmax = 40
Model parameters:
c = 1.00 tau = 10.50 # nu = 3.00 lambda = 0.000E+00 psi e = 0.000E+00 psi m = 0.000E+00 psi t = 0.000E+00

Temp	N/H	P	D/H	T/H	He3/H	He4	Li6/H	Li7/H	Be7/H	Li6/H&up
1.000E+02	8.806E-01	5.376E-01	1.179E-12	1.861E-25	1.861E-25	4.000E-25	1.861E-25	1.861E-25	1.861E-25	3.183E-24
3.433E+01	8.443E-01	6.082E-01	3.236E-13	1.644E-25	1.644E-25	4.000E-25	1.644E-25	1.644E-25	1.644E-25	2.795E-24
6.576E+00	2.600E-01	8.000E-01	3.189E-13	3.579E-23	3.724E-23	1.804E-20	1.250E-25	1.250E-25	1.250E-25	2.125E-24
3.101E+00	2.027E-01	8.315E-01	5.990E-12	3.488E-17	9.947E-18	3.436E-16	1.203E-25	1.203E-25	1.203E-25	1.886E-22
2.138E+00	1.922E-01	8.388E-01	1.164E-10	5.631E-12	4.643E-13	1.827E-11	1.192E-25	1.192E-25	1.192E-25	2.406E-21
1.252E+00	1.728E-01	8.527E-01	1.674E-07	1.038E-08	5.156E-11	1.907E-08	2.117E-25	2.074E-22	1.173E-25	3.101E-21
9.829E-01	1.570E-01	8.642E-01	2.694E-05	7.541E-07	7.278E-10	1.599E-06	1.395E-20	1.872E-16	8.136E-24	3.226E-21
8.449E-01	1.424E-01	8.722E-01	1.328E-03	2.603E-05	1.140E-07	1.247E-03	4.345E-16	7.015E-12	7.669E-19	1.468E-16
6.921E-01	3.228E-02	7.943E-01	5.871E-03	9.079E-05	7.012E-06	1.706E-01	5.631E-13	4.946E-09	1.306E-14	1.330E-13
2.655E-01	2.042E-05	7.737E-01	5.927E-04	3.438E-06	1.991E-05	2.253E-01	6.564E-14	7.624E-10	4.819E-13	1.845E-15
9.562E-02	1.163E-07	7.737E-01	4.892E-04	1.401E-06	3.007E-05	2.254E-01	3.238E-14	6.915E-10	3.223E-13	1.773E-15
2.274E-02	7.640E-12	7.737E-01	4.862E-04	1.456E-06	3.071E-05	2.254E-01	3.165E-14	6.914E-10	3.203E-13	1.773E-15
9.939E-03	7.869E-15	7.737E-01	4.862E-04	1.452E-06	3.217E-05	2.254E-01	3.165E-14	6.917E-10	3.203E-13	1.773E-15

Temp	T	rhog	rhoe	rhone	rhob	phie	dt	eta	H
1.000E+02	1.347E-02	8.418E+08	1.474E+09	2.210E+09	9.271E+00	1.061E-10	1.000E-06	2.761E-10	-5.029E+01
3.433E+01	8.860E-02	1.170E+07	2.043E+07	3.063E+07	3.745E-01	1.205E-10	1.411E-02	2.746E-10	-5.922E+00
6.576E+00	2.390E+00	1.574E+04	2.582E+04	3.841E+04	2.496E-03	1.713E-10	8.415E-02	2.605E-10	-2.114E-01
3.101E+00	1.179E+01	7.788E+02	1.004E+03	1.526E+03	2.220E-04	2.216E-10	4.136E-01	2.209E-10	-4.300E-02
2.138E+00	2.763E+01	1.758E+02	1.629E+02	2.702E+02	6.061E-05	2.986E-10	7.637E-01	1.841E-10	-1.644E-02
1.252E+00	9.964E+01	2.072E+01	6.773E+00	1.966E+01	8.492E-06	8.096E-10	2.616E+00	1.283E-10	-6.133E-03
9.829E-01	1.749E+02	7.856E+00	1.118E+00	6.225E+00	3.586E-06	1.950E-09	2.389E+00	1.120E-10	-2.914E-03
8.449E-01	2.442E+02	4.289E+00	3.093E-01	3.158E+00	2.155E-06	4.131E-09	2.264E+00	1.060E-10	-2.082E-03
6.921E-01	3.713E+02	1.931E+00	4.484E-02	1.349E+00	1.138E-06	1.461E-08	1.553E+01	1.019E-10	-1.363E-03
2.655E-01	2.530E+03	4.185E-02	8.860E-09	2.850E-02	6.309E-08	2.483E-03	2.296E+02	9.999E-11	-1.983E-04
9.562E-02	1.947E+04	7.038E-04	0.000E+00	4.792E-04	2.946E-09	1.772E+01	8.474E+02	9.999E-11	-2.671E-05
2.274E-02	3.447E+05	2.251E-06	0.000E+00	1.532E-06	3.962E-11	1.772E+01	3.172E+04	9.999E-11	-1.454E-06
9.939E-03	1.884E+06	8.213E-08	0.000E+00	5.592E-08	3.308E-12	1.772E+01	1.626E+05	9.999E-11	-2.777E-07

Figure IV.3. Sample output file.

Computational parameters:
reg = 0.300 regm = 0.600 t9i = 1.00E+02 t9f = 1.00E-02
ytmin = 1.00E-25 im = 1 itmax = 40
Model parameters:
c = 1.00 tau = 10.50 # nu = 3.00 lambda = 0.000E+00
psi e = 0.000E+00 psi m = 0.000E+00 psi t = 0.000E+00

Temp	D/H	T/H	He3/H	He4	Li7/H
1.000E+02	1.179E-12	1.861E-25	1.861E-25	4.000E-25	1.861E-25
3.433E+01	3.236E-13	1.644E-25	1.644E-25	4.000E-25	1.644E-25
6.576E+00	3.189E-13	3.579E-23	3.724E-23	1.804E-20	1.250E-25
3.101E+00	5.990E-12	3.488E-17	9.947E-18	3.436E-16	1.203E-25
2.138E+00	1.164E-10	5.631E-12	4.643E-13	1.827E-11	1.192E-25
1.252E+00	1.674E-07	1.038E-08	5.156E-11	1.907E-08	2.074E-22
9.829E-01	2.694E-05	7.541E-07	7.278E-10	1.599E-06	1.872E-16
8.449E-01	1.328E-03	2.603E-05	1.140E-07	1.247E-03	7.015E-12
6.921E-01	5.871E-03	9.079E-05	7.012E-06	1.706E-01	4.946E-09
2.655E-01	5.927E-04	3.438E-06	1.991E-05	2.253E-01	7.624E-10
9.562E-02	4.892E-04	1.401E-06	3.007E-05	2.254E-01	6.915E-10
2.274E-02	4.862E-04	1.456E-06	3.071E-05	2.254E-01	6.914E-10
9.939E-03	4.862E-04	1.452E-06	3.217E-05	2.254E-01	6.917E-10

Figure IV.4. Sample screen output display.

B. Specific Output File

It can be that these output modes do not satisfy Rocky's particular needs; perhaps he just wants a bunch of numbers that can be used with a plotting routine. No problem – I have provided a flexible subroutine `CHECK` which can be used to produce an output file `CHECK.DAT` which can be of any fashion that Rocky pleases. At the moment, `CHECK` is called after every run and it writes out some basic information such as the final baryon-to-photon ratio and the final abundances for ^{4}He, ^{3}He, D, and ^{7}Li into `CHECK.DAT`. However, subroutine `CHECK` has in it the entire list of global variables used in *NUC123* and all the necessary `COMMON` statements to access those variables so the subroutine can easily be modified. The subroutine can be called from anywhere in the main program through the use of a `CALL CHECK` statement along with a value for variable `itime` which serves to identify the location of the `CALL` statement in the program. Several such checkpoints have been placed in the program already: at the beginning of the program (`itime` = 1) and at its end (`itime` = 10); at the beginning of the `RUN` section (`itime` = 2) and at its end (`itime` = 9); and at the beginning (`itime` = 3) and end (`itime` = 8) of every run.

V. Modifying The Program

A. Using The Adaptive Subroutine

Rocky, being the speculative kind of guy he is, occasionally will want to venture out to where no man has ever gone before. He thus may want the nucleosynthesis code to do some crazy whiz-bang stuff that's just not built into it. If these changes are not to be too drastic, they may be all accommodated in subroutine `CHECK`, the same subroutine mentioned in the previous section. Subroutine `CHECK` is essentially an empty subroutine with a complete roster of `COMMON` statements so that it has the connections to all of the global variables (any variable passed through a `COMMON` statement). It can be called from anywhere in the program using a `CALL CHECK` statement but it can be made to do different things depending on where it is called from through the use of the variable `itime` which takes on a distinct value depending on the location of the `CALL` statement. One may find that putting all code modifications into subroutine `CHECK` carries some advantages over simply dispersing the modifications all over the program. For instance, once all of the necessary checkpoints (places with `CALL CHECK` statement and which set `itime` to a distinct value) are put into *NUC123*, one need only compile subroutine `CHECK` each time further changes are made instead of compiling the whole program. Furthermore, all of the changes are isolated in `CHECK` so it is easy to see how much modification has been made to the program. If a modification fouls up the whole computation, it is a simple matter to test a particular set of statements in the single subroutine.

B. Suggestions For Implementing Changes

In section A above, I have provided a way of making modifications while at the same time minimizing alterations to the main body of the program. However, one may still be forced to modify the main program for two reasons: either the changes are necessary simply to get the program to run at all on one's computer or the changes are too drastic to limit them to one subroutine.

In the first instance, suppose Rocky obtains a copy of *NUC123* and puts it into his

area. He may find that he has trouble running it for a number of reasons. It could be that his computer is not a VAX machine in which case the assignment of READ/WRITE units to the terminal via SYS$COMMAND in the OPEN statement in NUC123 will cause trouble. Rocky will then have to put in the appropriate substitute to make the assigenments to the terminal. It also could be that his compiler may not understand the DO...END DO arrangement for which he will have to write a macro to make the appropriate conversion to the more conventional DO statement with statement labels. Or it can be that *NUC123* is just too big for his liking. In this case, he can chop out various subroutines and statements to reduce *NUC123* down to the desired size. He can drop any or all of the user interface subroutine with the exception of RUN if he can do without the menu options. Subroutines NUDENS, RATE5, and FLOW and functions EVAL and XINTD can be chopped out if Rocky is not interested in doing neutrino degeneracy, new reaction rates, abundance flow traces, and numerical integration. He can also cut out the myriad declaration statements (as long as an array is not being dimensioned) and comment statements if he so wishes.

In the second instance, Rocky might be planning to do something drastic such as expanding the network to look at the production of heavier elements. Here, major modification of various parts of *NUC123* may be unavoidable. To follow up what Rocky might do in this case, he can first simply alter the PARAMETER assignments of **recmax** (the maximum number of reactions which is set to 88) and of **nucmax** (the maximum number of nuclides which is set to 26) to reflect the specifications of the larger network. The options involving new reaction rates and different network sizes are a bit messy to adapt to a larger network so it may be easiest for him to bag this features. He can then could add in the characteristics for the added reactions into BLOCK DATA and put in the corresponding rates at the end of RATE4.

There may be times in which one wishes to go beyond the accuracy provided by the analytic approximation to the n $\leftrightarrow$ p rates and utilize numerical integration to solve the actual equations. I will mention here a simple trick that one can use to induce *NUC123* to do just this. One can set the electron degeneracy parameter, ξ_e, to a very small nonzero value in which case the numerical integration is done to compute these rates. A small enough value for ξ_e will not affect the reaction rates or the neutrino energy densities.

We are at the end of our budget journey into the early universe to investigate primordial nucleosynthesis. By now, Rocky is obviously a fully qualified nucleosynthesis professional and if you've been following along carefully, so are you.

VI. Acknowledgements

I would like to thank R. V. Wagoner for the use of his computer code and Rocky Kolb for providing me with the nucleosynthesis code, with valuable assistance with the code, and with a bit of fun. I would also like to acknowledge valuable discussions with M. H. Reno and H. Hodges.

This work was supported by NSF and DOE grants at the University of Chicago and by the NASA/Fermilab astrophysics group.

References

Beaudet, G. and Goret P. , 1976, Astron. & Astrophys. **49**, 415.

Boesgaard, A. M. and Steigman, G., 1985, Ann. Rev. Astron. Astrophys. **23**, 319.

Clayton, D. D.,1983, *Principles of Stellar Evolution and Nucleosynthesis*, (Chicago: University of Chicago Press).

David, Y. and Reeves, H., 1980, Phil. Trans. Roy. Soc. **A296**, 415.

Fowler, W. A., Caughlan, G. R., and Zimmerman, B. A., 1967, Ann. Rev. Astron. Astrophy. **5**, 525.

Fowler, W. A., Caughlan, G. R., and Zimmerman, B. A., 1975, Ann. Rev. Astron. Astrophy. **13**, 69.

Harris, M. J., Fowler, W. A., Caughlan, G. R., and Zimmerman, B. A., 1983, Ann. Rev. Astron. Astrophys. **21**, 165.

Kawano, L., Schramm, D. N., and Steigman, G., 1988, Ap. J. (to be published April 15).

Peebles, P. J. E., 1971, *Physical Cosmology* (Princeton: Princeton University Press).

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, *Numerical Recipes* (Cambridge: Cambridge University Press).

Scherrer, R. J., 1983, M.N.R.A.S. **205**, 683.

Schramm, D. N. and Wagoner, R. V., 1977, Ann. Rev. Nucl. Part. Sci. **27**, 37.

Wagoner, R. V., 1969, Ap. J. Suppl. No. 162, **18**, 247.

Wagoner, R. V., 1972, Ap. J. **179**, 343.

Wagoner, R. V., Fowler, W. A., and Hoyle, F., 1967, Ap. J. **148**, 3.

Weinberg, S., 1972, *Gravitation and Cosmology* (New York: John Wiley & Sons).

If you have questions concerning details of *NUC123* or wish to obtain a copy of the program, I can be contacted at either of the following locations:

Astronomy and Astrophysics Center
The University of Chicago
5640 South Ellis Avenue
Chicago, Illinois 60637
Office Phone: (312) 702-6041

NASA/Fermilab Astrophysics Center
MS 209
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510-0500
Office Phone: (312) 840-2228
Bitnet Address: Larry@FNAL