Fermi National Accelerator Laboratory

FERMILAB-Pub-92/04-A
January 1992

LET'S GO: EARLY UNIVERSE II

Primordial Nucleosynthesis
The Computer Way

Lawrence Kawano

Kellogg Radiation Laboratory
Caltech

Abstract

This is a revised description and manual for the primordial nucleosynthesis program *NUC123*, an updated and modified version of the code of R.V. Wagoner. *NUC123* has undergone a number of modest changes further enhancing its documentation and ease of use. Presented is a guide to its use followed by a series of appendices containing specific details such as a summary of the basic structure of the program, a description of the computational algorithm and a presentation of the theory incorporated into the program.

Supported in part by NASA, NAGW-2381 at the NASA/Fermilab
Astrophysics Center, Batavia, IL 60510-0500

– Program Summary –

Title of program: NUC123

Program obtainable from: Rocky Kolb and Michael Smith (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: 398 blocks total

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

No. of lines in program: 4601

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 allows 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 equation using Gaussian elimination with back substitution.

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

– Outline –

I. Introduction

A. General Comments

In 1988, I modified the nucleosynthesis program of R.V. Wagoner (Wagoner 1969 and Wagoner 1972) and introduced it as *NUC123* in a Fermilab preprint (Kawano 1988). Since then, it has seen extensive use by a number of research groups: Krauss and Romanelli (1990), Olive *et al.* (1990), Madsen (1990), Bianconi, Lee, and Ruffini (1991), and Walker *et al.* (1991) to name but a few. Since the 1988 program (as described in the Fermilab preprint) was issued, I have continued to make some modest changes to the program, generally with an aim toward simplicity and clarity. The biggest change I have made along these lines has been the elimination of the menu options for new reaction rates and reaction flow monitoring which, while having rather limited utility, imposed undue complexity upon the program (this elimination has reduced the size of the program by essentially 20%). The 1988 program is distinguished in that it does not have a version number; subsequent intermediate versions – which I have continued to distribute from time to time – are identified by a version number. The most current version (the one described in this preprint) has number 4.1. Important differences between the current and original versions are described in appendix A.

The purpose of this preprint is twofold: 1) it is a manual for anyone wanting to know how to use the program; 2) it is a comprehensive description for anyone wanting to know everything about the program. The main part of this preprint serves as a convenient manual for the first purpose while the additional appendices fulfill the second.

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. I have tried to use standard FORTRAN as much as possible to allow the running of NUC123 on a variety of machines but even so some adjustments are necessary for machines other than a VAX (a SUN in particular). Some of the changes I have made have been directed toward making the code more universal. NUC123 was further developed on the VAX/VMS system of the Kellogg Radiation Laboratory at Caltech.

B. Menu Interface

The nucleosynthesis program has been split up into 4 files: `NUC123.FOR` (containing the user interface subroutines); `NUCCOM.FOR` (containing the COMputational subroutines); `NUCRAT.FOR` (containing the reaction RATes); and `NUCINT.FOR` (with an INTerface subroutine). This segmentation of the program has facilitated the mailing of the program; it has also simplified the making of program modifications – changes may require the compilation of a single file instead of the whole program.

To run *NUC123*, all you need to do is compile each of the 4 files, link them together (`$ link NUC123, NUCCOM, NUCRAT, NUCINT` for the VAX/VMS) then run (`$ run NUC123` as long as you had listed `NUC123` first in the link statement). Having done so, you will get a nice, big greeting by the program. Pressing the <*RETURN*> key will get you to the main menu as shown in figure 2.

- HELP is intended as a place of refuge for the uninitiated. Although the program itself is generally straightforward and friendly enough on its own to let people "wing" their way through, the HELP section is sort of an "on-line" version of this manual (albeit a bit scaled-down), intended to familiarize any potential user with the basic ins and outs of the program. Entering a "1" and a <*RETURN*> will get you into the HELP submenu as shown in figure 3.
- SET COMPUTATION PARAMETERS allows you to set the values of various parameters used in determining the accuracy and duration of the computation. These include constants which constrain changes in the time-step size, the initial time-step size, initial and final temperatures, and minimum level for nuclide abundances. The options here will be covered in section II.A
- SET MODEL PARAMETERS lets you play "God": you get to set the characteristics of the universe at the time of nucleosynthesis, be it the baryon-to-photon ratio (η), neutron lifetime, or number of neutrino types. All this is discussed in detail in section II.B.
- RUN is where the real action takes place. The RUN submenu allows you to determine the size of the reaction network to be used in the computation and to specify multiple loopings of runs to cover a range of parameter space before you start the Universe sizzling in a cauldron of nucleons. More on this in section III.
- OUTPUT lets you see the results of the computation in 2 ways: 1) as an output file which can be printed out; or 2) as a listing right there on the screen. This is elaborated upon in section IV.
- Exit – It's Miller time.

(A tip for the menu user: If a response to any option request from the menu or any of the submenus is not in the given range or if one simply hits <*RETURN*>, then the response is assumed to be the last option, the one for Exit.)

C. Batch Runs

Although the menu interface is nice and convenient, there may be times you want to do a lot of runs without taking up a lot of terminal time. In this case, file `NUC123.FOR` (the only one of the 4 files that really needs any changing) can be easily converted to run in batch mode through a convertion macro such as the one listed in figure 1 (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 4 for input from file `BATIN.INC` and with unit 5 for output to file `BATOUT.INC` in the `OPEN` and `CLOSE` file statements (I had used units 7 and 8 in the 1988 preprint but no big difference). These substitutions are also made in the `PARAMETER` statements of all the subroutines which contain the appropriate `READ/WRITE` statements.

All you need to do to run this batch version of *NUC123* (given the name `NUCBAT.FOR` here for `NUC`leosynthesis `BAT`ch program) is, as before, compile `NUCBAT` and then link with `NUCCOM.OBJ`, `NUCRAT.OBJ`, and `NUCINT.OBJ`. Then, before submitting `NUCBAT` to a batch queue, you need only to edit into the file BATIN.INC all the selections you would normally put in at a terminal session. The responses from the program – which you would normally see on the screen – are stored in the file BATOUT.INC. File BATOUT.INC is nice to have because if the program bombs, you can look into BATOUT.INC to see how your commands were carried out. For instance, had you put commands into BATINT.INC incorrectly, it should be obvious from BATOUT.INC. Notice that, even in a batch run, you retain all the flexibility of an interactive session. (Hint: Erase the BATOUT.INC files to keep them from eating up your disk space.)

II. Input Parameters

A. Computational Parameters

Once you're basically oriented on the workings of the program, you are ready to roll up your sleeves to do some heavy-duty nucleosynthesis computing. The first order of business might then be for you to go into menu selection 2 (corresponding to subroutine SETCOM) to adjust the computation mechanism of the program to your liking. The submenu shown in figure 4 gives you control over the adjustments of a number of computational parameters: 2 constants which set a limit on the adjustments of the time-step, the size of the initial time-step, the starting and ending temperatures, a lower limit on nuclide abundances, and the interval of data accumulation. You also are given the opportunity to restore, in one bold stroke, all parameters to their default values (the ones shown in the figure). Each of these input parameters will now be discussed in detail.

The first two 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 the error in the quantity being evolved is limited by a predetermined accuracy. In the present case, the size of the time-step is determined by the requirement that the nuclide abundances and temperature not change too much: the first parameter (corresponding to variable cy in the program) limits the time-step from abundance changes; the second parameter (corresponding to variable ct) does the same from temperature changes. Default values for cy and ct are 0.3 and 0.03 respectively and appendix C has more details on the time-step control. Note: I have made some changes from 1988 in defining these constants (see appendix A).

The initial time-step (the third item in the submenu) has a default value of 10^{-4}. 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 a time evolution from thereon. It continues in this manner until the temperature falls below the final temperature or else accumulates the maximum number of lines of data allowed by an internal parameter in the program (set to 40 lines). Y_{min} (the sixth parameter in the submenu) puts a lower limit on nuclide abundances during the calculations (nonzero values for the Y_is are necessary to keep the matrix nonsingular in the computation of dY/dt [see appendix E]) and the default value is set to 10^{-25}. The accumulation increment (the seventh parameter) specifies the interval at which information (temperature, nuclide abundances, etc.) is recorded. The default setting is to record every 30 iterations of the Runge-Kutta routine. The accumulation interval also serves to determine the monitoring times of the Gaussian elimination process (see appendix E on this aspect). (Helpful hint: If the program runs through an unusually large number of cycles – due to strict time-step settings for instance – the program may exceed the allotted number of run cycles [30 × 40 = 1200] unless you have upped the accumulation increment. This is most likely what has happened in the case you are surprised by a negative final abundance for ^{4}He: the program stopped while the ^{4} abundance was still very small, resulting in a negative final value when 0.0025 was subtracted for radiative corrections in subroutine CHECK.)

The default values of all of the computational parameters (as well as of the model parameters) are found in the BLOCK DATA at the end of the file NUCRAT.FOR. There you can alter the default values to insert numbers more to your liking.

B. Model Parameters

With the computation mechanism tuned to desired specification, you can put together your very own nucleosynthesis recipe by fiddling around with the model parameter settings. Selection 3 (contained in subroutine SETMOD) presents the submenu shown in figure 5. You can investigate various regions of the standard model by ranging through values of the second, third, and fourth parameters; the first and last four parameters let you do some roaming outside of the standard picture.

The first parameter is a multiplicative factor which is applied to the accepted value of the gravitational constant to produce a variant value. The default value for this parameter is 1. The second parameter is the neutron lifetime in seconds and its default value is 888.54 seconds (from Smith, Kawano, Malaney 1992). 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 now in the current program version (see appendix A) the value it assumes after electron-positron annihilation; the starting value of η is calculated by multiplying this by a factor of 11/4 (see Weinberg p. 536 or Peebles p. 250). The default setting for η is $10^{-9.5}$. The fifth parameter is the value of the cosmological constant (default value is 0) in the usual units as it appears in equation D.17. The last three parameters deal with neutrino degeneracy and in the the default case, the neutrinos are considered nondegenerate. Neutrino degeneracy is discussed further in appendix G.

III. Running The Code

Having set the computation and model parameters, you're ready 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 4 of the main menu (subroutine RUN) which offers the submenu depicted in figure 6. From the figure, you can see there are two options relating to the running of the program and one option for selecting the reaction network size.

The "Go" option will set the program off on solving the differential equations for evolving the nuclide abundances, etc. When the computation is complete, a message to that effect is displayed on the screen so you can now go into the Output section to look at the final results (see section IV for discussion of the Output section).

Another run option is to cover a range of parameter space by doing a series of runs at different values of particular parameters. In the "Do Multiple Runs" option, you can specify up to three model parameters which are to be incremented through specified step sizes through a specified range. (Note: at the beginning of the runs, values in the SET MODEL PARAMETERS section will be overridden. At the end of the runs, the values will reflect the parameters settings of the final run.) You 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 starting value, final value and incremental value. If a 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 and 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) – some other arbitrary value must be submitted. The baryon-to-photon ratio undergoes a incrementation in the value of the exponent while the rest undergo a linear increase. You are then asked to confirm your specifications (in case of a mistaken entry). Once this is done for all of the parameters, the program goes off to do the multiple computations and when everything is done, it comes back with an appropriate message.

For a large number of loopings, these multiple runs can take a long time. Therefore, you can either doze off in front of the terminal, run everything in batch mode (for batch runs see section I.C), or, if you are not so picky about small computational errors, utilize beforehand the "Set Run Network" option in which you 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 60 reactions or even down to 9 nuclides and 25 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 16.

IV. Output Options

Once your computations are completed, you can go to option 5 in the main menu (handled by subroutine `OUTPUT`) to display the results. This leads to the submenu shown in figure 7 which offers you the choice of creating an output file (named `NUC123.DAT`) which can be printed or of seeing the results right there on the screen. The sample `NUC123.DAT` output file shown in figure 10 was produced using the default parameters. The parameter values for the run are shown listed first, followed by the temperature (in units of 10^9 K) and the nuclide abundances which are given in terms of the mass fraction for ^{4}He and the number density relative to that of H for the rest of the nuclides. In the second section, the output file lists the following: temperature; time (in seconds); energy density (in units of g·cm^{-3}) of photons, electrons, electron neutrinos and baryons; electron chemical potential; time-step size; baryon-to-photon ratio; and expansion rate.

If you are impatient to see the results, you can choose to flash them right there on the screen with option 2. The tables which appear on the screen are arranged in the same manner as the output file but they have been split up into four parts in order to accommodate them on the screen. You can select which one to see by using the submenu shown in figure 8: the first part contains the abundances for the usual light nuclides; the second part contains the abundances for other 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 9.

Rather than relying on the output options just mentioned, you can also create your own output file (for use with plotting packages like MONGO and TOPDRAWER). For this purpose, use the interface subroutine `CHECK` to produce an output file `NUCINT.DAT` of your own specification. Subroutine `CHECK` is presently configured to write into a file `NUCINT.DAT`, after every run, the final baryon-to-photon ratio and the final abundances for D, ^{3}He (+^{3}H), ^{4}He and ^{7}Li (+^{7}Be). Subroutine `CHECK` is equipped with the entire list of global variables (variables passed through a `COMMON` statement) used in *NUC123* and all the necessary `COMMON` statements to access those variables – it can easily be modified to suit any need. The subroutine can be called from anywhere in the main program with a `CALL CHECK` statement along with a value for variable `itime` which serves to uniquely 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); at the beginning (`itime` = 3) and end (`itime` = 8) of every run; and at the beginning of the first (`itime` = 4) and second (`itime` = 7) Runge-Kutta loops.

V. Modifying The Program

If you are a real razzle-dazzle kind of person, you'd probably want the nucleosynthesis code to do some crazy whiz-bang stuff that's just not built into it. If these contemplated changes are not too drastic, they might be all accommodated in subroutine `CHECK`, the same subroutine mentioned in the previous section. Subroutine `CHECK` is essentially an empty interface subroutine with a complete roster of `COMMON` statements so that one has easy access to all of the global variables. You can call it up from anywhere in the program using a `CALL CHECK` statement; you also can do different things with it depending on where one calls it up through the use of the variable `itime` which uniquely labels the location of the `CALL` statement. The advantage of putting all code modifications into subroutine `CHECK` (over dispersing changes throughout the program) is that one need compile only subroutine `CHECK` (file `NUCINT.FOR`) each time changes are made – rather than recompile the whole program. Furthermore, with all of the modifications localized in `CHECK`, it isolates trouble-shooting to that one subroutine.

Even with this handy subroutine around, you might still be forced to modify the main program for two reasons: the changes are necessary to get the program to run on a system different from VAX/VMS or the changes are too drastic to limit them to one subroutine.

I have tried to make the program more compatible with other systems; in particular, I have renamed the input/output unit parameters to integer names (see appendix A). There is no doubt I've overlooked other problems but I can bring to your attention these two potential difficulties: 1) the assignment of `READ/WRITE` units to the terminal via `SYS$COMMAND` in the `OPEN` statement in the program `NUC123`; the `DO...END DO` statement which is not part of the ANSI FORTRAN 77 standard.

To those contemplating major changes, I offer a few guidelines. One thing you can do is start by paring down the size of the program: if your work has nothing to do with neutrino degeneracy, you can reduce the program size somewhat by lopping out subroutine `NUDENS` and functions `EVAL` and `XINTD` as well as related statements in subroutines `START` and `THERM`. With the elimination of the NEW REACTIONS option, changing the number of nuclides and of reactions is a simple matter of changing parameters `nrec` (number of reactions) and `nnuc` (number of nuclides) and their related parameters. Finally, I hope that the documentation within the program itself, and the detailed descriptions of the program in the appendices will make the workings of the program obvious enough for you to figure out how to insert your modifications.

VI. Final Comments and Acknowledgements

The original papers (Wagoner, Fowler, and Hoyle 1967; Wagoner 1969), a review paper by Schramm and Wagoner (1977), a number of books (Peebles; Weinberg; Kolb and Turner) provide good resources for discussions on the physics of primordial nucleosynthesis. The work of Krauss and Romanelli (1990), Walker *et al.* (1991), and most recently, Smith, Kawano, and Malaney (1992) give the current status report on standard nucleosynthesis. In our paper, we compiled the latest experimental data on 12 of the most important reactions in the nucleosynthesis network (see table 2) and computed new reaction rates and their uncertainties. Then, using the program described in this manual, we have, in the manner of Krauss and Romanelli, done an extensive Monte Carlo analysis to produce results reflecting these reaction uncertainties. We used these results, together with deduced limits on primoridal abundances gleaned from a thorough and careful look at observations of the light element abundances, to put constraints on Ω_b.

I mention in passing that one can perhaps get a better feel of certain aspects of nucleosynthesis by looking at some papers which do analytical derivations of nucleosynthesis (Berstein, Brown, and Feinberg 1989; Esmailzadeh, Starkman, and Dimopoulos 1991).

I would like to express my thanks to Michael Smith for his computation of reaction rates and to David and Mimi Dickstein for their careful reading of this manuscript. This work was supported by NSF at Caltech under grant PHY90-13248.

Appendix A. Changes in The Program

- Splitting of the program into 4 separate files –
  The program is now contained in 4 separate files:
  1) NUC123.FOR
     (menu interface subroutines NUC123, HELP, SETCOM, SETMOD, RUN, OUTPUT)
  2) NUCCOM.FOR
     (computation subroutines DRIVER, START, DERIVS, ACCUM, THERM, BESSEL, KNUX, NUDENS, EVAL, XINTD, EX, SOL, EQSLIN)
  3) NUCRAT.FOR
     (reaction rate and nuclide data RATE0, RATE1, RATE2, RATE3, RATE4, BLOCK DATA)
  4) NUCINT.FOR
     (user interface subroutine CHECK)

  Each of the files can be individually compiled then linked together.
- Elimination of subroutines SETRAT and FLOW –
  Since setting new rates and monitoring processing rates can probably just as easily be accomplished by a few extra statements in the interface subroutine CHECK, both of these subroutines were chopped out. The elimination of these subroutines have greatly simplied matters in other subroutines, namely in RUN (where the incorporation of new reactions within different reaction network sizes was a real pain) and SOL. It is now much easier to change the number of reaction rates and nuclides (it just involves resetting nrec and nnuc and their related parameters).
- Elimination of other menu options –
  The units change option in the output menu has been reduced to a few lines in the program. The removal of the comment characters C in column 1 will activate the instructions to change the temperature units to MeV and mass densities to their fractional contributions to the total. The mode-of-accumulation option in the computation parameters menu has also been eliminated; the mode-of-accumulation is now set to number of Runge-Kutta cycles.
- Changes in menu default values –
  In the SET COMPUTATION PARAMETERS section, the initial time-step has been increased from 10^{-6} to 10^{-4} and ct (which as dlt90 was set at 0.1) is now set to 0.03. The consequence of the latter change has been a slight improvement in accuracy with only a small additional cost in time required (see figure 14). In the SET MODEL PARAMETERS section, the neutron lifetime (originally given as the half-life; see changes in model variables below) has a default value of 888.54 seconds (from Smith, Kawano, and Malaney 1992). In addition, the default baryon-to-photon ratio (see changes in model variables below) has been set to $10^{-9.5}$.
- Changes in COMMON statements –
  With the elimination of subroutine FLOW, common area /coeff/ disappeared; with subroutine SETRAT gone, common areas /newrat/ and /roster/ also disappeared while common area /runopt/ suffered the loss of variables trace, gtrace, itnucl, and itreac. Common areas /evolp1/, /evolp2/, and /evolp3/ replaced /evolvpr/ and /abund/ and /evolv/ was eliminated (v and associated variables were made local)

as a result of the changes (described below in "Cosmetic Changes") in subroutines DRIVER and DERIVS. Common area /compr/ reflects the changes in the computational parameters: cy and ct have replaced reg and regm; modac and itmax were removed (the latter of which was made local); and inc replaced acinc. Changes were made similarily for /compr0/. The ordering of variables was changed slightly in common area /modpr/. Common area /flags/ is somewhat smaller with the elimination of variables t9c and id

- Changes in parameter names and values –
  Parameter nucmax was renamed nnuc (for number of nuclides), nucmx2 was eliminated (replaced by nnuc), recmax=100 was redone as nrec=88 (for number of reactions), recext was deleted because of the elimination of subroutine SETRAT, and the total number of variables to be evolved was changed to nvar=29 from nmax=103 (which reflected the mistake of adding 3 to the number of reactions instead of the number of nuclides). The parameters making input/output assignments were changed from iu and ou to ir and iw because ou did not pass as an integer on some computer systems. Computation parameter itmax was changed to a fixed local parameter.
- Changes in computational variables –
  Parameter reg was renamed cy (for constraint on abundances y) and regm was eliminated, its place in the menu taken by ct (for constraint on temperature), originally the local parameter dlt90. With the restriction of the data accumulation mode to just the number of Runge-kutta cycles, the mode-of-accumulation parameter modac was eliminated and the accumulation increment acinc was reduced to an integer inc indicating the number of Runge-Kutta cycles between each accumulation of data.
- Changes in model variables –
  The baryon-to-photon ratio originally reflected the value at the start of nucleosynthesis (before the electron-photon annihilation); it is now the final value as it is today. The starting value is obtained by multiplying the inputted value by 11/4 in subroutine START at the beginning of the nucleosynthesis computation. The neutrino degeneracy variables psi have been renamed xi to correspond to its usual Greek designation.
- Elimination of some variables –
  Because of the changes mentioned above for the computational variables, dtfac (defined by regm and reg) and t9c (used in a temperature-change mode-of-accumulation) were eliminated. Counter id, used to track the monitoring of the Gaussian elimination process, has been eliminated, its role assumed by ip which had been used solely to track the accumulation of information. Variable elec has been dropped in favor of using y(2) instead.
- Changes in subroutine KNUX –
  The Chebyshev polynomials used to compute the values of modified Bessel functions $K_0(z)$ and $K_1(z)$ for $z = (m_e c^2/kT_9) \leq 2$ were replaced by polynomial approximations given by Abramozitz and Stegun 1968 (page 378, equations 9.8.1 and 9.8.3. and page 379, equations 9.8.5 and 9.8.7), the same source for the polynomial approximations given for $z = (m_e c^2/kT_9) > 2$. The resultant numerical differences are insignificant.
- Changes in subroutine EQSLIN –
  Some cosmetic changes were made, including the relabeling nit → nord and mit →

mord, the resetting of tolerance level eps to 2×10^{-4} to reduce the number of error messages, and the replacement of iloop with icnvm which, though inequivalent to iloop, serves the same purpose of providing the proper signal for doing a convergence test.

- Changes in reaction rates –
  See appendix F for details. In addition to changes in reaction rates, the index assignments for decay rates in subroutine RATE0 were changed.
- Changes in numerical constants and minus signs –
  The value of $M_u n_\gamma / T_9^3$ (M_u is the atomic mass unit), given originally as 3.37×10^4, was made more precise as 3.3683×10^4. The proportionality constant between time and temperature and the graviational constant were not themselves changed. They were, however, given separate local parameter names (const1 and const2 respectively). The elimination of the negative sign from the Hubble constant hubcst and the insertion of one in the definition of thm(11) in subroutine THERM has resulted in sign changes in variables dphdln, bar, dlndt9, and dhv in subroutine DERIVS. This was done to make the expressions in the code for the derivatives of T_9, h, and ϕ_e correspond more closely to the equations describing them in the text (see appendix D).
- Changes in interfacing with subroutine CHECK –
  In the interface subroutine (which I originally referred to as an adaptive subroutine), I have added 2 new connections: (itime = 4) in the first Runge-Kutta loop right after the derivatives calculation in DERIVS; (itime = 7) in the second Runge-Kutta loop after calling DERIVS.
- Changes in input/output format –
  I have replaced many of the different input/output formats with list-directed READ/WRITE statements.
- Cosmetic changes –
  To make the program look simpler, I have: 1) eliminated a series of stacked IF statements in the subroutine RUN and replaced them with a smaller set of statements incorporating an array qvary EQUIVALENCEd to variables c, cosmo, and xi; 2) used EQUIVALENCE statements to eliminate the numerous statements equating variables t9, hv, and phie with array v in subroutines DRIVER and DERIVS; and 3) changed index assignments (in subroutine THERM) for the thm array while reducing the array size from 20 to 14.

Appendix B. Subroutine Hierarchy and Description

Figure 11 gives the picture of the subroutine interconnections. For simplicity, I have excluded from the diagram some of the relatively minor connections, particularily those involved in neutrino degeneracy calculations. If the electron neutrino degeneracy parameter $\xi_e \neq 0$, subroutine START calls subroutine RATE1 to compute the normalization constant K for the reaction rate integrals (equation G.3). Any time these integrals need to be evaluted, subroutine RATE1 calls function EVAL (which contains the integrands to be evaluated) and function XINTD (which does the numerical integration). These same two functions are called by subroutine NUDENS to compute the energy density integals (equation G.4). Not related to neutrino degeneracy, subroutine START also calls subroutine BESSEL to compute a starting value of the electron chemical potential ϕ_e.

I will initially give here the basic picture of the operation of the code. When *NUC123* is activated, the main program NUC123 begins by calling up a set of numerical values stored in BLOCK DATA. The program next presents the user with a menu option which leads into subroutines HELP, SETCOM, SETCOM, RUN, or OUTPUT: HELP presents the user with helpful information on the code; SETCOM allows the user to change a variety of computational parameters; SETMOD offers the same for model parameters; RUN does computation runs and network size adjustments; and OUTPUT gives a look at the results. When a computation run is initated, RUN calls subroutine DRIVER which begins by calling START to get initial abundances and decay rates (from subroutine RATE0). Subroutine DRIVER uses a second-order Runge-Kutta routine (described in appendix C below) to evolve 4 quantities – the temperature T_9, the quantity h, the electron chemical potential ϕ_e, and the abundances Y_i – by computing their derivatives in each of the Runge-Kutta loops via subroutine DERIVS. Subroutine DERIVS first calls subroutine THERM to compute energy densities and pressures; BESSEL and KNUX work out the necessary Bessel functions. DERIVS then uses reaction rates computed in RATE1, RATE2, RATE3, and RATE4 to construct a matrix equation for the abundance derivatives (in subroutine SOL) that is then solved by Gaussian elimination (in EQSLIN). With the new dY/dt values, DERIVS computes the derivatives for T_9, h, and ϕ_e. As the quantities are evolved forward in the Runge-Kutta iterations, subroutine ACCUM records their values at appropriate times.

Next, I give a subroutine-by-subroutine account of the workings of the program.

- NUC123 –
  This program serves as the heart of the menu display system, presenting the user with the main menu options and calling subroutines in response to the user's selections. The program begins by initially opening files corresponding to the terminal and to data file NUC123.DAT. It then prompts the user with a "NUC123" greeting. When the user responds with a <*RETURN*>, the program intalls a set of default values into its working variables – these include reaction parameters, run options, output options, computational parameters, and model parameters. It next presents the user with the main menu selection as shown in figure 2. Depending on the selection, it calls subroutines HELP, SETCOM, SETMOD, RUN, or OUTPUT. If option 6 (or <*RETURN*>) is entered, the program exits, saving the data file NUC123.DAT if it was requested.
- HELP –
  This subroutine provides the user with a brief description about the program. Upon entry, the subroutine produces a submenu of options (figure 3) and proceeds to provide information on the selected subject. A user input of "1" advances the screen whereas anything else returns the user to the HELP submenu.
- SETCOM –
  The purpose of this subroutine is to allow the user to adjust various computation-related settings. Upon entry, the subroutine provides the user with a submenu (figure 4) which lists the computational parameters: time-step constraint from abundance change, time-step constraint from temperature change, initial time-step, initial tem-

perature, final temperature, lower limit on abundances, and accumulation increment. The user also has the option of returning everything to their default values.

- SETMOD –
  The offered submenu (figure 5) allows the user the opportunity to set the values for the gravitational constant, the neutron lifetime, the number of neutrino species, the baryon-to-photon ratio, the cosmological constant and the neutrino chemical potentials. If a non-zero value is entered for the tau neutrino chemical potential and the number of neutrino species is set less than 3, the subroutine changes the number of neutrino back to 3.
- RUN –
  This subroutine presents the user with a submenu (figure 6) which gives the user his choice of either changing the reaction network size, doing a single run, or doing multiple computations. If the user exercises the first option, he is asked to set the network size. If the entered value is not 1, 2, or 3, the subroutine assumes a default value of 1. The subroutine then adjusts the reaction network by setting variable `isize` to the appropriate number of nuclides and variable `jsize` to the appropriate number of reactions. If the user requests a single run, the subroutine then calls DRIVER. If the user requests a multiple run, the subroutine first asks for the number of variables to be varied (= "number of loopings to be done"). If the response, given in variable `jnum`, is not 1, 2, or 3, the subroutine assumes a default value of 1. The subroutine then shows the user a sub-submenu of the model parameters that are to be varied. If no selection is made, then rejection counter `knum` is incremented. Variable `inum(i)` contains the selection number and variables `rnum1(i)`, `rnum2(i)`, and `rnum3(i)` store the starting value, the terminating value, and the increment for the quantity to be varied. Asking the user for such information `jnum` number of times, the subroutine skips the runs if `jnum - knum` is zero and displays the run selections if it is not. The subroutine uses `rnum1(i)`, `rnum2(i)`, and `rnum3(i)` to compute the number of times it is going to do each DO loop (of which there are three). The workings within each loop is the same; I will discuss these in terms of the particulars of the innermost loop. The loop index here is given by `lnumb3` and the number of loopings is given by `lnum(3)`. The subroutine computes `rnumb3`, the value of the quantity varied in the innermost DO loop for that run, and if the DO loop has a valid selection number, it equates `rnumb3` to the appropriate variable name. If the quantity is in particular the baryon-to-photon ratio, the variable involved is `etal` and for this case, `rnumb3` is the log of its value. The subroutine then calls subroutine DRIVER to do a run and when a run is done, it steps up `rnumb3` to its new value. It continues in this fashion until all the requested loopings are done.
- OUTPUT –
  The OUTPUT subroutine shows the user the submenu shown in figure 7, giving the user the choice of requesting an output file or of seeing output on the screen. If the user opts for a file, the subroutine records in file NUC123.DAT the information for the last run, putting the computational and model parameters at the top then listing the temperature, nuclide abundances, energy densities, the baryon-to-photon ratio, the expansion rate, etc. (figure 10). There are 9 lines of additional code which will –

when activitated by the elimination of the "C" in column 1 – convert the temperature listing to units of MeV and the densities to their fractional contributions. If the user opts for screen-viewing, the subroutine presents the sub-submenu shown in figure 8. The selections lead to WRITE statements which reproduce the information discussed above in the form seen in figure 9.

- DRIVER –
  Appendix C, which follows, gives a fairly thorough look at this subroutine.
- START –
  Initializes flags and counters, sets up computational and model variables, and computes the initial neutron and proton abundances as well as the starting values of h, ϕ_e, and baryon density ρ_b. It also computes the deuterium abundance while setting the abundances of the other nuclides to Y_{min}. The initial neutron and proton abundances (also the deuterium abundance) are found from nuclear statistical equilibrium (complications from a nonzero ξ_e are discussed in appendix G) and h and ϕ_e are computed from equations D.1 and D.2 respectively.
- DERIVS –
  Computes the derivatives for Y_i (through subroutines SOL and EQSLIN), T_9, h, and ϕ_e. It uses energy densities and pressures (computed in subroutine THERM) to calculate the Hubble constant and reaction rates before calling SOL and EQSLIN to figure out the change in abundance, dY_i/dt. From the energy densities, pressures, and dY_i/dt, it arrives at dT_9/dt, dh/dt, and $d\phi_e/dt$ (see the latter half of appendix D).
- ACCUM –
  Transfers information to a set of output buffers (identified by -out). The abundance information is given as the density ratio to hydrogen except for hydrogen and helium which are given in terms of the mass fraction. The abundances of nuclides for ^{8}Li and up are given as a total sum. Other information that is recorded: temperature, time, energy densities ($\rho_\gamma, \rho_e, \rho_\nu, \rho_b, \rho_{total}$), ϕ_e, dt, η, and H (Hubble constant). If the termination criterion is met (output buffer filled or the accumulator called off normal accumulation interval), the termination flag ltime is raised.
- THERM –
  Computes energy densities, derivatives of the densities, pressures, and analytic approximations to the n ↔ p reaction. See the first half of appendix D (equations D.7 to D.16) and appendix F (equations F.1 and F.2).
- BESSEL –
  Computes the functions $L(z)$, $M(z)$ and $N(z)$ as defined by equations D.11-D.13.
- KNUX –
  Solves for the modified Bessel functions $K_0(z)$, $K_1(z)$, $K_2(z)$, $K_3(z)$ and $K_4(z)$ using formulas found in Abramowitz and Stegun 1968. Functions $K_0(z)$ and $K_1(z)$ for 1) $z \leq 2$ are given by polynomial approximations 9.8.5 and 9.8.7 (page 379) together with the expressions for $I_0(z)$ and $I_1(z)$ (9.8.1 and 9.8.3 on page 378); 2) $z > 2$ are given by polynomial approxmations 9.8.6 and 9.8.8 (page 379). Functions $K_2(z)$, $K_3(z)$ and $K_4(z)$ are solved by the recursion relation from the first line of 9.6.26 (page 376).
- NUDENS, EVAL, XINTD –
  See appendix G.

- SOL –
SOL begins by evaluating jsize number of reactions, first by putting the number and identification number of the reaction nuclides into convenient index variables, then going to the appropriate set of code to compute the reverse reaction rate and the coefficients for the second equality in equation E.7, putting in the coefficients in reverse order into matrix A. Having done this for all jsize reactions, SOL modifies the matrix to the form seen in equation E.8 by multiplying by dt and adding 1 to all the diagonal elements. With matrix A in its final form, SOL calls subroutine EQSLIN to perform the gaussian elimination, requesting a convergence test if: 1) the program is in the first Runge-Kutta loop and 2) counter ip has a value equal to the accumulation increment inc. Having obtained $\bar{Y}_i(t)$ from EQSLIN (in variable yx), SOL computes dY_i/dt from the first equality in equation E.7.
- EQSLIN –
Much of what goes on here is described in the latter half of appendix E (equations E.8 - E.15). Let me just detail here the nature of the convergence test which is done when the convergence monitor iconvm is equal to the accumulation increment inc. The convergence test compares xdy against a convergence tolerance eps. The first time around, this is not a real test but a trick to get EQSLIN to automatically go into solving for a correction $\delta\bar{Y}$ as xdy is just $\bar{Y}_i/\bar{Y}_i$ and this is nearly equal to 1. It computes the righthand side of equation E.15, does the triangularization adjustments on it, and then uses the already triangulatized matrix to find $\delta\bar{Y}$ from back substitution E.11. The second time around we have a true convergence test in which xdy $= \delta\bar{Y}_i/\bar{Y}_i$ is compared against the convergence tolerance eps. If any of the nuclides fail this test, 1) an error flag is raised if the current order of correction nord is equal to the desired order of correction mord; 2) it goes back to doing another iteration if nord is less then mord. (Note: the program assumes error $-\delta\bar{Y}_i$ for equation E.13 so the sign convention in the program is different than that in equations E.13-E.15.)
- RATE0 –
Called once in the beginning by subroutine START, subroutine RATE0 computes the decay rates for 10 β^- and β^+ weak interactions.
- RATE1 –
Computes the forward and reverse rates for the charged weak interactions n ↔ p. If the electron neutrino degeneracy parameter $\xi_e = 0$, then the rates are given as polynomial expansions (computed in subroutine THERM) normalized by the neutron lifetime (see appendix F). For the case it is not zero, see appendix G.
- RATE2 –
Computes the forward reactions rates for 23 reactions up to those involving mass 7 nuclei (see figure 16).
- RATE3 –
Computes the forward reactions rates for an additional 30 reactions up to those involving mass 12 nuclei (see figure 16).
- RATE4 –
Computes the forward reactions rates for an additional 24 reactions up to those involving ^{16}O (see figure 16).

- BLOCK DATA –
  This is a block of DATA statements which holds information on nuclides and reactions and contains the default values for the computational and model parameters.

Appendix C. Runge-Kutta Driver

In subroutine DRIVER, the Runge-Kutta driving routine is used to time-evolve T_9, h, and ϕ_e and the nuclide abundances Y_i under the label of the variable v. More information 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 illustrated in figure 12. We wish to evolve quantity v from time t_1 to time t_2. If we achieve this using only the derivative calculated at time t_1, the solution (point 2) is neither very accurate nor the process stable. To get beyond this first-order correction, we symmetrize the method by evaluating derivatives at both points 1 and 2 and averaging to get to a final point f.

More specifically, we have a quantity $v(t_1)$ for a particular value of t_1 (point 1) and we wish to find the value of this quantity, $v(t_2)$, at time some t_2 (point f). We begin by computing the derivative at t_1, $dv/dt(t_1)$, and using this (via linear extrapolation over the interval $\Delta t = t_2 - t_1$) to get an initial trial value $\tilde{v}(t_2)$ (point 2). This by itself is, as mentioned, accurate to only first-order. We therefore continue by computing the derivative at t_2, $d\tilde{v}/dt(t_2)$ and then averaging the two values of the derivatives to get

$$\frac{d\bar{v}}{dt}(t_1) = \frac{1}{2}\left[\frac{dv}{dt}(t_1) + \frac{d\tilde{v}}{dt}(t_2)\right] \tag{C.1}$$

from which we can extrapolate to point f to arrive at the new value $v(t_2)$:

$$v(t_2) = v(t_1) + \frac{d\bar{v}}{dt}(t_1)\Delta t \quad . \tag{C.2}$$

In the program, the first Runge-Kutta loop begins the process by finding the derivatives for the array v and returning them in the array dvdt. These quantities for point 1 are then renamed v0 and dvdt0, with v assuming the values at point 2. In the second Runge-Kutta loop, the derivatives at point 2 are computed and returned in array dvdt. Variables v then take on the values at point f.

After computing the derivatives in the first Runge-Kutta loop, subroutine DRIVER checks the accumulation criteria: 1) Is the temperature below threshold? 2) Is the time-step too small? 3) Does counter id indicate that it is time to accumulate data again? If any of these conditions are met, it calls subroutine OUTPUT to store the current values of T_9, h, ϕ_e, nuclide abundances, etc. If in addition a termination criterion (temperature too low; time-step too small; output buffer full) is satisfied, the computation is terminated; otherwise, the program goes on to adjust the time-step before computing the trial value $d\tilde{v}/dt(t_2)$ in the second Runge-Kutta loop.

Truncation errors are introduced by the linearization in the Runge-Kutta method (equation C.1) and in the abundance change equation E.7; these errors are controlled by the size of the time-step. The time-step size is determined by variables cy and ct (corresponding to the first two parameters in the SET COMPUTATION PARAMETERS submenu [figure 4]) which control the time-step by limiting the change in the abundances and temperature respectively. The parameter cy produces a prospective size for the time-step for each nuclide from the following equation:

$$\Delta t_{min} = \left| \frac{Y_i(t)}{\frac{dY_i(t)}{dt}} \right| \mathtt{cy} \left[1 + \left(\frac{log(Y_i)}{log(Y_{min})} \right)^2 \right] \tag{C.3}$$

in which the abundances are given by $Y_i = X_i/A_i$, X_i the mass fraction contained in nuclide i having atomic weight A_i. Y_{min} (the sixth parameter in the submenu) puts a lower limit on nuclide abundances during the calculations (nonzero values for Y_i are necessary to keep the matrix nonsingular in the computation of dY/dt [see appendix E]). The smallest value of Δt_{min} found amongst all of the abundance changes is retained and is compared to the Δt_{min} computed from ct and the temperature change using equation

$$\Delta t_{min} = \left| \frac{T_9}{\frac{dT_9}{dt}} \right| \mathtt{ct} \tag{C.4}$$

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

The effect of adjusting these parameters on the time-step is shown in figure 13. The unbroken line in the plot shows the size of the time-step as a function of the temperature for a run with the default values of cy = 0.3 and ct = 0.03. The horizontal line segments near the top of the plot indicate the temperatures for which equation C.4 produced a smaller Δt_{min} than equation C.3 for the default run. When we lower cy down to 0.03, we get the dashed line constrained mainly by equation C.3. When instead we lower the value of ct by an order of magnitude, the Δt_{min} from equation C.4 dominates and we get the dotted line (which traces out the $t \propto T^{-2}$ relationship). The plot suggests that as we lower ct, we lower the initial time-step accordingly.

Figures 14a-c shows the effect of changing cy (left side) and ct (right side) on the computed nuclide abundances (on top) and on the computation time (on the bottom). Figure 14a displays the results for $\eta = 10^{-10}$, figure 14b for $\eta = 10^{-9.5}$, and figure 14c for $\eta = 10^{-9}$. For the plots on the left-hand side, the nuclide abundances and computation times are given for the values of $1.0 \geq$ cy ≥ 0.03 and ct = 0.03; on the right-hand side, the plots for the nuclide abundances and computation times are for values $0.1 \geq$ ct ≥ 0.003 and cy = 0.3. All the abundances and computation times are normalized by dividing them through by the abundances and time obtained by a benchmark run which had both cy and ct at their lowest settings: cy = 0.03 and ct = 0.003. (Parameter cy cannot be taken much below 0.03, particularily for low values of η, because an instability causes the computations to bog down. In equation C.3, the $[log(Y_i)/log(Y_{min})]^2$ term biases the time-step selection toward elements with reasonably large abundances. However, if cy is pressed too small, Δt_{min} will be determined by elements whose abundances are essentially at Y_{min}. Since these abundances are held artificially at Y_{min}, the abundance change is of order $Y_{min}/\Delta t_{min}$ and the new time-step from C.3 comes out $\Delta t_{new} \sim \mathrm{cy}\Delta t_{old}$, becoming smaller every time.)

We can see from figures 14 that, as constraints on the time-step are made tighter, nuclide abundances can diverge from those computed from runs with default values by as much as 5% for Li^7 (6% for the old default value of $\mathrm{ct} = 0.1$). Given the significance of these corrections, I have produced a plot (figure 15) showing the necessary multiplicative correction factors to be used for results from default runs to produce values more closely approximating those computed with smaller time-steps. The graph shows the abundances from runs with $\mathrm{cy} = 0.03$ and $\mathrm{ct} = 0.003$ divided by the abundances from runs with default values. These muliplicative correction factors are to be applied after one has already added ^{3}H to ^{3}He, ^{7}Be to ^{7}Li, and 0.0025 (for radiative corrections, etc.) to ^{4}He and assume that ^{4}He is given by its mass fraction and the other elements by their number density relative to hydrogen. These corrections are good to better than .2% even for one more or one less number of neutrinos and for somewhat different neutron lifetimes.

Appendix D. Time Evolution of Variables

The basic purpose of the computational part of the code (which retains the structure of the old Wagoner code) is to time-evolve three quantities – the temperature T_9 (in units of 10^9 K), the electron chemical potential ϕ_e, and h 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. The time derivatives of these quantities – used by a second-order Runge-Kutta routine (appendix C) to do the time evolution – is first found for the abundances Y_i which is then used to help find the time derivatives of the other three quantities being evolved. The calculation of the time derivatives for T_9, ϕ_e, and h is the main topic of this appendix; the following appendix (appendix E) discusses in detail the computation of the time derivatives for Y_i.

I will discuss this time evolution in sequence, following the computational steps in the program (all quantities are given in cgs units). At the beginning of a run, the driving subroutine DRIVER calls subroutine START to compute 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,

$$h = M_u \frac{n_\gamma}{T_9^3} \eta = 3.3683 \times 10^4 \eta \quad , \tag{D.1}$$

and the electron chemical potential is calculated from h and the temperature,

$$\phi_e \sim \frac{ChY_p}{z^3} \left[\frac{1}{\sum_n (-)^{n+1} n L(nz)} \right] \quad , \tag{D.2}$$

with $C = (\pi^2/2) N_A (\hbar c/k)^3$ (N_A is Avogadro's number) and $L(z)$ is given by equation D.11 (equation D.2 itself derives from equations D.16 and D.22). The initial abundances of neutrons and protons given by

$$Y_n = \frac{1}{1 + e^{q/kT}} \tag{D.3}$$

$$Y_p = \frac{1}{1 + e^{-q/kT}} \tag{D.4}$$

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

$$\rho_b \sim hT_9^3 \quad , \tag{D.5}$$

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

$$t = (12\pi Gac^{-2})^{1/2} T_9^{-2} = (10.4)^2 T_9^{-2} \tag{D.6}$$

with a the Stefan-Boltzmann constant, and the weak decay rates (in subroutine RATE0).

Subroutine DRIVER then calls subroutine DERIVS to find the time derivatives of the quantities being evolved. DERIVS first goes into subroutine THERM to compute energy densities and pressures. THERM computes the photon energy density (or more precisely, the mass density in units of gm cm^{-3})

$$\rho_\gamma = \frac{\pi}{15} \frac{k^4}{(c\hbar)^3} T^4 = 8.418 T_9^4 \quad , \tag{D.7}$$

its derivative $d\rho_\gamma / dT_9$, and the photon pressure

$$\frac{p_\gamma}{c^2} = \frac{1}{3} \rho_\gamma \quad . \tag{D.8}$$

THERM then calculates the sum of the electron and positron energy densities (see Fowler and Hoyle 1964, equation B44)

$$\rho_{e^-} + \rho_{e^+} = \frac{2}{\pi^2} \frac{(m_e c^2)^4}{(\hbar c)^3} \sum_{n=1}^{\infty} \frac{(-)^{n+1}}{nz} \cosh(n\phi_e) M(nz) \quad , \tag{D.9}$$

the sum of the electron and positron pressures (Fowler and Hoyle 1964, equation B27)

$$\frac{p_{e^-} + p_{e^+}}{c^2} = \frac{2}{\pi^2} \frac{(m_e c^2)^4}{(\hbar c)^3} \sum_{n=1}^{\infty} \frac{(-)^{n+1}}{nz} \cosh(n\phi_e) L(nz) \quad , \tag{D.10}$$

and the derivatives $\partial(\rho_{e^-} + \rho_{e^+})/\partial T_9$ and $\partial(\rho_{e^-} + \rho_{e^+})/\partial \phi_e$, the latter two for use in equations D.29 and D.34. In these expressions, the sum is truncated to 5 terms, $z = m_e c^2 / kT_9$ and the functions $L(z)$, $M(z)$, and $N(z)$ (computed in subroutine BESSEL) are defined as

$$L(z) = \frac{K_2(z)}{z} \tag{D.11}$$

$$M(z) = \frac{1}{z} \left[\frac{3}{4} K_3(z) + \frac{1}{4} K_1(z) \right] \tag{D.12}$$

$$N(z) = \frac{1}{z} \left[\frac{1}{2} K_4(z) + \frac{1}{2} K_2(z) \right] \tag{D.13}$$

with $K_n(z)$ = modified Bessel functions (computed in subroutine KNUX). To round out these energy density calculations, THERM also determines the neutrino energy density

$$\rho_\nu = \frac{7}{8}\frac{\pi}{15}\frac{k^4}{(c\hbar)^3}T^4 \tag{D.14}$$

(actually computed by either tracking the decrease of the baryon density or, in the case of neutrino degeneracy, doing a numerical integration [see appendix G]), the baryon energy density (via equation D.5), and the total energy density

$$\rho_{total} = \rho_\gamma + (\rho_{e^-} + \rho_{e^+}) + \rho_\nu + \rho_b \quad . \tag{D.15}$$

In addition, THERM takes the difference of the electron and positron number densities (Fowler and Hoyle 1964, equation B6)

$$\frac{\pi^2}{2}\left[\frac{\hbar c}{m_e c^2}\right]^3 z^3(n_{e^-} - n_{e^+}) = z^3 \sum_{n=1}^{\infty}(-)^{n+1}\sinh(n\phi_e)L(nz) \tag{D.16}$$

and also computes the derivatives $\partial/\partial T_9$ (for use in equation D.23) and $\partial/\partial\phi_e$ (for equations D.23-D.25). Finally, THERM also does analytic approximations to the reaction rates for n ↔ p (see appendix F).

With the total energy density thus determined, DERIVS uses the Friedmann equation

$$H^2 = \frac{8\pi}{3}G\left(\rho_{total} + \frac{\lambda}{3}\right) \tag{D.17}$$

(H is the expansion rate, G is Newton's constant, and λ is the cosmological constant) to get the expansion rate and with the computed baryon densities, calls up RATE1 - RATE4 to get the reaction rates. It then calls SOL and EQSLIN to find the abundance derivatives dY_i/dt (appendix E).

Subroutine DERIVS then uses the abundance derivatives, the energy densities and pressures, and the expansion rate to compute the derivatives for T_9, h, and ϕ_e from

$$\frac{dT_9}{dt} = \frac{dr}{dt}\bigg/\frac{dr}{dT_9} \tag{D.18}$$

$$\frac{dh}{dt} = -3h\left[\frac{1}{R}\frac{dR}{dt} + \frac{1}{T_9}\frac{dT_9}{dt}\right] \tag{D.19}$$

$$\frac{d\phi_e}{dt} = \frac{\partial\phi_e}{\partial T_9}\frac{dT_9}{dt} + \frac{\partial\phi_e}{\partial r}\frac{dr}{dt} + \frac{\partial\phi_e}{\partial S}\frac{dS}{dt} \tag{D.20}$$

where R is the scale factor, $r = \ln(R^3)$, and $S = \sum_i Z_i Y_i$. I will now go into the details of these 3 expressions and I might as well start with the simple one for h. Equation D.19 comes simply from expression D.5: $h \sim \rho_b/T_9^3 \sim 1/R^3T_9^3$.

The partial derivatives for equation D.20 are computed from the equation for charge conservation

$$n_{e^-} - n_{e^+} = N_A h T_9^3 S \tag{D.21}$$

where N_A is Avogadro's number. This equation can be rewritten so that the left-hand side is made the same as the left-hand side of equation D.16:

$$\frac{\pi^2}{2}\left[\frac{\hbar c}{m_e c^2}\right]^3 z^3 (n_{e^-} - n_{e^+}) = \frac{\pi^2}{2}\left[N_A \left(\frac{\hbar c}{k}\right)^3 hS\right] \ . \tag{D.22}$$

We can call the left-hand side function $N = N(T_9, \phi_e)$ and the right-hand side function $M = M(r, S, T_9)$. Taking derivatives of both sides with respect to T_9, r, and S, we get the needed partial derivatives:

$$\left.\frac{\partial M}{\partial T_9}\right|_{r,S} = \frac{\partial N}{\partial T_9} + \frac{\partial N}{\partial \phi_e}\frac{\partial \phi_e}{\partial T_9} \Longrightarrow \frac{\partial \phi_e}{\partial T_9} = \left(\frac{\partial N}{\partial \phi_e}\right)^{-1}\left[\frac{\partial M}{\partial T_9} - \frac{\partial N}{\partial T_9}\right] \tag{D.23}$$

$$\left.\frac{\partial M}{\partial r}\right|_{T_9,S} = \frac{\partial N}{\partial \phi_e}\frac{\partial \phi_e}{\partial r} \Longrightarrow \frac{\partial \phi_e}{\partial r} = \left(\frac{\partial N}{\partial \phi_e}\right)^{-1}\frac{\partial M}{\partial r} \tag{D.24}$$

$$\left.\frac{\partial M}{\partial S}\right|_{T_9,r} = \frac{\partial N}{\partial \phi_e}\frac{\partial \phi_e}{\partial S} \Longrightarrow \frac{\partial \phi_e}{\partial S} = \left(\frac{\partial N}{\partial \phi_e}\right)^{-1}\frac{\partial M}{\partial S} \ . \tag{D.25}$$

The partial derivatives of N are computed in subroutine `THERM`, those for M in `DERIVS`.

For the temperature derivative in equation D.18, $dr/dt = 3H$ (H the expansion rate) and for dr/dT_9, we need to start from the conservation of energy

$$\frac{d}{dt}(\rho R^3) + \frac{p}{c^2}\frac{d}{dt}(R^3) + R^3 \left.\frac{d\rho}{dt}\right|_{T_9=const} = 0 \tag{D.26}$$

with the third term taking into account the energy changes introduced by nucleosynthesis. This can be converted to an equation for dr/dT_9,

$$\frac{dr}{dT_9} = -\frac{\frac{d\rho}{dT_9}}{\rho + \frac{p}{c^2} + (\frac{1}{dr/dt})\frac{d\rho}{dt}|_{T_9}} \ , \tag{D.27}$$

and using ρ_e to express $(\rho_{e^-} + \rho_{e^+})$ and p_e for $(p_{e^-} + p_{e^+})$, this can be written out as

$$\frac{dr}{dT_9} = -\frac{\frac{d\rho_\gamma}{dT_9} + \frac{d\rho_e}{dT_9} + \frac{d\rho_b}{dT_9}}{\rho_\gamma + \frac{p_\gamma}{c^2} + \rho_e + \frac{p_e}{c^2} + \frac{p_b}{c^2} + \frac{1}{dr/dt}(\frac{d\rho_b}{dt}|_{T_9} + \frac{d\rho_e}{dt}|_{T_9})} \ . \tag{D.28}$$

The photon terms are straightforward and are given by equations D.7 and D.8. For the electrons and positrons, the term $d\rho_e/dT_9$ is given as

$$\frac{d\rho_e}{dT_9} = \frac{\partial \rho_e}{\partial T_9} + \frac{\partial \rho_e}{\partial \phi_e}\frac{\partial \phi_e}{\partial T_9} \tag{D.29}$$

and the energy density and pressure come from equations D.9 and D.10. Neutrinos do not appear in equation D.28 because the program assumes that they have decoupled. The baryon energy density, as given by equation D.5, would go strictly as $1/R^3$ and thus would drop out of equation D.28. However, one derivative term remains as the baryon energy density is more completely given by (Wagoner 1969, equation 2)

$$\begin{aligned} \rho_b &= n_b\left[M_u + \sum_i \left(\Delta M_i + \frac{3kT}{2c^2}\right)Y_i\right] \\ &= hT_9^3\left[1 + \sum_i \left(\frac{\Delta M_i}{M_u} + \zeta T_9\right)Y_i\right] \end{aligned} \tag{D.30}$$

where ζ is just a constant equal to 1.388×10^{-4}. This remaining part of the temperature derivative is given by

$$\frac{d\rho_b}{dT_9} = hT_9^3\zeta\sum_i Y_i \quad , \tag{D.31}$$

while the baryon pressure is given by

$$\begin{aligned} p_b &= n_b kT\sum_i Y_i \\ &= hT_9^3\left(\frac{2}{3}\zeta T_9\sum_i Y_i\right) \quad . \end{aligned} \tag{D.32}$$

The final two terms in the denominator of equation D.28 are the time derivative of the baryon energy density, given as

$$\left.\frac{d\rho_b}{dt}\right|_{T=const} = hT_9^3\sum_i\left(\zeta T_9 + \frac{\Delta M_i}{M_u}\right)\frac{dY_i}{dt} \quad , \tag{D.33}$$

and the time derivative of the electron-positron energy density,

$$\left.\frac{1}{(dr/dt)}\frac{d\rho_e}{dt}\right|_{T=const} = \frac{\partial\rho_e}{\partial\phi_e}\left(\frac{\partial\phi_e}{\partial r} + \frac{\partial\phi_e}{\partial S}\frac{\partial S}{\partial t}\frac{1}{dr/dt}\right) \quad . \tag{D.34}$$

After DERIVS obtains the time derivatives of T_9, h, and ϕ_e from equations D.18 - D.20, it transfers these values to DRIVER which then continues along the Runge-Kutta process described in appendix C.

Appendix E. Calculating dY/dt

All reactions considered in the program contain at most four different nuclides and thus are of the form

$$N_i({}^{A_i}Z_i) + N_j({}^{A_j}Z_j) \leftrightarrow N_k({}^{A_k}Z_k) + N_l({}^{A_l}Z_l) \tag{E.1}$$

where N_i is the number and A_i is the atomic number of nuclide i. The abundance changes for nuclide i are given (see equation 28 of Wagoner 1969) as changes in Y_i ($Y_i = X_i/A_i$, X_i the mass fraction contained in nuclide i) through the equation

$$\frac{dY_i}{dt} = \sum_{j,k,l} N_i\left(- \frac{Y_i^{N_i} Y_j^{N_j}}{N_i! N_j!}[ij]_k + \frac{Y_l^{N_l} Y_k^{N_k}}{N_l! N_k!}[lk]_j \right) \tag{E.2}$$

in which $[ij]_k$ is the forward reaction rate, $[lk]_j$ is the reverse rate and abundance changes are summed over all forward and reverse reactions involving nuclide i.

Once the forward reaction rates are computed in subroutines `RATE1` - `RATE4`, subroutine `DERIVS` then calls subroutine `SOL` which builds the matrix equation for the abundance changes. Subroutine `SOL` uses the method of implicit differencing because the right-hand side of equation E.2 is a small difference of large numbers due to the near equality of the forward and reverse rates at high temperatures – normally requiring 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

$$Y' = -CY \tag{E.3}$$

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

$$Y_{n+1} = Y_n + \Delta t Y_n' = (1 - C\Delta t)Y_n \tag{E.4}$$

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 uses instead

$$Y_{n+1} = Y_n + \Delta t Y_{n+1}' = \frac{Y_n}{(1 + C\Delta t)} \tag{E.5}$$

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

$$\mathbf{Y}_{n+1} = (\mathbf{1} + \mathbf{C\Delta t})^{-1} \cdot \mathbf{Y_n} \quad . \tag{E.6}$$

In subroutine `SOL`, the abundance change equation E.2 is replaced by the linearized equation

$$\begin{aligned} \frac{dY_i}{dt}(t) &= \frac{\bar{Y}_i(t) - \tilde{Y}_i(t - \Delta t)}{\Delta t} \\ &= \sum_{i,j,k} - \frac{N_i[ij]_k}{N_i! N_j! (N_i + N_j)} \\ &\quad \times [N_i Y_i^{N_i - 1} Y_j^{N_j} \bar{Y}_i(t) + N_j Y_i^{N_i} Y_j^{N_j - 1} \bar{Y}_j(t)] \\ &\quad + - \frac{N_i[lk]_j}{N_k! N_l! (N_k + N_l)} \\ &\quad \times [N_k Y_k^{N_k - 1} Y_l^{N_l} \bar{Y}_k(t) + N_l Y_k^{N_k} Y_l^{N_l - 1} \bar{Y}_l(t)] \end{aligned} \tag{E.7}$$

in which the barred abundances are computed solely to find the abundance change from the first equality and the abundances with the tildes are those of the previous time-step point. (Note: the convention used here for the bars and tildes are different from that in appendix C.) The equation for the second equality can be written in the form of equation E.6

$$(A_{ij})[\bar{Y}_j(t)] = [\tilde{Y}_i(t-\Delta t)] \tag{E.8}$$

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 E.8 can be triangularized to look like

$$\begin{pmatrix} A'_{11} & A'_{12} & A'_{13} & \dots & A'_{1n} \\ 0 & A'_{22} & A'_{23} & \dots & A'_{2n} \\ 0 & 0 & A'_{33} & \dots & A'_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A'_{nn} \end{pmatrix} \cdot \begin{pmatrix} \bar{Y}_1 \\ \bar{Y}_2 \\ \bar{Y}_3 \\ \vdots \\ \bar{Y}_n \end{pmatrix} = \begin{pmatrix} \tilde{Y}'_1 \\ \tilde{Y}'_2 \\ \tilde{Y}'_3 \\ \vdots \\ \tilde{Y}'_n \end{pmatrix} \tag{E.9}$$

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 $\bar{Y}_n$, the final line of the matrix equation gives

$$\bar{Y}_n = \tilde{Y}'_n / A'_{nn} \tag{E.10}$$

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

$$\bar{Y}_i = \frac{1}{A'_{ii}}\left[\tilde{Y}'_i - \sum_{j=i+1}^{N} A'_{ij}\bar{Y}_j\right] \quad . \tag{E.11}$$

In the program, entries for the matrix equation E.9 are inserted in reverse order so that the equations for nuclides with substantial abundances are solved first.

At monitoring intervals specified by the accumulation increment, subroutine EQSLIN checks for convergence and, for good measure, does an iterative improvement of these solutions. Drawing upon material from *Numerical Recipes* (section 2.7) again, we see that when we solve for the equation

$$\mathbf{A} \cdot \bar{\mathbf{Y}} = \mathbf{Y} \quad , \tag{E.12}$$

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

$$\mathbf{A} \cdot (\bar{\mathbf{Y}} + \delta\bar{\mathbf{Y}}) = \mathbf{Y} + \delta\mathbf{Y} \quad . \tag{E.13}$$

Subtracting equation E.12 from E.13 gives

$$\mathbf{A} \cdot \delta\bar{\mathbf{Y}} = \delta\mathbf{Y} \tag{E.14}$$

into which the solution for $\delta\mathbf{Y}$ from equation E.13 can be inserted to give

$$\mathbf{A} \cdot \delta\bar{\mathbf{Y}} = \mathbf{A} \cdot (\bar{\mathbf{Y}} + \delta\bar{\mathbf{Y}}) - \mathbf{Y} \quad . \tag{E.15}$$

The right-hand side of this equation is known: we can thus solve for $\delta\bar{\mathbf{Y}}$. Since equation E.14 is in the form of the original equation (equation E.8), this can be solved using the same matrix solution given in equations E.9 - E.11. This process, however, does not improve the solution by much: $\sim$ .1% difference even if correction $\delta\bar{\mathbf{Y}}$ is added to $\bar{\mathbf{Y}}$ every time (at the cost of 35% more computing time). The procedure serves mainly to check that the relative error $\delta\bar{Y}_i/\bar{Y}_i$ is smaller than some tolerance level given by eps.

Appendix F. Reaction Rates

The program incorporates a network of 88 reactions interlinking 26 nuclides (figure 16). The functional forms of the reaction rates are discussed in Fowler, Caughlan, and Zimmerman 1967, in Wagoner 1969 and in Clayton's book *Principles of Stellar Evolution and Nucleosynthesis*. Numerical values for the rates themselves can be found in the two papers and in the more recent work of Fowler, Caughlan, and Zimmerman 1975, Harris et al. 1983, and Caughlan and Fowler 1988. Table 1 provides information on literature references for the reaction rates.

In this appendix, I will focus on the n ↔ p reaction and on 11 other primary reactions which are illustrated in figure 17. The n → p and the p → n reaction rates are approximated in subroutines RATE1 and THERM with the polynomial expansions

$$\lambda_{n\to p} = \frac{1}{\tau}\left(1 + \frac{0.565}{z} - \frac{6.382}{z^2} + \frac{11.108}{z^3} + \frac{36.492}{z^4} + \frac{27.512}{z^5}\right) \tag{F.1}$$

$$\lambda_{p\to n} = \frac{1}{\tau}\left(\frac{5.252}{z} - \frac{16.229}{z^2} + \frac{18.059}{z^3} + \frac{34.181}{z^4} + \frac{27.617}{z^5}\right)e^{-qz} \tag{F.2}$$

with τ the neutron lifetime, $q = (m_n - m_p)/m_e$ and $z = m_e c^2/kT_\gamma$. A plot of these rates versus temperature is shown in figure 18a. Figure 18b shows the ratio of these analytical rates to those obtained from the numerical integration of the integrals given in G.3. We see that for the most part, the approximations stay within a percent of the numerically computed rates. (The numerical integration can be activated by inputting a nonzero but negligibly small value for ξ_e.) The resulting differences in the final abundances of the light elements are rather small: .2% for d, .1% for ^{3}He, .4% for ^{4}He and .4% for ^{7}Li (at $log(\eta) = -9.5$).

Dicus *et al.* 1982 have made a study of corrections to the computed light element abundances which include not only this change due to doing the integration numerically but also correct treatment of Coulomb corrections, radiative corrections, the effect of the plasma on the mass of the electron, and the heating of electron neutrinos in e^+e^- annihilation. They conclude that the sum total of these effects is to systematically reduce the ^{4}He abundance by $\Delta Y = 0.0025$ and to change the abundances in the other elements by $1-2\%$. The nucleosynthesis program accounts for the decrease in ^{4}He by subtracting $\Delta Y = 0.0025$ from the ^{4}He abundance in subroutine CHECK.

In a recent paper (Smith, Kawano, and Malaney 1992), a number of us worked out new values and uncertainties for the neutron lifetime – which normalizes the n → p reaction –

and for 11 of the most important reactions in the network. In this paper, we investigated the relative importance of the reactions (determining the primacy of these 12), compiled experimental information on these reactions, and determined their rates and uncertainties. I reproduce here some of our results in Table 2 which displays the values of neutron lifetime and the reaction rates and gives the uncertainties of these values. The experiments and data relevant to the derivation of these reaction rates are discussed in detail in our paper.

Appendix G. Neutrino Degeneracy

The neutrino degeneracy parameters are defined as $\xi = \mu/T$ with μ the chemical potential and T the temperature; in the program, there are three of these parameters corresponding to the 3 neutrino species. Details concerning the cosmology of neutrino degeneracy can be found in the literature (Wagoner, Fowler, and Hoyle 1967, Beaudet and Goret 1976, David and Reeves 1980, Scherrer 1983, Boesgaard and Steigman 1985, Bianconi *et al.* 1991, Malaney and Mathews 1991, Kang 1991, and Kang and Steigman 1991). The basic consequences are the effect on the neutron-proton ratio from a nonzero ξ_e and the change in the neutrino energy densities from a nonzero ξ_ν in general. With a nonzero ξ_e, the initial neutron and proton abundances (computed in subroutine START) are altered from their forms in equations D.3 and D.4 to

$$Y_n = \frac{1}{1 + e^{q/kT+\xi_e}} \tag{G.1}$$

$$Y_p = \frac{1}{1 + e^{-q/kT-\xi_e}} \tag{G.2}$$

and the n → p reaction rate (see Scherrer 1983; Beaudet and Goret 1976) generalizes to

$$\lambda_{n\rightarrow p} = K \int_1^\infty dx \frac{x(x+q)^2(x^2-1)^{1/2}}{(1+e^{-xz})[1+e^{(x+q)z_\nu+\xi_e}]} \tag{G.3}$$

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

with $q = (m_n - m_p)/m_e, z = m_e c^2/kT_\gamma, z_\nu = m_e c^2/kT_\nu$, and K the normalization constant computed by requiring that equation G.3 go to $1/\tau$ at low temperature. The p → n reaction rate is given by $\lambda_{p\rightarrow n} = \lambda_{n\rightarrow p}(-q, -\xi_e)$. In the program, these rates are numerically integrated as opposed to the nondegenerate case in which the rate is analytically evaluated using an approximation in subroutine THERM (see appendix F for comparison of analytic vs. numerical rates). For nonzero ξ_ν in general, the neutrino energy densities (see Scherrer 1983) become

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

which are evaluated in subroutine NUDENS.

The numerical integration of the n $\leftrightarrow$ p reactions is set up in subroutine RATE1 where appropriate upper limits to the reaction rate integrals are computed, determined by the overflow and underflow limits of the VAX machine: for each term of integral G.3 (and its counterpart for p $\rightarrow$ n), the maximum allowable value for x for each of the exponents in the denominator is determined and the greater of the two is chosen as the upper limit. Subroutine RATE1 then calculates these integrals numerically (using the integrands contained in function EVAL) via function XINTD which uses the method of $n = 6$ Gaussian integration (see Abramowitz and Stegun 1968, Table 25.4).

The evaluation of the neutrino energy densities (equation G.4) is done in subroutine NUDENS. The subroutine applies the following approximations: in the case of small ξ_ν,

$$\rho_\nu + \rho_{\bar{\nu}} = \frac{\pi^2}{15} T_\nu^4 \left(\frac{7}{8} + \frac{15}{4\pi^2} \xi_\nu^2 + \frac{15}{8\pi^4} \xi_\nu^4 \right) \quad ; \tag{G.5}$$

in the case of large ξ_ν,

$$\rho_\nu + \rho_{\bar{\nu}} = \frac{1}{8\pi^2} (T_\nu \xi_\nu)^4 \left[1 + \left(\frac{2\pi^2}{\xi_\nu^2} \right) \right] \quad . \tag{G.6}$$

If these approximations do not apply, then NUDENS calls function XINTD to numerically integrate expression G.4, the integrands contained in function EVAL. In subroutine THERM (which calls NUDENS), the energy densities found from G.4, G.5, and G.6 are multiplied by $12.792 = k^4/(\hbar^3 c^5)$ to include constants that were left out of these expressions.

(Note: The program uses units of MeV in computing the reaction rate integrals in equation G.3 but retains units of 10^9K in computing the energy density expressions G.4, G.5, and G.6. This may cause some confusion.)

Appendix H. Listing and Glossary of Variable Names

a (nnuc,nnuc) DOUBLE PRECISION [/lncoef/] The matrix A in equations E.8 - E.15.

a0 (nnuc,nnuc) DOUBLE PRECISION [local to EQSLIN] Same as above matrix A except in its pristine form before triangularization.

am (nnuc) REAL [/nucdat/] An array containing the atomic numbers of the 26 nuclides from n to ^{16}O. Values stored in BLOCK DATA.

b (nnuc) REAL [/lncoef/] The right-hand vector $\tilde{Y}_i(t-\Delta t)$ from equation E.8 (contains the values from array y0 in inverse order).

bar REAL [local to DERIVS] Contains the baryon pressure, expression D.32 and $1/(dr/dt)$ times D.33.

bdln REAL [local to SOL] Equal to $10^{-5}(dr/dt) = 3 \times 10^{-5} H$ and is used to determine if the matrix elements of a0 above are to be set to zero.

bk0 - bk4 REAL [/kays/] Contains the values of the modified Bessel functions $K_0(z)$, $K_1(z)$, $K_2(z)$, $K_3(z)$ and $K_4(z)$.

bl1 - bl5 REAL [/bessel/] Evaluation of function $L(z)$ defined in equation D.11.

blz (5) REAL [local to BESSEL] Equivalenced to bl1 - bl5.

bm1 - bm5 REAL [/bessel/] Evaluation of function $M(z)$ defined in equation D.12.

bmz (5) REAL [local to BESSEL] Equivalenced to bm1 - bm5.

bn1 - bn5 REAL [/bessel/] Evaluation of function $N(z)$ defined in equation D.13.

bnz (5) REAL [local to BESSEL] Equivalenced to bn1 - bn5.

c (3) REAL [/modpr/] This array is used for input in the SET COMPUTATIONAL PARAMETERS section. c(1) is variation of gravitational constant and is used in START to set the gravitational constant. c(2) is neutron lifetime (sec) and is equated to tau in START. c(3) is number of neutrino species and its value is passed onto xnu in START.

c0 (3) REAL [/modpr0/] Default values for the above array c (values of 1., 888.541 and 3. stored in BLOCK DATA).

ci,cj,ck,cl REAL [local to SOL] The coefficients for the terms in the rate equation E.7. ci, for instance, is the coefficient in front of $\bar{Y}_i(t)$ with the first N_i coming from ri (see ri, rj, rk, rl below).

cl PARAMETER [=1.e-16] Lower limit on size of time-step used in DRIVER. If the time-step dips below this value times dlt9dt (see below), the computation is terminated.

cnorm REAL [/nupar/] Normalizing constant K for the n ↔ p seen in equation G.3.

const1 PARAMETER [=0.09615] Relation between time and temperature as given by D.6. This number is dependent on the relativistic degrees of freedom (on the temperature that you start at and on the number of particles you put in).

const2 PARAMETER [=6.6700e-8] Value for the gravitational constant.

cosmo REAL [/modpr/] Cosmological constant as seen in the Friedmann equation D.17.

cosmo0 REAL [/modpr0/] Default value of cosmological constant (value of 0 stored in BLOCK DATA).

ct REAL [/compr/] Time-step limiting constant on the temperature (see equation C.4).

ct0 REAL [/compr0/] Default value for ct. Value of 0.03 stored in BLOCK DATA.

cx DOUBLE PRECISION [local to EQSLIN] Scaling factor in triangularization of matrix A.

cy REAL [/compr/] Time-step limiting constant on abundances (see equation C.3).

cy0 REAL [/compr0/] Default value for cy. Value of 0.3 stored in BLOCK DATA.

dhv REAL [/evolp2/] Change in hv as defined by equation D.19.

dlndt9 REAL [local to DERIVS] Expression for dr/dT_9 as given by equation D.28.

dlt9dt REAL [/time/] $(1/T_9)dT_9/dt$. Used to solve for dh/dt (equation D.19) and to limit the time-step in conjunction with parameter cl (see cl above).

dm (nnuc) REAL [/nucdat/] Mass excess of nuclide. The $\Delta M_i/M_u$ seen in equation D.30. Values stored in BLOCK DATA.

dphdln REAL [local to DERIVS] $d\phi_e/dr$ as given in equation D.24.

dphdt9 REAL [local to DERIVS] $d\phi_e/dT_9$ as given in equation D.23.

dphdzy REAL [local to DERIVS] $d\phi_e/dS$ as in equation D.25.

dphie REAL [/evolp2/] Change in chemical potential as expressed in equation D.20.

dt REAL [/time/] The time-step.

dt0 REAL [/varpr0/] Default initial time-step. Value of 10^{-4} stored in BLOCK DATA.

dt1 REAL [/varpr1/] Initial time-step. dt is set to dt1 in START.

dt9 REAL [/evolp2/] Change in temperature as given by equation D.18.

dtl REAL [local to DRIVER] The smallest Δt_{lim} gotten from equation C.3.

dtmin REAL [local to DRIVER] The smallest Δt_{lim} gotten from both equation C.3 and C.4.

dtout (itmax) REAL [/outdat/] The output time-step accumulation buffer.

dvdt (nvar) REAL [local to DRIVER] Time derivatives for T_9, h, ϕ_e and Y_i.

dvdt0 (nvar) REAL [local to DRIVER] Value of derivatives computed during first Runge-Kutta loop. See discussion following equation C.2.

dydt (nnuc) REAL [/evolp2/] Change in number abundances: dY_i/dt.

eps PARAMETER [=2.e-4] Tolerance for convergence of Gaussian elimination process in EQSLIN.

eta0 REAL [/varpr0/] Default baryon-to-photon ratio.

eta1 REAL [/varpr1/] Baryon-to-photon ratio as it is today. (It is multiplied by 11/4 in subroutine START to get the value before e^-e^+ annihilation.)

etaout (itmax) REAL [/outdat/] The output buffer for the baryon-to-photon ratio.

f (nrec) REAL [/rates/] Forward reaction rate $N_A{<}\sigma v{>}$. Computed in RATE0 - RATE4.

g REAL [/modpr/] Gravitational constant set to const1*c(1) in START.

hubcst REAL [/therm/] The Hubble Constant; the expansion rate of the universe. Value determined by Friedmann equation (D.17) in DERIVS.

hubout (itmax) REAL [/outdat/] The output buffer for the expansion rate.

hv REAL [/evolp1/] Defined by equation D.1 and initialized in subroutine START by eta1.

i,j,k,l INTEGER [local to SOL] Short index names equated to ii, jj, kk, ll (which contain the reaction nuclide numbers).

icnvm INTEGER [local to EQSLIN] Convergence monitor. Initiates the convergence check described by equations E.12 - E.15 when counter ip is equal to inc (the desired accumulation frequency) during the first Runge-Kutta loop.

ierror INTEGER [local to SOL and EQSLIN] Nuclide number of matrix element which does not converge. This means that the $\delta\bar{Y}/\bar{Y}$ for this nuclide was larger than the given tolerance set by eps.

iform (nrec) INTEGER [/recpr/] Reaction-type code (1-11). The eleven types of reactions are encoded in si, sj, sk and sl which give the number of nuclides involved in the reaction (see the DATA statements in subroutine SOL); SOL uses iform to treat each configuration differently.

ii (nrec) INTEGER [/recpr/] Incoming nuclide type (1-26). Indicates heavier of the incoming nuclides.

inc INTEGER [/compr/] Accumulation increment. Used to determine the number of Runge-Kutta iterations between putting data into the output buffers and also doing the convergence test for the Gaussian elimination solution.

inc0 INTEGER [/compr0/] Default accumulation increment. Value of 30 stored in BLOCK DATA.

ind INTEGER [local to SOL] Equated to iform for use as a convenient index.

inum INTEGER [local] Used throughout the subroutines of file NUC123.FOR, it contains the selection number inputted by the user.

inum (3) INTEGER [local to RUN] In multiple runs option. Selection number for parameter to be varied.

inumb INTEGER [local to RUN] Selection number for reaction network size.

ip INTEGER [/flags/] The accumulation counter used to compare with inc. It gives the number of iterations after outputting a line and if this number equals inc, then a convergence test is done in EQSLIN and the accumulator ACCUM is called.

ir PARAMETER [=1] Input unit number. During an interactive session, this is set to the terminal.

irun INTEGER [/runopt/] The run network size (3,2,1) which depends on the network size selection made in subroutine RUN.

is INTEGER [/flags/] Number of total iterations for computation run. Unlike ip, this counter is never reset. Its only purpose is to delay time-step adjustments in DRIVER until the second Runge-Kutta iteration.

isize INTEGER [/runopt/] Number of nuclides in computation. This can be set to 3 different values (9, 18, 26) depending on the network size selection made in subroutine RUN.

it INTEGER [/flags/] Accumulation counter which tracks the number of times information has been stored in the output buffer.

iter PARAMETER [=50] Set in subroutines NUDENS and RATE1 and transferred to integration function XINTD as nq. The segment between the upper limit xhi and the lower limit xlo is divided into nq = iter intervals of equal spacing for computation using gaussian integration.

itime INTEGER [/check/] Time check. It uniquely identifies a checkpoint in the program, allowing the activation of a particular set of statements in subroutine CHECK. The prepositioned checkpoints are listed at the end of section V.

itmax PARAMETER [=40] The size of the output accumulation buffer. It sets the maximum number of output lines that can be printed.

iw PARAMETER [=1] Output unit number. During an interaction session, this is set to the terminal.

jj (nrec) INTEGER [/recpr/] Incoming light nuclide type (1-6). Indicates lighter of incoming nuclides.

jnum INTEGER [local to RUN] In multiple runs option. Number of loopings to be done.

jsize INTEGER [/runopt/] Number of reactions in computation. This can be set to 3 different values (34, 64, 88) depending on the network size selection made in subroutine RUN.

kk (nrec) INTEGER [/recpr/] Outgoing light nuclide type (1-6). Indicates lighter of outgoing nuclides.

knuc PARAMETER [=9] Total number of nuclides for irun = 3.

knum INTEGER [local to RUN] In multiple runs option. Number of loopings rejected.

krec PARAMETER [=34] Total number of nuclear reactions for irun = 3.

lchose INTEGER [local to RUN] In multiple runs option. Alphanumeric user response for confirming selected parameter range.

ll (nrec) INTEGER [/recpr/] Outgoing nuclide type (1-26). Indicates heavier of outgoing nuclides.

lnuc PARAMETER [=18] Total number of nuclides for irun = 2.

lnum (3) INTEGER [local to RUN] In multiple runs option. End values for loop indices.

lnumb1 INTEGER [local to RUN] In multiple runs option. Loop index (outer loop).

lnumb2 INTEGER [local to RUN] In multiple runs option. Loop index (middle loop).

lnumb3 INTEGER [local to RUN] In multiple runs option. Loop index (inner loop).

loop INTEGER [local to DRIVER, DERIVS, SOL] Indicates either first or second part of Runge-Kutta iteration.

lrec PARAMETER [=64] Total number of nuclear reactions for irun = 2.

ltime INTEGER [/flags/] Indicates if termination criterion is also satisfied.

mbad INTEGER [/flags/] Indicates if gaussian elimination terminated because of a zero value in a diagonal element of matrix A or if the convergences of the solutions were larger than tolerance eps.

mord PARAMETER [=1] The order of iterative corrections desired.

mvar INTEGER [local to DRIVER] Total number of variables to be evolved = the number of nuclides + 3 (for T_9, h and ϕ_e).

nnuc PARAMETER [=26] Number of nuclear reactions (maximum available).

nord INTEGER [local to EQSLIN] Counter giving the order of iterative correction.

nout INTEGER [/outopt/] Counter keeping track of the number of output requests. For the first request, subroutine OUTPUT inserts a title for the output file.

nrec PARAMETER [=88] Number of nuclides in calculation (maximum available).

nu INTEGER [/nupar/] Index for neutrino type (1-3).

nvar PARAMETER [=29] Maximum number of variables to be evolved = the maximum number of nuclides + 3 (for T_9, h and ϕ_e).

outfile LOGICAL [/outopt/] Indicates if output file has been requested (in which case the output file is saved).

phie REAL [/evolp1/] Chemical potential of electrons and positrons.

q PARAMETER [=2.531] The neutron-proton mass difference $(m_n - m_p)/m_e$.

q9 (nrec) REAL [/recpr/] Energy release in reaction (in units of T_9).

qvary (7) REAL [local to RUN] In multiple runs option. Array set equal to c, cosmo and xi.

r (nrec) REAL [/rates/] Reverse reaction rates (worked out in SOL).

reacpr (nrec,8) REAL [/recpr0/] Reaction parameters which include reaction number, reaction type, 4 nuclide numbers, reverse reaction coefficient and the energy of the reaction (values passed onto iform, ii, jj, kk, ll, rev and q9).

rev (nrec) REAL [/recpr/] Reverse reaction coefficient for use in computing the reverse reaction rate.

rhob REAL [/endens/] Baryon energy density as given by equation D.5.

rhob0 REAL [/endens/] Initial baryon energy density.

rhone0 REAL [/endens/] Initial electron neutrino energy density.

rhonu REAL [/nupar/] Neutrino energy density as computed by subroutine NUDENS.

ri,rj,rk,rl REAL [local to SOL] Equated to arrays si, sj, sk and sl to hold the numbers N_i, N_j, N_k and N_l for equation E.7.

rnb REAL [/endens/] The ratio of the current baryon energy density to the initial value. This ratio to the 4/3 power gives the ratio of the current neutrino energy density to its initial value.

rnum1 (3) REAL [local to RUN] In multiple runs option. Parameter starting value.

rnum2 (3) REAL [local to RUN] In multiple runs option. Parameter end value.

rnum3 (3) REAL [local to RUN] In multiple runs option. Parameter increment.

rnumb1 REAL [local to RUN] In multiple runs option. Parameter value (outer loop).

rnumb2 REAL [local to RUN] In multiple runs option. Parameter value (middle loop).

rnumb3 REAL [local to RUN] In multiple runs option. Parameter value (inner loop).

si,sj,sk,sl (11) REAL [local to SOL] The eleven types of reactions are encoded in si, sj, sk and sl which give the number of nuclides involved in the reaction (see the DATA statements in subroutine SOL) and these values are passed onto ri, rj, rk and rl.

sum DOUBLE PRECISION [local to EQSLIN] Sum for backsubstitution as shown in equation E.11.

sumdy REAL [local to DERIVS] $\sum_i dY_i/dt$ as used in equation D.33.

summdy REAL [local to DERIVS] $\sum_i(\Delta M_i/M_u)dY_i/dt$ as used in equation D.33.

sumy REAL [local to DERIVS] $\sum_i Y_i$ as used in equations D.31 and D.32.

sumzdy REAL [local to DERIVS] $\sum_i Z_i dY_i/dt = dS/dt$ as used in equations D.20, D.24 (in $\partial M/\partial r$), D.25 (in $\partial M/\partial S$) and D.34.

sumzy REAL [local to DERIVS] $\sum_i Z_i Y_i = S$.

t REAL [/time/] Time.

t9 REAL [/evolp1/] Temperature of photons T_9 (units of 10^9 K).

t9f REAL [/compr/] Final (termination) temperature (in 10^9 K).

t9f0 REAL [/compr0/] Default value for t9f. Value of 10^{-2} stored in BLOCK DATA.

t9i REAL [/compr/] Initial (starting) temperature (in 10^9 K).

t9i0 REAL [/compr0/] Default value for t9i. Value of 10^2 stored in BLOCK DATA.

t9mev REAL [/nupar/] Temperature of photons (in units of MeV).

t9out (itmax) REAL [/outdat/] The output accumulation buffer for the temperature (in units of 10^9 K).

tau REAL [/modpr/] Neutron lifetime (sec).

thm (14) REAL [/therm/] Thermodynamic variables (energy densities, energy density derivatives, pressures). See discussion from D.7 to D.16.

thmout (itmax,6) REAL [/outdat/] The output accumulation buffer for energy densities and ϕ_e.

tnmev REAL [/nupar/] Neutrino temperature (in units of MeV).

tnu REAL [/nupar/] Neutrino temperature (in units of 10^9).

tout (itmax) REAL [/outdat/] The output accumulation buffer for the time.

v (nvar) REAL [local to DRIVER] Variables to be time evolved (T_9, h, ϕ_e and Y_i).

v0 (nvar) REAL [local to DRIVER] Value of variables at original point (see discussion following equation C.2).

vtype (8) CHARACTER [local to RUN] In multiple runs option. Label shown to user for quantities being varied by subroutine RUN.

w (2) REAL [local to RATE1] Upper limit for exponentials in the n → p rate.

x (2) REAL [local to RATE1] Upper limit for exponentials in the n → p rate.

x (nnuc) DOUBLE PRECISION [local to EQSLIN] Right-hand vector seen in equation E.8 and E.9.

xdy REAL [local to EQSLIN] The first time around, this is the right-hand value over the solution value, a ratio close to one and certainly greater than eps. This is a trick to get EQSLIN to go further and do the iterative improvement of the gaussian elimination solution. Thus, the second time around, this becomes $\delta\bar{Y}/\bar{Y}$ which can be properly compared to eps.

xi (3) REAL [/modpr/] Neutrino degeneracy parameters. xi(1) is ν_e degeneracy parameter ξ_e; xi(2) is the ν_μ neutrino degeneracy parameter ξ_μ; xi(3) is the ν_τ neutrino degeneracy parameter ξ_τ.

xi0 (3) REAL [/modpr0/] Default neutrino degeneracy parameters. Values of 0 are stored in BLOCK DATA.

xnu REAL [/modpr/] Number of neutrino species.

xout (itmax,nnuc2) REAL [/outdat/] The output accumulation buffer for nuclide mass fractions (H and ^{4}He) and ratios of number densities.

y (2) REAL [local to RATE1] Upper limit for exponentials in the p → n rate.

y (nnuc) REAL [/evolp1/] Number densities of nuclides Y_i defined by X_i/A_i, X_i the mass fraction contained in nuclei having atomic weight A_i.

y0 (nnuc) REAL [/evolp3/] Abundances of nuclides Y_i at end of 1st R-K loop (see discussion following equation C.2).

ytmin REAL [/compr/] Smallest abundances allowed. See discussion following equation C.3.

ytmin0 REAL [/compr0/] Default **ytmin**. Value of 10^{-25} stored in BLOCK DATA.

yx (nnuc) REAL [/lncoef/] The solution vector $\bar{Y}_i$ seen in equations E.8 - E.11. In subroutine **EQSLIN**, **yx** is just called **y**, not to be confused with **y** in common **/evolp1/**.

yy (nnuc) REAL [local to SOL] The solution vector $\bar{Y}_i$ seen in equations E.8 - E.11. The same as **yx** but with the nuclides in the original order (reverse of the order for **yx**).

z (2) REAL [local to RATE1] Upper limit for exponentials in the p → n rate.

z REAL [local to START and THERM] Defined by $z = m_e c^2/kT_9$.

zm (nnuc) REAL [/nucdat/] The nuclide charge. Values stored in BLOCK DATA.

References

Abramowitz, M., and Stegun, I.A., 1964, *Handbook of Mathematical Functions*, Applied Mathmatics Series, vol. 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York).

Ajzenberg-Selove, F., 1986, Nucl. Phys. **A449**, 1.

Ajzenberg-Selove, F., 1988, Nucl. Phys. **A490**, 1.

Ajzenberg-Selove, F., 1990, Nucl. Phys. **A506**, 1.

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

Berstein, J., Brown, L.S., and Feinberg, G., 1989, Rev. Mod. Phys., **91**, 25.

Bianconi, A., Lee, H.W., Ruffini, R., 1991, Astron. & Astrophys. **241**, 343.

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

Caughlan, G.R., and Fowler, W.A., 1988, Atomic Data and Nuclear Data Tables, **40**, 291.

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.

Dicus, D.A., Kolb, E.W., Gleeson, A.M., Sudarshan, E.C.G., Teplitz, V.L., and Turner, M.S., 1982, Phys Rev. **D26**, 2694.

Esmailzadeh, R., Starkman, G.D., and Dimopoulos, S., 1991, Ap. J, **378**, 504.

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

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

Fowler, W.A., and Hoyle, F., 1964, Ap. J. Suppl., No. 91, **9**, 201.

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

Kang, H.-S., 1991, Ohio State preprint OSU-TA-8/91, to appear in the *Proceedings of the Second Winter School of Physics*, University of Puerto Rico, Puerto, Rico, 27 March-5 April, 1991.

Kang, H.-S., and Steigman, G., 1991, Ohio State preprint OSU-TA-7/91, submitted to Nucl. Phys. B.

Kawano, L.H., 1988, Fermilab preprint FERMILAB-PUB-88/34-A.

Kolb, E.W., and Turner, M.S., 1990, *The Early Universe* (New York: Addison-Wesley).

Krauss, L.M., and Romanelli, P., 1990, Ap. J. **358**, 47.

Madsen, J., 1990, Phys. Rev. **D41**, 2472.

Malaney, R.A., and Mathews, G., 1991, Physics Reports, submitted.

Olive, K.A., Schramm, D.N., Steigman, G., and Walker, T.P., 1990, Phys. Lett. **B236**, 454.

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.

Smith, M., Kawano, L.H., and Malaney, R.A., 1992, submitted to Ap. J.

Tilly, D.R., Weller, H.R., and Hasan, H.H., 1987, Nucl. Phys. **A474**, 1.
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.
Walker, T.P., Steigman, G., Schramm, D.N., Olive, K.A., and Kang, H.-S., 1991, Ap. J. **376**, 51.
Weinberg, S., 1972, *Gravitation and Cosmology* (New York: John Wiley & Sons).

If you wish to obtain a copy of the program, please contact either Michael Smith at Caltech or Rocky Kolb at Fermilab:

Michael Smith
W. K. Kellogg Radiation Laboratory
Mail Code 106-38
California Institute of Technology
1201 California Blvd.
Pasadena, California 91125
MSmith@Erin.Caltech.EDU

Rocky Kolb
NASA/Fermilab Astrophysics Center
Mail Station 209
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510-0500
Rocky@FNAL.Bitnet

Table 1.

REACTIONS AND SOURCES FOR REACTION RATES

Reaction	Number	Literature Source
1. $n \leftrightarrow p$	01	Smith, Kawano and Malaney 1992
2. $H(n,\gamma)^2H$	12	Smith, Kawano and Malaney 1992
3. $^2H(n,\gamma)^3H$	13	Wagoner 1969
4. $^2H(p,\gamma)^3He$	20	Smith, Kawano and Malaney 1992
5. $^2H(d,n)^3He$	28	Smith, Kawano and Malaney 1992
6. $^2H(d,p)^3H$	29	Smith, Kawano and Malaney 1992
7. $^2H(\alpha,\gamma)^6Li$	25	Caughlan and Fowler 1988
8. $^3H \rightarrow e^- + \nu + {}^3He$	02	Tilly, Weller and Hasan 1987
9. $^3H(p,\gamma)^4He$	21	Caughlan and Fowler 1988
10. $^3H(d,n)^4He$	30	Smith, Kawano and Malaney 1992
11. $^3H(\alpha,\gamma)^7Li$	26	Smith, Kawano and Malaney 1992
12. $^3He(n,\gamma)^4He$	14	Wagoner 1969
13. $^3He(n,p)^3H$	16	Smith, Kawano and Malaney 1992
14. $^3He(d,p)^4He$	31	Caughlan and Fowler 1988
15. $^3He(^3He,2p)^4He$	32	Caughlan and Fowler 1988
16. $^3He(\alpha,\gamma)^7Be$	27	Smith, Kawano and Malaney 1992
17. $^4He(\alpha n,\gamma)^9Be$	58	Caughlan and Fowler 1988
18. $^4He(2\alpha,\gamma)^{12}C$	59	Caughlan and Fowler 1988
19. $^6Li(n,\gamma)^7Li$	15	Malaney and Fowler 1989
20. $^6Li(n,\alpha)^3H$	18	Caughlan and Fowler 1988
21. $^6Li(p,\gamma)^7Be$	22	Caughlan and Fowler 1988
22. $^6Li(p,\alpha)^3He$	23	Caughlan and Fowler 1988
23. $^6Li(\alpha,\gamma)^{10}B$	49	Caughlan and Fowler 1988
24. $^7Li(n,\gamma)^8Li$	35	Wagoner 1969
25. $^7Li(p,\alpha)^4He$	24	Smith, Kawano and Malaney 1992
26. $^7Li(d,n\alpha)^4He$	33	Caughlan and Fowler 1988
27. $^7Li(\alpha,\gamma)^{11}B$	50	Caughlan and Fowler 1988
28. $^8Li \rightarrow e^- + \nu + 2{}^4He$	03	Ajzenberg-Selove 1988
29. $^8Li(p,n\alpha)^4He$	60	original Wagoner code
30. $^8Li(\alpha,n)^{11}B$	53	Malaney and Fowler 1989

Table 1 – Continued.

Reaction	Number	Literature Source
31. ^{7}Be(n,p)^{7}Li	17	Smith, Kawano and Malaney 1992
32. ^{7}Be(n,α)^{4}He	19	Wagoner 1969
33. ^{7}Be(p,γ)^{8}B	40	Caughlan and Fowler 1988
34. ^{7}Be(d,pα)^{4}He	34	Caughlan and Fowler 1988
35. ^{7}Be(α,γ)^{11}C	51	Caughlan and Fowler 1988
36. ^{9}Be(p,γ)^{10}B	41	Caughlan and Fowler 1988
37. ^{9}Be(p,α)^{6}Li	46	Caughlan and Fowler 1988
38. ^{9}Be(p,dα)^{4}He	62	Caughlan and Fowler 1988
39. ^{9}Be(d,n)^{10}B	55	original Wagoner code
40. ^{9}Be(α,n)^{12}C	54	Caughlan and Fowler 1988
41. ^{8}B $\rightarrow$ e^{+} + ν + 2^{4}He	06	Ajzenberg-Selove 1988
42. ^{8}B(n,pα)^{4}He	61	original Wagoner code
43. ^{8}B(α,p)^{11}C	52	Wagoner 1969
44. ^{10}B(n,γ)^{11}B	36	Wagoner 1969
45. ^{10}B(n,α)^{7}Li	39	Caughlan and Fowler 1988
46. ^{10}B(p,γ)^{11}C	42	Caughlan and Fowler 1988
47. ^{10}B(p,α)^{7}Be	47	Caughlan and Fowler 1988
48. ^{10}B(d,p)^{11}B	56	original Wagoner code
49. ^{10}B(α,n)^{13}N	85	Caughlan and Fowler 1988
50. ^{10}B(α,p)^{13}C	80	Wagoner 1969
51. ^{11}B(n,γ)^{12}B	37	Malaney and Fowler 1989
52. ^{11}B(p,γ)^{12}C	43	Caughlan and Fowler 1988
53. ^{11}B(p,2α)^{4}He	63	Caughlan and Fowler 1988
54. ^{11}B(d,n)^{12}C	57	original Wagoner code
55. ^{11}B(α,n)^{14}N	86	Caughlan and Fowler 1988
56. ^{11}B(α,p)^{14}C	81	Caughlan and Fowler 1988
57. ^{12}B $\rightarrow$ e^{-} + ν + ^{12}C	04	Ajzenberg-Selove 1990
58. ^{12}B(p,n)^{12}C	45	Wagoner 1969
59. ^{12}B(p,α)^{9}Be	48	Wagoner 1969
60. ^{12}B(α,n)^{15}N	87	Wagoner 1969

Table 1 – Continued.

Reaction	Number	Literature Source
61. $^{11}C \rightarrow e^{+} + \nu + {}^{11}B$	07	Ajzenberg-Selove 1990
62. $^{11}C(n,p)^{11}B$	38	Caughlan and Fowler 1988
63. $^{11}C(n,2\alpha)^{4}He$	64	Wagoner 1969
64. $^{11}C(p,\gamma)^{12}N$	44	Caughlan and Fowler 1988
65. $^{11}C(\alpha,p)^{14}N$	82	Caughlan and Fowler 1988
66. $^{12}C(n,\gamma)^{13}C$	65	Wagoner 1969
67. $^{12}C(p,\gamma)^{13}N$	72	Caughlan and Fowler 1988
68. $^{12}C(\alpha,\gamma)^{16}O$	79	Caughlan and Fowler 1988
69. $^{13}C(n,\gamma)^{14}C$	66	Wagoner 1969
70. $^{13}C(p,\gamma)^{14}N$	73	Caughlan and Fowler 1988
71. $^{13}C(\alpha,n)^{16}O$	88	Caughlan and Fowler 1988
72. $^{14}C \rightarrow e^{-} + \nu + {}^{14}N$	05	Ajzenberg-Selove 1986
73. $^{14}C(p,\gamma)^{15}N$	74	Caughlan and Fowler 1988
74. $^{12}N \rightarrow e^{+} + \nu + {}^{12}C$	08	Ajzenberg-Selove 1990
75. $^{12}N(\alpha,p)^{15}O$	83	Caughlan and Fowler 1988
76. $^{13}N \rightarrow e^{+} + \nu + {}^{13}C$	09	Ajzenberg-Selove 1986
77. $^{13}N(n,p)^{13}C$	68	Caughlan and Fowler 1988
78. $^{13}N(p,\gamma)^{14}O$	75	Caughlan and Fowler 1988
79. $^{13}N(\alpha,p)^{16}O$	84	Caughlan and Fowler 1988
80. $^{14}N(n,\gamma)^{15}N$	67	Wagoner 1969
81. $^{14}N(n,p)^{14}C$	69	Caughlan and Fowler 1988
82. $^{14}N(p,\gamma)^{15}O$	76	Caughlan and Fowler 1988
83. $^{15}N(p,\gamma)^{16}O$	77	Caughlan and Fowler 1988
84. $^{15}N(p,\alpha)^{12}C$	78	Caughlan and Fowler 1988
85. $^{14}O \rightarrow e^{+} + \nu + {}^{14}N$	10	Ajzenberg-Selove 1986
86. $^{15}O \rightarrow e^{+} + \nu + {}^{15}N$	11	Ajzenberg-Selove 1986
87. $^{15}O(n,p)^{15}N$	70	Caughlan and Fowler 1988
88. $^{15}O(n,\alpha)^{12}C$	71	Caughlan and Fowler 1988

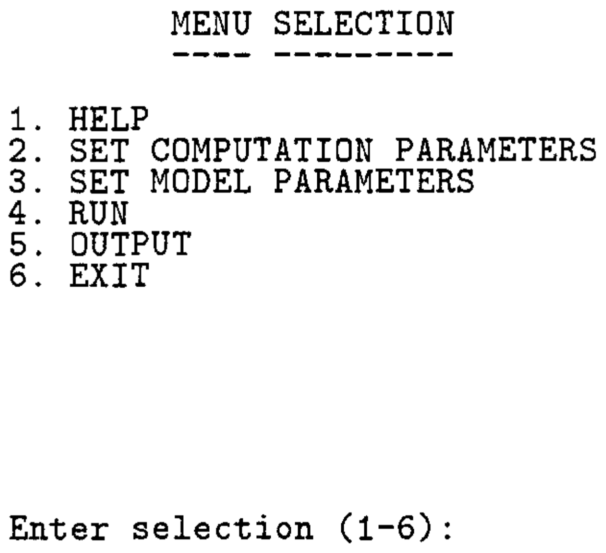
Table 2.

DEDUCED RATES AND UNCERTAINTIES OF 12 MAJOR BBN REACTIONS

	Reaction	Nuclear Reaction Rate ($\mathrm{cm^3 s^{-1} mole^{-1}}$)	1σ Uncertainty
1.	neutron τ	888.5 seconds	3.7 sec
2.	p(n,γ)d	4.742×10^4 $\times(1. - .850T_9^{1/2}+.490T_9-.0962T_9^{3/2}$ $+8.47\times 10^{-3}T_9^2-2.80\times 10^{-4}T_9^{5/2})$	7%
3.	d(p,γ)^{3}He	$2.65\times 10^3 T_9^{-2/3}\exp(-3.720/T_9^{1/3})$ $\times(1.+.112T_9^{1/3}+1.99T_9^{2/3}+1.56T_9+.162T_9^{4/3}+.324T_9^{5/3})$	10
4.	d(d,n)^{3}He	$3.95\times 10^8 T_9^{-2/3}\exp(-4.259/T_9^{1/3})$ $\times(1.+.098T_9^{1/3}+.765T_9^{2/3}+.525T_9+9.61\times 10^{-3}T_9^{4/3}+.0167T_9^{5/3})$	10
5.	d(d,p)t	$4.17\times 10^8 T_9^{-2/3}\exp(-4.258/T_9^{1/3})$ $\times(1.+.098T_9^{1/3}+.518T_9^{2/3}+.355T_9-.010T_9^{4/3}-.018T_9^{5/3})$	10
6.	t(d,n)^{4}He	$1.063\times 10^{11} T_9^{-2/3}\exp[-4.559/T_9^{1/3}-(T_9/.0754)^2]$ $\times(1.+.092T_9^{1/3}-.375T_9^{2/3}-.242T_9+33.82T_9^{4/3}+55.42T_9^{5/3})$ $+8.047\times 10^8 T_9^{-2/3}\exp(-0.4857/T_9)$	8
7.	t(α,γ)^{7}Li	$3.032\times 10^5 T_9^{-2/3}\exp(-8.090/T_9^{1/3})$ $\times(1.+.0516T_9^{1/3}+.0229T_9^{2/3}+8.28\times 10^{-3}T_9$ $-3.28\times 10^{-4}T_9^{4/3}-3.01\times 10^{-4}T_9^{5/3})$ $+5.109\times 10^5 T_{9a}^{5/6}T_9^{-3/2}\exp(-8.068/T_{9a}^{1/3})$ $T_{9a}=T_9/(1.+.1378T_9)$	$T_9>10$: 8.1 $T_9\leq 10$: $29.-5.9T_{9b}^{1/2}-7.2T_{9b}+4.0T_{9b}^{3/2}-.56T_{9b}^2$ $T_{9b}=T_9+.0419$
8.	^{3}He(n,p)t	$7.21\times 10^8(1.-.508T_9^{1/2}+.228T_9)$	10
9.	^{3}He(d,p)^{4}He	$5.021\times 10^{10} T_9^{-2/3}\exp[-7.144/T_9^{1/3}-(T_9/.270)^2]$ $(1.+.058T_9^{1/3}+.603T_9^{2/3}+.245T_9+6.97T_9^{4/3}+7.19T_9^{5/3})$ $+5.212\times 10^8 T_9^{-1/2}\exp(-1.762/T_9)$	8
10.	^{3}He(α,γ)^{7}Be	$4.817\times 10^6 T_9^{-2/3}\exp(-14.964/T_9^{1/3})$ $\times(1.+.0325T_9^{1/3}-1.04\times 10^{-3}T_9^{2/3}-2.37\times 10^{-4}T_9$ $-8.11\times 10^{-5}T_9^{4/3}-4.69\times 10^{-5}T_9^{5/3})$ $+5.938\times 10^6 T_{9a}^{5/6}T_9^{-3/2}\exp(-12.859/T_{9a}^{1/3})$ $T_{9a}=T_9/(1.+.1071T_9)$	$T_9>10$: 9.7 $T_9\leq 10$: $27.-15.T_{9b}^{1/2}+4.0T_{9b}-.25T_{9b}^{3/2}-.02T_{9b}^2$ $T_{9b}=T_9+.783$
11.	^{7}Li(p,α)^{4}He	$1.096\times 10^9 T_9^{-2/3}\exp(-8.472/T_9^{1/3})$ $-4.830\times 10^8 T_{9a}^{5/6}T_9^{-3/2}\exp(-8.472/T_{9a}^{1/3})$ $+1.06\times 10^{10} T_9^{-3/2}\exp(-30.442/T_9)$ $+1.56\times 10^5 T_9^{-2/3}\exp[-8.472/T_9^{1/3}-(T_9/1.696)^2]$ $\times(1.+.049T_9^{1/3}-2.50T_9^{2/3}+.860T_9+3.52T_9^{4/3}+3.08T_9^{5/3})$ $+1.55\times 10^6 T_9^{-3/2}\exp(-4.478/T_9)$ $T_{9a}=[T_9/(1.+.759T_9)]$	8
12.	^{7}Be(n,p)^{7}Li	2.675×10^9 $\times(1.-.560T_9^{1/2}+.179T_9-.0283T_9^{3/2}+2.21\times 10^{-3}T_9^2-6.85\times 10^{-5}T_9^{5/2})$ $+9.391\times 10^8 T_{9a}^{3/2}T_9^{-3/2}+4.467\times 10^7 T_9^{-3/2}\exp(-0.07486/T_9)$ $T_{9a}=[T_9/(1.+13.08T_9)]$	9

```
$ edt nuc123.for
find ’unit=1’
sub/unit=1/unit=4
sub/sys$command/batin.inc
copy .  to .+
sub/unit=4/unit=5/.-
sub/batin/batout
sub/old/new
find ’close (unit=1)’
sub/unit=1/unit=4
copy .  to .+
sub/unit=4/unit=5/.-
sub/ir=1/ir=4/whole
sub/iw=1/iw=5/whole
exit
$ rename nuc123.for nucbat.for
$ write sys$output "NUCBAT ready"
```

Figure 1. VAX/VMS command file for converting NUC123 into NUCBAT, a version for batch runs.



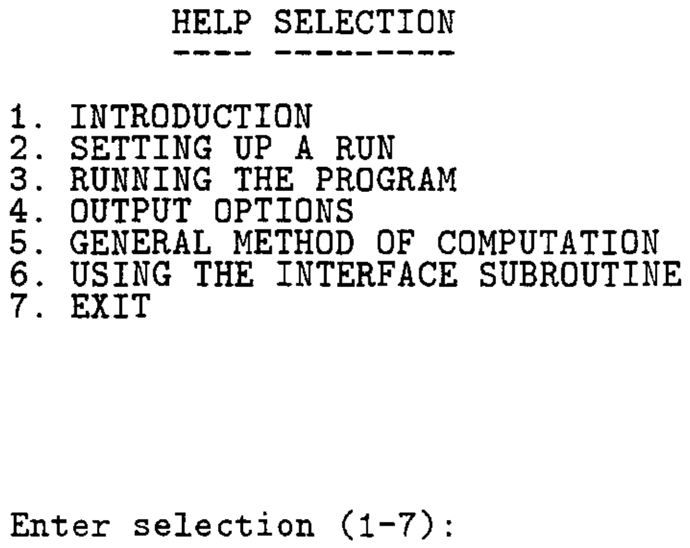
```
           MENU SELECTION
           ---- ---------

1. HELP
2. SET COMPUTATION PARAMETERS
3. SET MODEL PARAMETERS
4. RUN
5. OUTPUT
6. EXIT




Enter selection (1-6):
```

Figure 2. NUC123 main menu.



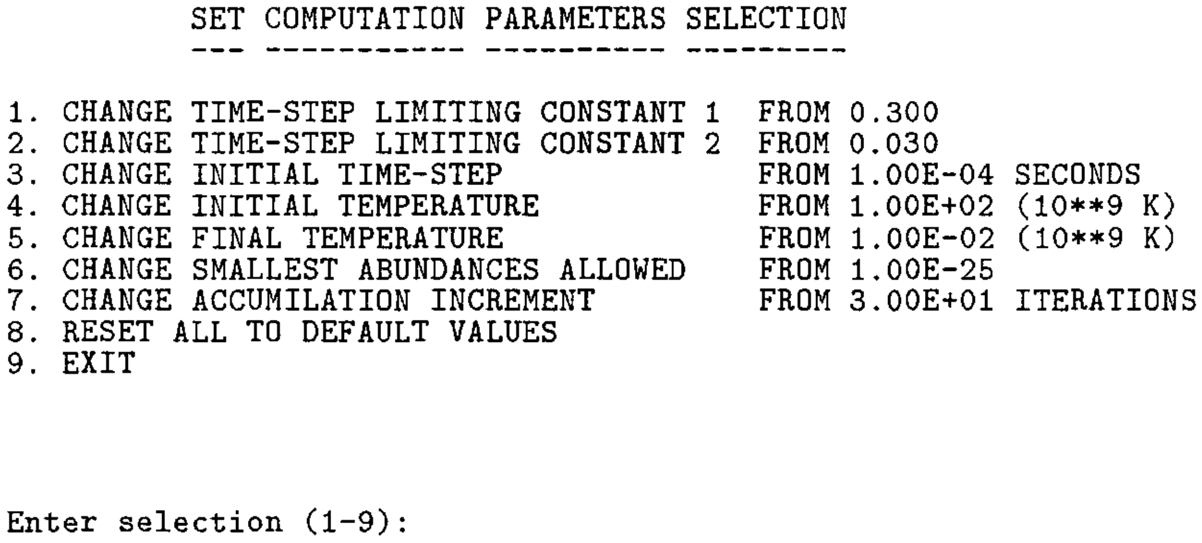
```
           HELP SELECTION
           ---- ---------

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




Enter selection (1-7):
```

Figure 3. HELP section submenu.

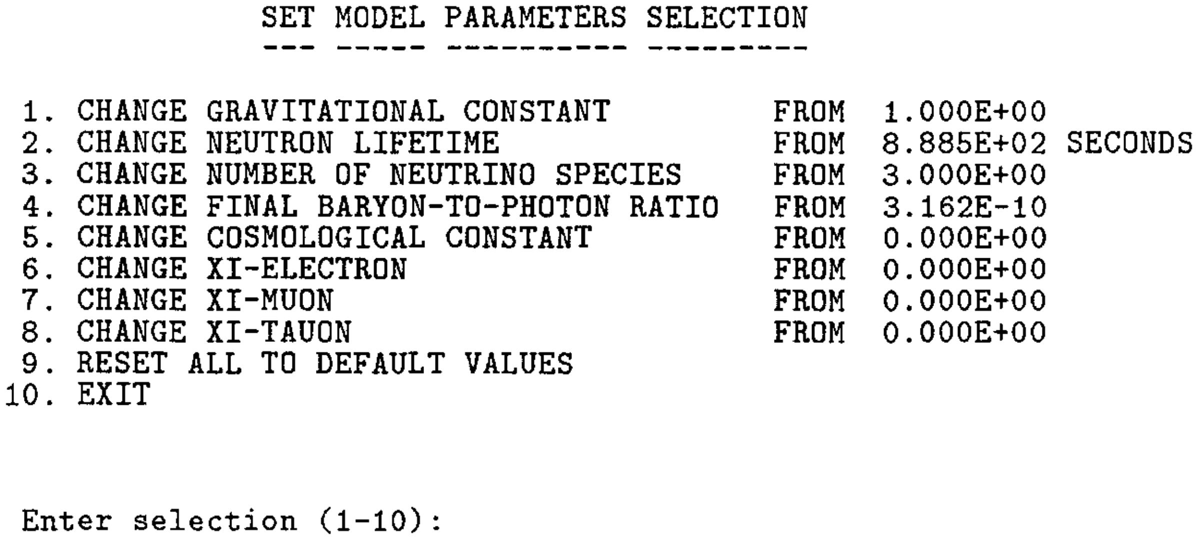


```
          SET COMPUTATION PARAMETERS SELECTION
          --- ----------- ---------- ---------

1. CHANGE TIME-STEP LIMITING CONSTANT 1  FROM 0.300
2. CHANGE TIME-STEP LIMITING CONSTANT 2  FROM 0.030
3. CHANGE INITIAL TIME-STEP              FROM 1.00E-04 SECONDS
4. CHANGE INITIAL TEMPERATURE            FROM 1.00E+02 (10**9 K)
5. CHANGE FINAL TEMPERATURE              FROM 1.00E-02 (10**9 K)
6. CHANGE SMALLEST ABUNDANCES ALLOWED    FROM 1.00E-25
7. CHANGE ACCUMILATION INCREMENT         FROM 3.00E+01 ITERATIONS
8. RESET ALL TO DEFAULT VALUES
9. EXIT


Enter selection (1-9):
```

Figure 4. SET COMPUTATIONAL PARAMETERS section submenu.

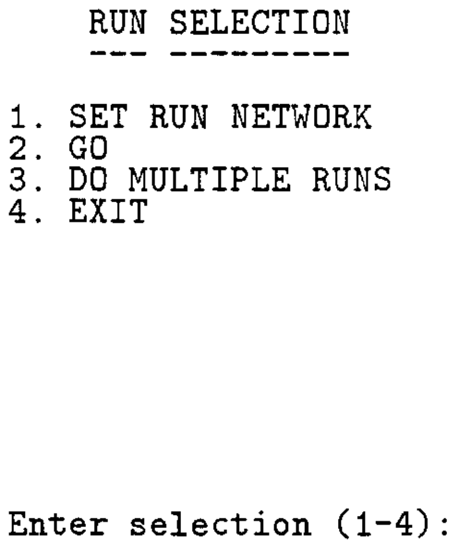


```
              SET MODEL PARAMETERS SELECTION
              --- ----- ---------- ---------

 1. CHANGE GRAVITATIONAL CONSTANT        FROM  1.000E+00
 2. CHANGE NEUTRON LIFETIME              FROM  8.885E+02 SECONDS
 3. CHANGE NUMBER OF NEUTRINO SPECIES    FROM  3.000E+00
 4. CHANGE FINAL BARYON-TO-PHOTON RATIO  FROM  3.162E-10
 5. CHANGE COSMOLOGICAL CONSTANT         FROM  0.000E+00
 6. CHANGE XI-ELECTRON                   FROM  0.000E+00
 7. CHANGE XI-MUON                       FROM  0.000E+00
 8. CHANGE XI-TAUON                      FROM  0.000E+00
 9. RESET ALL TO DEFAULT VALUES
10. EXIT


 Enter selection (1-10):
```

Figure 5. SET MODEL PARAMETERS section submenu.



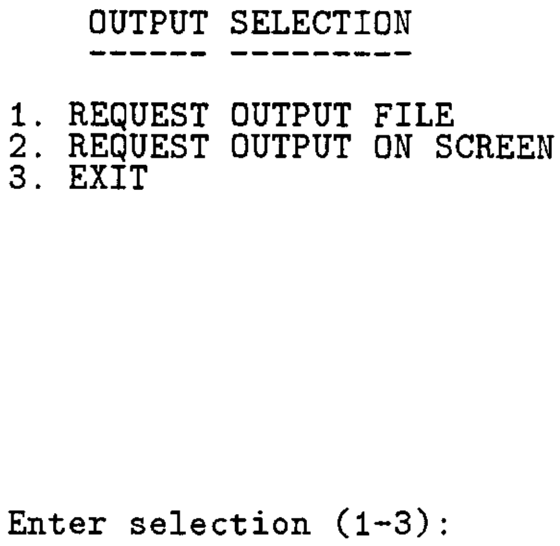
```
      RUN SELECTION
      --- ---------

  1. SET RUN NETWORK
  2. GO
  3. DO MULTIPLE RUNS
  4. EXIT




  Enter selection (1-4):
```

Figure 6. RUN section submenu



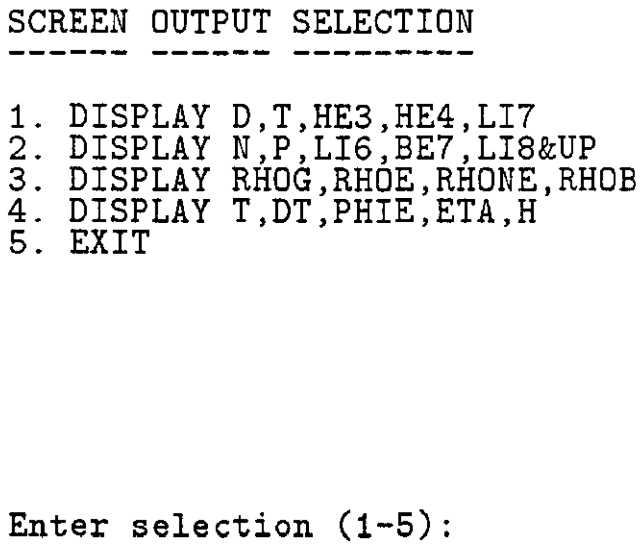
```
      OUTPUT SELECTION
      ------ ---------

  1. REQUEST OUTPUT FILE
  2. REQUEST OUTPUT ON SCREEN
  3. EXIT




  Enter selection (1-3):
```

Figure 7. OUTPUT section submenu.



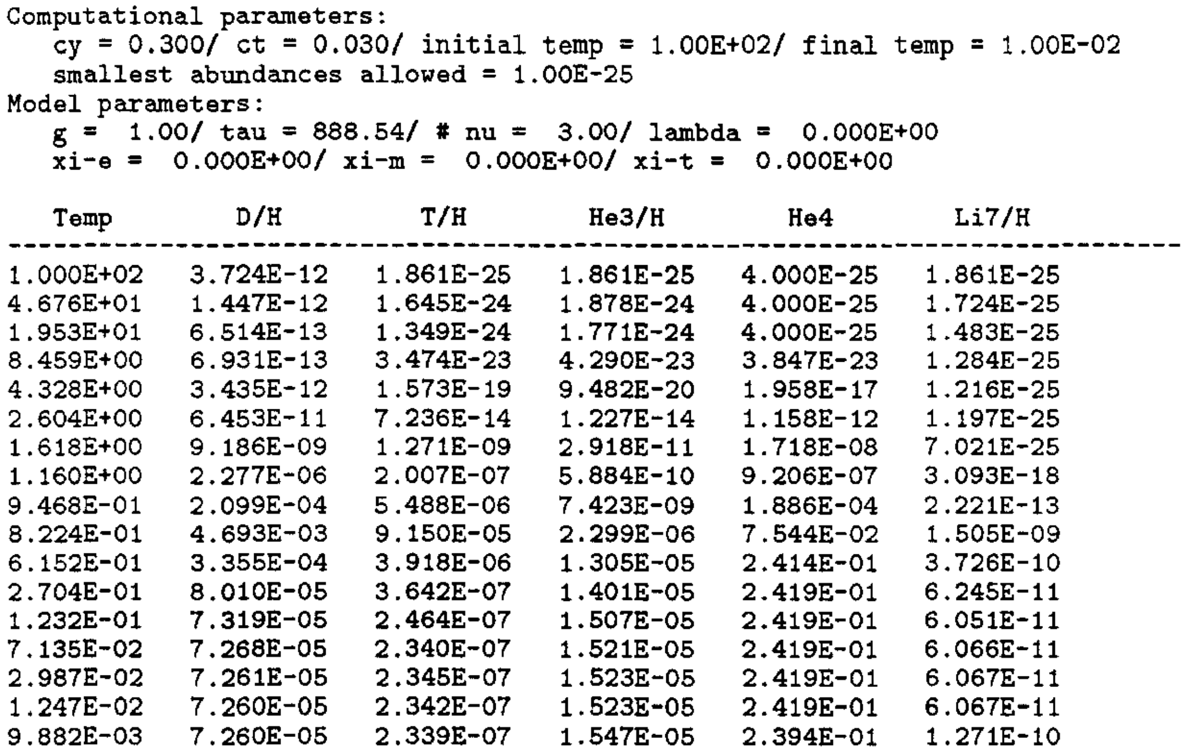
```
SCREEN OUTPUT SELECTION
------ ------ ---------

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):
```

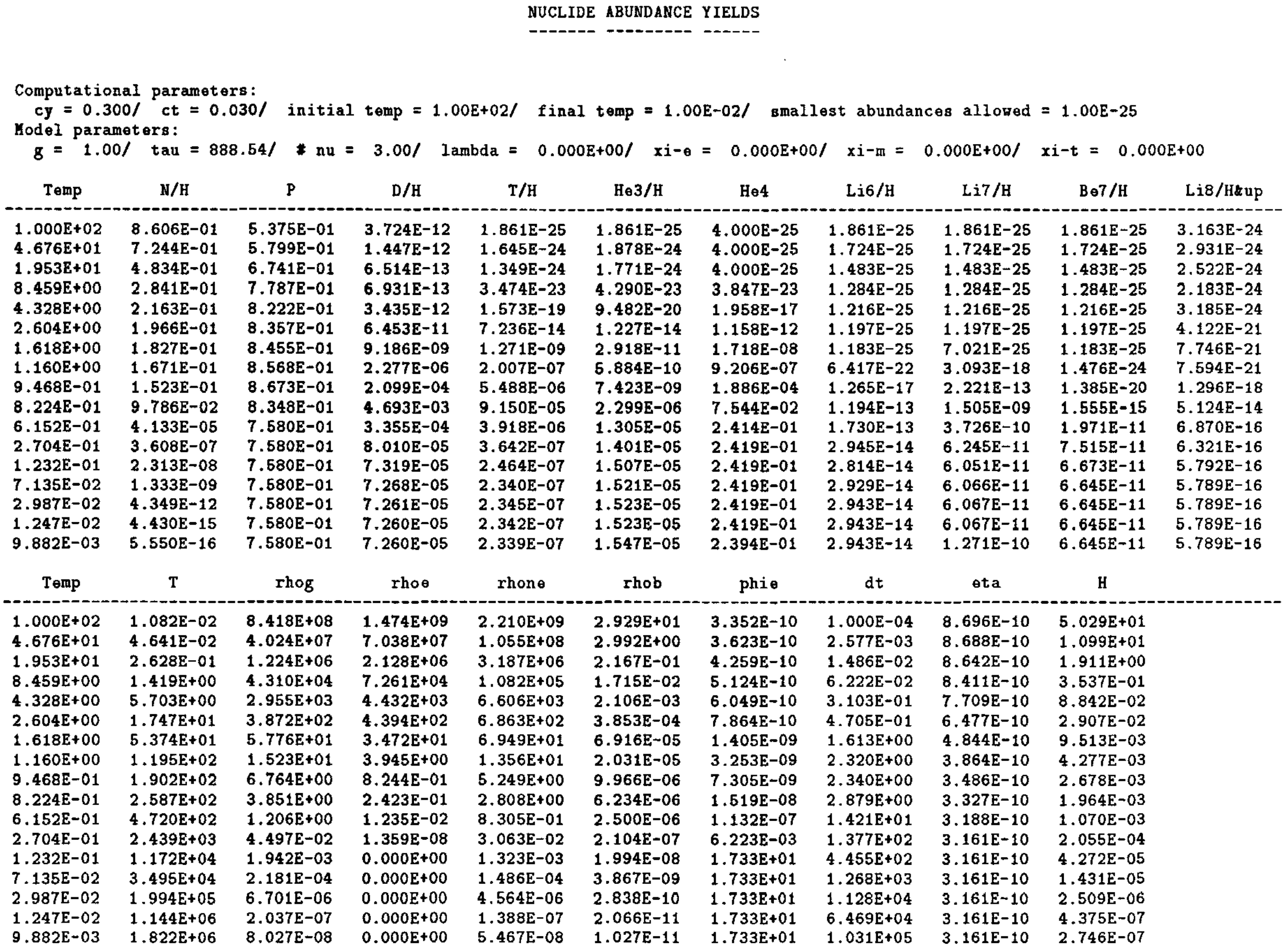
Figure 8. Screen output sub-submenu.



```
Computational parameters:
   cy = 0.300/ ct = 0.030/ initial temp = 1.00E+02/ final temp = 1.00E-02
   smallest abundances allowed = 1.00E-25
Model parameters:
   g =  1.00/ tau = 888.54/ # nu =  3.00/ lambda =  0.000E+00
   xi-e =  0.000E+00/ xi-m =  0.000E+00/ xi-t =  0.000E+00

   Temp        D/H         T/H         He3/H        He4         Li7/H
-----------------------------------------------------------------------------
1.000E+02   3.724E-12   1.861E-25   1.861E-25   4.000E-25   1.861E-25
4.676E+01   1.447E-12   1.645E-24   1.878E-24   4.000E-25   1.724E-25
1.953E+01   6.514E-13   1.349E-24   1.771E-24   4.000E-25   1.483E-25
8.459E+00   6.931E-13   3.474E-23   4.290E-23   3.847E-23   1.284E-25
4.328E+00   3.435E-12   1.573E-19   9.482E-20   1.958E-17   1.216E-25
2.604E+00   6.453E-11   7.236E-14   1.227E-14   1.158E-12   1.197E-25
1.618E+00   9.186E-09   1.271E-09   2.918E-11   1.718E-08   7.021E-25
1.160E+00   2.277E-06   2.007E-07   5.884E-10   9.206E-07   3.093E-18
9.468E-01   2.099E-04   5.488E-06   7.423E-09   1.886E-04   2.221E-13
8.224E-01   4.693E-03   9.150E-05   2.299E-06   7.544E-02   1.505E-09
6.152E-01   3.355E-04   3.918E-06   1.305E-05   2.414E-01   3.726E-10
2.704E-01   8.010E-05   3.642E-07   1.401E-05   2.419E-01   6.245E-11
1.232E-01   7.319E-05   2.464E-07   1.507E-05   2.419E-01   6.051E-11
7.135E-02   7.268E-05   2.340E-07   1.521E-05   2.419E-01   6.066E-11
2.987E-02   7.261E-05   2.345E-07   1.523E-05   2.419E-01   6.067E-11
1.247E-02   7.260E-05   2.342E-07   1.523E-05   2.419E-01   6.067E-11
9.882E-03   7.260E-05   2.339E-07   1.547E-05   2.394E-01   1.271E-10
```

Figure 9. Sample screen output display



```
                                                     NUCLIDE ABUNDANCE YIELDS
                                                     ------- --------- ------


 Computational parameters:
   cy = 0.300/  ct = 0.030/  initial temp = 1.00E+02/  final temp = 1.00E-02/  smallest abundances allowed = 1.00E-25
 Model parameters:
   g =  1.00/  tau = 888.54/  # nu =  3.00/  lambda =  0.000E+00/  xi-e =  0.000E+00/  xi-m =  0.000E+00/  xi-t =  0.000E+00

    Temp        N/H          P          D/H         T/H        He3/H        He4        Li6/H       Li7/H       Be7/H     Li8/H&up
-----------------------------------------------------------------------------------------------------------------------------------
 1.000E+02   8.606E-01   5.375E-01   3.724E-12   1.861E-25   1.861E-25   4.000E-25   1.861E-25   1.861E-25   1.861E-25   3.163E-24
 4.676E+01   7.244E-01   5.799E-01   1.447E-12   1.645E-24   1.878E-24   4.000E-25   1.724E-25   1.724E-25   1.724E-25   2.931E-24
 1.953E+01   4.834E-01   6.741E-01   6.514E-13   1.349E-24   1.771E-24   4.000E-25   1.483E-25   1.483E-25   1.483E-25   2.522E-24
 8.459E+00   2.841E-01   7.787E-01   6.931E-13   3.474E-23   4.290E-23   3.847E-23   1.284E-25   1.284E-25   1.284E-25   2.183E-24
 4.328E+00   2.163E-01   8.222E-01   3.435E-12   1.573E-19   9.482E-20   1.958E-17   1.216E-25   1.216E-25   1.216E-25   3.185E-24
 2.604E+00   1.966E-01   8.357E-01   6.453E-11   7.236E-14   1.227E-14   1.158E-12   1.197E-25   1.197E-25   1.197E-25   4.122E-21
 1.618E+00   1.827E-01   8.455E-01   9.186E-09   1.271E-09   2.918E-11   1.718E-08   1.183E-25   7.021E-25   1.183E-25   7.746E-21
 1.160E+00   1.671E-01   8.568E-01   2.277E-06   2.007E-07   5.884E-10   9.206E-07   6.417E-22   3.093E-18   1.476E-24   7.594E-21
 9.468E-01   1.523E-01   8.673E-01   2.099E-04   5.488E-06   7.423E-09   1.886E-04   1.265E-17   2.221E-13   1.385E-20   1.296E-18
 8.224E-01   9.786E-02   8.348E-01   4.693E-03   9.150E-05   2.299E-06   7.544E-02   1.194E-13   1.505E-09   1.555E-15   5.124E-14
 6.152E-01   4.133E-05   7.580E-01   3.355E-04   3.918E-06   1.305E-05   2.414E-01   1.730E-13   3.726E-10   1.971E-11   6.870E-16
 2.704E-01   3.608E-07   7.580E-01   8.010E-05   3.642E-07   1.401E-05   2.419E-01   2.945E-14   6.245E-11   7.515E-11   6.321E-16
 1.232E-01   2.313E-08   7.580E-01   7.319E-05   2.464E-07   1.507E-05   2.419E-01   2.814E-14   6.051E-11   6.673E-11   5.792E-16
 7.135E-02   1.333E-09   7.580E-01   7.268E-05   2.340E-07   1.521E-05   2.419E-01   2.929E-14   6.066E-11   6.645E-11   5.789E-16
 2.987E-02   4.349E-12   7.580E-01   7.261E-05   2.345E-07   1.523E-05   2.419E-01   2.943E-14   6.067E-11   6.645E-11   5.789E-16
 1.247E-02   4.430E-15   7.580E-01   7.260E-05   2.342E-07   1.523E-05   2.419E-01   2.943E-14   6.067E-11   6.645E-11   5.789E-16
 9.882E-03   5.550E-16   7.580E-01   7.260E-05   2.339E-07   1.547E-05   2.394E-01   2.943E-14   1.271E-10   6.645E-11   5.789E-16

    Temp         T          rhog        rhoe       rhone        rhob        phie         dt         eta          H
-----------------------------------------------------------------------------------------------------------------------------------
 1.000E+02   1.082E-02   8.418E+08   1.474E+09   2.210E+09   2.929E+01   3.352E-10   1.000E-04   8.696E-10   5.029E+01
 4.676E+01   4.641E-02   4.024E+07   7.038E+07   1.055E+08   2.992E+00   3.623E-10   2.577E-03   8.688E-10   1.099E+01
 1.953E+01   2.628E-01   1.224E+06   2.128E+06   3.187E+06   2.167E-01   4.259E-10   1.486E-02   8.642E-10   1.911E+00
 8.459E+00   1.419E+00   4.310E+04   7.261E+04   1.082E+05   1.715E-02   5.124E-10   6.222E-02   8.411E-10   3.537E-01
 4.328E+00   5.703E+00   2.955E+03   4.432E+03   6.606E+03   2.106E-03   6.049E-10   3.103E-01   7.709E-10   8.842E-02
 2.604E+00   1.747E+01   3.872E+02   4.394E+02   6.863E+02   3.853E-04   7.864E-10   4.705E-01   6.477E-10   2.907E-02
 1.618E+00   5.374E+01   5.776E+01   3.472E+01   6.949E+01   6.916E-05   1.405E-09   1.613E+00   4.844E-10   9.513E-03
 1.160E+00   1.195E+02   1.523E+01   3.945E+00   1.356E+01   2.031E-05   3.253E-09   2.320E+00   3.864E-10   4.277E-03
 9.468E-01   1.902E+02   6.764E+00   8.244E-01   5.249E+00   9.966E-06   7.305E-09   2.340E+00   3.486E-10   2.678E-03
 8.224E-01   2.587E+02   3.851E+00   2.423E-01   2.808E+00   6.234E-06   1.519E-08   2.879E+00   3.327E-10   1.964E-03
 6.152E-01   4.720E+02   1.206E+00   1.235E-02   8.305E-01   2.500E-06   1.132E-07   1.421E+01   3.188E-10   1.070E-03
 2.704E-01   2.439E+03   4.497E-02   1.359E-08   3.063E-02   2.104E-07   6.223E-03   1.377E+02   3.161E-10   2.055E-04
 1.232E-01   1.172E+04   1.942E-03   0.000E+00   1.323E-03   1.994E-08   1.733E+01   4.455E+02   3.161E-10   4.272E-05
 7.135E-02   3.495E+04   2.181E-04   0.000E+00   1.486E-04   3.867E-09   1.733E+01   1.268E+03   3.161E-10   1.431E-05
 2.987E-02   1.994E+05   6.701E-06   0.000E+00   4.564E-06   2.838E-10   1.733E+01   1.128E+04   3.161E-10   2.509E-06
 1.247E-02   1.144E+06   2.037E-07   0.000E+00   1.388E-07   2.066E-11   1.733E+01   6.469E+04   3.161E-10   4.375E-07
 9.882E-03   1.822E+06   8.027E-08   0.000E+00   5.467E-08   1.027E-11   1.733E+01   1.031E+05   3.161E-10   2.746E-07
```

Figure 10. Sample output file.

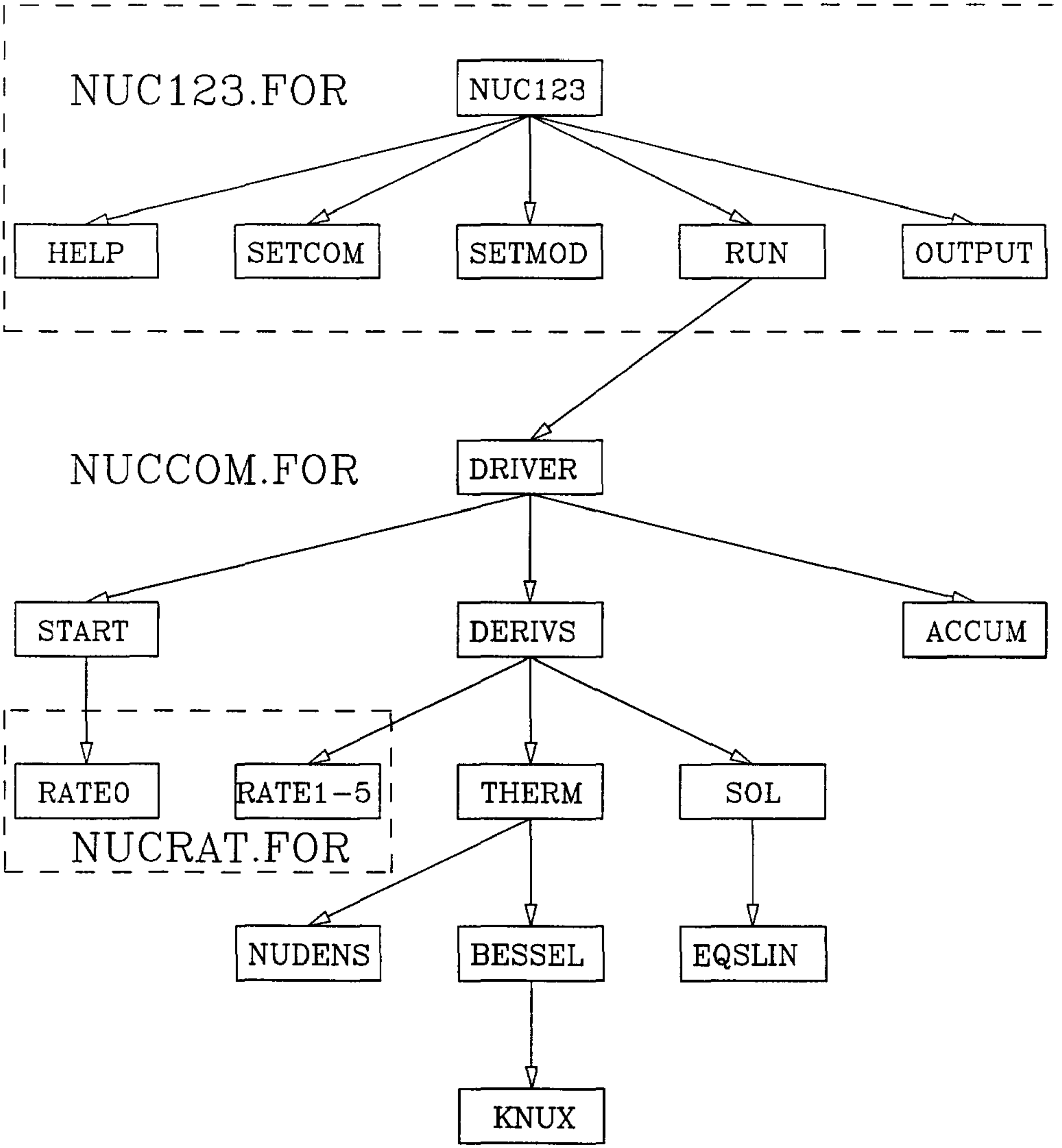


Figure 11. Subroutine hierarchy.

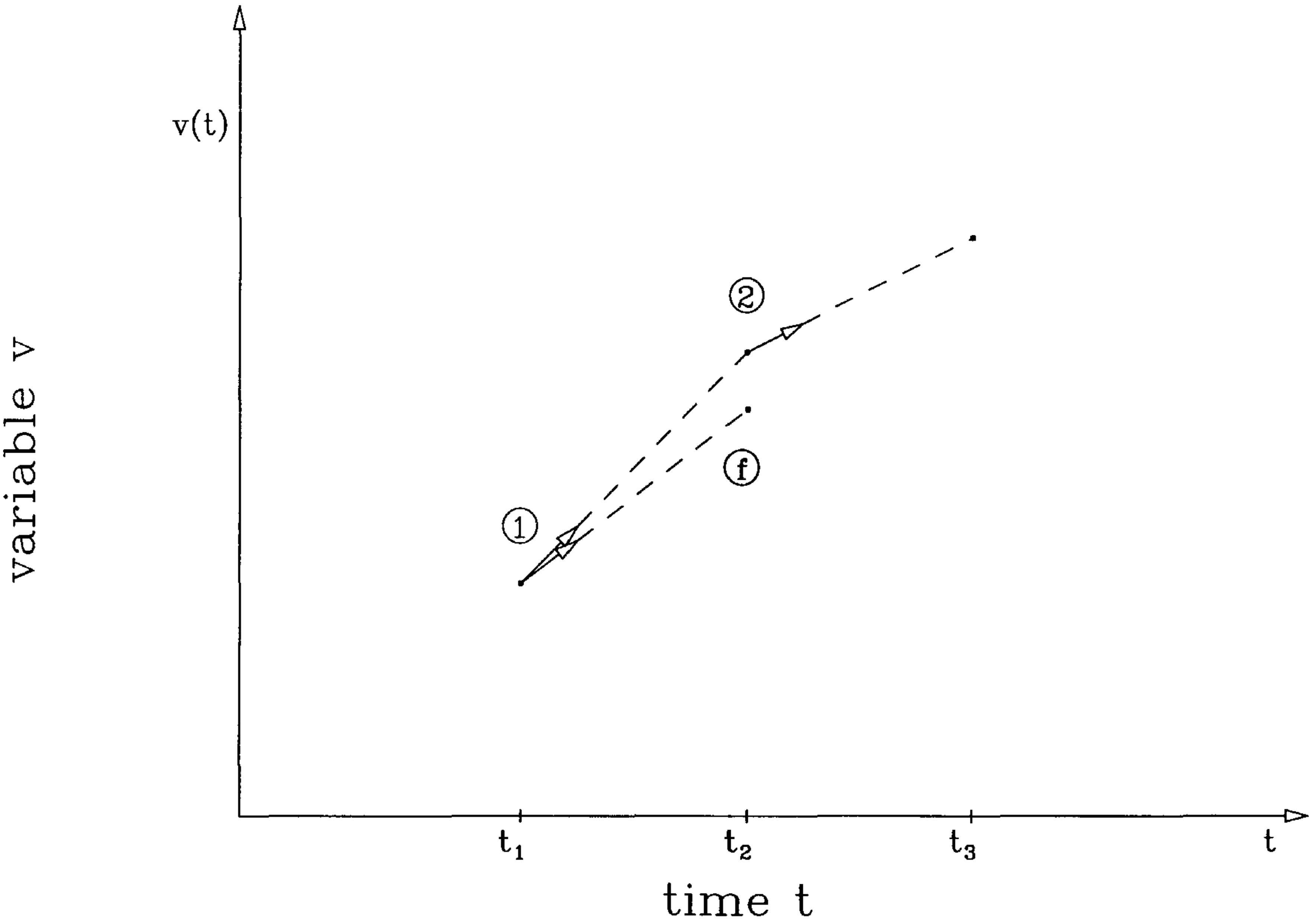


Figure 12. 2nd order Runge–Kutta procedure.

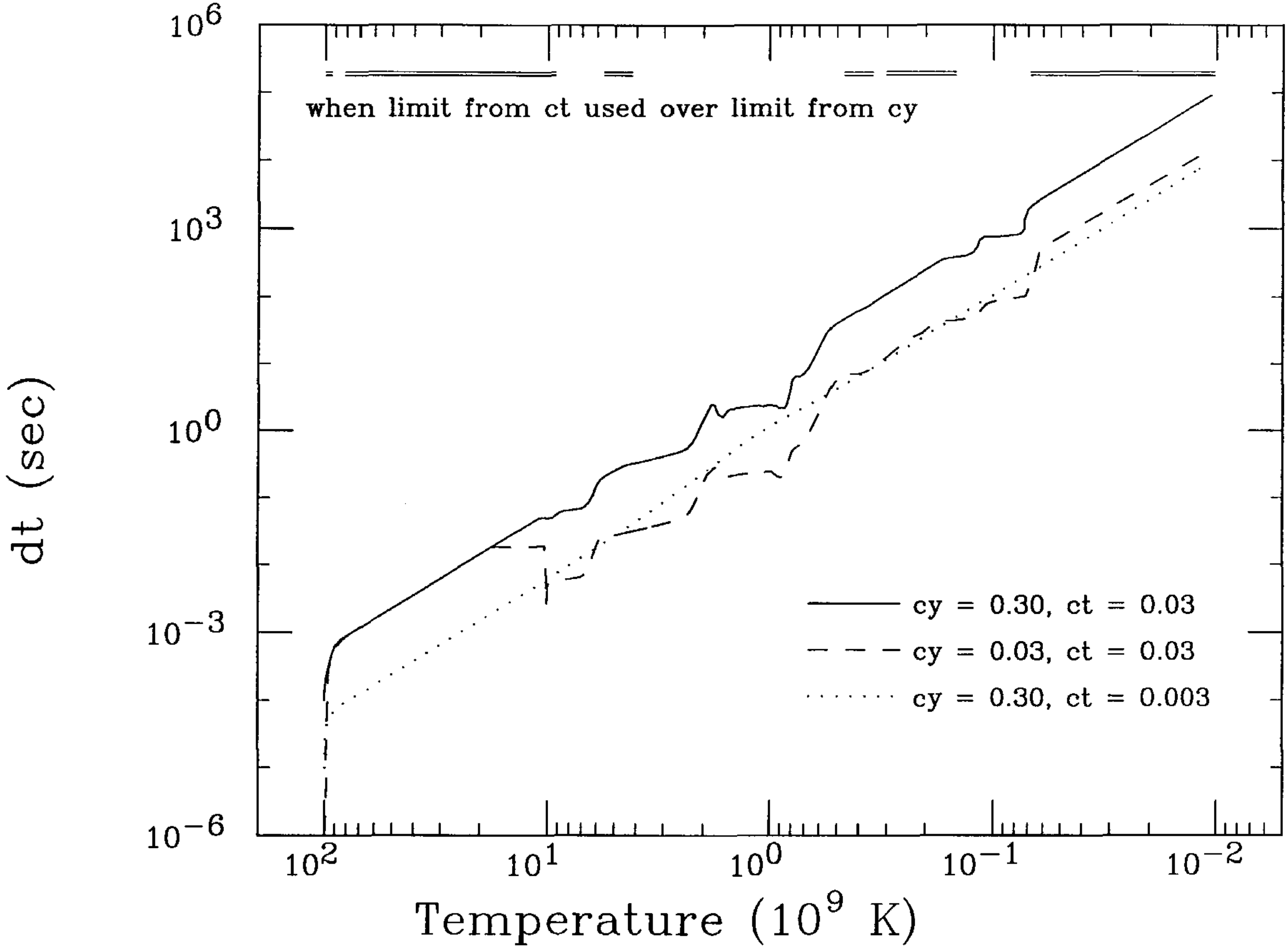


Figure 13. Time–step vs. temperature.

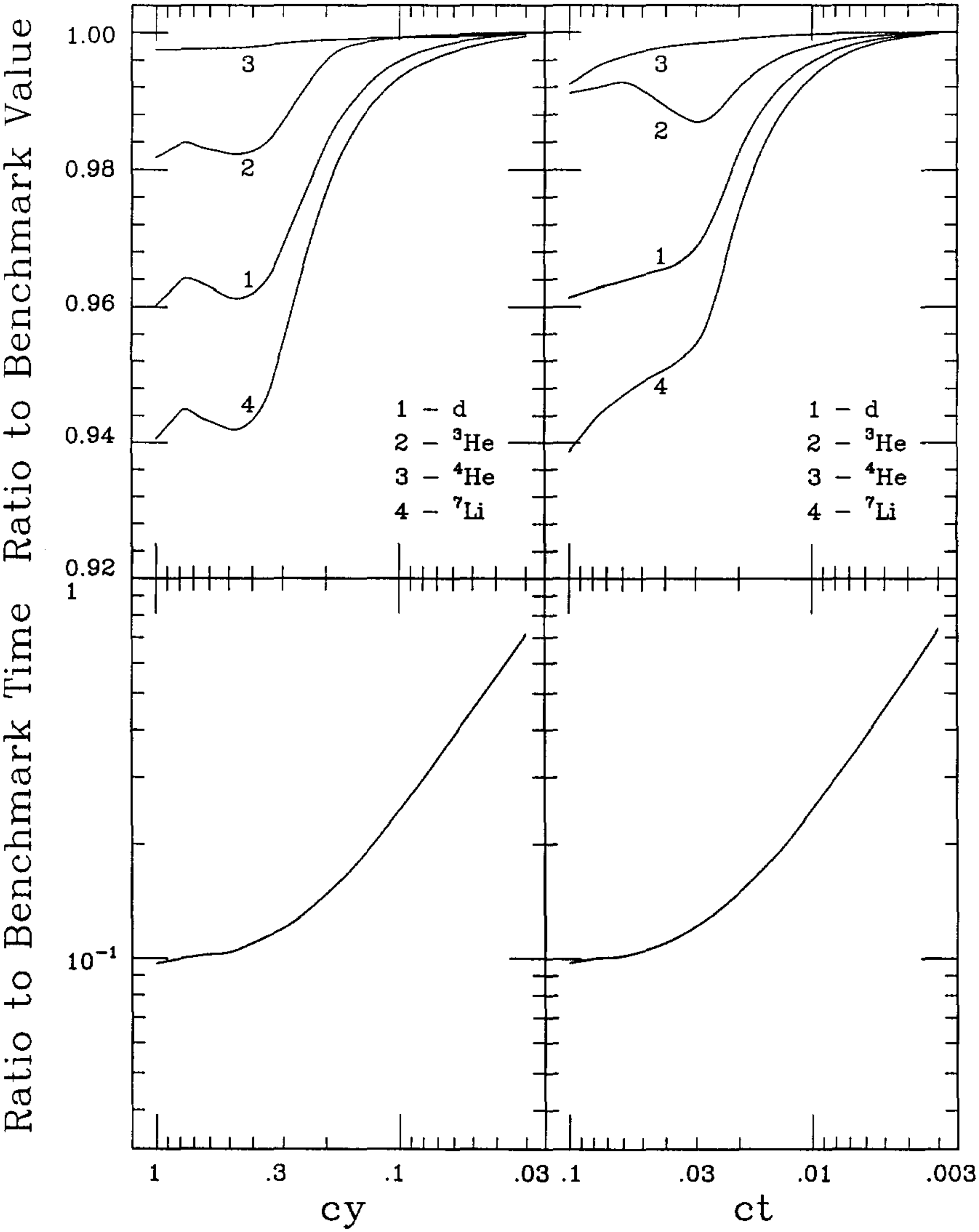


Figure 14a. For $\eta = 10^{-10}$

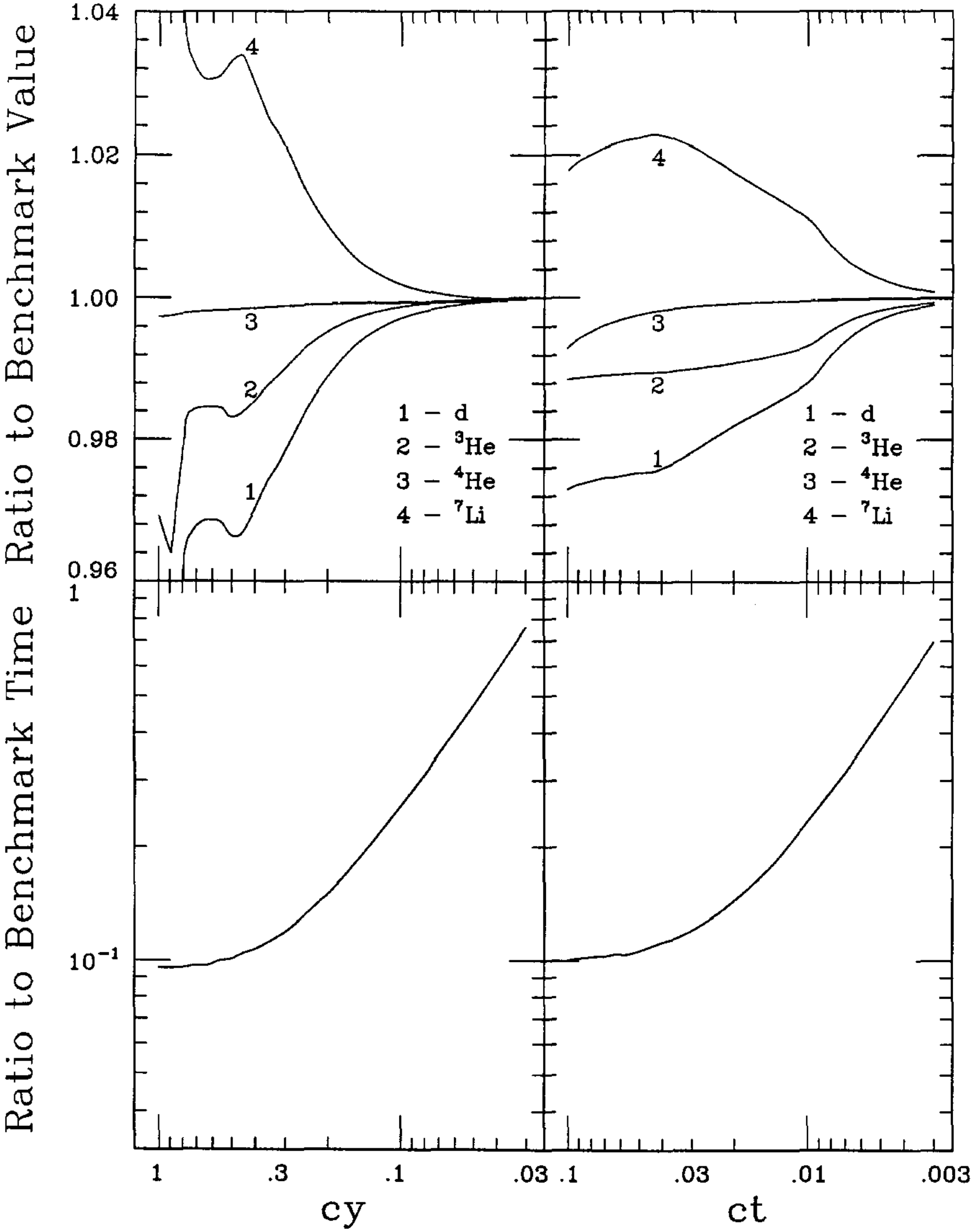


Figure 14b. For $\eta = 10^{-9.5}$

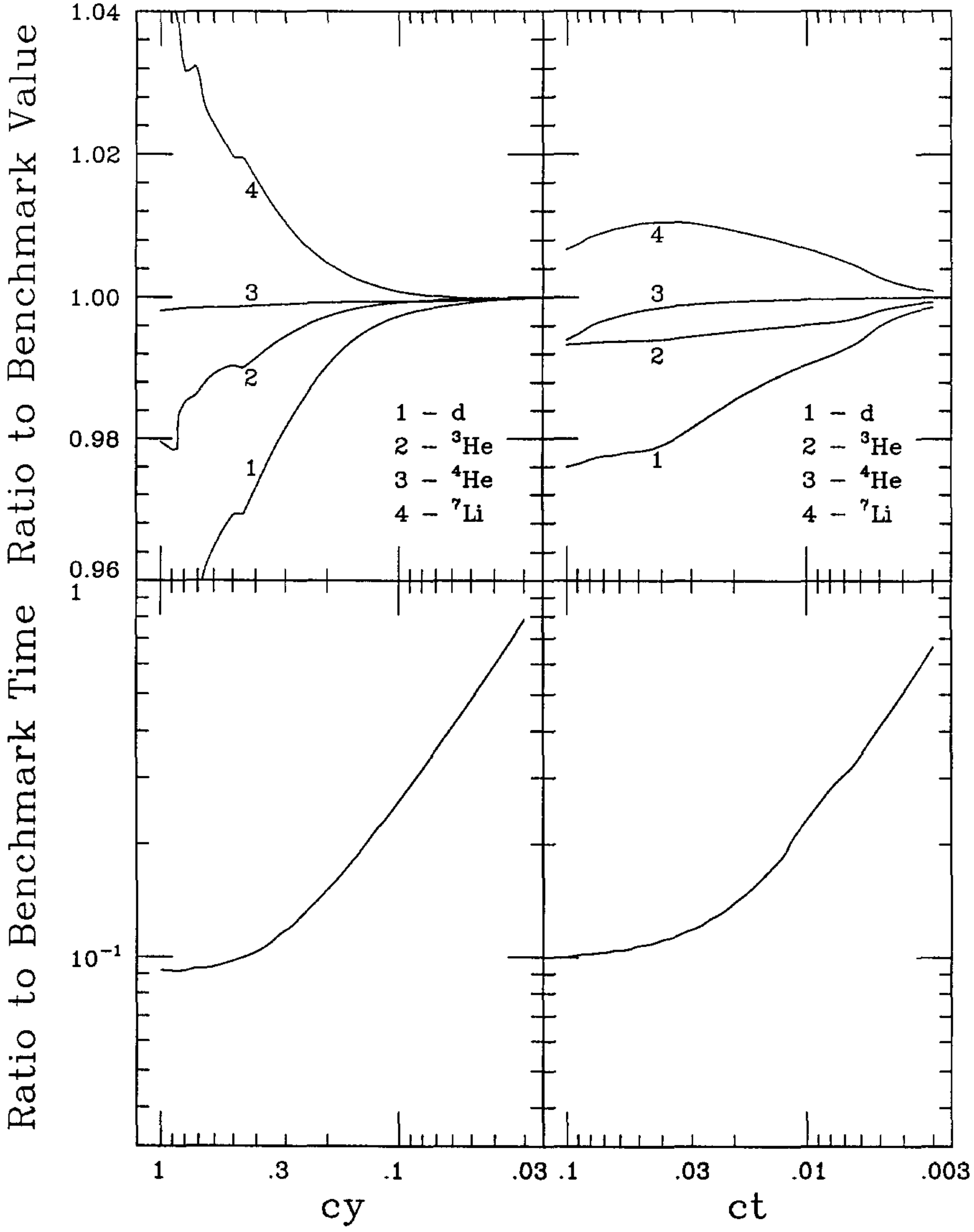


Figure 14c. For $\eta = 10^{-9}$

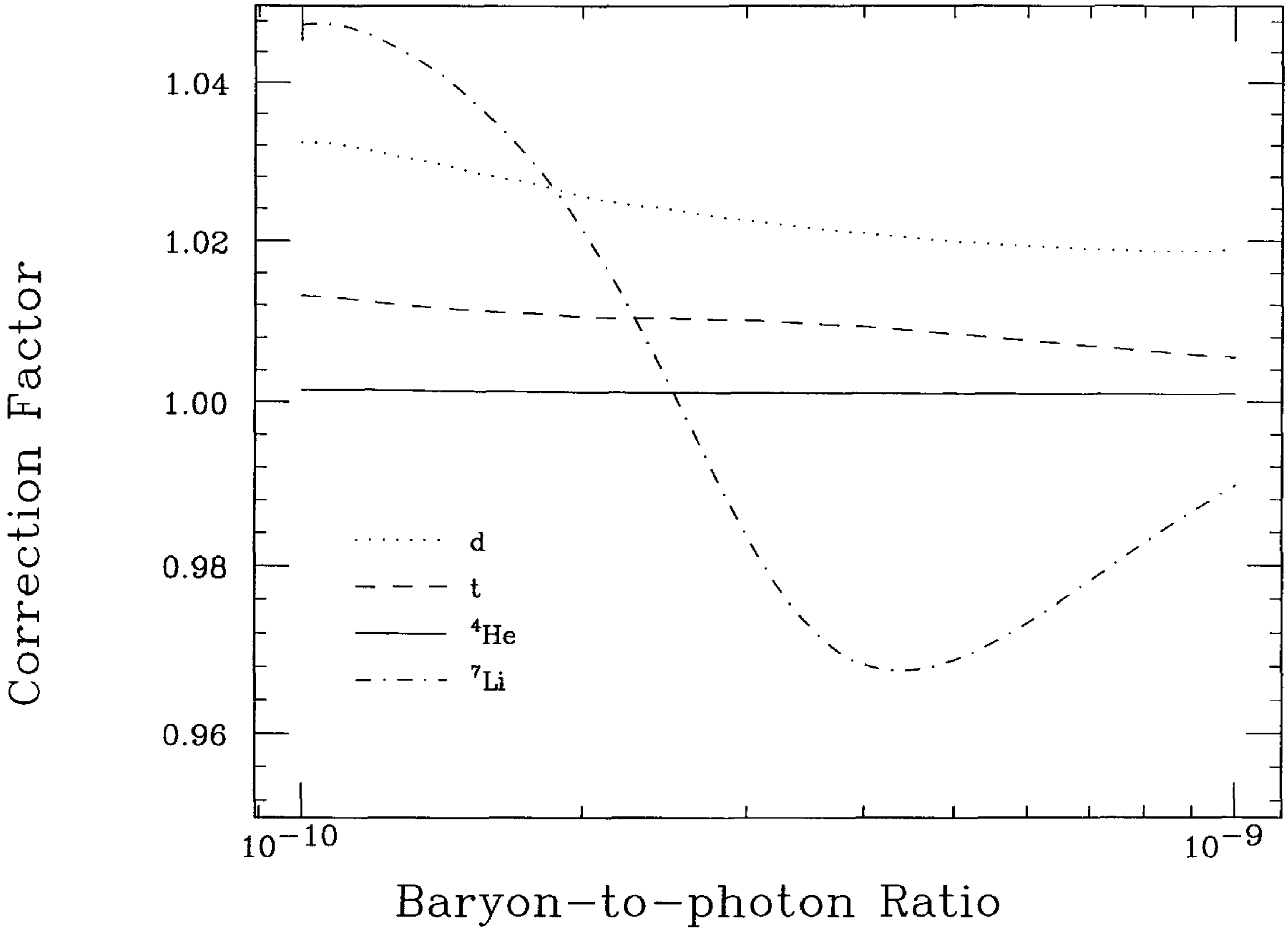


Figure 15. Abundance Correction Factor.

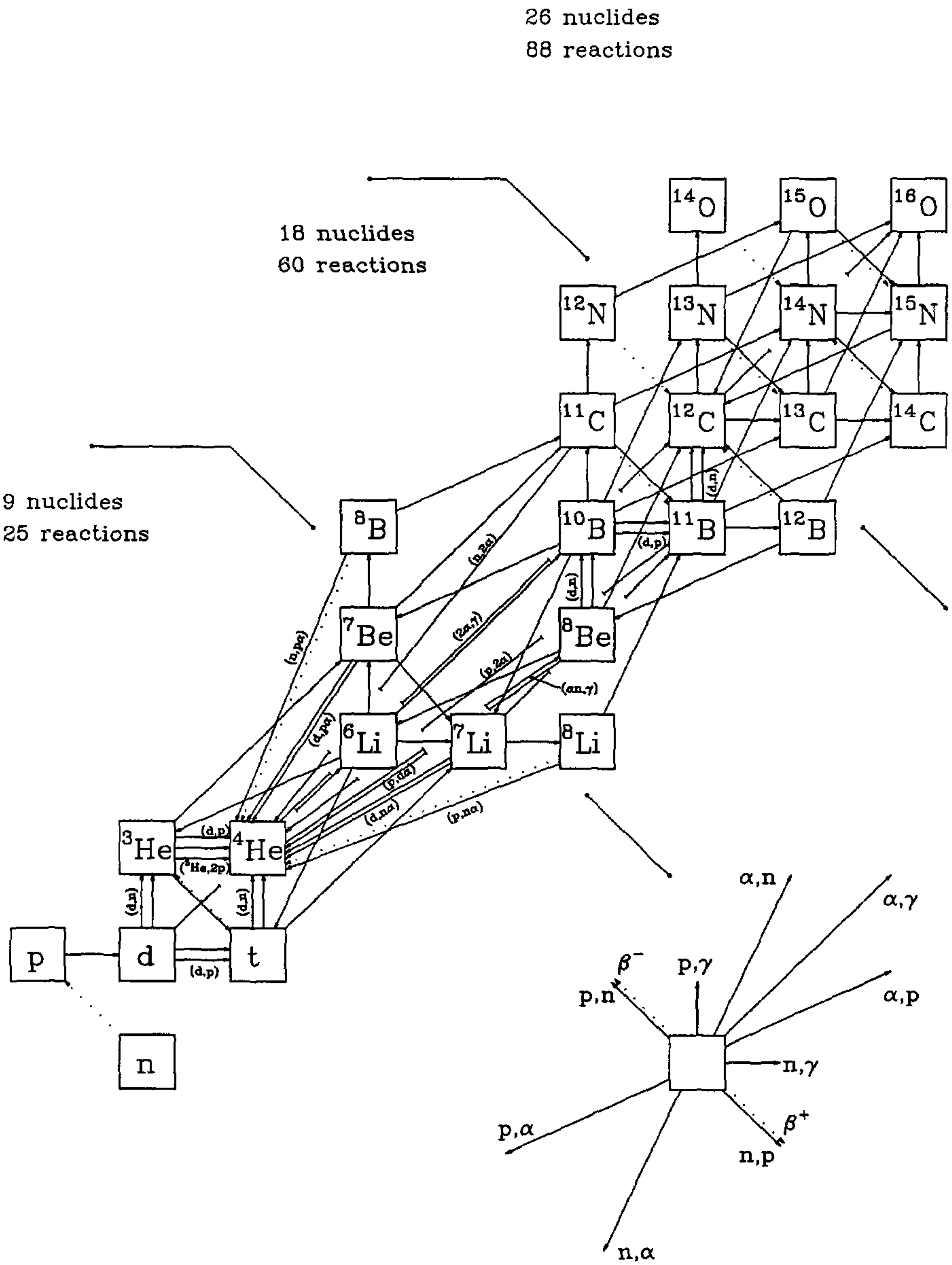
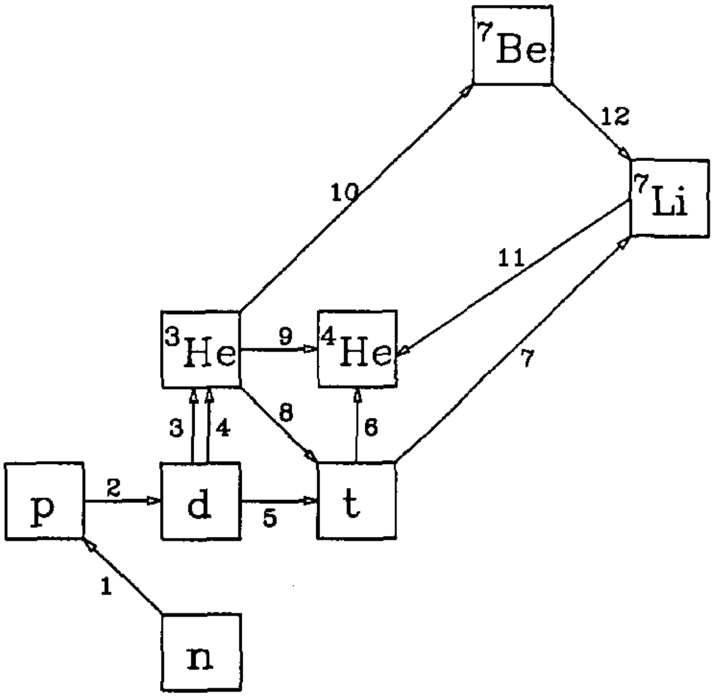


Figure 16. Reaction network.



1. n <-> p
2. p(n,γ)d
3. d(p,γ)^{3}He
4. d(d,n)^{3}He
5. d(d,p)t
6. t(d,n)^{4}He
7. t(α,γ)^{7}Li
8. ^{3}He(n,p)t
9. ^{3}He(d,p)^{4}He
10. ^{3}He(α,γ)^{7}Be
11. ^{7}Li(p,α)^{4}He
12. ^{7}Be(n,p)^{7}Li

Figure 17. Primary Reactions.

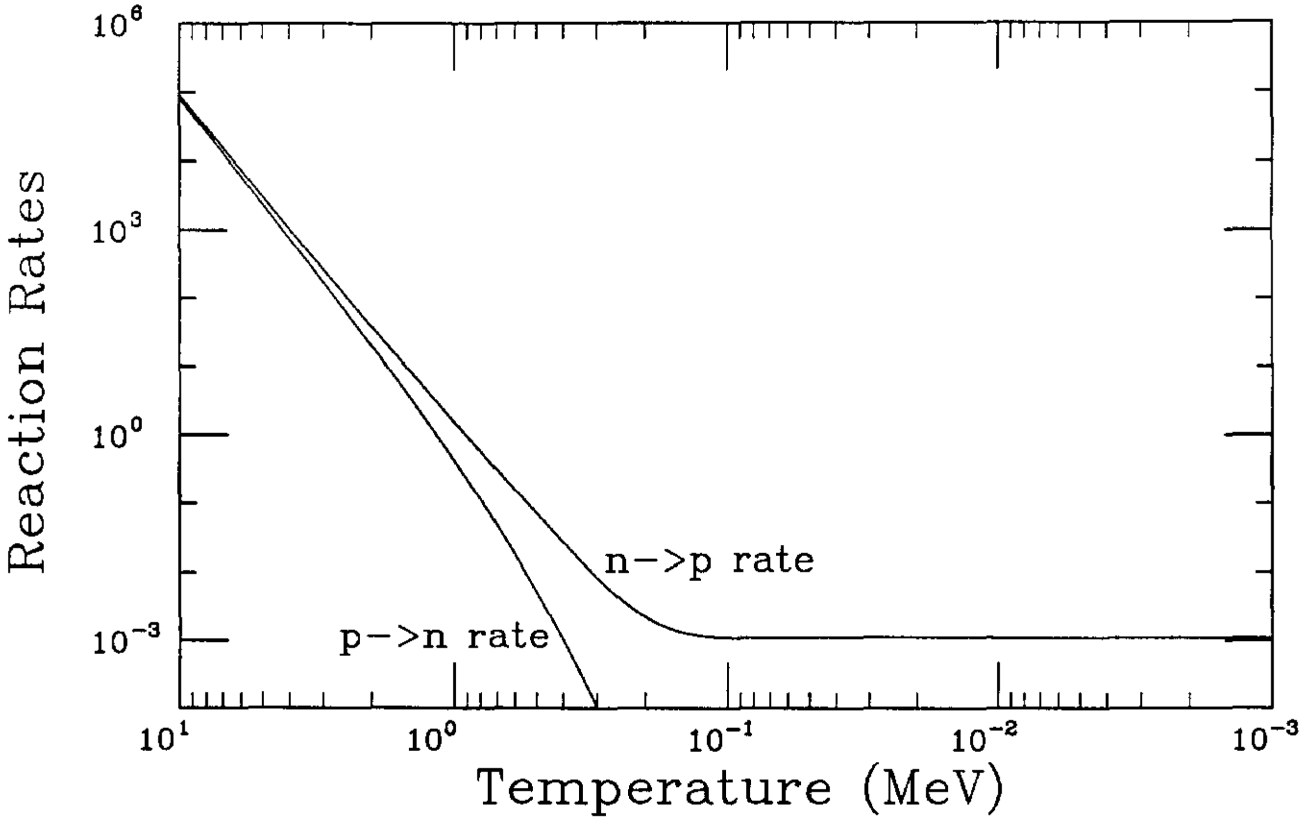


Figure 18a.

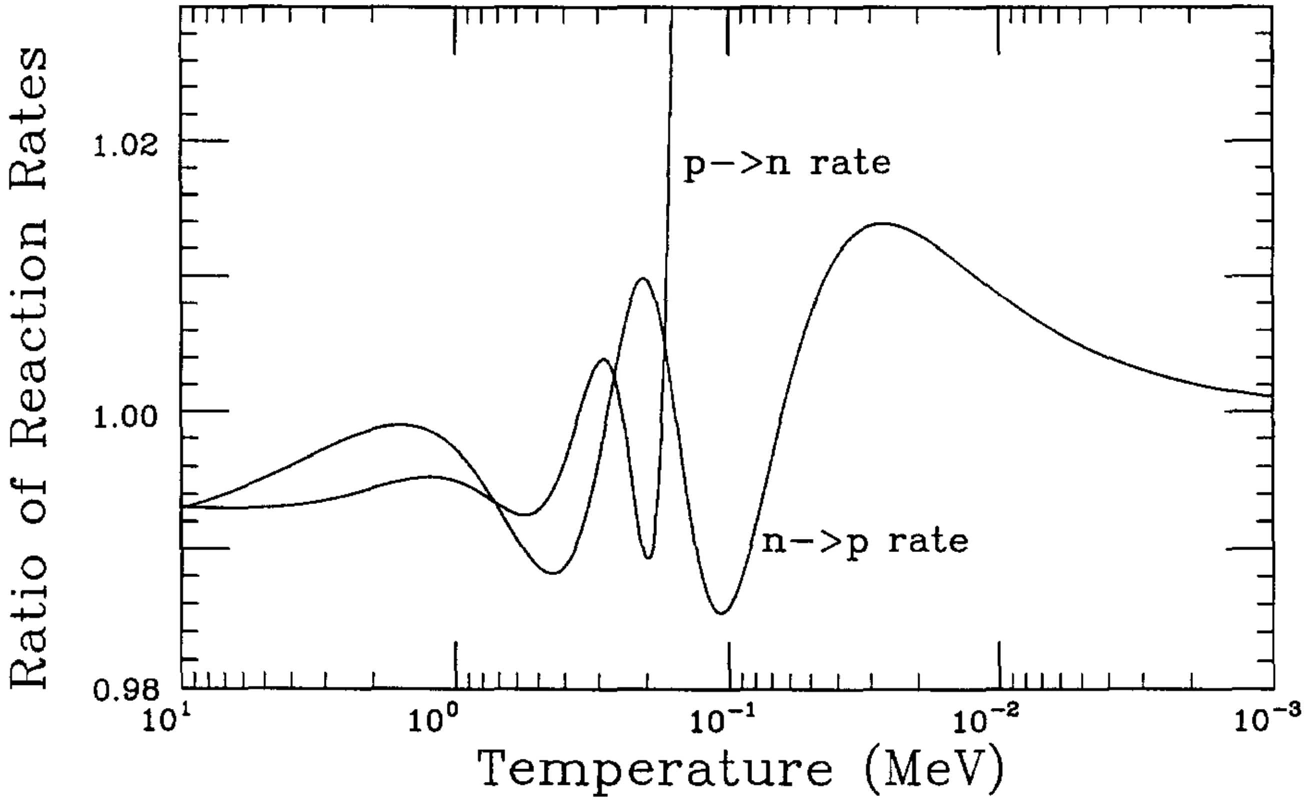


Figure 18b.