AIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100

Silver Spring, Maryland 20910

301/587-8202

Centimeter

Inches

MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

DCHAIN: A User-Friendly Computer Program for Radioactive Decay and Reaction Chain Calculations

L. V. East

Published May 1994

**Idaho National Engineering Laboratory
EG&G Idaho, Inc.
Idaho Falls, ID 83415**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

**Prepared for the
U.S. Department of Energy
Office of Environmental Restoration and Waste Management
Under DOE Idaho Operations Office
Contract DE-AC07-76ID01570**

MASTER
Se

Summary

A computer program for calculating the time-dependent daughter populations in radioactive decay and nuclear reaction chains is described. Chain members can have non-zero initial populations and be produced from the preceding chain member as the result of radioactive decay, a nuclear reaction, or both. As presently implemented, chains can contain up to 15 members. Program input can be supplied interactively or read from ASCII data files. Time units for half-lives, etc. can be specified during data entry. Input values are verified and can be modified, if necessary, before used in calculations. Output results can be saved in ASCII files in a format suitable for including in reports or other documents. The calculational method, described in some detail, utilizes a generalized form of the Bateman equations. The program is written in the C language in conformance with current ANSI standards and can be used on multiple hardware platforms.

1. Introduction

The requirement to determine the populations of individual members of radioactive decay chains arises in variety of areas, ranging from environmental monitoring to nuclear structure experiments. Likewise, it is often required to determine individual nuclide populations in nuclear reaction chains, for example when irradiating materials in a high-flux nuclear reactor. These types of calculations are closely related since the underlying time dependent processes are described by the same form of differential equations.

This report describes a general purpose user-friendly computer program written to handle these problems. The program can handle decay, reaction, or combination decay and reaction chains having up to 15 members with non-zero values allowed for initial populations of any or all of the chain members. Discussions of the underlying calculational principles and program operation are presented in the following sections. Source file listings of the main calculational subroutine are given in Appendix A. Appendix B contains sample output reports.

2. Radioactive Decay and Nuclear Reaction Equations

2.1 Radioactive Decay

The decay of radioactive atoms is described by the differential equation

$$\frac{dN}{dt} = -\lambda N \quad (1)$$

where N represents the number of atoms at a given time and λ is a decay constant characteristic of a particular atomic species. This equation has the well known solution

$$N = N_0 e^{-\lambda t} \quad (2)$$

which gives the number of atoms, N , at time t in terms of the number of atoms, N_0 , at time $t = 0$.

A radioactive decay series of the form $N_1 \rightarrow N_2 \rightarrow \dots \rightarrow N_n$ is described by the set of differential equations

$$\begin{aligned}
\frac{dN_1}{dt} &= -\lambda_1 N_1 , \\
\frac{dN_2}{dt} &= -\lambda_2 N_2 + \lambda_1 N_1 , \\
&\vdots \\
\frac{dN_n}{dt} &= -\lambda_n N_n + \lambda_{n-1} N_{n-1} .
\end{aligned} \tag{3}$$

The solution to this set of equations usually presented in nuclear physics texts is the one first derived by Bateman [1]. This solution assumes that only the parent substance is initially present; i.e.,

$$t = 0; N_1 = N_1^0, N_2^0 = N_3^0 = \dots = N_n^0 = 0 . \tag{4}$$

The number of atoms of the i^{th} member of the chain at time t is given by

$$N_i(t) = C_1 e^{-\lambda_1 t} + C_2 e^{-\lambda_2 t} + \dots + C_i e^{-\lambda_i t} \tag{5}$$

where

$$\begin{aligned}
C_1 &= \frac{\lambda_1 \lambda_2 \dots \lambda_i}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1) \dots (\lambda_i - \lambda_1)} N_1^0 , \\
C_2 &= \frac{\lambda_1 \lambda_2 \dots \lambda_i}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2) \dots (\lambda_i - \lambda_2)} N_1^0 , \\
&\vdots \\
C_i &= \frac{\lambda_1 \lambda_2 \dots \lambda_i}{(\lambda_1 - \lambda_i)(\lambda_2 - \lambda_i) \dots (\lambda_{i-1} - \lambda_i)} N_1^0 .
\end{aligned} \tag{6}$$

To obtain solutions when non-zero initial daughter populations are present, the usual method is to perform an iterative solution to the above Bateman equations by substituting the solution for N_i in an $(i - 1)$ member chain with substance 2 as the parent and a solution for an $(i - 2)$ member chain with substance 3 as the parent, etc. This becomes very tedious for long decay chains. However, Kerr and Walton [2] have re-formulated the Bateman equations into a more general form not requiring zero initial daughter populations and well structured for automatic computation. In their formulation, the population of the k^{th} member of a radioactive decay chain at time t is given by

$$N_k = \sum_{j=0}^k Q_{kj} \quad (7)$$

where

$$\begin{aligned} Q_{kk} &= S_{kk} e^{-\lambda_k t} , \\ Q_{kj} &= \frac{\lambda_{k-1}}{\lambda_k - \lambda_j} Q_{k-1,j} , \quad 1 < k < j \end{aligned} \quad (8)$$

and

$$\begin{aligned} S_{1k} &= N_1^0 , \\ S_{ik} &= N_i^0 + \frac{\lambda_{i-1}}{\lambda_{i-1} - \lambda_k} S_{i-1,k} , \quad 1 < i \leq k . \end{aligned} \quad (9)$$

N_i^0 is the initial population of the i^{th} chain member.

For the more general case in which one or more of the members of the decay chain can undergo multiple modes of decay, one simply multiplies the appropriate λ_i 's in the *numerators* of the above equations by b_i , the branching fraction leading from the i^{th} to the $(i + 1)^{\text{th}}$ member of the chain. Multiple branches must be calculated separately.

Note that S is an upper diagonal matrix and Q is a lower diagonal matrix. Since the diagonal terms of Q are calculated from the diagonal terms of S , the S and Q matrices can share the same computer memory if the S_{ik} terms are calculated before the Q_{kj} terms.

2.2 Nuclear Reactions

If a target is irradiated by particles that induce nuclear reactions, atoms in the target will be transformed into atoms of a new species. The rate of depletion of the target atoms is given by

$$\frac{dN}{dt} = -\sigma \phi N \quad (10)$$

where

σ = cross-section for the reaction (cm^2), and
 ϕ = particle flux (particles/ cm^2/sec).

Eq. (10) has the same form as Eq. (1) for radioactive decay. The number of target atoms, N , present at time t after the beginning of irradiation is given by

$$N = N_0 e^{-\sigma \phi t} \quad (11)$$

where N_0 is the number of target atoms present at $t = 0$.

Typically, the atoms produced by such reactions will undergo radioactive decay. In some instances, a product species can be formed both by irradiation and decay of a parent species. It is also possible that product atoms can undergo nuclear reactions as the result of the incident particles, thus producing a "reaction chain" analogous to the decay chains discussed above. The resulting reaction/decay chain(s) so produced can, indeed, be described in a manner analogous to radioactive decay chains -- as discussed in the following section.

2.3 General Decay/Reaction Chains

When a nuclear reaction is induced in radioactive atoms, the rate of disappearance of the atoms can be described by a modified form of Eq. (1) [or Eq. (10)] that takes into account their disappearance by both transmutation reactions and radioactive decay. This modified equation is

$$\frac{dN}{dt} = -(\lambda + \sigma \phi) N = -\mu N \quad (12)$$

where λ , σ and ϕ are as defined above. The quantity μ can be considered an "effective" decay constant, and the solution has the same form as Eq. (2) with λ replaced by μ .

In the most general case of multiple reactions and decays, the transformation from one member of the chain to the next may occur by a nuclear reaction, radioactive decay, or both. Furthermore, the radioactive decay of one or more chain members may have multiple modes of decay, and multiple reactions can occur in any given chain member. For this rather complex general case, Eq. (3) above can be re-written as [3]

$$\begin{aligned} \frac{dN_1}{dt} &= -(\lambda_1 + \sigma_1 \phi) N_1 = -\mu_1 N_1, \\ \frac{dN_i}{dt} &= -(\lambda_i + \sigma_i \phi) N_i + (b_i \lambda_{i-1} + \sigma_{i-1}^p \phi) N_{i-1} \\ &= -\mu_i N_i + \mu_{i-1}^p N_{i-1}, \quad i > 1 \end{aligned} \quad (13)$$

where

- λ = decay constant,
- b = branching fraction,
- σ = total removal cross-section,
- σ^p = partial cross-section leading to next chain member,
- ϕ = particle flux,
- μ = "effective" decay constant,
- μ^p = partial "effective" decay constant leading to next chain member.

This set of differential equations can be used to represent any combination of nuclear transmutation and radioactive decay. The general solution, analogous to Eqs. (7 - 9), for the k^{th} member of the chain at time t is

$$N_k = \sum_{j=0}^k Q_{kj} \quad (14)$$

where

$$Q_{kk} = S_{kk} e^{-\mu_k t},$$

$$Q_{kj} = \frac{\mu_{k-1}^p}{\mu_k - \mu_j} Q_{k-1,j}, \quad 1 < k < j \quad (15)$$

and

$$S_{1k} = N_1^0,$$

$$S_{ik} = N_i^0 + \frac{\mu_{i-1}^p}{\mu_{i-1} - \mu_k} S_{i-1,k}, \quad 1 < i \leq k. \quad (16)$$

Note that in the case of nuclear transmutations due to particle irradiation, these equations assume that the particle flux is constant during irradiation and that t does not extend beyond the end of the irradiation. To determine populations (modified by radioactive decay only) after irradiation, a second calculation must be carried out (with the particle flux set to zero) using the populations at the end of irradiation as the new initial populations. Separate calculations must also be performed for multiple decay/reaction branches.

2.4 Parent/Daughter Nuclide Relations

Many radioactive decay situations of practical interest consist of a parent and single daughter product. From the discussion above, the number of daughter atoms, N_2 , present at time t can be determined from

$$N_2 = \frac{\lambda_1}{\lambda_2 - \lambda_1} N_1^0 (e^{-\lambda_1 t} - e^{-\lambda_2 t}) + N_2^0 e^{-\lambda_2 t} . \quad (17)$$

Two general cases are of interest:

1. Parent longer lived than daughter ($\lambda_1 < \lambda_2$), and
2. Daughter longer lived than parent ($\lambda_2 < \lambda_1$).

In the first case, a state of equilibrium is reached in which the ratio of the disintegration rates, and hence the activities, become constant. This can be deduced from Eqs. (17) and (2); when t becomes sufficiently large, we obtain

$$\frac{N_1}{N_2} \approx \frac{\lambda_2 - \lambda_1}{\lambda_1} . \quad (18)$$

This condition is referred to as *transient equilibrium*. The limiting case in which $\lambda_1 \ll \lambda_2$ is referred to as *secular equilibrium*. Before equilibrium is reached, the sum of the parent and daughter disintegration rates (and hence the total activity) when no daughter atoms are initially present will pass through a maximum. After equilibrium is reached, the total activity decays with a half-life corresponding to that of the parent.

When the daughter is longer lived than the parent, no equilibrium is reached and the maximum activity occurs at $t = 0$. The activity of the daughter, when no atoms are initially present, increases to a maximum and then the total activity decays with the half-life of the daughter.

Note that for both equilibrium and non-equilibrium situations, the daughter activity reaches a maximum when growing into a freshly separated parent fraction. The time at which this maximum occurs, t_m , can be obtained by differentiating Eq. (17) and setting the result equal to zero at $t = t_m$. Doing this yields

$$t_m = \frac{1}{\lambda_2 - \lambda_1} \ln \frac{\lambda_2}{\lambda_1} . \quad (19)$$

For the limiting case $\lambda_2 \approx \lambda_1 = \lambda$,

$$t_m = \frac{1}{\lambda} , \quad \lambda = \lambda_1 = \lambda_2 . \quad (20)$$

3. Computer Program Description

3.1 General Features

The computer program "DCHAIN" calculates nuclide populations present in radioactive decay and/or nuclear reaction chains. These calculations are performed using the Kerr-Walton formulation of the generalized Bateman equations, Eqs. (14 - 16); a listing of the subroutine that implements these equations is given in Appendix A. Up to 15 chain members can be accommodated,¹ and any or all of the members of the chain can have non-zero initial populations. The program can also be used to calculate the time for maximum parent/daughter activities and approximate time for equilibrium (defined as the time at which the contribution from the exponential term in Eq. (17) is ≤ 0.01 ² assuming $N_2^0 = 0$) to occur.

DCHAIN is written in ANSI C [4] and should be compilable under any ANSI (or POSIX) compliant C or C++ compiler.³ A simple "character cell" user interface is employed that should be compatible with any character cell I/O device, such as a CRT terminal or terminal emulator. A complete package consisting of source files, an executable file for use with xx86 personal computers running DOS⁴ (V2.1 or later), sample data files and documentation can be obtained from the U.S. DOE's Energy Science and Technology Software Center (ESTSC). All requests for copies of the program should be addressed to the ESTSC.

The program is user-friendly with extensive "help" information available and validity checking of input values. Help can be obtained by entering "?" or "help" in response to an input request. Decay and reaction parameters can be entered manually or read from an ASCII file; new or changed parameters can be saved in an ASCII file for reuse. Output results can also be saved in an ASCII file, the contents of which can optionally be sent to a hard-copy device.

¹ The number of chain members has been arbitrarily limited to 15 in order to reduce round-off errors. Maximum chain length is set by a parameter in an "include file" (see Appendix A) and can be easily changed by editing this file and re-compiling the program.

² This corresponds to approximately seven half-lives of the daughter nuclide.

³ The program has been successfully compiled with Microsoft QuickC V2.5 under MS-DOS, VAX C V3.1 under VMS, DEC C V1.3 under OpenVMS/Alpha, GNU C V2.3.3 and other compilers under various versions of unix. The symbol "unix" or "__unix__" must be defined for proper operation under unix (see listing of "dchain.h" in Appendix A).

⁴ The term "DOS" is used in a generic sense to refer to both IBM and Microsoft personal computer disk operating systems (PC-DOS® and MS-DOS®).

3.2 Program Execution

The program is typically executed by entering its name preceded by any system required execution command. On systems that support command-line arguments⁵, the following format can be used:

```
DCHAIN [param1] [param2]
```

where the optional parameters are:

param1 : Name of input data file; or
 "*" to list available input files; or
 "#help" to list help information, where "#" can be "-", "/" or
 blank.

param2 : Name of output (report) file.

If only one parameter is present, it is assumed to be the name of an input file unless a special character or keyword as noted above is present. If an input file is specified, it will be read and verified to be in the proper format. If it is a proper input file, its contents (nuclide names, half-lives, cross-sections, etc.) will be displayed and you will be given the option to use the data as read from the file or edit it before use.

If no parameters are present, you will be presented with the following options menu:

Your options are:

- 1 - Perform General Decay/Reaction Chain Calculation.
- 2 - Perform Decay-Only Chain Calculations.
- 3 - Perform Parent/Child Equilibrium Calculations.
- 4 - See Program Usage Information.
- 5 - See Program Calling Information.
- 6 - Exit Program.

Enter your choice (1 to 6) [1]:

At each request for input, default values (if any) will be shown in "[]" and can be selected by just pressing the RETURN (or ENTER) keyboard key. For example, just pressing RETURN in response to the above input request would automatically select the first menu option.

⁵ DOS, OS/2 and unix systems directly support command-line arguments. Under VMS, an "external command" must be defined in order to pass command-line parameters to DCHAIN.

If "Option 1" is selected, you will be asked if you need instructions on program usage and whether data will be read from a file or entered manually. If manual entry is selected, the following information will be requested:

- Problem title; up to 68 characters of alphanumeric information (optional).
- Number of chain members.
- Irradiation time; default time units are seconds, but other units (minutes, hours, days or years) can be specified by appending the first letter of the desired units (m, h, d or y in upper or lower case) to the value (e.g., 20d = twenty days).
- Particle flux (particles/cm²/sec); not requested if "0" is entered for irradiation time.
- Names of chain members; alphanumeric identifiers such as Co-60, Am-241, etc.
- Initial populations of chain members; can be in any arbitrary units.
- Half-lives of chain members; default units are seconds, but other units can be specified by appending m[inutes], h[ours], d[ays] or y[ears] (see above). (For stable isotopes, enter some astronomically large number like 1e20y.)
- Branching fractions for chain members; will be between 0 and 1 for radioactive decay branches, and 0 for pure reaction branches.
- Total removal cross-sections in barns; enter 0 for pure decay branches.
- Partial cross sections giving rise to the next chain member (0 if the next member chain member is the result of radioactive decay only).

Additional information can be obtained by entering "?" or "help" in response to an input request. Default values, shown in "[]", can be selected by just pressing the RETURN (or ENTER) key. After all values have been entered, they will be displayed in a table and you will be given the option of continuing, editing the input values, restarting or exiting the program. When you chose to continue, you will be given options for saving the input values in a file (with a name of your choice) for later use and saving results in a report file. Times can then be entered for which chain populations will be calculated.

After a time value is entered, the chain populations at the specified time will be immediately calculated and displayed. In order to eliminate displaying meaningless small numbers, population values that are less than a limiting small fraction of the initial population (of the parent, if initial daughter population is zero) will be set to zero. This limiting fractional value will depend on the number of bits used to represent double precision floating point numbers, and is typically $\sim 10^{-16} - 10^{-18}$.

Time value entries are terminated by entering a "Ctrl/Z" character ("Ctrl/D" under unix). After all time values have been entered, the results will be written to an ASCII file if that option was chosen. When results are saved in a file, you will also be given the option of printing the file contents. The options menu shown above is then redisplayed.

"Option 2" is essentially the same as "Option 1", but you will not be prompted for information pertaining to irradiations (irradiation time, particle flux and cross-section values).

If "Option 3" is selected, you will be requested to enter parent and daughter half-lives and the following information will be displayed:

- Time at which maximum daughter activity occurs.

And, for secular or transient equilibrium conditions:

- Time at which maximum total activity occurs.
- Approximate time for equilibrium to be reached.
- Daughter/Parent activity ratio at equilibrium (will be 1.0 for secular equilibrium, > 1.0 for transient equilibrium).

"Option 4" and "Option 5" display program usage and calling parameter information, respectively. "Option 6" terminates program execution; selecting this option is the normal method of exiting the program.

3.3 Data Files

3.3.1 Decay and Reaction Chain Data Files

Decay and reaction chain data can be saved in named ASCII files for later use. The default location of these data files can be specified using an "environment variable" ("logical name" under VMS) named "DCHAIN_IN" (must be specified in upper case under unix). This variable can be defined by using the appropriate operating system command ("set" under DOS, OS/2 and unix, "define" or "assign" under VMS). For example, to specify "D:\DCHAIN\DATA" as the default location of input files under DOS, enter the following command *before* executing DCHAIN:

```
set dchain_in=d:\dchain\data\
```

This command could also be placed in your AUTOEXEC.BAT (or equivalent startup) file. The trailing "\" ("/" under unix) *must* be specified.

If no directory (or "path") is specified when a data file name is entered and DCHAIN_IN is not defined, then the file will be placed (or searched for, if being used for input) in the

current working directory. If no extension is specified for a data file, ".dci" will be assumed. You will be asked for permission before an existing file will be overwritten.⁶

Decay/reaction chain data files consist of alphanumeric strings that mimic keyboard input to DCHAIN. All entries must be present, however, even for input requests that would be skipped during manual data entry. These files can be edited off-line with a text editor, or by using them as input to DCHAIN, editing values interactively, and saving the edited values in a new file.

The following sample data files are supplied with DCHAIN:

EXAMPLE1.DCI : ²⁴³Am decay chain data.

EXAMPLE2.DCI : ²³⁸U neutron capture and decay chain data.

Sample reports generated using these files are shown in Appendix B.

3.3.2 Report Files

Results from DCHAIN calculations can be saved in named ASCII files. The default location of report files can be specified using an "environment variable" ("logical name" under VMS) named "DCHAIN_OUT" (must be specified in upper case under unix). This variable can be defined by using the appropriate operating system command ("set" under DOS, OS/2 and unix, "define" or "assign" under VMS). For example, to specify "D:\DCHAIN\RESULTS" as the default location of output files under DOS, enter the following command *before* executing DCHAIN:

```
set dchain_out=d:\dchain\results\
```

This command could also be placed in your AUTOEXEC.BAT (or equivalent startup) file. The trailing "\" ("/" under unix) *must* be specified.

If no directory (or path) is specified when a report file name is entered and DCHAIN_OUT is not defined, then the file will be placed in the current working directory. If no extension is specified for a report file, ".dco" will be used. An existing file will not be overwritten without your permission.

Report files have a tabular format suitable for printing or including in a document. Times are sorted in increasing order, with duplicate values removed. Sample report file contents are shown in Appendix B.

⁶ Under VMS, a new file version will be created if the name of an existing file is specified for output.

3.4 Hardcopy Output

Results must be saved in a report file if you wish to obtain hardcopy output. When a report file is specified, you will be asked if you wish to print the results before returning to the main options menu. You can specify the desired hardcopy device (or print queue, under VMS) to receive the output.

3.5 Exit Codes

When DCHAIN is executed from a batch (command, script) file, its exit code value can be checked for normal and abnormal program termination conditions. The following exit (decimal) values are returned when running under operating systems other than VMS:

- 0 - No error (normal termination).
- n - System dependent value indicating non-standard program termination (e.g., via Ctrl/C).
- 77 - Memory allocation failure.
- 100 - End of input file (command or data) reached.

Under VMS, the exit 32 bit status word will contain 1 on normal termination. The values 292 and 2160 are returned for memory and EOF errors, respectively, in the lower half of the status word, with the "user error" and "no message" bits set in the upper half.

4. **References**

1. H. Bateman, "Solution of a System of Differential Equations Occurring in the Theory of Radio-active Transformations," *Proc. Cambridge Phil. Soc.* **15**, 423 (1910).
2. T. B. Kerr and R. B. Walton, "Results Obtained Using an IBM 704 Program for Computing ²³⁵U Fission-Product Populations, Activities, and Powers," U.S. Air Force Report AFSWC-TN-59-19 (1959).
3. W. Robinson, "The Equations of Radioactive Transformation in a Neutron Flux," *J. Chem Phys.* **17**, 542 (1949).
4. "American National Standard for Information Systems - Programming Language C," *American National Standards Institute*, X3.159-1989.

Appendix A.

Source Listing of Subroutine Performing Chain Calculations

A.1 Include File DCHAIN.H

```

/*****
 *
 * Include file for DCHAIN Program - macros, global definitions, etc.
 *
 * L. V. East, INEL, March 1993
 *
 *****/

#include <float.h>
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>

/***** Macro Definitions *****/

#ifndef unix
# ifdef __unix__
#   define unix 1
# endif
#endif

#ifndef ABS
# define ABS(x) ( (x) >= 0 ? (x) : (-(x)) )
#endif

#ifndef __DATE__
# define __DATE__ "NO DATE"
#endif

#ifdef unix
# define EOF_CHAR "Ctrl/D"
#else
# define EOF_CHAR "Ctrl/Z"
#endif

#ifndef False
# define False 0
#endif

#define LINE_LEN 78 /* Maximum length of a line of text. */

#ifndef MAX
# define MAX(x,y) ( (x) > (y) ? (x) : (y) )
#endif

```

```

#define MAX_LEN          7          /* Maximun no. characters in a nuclide name. */

#ifndef MAX_NAME
#  ifdef _MAX_PATH          /* Defined by MicroSoft C. */
#    define MAX_NAME      _MAX_PATH
#  else
#    define MAX_NAME      255    /* Maximum file name length. */
#  endif
#endif

#define MAX_NC          15          /* Maximum number of decay chain members. */

#define MAX_TIMES       100        /* Maximun size of results array(s). */

#ifndef MIN
#  define MIN(x,y)        ( (x) > (y) ? (y) : (x) )
#endif

#define TIME_ERROR      -7.070E+07 /* Indicats time conversion error. */

#define TOO_SMALL       DBL_EPSILON * 0.5 /* Value small enough to ignore. */

#ifndef True
#  define True          1
#endif

#define VERSION         "V1.3"      /* Program version ID. */

/***** Exit Status values *****/

#ifdef VMS
#  define NoError        1
#  define MemoryError    292 | 0x30000000 /* VMS INSMEM error code,
                                           * with "no message" and
                                           * "user error" bits set. */
#  define EOFError       2160 | 0x30000000 /* VMS ENDOFFILE error code. */
#else
#  define NoError        0
#  define MemoryError    77          /* Want value > largest signal value. */
#  define EOFError       100
#endif

/***** Global Variables *****/

extern char      name[MAX_NC][MAX_LEN+1], /* Chain member names. */
               title[LINE_LEN+1]; /* Problem description. */

extern double    br[MAX_NC-1], /* Branching ratios. */
               lamba[MAX_NC], /* Decay constants (/sec). */
               NO[MAX_NC], /* Initial populations. */
               sigma[MAX_NC], /* Total cross sections (barns). */
               sigma_p[MAX_NC-1]; /* Partial cross sections (barns). */

extern struct STUFF
{
    float flux; /* Particle flux (/cm^2-sec). */

```

```

float   rtime;           /* Irradiation time (sec).          */
int     number;          /* Number of chain members (<= MAX_NC) */
} param;

```

A.2 Subroutine File CH_CALC.C

```

/* ANSI C function to perform generalized decay/capture chain calculations
 * using Kerr-Walton iterative procedure. The following double precision
 * arrays must be defined globally:
 *
 *   N_O[nc]           Initial populations of chain members.
 *   lambda[nc]        Decay constants (1/sec) of each member.
 *   br[nc-1]          Branching ratios for decay to next chain member.
 *   sigma[nc]         Total reaction removal cross section of each
 *                     chain member (barns).
 *   sigma_p[nc-1]     Cross section leading to next chain member (barns).
 *
 * Returns a pointer to an array containing the results (nc elements), which
 * are the chain populations at time T; pointer will be NULL if nc > MAX_NC,
 * Time < 0 or flux < 0, or if a memory allocation failure encountered.
 * The memory used by the returned array should be deallocated by the caller
 * (using free()) when it is no longer needed.
 *
 * Written by L. V. East, INEL, 24 Feb 1993
 *
 * Edit History:
 *
 *   01 Mar 93   Fine tuned; just return initial populations if
 *               time = 0, check for small differences between
 *               decay constants, made union of S and Q, etc. /lve
 *   18 Apr 93   Allocate space for returned vector dynamically. /lve
 *   21 Apr 93   Fixed spelling errors. /lve
 */

#include "dchain.h"      /* Globals, macros, standard includes, etc. */
#include <math.h>

double *chain_calc( int   nc,          /* Number of decay chain members. */
                   float flux,        /* Neutron flux (neutrons/cm^2*sec). */
                   float T_ir,        /* Irradiation time (seconds). */
                   float Time )       /* Total decay time (seconds). */
{
    double factor, sum, temp, t;
    short count = 1,
           loop;
    register int i, j, k;
    static char *message = "\007\nError: Decay constants %9.3E and %9.3E \
are too close together.\n\n";
    static double mu[MAX_NC],          /* lambda[i] + sigma[i]*flux. */
                  mu_p[MAX_NC-1],      /* b[i]*lambda_p[i] + sigma_p[i]*flux. */
                  nx_O[MAX_NC],        /* N_O[] copy (initial populations). */
                  *N;                  /* Pointer to array containing final
                                     * chain member populations (returned). */

```

```

/* The matrix S(i,k) is upper diagonal and Q(j,k) is lower diagonal, with
 * its diagonal elements computed from S's diagonal elements. Thus, with
 * proper attention to the order of calculations, S and Q can share memory.
 */
static union
{
    double s[MAX_NC][MAX_NC];      /* S(i,k) coefficients. */
    double q[MAX_NC][MAX_NC];      /* Q(j,k) coefficients. */
} x;

/* Check that arguments are within range; return NULL if not.          */
if (nc > MAX_NC || Time < 0.0 || (T_ir > 0.0 && flux <= 0.0))
    return NULL;

/* Allocate space for the final chain population array; return NULL if
 * insufficient memory available (not likely, but have to check anyway...)
 */
if ((N = (double *)calloc( nc, sizeof( double ) )) == NULL)
{
    fprintf( stderr, "\007\nError: Memory allocation failure.\n\n" );
    return NULL;
}

/* Make local copy of initial populations array; setup to return initial
 * populations if Time = 0 (see below).
 */
for (i = 0; i < nc; i++)
    if (Time != 0.0)
        nx_0[i] = N_0[i];
    else
        N[i] = N_0[i];

/* If Time = 0, just return copy of initial populations.                */
if (Time == 0.0)
    return N;

if (T_ir > 0.0)
    factor = flux * 1.0E-24;      /* (particle flux)*barn.          */
else
    factor = 0.0;

/* Note that if Time > T_ir, then we must do the calculation in
 * two parts: 1) effect of reactions + decay up to T_ir, and
 * 2) effect of decay only after T_ir.
 */
if (T_ir > 0.0 && Time > T_ir)
{
    loop = 2;
    t = T_ir;
}
else
{
    loop = 1;
    t = Time;
}

```

```

/* Now do what we came here for... */
while (loop > 0)
{
    if (count > 1)
    {
        factor = 0.0;
        t = Time - T_ir;
        for (i = 0; i < nc; i++)
            nx_0[i] = N[i]; /* Reset initial populations. */
    }

    for (k = 0; k < nc; k++)
    {
        x.s[0][k] = nx_0[0];
        mu[k] = factor * sigma[k] + lamba[k];
        if (k != nc - 1)
            mu_p[k] = factor * sigma_p[k] + br[k] * lamba[k];
        for (i = 1; i <= k; i++)
        {
            temp = mu[i-1] - mu[k];
            if (ABS(temp) < TOO_SMALL)
            {
                fprintf( stderr, message, mu[i-1], mu[k] );
                return NULL;
            }
            x.s[i][k] = nx_0[i] + x.s[i-1][k] * mu_p[i-1] / temp;
        }
    }

    for (k = 0; k < nc; k++)
    {
        x.q[k][k] = x.s[k][k] * exp( -mu[k] * t );
        for (j = 0; j < k; j++)
        {
            temp = mu[k] - mu[j];
            if (ABS(temp) < TOO_SMALL)
            {
                fprintf( stderr, message, mu[k], mu[j] );
                return NULL;
            }
            x.q[k][j] = x.q[k-1][j] * mu_p[k-1] / temp;
        }
    }

    for (k = 0; k < nc; k++)
    {
        sum = 0.0;
        for (j = 0; j <= k; j++)
            sum = sum + x.q[k][j];
        N[k] = ( sum / MAX(nx_0[0], nx_0[k]) > TOO_SMALL ? sum : 0.0 );
    }

    ++count;
    --loop;
}

return N;
}

```

(blank page)

Appendix B.

Sample Output Reports

B.1 Using EXAMPLE1.DCI as Input

This input file contains data for the decay chain starting with ^{243}Am and ending with ^{207}Tl . Note that the short lived nuclide ^{215}Po never attains a significant population and could have been left out of the chain without effecting the resulting populations of successive chain members.

DCHAIN V1.3

Example #1: Am-243 decay chain

22-Apr-94

Initial Conditions:

Nuclide	N O	Half-life	Br. Fr.
Am 234	1.000E+000	7.380E+003 Y	1.000E+000
Np 239	0.000E+000	2.350E+000 D	1.000E+000
Pu 239	0.000E+000	2.410E+004 Y	1.000E+000
U 235	0.000E+000	7.100E+008 Y	1.000E+000
Th 231	0.000E+000	2.550E+001 H	1.000E+000
Pa 231	0.000E+000	3.250E+004 Y	1.000E+000
Ac 227	0.000E+000	2.160E+001 Y	1.000E+000
Th 227	0.000E+000	1.820E+001 D	1.000E+000
Ra 223	0.000E+000	1.143E+001 D	1.000E+000
Rn 219	0.000E+000	4.000E+000 S	1.000E+000
Po 215	0.000E+000	1.780E-003 S	1.000E+000
Pb 211	0.000E+000	3.610E+001 M	1.000E+000
Bi 211	0.000E+000	2.150E+000 M	1.000E+000
Tl 207	0.000E+000	4.790E+000 M	

Calculated Populations:

Time	Am 234	Np 239	Pu 239	U 235	Th 231
1.0000E+001 Y	9.9906E-001	8.7101E-007	9.3778E-004	1.3476E-007	0.0000E+000
1.0000E+002 Y	9.9065E-001	8.6368E-007	9.3340E-003	1.3449E-005	0.0000E+000
1.0000E+003 Y	9.1035E-001	7.9368E-007	8.8349E-002	1.2968E-003	5.3131E-015
5.0000E+003 Y	6.2524E-001	5.4511E-007	3.4710E-001	2.7654E-002	1.1330E-013
1.0000E+004 Y	3.9093E-001	3.4083E-007	5.1763E-001	9.1436E-002	3.7464E-013
5.0000E+004 Y	9.1306E-003	7.9604E-009	3.2901E-001	6.6185E-001	2.7118E-012
1.0000E+005 Y	8.3369E-005	7.2684E-011	8.1106E-002	9.1875E-001	3.7644E-012

Time	Pa 231	Ac 227	Th 227	Ra 223	Rn 219
1.0000E+001 Y	4.3758E-016	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
1.0000E+002 Y	4.3778E-013	1.3784E-016	0.0000E+000	0.0000E+000	0.0000E+000
1.0000E+003 Y	4.2403E-010	2.5732E-013	5.9350E-016	3.7268E-016	0.0000E+000
5.0000E+003 Y	4.6029E-008	3.0059E-011	6.9342E-014	4.3547E-014	0.0000E+000
1.0000E+004 Y	3.1031E-007	2.0453E-010	4.7184E-013	2.9632E-013	0.0000E+000
5.0000E+004 Y	1.1776E-005	7.8185E-009	1.8037E-011	1.1328E-011	0.0000E+000
1.0000E+005 Y	2.9285E-005	1.9458E-008	4.4889E-011	2.8191E-011	1.1419E-016

Time	Po 215	Pb 211	Bi 211	Tl 207
1.0000E+001 Y	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
1.0000E+002 Y	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
1.0000E+003 Y	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
5.0000E+003 Y	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
1.0000E+004 Y	0.0000E+000	6.4993E-016	0.0000E+000	0.0000E+000
5.0000E+004 Y	0.0000E+000	2.4845E-014	1.4797E-015	3.2966E-015
1.0000E+005 Y	0.0000E+000	6.1831E-014	3.6825E-015	8.2042E-015

B.2 Using EXAMPLE2.DCI as Input

This input file contains data for the ^{238}U neutron capture and decay chain ending with ^{241}Am . Note that ^{239}U and ^{239}Np quickly decay away after the end of neutron irradiation.

DCHAIN V1.3

Example #2: U-238 neutron capture.

22-Apr-94

Flux = 1E+013 particles/(cm² sec).
Irradiation Time = 463 D.

Initial Conditions:

Nuclide	N_0	Half-life	Br. Fr.	Sigma	Sigma_p
U 238	1.000E+000	4.470E+009 Y	0.000E+000	1.430E+001	1.420E+001
U 239	0.000E+000	2.350E+001 M	1.000E+000	0.000E+000	0.000E+000
Np 239	0.000E+000	2.355E+000 D	1.000E+000	6.500E+001	0.000E+000
Pu 239	0.000E+000	2.415E+004 Y	0.000E+000	1.240E+003	4.800E+002
Pu 240	0.000E+000	6.537E+003 Y	0.000E+000	2.850E+003	2.850E+003
Pu 241	0.000E+000	1.450E+001 Y	1.000E+000	1.350E+003	0.000E+000
Am 241	0.000E+000	4.330E+002 Y		1.200E+001	

Calculated Populations:

Time	U 238	U 239	Np 239	Pu 239	Pu 240
6.0000E+001 D	9.9926E-001	2.8864E-007	4.1647E-005	6.7310E-004	7.6430E-006
1.2000E+002 D	9.9852E-001	2.8843E-007	4.1616E-005	1.3433E-003	3.0189E-005
4.6300E+002 D	9.9430E-001	2.8721E-007	4.1440E-005	4.4381E-003	3.2034E-004
4.8000E+002 D	9.9430E-001	0.0000E+000	2.8016E-007	4.4795E-003	3.2034E-004
2.0000E+000 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4797E-003	3.2032E-004
5.0000E+000 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4793E-003	3.2022E-004
1.0000E+001 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4787E-003	3.2005E-004
2.0000E+001 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4774E-003	3.1971E-004
1.0000E+002 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4671E-003	3.1701E-004
5.0000E+002 Y	9.9430E-001	0.0000E+000	0.0000E+000	4.4161E-003	3.0384E-004

Time		Pu 241	Am 241
6.0000E+001	D	3.5663E-007	6.7557E-010
1.2000E+002	D	2.8816E-006	1.1328E-008
4.6300E+002	D	1.1804E-004	1.9786E-006
4.8000E+002	D	1.1778E-004	2.2408E-006
2.0000E+000	Y	1.1398E-004	6.0347E-006
5.0000E+000	Y	9.8750E-005	2.1196E-005
1.0000E+001	Y	7.7756E-005	4.1933E-005
2.0000E+001	Y	4.8209E-005	7.0560E-005
1.0000E+002	Y	1.0526E-006	1.0487E-004
5.0000E+002	Y	5.2239E-015	5.5854E-005

DATE

FILMED

8/17/94

END