PCS
A Code System for Generating
Production Cross Section Libraries

Lawrence J. Cox

April 1997

This is an informal report intended primarily for internal or limited external
distribution. The opinions and conclusions stated are those of the author and may or
may not be those of the Laboratory.
Work performed under the auspices of the U.S. Department of Energy by the
DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California 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 the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd., Springfield, VA 22161
PCS

A Code System for Generating Production Cross Section Libraries

Larry Cox

April 27, 1995

Abstract

This document outlines the use of the PCS Code System. It summarizes the execution process for generating FORMAT2000 production cross section files from FORMAT2000 reaction cross section files. It also describes the process of assembling the ASCII versions of the high energy production files made from ENDL and Mark Chadwick's calculations. Descriptions of the function of each code along with its input and output and use are given.

This document is under construction. Please submit entries, suggestions, questions and corrections to (ljc@llnl.gov)
Contents

Introduction

Overview ............................................. 1
Platforms ............................................. 2
Availability .......................................... 2
Contacts ............................................. 3
Contributors ......................................... 3

Using PCS

PCScratch: Generating 0-20 MeV PCS files ........... 4
   Input Data Sources .................................. 5
Assembling High Energy Files from Two Sources .... 6
Thinning I3 data ...................................... 7
Energy Balance ....................................... 8
   Elastic Scattering Distributions .................... 8
   Energy-Angle Correlated Distributions .............. 9
   Isotropic Energy Distributions .................... 9
   Average Q-value ................................... 9
Angle-Integrated Distributions and Kerma Factors .... 10
   Generation of Angle-Integrated Distributions ....... 10
   TeX formated Tables ............................... 11
Codes  
Source Code Version Control ........................................... 12

A Algorithms  
Goal ........................................................................... 13
Output files ....................................................................... 14
Input files .......................................................................... 14
Accumulating the reaction list ........................................... 15
C\YO list for ENDL ............................................................ 15
Input Data format variations .............................................. 17
Contents of the output files ................................................ 19
One Algorithm for accumulating these values .................... 20
Master lists of coordinate values ....................................... 20
Another Algorithm for accumulating these values ............... 21
Two body reactions ............................................................ 21
Making I=(0,1,3)(LAB FRAME) from ENDL I=(0,1)(CM FRAME) 22
Renormalizing the probability distributions: (I=0,1)CM .... 23
Multi-body reactions with I=4(l=0) data (no I=1 data) ....... 23
Making I=(0,1,3) from ENDL I=(0,4) ................................. 23
S=1 switch special case .................................................... 25
Other special cases .......................................................... 25
Adding two I=(0,[1,3] or [4]) sets ..................................... 26
I=0 ............................................................................. 26
List of Tables

1 Production Particle Types to Consider ............ 13
2 Neutron Producing Reaction Numbers ............ 16
3 YO occurrences for C values in ENDL, ECPL and EGDL .... 18
Introduction

The PCS code system was developed to generate a new set of nuclear data libraries in the production cross section format.

The standard representation of nuclear data in ENDL has historically been reaction based. That is, each reaction is tabulated separately.

With the advent of the PEREGRINE Project, the need for nuclear data to higher energy arose. With higher incident energies, the number of specific reaction channels available increases dramatically. It was decided to process the data into only two tabulations: Elastic scattering and Non-elastic particle production. With this simplification, an increasing number of reaction channels does not increase the complexity of the data library. Elastic scattering is kept separate because of its simplicity. All other reactions—inelastic-scattering, capture, etc.—are added together to produce the non-elastic cross section.

Overview

The main purpose of this set of codes is to generate FORMAT2000 Production Cross Section libraries from the reaction based data in ENDL, ECPL and EGDL together with the new, high energy evaluations from Mark Chadwick.

The main module, PCScreate, adds together all available reaction data for a given incident particle onto a specific target isotope from the chosen data library excluding elastic scattering. Angular distributions, Energy-angle correlated spectra, average energy, and multiplicities are generated for each particle type produced as determined from the reaction data source files. In addition, the average Q-value is calculated and tabulated.

As input, PCScreate, by default, reads the ENDL, ECPL, or EGDL libraries directly, based on the incident particle type, and processes all files listed in the .index file of the appropriate NDS subdirectory. It can also be told to read private copies of data files via a command line option.

The result of processing an isotope from ENDL is a set of files comprising a subdirectory entry for that isotope into PCSL20, the 0-20 MeV production cross section
INTRODUCTION

library. The files are created in the directory in which PCScreate is executed.

Codes and scripts are supplied to facilitate concatenation of these low energy files to the 20+ MeV files being generated by Mark Chadwick’s new evaluations of biologically important nuclei. At this time, the process is not yet fully automated. Some hand processing is still necessary, but the amount is minimal if all steps in the processing are followed.

Platforms

PCScreate and its ancillary codes were written and are executable on the following platforms:

- Sparc 2 running Sun OS4.1.3
- Sparc 10, 20 running Solaris 5.3

The various codes and scripts should be easily portable to other platforms.

Availability

Currently, the code is available for execution on PDNET only. (See system manager, Jay McGowan (jdm@llnl.gov), for an account.)

Executables for all platforms are found in one place:

/h/pd7/cox/projects/pcs/bin

This user manual is available in postscript form:

/h/pd7/cox/projects/pcs/doc/ps/pcs.ps

It is also available as online HTML:

/h/pd7/cox/projects/pcs/doc/html/pcs.html
INTRODUCTION

Contacts

The primary contact for PCScreate is Larry Cox (x3-6610) (ljc@llnl.gov). As he is no longer a Nuclear Data Group member, the codes will be transferred to Dave Resler (daresler@llnl.gov) for keeping and long term support. Mark Chadwick (mbchadwick@llnl.gov) is the primary contact for the high energy ASCII data files.

For computer systems support contact Jay McGowan.

Contributors

The PCScreate code system is maintained under the auspices of the Nuclear Data Group of N-Division in the Physics and Space Technology Directorate at the Lawrence Livermore National Laboratory.

Many individuals at the Laboratory have contributed in some way to this project.

<table>
<thead>
<tr>
<th>Name</th>
<th>Contact</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Larry Cox</td>
<td>(<a href="mailto:ljc@llnl.gov">ljc@llnl.gov</a>)</td>
<td>PCS source code; Other codes and scripts</td>
</tr>
<tr>
<td>Dave Resler</td>
<td>(<a href="mailto:daresler@llnl.gov">daresler@llnl.gov</a>)</td>
<td>Nuclear Data; Algorithm Discussions</td>
</tr>
<tr>
<td>Mark Chadwick</td>
<td>(<a href="mailto:mbchadwick@llnl.gov">mbchadwick@llnl.gov</a>)</td>
<td>Extended Nuclear Data</td>
</tr>
<tr>
<td>Gerry Hedstrom</td>
<td>(<a href="mailto:ghedstrom@llnl.gov">ghedstrom@llnl.gov</a>)</td>
<td>Software Support</td>
</tr>
<tr>
<td>Jim Rathkopf</td>
<td>(<a href="mailto:rathkopf@llnl.gov">rathkopf@llnl.gov</a>)</td>
<td>Library Testing (MCAPM, DATACC)</td>
</tr>
<tr>
<td>Roger White</td>
<td>(<a href="mailto:rmwhite@llnl.gov">rmwhite@llnl.gov</a>)</td>
<td>Group Leader; Nuclear Data Algorithm Discussions</td>
</tr>
<tr>
<td>Ernie Plechaty</td>
<td>(retired)</td>
<td></td>
</tr>
<tr>
<td>S. T. Perkins</td>
<td>(deceased)</td>
<td></td>
</tr>
<tr>
<td>Bob Gulliford</td>
<td>(<a href="mailto:gulliford@llnl.gov">gulliford@llnl.gov</a>)</td>
<td></td>
</tr>
<tr>
<td>Lila Chase</td>
<td>(<a href="mailto:lchase@llnl.gov">lchase@llnl.gov</a>)</td>
<td></td>
</tr>
<tr>
<td>Jay McGowan</td>
<td>(<a href="mailto:jdm@llnl.gov">jdm@llnl.gov</a>)</td>
<td></td>
</tr>
</tbody>
</table>
Using PCS

This chapter gives the details of processing FORMAT2000 reaction cross section data files into production cross section files. It also gives the steps needed to assemble a complete high energy library.

The executable codes are all kept in one place. The current floor versions are located in

`cox/projects/pcs/bin/`

The sources for the fortran codes are located in subdirectories of

`cox/projects/pcs/src/`

by the name of the code module. The code documentation, such as it is, is kept in

`cox/projects/pcs/doc/`

PCScreate: Generating 0-20 MeV PCS files

COMMAND: PCScreate

USAGE: PCScreate <YI> <YO> <ZA> [datapath]

Argument Definition
YI YI number [1-7]
YO YO number [0-7]
ZA ZA number: 1000*Z+A

Option Definition
datapath alternate path to data files

The main module in the PCS code system is PCScreate. This is the package that reads reaction cross section files, processes them into energy-angle correlated form and adds them up into production cross sections. It is easy to use and has been thoroughly tested.

One execution of PCScreate generates a set of files specific to the projectile, the ejectile and the target. If there are no reactions given between the projectile and the target that generate the ejectile, no files are produced.
Input Data Sources

Processing /nds/ Directly

By default, PCScreate directly access the /nds/ data structure based on the projectile (YI), the ejectile (YO), and the target (ZA). For incident neutrons (YI=1), PCScreate looks for and reads the file /nds/endl/za<ZZZ><AAA>/index to determine what particle producing reactions need processing. For incident charged particles (YI=n ∈ {2 – 6}), PCScreate looks for and reads the file /nds/ecpl/yi<nn>/za<ZZZ><AAA>/index to determine what particle producing reactions need processing.

Processing Private Data Files

Private copies of data files can be processed by PCScreate by using the [datapath] command line option. By specifying either a relative or absolute path as the fourth command line argument, PCScreate can be told exactly where to look for input data. It will look for the file [datapath]/index for a list of data files. All files that you wish to have processed must be listed in this file and must be located in the same directory. PCScreate will inform you if it cannot find a complete, valid set of files or if any files given in [datapath]/index are not found.

The index file needs to be in a certain order. The following command will generate the file in the correct order assuming you want all files matching the pattern yo*.

```
ls yo* | sort -tc +1.0n -1.2 +0.2n -0.4 +1.3n -1.6 >.index
```

The use of the index file allows the user the flexibility to exclude certain reactions or to include modified cross section files. This was used extensively during development and may be useful in exploring what cross sections are important to PEREGRINE.

It is not currently possible to mix data files from different directories by the use of a custom index file specifying the paths of each file therein. The file names as read from index are parsed for the YO, C, I and S numbers assuming a fixed length of 15 characters. Adding paths to the index file would lead to parsing errors. If this feature is ever needed, path parsing could be added. The subroutines that read values from the file name array of filenames “file(i)” would be the affected code. The index to be used will be the one found at [datapath]/index.
USING PCS

Assembling High Energy Files from Two Sources

COMMAND: concat

USAGE: concat [-merge_energy] <Low-E path> <High-E path>

<table>
<thead>
<tr>
<th>Option</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>-merge_energy</td>
<td>The energy at which to switch from the low energy files to the high energy files (default 20.0MeV, equivalent to concat -20.0 ...)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Argument</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-E path</td>
<td>Directory containing low energy files</td>
</tr>
<tr>
<td>High-E path</td>
<td>Directory containing high energy files</td>
</tr>
</tbody>
</table>

CONCAT is a FORTRAN code that reads data files from two sources and merges them into a single ASCII data library. The user must specify the complete paths to both the low and high energy files, giving the low-E path first. To determine which files to use, CONCAT reads the file .file_match to determine which files to use and what may be missing. CONCAT looks specifically for ../../.file_match. That is, it look two directories above the current directory. The output is placed in the current directory. Any missing files are reported to standard output. Files without matches will result only in a report of what is missing. Files with matches will have the information from the low energy file up to and including the merge energy (default 20 MeV) followed by the information the high energy file that is above the merge energy. The files produced are terminated with the usual “1” in column 72 preceded by 71 blanks.

The merge energy option is useful in assembling files with a different merge energy. If that energy occurs in both file (and is not the default 20 MeV) then that energy is taken from the low energy file. Only energies greater than the merge energy are taken from the high energy files.

For the default case of merging 0-20 MeV data onto the new high energy evaluations, the 20 MeV data from the high energy file is shifted to 20.001 MeV. No attempt to smooth across the transition is made. The header information is taken from the low energy file.

No provision is made to partially process the directories. However, by editing the .file_match file, the user could select the data he/she wishes to process. This may be useful if a concatenation of two large data sets is mostly successful with only one--or a few--missing pieces. Those pieces could then be provided and processed singly.
by using a `file_match` with all other lines commented out. I haven't tried this, but I can see no reason it wouldn't work.

In some cases, manual work is needed on the output files. One specific example of this is the production of tritons off of nitrogen. Chadwick does not routinely provide triton production data above 20 MeV. Therefore, CONCAT, when merging PCSL20/za007014 with a set of Chadwick's calculation on nitrogen, will return the messages:

Matching File not found for:
/nds/data/pcs120/za007014/yi01/yo04c05i001s000
/nds/data/pcs120/za007014/yi01/yo04c05i003s000
/nds/data/pcs120/za007014/yi01/yo04c05i009s000

In these cases, I copy the files listed into the current workspace and modify the multiplicity file to show zero multiplicity from 20 MeV to the maximum energy of the high energy file set. No extra distributions are needed in the I1 an I3 files since no particles are produced at the extended energies.

There have also been cases where Chadwick produces secondary particles not produced in PCSL20. In these cases, I copy the files from the high energy data set and modify the multiplicity file to show zero multiplicity below 20 MeV. These cases occurred when Chadwick was producing tritons off of nuclei other than nitrogen for kerma testing purposes.

### Thinning I3 data

**COMMAND:** thinI3

**USAGE:** thinI3 <file> [t]

- **Argument** *Definition*
  - file: The name of the I3 input file; It is also used as the pattern for the I1 file.

- **Option** *Definition*
  - t: truncates incident energies at 20 MeV

The I3 data files that result from PCScreate are sometimes very large. Angular data is generated for each incident energy found in the original files at 21 distinct angles.
THINI3 removes entirely some of the tabulated incident energies in the I3 data file and the associated II data file. A hard coded tolerance of 0.01 is used in average secondary energy removing incident energy points that don’t change the average by more than the tolerance. A 20 MeV incident energy is always retained.

Both II and I3 files are thinned because MCFGEN assumes that the set of incident energies and angles are identical between them the two. Both files must be present for processing to generate thinned files.

The output files have the same name as the original file with the added suffix of *.thn. The following shell script will rename the original files to be *.fat and move the new *.thn files into the original names.

```
#!/bin/csh
# get the file list and its length
set var='ls *.thn'
set n=$#var
while ( $n )
  set new='echo $var[$nl | tr '[.' ']' '
  mv $new[1] $new[1].fat
  mv ${var[$nl]} ${new[1]}
  @n=($n - 1)
end
exit
```

**Energy Balance**

**Elastic Scattering Distributions**

**COMMAND:** edepCM

**USAGE:** edepCM <file>

**Argument**
- **file** Definition
  - II file name. Also used as pattern for other file names needed.

**Option**
- **t** Definition
  - truncate incident energies at 20 MeV
USING PCS

Energy-Angle Correlated Distributions

COMMAND: edepI3
USAGE: edepI3 <file>[t]

Argument Definition
file I3 file name. Also used as pattern for other file names needed.

Option Definition
  t truncate incident energies at 20 MeV

Isotropic Energy Distributions

COMMAND: edepI4
USAGE: edepI4 <file>[ t | no9 ]

Argument Definition
file I4 file name. Also used as pattern for other file names needed.

Option Definition
  t truncate incident energies at 20 MeV
  no9 Do not use I9 multiplicity file

Average Q-value

COMMAND: qbar
USAGE: qbar [path to files]

COMMAND: qbardrv
USAGE: qbardrv [path to files]

Option Definition
  path uses other directory for input

These two codes calculate the reaction averaged Q-value by summing over the average energy produced in secondary particles and subtracting that value from the incident
energy. Also subtracted is the average energy deposited in heavy recoils. The result is written to an I12 file created in standard FORMAT2000.

\[ Q = E_{\text{inc}} - \sum_{yo=1}^{7} \nu E_{yo} - \overline{E}_{\text{recoil}} \]

The average recoil energy is from the I11 file. The average particle energies are taken from an I10 file. Except in the case of yo=7 (gammas), the multiplicities are folded into the I10 files. For gammas, the I9 file is used.

The two codes different only in the choice of I10 files:
- Qbar uses the I10 files produced by PCScreate which are calculated from the reaction based I10 files.
- Qbardrv tries to use the files generated by the various edep* routines, specifically looking for the suffix ".drv". It generates an I12 file with the "drv" suffix.

Both codes will fail if no I11 (recoil energy) file is present. Given an I11 file, they will both pick up and report the use of any available appropriate I10 files.

The use of both of these codes yields a three way comparison of the average Q-value since PCScreate also generates an I12 file during yo=0 processing. This was very useful in the debugging phase of PCS development.

The other use of qbar is to generate the average Q-value from Chadwick's calculates as that is the one set of values he does not yet provide.

Angle-Integrated Distributions and Kerma Factors

Generation of Angle-Integrated Distributions

COMMAND: I3tol4

USAGE: I3tol4 <file>
Argument Definition

file I3 file name. Also used as pattern for other file names needed.

For the purposes of comparison to angle-integrated sampling of the energy-angle correlated distributions, it is convenient to have angle-integrated distributions available. The code I3toI4 does this translation by performing the angular integration over the I0/I1/I3/I9 distributions. Angular steps of 5 degrees and unit-based transformations/interpolations are utilized in the integration. The data is written to a FORMAT2000 I4/ℓ = 0 file. This file can then be accessed using QPX.

The distributions provided are not normalized. They contain the actual double differential cross section.

This is the code that is used to generate the I4 files needed to generate the cross section/kerma tables in the series of documents describing the new evaluations by Chadwick. They are used as input by the code I4table.

TeX formated Tables

COMMAND: I4table

USAGE: I4table <atw>

Argument Definition

atw Atomic weight of target in AMU

I4table calculates kerma factors from angle-integrated energy distribution cross section files (I4). It then produces a \LaTeX formated table suitable for publication. The table includes the angle-integrated energy distributions for the various particle types and also gives total production cross sections and partial kerma factors for the different particle types. Also calculated and placed in the table are total elastic and non-elastic kerma as well as the overall total kerma.
Codes

Source Code Version Control

All of the codes in the PCScreate code system are managed under SCCS. Each source directory has an SCCS subdirectory containing the development history of its code. For more information on SCCS, see the man pages: man sccs
Appendix A

Algorithms

Goal

For files generated from ENDL: (eventually I will also process ECPL and EGDL) We want to generate a library containing the following:

- The total cross section for interaction (in any way) vs incident energy (y00c01i000s000). This can be copied directly from ENDL.
- The cross section for elastic scattering of the incident particle (y00c10i000s000) and its angular distribution (y01c10i001s000). These, too, can be directly copied from ENDL.
- The production cross section for other particles produced by the interaction, including inelastically scattered incident particles.

Details of the specific reactions will be lost in the process.

It was decided in a discussion between myself, MBC, DAR and JAR to put the production cross section under a single new C value C=5 (particle emission). This was also discussed with RJH and STP and received their blessing.

Table 1: Production Particle Types to Consider

<table>
<thead>
<tr>
<th>type</th>
<th>yo</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>p</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>3</td>
</tr>
<tr>
<td>t</td>
<td>4</td>
</tr>
<tr>
<td>(^3)He</td>
<td>5</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>6</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>7</td>
</tr>
</tbody>
</table>
The gamma distributions are tabulated separately from the other particle production. The C value used is 55 as in ENDL. This is done because most of the gamma information is isotropic whereas the other particles’ energy distributions depend on angle. The processing codes that generated the binary application files require that each particle type for a given C value be given in the same format.

**Output files**

The following files are generated for each incident particle type on each target isotope.

1) The "I=0" file: non-elastic reaction cross section as a function of incident energy, Einc.

   For each particle type: (Files not present for YO’s that are not produced from a given YI and ZA)

2) The "I=1" file: angular distribution (cos(theta)==CT) as a function of Einc; dP/d(cos(theta)) at Einc. This is a probability distribution normalized for each tabulated Einc with cos(theta) ranging from [-1,1].

3) The "I=3" file: production energy (E’) distribution vs (Einc, cos(theta)). For each tabulated Einc and cos(theta), a normalized probability distribution in E’is given.

4) The "I=9" file: particle multiplicity per non-elastic reaction. (can be less than one) There will be only one I=0 file for C=5 that contains the non-elastic cross section. For each particle type, there will be an I=9 file containing the average number of that type generated per non-elastic reaction.

5) The "I=4" file: (for gamma production only in place of I1,I3) Since MBC is producing isotropic gamma spectra above 20MeV, I am at this time also doing so below 20MeV. No energy distribution information is lost. Four columns. (Einc, E’, I[=0], dP/dE’) dP/dE’ is normalized for each Einc (and each I, if appropriate)

**Input files**

The files used to create the above files for each particle are in the ENDL library:
For each isotope, there is a subdirectory under this path given by
\[za<ZZZ><AAA>\]
where \(<ZZZ><AAA>\) is a 6 digit integer given by \(1000*Z+A\).

For neutron production from ENDL (the first to be considered), the following C values have to be processed if they are present. For other particle production and other input libraries, the list will vary.

**Accumulating the reaction list**

The first step in determining the production spectrum for a given particle will be the creation of a list of the tabulated reactions. Each ENDL ZA subdirectory has a file named "index" that lists the available files for that ZA number. The filenames are organized by ascending C value and alphabetically for a given C (0<1, a<b, etc). Hence, a list of reactions present is determined by a sequential scan of the index while comparing against a list of C values for particle production. By coding the reaction list in the same order, the scan need only be done once to get all reactions present that will produce the current particle.

The C value for a file is a 2 character integer starting in the 6th column of the file name: \(yo<NN>c<NN>i<NNN>s<NNN>\). For a given C value from the list, I scan '.index' until I find it or until the C value in '.index' is larger. Then I advance the reaction list so that the C value is greater than or equal to the current value found in '.index'.

**C\YO list for ENDL**

This table was created by using variations of the command

```bash
ls /nds/data/endl/za* — egrep '(za—c<nn>)' — more
```

for each defined C value in <nn> format. The resulting list of files was scanned for YO values, including residual nuclei YO's (11-16). Particle and residual nuclei
### Table 2: Neutron Producing Reaction Numbers

<table>
<thead>
<tr>
<th>c</th>
<th>reaction</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>(n,n')</td>
<td>2 body reaction; no E' data (determined by kinematics)</td>
</tr>
<tr>
<td>12</td>
<td>(n,2n)</td>
<td>mult. 2X</td>
</tr>
<tr>
<td>13</td>
<td>(n,3n)</td>
<td>mult. 3x</td>
</tr>
<tr>
<td>14</td>
<td>(n,4n)</td>
<td>mult. 4x</td>
</tr>
<tr>
<td>15</td>
<td>(n,fission)</td>
<td>mult. from i=7 (nubar) file</td>
</tr>
<tr>
<td>20</td>
<td>(n,n p)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>(n,p n)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>(n,n d)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>(n,n d α)</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>(n,n t)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>(n,n $^3He$)</td>
<td>neutron from (n,n') in C12: no E' data (determined by kinematics)</td>
</tr>
<tr>
<td>26</td>
<td>(n,n α)</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>(n,n 2α)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>(n,n t α)</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>(n,2n p)</td>
<td>mult. 2x</td>
</tr>
<tr>
<td>30</td>
<td>(n,γ n α)</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>(n,2n p α)</td>
<td>mult. 2x</td>
</tr>
<tr>
<td>32</td>
<td>(n,2n d)</td>
<td>mult. 2x</td>
</tr>
<tr>
<td>33</td>
<td>(n,2n α)</td>
<td>mult. 2x</td>
</tr>
<tr>
<td>34</td>
<td>(n,n p α)</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>(n,d n)</td>
<td></td>
</tr>
</tbody>
</table>
files will be treated separately. No reaction files were found for outgoing electrons or positrons (y0=8,9,10).

The table contains all the utilized C values in ENDL as of 3/1/94. C values that are assigned for specific uses but that are unused are not included in the table.

A numeral in the YO column across from a C value means that files exist in the ENDL library pertaining to production of that particle. If the numeral is >1, it is the explicit multiplication factor: Example, for (n,2n'), there is a '2' under yo=1 across from C=12. There may be multiplicity files for reactions with a numeral '1'. For neutron production (yo=1) from fission (c=15), there is a 'nubar' multiplicity file (I=7). For gamma production (yo=7) there is an 'I=9' file.

Note: This table has been updated to contain all the 'particle producing' C values and hence can be used for ENDL, ECPL, and EGDL as of 5/94.

**Input Data format variations**

In most cases, the specific reactions do not have "I=3" information. Some are not in "I=1" format, either. In this later case, the data will usually be in "I=4,l=0" format which contains the E' and angular distribution data together.

The 'l' descriptor refers to a Legendre polynomial in cos(θ). The 'l=0' coefficient means isotropic emission. In these cases, the probability distribution given, dPe(E';Einc), doesn't depend on the angle, cos(θ). If there was angular dependence, it would be described by higher order l-values. For data of this form, there is no need for an input "I=1" file. To get a uniform normalized probability distribution in cos(θ), I only need to set dPt(cos(θ)) = 0.5 for all value of cos(θ).

Most reactions involving only two particles [E.G. (n,n')] have no E' data given. In these cases, E' is determined by 2-body relativistic kinematics. These reactions have "I=1" data files. It is important to note that the angular distributions ("I=1") are given in the center of mass frame. Dave Resler has provided a routine 'rcmtol.f'(orig. by S.Warshaw; modified by LJC) that will convert from the CM frame to the lab frame. Given the kinetic energy of the incident particle and the angle between the projectile and the ejectile (both in the lab frame), it returns the angle of the ejectile in the CM frame with respect to the incident direction. It also returns the new lab kinetic energy of the ejectile and the scale factor to convert the CM angle probability to the lab frame. (It actually has 8 input values: see the code for a description. A copy
### Table 3: YO occurrences for C values in ENDL, ECPL and EGDL

(Not all C values were found)

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Type</th>
<th>YO</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
<th>C15</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n,n)</td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,n')</td>
<td></td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2n)</td>
<td></td>
<td>12</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,3n)</td>
<td></td>
<td>13</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,4n)</td>
<td></td>
<td>14</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,fission)</td>
<td></td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,np)</td>
<td></td>
<td>20</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,pn)</td>
<td></td>
<td>21</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,nd)</td>
<td></td>
<td>22</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,ndα)</td>
<td></td>
<td>23</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,ut)</td>
<td></td>
<td>24</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,n³He)</td>
<td></td>
<td>25</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,α)</td>
<td></td>
<td>26</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2α)</td>
<td></td>
<td>27</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,ntα)</td>
<td></td>
<td>28</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2np)</td>
<td></td>
<td>29</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,γα)</td>
<td></td>
<td>30</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2npα)</td>
<td></td>
<td>31</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2nd)</td>
<td></td>
<td>32</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2nα)</td>
<td></td>
<td>33</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,nα)</td>
<td></td>
<td>34</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,dn)</td>
<td></td>
<td>35</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2α)</td>
<td></td>
<td>37</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,n³Heα)</td>
<td></td>
<td>38</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,tp)</td>
<td></td>
<td>39</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,p)</td>
<td></td>
<td>40</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,d)</td>
<td></td>
<td>41</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,t)</td>
<td></td>
<td>42</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,tα)</td>
<td></td>
<td>43</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,n³He)</td>
<td></td>
<td>44</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,α)</td>
<td></td>
<td>45</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,γ)</td>
<td></td>
<td>46</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,dα)</td>
<td></td>
<td>47</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,pa)</td>
<td></td>
<td>48</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,2pa)</td>
<td></td>
<td>49</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(yi,Xγ)</td>
<td></td>
<td>55</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
has been put in src/SCCS.) I have modified it to also check that the incident energy is above the backscatter threshold. If the first tabulated \( E_{\text{inc}} \) is below the backscatter threshold, it shifts the energy upwards to allow complete angular distributions. This maximum shift seen on the biologically important nuclei has been less than a few ten’s of keV.

**Contents of the output files**

The "I=0" data file will contain the non-elastic cross section, \( Sp(E_{\text{inc}}) \). This quantity is defined as the sum of the various reaction cross sections at \( E_{\text{inc}}, Si(E_{\text{inc}}) \):

\[
Sp(E_{\text{inc}}) = \text{SUM}( Si(E_{\text{inc}}) )
\]

The "I=9" data file will contain the average multiplicity of the produced particle per non-elastic reaction, \( Mp(E_{\text{inc}}) \), defined as:

\[
Mp(E_{\text{inc}}) = \frac{\text{SUM}( Mi(E_{\text{inc}})\times Si(E_{\text{inc}}) )}{Sp(E_{\text{inc}})}
\]

where \( Mi(E_{\text{inc}}) \) is the multiplicity for particle production for reaction ‘i’. In most cases, \( Mi(E_{\text{inc}}) \) is constant.

The "I=1" data file will contain the unit-probability angular distribution of the particle production spectrum in the laboratory frame of reference, \( dP_t(cos(\theta);E_{\text{inc}}) \), defined as follows:

\[
dP_t(cos(\theta);E_{\text{inc}}) = \frac{\text{SUM}( Mi(E_{\text{inc}})\times Si(E_{\text{inc}}) \times dP_d(cos(\theta);E_{\text{inc}}) )}{Sp(E_{\text{inc}})}
\]

where \( dP_d(cos(\theta);E_{\text{inc}}) \) is the differential probability of emission at \( cos(\theta)(\text{lab}) \) for the incident energy \( (E_{\text{inc}}) \) for reaction ‘i’.

The "I=3" data file will contain the unit probability energy distribution of the production spectrum, \( dP_e(E';E_{\text{inc}},cos(\theta)) \) defined as follows

\[
dP_e(E';E_{\text{inc}},cos(\theta)) = \frac{\text{SUM}( Mi(E_{\text{inc}})\times Si(E_{\text{inc}}) \times dP_d(cos(\theta);E_{\text{inc}}) \times dP_e(E';E_{\text{inc}},cos(\theta)) )}{(Sp(E_{\text{inc}})\times dP_d(cos(\theta);E_{\text{inc}}))}
\]

where \( dP_e(E';E_{\text{inc}},cos(\theta)) \) is the differential probability of emission at energy \( E' \) for reaction ‘i’ given the incident energy and emission angle, \( (E_{\text{inc}},cos(\theta)) \).
APPENDIX A ALGORITHMS

The "I=4" data file (for gammas, in place of the I=3 files) will contain the unit probability energy distributions, \( dP(E';E_{inc}) \) defined as follows:

\[
dP(E';E_{inc}) = \frac{\text{SUM}( \text{Mi}(E_{inc}) \times \text{Si}(E_{inc}) \times dP_{i}(E';E_{inc}) )}{\text{Sp}(E_{inc})}
\]

**One Algorithm for accumulating these values**

With the definitions given above for the file contents, a simple nested loop could calculate all the quantities after the data is loaded. In the following conceptual coding, 'n' refers to \( E_{inc}(n) \), 'm' to \( \cos(\theta)(m) \), 'k' to \( E'(k) \) where each is chosen from a master list made from the values contained in the available reaction cross section files. Also, the notation \( \text{VARindex} \) refers the complete master list of values for \( \text{VAR} \).

\[
\text{loop on (E_{inc}n) loop on (\cos(\theta)m) loop on (E'k)}
\]

\[
\text{Sp}(n) = 0.0 \quad dP(m;n) = 0.0 \quad dP(k;n,m) = 0.0
\]

\[
\text{loop on (reactions i) Sp}(n) = \text{Sp}(n) + \text{Mi}(n) \times \text{Si}(n) \times dP_{i}(m) ;
\]

\[
dP(k;n,m) = dP(k;n,m) + \text{Mi}(n) \times \text{Si}(n) \times dP_{i}(m) ;
\]

\[
\text{dP}(m;n) = dP(m;n) / \text{Sp}(n) \quad dP(k;n,m) = dP(k;n,m) / ( dP(m;n) \times \text{Sp}(n) )
\]

\[
\text{write('I=0') } E_{inc}(n), \text{ Sp}(n) \text{ write('I=1') } E_{inc}(n), \cos(\theta)(m), \text{ dP}(m;n) \text{ write('I=3') } E_{inc}(n), \cos(\theta)(m), E'(k), \text{ dP}(k;n,m)
\]

\[
\text{endloop on E' endloop on cos(\theta) endloop on E_{inc}}
\]

Drawbacks of this technique: 1) All reaction data must be in memory at the same time. 2) Master lists of the incident energies, angles, and exit energies must be compiled from all three data formats.

**Master lists of coordinate values**

Before the loop described above can be executed, the master lists of \( E_{inc} \), \( \cos(\theta) \), and \( E' \) values will have to tabulated. The obvious method for producing these lists is to use all values that occur in any of the reaction files present. This may be a large list, however. The choice may have to be thinned, either before or after processing. In order to preserve fine details, I should probably use as complete a list as possible.
during accumulation and do the data thinning later. As of 7/94 no thinning is being done at all.

**Another Algorithm for accumulating these values**

The main idea behind this technique is to process one reaction at a time and to keep a running sum. After each reaction, "add" it to the previous total with weights from the reaction and running sum multiplicities and cross sections. After each addition, the I1 and I3 or I4 arrays are renormalized.

The identification of the reactions can utilize the '.index' file found in each ZA directory of ENDL and the C\YO table given above. For each reaction found in the index, I can check the C\YO table to see if it contributes to the production of the YO particle.

For each reaction, I will need to calculate weighted "I=1" and "I=3" data. For I=1, the usual probability distribution in \(\cos(\theta)\) will need to be weighted by the multiplicity and the cross section (lab frame). For I=3, the usual probability distribution in \(E'\) will need to be weighted by the weighted I=1 distribution: OR directly by the multiplicity, the cross section, and the probability distribution in \(\cos(\theta)\).

Summing the reactions will require padding each reaction array to contain all the points in the other. This will require 1, 2, and 3D interpolation. The I=0 files can be linearly interpolated in Einc to fill in missing points in each array. The I=1 files can also be linear interpolated in both Einc and CT, since the endpoints of CT don't move. However, interpolation in the I=3 file will require the use of the UBT (unit based interpolation) along the E' axis only: the other two axes can be treated using simple linear interpolation.

**Two body reactions**

For two body reactions such as (n,n) or (n,n'), the only data available will be the I=1 (integrated reaction cross section for incident energies given in the lab frame) and I=3CM (angular distribution in CM system for incident energies given in the lab frame). To get the angular distribution and E' distribution in the lab frame requires a little work. (The I=0 file will actually be unchanged because the multiplicity for a two body reaction is "1".)
APPENDIX A ALGORITHMS

Making I=(0,1,3)(LAB FRAME) from ENDF I=(0,1)(CM FRAME)

Step 1) Obtain a list of incident energies from I=1CM: E(n)—n=1,N

Step 2) Define a list of angles desired in the lab system. Since the cos(θ) axis in the CM frame is distorted when converted to the lab frame, some minimum granularity will be required for accuracy. To start, I will use step of 0.1 in cos(θ). cos(θ)(m)—m=1,M

Step 3) Define an energy resolution for E'. For a given E(n) and cos(θ)(m), there will be a specific E'due to relativistic kinematics. We need to define a dE' with which to make a normalized energy distribution.

Step 4) Make new I=1 and I=3 files.

Loop on E(n), n=(1,N) : incident energies
Loop on cos(θ)(m), m=(1,M) : lab angles

convert angle to CM frame and get E' and probability scale: call rcmtol(argsin, argsout) argsin = ( Myi amu of yi Mtarget amu of target nucleus Myo amu of yo Mrn amu of residual nucleus Exrn excitation level of residual E(n) kinetic energy of yi cos(θ)(m) cosine of yo in lab frame ) argsout = ( Eshift the maximum of Einc and the backscatter threshold cos(θCM) cosine of yo in CM frame csltoc scale factor for probability to be at CTCM to get probability to be at cos(θ)(m) in lab frame E'(n,m) energy of yo in lab frame )

get dPt(E(n),cos(θ)CM) from I=1 (lin. interp. on cos(θ))

calculate lab frame normalized angular distribution: dPt(Eshift,cos(θ)(m)) = dPt(Eshift,cos(θ)CM) / csltoc

write(I=1): Eshift, cos(θ)(m), dPt(Eshift,cos(θ)(m))

write the normalized triangular energy distribution: write(I=3 weighted) Eshift, cos(θ)(m), E'-dE', 0.0 write(I=3 weighted) Eshift, cos(θ)(m), E', (1/dE') write(I=3 weighted) Eshift, cos(θ)(m), E'+dE', 0.0 NOTE: Care must be taken that E'-dE' is not less than zero. To this end, the lower end of the delta function is set by max[0.0,E'-dE'], and the height is set by max[1/dE', 2/(E'+ dE')] end Loop: m end Loop: n

check the normalization for each E(n) (see below)
finished with this reaction:

The $I=1$ array contains normalized angular distributions at the chosen lab angles for each $E_{\text{inc}}$ in the original $I=1$ data. The $I=3$ array contains normalized $E'$ distributions for each cosine at each $E_{\text{inc}}$.

**Renormalizing the probability distributions: $(I=0,1)\text{CM}$**

After converting the angular probability distribution to the lab frame, it may not be properly normalized due to the coordinate changes. Hence, it should be renormalized before it is added to the running sum.

**Multi-body reactions with $I=4(l=0)$ data (no $I=1$ data)**

Most multi-body reactions tabulated in ENDL have angular distribution energy given in the form of $I=4$ data with $l=0$ (isotropic). The discussion that follows assume $l=0$. For purposes of creating the PCSL, these data have to be turned into the $I=1,I=3$ form. In the two body case described above, the $I=0$ data didn’t have to be changed because the multiplicity was "1". In the current problem, there is a multiplicity factor that needs to be included: it will be "1" in most cases. The multiplicity may be implicit in the reaction—ex. $(n,2n)$—or it may be explicitly given in an additional data file. Possible additional data types are $I=7(\text{nubar})$ [for fission $(\gamma_0 = 1, c=15)$ only] and $I=9(\text{multiplicities})$ [for gamma $(\gamma_0=7)$ production only]. All other multiplicities are implicit in the reaction.

**Making $I=(0,1,3)$ from ENDL $I=(0,4)$**

Step 1) Obtain a list of incident energies from $I=4$: $E(n)\rightarrow n=1,N$

Step 2) Define a list of angles. If $l=0$, the distribution is isotropic. Therefore, I only need the endpoints of the $\cos(\theta)$ axis. $M=3$: $\cos(\theta)(1)=-1; \cos(\theta)(2)=0.0; \cos(\theta)(3)=1$

\[
dP_t(E,\cos(\theta)) = 0.5 \text{ (for all values of } E, \cos(\theta))\]
APPENDIX A ALGORITHMS

Step 3) For each incident energy, there is a list of E' values in the I=4 file. As each energy is processed, this list needs to be defined. It can be overwritten with the next set of E' values for each subsequent E(n). at n: E'(k)—k=1,K(n)

Step 4) Determine the YO multiplicity for this reaction: If yo=1 and c=15, look for an I=7(nubar) file; If yo=7, look for an I=9(mult.) file. Otherwise, take the multiplicity from the C\YO table entry as a constant multiplicity.

Step 5) Make the new I=0, I=1 and I=3 files.

Loop on E(n), n=(1,N)

get cross section S(E(n)) from I=1 (lin. interp. on Ein)

scale by multiplicity to get the weighted cross section S'(E(n)): get Mult(E(n)) from the I=7,9 file for E(n), if appropriate:

S'(E(n)) = S(E(n)) * Mult(E(n))

Loop on \cos(\theta)(m), m=(1,M=3) Loop on E'(k), k=(1,K(n))

determine the value to write to the I=1 file: dPt(E(n)) = 0.5 : no dependence \cos(\theta)

look up the probability for yo emission at E'(k) for yi at E(n) from the I=4 file: dPe(E(n),E'(k)):

write(I=3) E(n), \cos(\theta)(m), E'(k), dPe(E(n),E'(k))

end Loop: k

write(I=1) E(n), \cos(\theta)(m), dPt(E(n))

end Loop: m

write(I=0 weighted) E(n), S'(E(n))

end Loop: n

finished with this reaction

These reactions should not require renormalization since we are explicitly creating normalized areas.
APPENDIX A ALGORITHMS

S=1 switch special case

Excitation levels; More than one distribution per file:

For both ENDL I=(0,1) and ENDL I=(0,4) defined reactions, there are cases where S=1. The meaning of this switch is to mark the presence of one or more excitation levels of the residual nucleus. The excitation energy is given in the second line of the header as X1. (See /nds/misc/read.ENDL.format). For the cases with more than one level, each level can be treated as a completely separate reaction. There should be blocks for each given energy level in both files: either I=(0,1) or I=(0,4). Hence, for the special case of S=1, I must be alert for the presence of more than one 'reaction' in a single file set. For I=(0,1) type (2 body) reactions, the excitation level needs to be passed to the routine 'rcmtol' in the 'argsin' list. (See discussion above on ENDL I=(0,1) reactions.)

Other special cases

There is exactly one case where there is I=(0,1,3) data in the ENDL database. That is for ZA004009 (be-9) with the (n,2n) reaction. Is this data given in the lab frame or the CM frame? We will probably not need to process this data into production form for biological applications.

There is also one case where there is I=(0,4)l(>0) data in ENDL. That is for the d(n,2n)p reaction. In this case, the Legendre polynomial distributions would have to be converted to angular probability distributions. The information in the *i004* file is apparently given in normalized form for each (E(n),E'(k),l=0) set. The information for other l-values is not normalized to unitly. Are the different distribution to be weighted by their relative normalization and the result re-normalized or is there some implicit multiplicity here? There are also occurrences of l>0 in EGDGL, but not in particle production data files.

There is another special case for gamma production where the data is given in I=1 format with S=3. The switch (S) of 3 means that the first X-field contains the discrete gamma energy. The I=1 data is the angular distribution of this gamma. This case is handled by assuming that the angular distribution is given in the lab system and the photon energy in the lab frame is independent of angle and equals the value found in the X1 field. These files can potentially contain more than one gamma line per file. In this event, the different data blocks are treated as separate contributions as
in the case of S=1 described above. Since we are treating gammas as isotropic at this time, data of this type is converted to an I4 format with three points for each incident energy defining a triangular delta function of specified width.

**Adding two I=(0,[1,3] or [4]) sets**

_1=0_

1) Pick a common set of E-values from the two arrays: \( E(n) = E1(n) \cup E2(n) \)

2) Padd the two arrays to contain all of the set \( E_f \) using linear interpolation in \( E \).

4) Add the \( S1(E1) \) and \( S2(E2) \) values directly to get the summed \( S(E) \) distribution.

_1=1_

1) Pick a common set of E-values from the two arrays: \( E(n) = E1(n) \cup E2(n) \)

2) For each array, add the missing E-values and fill the \( \cos(\theta) \) distributions for the new E-values. Since the adjacent tabulated E-values may not have the same \( \cos(\theta) \) points, pick the \( \cos(\theta) \)-values for the new E-value from those common to the two adjacent energies. Then, using double linear interpolation, calculate the weighted distribution at the new E-value for each of those \( \cos(\theta) \)-values.

3) At this point the two arrays to add have the same tabulated incident energy values but, for each energy, may not have the same tabulated \( \cos(\theta) \)-values. So, for each energy, pick a set of common \( \cos(\theta) \)-values from the two arrays and padd the arrays with the missing \( \cos(\theta) \)-values using linear interpolation in \( \cos(\theta) \). The two arrays will then have the same coordinate sets in both \( E \) and \( \cos(\theta) \).

4) Scale the two arrays by the multiplicity weighted cross sections linearly interpolated from associated \( I=0 \) arrays.

5) Add the two padded and scaled arrays point by point to get the new summed \( dP(E,\cos(\theta)) \) distribution.

6) Renormalize each \( (E(n),\cos(\theta)) \) distribution in the sum array.
NOTE: This process is actually done fully for each element of E(n) before proceeding to the next element to conserve memory.

I=3

1) Pick a common set of E-values from the two arrays: E(n) = E1(n) U E2(n)

2) For each array (separately), the missing E-values with their energy distributions at appropriate cos(θ) values need to be filled in. For each new E-value, E(n): a) Form a list of the cos(θ) values found under the two existing energies bracketing E(n). b) For the two bracketing incident energies, temporarily insert the E' distributions at the missing cos(θ) values using a UBT interpolation along the cos(θ) axis. c) Using UBT on the E axis, determine the new E' distribution at each cos(θ) value for the new incident energy E(n).

3) At this point, the two arrays have the same E values. However, they may not have the same set of cos(θ) values for matching E values, however. For each energy E(n): a) Form a list of cos(θ) values common to both arrays at E(n). cos(θ)(n) = cos(θ)1(n) U cos(θ)2(n) b) Insert the E' distributions (using UBT along the cos(θ) axis) for the cos(θ) values not present in each array.

4) The two arrays should now match with respect to the cosines at each E(n). Scale each array by the multiplicity weighted cross section (I=1) and the angular probability distribution (I=3).

5) Add the two arrays, (E, cos(θ), E') point by (E, cos(θ), E') point.

6) Renormalize each (Einc, cos(θ), E') distribution in the sum array.

NOTE: This process is actually done fully for each element of E(n) and cos(θ)(n,m) before proceeding to the next element to conserve memory.

I=4

1) Pick a common set of E-values from the two arrays: E(n) = E1(n) U E2(n)

2) In each array, insert the E' distributions (using UBT) for each missing E(n).
APPENDIX A ALGORITHMS

3) For each $E(n)$: a) Form a list of $E'$ values common to both arrays at $E(n)$. $E'(n) = E1'(n) \cup E2'(n)$ b) In each array, fill in the missing points in the distribution using linear interpolation in $E'$.

4) Scale each array by its multiplicity weighted cross section ($I=0$).

5) Now both arrays have the same set of incident energies and the same set of $E'$ energies at each incident energy. Add the two arrays point by point.

6) Renormalize each $(E(n),E')$ distribution in the sum array.

NOTE: This process is actually done fully for each element of $E(n)$ before proceeding to the next element to conserve memory.
Index

Commands
  concat, 6
  edepCM, 8
  edepI3, 9
  edepI4, 9
  I3toI4, 10
  I4table, 11
  PCScreate, 4
  qbar, 9
  qbardrv, 9
  thinI3, 7
  concat, 6
  .file_match file, 6

documentation
  location, 4
  edepCM, 8
  edepI3, 9
  edepI4, 9
  executables
    location, 4
    .file_match file, 6
  I3toI4, 10, 11
  I4table, 11
  .index file, 5
    mixed sources, 5
    ordering, 5
    selection, 5
  Kerma, 11
    elastic, 11
    non-elastic, 11
    partial, 11
    total, 11
  PCScreate, 4

.input data
  directories, 5
  mixed sources, 5
/nads/, 5
  private, 5

PEREGRINE
  sensitivity, 5
  qbar, 9
  qbardrv, 9

SCCS, 12
  source code
    location, 4
    SCCS, 12
    version control, 12

thinI3, 7