

Understanding NJOY

R.E. MacFarlane*

Los Alamos National Laboratory, Los Alamos, USA

*Lectures given at the
Workshop on Nuclear Data and Nuclear Reactors:
Physics, Design and Safety
Trieste, 13 March - 14 April 2000*

LNS015013

*ryxm@lanl.gov

Abstract

The NJOY Nuclear Data Processing System is used to convert evaluated nuclear data in ENDF format into forms useful for applications. As a bridge between physics and engineering, it is best used by people with some knowledge of things like nuclear reaction theory, resonance theory, or scattering theory on one side, and some knowledge of things like particle transport codes, reactor core calculations, or radiation medicine on the other. These lectures provide an introduction you to NJOY system (using version 97.107), give a quick outline of the physics of nuclear data, and show how some of the main applications in nuclear technology link to NJOY results. This material is also available on line, including more details and exercises, at <http://t2.lanl.gov/njoy>.

1 ENDF, NJOY, and Applications

The ENDF formats were originally developed for use in the US national nuclear data files called ENDF/B (the Evaluated Nuclear Data Files). These files went through various versions with names like ENDF/B-VI, each version adding both improved data and new capabilities for representing nuclear data. The current ENDF-6 format can represent cross sections for neutrons, photons, and charged particles, including particle yields and distributions in angle and energy, for energies up to several hundred MeV, the radioactive decay properties of reaction products, and estimated errors and covariances of the various nuclear parameters. The ENDF format is now widely used around the world, including the JEF files in Europe, the JENDL files in Japan, and the BROND files in Russia. Thus, even though NJOY was originally designed to work with the US ENDF/B libraries, it now provides a universal capability to work with nuclear data libraries all over the world. For more information on the ENDF format, see “Introduction to the ENDF Format” elsewhere in this volume, or try the online version at <http://t2.lanl.gov/endl>.

The NJOY Nuclear Data Processing System is a modular computer code designed to read evaluated data in ENDF format, transform the data in various ways, and output the results as libraries designed to be used in various applications. Each module performs a well defined processing task. The modules are essentially independent programs, and they communicate with each other using input and output files, plus a very few common variables.

NJOY directs the flow of data through the other modules and contains a library of common functions and subroutines used by the other modules

RECONR reconstructs pointwise (energy-dependent) cross sections from ENDF resonance parameters and interpolation schemes.

BROADR Doppler broadens and thins pointwise cross sections.

UNRESR computes effective self-shielded pointwise cross sections in the unresolved energy range.

HEATR generates pointwise heat production cross sections (KERMA coefficients) and radiation-damage cross sections.

THERMR produces cross sections and energy-to-energy matrices for free or bound scatterers in the thermal energy range.

GROUPR generates self-shielded multigroup cross sections, group-to-group scattering matrices, photon-production matrices, and charged-particle cross sections from pointwise input.

GAMINR calculates multigroup photoatomic cross sections, KERMA coefficients, and group-to-group photon scattering matrices.

- ERRORR** computes multigroup covariance matrices from ENDF uncertainties.
- COVR** reads the output of ERRORR and performs covariance plotting and output formatting operations.
- MODER** converts ENDF “tapes” back and forth between ASCII format and the special NJOY blocked-binary format.
- DTFR** formats multigroup data for transport codes that accept formats based in the DTF-IV code.
- CCCCR** formats multigroup data for the CCCC standard interface files ISOTXS, BRKOXS, and DLAYXS.
- MATXSR** formats multigroup data for the newer MATXS material cross-section interface file, which works with the TRANSX code to make libraries for many particle transport codes.
- RESXSR** prepares pointwise cross sections in a CCCC-like form for thermal flux calculators.
- ACER** prepares libraries in ACE format for the Los Alamos continuous-energy Monte Carlo code MCNP.
- POWR** prepares libraries for the EPRI-CELL and EPRI-CPM codes.
- WIMSR** prepares libraries for the thermal reactor assembly codes WIMS-D and WIMS-E.
- PLOTR** reads ENDF-format files and prepares plots of cross sections or perspective views of distributions for output using VIEWR.
- VIEWR** takes the output of PLOTR, or special graphics from HEATR, COVR, DTFR, or ACER, and converts the plots into Postscript format for printing or screen display.
- MIXR** is used to combine cross sections into elements or other mixtures, mainly for plotting.
- PURR** generates unresolved-resonance probability tables for use in representing resonance self-shielding effects in the MCNP Monte Carlo code.
- LEAPR** generates ENDF scattering-law files (File 7) for moderator materials in the thermal range. These scattering-law files can be used by THERMR to produce the corresponding cross sections.
- GASPR** generates gas-production cross sections in pointwise format from basic reaction data in an ENDF evaluation. These results can be converted to multigroup form using GROUPT, passed to ACER, or displayed using PLOTR.

Note that some of these modules do calculations to transform the evaluated nuclear data, and others format the results of the calculations for various nuclear applications. One very basic application is the multigroup particle transport code, which is used to compute neutron and photon fluxes and reaction rates for reactor design, shielding and radiation protection, criticality safety, experimental facility design, medical applications, and so on. Familiar examples of such codes include ANISN from Oak Ridge and the “DANT” series from Los Alamos. NJOY can prepare data for these codes through several paths, including DTFR, CCCCR, and MATXSR. These days, people are making more and more use of the powerful Monte Carlo method, which uses very detailed and faithful representations of complex problems. Data for the popular MCNP code can be produced with ACER. For direct thermal-reactor core calculations, the WIMS code series from Great Britain is very widely used, and NJOY’s WIMSR module can produce the appropriate libraries for it. The modular structure of NJOY lends itself to adding additional applications without having to reinvent the core transformations that NJOY does to evaluated nuclear data.

1.1 NJOY Data Flow

NJOY uses a data-flow model to process evaluated nuclear data. Each module reads some data from one or more input files, transforms it, and writes the results on one or more output files. The order that the modules are used and the particular input and output files used are specified as in the following example:

```
[copy an ENDF file onto "tape20"]
reconr
20 21
...specific input for RECONR...
broadr
20 21 22
...specific input for BROADR...
groupr
20 22 0 23
...specific input for GROUPE...
dftr
23 24 22 25
...specific input for DTFR...
stop
[copy tape24 to the DTF library]
```

In this case, RECONR reads the original ENDF tape and reconstructs the resonances and nonlinear interpolation schemes to prepare a PENDF (pointwise ENDF) tape on unit 21. BROADR reads the PENDF tape, Doppler broadens the pointwise cross sections, and writes a new temperature-dependent PENDF tape on unit 22. Sometimes it needs some data from the ENDF tape, and this is why `tape20` is also provided as input. Next, the information from the ENDF and PENDF tapes is run

through the multigroup averaging process and written in GENDF format (a special groupwise variation of the ENDF format) on `tape23`. Finally, the multigroup constants on `tape23` are converted to DTF format on `tape24`. The PENDF input for DTFR is used for the plots, which appear on `tape25`.

All NJOY jobs are variations of this basic data-flow structure. I/O units with numbers from 20 up can be used to couple the modules together, for the original input files, or for output files. NJOY always names these “tape20”, “tape21”, and so on. Units from 10 to 19 are used as scratch files. They may be named as tapes, or have special system names. Units with numbers below 10 are used for system input and output files, such as “output”.

NJOY is normally run under the control of some kind of system shell. For unix, the NJOY input commands go into the standard input, and runtime messages go out on the standard output. Therefore, if the input “deck” (such as the one outlined above) is named “inx”, the following would cause a normal NJOY run to execute:

```
njoy < inx
```

The short listing containing progress messages, notes, and error messages would appear on the screen (or it could be redirected to a file, *e.g.*, `outx`), and the full listing would appear on a file named “output”.

Please don't be shocked at our pleasure in using the terms “tape” and “deck.” Later, you will find us talking about input “cards.” We do this out of respect for the history of computing, and because we can still remember using tapes and decks of cards!

1.2 PENDF and GENDF

The data-flow example above introduces the terms “PENDF tape” and “GENDF tape.” Because these files will be very important for all the NJOY examples, we define them more completely here:

PENDF tape: a pointwise version of an ENDF tape where any resonance parameters or nonlinear interpolation ranges have been converted into pointwise cross sections suitable for linear interpolation to within specified accuracy criteria. The File 1 comments have been simplified, although the directory is correct. Resonance parameters have been removed from File 2. Only Files 3, 6 (for thermal data only), and 13 are given for data; the other information, such as angle and energy distributions, must be read from the corresponding ENDF tape. Multiple temperatures may be given for a material by repeating each “MAT” in order of increasing temperature.

GENDF tape: a multigroup version of an ENDF tape with the group structures, temperatures, and background cross section defined in File 1, “vector” cross

sections given in File 3, and group-to-group matrices given in File 6. Higher MF Files are used for photons and charged particles.

Using these standardized files supports the modular nature of NJOY. Different modules can sometimes be run in different orders and combined in various ways. Internally, all NJOY modules use a common set of subroutines for reading and writing the ENDF, PENDF, and GENDF “tapes” (they are located in the NJOY module).

1.3 Formatted and Binary Modes

ENDF tapes received from a distribution point, be they ENDF/B, JEF, JENDL, CENDL, or BROND, come in a formatted ASCII mode that can be easily read or printed. However, this is not the most efficient form to use for communicating between different processing modules in NJOY. Binary files are much more efficient because it is not necessary to repeatedly convert the data between the binary forms used in memory and the ASCII forms.

Although the ENDF format specifies a standard binary mode, NJOY uses a special blocked-binary mode that divides each ENDF “record” into one or more blocks of data of bounded size. The NJOY subroutines that read and write ENDF records know how to handle the binary mode and how to convert back and forth between binary and ASCII. NJOY normally uses a page size of 326 words for these binary blocks, which is small enough to use conveniently and large enough to reduce the total number of I/O operations to a reasonable value.

NJOY contains a special module called MODER for converting back and forth between the ASCII and binary forms. It uses negative unit numbers to indicate binary files. Therefore, the previous example of NJOY data flow would like like this when using binary mode:

```
[copy an ENDF file onto "tape20"]
moder
20 -21
reconr
-21 -22
...specific input for RECONR...
broadr
-21 -22 -23
...specific input for BROADR...
groupr
-21 -23 0 -24
...specific input for GROUPE...
dftr
-24 25 -23 26
...specific input for DTFR...
stop
[copy tape25 to the DTF library]
```

MODER also has other capabilities. See the MODER Input Instructions in the online version of this tutorial, the `userin` file in the NJOY distribution, or a printed NJOY manual.

1.4 NJOY Input

Beginning with NJOY97, the standard Fortran “read *” method is used to read input quantities and strings. Earlier versions used a home-grown scheme with similar properties. The main difference between these two approaches is that the new one delimits string data with single quote characters, but the old one used star characters.

All input is free form, and different quantities can be separated with spaces or commas. Numerical quantities can be given with or without decimals, and “E” or “e” can be used for scientific notation. String input consisting of single words that start with an alphabetic character do not require delimiters. Longer strings with spaces and special characters must be surrounded by single quote characters.

Lines of input for NJOY are often organized so that the parameters at the ends of the lines have default values. Therefore, you can often stop giving numbers before the entire I/O list is exhausted by using the slash (/), or division sign, character. Some examples follow:

```

1   reconr
2   20 21
3   1395 1/
4   .01/
5   '92-U-235' /
```

Line 1 is a string value that doesn’t require delimiters. Line 2 is a line with only two values in the I/O list, and both are given. No terminating / is required. Line 3 is a line with three items in the I/O list. The last variable, namely `ngrid`, will take its default value of 0. Line 4 has 4 values, and the last three will all be defaulted. The value .01 could also be written as 1e-2 or 1.0e-2. Line 5 has only one value, which is a string. Since the string has spaces and starts with a digit, it has to be delimited with quote characters. Any line that consists of string data must end with the / character. The NJOY97 test problems provide numerous examples of the input syntax.

2 RECONR: Linearization and Resonance Reconstruction

Most NJOY processing sequences start with RECONR. It fills two roles. First, it goes through all the reactions included on the ENDF tape and chooses a union grid that allows all cross sections to be represented using linear interpolation to

a specified accuracy. This step removes any nonlinear interpolation ranges (*e.g.*, log-log, linear-log). It also makes it possible for all summation reactions to be reconstructed as the sum of their parts (*e.g.*, total, total inelastic, total fission). Second, for resonance materials, it reconstructs the resonance cross sections (elastic, fission, capture) on a union grid that allows them all to be represented within certain accuracy criteria, and then combines the resonance data with the other linearized and unionized cross sections. The results are written in PENDF format.

2.1 Linearization and Unionization

All ENDF-format evaluations define their cross sections over at least part of the energy range using tabulations. In many cases, part of some of these tabulations specify that a nonlinear interpolation scheme is to be used to obtain intermediate values. NJOY wants to get everything in linear form for several reasons:

- the method used for Doppler broadening requires linearly interpolated piecewise functions;
- the ACE files for use with MCNP require linear-linear functions;
- linear functions are easy to average into multigroup form;
- linear functions are easy to plot; and
- summation cross sections, such as the total cross section, total inelastic, or total fission, cannot be properly defined as the sum of their parts except when given as linear.

The classic example of the last point is the thermal energy range, where the zero-degree elastic cross section is usually constant with energy (and uses linear-linear interpolation), but the radiative capture cross section follows a $1/v$ law and uses log-log interpolation. There is no simple “graph-paper” interpolation scheme that can represent the sum of the elastic and capture cross sections to the same accuracy as the original parts. However, if both the elastic and the capture are linearized to a given accuracy on the same “union grid,” their sum is also linearized to the same accuracy on this grid.

RECONR solves this problem as follows. It first skips the total cross section and goes to the elastic cross section (MF=3, MT=2). It starts with a list of “nodes” that may include energy values given by the user, may include energies derived from the resonance parameters, and certainly includes the thermal value .0253 eV. It then adds in any energy points given in the elastic section of File 3, and if necessary, it adds in points on the 1, 2, 5, 10, ... grid for each energy decade. Now it looks at each “panel” described by a pair of these nodes and decides whether to divide the panel in half. It will divide the panel if the energy step is too large to represent the $1/v$ function to within the specified accuracy, or if the interpolation scheme is nonlinear,

it will divide the panel if the difference between the specified interpolated function and the linear interpolate is too large. It continues subdividing and checking until the desired accuracy has been achieved over the entire energy range.

The reason for checking the $1/v$ accuracy for elastic scattering is that Doppler broadening tends to add a $1/v$ component to the elastic cross section, and this grid will help to handle that correctly in BROADR, even when the radiative capture reaction is missing or extraordinarily small.

Once the elastic cross section has been linearized, the code moves on to the next reaction, perhaps (n,2n), and determines whether any additional points have to be inserted in the union grid to represent that reaction to within the desired tolerance. RECONR continues in this way until all the reactions have been checked. The result is therefore a union grid able to represent all the cross sections with linear interpolation, and that implies that it can also represent the sums of those cross sections with the same accuracy.

The following input deck will carry out the linearization and unionization process for H-1 from ENDF/V-VI (assuming that the appropriate ENDF file has been mounted on unit 20):

```
reconr
20 21
'1-H-1 from ENDF/B-VI'/
125 1/
.01/ 1% linearization
'1-H-1 from ENDF/B-VI'/
0/
stop
```

For a complete description of these input lines, see the online RECONR Input Instructions, a printed NJOY manual, the “User Input” file, from the NJOY distribution, or the comment cards at the beginning of the RECONR module in the NJOY source listing. Briefly, this deck says, read from an ENDF file on tape20 and write the PENDF file to tape21 with a new title. Work on MAT 125, use 1% reconstruction, and provide one new line of text for the File 1 comments on the output PENDF file.

2.2 Resonance Reconstruction

ENDF-format files for the lighter nuclides usually represent the entire cross section using tabulated values. However, as the target mass increases, more and more resonances occur for reactions like elastic scattering, radiative capture, and fission. The number of energy grid points required to represent the function in tabulated form becomes excessive. At this point, the evaluators start to use resonance parameters to represent all or part of the cross section over a range of energies called the “resonance range.”

The resonance range may consist entirely of resolved resonances, or there may be both resolved and unresolved ranges. We will discuss the unresolved part later. The resolved range usually starts at $1e-5$ eV, or some other low energy (such as 1 eV) and extends until the resonance effects are small, or until a changeover to the unresolved representation occurs. The resonance parameters are given in File 2 with MT=151. For a more detailed description of the ENDF resonance formats, see our “Introduction to the ENDF Formats” elsewhere in this volume or online. The cross sections in File 3 over the resolved range are usually zero, but in some cases, there may be small “background” cross sections that are to be added to the cross sections computed from the resonance parameters to correct for problems (such as the inadequacy of a single-level Breit-Wigner representation). ENDF provides discontinuities represented as double energy points at the ends of the resonance ranges. These regions of background cross section often confuse people who look at File 3 in ENDF evaluations and ask, “Why is the cross section zero? That can’t be right!”

The ENDF format provides several options for representing resonance parameters. The most commonly used are

Single-Level Breit-Wigner - simple and easy to use, but not always as accurate as needed (can produce negative cross sections),

Multi-Level Breit-Wigner - gets rid of the negative cross sections, but much more expensive to use. Can’t represent multichannel effects that are sometimes important.

Reich-Moore - very faithful for all but the lightest nuclides. It is becoming the preferred representation for important materials.

Except for run time and negative cross sections, the representation used is transparent to the RECONR user.

RECONR reads in the resonance parameters, and it adds the resonance energies and the half-height energies to the list of nodes available from the linearization and unionization process. The resonance nodes are truncated to five significant digits. This makes it easier to find them on a PENDF listing, and it prevents them from being thinned away in BROADR. This choice of starting nodes for the resonance reconstruction process assures that no important structures will be missed. The code then works through each panel defined by these nodes, compares the cross section computed at the midpoint with a linearly interpolated value, and decides whether to add the new midpoint to the grid. This process of halving and checking continues until convergence is achieved over the entire energy range using a special inverted-stack logic that is also used elsewhere in NJOY. Once the resonance energy grid and the resonance cross sections have been determined, RECONR merges them with the linearized and unionized ENDF cross sections, and write the results on an output file in PENDF format.

2.3 Integral Thinning

The criteria used for deciding when to stop halving intervals during resonance reconstruction are complex. Modern resonance evaluations can be very detailed, yielding hundreds of thousands of energy points when fully expanded. These very large tabulations are expensive to produce, to store, and to Doppler broaden or average. However, almost all of the uses of RECONR output result in integrating over the cross sections; this is true for both Doppler broadening and multigroup averaging, and it is also true for all ultimate applications, even continuous-energy Monte Carlo transport calculations. Therefore, if the reconstructed resonance cross sections contain features that cause negligible effect on these integrals, these features can be removed without affecting the ultimate results significantly. This is why RECONR includes an option for carrying out “integral thinning” while it is generating its resonance cross sections.

The main user input to RECONR is the reconstruction tolerance, `ERR`. It is given as a fractional quantity, *e.g.*, 0.001, to represent 0.1%. The default behavior is to set the auxiliary thinning parameters to standard values `ERRMAX=20*ERR` and `ERRINT=ERR/10000`. RECONR will then always halve intervals until the fractional difference between the computed cross sections and the interpolates are less than `ERRMAX` for all reactions. Once this has been achieved for an interval, it will check to see how much the panel will change the resonance integral. If the change is smaller than `ERRINT`, the subdivision process will be terminated. When the subdivision terminates, it is possible to make a rough estimate of the effect of the termination on the resonance integral. As an example of this process, here is an example of the output provided by RECONR when processing U-238 from ENDF/B-VI:

```

number of user and resonance nodes = 3834
points in initial unionized grid = 4230
points added by linearization = 215

      estimated maximum error due to
      resonance integral check (errmax,errint)
      and significant figure truncation

      upper      elastic      percent error      capture      percent error
      energy     integral     res-int sig-fig     integral     res-int sig-fig
1.00E-05
1.00E-04      2.15E+01      0.000      0.000      1.86E+02      0.000      0.000
1.00E-03      2.15E+01      0.000      0.000      5.88E+01      0.000      0.000
1.00E-02      2.15E+01      0.000      0.000      1.86E+01      0.000      0.000
1.00E-01      2.15E+01      0.000      0.000      5.91E+00      0.000      0.000
1.00E+00      2.14E+01      0.000      0.000      1.96E+00      0.000      0.000
2.00E+00      6.32E+00      0.000      0.000      3.32E-01      0.000      0.000
5.00E+00      7.90E+00      0.000      0.000      6.29E-01      0.000      0.000
1.00E+01      1.47E+01      0.000      0.000      1.30E+02      0.000      0.000
2.00E+01      5.27E+00      0.000      0.000      6.59E-01      0.000      0.000
5.00E+01      1.02E+02      0.000      0.000      1.09E+02      0.000      0.000
1.00E+02      1.89E+01      0.000      0.000      1.26E+01      0.000      0.000
2.00E+02      5.05E+01      0.000      0.000      1.34E+01      0.000      0.000

```

5.00E+02	1.93E+01	0.000	0.000	5.14E+00	0.004	0.000
1.00E+03	1.50E+01	0.001	0.000	2.45E+00	0.014	0.000
2.00E+03	1.47E+01	0.002	0.000	1.33E+00	0.042	0.000
5.00E+03	1.77E+01	0.006	0.000	1.09E+00	0.113	0.000

```

points added by resonance reconstruction =140468
points affected by resonance integral check =103592
points affected by significant figure reduction =      0
final number of resonance points =144493
number of points in final unionized grid =144913

```

Note that the output shows the elastic and capture integrals over roughly logarithmic energy bands running through the resolved range (1e-5 eV to 5 keV). The “res-int” column shows an estimate of the error made by integral thinning, *e.g.*, a maximum of 0.1% for U-238 capture at high energies. About 100000 points were removed from the final energy grid by the integral check. This is a major savings for a fairly small impact on the least known part of the U-238 cross section. If this increased error is unacceptable, it can be reduced in two ways: either set `ERRMAX` closer to `ERR`, or make `ERRINT` smaller than `ERR/10000`.

2.4 Numerical Precision

Standard ENDF-format files use an 11-column field to represent floating-point numbers. In order to maximize the number of significant digits possible within this limit, unnecessary characters are eliminated. Therefore, energies usually take on forms like “1.234567+3”, which provides for 7 significant digits. If larger exponents are needed, the format drops back to 6 or 5 significant digits. One of the few places that this standard numerical format can be inadequate is for cross sections reconstructed for very narrow resonances. Since narrow resonances only occur for a restricted range of energies (say, 1 eV to 1 MeV), 11-column forms such as “123.456789” can provide an accuracy of 9 significant digits, which is always sufficient in practice.

However, the need for more than 7 significant digits is rare. Moreover, the smoothing effect of Doppler broadening usually assures that only 7 digits are needed at 300K, even if 11 are needed at 0K. In addition, we want to preserve as much similarity between PENDF tapes and ENDF tapes as possible. For these reasons, we have adopted a complex system for the precision of energy values used in NJOY97. The initial nodes, the values read from standard ENDF tapes, and values added by RECONR (such as those from the 1, 2, 5, 10, 20, ... sequence) will normally have only a few significant digits (typically 5 or fewer). This helps to identify them on listings, and BROADR will try not to thin out energy values with several zeros on the end. In subdividing energy panels to linearize resonance cross sections, RECONR will first try to fit the actual function with numbers limited to the 7-digit standard form. Only when this is impossible will RECONR resort to the 11-digit form. It is fairly easy to spot such numbers on a listing of the 0K PENDF tape.

In BROADR, the code will preferentially use the numbers with lower numbers of significant digits as nodes in its reconstruction process. Thus, if 11-digit numbers are not needed to represent the energies for the Doppler-broadened cross section, the PENDF tape from BROADR will contain only the ENDF-standard 7-digit form, and our goal of maximum compatibility with the appearance of standard ENDF tapes will have been achieved. However, if the more precise 11-digit values are needed to get the specified accuracy, they will be used. All cross sections are adjusted to fit into the normal ENDF 7 significant-figure format.

NJOY97 binary ENDF tapes use 64-bit representations for all numbers in order to support the scheme described above. The number of significant digits in the binary versions of each number matches the number in the ASCII versions, which helps to preserve consistency between NJOY runs made in binary and ASCII modes.

2.5 Unresolved Range

For the heavier nuclei, there is a region above the resolved range where the resonances get so close together that they can no longer be separated. However, there can still be important resonance effects, such as self shielding, active in this “unresolved” range. In ENDF-format evaluations, the unresolved range is represented by giving average resonance widths and spacings and the statistical rules obeyed by these widths and spacings. RECONR converts these statistical representations into actual “infinitely-dilute” cross sections. These are the values appropriate for thin samples or dilute mixtures where resonances don’t interfere with each other due to their effects on the neutron flux. The effects of each resonance simply add together in computing the net cross section, and the cross section can be computed by doing weighted averages using the specified probability distributions.

There are some complicated rules used for determining the energy grid in the unresolved range. Modern evaluations normally give the parameters as functions of energy, and the rule is to first compute the cross sections at these energy points, and then to interpolate between the cross sections to get intermediate values. However, some of the older evaluations contain rather large jumps in energy in between the grid values given, *e.g.*, decade steps. The evaluators did not expect that their work would be linearly interpolated across such large steps, because earlier ENDF rules did not require that. Therefore, RECONR watches for large steps, and it automatically subdivides them using points from a standard grid with roughly logarithmic steps. This gives a better approximation of the roughly $1/v$ shapes expected. Some evaluations for minor isotopes that were made without benefit of detailed experimental data just assume that the average resonance parameters are constant across the unresolved range. RECONR automatically subdivides the range using its standard grid, thus yielding a cross section with a reasonable shape resulting from the natural energy dependences in the resonance formulas.

RECONR includes the unresolved contributions in the normal elastic, fission, capture, and total tabulations in File 3 of the PENDF tape, and it generates a new

section of File 2 with the special value MT=152 that also contains the unresolved cross sections. This section may also be used by UNRESR or PURR for self-shielded cross sections.

3 BROADR: Doppler Broadening

BROADR adds temperature dependence to the pointwise cross sections generated by the RECONR module (see UNRESR and PURR for temperature dependence in the unresolved energy range). It can also be used to rebroaden the results of a previous BROADR run to a higher temperature.

In a material at temperature T , the target atoms and molecules are moving around randomly with a distribution of velocities given by the Maxwell-Boltzmann function:

$$P(v', T)dv' = \frac{\alpha^{3/2}}{\pi^{3/2}} \exp(-\alpha v'^2) dv', \quad (1)$$

where $\alpha = M/(2kT)$. When a neutron moves through the material, it is the *relative velocity* between the neutron and the target nucleus that determines the proper cross section to use. We can define a single effective cross section for neutrons with velocity v by conserving the reaction rate

$$v\bar{\sigma}(v, T) = \int dv' |v - v'| \sigma(|v - v'|) P(v', T). \quad (2)$$

BROADR uses methods based on early versions of Cullen's SIGMA1 code to compute these effective cross sections for linear-linear representations of the basic reaction cross sections. The basic cross sections are obtained from a PENDF tape generated by RECONR or produced in a previous BROADR run. The Doppler-broadened effective cross sections are written out as a multi-temperature PENDF tape.

The following is an example of a simple BROADR run to Doppler broaden the results of a previous RECONR run (on `tape21`) for the H-1 evaluation (on `tape20`):

```
20 21 22
125 1/
.001/
300/
0/
```

To understand this deck, study the BROADR input instructions. It basically says, broaden MAT125 to 1 temperature using a tolerance of 0.1% and a temperature of 300K.

3.1 Typical Broadened Cross Sections

The effects of temperature on typical cross sections is best illustrated by giving several examples. The left side of the following figure shows the (n,α) cross section for B-10 at temperatures of 0K, 3000K, and 3000000K. This shows that a $1/v$ cross section is invariant under Doppler broadening. The right side shows the elastic cross section for carbon at temperatures of 0K, 300K, 3000K, and 30000K. This shows that a constant cross section will develop a $1/v$ tail under Doppler broadening.

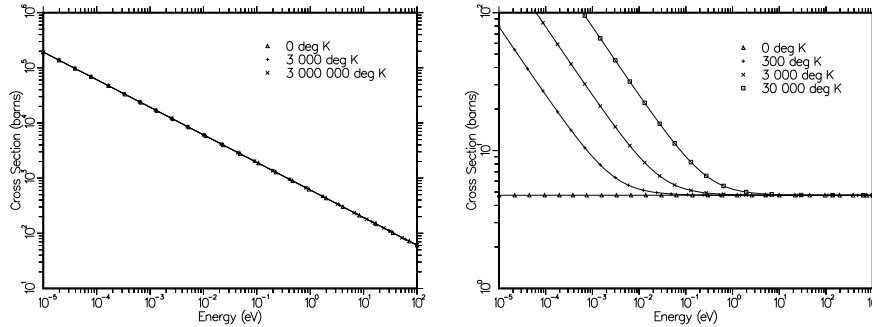


Figure 1: Broadening for constant and $1/v$ cross sections

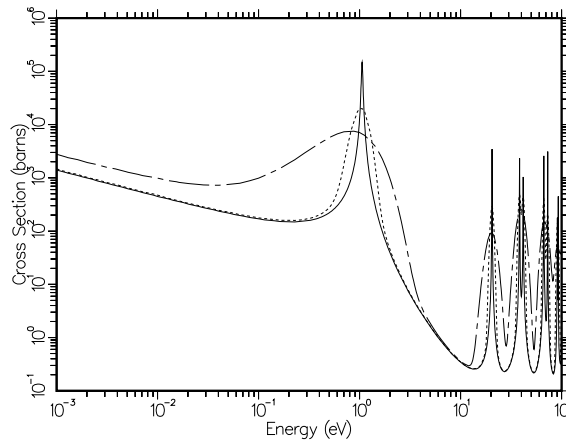


Figure 2: Broadening for resonance cross sections

The next figure demonstrates how resonance cross sections behave under Doppler broadening. The (n,γ) cross section for Pu-240 is shown for temperatures of 0K,

30000K, and 300000K. Resonances with energies larger than kT/A broaden symmetrically (and their areas tend to remain constant). Low energy resonances develop an additional $1/v$ tail, and their areas do not remain constant under Doppler broadening. These effects are best understood by noting that Doppler broadening preserves the reaction rate $v\sigma(v)$, and a finite reaction rate is expected for T larger than zero even as v goes to zero.

3.2 Energy Grids

As discussed in the RECONR section, the zero-temperature PENDF tape uses a unionized energy grid that represents the cross sections to within certain accuracy criteria using linear-linear interpolation. As shown on the preceding page, the shapes of the cross sections can change radically after Doppler broadening, and we should expect that the optimum unionized energy grid will have to change also.

As an example of this, the following graph shows the 20 eV resonance in the (n,γ) cross section of Pu-240 for the two temperatures 0K and 300000K. Note that the broadened curve doesn't need as many energy points near the center of the resonance as the zero-temperature curve, but it needs more points out on the steep sides of the wings of the resonance near 15 and 25 eV.

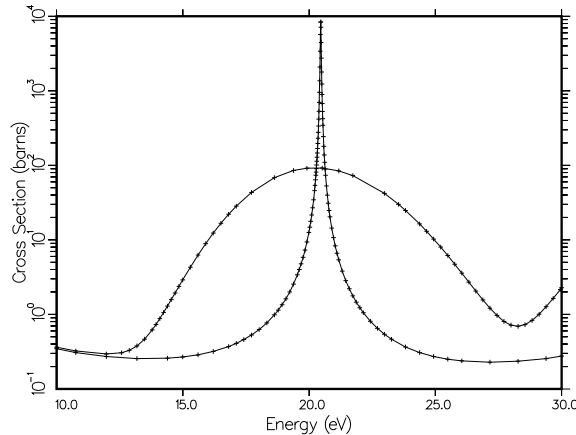


Figure 3: Effect on energy grid of Doppler broadening

BROADR generates the new grid using a method very similar to the one used for resonance reconstruction by RECONR. Starting at one particular energy point, BROADR skips forward over some of the energy points in the input grid to find the top of a “panel” to be relinearized. The top of the panel is determined by trying several criteria, such as a maximum lethargy step, or a maximum number of

points skipped. But most important, in NJOY97, BROADR tries to keep energy values that end with several zeros. This means that the original “node” values from RECONR will be kept in the new grid. These include standard values like .02530000, 1.000000+2, 2.000000+2, 5.000000+2, and 1.000000+3. But they also include the center energies of the resonances and the two half-height values used to seed resonance reconstruction. This approach keeps all the convenient simple points, preserves all the important structure, and allows BROADR the most possible freedom for generating an ideal new energy grid within these constraints.

Once the upper and lower limits of the panel have been determined, the code then checks whether it should be divided in half by comparing the computed Doppler-broadened cross section at the midpoint to the linearly interpolated value. It uses the same stack logic, fractional accuracy checks, and resonance integral checks as RECONR. Of course, all of the broadened reactions are checked at each midpoint, which preserves the union nature of the energy grid.

3.3 Numerical Precision

The energy grid on the input PENDF tape may have been produced by RECONR, or it may be the result of a previous BROADR run. In either case, most of the energy values will be given to seven significant figures using the standard 11-column ENDF format, *e.g.*, 1.234567+1. It is possible that some of the input energies will use the nine-digit form, *e.g.*, 1.23456789. All of the cross section values will use seven significant figures.

When BROADR determines its new union grid, all the “nodes” used as starting points for the relinearization can be represented with the seven-digit form. The new points in the grid are produced just as in RECONR, and it is possible that nine-digit energies may be required in some cases. However, broadening tends to smooth the cross sections, and the nine-digit energies will often disappear. All Doppler-broadened cross sections are reduced to seven significant figures on output.

3.4 Broadenable Reactions

As demonstrated in the preceding sections, temperature dependence can be introduced by Doppler broadening smooth cross sections at low energies (less than about $16kT/A$). In addition, cross sections with features that are sharp with respect to kT/A , such as resonances, will show temperature dependence. Therefore, the reactions that need to be Doppler broadened are normally elastic, fission, and capture. Sometimes (n, α) reactions are present at low energies also. Cross sections in the unresolved resonance range cannot be broadened (see UNRESR and PURR).

The maximum energy for broadening, `thnmax` is defaulted to either the first reaction threshold (typically MT=51, the first inelastic level), the start of the unresolved energy range, or 1 MeV, whichever is the smallest. BROADR selects the

set of reactions to be broadened by looking for those that are present in this energy range, which is usually elastic, fission, and capture.

It is possible that these reactions will show no changes under Doppler broadening over much of this energy range. For example, resonances in light isotopes tend to be broad with respect to thermal energies. However, we usually just go ahead and run BROADR on everything.

If you want to use NJOY cross sections for supernova calculations, you may want to include more reactions, such as the first few inelastic levels. There is an option with `thnmax` negative that allows this.

3.5 Thermal Quantities and Resonance Integrals

In the thermal reactor business, quite a bit of insight into the behavior of a particular evaluation can be obtained by examining certain thermal cross sections and resonance integrals. In a reactor, most fission neutrons are born from fission reactions at thermal energies. The quantities of interest there are the fission cross section itself, the competing capture cross section, and the number of fission neutrons produced per reaction, ν -bar. The deviation of the shape of the cross sections from a $1/v$ shape is also of interest. The fission neutrons are produced at energies up around 1 MeV, and they have to slow down to thermal energies before producing more neutrons to continue the chain reaction. The quantities of interest here are the fission and capture resonance integrals, which help determine the probability that a neutron will disappear before reaching thermal energies. For these reasons, the following simple parameters have come into wide use:

- fission cross section at thermal
- fission ν -bar at thermal
- capture cross section at thermal
- thermal Maxwellian fission integral
- thermal Maxwellian capture integral
- fission g -factor (deviation from $1/v$)
- capture g -factor
- thermal alpha integral
- thermal eta integral
- thermal K1 integral
- fission resonance integral
- capture resonance integral

The conventional thermal energy is 0.0253 eV (296.3K). If the cross section has a $1/v$ shape, its integral weighted against a Maxwellian spectrum for 0.0253 eV is given by

$$I = \int \sigma(E)(E/kT) \exp(-E/kT)d(E/kT), \quad (3)$$

$$I = \int \sigma(kT) \sqrt{kT/E} (E/kT) \exp(-E/kT) d(E/kT) = \frac{\sqrt{\pi}}{2} \sigma(kT), \quad (4)$$

$$G = \frac{2}{\sqrt{\pi}} \frac{I}{\sigma(kT)}, \quad (5)$$

where the factor G provides a measure of the deviation from a $1/v$ shape. It will be unity for a pure $1/v$. Similarly, the alpha, eta, and K_1 integrals are Maxwellian averages of the following quantities:

$$\alpha = \frac{\sigma_c}{\sigma_f}, \quad (6)$$

$$\eta = \frac{\nu \sigma_f}{\sigma_f + \sigma_c}, \quad (7)$$

$$K_1 = (\nu - 1) \sigma_f - \sigma_c. \quad (8)$$

The quantity K_1 is known to be an especially good indicator for k_{eff} . The following is an example for U-235 from Release 3 of ENDF/B-VI:

```

thermal quantities at 300.0 K = 0.0259 eV
-----
fission xsec at 0.0253: 5.8472E+02
fission nubar at 0.0253: 2.4338E+00
capture xsec at 0.0253: 9.8575E+01
fission xsec at tev: 5.7830E+02
fission nubar at tev: 2.4338E+00
capture xsec at tev: 9.7349E+01
thermal capture integral: 8.5429E+01
thermal capture g-factor: 9.9022E-01
capture resonance integral: 1.4337E+02
thermal fission integral: 4.9323E+02
thermal fission g-factor: 9.6239E-01
thermal alpha integral: 1.6117E-01
thermal eta integral: 2.0059E+00
thermal k1 integral: 6.2301E+02
fission resonance integral: 2.7750E+02
-----

```

A good reference for experimental values of these quantities is S. F. Mughabghab, *Neutron Cross Sections, Volume I, Neutron Resonance Parameters and Thermal Cross Sections, Part B, Z=61-100*, Academic Press, 1984. Its values for the quantities in the table above are

- thermal capture cross section = 98.3 ± 0.8 b
- thermal fission cross section = 582.6 ± 1.1 b
- thermal nu-bar = $2.4251 \pm .0034$
- thermal alpha = $0.1687 \pm .0015$
- thermal eta = $2.0751 \pm .0033$
- fission g factor = $0.9761 \pm .0012$

- capture resonance integral = 144 ± 6 b
- fission resonance integral = 275 ± 5 b

which compare fairly well with the numbers from ENDF and BROADR.

4 Unresolved Resonance Self Shielding

At higher energies in heavier nuclei, the resonances get so close together that they cannot be given separately. Instead of giving individual resonances with their energies and characteristic widths, ENDF-format evaluations give average values for the resonance spacing and the various characteristic widths, together with the probability distributions needed to describe the quantities.

In this unresolved range, it is no longer possible to compute simple cross section *vs* energy tables; instead, you can calculate the effective cross section in the region of a given energy, which depends on the environment through self-shielding effects, or you can calculate probability distributions for the total cross section and the related elastic, fission, and capture distributions. The self-shielded cross sections are computed by UNRESR, and the probability tables are computed by PURR.

The effective self-shielded cross sections from UNRESR are normally used by the GROUPT module for generating self-shielded multigroup constants. The probability tables from PURR are usually processed by the ACER module and made available to the MCNP Monte Carlo code.

5 Physics Add Ons To PENDF Tapes

There are a number of physical quantities that can be computed from ENDF data, but that are not explicitly included in most ENDF files. NJOY uses its data-flow model to generate such quantities and add them to a PENDF tape. Several of these physical quantities will be discussed in the following pages:

- Gas-production cross sections (sometimes called charged-particle production) are produced by the GASPR module and add to File 3 of a PENDF tape using the MT numbers from 203 (H-1 production) to 207 (He-4 production). These cross sections are normally sums over several of the reactions found on a normal ENDF-format evaluation.
- Heating and radiation damage cross sections are generated by the HEATR module and added to File 3 of a PENDF tape using special MT numbers. Values of 300+MT are used to represent heating associated with MT; normally only MT=301 for total heating is given. A special value of MT=444 is used for total radiation damage energy production, and some other values nearby are used for the components of damage, if desired.

- Thermal scattering cross sections that take the binding of the target atom into account are produced in the THERMR module. Results are added to both File 3 (cross sections) and File 6 (energy-angle distributions) using a special set of MT numbers including MT=221 for free gas scattering, MT=222 for hydrogen bound in water, and higher MT values for other important moderating materials.

The modular nature of the NJOY data-flow model allows you to include these additional physical quantities or omit them, depending on the needs of your particular applications.

6 GASPR: Gas Production Cross Sections

In many practical applications, it is important to know the total production of protons (hydrogen), alphas (helium), and other light charged particles resulting from the neutron flux. Therefore, it is convenient to have a set of special “gas production” or “charged-particle production” cross sections that can be used in application codes. The ENDF format provides a set of MT numbers for these quantities, but only a few evaluators have added them to their files:

- MT=203 – total proton production
- MT=204 – total deuteron production
- MT=205 – total triton production
- MT=206 – total He-3 production
- MT=207 – total alpha production

The GASPR module goes through all of the reactions given in an ENDF-format evaluation, determines which charged particles would be produced by the reaction, and adds up the particle yield times the reaction cross section to produce the desired gas production cross sections. It uses data from an input ENDF tape and an input PENDF tape, and it writes the results on an output PENDF tape. They are then available for plotting, multigroup averaging, or reformatting for the MCNP Monte Carlo code.

6.1 Gas Producing Reactions

Nuclear reactions can result in a number of different products, including photons, neutrons, light charged particles, and heavy residual nuclei. The identities of the products are sometimes implicit in the MT number for the reaction, and they are sometimes given explicitly in the section of File 6 corresponding to the reaction. When the products are implicit, they are produced with integer yields. The following table gives some common examples:

MT	products
22	1 neutron, 1 alpha, residual
103	1 proton, residual
111	2 protons, residual
113	1 triton, 2 alphas, residual

Of course, the residual could also be a light charged particle, and it may contribute to the gas production cross section.

Many of the newer evaluations now available use File 6 to provide a complete accounting of all the particles produced in a nuclear reaction, including their yields and their distributions in angle and energy. If the section in File 6 uses a simple MT number, like those tabulated above, the explicit yields will be the same as the implicit ones. However, it is also possible to use MT=5 with File 6. This MT is the sum over a number of reaction channels, and the particle yields do not have to be integers. GASPR is able to take account of these non-integer particle yields and add the correct contributions to the total gas production.

7 HEATR: Heating and Radiation Damage

Nuclear heating results from the slowing down of energetic charged particles produced in nuclear reactions, including the recoil nucleus from scattering reactions. It is a very important quantity. Sometimes it is the product being sold (as in power reactors), and sometimes it is a damaging corollary of the nuclear reactions (as in melting of important structural elements). The HEATR module of NJOY can be used to compute estimates of energy-deposition cross sections for neutrons that can be combined with calculations of neutron fluxes in nuclear systems to compute the neutronic contributions to nuclear heating. The heating due to the photon flux in a nuclear system is usually even more important; it is computed in the GAMINR module.

The same energetic charged particles and recoil nuclei that lead to heating can also cause damage to the crystalline structure of the materials that they pass through. An important case of this is the embrittlement of power reactor containment vessels that is one of the main limiting factors in the useful life of a commercial reactor. HEATR computes the damage-production energy, which can be correlated to macroscopic damage, such as tensile strength, ductility, or resistivity, through phenomenological factors like DPA (displacements per atom).

8 THERMR: Thermal Cross Sections and Scattering

At thermal energies, *e.g.*, up to about 0.5 eV for temperatures around room temperature and maybe up to as high as 4 eV for hotter materials, the energy transferred

by the scattering of a neutron is similar to the kinetic energies of motion of the atoms in liquids and to the energies of excitations in molecules and crystalline lattices. Therefore, you cannot picture the target atoms as being initially stationary and recoiling freely as is normally done for higher neutron energies. The motion of the target atoms and their binding in liquids and solids affects both cross sections and the distribution in energy and angle of the scattered neutrons. The THERMR module of NJOY is used to compute these effects and add them to a PENDF tape for use by other modules.

For free-gas scattering, where only the thermal motion of the targets is taken into account, not internal modes of excitation, THERMR can generate the cross sections and scattering distributions using analytic formulas. For real bound scattering, it uses an input scattering function and other parameters from an ENDF-format thermal evaluation in File 7 format. A number of such evaluations for common moderator materials have been available for years in various ENDF-format libraries, and new ones have been produced recently using the LEAPR module of NJOY. The results of THERMR's work are stored into the new PENDF tape in Files 3 and 6 using a special set of MT numbers:

MT	Moderator
221	free gas
222	H in H ₂ O
223,224	H in polyethylene
225,226	H in ZrH _n
227	benzene
228	D in D ₂ O
229,230	C in graphite
231,232	Be
233,234	BeO
235,236	Zr in ZrH _n

As will be discussed below, the lines with two MT values refer to the inelastic and elastic components of scattering, respectively.

8.1 Incoherent Inelastic Scattering

The most interesting component of thermal scattering is called “incoherent inelastic” because the neutron exchanges energy with the target molecule or crystalline lattice and all the scattered waves are assumed to combine incoherently without interference effects. This component is described in terms of a scattering function called “S of alpha and beta,” where alpha and beta are reduced values for momentum transfer and energy transfer, respectively:

$$\sigma^{\text{inc}}(E, E', \mu) = \frac{\sigma_b}{2kT} \sqrt{\frac{E'}{E}} e^{-\beta/2} S(\alpha, \beta), \quad (9)$$

$$\beta = \frac{E' - E}{kT}, \tag{10}$$

$$\alpha = \frac{E' + E - 2\mu\sqrt{EE'}}{Akt}, \tag{11}$$

$$\sigma_b = \sigma_f \frac{(A + 1)^2}{A^2}. \tag{12}$$

Here σ_b is the bound scattering cross section, σ_f is the free scattering cross section, E and E' are initial and final neutron energies, μ is the scattering cosine, and A is the ratio of the target mass to the neutron mass.

The scattering function for a gas of particles with no internal structure (free gas) is given by

$$S(\alpha, \beta) = \frac{1}{\sqrt{4\pi\alpha}} \exp\left\{-\frac{\alpha^2 + \beta^2}{4\alpha}\right\} \tag{13}$$

A bound scattering function is considerably more complex, as shown by the following two figures for hydrogen bound in water:

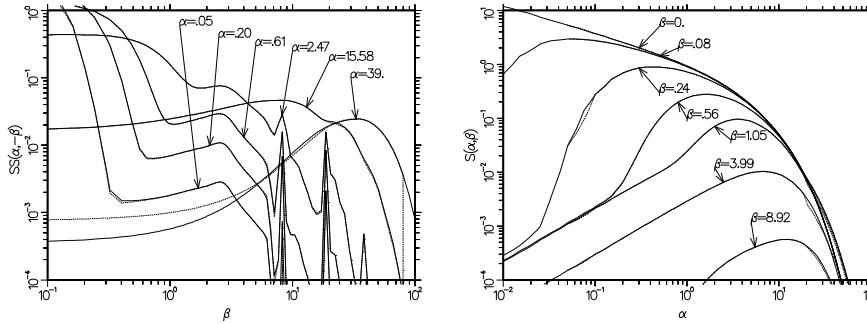


Figure 4: Views of $S(\alpha, \beta)$ for H in H_2O

The sharp features in the left figure are due to the discrete molecular vibrations at .205 and .408 eV. The peak that develops at small alpha and beta is a pseudo-elastic peak resulting from diffusion. For more information on the physics going on in these figures, see the LEAPR write up.

THERMR processes the scattering function into an incoherent-inelastic cross section *vs* energy and a normalized energy-angle distribution. An example of such a distribution is shown below for beryllium. A number of “up-scatter” peaks corresponding to the deexcitation of lattice vibrations are seen for low incident energies.

8.2 Coherent Elastic Scattering

In crystalline materials consisting of coherent scatterers (*e.g.*, graphite, beryllium, beryllium oxide), the scattering from different planes of atoms can interfere, leading

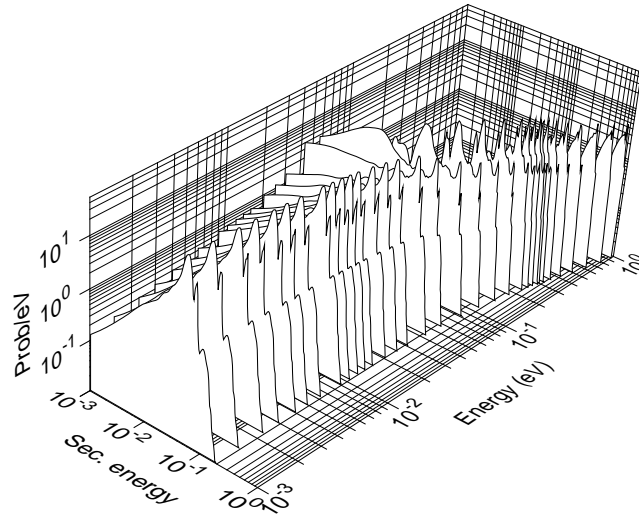


Figure 5: Thermal scattering from beryllium metal at 300K

to a series of “Bragg edges” as the neutron wavelength hits various possible atomic spacings. The thermal elastic cross section of graphite is shown below:

The mathematical expression of the energy-angle distribution and integrated cross section for coherent elastic scattering is as follows:

$$\sigma^{\text{coh}}(E, E', \mu) = \frac{\sigma_c}{E} \sum_{E_i < E} f_i e^{-2W E_i} \delta(\mu - \mu_0) \delta(E - E'), \quad (14)$$

$$\mu_0 = 1 - \frac{E_i}{E}, \quad (15)$$

$$\sigma^{\text{coh}} = \frac{\sigma_c}{E} \sum_{E_i < E} f_i e^{-2W E_i}. \quad (16)$$

Note that the scattering distribution is completely defined by the the energy locations and sizes of the steps in the integrated cross section. The GROUPR module and the MCNP Monte Carlo code take advantage of this feature.

8.3 Incoherent Elastic Scattering

In solids containing hydrogen, such as polyethylene or frozen methane, the strong incoherence of the scattering from hydrogen simplifies the elastic term:

$$\sigma^{\text{iel}}(E, E', \mu) = \frac{\sigma_b}{2} e^{-2W E(1=\mu)} \delta(E - E'), \quad (17)$$

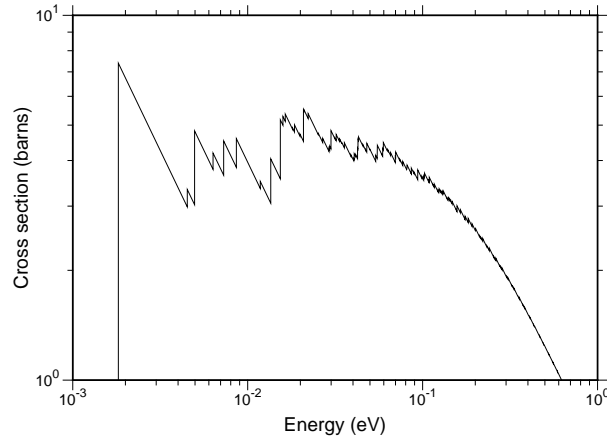


Figure 6: Coherent scattering for graphite

$$\sigma^{\text{iel}}(E) = \frac{\sigma_b}{2} \left\{ \frac{1 - e^{-4We}}{2WE} \right\}. \quad (18)$$

The cross section only depends on the bound scattering cross section and the Debye-Waller coefficient, W , which is computed from the excitation spectrum for the solid. The following shows an example of the elastic (dashed) and inelastic (solid) terms for frozen methane at 22K:

8.4 Data Representation

The cross sections *vs* energy generated by THERMR are put into File 3 as simple tabulations using the MT number (or numbers) requested by the user. The energy-angle distributions appear as normalized distributions in File 6. For a specified grid of incident energies, a distribution *vs* secondary energy is given on a grid chosen to represent the actual angle-integrated function to a specified accuracy. At each secondary energy, a specified number of discrete scattering cosines is given with a predetermined set of probability values. This format allows for easy sampling in a Monte Carlo code, and it can also produce good values for Legendre coefficients when needed. As an example of a THERMR run, consider the following input deck:

```

26 22 23
1065 1306 8 1 4 1 1 229 0
300.
.01 1.2

```

To understand this input, compare it with the THERMR input instructions in the online version. Briefly, this run is for carbon from ENDF/B-V (MAT=1306).

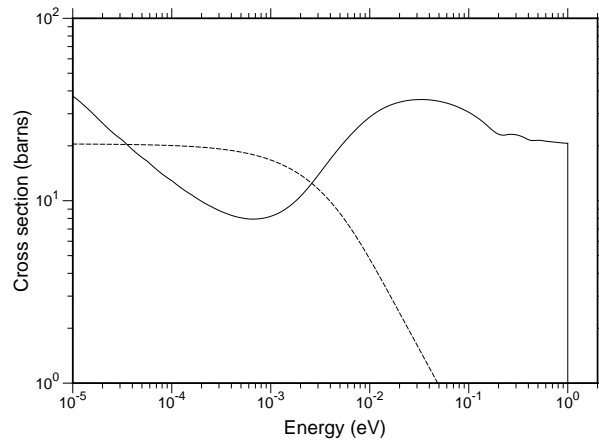


Figure 7: Incoherent elastic scattering for solid methane at 22K

Thermal MAT=1065 is graphite from the thermal ENDF file Tape 322 mounted on unit 26. The results will be written on the new PENDF tape on unit 23 using MT=229 and 230. There will be 8 discrete secondary cosines given for each $E \rightarrow E'$. Reconstruction will be a 1%, and the incident energies will range up to 1.2 eV.

9 GROUPR: Multigroup Constants

It is very time consuming to carry out full calculations of neutron transport using the detailed PENDF cross sections that we have discussed in the preceding pages, although it can be done using continuous-energy Monte Carlo codes like MCNP, which we will discuss later. Therefore, it is very common to reduce the detail in the cross sections by averaging them over a set of energy ranges called “groups.” The GROUPR module of the NJOY system carries out such averages for the point-wise cross sections, and it also generates multigroup “matrices,” which describe the transfer of neutrons from one group to another.

The basis for GROUPR is the “multigroup Boltzmann transport equation,” which can be written as

$$\begin{aligned} \mu \frac{\partial}{\partial x} \phi_g(x, \mu) &+ \sum_{\ell=0}^{\infty} P_{\ell}(\mu) \sigma_{t\ell g}(x) \phi_{\ell g}(x) \\ &= \sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} P_{\ell}(\mu) \sum_{g'} \sigma_{X\ell g' \rightarrow g}(x) \phi_{\ell g'}(x) + Q_g(x, \mu), \quad (19) \end{aligned}$$

where slab geometry has been used for simplicity. The first term describes the

spatial transport of neutrons in group g , the second accounts for reactions in group g , the third gives the source into group g from other groups, and the last includes any fixed sources and the source from fission. The flux and cross sections that appear here are defined as follows:

$$\phi_{\ell g}(x) = \int_g \phi_{\ell}(x, E) dE , \quad (20)$$

$$\sigma_{t\ell g}(x) = \frac{\int_g \sigma_t(x, E) \phi_{\ell}(x, E) dE}{\int_g \phi_{\ell}(x, E) dE} , \quad (21)$$

$$\sigma_{X\ell g' \rightarrow g} = \frac{\int_g dE \int_{g'} dE' \sigma_{X\ell}(x, E' \rightarrow E) \phi_{\ell}(x, E)}{\int_g \phi_{\ell}(x, E) dE} . \quad (22)$$

One of the reasons for doing a transport calculation is to obtain various “responses,” such as heating, production of a radionuclide, or production of helium. The responses can be computed using the multigroup flux together with a multigroup reaction cross section:

$$\sigma_r(x) = \frac{\int_g \sigma_r(x, E) \phi_0(x, E) dE}{\int_g \phi_0(x, E) dE} , \quad (23)$$

It is the mission of GROUPT to compute the cross sections with “g” subscripts, which are often called “multigroup constants.” The Boltzmann equation itself is then solved in a separate neutron transport code, such as ANISN or DANT.

9.1 Weighting Function

“Wait a minute,” you ask, “the purpose of solving the transport equation is to get the flux, but I have to know the flux to compute the multigroup constants!” This conundrum is the source of much of the “art” in using multigroup methods. If you can make a good guess for the shape of the flux (mostly the *intragroup* flux) for the class of problems of interest, you can do very good multigroup calculations with only a few groups. If the flux can change shape between different problems or between different regions of one problem, then you have to use a large number of groups together with some default shape (such as flat weighting). The presence of resonance absorbers in a problem results in complex shapes for the weighting function, which leads to the effects called “self shielding.”

As a first example, consider a typical water-moderated critical assembly, such as a power reactor. Neutrons are born at high energies (about 2 MeV) with a

characteristic fission spectrum, many of them slow down by collisions with hydrogen nuclei in the water, resulting in a $1/E$ shape, and they finally come into equilibrium when they reach thermal energies, resulting in a Maxwellian shape appropriate to the temperature of the system. As shown in the left side of the following figure, GROUPR contains several builtin weighting functions that show this combination of shapes:

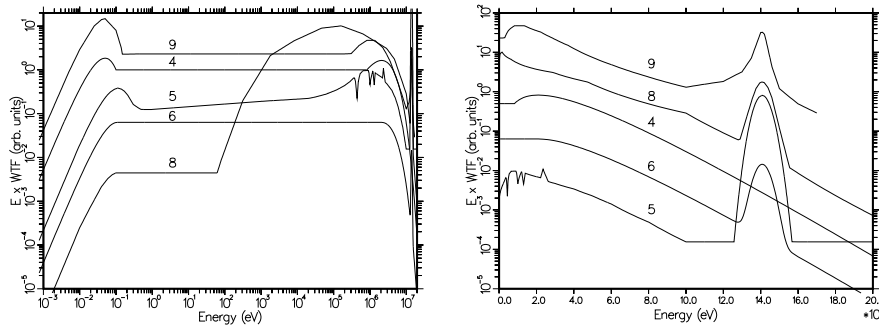


Figure 8: Two views of the NJOY built-in weighting functions

The curves are plotted as log-log “weight per unit lethargy” or $E \times W(E)$, which makes the central $1/E$ part of the curve flat. The parameters used to generate the curve for IWT=4 were a thermal temperature of .0253 eV joined to $1/E$ at 0.1 eV, and a fission temperature of 1.40 MeV joined to $1/E$ at 820.3 keV. Curve 8 is for a fast reactor or fusion blanket.

As a second example, consider a fusion system. The neutrons born in d-T reactions appear as a sharp peak centered near 14 MeV. They then scatter down to lower energies by elastic and inelastic processes, producing a shape in the 1 MeV range very similar to the fission spectrum. If there are few light isotopes around, few neutrons get to thermal energies, and a shape like Curve 8 results. The high-energy shapes of some of GROUPR’s built-in weighting functions are shown in the right side of the preceding figure.

Here is a brief summary of the built-in weighting functions:

- IWT=2, constant. Used for very fine group structures like the 640-group dosimetry structure.
- IWT=3, $1/E$. Used for calculating resonance integrals.
- IWT=4, analytic thermal+ $1/E$ +fission. Allows the user to adjust the relative amounts of thermal and fission flux.

- IWT=5, mid-life PWR flux spectrum with a fusion peak added. Includes some peaks and dips from oxygen, and some hardening of the $1/E$ shape. Used for EPRI data and for the LANL libraries using the WIMS structure.
- IWT=6, similar to 4, but the breakpoints were chosen to make the curve continuous and a fusion peak was added.
- IWT=8, fast reactor weighting function. It has a fusion peak and an intermediate shape typical of a fast reactor or fusion blanket. A thermal part is provided at low energies for outer regions of systems.
- IWT=9, CLAW weighting function. A typical thermal $+1/E+$ fission + fusion shape used for many years at Los Alamos.
- IWT=10, same as IWT=9, except the thermal part is automatically recalculated to follow a Maxwellian law for the actual temperature.
- IWT=11, VITAMIN-E weighting function. Another thermal $+1/E+$ fission + fusion shape.

9.2 Group Structures

If the weighting function is very well known, you can do good transport calculations with very few groups, but with modern computers, the trend has been to use more and more groups. GROUPT contains a number of built in options for group structures (some of them of mostly historical interest), and the user can enter any arbitrary structure that matches his or her needs.

Reactor physicists have always been fond of the variable called “lethargy,” which is defined as follows:

$$u = \log \left(\frac{10\text{MeV}}{E} \right), \quad (24)$$

$$E = (10\text{MeV})e^{-u}. \quad (25)$$

Lethargy increases as neutrons slow down. Note that in the slowing-down region, where the flux varies like $1/E$, groups with constant lethargy width all contain the same portion of the flux. In elastic scattering, neutrons lose a fraction of their energy with each scattering; thus, the lethargy increases by a fixed amount with each scattering. Because lethargy is such a natural variable for neutron slowing-down problems, many group structures are based on groups with certain lethargy widths. For example,

- the LANL 30-group structure uses one-lethargy steps at low energy,
- the ANL 27-group structure uses even half-lethargy steps, and
- the GAM-I 68-group structure uses even quarter-lethargy steps.

Some group structures start with even lethargy groups and then subdivide some groups that contain especially interesting structure, such as the Fe-56 resonance at 27 keV. In other cases, the group structure starts with even lethargy groups and then moves some bounds to try to center important resonances (*e.g.*, the WIMS 69-group structure).

Another factor when choosing a group structure is its range. Many group structures intended for reactor analysis stop at 10 MeV. They are clearly not suitable for fusion problems. Other group structures only provide a few groups in the thermal range, and they are most suitable for fast-reactor and fusion problems.

The neutron group structures available in GROUPT include:

IGN=3 LANL 30-group structure, used for fusion problems.

IGN=4 ANL 27-group structure, even half-lethargy groups used for fast-reactor analysis after collapse from more detailed structures.

IGN=6 GAM-I 68-group structure, even quarter-lethargy set used for the epithermal part of thermal reactor codes (most recently, EPRI-CELL).

IGN=8 LASER-THERMOS 35-group structure, a thermal structure for the low energy part of power-reactor codes (most recently EPRI-CELL).

IGN=9 EPRI-CPM 69-group structure, the WIMS structure, still widely used all over the world for reactor calculations.

IGN=10 LANL 187-group structure, a single set of fine-lethargy bounds suitable for thermal reactors, fast reactors, and fusion reactors.

IGN=13 LANL 80-group structure, optimized for fast-reactor and fusion systems. Still in use through MATXS files.

IGN=15 SAND-II 640-group structure, used for dosimetry work with flux unfolding codes. It extends the earlier SAND-II 620-group structure to higher energies.

IGN=17 VITAMIN-J 175-group structure, an extension of the 174-group structure. This structure is currently widely used in the fusion community, *e.g.*, ITER design.

9.3 Self Shielding

When the neutrons slow down in a medium with resonance absorption present, the smooth weighting function discussed above will be modified. Dips will appear in the flux corresponding to resonance peaks, and sometimes sharp peaks will occur in the flux corresponding to deep minima in the cross section, or “windows.” The

products of cross section and flux that appear in the definitions of the multigroup constants will clearly be reduced (self shielded) when the dips are large.

The classical method for handling self shielding in multigroup codes is the Bondarenko model. For narrow resonances in large systems, the flux takes the form:

$$\phi(E) = \frac{W(E)}{\Sigma_t(E)}. \quad (26)$$

The denominator contains the total macroscopic cross section for the material. To obtain the part of the flux that provides self shielding for isotope i , it is assumed that all the other isotopes can be represented with a constant “background cross section” called σ_0 . Therefore,

$$\phi^i(E) = \frac{C(E)}{\sigma^i(E) + \sigma_0^i}. \quad (27)$$

The qualitative behavior of this equation is easy to understand. If σ_0 is larger than the tallest peaks in σ_t , the weighting flux is approximately proportional to the smooth weighting function $C(E)$. This is called infinite dilution; the cross section in the material of interest has little or no effect on the flux. On the other hand, if σ_0 is small with respect to σ_t , the weighting flux will have large dips at the locations of the peaks in σ_t , and a large self-shielding effect will be expected.

The following input deck will prepare self shielded cross sections for three temperatures and seven values of σ_0 for Pu-238 from ENDF/B-VI. To understand the deck, compare it with the GROUPT input instructions from the online version of this tutorial or an NJOY manual.

```

groupr
21 22 0 23
9434 6 0 4 0 3 7 1
'94-pu-238'/
300. 900. 2100.
1.e10 1.e5 1.e4 1000. 100. 10. 1
.1 0.025 0.8208e06 1.4e06
3 1 'total'/
3 2 'elastic'/
3 18 'fission'/
3 102 'capture'/
0/
3 1 'total'/
3 2 'elastic'/
3 18 'fission'/
3 102 'capture'/
0/
3 1 'total'/
3 2 'elastic'/
3 18 'fission'/
3 102 'capture'/
0/
0/

```

The corresponding output from GROUPR is shown below (with some excisions to make it fit the page).

```

group constants at t=3.000E+02 deg k
for mf 3 and mt102 capture

energy group constants vs sigma zero
group infinity ... 1.000E+03 1.000E+02 1.000E+01 1.000E+00

 1  3.494E+02 ... 2.927E+02 1.995E+02 1.487E+02 1.396E+02
 2  1.254E+01 ... 1.253E+01 1.246E+01 1.227E+01 1.219E+01
 3  5.580E+00 ... 5.578E+00 5.565E+00 5.521E+00 5.495E+00
 4  7.307E+01 ... 5.052E+01 2.244E+01 1.204E+01 1.029E+01
 5  2.510E+00 ... 2.500E+00 2.434E+00 2.227E+00 2.115E+00
 6  5.200E-01 ... 5.200E-01 5.198E-01 5.189E-01 5.181E-01
 7  1.425E+01 ... 1.099E+01 4.944E+00 2.225E+00 1.743E+00
 8  7.344E+01 ... 2.738E+01 8.348E+00 3.696E+00 2.899E+00
 9  9.300E-02 ... 9.297E-02 9.275E-02 9.174E-02 9.083E-02
10  3.183E+00 ... 2.645E+00 1.262E+00 4.791E-01 3.409E-01
11  1.694E+01 ... 8.739E+00 2.943E+00 1.165E+00 8.858E-01
12  2.627E+01 ... 1.489E+01 5.083E+00 1.979E+00 1.488E+00
13  2.337E+01 ... 1.254E+01 4.472E+00 1.981E+00 1.592E+00
14  1.348E+01 ... 8.083E+00 3.312E+00 1.459E+00 1.123E+00
15  1.081E+01 ... 6.994E+00 3.060E+00 1.373E+00 1.056E+00
16  7.869E+00 ... 5.761E+00 2.772E+00 1.277E+00 9.828E-01
17  6.324E+00 ... 4.939E+00 2.560E+00 1.219E+00 9.415E-01
18  5.154E+00 ... 4.223E+00 2.352E+00 1.165E+00 9.056E-01
19  4.195E+00 ... 3.579E+00 2.141E+00 1.109E+00 8.704E-01

```

And finally, we show a plot of the PENDF cross section for capture (dotted), the infinitely-dilute capture (solid), and the self-shielded capture for a background cross section of 1 barn (dashed).

9.4 Multigroup Matrices

In order to unify several different processing tasks, GROUPR uses the concept of a generalized group integral

$$\sigma_g = \frac{\int_g \mathcal{F}(E) \sigma(E) \phi(E) dE}{\int_g \phi(E) dE}, \quad (28)$$

where the integrals are over all initial neutron energies in group g , $\sigma(E)$ is a cross section at E , and $\phi(E)$ is an estimate of the flux at E . We call $\mathcal{F}(E)$ the “feed function”. It alone changes for different data types. To average a neutron cross section, \mathcal{F} is set to 1. To average a ratio quantity like $\bar{\mu}$ with respect to elastic scattering, \mathcal{F} is set to $\bar{\mu}$. For photon production, \mathcal{F} is the photon yield. For matrices, \mathcal{F} is the ℓ -th Legendre component of the normalized probability of scattering into secondary

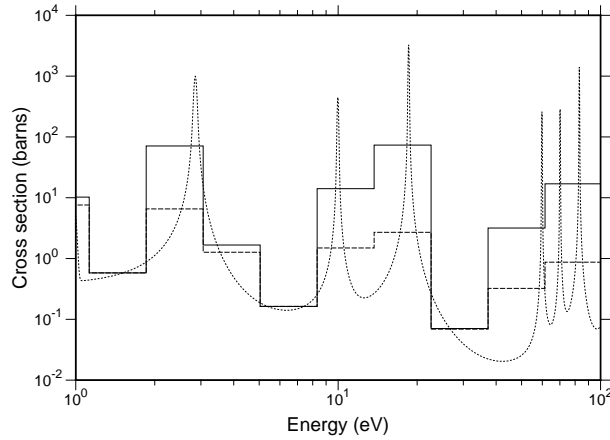


Figure 9: Self shielded cross sections for Pu-238 capture

energy group g' from initial energy E . This definition is clearly independent of whether the secondary particle is a neutron or a photon.

The question of integration grid or quadrature scheme is important for the evaluation this equation. Each factor in the integrands has its own characteristic features, and it is important to account for them all. First, a grid must be established for each factor. As an example, the grid of $\sigma(E)$ is generated in RECONR and BROADR such that sigma can be obtained to within a given tolerance by linear interpolation. GROUPE contains a subroutine GETSIG which carries out this interpolation at E and also returns the next grid energy in ENEXT. Subroutines GETFLX and GETFF perform similar functions for the flux and feed function. It is now easy to generate a union grid for the three-factor integrand by just selecting the next closest ENEXT from all the factors at each step of the integration.

9.5 Two-Body Scattering

Elastic scattering (ENDF MT=2) and discrete inelastic neutron scattering (with MT=51–90) are both examples of two-body kinematics and are treated together by GROUPE. The feed function required for the group-to-group matrix calculation may be written

$$\mathcal{F}_{\ell g'}(E) = \int_{g'} dE' \int_{-1}^{+1} d\omega f(E \rightarrow E', \omega) P_{\ell}(\mu[\omega]) , \quad (29)$$

where $f(E \rightarrow E', \omega)$ is the probability of scattering from E to E' through a center-of-mass cosine ω and P_{ℓ} is a Legendre polynomial for laboratory cosine μ . The

laboratory cosine corresponding to ω is given by

$$\mu = \frac{1 + R\omega}{\sqrt{1 + R^2 + 2R\omega}}, \quad (30)$$

and the cosine ω is related to secondary energy E' by

$$\omega = \frac{E'(1 + A)^2/A' - E(1 + R^2)}{2RE}, \quad (31)$$

where A' is the ratio of the emitted particle mass to the incident particle mass ($A'=1$ for neutron scattering). In these equations, R is the effective mass ratio

$$R = \sqrt{\frac{A(A + 1 - A')}{A'}} \sqrt{1 - \frac{(A + 1)(-Q)}{AE}}, \quad (32)$$

where A is the ratio of target mass to neutron mass, and $-Q$ is the energy level of the excited nucleus ($Q=0$ for elastic scattering). Integrating the defining equation for the two-body feed function over secondary energy gives

$$\mathcal{F}_{\ell g'}(E) = \int_{\omega_1}^{\omega_2} f(E, \omega) P_{\ell}(\mu[\omega]) d\omega, \quad (33)$$

where ω_1 and ω_2 are evaluated using the kinematics equations with E' equal to the upper and lower bounds of g' , respectively. The scattering probability is given by

$$f(E, \omega) = \sum_{\ell=0}^L f_{\ell}(E) P_{\ell}(\omega), \quad (34)$$

where the Legendre coefficients are either retrieved directly from the ENDF/B File 4 or computed from File 4 tabulated angular distributions using subroutines from the GROUPE module called GETFLE and GETCO.

The integration over ω is performed simultaneously for all Legendre components using Gaussian quadrature. The quadrature order is selected based on an estimate of the polynomial order of the integrand using the known order of the CM angular distribution plus a factor depending on atomic weight ratio AWR. More terms are needed for light targets.

The resulting two-body feed function for higher Legendre orders is a strongly oscillatory function of incident energy in some energy ranges. Such functions are very difficult to integrate with adaptive techniques. Gaussian methods, on the other hand, are capable of integrating such oscillatory functions exactly if they are polynomials. Since a polynomial representation of the feed function is fairly accurate, a Gaussian quadrature scheme was chosen for GROUPE. The scheme used is also well adapted to performing many integrals in parallel. In GROUPE, all Legendre components and all final groups are accumulated simultaneously. NJOY

takes pains to locate the “critical points” in incident energy so that the Gaussian integrations are performed over energy ranges that do not contain discontinuities.

Here is an example of a multigroup elastic scattering matrix for carbon computed using GROUPT for a 30-group structure:

```

group constants at t=3.000E+02 deg k           16.6s
for mf 6 and mt 2 elastic

initl final group constants vs legendre order
group group 0          1          2          3

  1     1   8.087E+00  4.532E-01  1.144E-02  4.205E-05
  2     1   7.572E-01 -2.232E-01 -1.690E-02 -4.363E-04
  2     2   4.003E+00  4.900E-01  2.363E-02  4.544E-04
  3     2   7.525E-01 -2.219E-01 -1.676E-02 -4.303E-04
  3     3   3.995E+00  4.879E-01  2.348E-02  4.484E-04
  4     3   7.574E-01 -2.234E-01 -1.686E-02 -4.318E-04
  4     4   3.985E+00  4.891E-01  2.357E-02  4.498E-04
  5     4   7.539E-01 -2.223E-01 -1.678E-02 -4.294E-04
  5     5   3.986E+00  4.880E-01  2.348E-02  4.474E-04
  ...
 28    26   8.308E-02 -4.928E-02  8.439E-03  1.234E-02
 28    27   2.545E-01  7.532E-02 -6.513E-03  2.423E-02
 28    28   5.256E-01  4.740E-01  3.849E-01  2.805E-01
 29    26   4.329E-03 -4.075E-03  3.606E-03 -2.988E-03
 29    27   1.217E-01 -4.973E-02 -1.258E-02  1.592E-02
 29    28   1.841E-01  1.053E-01  2.710E-02  4.737E-04
 29    29   5.124E-01  4.633E-01  3.788E-01  2.799E-01
 30    27   2.721E-02 -2.127E-02  1.255E-02 -5.134E-03
 30    28   6.865E-02 -1.719E-02 -1.846E-02  1.032E-02
 30    29   1.834E-01  1.016E-01  1.937E-02 -7.491E-03
 30    30   6.055E-01  5.428E-01  4.372E-01  3.182E-01

```

Note that the “in-group” cross section is much larger than the outscattering cross section, which is normally limited to a fairly small energy loss by kinematics. Also note that the scattering tends to be fairly isotropic at low energies (low group numbers here), but it becomes very forward peaked for in-scatter in group 30. This is demonstrated by the slow decrease for higher Legendre orders (a delta function of forward scattering would have the same cross section for each ℓ order).

9.6 Continuum Scattering and Fission

In ENDF/B, scattering from many closely-spaced levels or multibody scattering such as $(n,2n)$, $(n,n'\alpha)$, and fission can be represented using a separable function of scattering cosine and secondary neutron energy

$$f(E \rightarrow E', \mu) = F(E, \mu) g(E \rightarrow E'), \quad (35)$$

where F is the probability that a neutron will scatter through a laboratory angle with cosine μ irrespective of final energy E' . It is obtained from MF=4. Similarly,

g is the probability that a neutron's energy will change from E to E' irrespective of the scattering angle, and it is given in MF=5. Continuum reactions are easily identified by MT numbers of 6–49 and 91. The secondary-energy distributions can be represented using a two-dimensional tabulation or using an analytic law. The feed functions for continuum scattering are simply

$$\mathcal{F}_{\ell g'}(E') = \int_{-1}^{+1} P_{\ell}(\mu) f(E, \mu) d\mu \int_{g'} g(E, E') dE' . \quad (36)$$

As discussed in the companion article in this volume “An Introduction to the ENDF Formats,” the separability assumption is no longer considered adequate for reaction like (n,2n) or (n,n' α), because it doesn't describe the energy-angle correlation that occurs for these reactions. As a result, the continuum approach is only found for old carry-over materials and for minor actinides or fission products where neutron transport is a secondary concern. In these cases, the angular distribution is almost invariably given as isotropic in the laboratory system. However, the continuum approach remains the method of choice for fission, which is also modeled as isotropic in the laboratory frame. Therefore, in almost all cases of interest, the feed function reduces to

$$\mathcal{F}_{0g'}(E') = \int_{g'} g(E, E') dE' . \quad (37)$$

The integrals on the right hand side are returned by the GROUPT subroutine GETSED (for get secondary energy distribution), which either interpolates and integrates over the tabulation, or uses direct analytical integrals, as required. The integration over incident energy proceeds as for all other GROUPT quantities. The result is a scattering or fission matrix, $\sigma_{g \rightarrow g'}$. For scattering, the matrix will contain only downscattering elements for groups g from the threshold up. For fission, the matrix can be almost completely full with both energy increase and energy loss elements.

9.7 Fission Source, Nubar, and Chi

As discussed above, the natural representation of fission in GROUPT is a a group-to-group transfer matrix, $\sigma_{g \rightarrow g'}$. But the quantity required for the multigroup Boltzmann transport equation is the fission source into group g ,

$$S_g = \sum_{g'} \sigma_{f g' \rightarrow g} \phi_{0g'} , \quad (38)$$

Most existing transport code packages are not able to use a full fission matrix directly to compute this fission source; instead they count on the observation that, except for relatively high neutron energies, the spectrum of fission neutrons is only weakly dependent on initial energy. Therefore, the fission source can be written

$$S_g = \chi_g \sum_{g'} \bar{\nu}_{g'} \sigma_{f g'} \phi_{0g'} , \quad (39)$$

where $\bar{\nu}_g$ is the fission neutron yield, σ_{fg} is the fission cross section, and χ_g is the average fission spectrum (the familiar “chi” vector), which can be defined by

$$\chi_g = \frac{\sum_{g'} \sigma_{fg' \rightarrow g} \phi_{0g'}}{\sum_g \sum_{g'} \sigma_{fg' \rightarrow g} \phi_{0g'}} , \quad (40)$$

The fission neutron yield is given by

$$\bar{\nu}_g = \sum_{g'} \sigma_{fg' \rightarrow g} / \sigma_{fg} . \quad (41)$$

Clearly, χ_g computed in this way depends on the flux in the system of interest. In most cases, the built in weighting functions in GROUPT are not adequate for computing χ_g ; they are really only intended to provide reasonable *intragroup* fluxes. For this reason, GROUPT does not attempt to compute the fission chi itself—it just passes on the full fission matrix to some subsequent code that allows the user to enter an appropriate flux spectrum. Because the energy dependence is weak except for high incident energies, a rough guess for ϕ_{0g} usually gives an accurate fission chi. When this is not the case, the problem can be iterated. The DTFR and WIMSR modules know how to calculate chi from GROUPT output, and the TRANSX code working with MATXS files can support the fission-spectrum iteration described above.

There is an addition complication for fission. The fission matrix from GROUPT only describes the prompt part of fission. It is also necessary to obtain the delayed neutron yield and spectra from GROUPT and combine them with the prompt matrix to contain the complete total fission nu $\bar{\nu}$ and chi.

9.8 Running GROUPT

Setting up GROUPT runs can be daunting, because it has many possibilities for giving the user complete control over which reactions and data types are processed. However, for many applications, the user can use automatic modes that greatly simplify the input. Here is a simple example of a deck that computes infinitely dilute cross sections for carbon using a built-in group structure and weight function.

```

groupr
20 21 0 22 / ENDF on tape20, PENDF on tape21, GENDF on tape22
1306 3 3 3 3 1 1 1 / 30 groups, CLAW weight, P3
'carbon from ENDF/B-V'/
300 / one temperature
1.e10 / infinite dilution only
3/ process all cross sections on PENDF
6/ process all matrices
16/ process all photon production reactions
0/ end of this temperature
0/ end of groupr input

```

A clearer understanding of these input lines can be obtained by studying the GROUPT input specifications in the online version, a printed NJOY manual, or the files in the NJOY distribution. The notes after the “slash” terminator explain most of the features of this input deck.

Because not all users need all possible reactions, the automatic input process skips over thermal data (MT=221–250) and delayed neutron data (MT=455). If you need thermal data to be group averaged, include something like this in your deck:

```
3/ all reactions but thermal
3 229 'graphite inelastic'/
3 230 'graphite elastic'/
6/ all matrices but thermal
6 229 'graphite inelastic'/
6 230 'graphite elastic'/
```

Similarly, if you need delayed-neutron production and spectra, include something like this in your deck:

```
3 455/ delayed nubar
5 455/ delayed spectra
```

The code will automatically provide spectra for all the six time groups of delayed neutrons. As discussed in the previous subsection, it is the responsibility of a subsequent code to merge the delayed neutron data with the prompt data for the calculation of a proper fission source.

As a final example, to obtain self-shielded cross sections and scattering matrices, you have to ask for several temperatures and several σ_0 values. It is not necessary to reprocess all the reactions at the higher temperatures; just include the reactions that occur at resonance energies:

```
groupr
20 21 0 22
1050 3 0 3 3 2 3 1
'94-pu-238 from ENDF/B-IV'/
300 900
1e10 1e4 1000 100 10 1
3/ all reactions at 300K
6/ all matrices at 300K
0/ end of first temperature (300K)
3 1 'total'/ only resonance reactions at 900K
3 2 'elastic'/
3 18 'fission'/
3 102 'capture'/
6 2 'elastic'/ yes, you can self shield the elastic matrix
0/ end of second temperature (900K)
0/ end of groupr deck
```


GROUPR has other capabilities, including custom group structures, custom weight functions, the flux calculator, charged particle transfer matrices, and activation cross section processing, that can be explored in the detailed documentation.

10 Monte Carlo Libraries

In these days of inexpensive fast computers with huge memories, people are making more and more use of detailed Monte Carlo transport calculations. This approach is much more expensive than the multigroup approach, but it has two great advantages:

- detailed cross sections don't have to be approximated, and
- simplified models of the geometry don't have to be used.

There are Monte Carlo codes (*e.g.*, MORSE) that use multigroup data, and NJOY's multigroup data can be reformatted for these codes, but the most faithful physics modeling can be obtained by using a full continuous-energy Monte Carlo code like MCNP from Los Alamos. NJOY can produce libraries for the MCNP code by using its ACER module. Here is an example of how we normally do this using a unix script:

```

echo "ACE production and testing run:"
echo "getting files"
cfs get tape20:/endf/6u/neutron/c/nata [This is an existing ENDF tape]
cfs get tape21:/pendf/6u/c/nata [This is an existing PENDF tape]
cat>input <<EOF
acer
20 21 0 31 32
1 0 1/ fast data, no listing, type-1 format (ASCII)
'6-c-nata from endf-vi.1'/
600 296.3/ MAT and temperature
.01/ thin distributions to 1%
/
acer
0 31 33 34 35 / file33 is plotting output
7 1 2/ read type-1 file, make listing, write type-2 file (binary)
'6-c-nata from endf-vi.1'/
viewr
33 36/ convert the acer plot output to postscript
stop
EOF
echo "running njoy"
xnjoy<input
echo "saving njoy output"
cfs store dir=/acer/6u/f/c output:natao tape31:nata tape32:natax
cfs store dir=/acer/6u/f/c tape33:natav tape36:natap
echo "creating local xsdir file for material test ..."
cfs get /acer/xsdir2
xsdir2<<EOF
/acer/6u/f/c/nata

```

```

EOF
echo "creating mcnp inp file for material test ..."
cat>inp <<EOF
  material test
  1 1 .05 -1
  2 0 1

  1 so 1

  imp:n 1 0
  fcl:n 1 0
  mode n p
  fcl:p 1 0
  sdef erg=d1
  sil 1.e-8 .000001 .0001 .01 .1 1 18i 20
  sp1 0 1 23r
  nps 100000
  phys:p j 1
  ml 6000 1
EOF
echo "running mcnp ..."
mcnp
echo "cleaning up ..."
cfs store outp:/acer/6u/f/c/natar

```

The run uses a previously prepared PENDF tape. The first ACER section generates the ACE library and a file containing a line for a new "XSDIR" file, MCNP's index file for data. The second ACER section reads and prints this ACE file as an integrity check. It also makes a number of consistency checks on the file, and it prepares an extensive series of plots showing the cross sections and the distributions of secondary particles. The VIEWR section prepares these plots in Postscript for printing or for viewing on your screen. Finally, the script make an quick MCNP run on the new file as a final quality check. The "cfs" lines read and write from a Los Alamos archival storage system, and they can be changed to satisfy local requirements.

11 What Else Do You Need to Know?

This tutorial is necessarily brief, and it is slanted toward reactor applications. More details and a number of exercises designed to demonstrate important features of NJOY are available in the online version of "Understanding NJOY" at <http://t2.lanl.gov/njoy>. The tutorial is based on NJOY 97.107. Official versions of NJOY must be obtained from one of the official code centers, such as RSICC at Oak Ridge, or NEA-DB in Paris. Updates and other support information are available on line at <http://t2.lanl.gov/codes/njoy97>. The printed NJOY manual provided with the distribution is not quite up to date, but it provides a detailed description of the code. Just watch out for a few changes in input specifications.