8 GROUPR

GROUPR computes group-to-group scattering matrices, and anisotropic photon production matrices for neutrons from ENDF/B-IV and later evaluated nuclear data. With ENDF-6 format files, photonuclear data and incoming and outgoing charged particles can also be handled. Special features are provided for ratio quantities (for example, $\overline{\mu}$, $\overline{\nu}$, or photon yield), inverse velocity, delayed neutron spectra by time group, and anisotropic thermal neutron scattering. Fission is represented as a group-to-group matrix for full generality. Scattering matrices and photon production matrices may be self-shielded if desired.

The Bondarenko narrow-resonance weighting scheme[39] is normally used. Optionally, a weighting flux can be computed for various mixtures of heavy absorbers with light moderators. An accurate pointwise solution of the integral slowing down equation is used. This option is normally called on to account for intermediate resonance effects in the epithermal range.

Neutron data and photon-production data are processed in a parallel manner using the same weight function and quadrature scheme. This assures consistent cross sections for coupled neutron-photon problems. Two-body scattering is computed with a center-of-mass (CM) Gaussian quadrature, which gives accurate results even for small Legendre components of the group-to-group matrix.

User conveniences include free-form input and complete control over which reactions are processed. The neutron group structure, photon group structure, and weight function can each be read in or set to one of the internal options. Output can be printed and/or written to an output "groupwise-ENDF" (GENDF) file for further processing by a formatting module (DTFR, CCCCR, MATXSR, WIMSR), by the covariance module (ERRORR), or by the MCNP continuous-energy Monte Carlo module (ACER).

This chapter describes the GROUPR module in NJOY2016.0.

8.1 Multigroup Constants

Multigroup constants are normally used by computer codes that calculate the distributions of neutrons and/or photons in space and energy, and that compute various responses to these distributions, such as criticality, dose to personnel, or activation of materials. These distributions are solutions of the neutral particle transport equation.[5]

[5]The following development uses a notation based on Bell and Glasstone[53], where the lower-case sigma is used for both macroscopic and microscopic cross sections, depending on the context. One-dimensional

$$\begin{aligned} \mu\frac{\partial}{\partial x}\phi(x,\mu,E) &+ \sigma_t(x,E)\,\phi(x,\mu,E) \\ &= \int d\mathbf{\Omega}' \int dE'\,\sigma_X(x,E'\to E,\mathbf{\Omega}'\to\mathbf{\Omega})\,\phi(x,\mu',E') \\ &+ Q(x,\mu,E)\ , \end{aligned} \tag{240}$$

where the flux ϕ is allowed to vary with position x, direction $\mathbf{\Omega}$ with polar cosine μ, and energy E. Similarly, the macroscopic total cross section σ_t varies with position and energy. The right-hand side of the equation contains the source due to transfers from other directions $\mathbf{\Omega}'$ and energies E' (as described by the macroscopic transfer cross section σ_X), and a fixed or external source Q.

The macroscopic cross sections (in units of cm^{-1}) in Eq. 240 can be calculated from microscopic cross sections for the component isotopes or elements (in barns) using

$$\sigma_t(x,E) = \sum_i \rho_i(x)\,\sigma_t^i(T[x],E)\ , \tag{241}$$

where ρ_i is the number density for a constituent (in $\text{barns}^{-1}\text{cm}^{-1}$), which may vary with position, and T is the temperature, which may also vary with position. A similar formula holds for σ_X.

The transfer cross section σ_X (which includes both scattering and fission processes) is normally assumed to depend only on the cosine of the scattering angle, $\mu_0 = \mathbf{\Omega}\cdot\mathbf{\Omega}'$. This allows σ_X to be expanded using Legendre polynomials

$$\sigma_X(x,E'\to E,\mathbf{\Omega}'\to\mathbf{\Omega}) = \sum_{\ell=0}^{\infty}\frac{2\ell+1}{4\pi}\,\sigma_{X\ell}(x,E'\to E)\,P_\ell(\mu_0)\ . \tag{242}$$

Application of the addition theorem and integration over azimuthal angle then gives

$$\begin{aligned} \mu\frac{\partial}{\partial x}\phi(x,\mu,E) &+ \sigma_t(x,E)\,\phi(x,\mu,E) \\ &= \sum_{\ell=0}^{\infty}\frac{2\ell+1}{2}P_\ell(\mu)\int \sigma_{X\ell}(x,E'\to E)\,\phi_\ell(x,E')\,dE' \\ &+ Q(x,\mu,E)\ , \end{aligned} \tag{243}$$

slab geometry is used throughout for simplicity.

where

$$\phi_\ell(x,E) = \int P_\ell(\mu)\,\phi(x,\mu,E)\,d\mu \ . \tag{244}$$

The desired responses are then given by

$$R(x) = \int \sigma_r(x,E)\,\phi_0(x,E)\,dE \ , \tag{245}$$

where σ_r is the reaction cross section for the response. The next step is to integrate Eqs. 243 and 245 over a range of energies chosen to lie in group g. The results are

$$\begin{aligned} \mu\frac{\partial}{\partial x}\phi_g(x,\mu) \quad &+ \quad \sum_{\ell=0}^{\infty} P_\ell(\mu)\,\sigma_{t\ell g}(x)\,\phi_{\ell g}(x) \\ &= \quad \sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} P_\ell(\mu) \sum_{g'} \sigma_{X\ell g'\to g}(x)\,\phi_{\ell g'}(x) + Q_g(x,\mu) \ , \end{aligned} \tag{246}$$

and

$$R(x) = \sum_g \sigma_{rg}(x)\,\phi_{0g}(x) \ , \tag{247}$$

where

$$\phi_{\ell g}(x) = \int_g \phi_\ell(x,E)\,dE \ , \tag{248}$$

$$\sigma_{t\ell g}(x) = \frac{\displaystyle\int_g \sigma_t(x,E)\,\phi_\ell(x,E)\,dE}{\displaystyle\int_g \phi_\ell(x,E)\,dE} \ , \tag{249}$$

$$\sigma_r(x) = \frac{\displaystyle\int_g \sigma_r(x,E)\,\phi_0(x,E)\,dE}{\displaystyle\int_g \phi_0(x,E)\,dE} \ , \tag{250}$$

and

$$\sigma_{X\ell g'\to g} = \frac{\displaystyle\int_g dE \int_{g'} dE'\,\sigma_{X\ell}(x,E'\to E)\,\phi_\ell(x,E)}{\displaystyle\int_g \phi_\ell(x,E)\,dE} \ . \tag{251}$$

The last three equations provide the fundamental definitions for the multi-group cross sections and the group-to-group matrix. Note that the values of the group constants depend upon the basic energy-dependent cross sections obtained from an ENDF-format evaluation by way of the RECONR, BROADR, UNRESR, HEATR, THERMR, and PURR modules of NJOY, and the shape of ϕ within the group.

8.2 Group Ordering

Since neutrons normally lose energy in scattering, the scattering source into group g depends on the flux at higher-energy groups g' and the cross section for transferring neutrons from g' to g. For this reason, Eq. 246 is usually solved by sweeping from high energies to low energies. (Any thermal upscatter or fission is handled by iteration.) Data libraries for use with transport codes normally number the groups such that group 1 is the highest-energy group, and all the scattering matrix elements that transfer neutrons into group 1 are given first, followed by those for scattering into group 2, and so on.

However, in ENDF files the evaluated nuclear data are always given in order of increasing incident energy, and secondary neutron distributions are described by giving emission spectra for given incident energies. Therefore, GROUPR numbers its groups such that group 1 is the lowest-energy group, and it calculates that scattering out of group 1, followed by the scattering out of group 2, and so on.

The "backward" energy-group numbering convention used by GROUPR is a possible source of confusion in interpreting output produced by the various modules of NJOY. All group indices printed by GROUPR or written to the GROUPR output file use the increasing-energy order. The covariance modules and photon interaction module follow the GROUPR ordering convention. Output modules such as DTFR, CCCCR, MATXSR, and WIMSR invert the group order and rearrange the scattering matrices from the GROUPR outscatter organization to the "transport" inscatter form. Any group indices printed by these three output modules will be in the conventional transport decreasing-energy order.

8.3 Basic ENDF Cross Sections

The basic energy-dependent cross sections and energy-angle distributions needed for Eqs. 249, 250, and 251 are obtained from evaluated nuclear data in ENDF format[9]. These data are indexed by material (MAT), type of information (MF),

and reaction (MT). Materials can be single isotopes, elements, or compounds. Type of information includes energy-dependent cross section (MF=3), angular distributions (MF=4), secondary energy distributions (MF=5), and energy-angle distributions (MF=6). Reactions include the total (MT=1) required for Eq. 249, the partial scattering reactions that must be summed for Eq. 251 [that is, elastic (MT=2), discrete-level inelastic (MT=51 – 90), (n,2n) (MT=16), *etc.*], and the many partial reactions that can be used to calculate responses [for example, (n,2n) or (n,α) for activation, gas production, heat production (KERMA), radiation damage (DPA), *etc.*].

Before using GROUPR, the basic ENDF/B cross sections should have been converted into energy- and temperature-dependent pointwise cross sections in PENDF (pointwise ENDF) format using RECONR and BROADR. For heavy isotopes, unresolved self-shielding data should have been added to the PENDF tape[6] using UNRESR or PURR. If requested, heat production cross sections (KERMA), radiation damage production (or DPA), and thermal upscatter data can also have been added to the PENDF tape using HEATR and THERMR. See the chapters on these other modules for more information.

The detailed methods used for evaluating Eqs. 249, 250, and 251 from ENDF and PENDF tapes are given below (see Sections 8.11 through 8.15). An example of an ENDF pointwise cross section compared with group-averaged cross sections from GROUPR is given in Fig. 22.

8.4 Weighting Flux

In general, the weighting flux ϕ is not known; it is, after all, the particle distribution being sought in the transport calculation. However, it is often possible to obtain fairly accurate group constants for a particular application if the shape of the flux is reasonably well known over the broad energy ranges of a particular few-group structure (for example, a fission spectrum, thermal Maxwellian, or $1/E$ slowing-down spectrum). Alternatively, one can use many small groups so that mistakes in guessing the shape inside the group are not very important. The key to using the multigroup method effectively is balancing the tradeoffs between the choice of weight function and the number of groups used for each different class of problem being solved.

In many cases of practical interest, the flux ϕ will contain dips corresponding to the absorption resonances of the various materials. In the reaction rate

[6]The term "tape" is used loosely in this report to refer to any input or output file. Of course, such files would be on disk storage in a modern system.

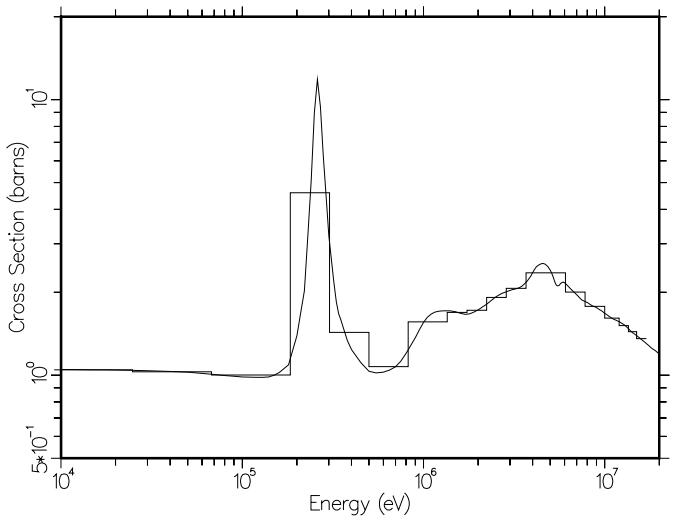

Figure 22: A comparison of the pointwise and multigroup representations for the total cross section of ^{7}Li. The Los Alamos 30-group structure is shown.

$\sigma(E) \times \phi(E)$, these dips clearly reduce (self-shield) the effect of the corresponding resonance. GROUPR provides two methods to estimate the effect of this self-shielding: the Bondarenko model and the flux calculator.

In the Bondarenko model[39], the narrow resonance (NR) approximation, and the B_N approximation for large systems[53] are invoked to obtain

$$\phi_\ell(E) = \frac{W_\ell(E)}{[\sigma_t(E)]^{\ell+1}} \ , \tag{252}$$

where ϕ_ℓ is the ℓ-th Legendre component of the angular flux, the $W_\ell(E)$ are smooth functions of energy (such as $1/E$+fission), and $\sigma_t(E)$ is the total macroscopic cross section for the material. GROUPR takes all of the W_ℓ to be equal to the single function $C(E)$, where $C(E)$ can be read in or set to one of several internally defined functions. It is further assumed that the important self-shielding effect of the flux can be obtained for isotope i by representing all the other isotopes with a constant "background cross section", σ_0. Therefore,

$$\phi^i_\ell(E) = \frac{C(E)}{[\sigma^i_t(E) + \sigma^i_0]^{\ell+1}} \ , \tag{253}$$

where σ^i_t is the microscopic total cross section for isotope i. The qualitative behavior of Eq. 253 is easy to understand. If σ_0 is larger than the tallest peaks in σ_t, the weighting flux ϕ 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.

Each component material of a mixture has a different weight function. The macroscopic total cross sections are given by

$$\sigma_{t\ell g} = \sum_i \rho_i \, \sigma^i_{t\ell g}(\sigma^i_0, T) \ , \tag{254}$$

where

$$\sigma^i_{t\ell g}(\sigma_0, T) = \frac{\displaystyle\int_g \frac{\sigma^i_t(E,T)}{[\sigma_0 + \sigma^i_t]^{\ell+1}} C(E)\, dE}{\displaystyle\int_g \frac{1}{[\sigma_0 + \sigma^i_t]^{\ell+1}} C(E)\, dE} \ . \tag{255}$$

Similar equations are used for σ_R and $\sigma_{X\ell g\to g'}$. On the GROUPR level, σ_0 and T are simply parameters. Subsequent codes, such as 1DX[54] or TRANSX[12, 41], can compute an appropriate value for σ_0, and then interpolate in tables of cross section versus σ_0 and T to get the desired self-shielded group constants.

The appropriate value for σ^i_0 is obvious when a single resonance material is mixed with a moderator material (for example, $^{238}UO_2$), because the admixed materials typically have a constant cross section in the energy range where the heavy isotopes have resonances. For a mixture of resonance materials, the normal procedure is to preserve the average of Eq. 253 in each group by using

$$\sigma^i_{0g} = \frac{1}{\rho_i} \sum_{j\neq i} \rho_j \, \sigma^j_{t0g}(\sigma^j_{0g}, T) \ , \tag{256}$$

where the ρ_i are atomic densities or atomic fractions. Eq. 256 is solved by iteration. Interference between resonances in different materials is handled in an average sense only.

In the unresolved energy range (see UNRESR and PURR), the explicit dependence of cross section on energy is not known. The integrands are replaced by their expected values

$$\sigma_{x\ell g}(\sigma_0) = \frac{\displaystyle\int_g \left\langle \frac{\sigma_x}{[\sigma_0 + \sigma_t]^{\ell+1}} \right\rangle W_\ell \, dE}{\displaystyle\int_g \left\langle \frac{1}{[\sigma_0 + \sigma_t]^{\ell+1}} \right\rangle W_\ell \, dE} \ , \tag{257}$$

where the expected values are averages over the distributions of resonance posi-

tion and width expected in the vicinity of energy E. The UNRESR and PURR modules produce effective self-shielded point cross sections defined by

$$\langle \sigma_x \rangle_\ell = \frac{\left\langle \frac{\sigma_x}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle}{\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle} \ . \tag{258}$$

Substituting Eq. 258 into Eq. 257 gives an equation of the form of Eqs. 249 and 250, except that σ is replaced by $\langle\sigma\rangle$, and the flux is replaced by an average effective flux. This effective flux can be obtained by manipulating the effective total cross section as follows:

$$\langle \sigma_t \rangle_\ell = \frac{\left\langle \frac{\sigma_0+\sigma_t-\sigma_0}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle}{\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle} = \frac{\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell}} \right\rangle}{\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle} - \sigma_0 \ , \tag{259}$$

from which

$$\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle = \frac{\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell}} \right\rangle}{\sigma_0 + \langle \sigma_t \rangle_k} \ . \tag{260}$$

Eq. 260 defines a recursion relation which can be used to compute the effective flux to any order

$$\left\langle \frac{1}{[\sigma_0+\sigma_t]^{\ell+1}} \right\rangle = \prod_{k=0}^{\ell} \frac{1}{\sigma_0 + \langle \sigma_t \rangle_k} \ , \tag{261}$$

This equation reduces to Eq. 253 in the resolved range. It is the formula used in `genflx` to compute $\phi_\ell(E)$ for the Bondarenko option.

When heterogeneity effects are important, the background cross section method can be extended as follows. In an infinite system of two regions (fuel and moderator), the neutron balance equations are

$$V_f \sigma_f \phi_f = (1-P_f) V_f S_f + P_m V_m S_m \ , \tag{262}$$

and

$$V_m \sigma_m \phi_m = P_f V_f S_f + (1-P_m) V_m S_m \ , \tag{263}$$

where V_f and V_m are the region volumes, σ_f and σ_m are the corresponding total

macroscopic cross sections, S_f and S_m are the sources per unit volume in each region, P_f is the probability that a neutron born in the fuel will suffer its next collision in the moderator, and P_m is the probability that a neutron born in the moderator will suffer its next collision in the fuel. As usual, use is made of the reciprocity theorem,

$$V_f \sigma_f P_f = V_m \sigma_m P_m \ , \tag{264}$$

and the Wigner rational approximation to the fuel escape probability,

$$P_f = \frac{\sigma_e}{\sigma_e + \sigma_f} \ , \tag{265}$$

where σ_e is a slowly varying function of energy called the escape cross section, to obtain an equation for the fuel flux in the form

$$(\sigma_f + \sigma_e)\phi_f = \frac{\sigma_e S_m}{\sigma_m} + S_f \ . \tag{266}$$

In the limit where the resonances are narrow with respect to both fuel and moderator scattering, the source terms S_f and S_m take on their asymptotic forms of σ_p/E and σ_m/E respectively, and this equation becomes equivalent to the Bondarenko model quoted above with

$$\sigma_0^f = \frac{\sigma_e}{\rho_f} \ , \tag{267}$$

and

$$C(E) = \frac{\sigma_e + \sigma_p}{\rho_f E} \ . \tag{268}$$

Note that a large escape cross section (a sample that is small relative to the average distance to collision), corresponds to infinite dilution as discussed above. To illustrate the general case, consider a neutron traveling through a lump of uranium oxide with an energy close to a resonance energy. If the neutron scatters from an oxygen nucleus, it will lose enough energy so that it can no longer react with the uranium resonance. Similarly, if the neutron escapes from the lump, it can no longer react with the uranium resonance. The processes of moderator scattering and escape are equivalent in some way. Comparing Eq. 267 with Eq. 256 gives an "equivalence principle" that says that a lump of particular dimensions and a mixture of particular composition will have the same self-shielded cross sections when the narrow resonance approximation is valid. The

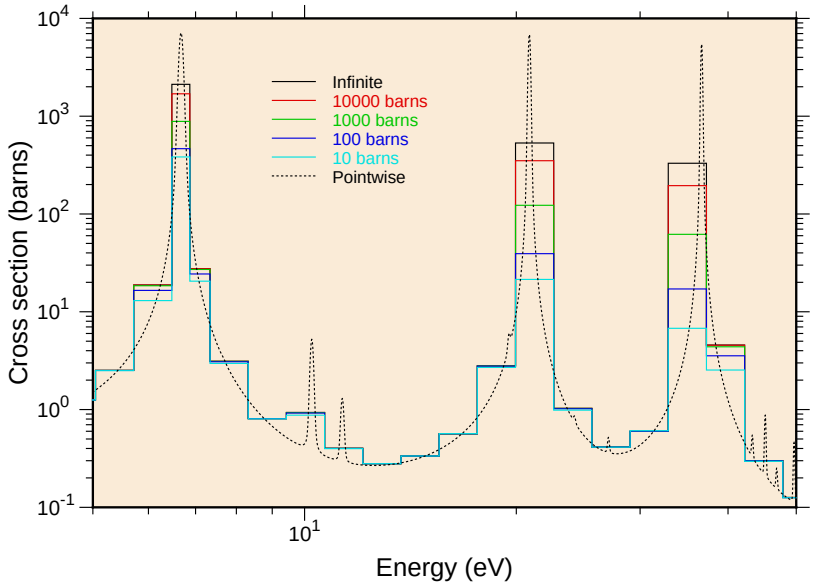

Figure 23: The self-shielding effect on the first three ^{238}U capture resonances at room temperature in the 5 to 50 eV range. The multigroup boundaries are from the Los Alamos 187-group structure.

effects of material mixing and escape can simply be added to obtain the effective σ_0 for a lump containing admixed moderator material. Therefore, Eq. 256 is extended to read

$$\sigma_{0g}^{i} = \frac{1}{\rho_i}\left\{\sigma_e + \sum_{j\neq i} \rho_j \, \sigma_{t0g}^{j}(\sigma_{0g}^{j}, T)\right\} , \tag{269}$$

where the escape cross section for simple convex objects (such as plates, spheres, or cylinders) is given by $(4V/S)^{-1}$, where V and S are the volume and surface area of the object, respectively. Many codes that use the background cross section method modify the escape cross section as defined above to correct for errors in the Wigner rational approximation ("Bell factor", "Levine factor"), or to correct for the interaction between different lumps in the moderating region ("Dancoff corrections"). These enhancements will not be discussed here.

As an example of self-shielded cross sections and how they vary with the background cross section, Fig. 23 shows the first three capture resonances of ^{238}U (which are very important for thermal power reactors) at room temperature for σ_0 values ranging from infinity down to 10 barns. Background cross sections that range between 20 and 50 barns are typical for uranium-oxide pin cells.

The BROADR chapter of this report showed its capability to compute stan-

dard resonance integrals. However, when self shielding and material temperature come into play, the effective resonance integral changes. GROUPR can calculate these quantities by doing one-group calculations with $1/E$ weighting over a standard energy range. Using the range 0.5 eV to the upper limit of the evaluation will match what BROADR does. Fig. 24 shows the temperature dependence of the capture resonance integral for ^{238}U from ENDF/B-VII at several different background cross sections. The range between 20 and 50 barns is typical for reactor pin cells.

8.5 Flux Calculator

This narrow-resonance approach is quite useful for practical fast reactor problems. However, for nuclear systems sensitive to energies from 1 to 500 eV, there are many broad- and intermediate-width resonances which cannot be self-shielded with sufficient accuracy using the Bondarenko approach. The GROUPR flux calculator is designed for just such problems.

Consider an infinite homogeneous mixture of two materials and assume isotropic scattering in the center-of-mass system. The integral slowing-down equation becomes

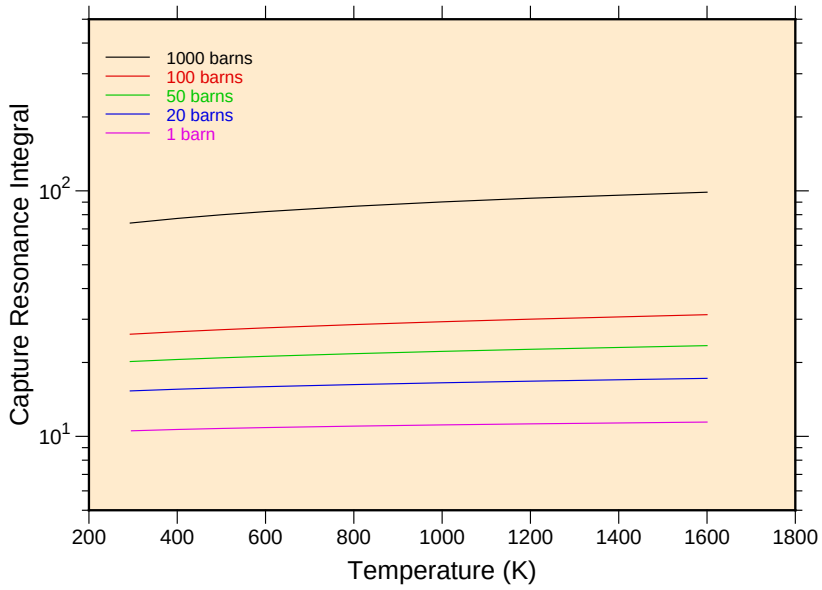

Figure 24: The capture resonance integral for ^{238}U showing its variation with temperature and background cross section σ_0. The resonance integral at infinite dilution is 275.58 barns. Note the slope with temperature, which helps to produce a negative temperature coefficient for uranium systems.

$$\sigma(E)\,\phi(E) = \int_E^{E/\alpha_1} \frac{\sigma_{s1}(E')}{(1-\alpha_1)E'}\,\phi(E')\,dE' + \int_E^{E/\alpha_2} \frac{\sigma_{s2}(E')}{(1-\alpha_2)E'}\,\phi(E')\,dE' \ . \tag{270}$$

Furthermore, assume that material 1 is a pure scatterer with constant cross section and transform to the σ_0 representation. The integral equation becomes

$$[\sigma_0 + \sigma_{t2}(E)]\,\phi(E) = \int_E^{E/\alpha_1} \frac{\sigma_0}{(1-\alpha_1)E'}\,\phi(E')\,dE' + \int_E^{E/\alpha_2} \frac{\sigma_{s2}(E')}{(1-\alpha_2)E'}\,\phi(E')\,dE' \ . \tag{271}$$

Finally, assume that the moderator (material 1) is light enough so that all the resonances of material 2 are narrow with respect to scattering from material 1. This allows the first integral to be approximated by its asymptotic form, $1/E$. More generally, the integral is assumed to be a smooth function of E given by $C(E)$. In this way, material 1 can represent a mixture of other materials just as in the Bondarenko method. Fission source and thermal upscatter effects can also be lumped in $C(E)$. The integral equation has now been reduced to

$$[\sigma_0 + \sigma_t(E)]\,\phi(E) = C(E)\,\sigma_0 + \int_E^{E/\alpha} \frac{\sigma_s(E')}{(1-\alpha)E'}\,\phi(E')\,dE' \ . \tag{272}$$

This is the simplest problem that can be solved using the flux calculator. The results still depend on the single parameter σ_0, and they can be used easily by codes that accept Bondarenko cross sections.

For heterogeneous problems, when the narrow-resonance approximation fails, both S_f and S_m in Eq. 266 will show resonance features. To proceed further with the solution of this equation, it is necessary to eliminate the moderator flux that is implicit in S_m. As a sample case, consider a fuel pin immersed in a large region of water. The fission neutrons appear at high energies, escape from the pin, slow down in the moderator (giving a $1/E$ flux), and are absorbed by the resonances in the pin. In this limit, any dips in the moderator flux caused by resonances in the fuel are small. On the other hand, in a closely packed lattice, the flux in the moderator is very similar to the flux in the fuel, and

resonance dips in the moderator flux become very evident. Intermediate cases can be approximated[55] by assuming

$$\phi_m = (1-\beta)\, C(E) + \beta\phi_f \ , \tag{273}$$

where β is a heterogeneity parameter given by

$$\beta = \frac{V_f \sigma_e}{V_m \sigma_m} \ . \tag{274}$$

Note that $\beta \to 0$ gives the isolated rod limit and $\beta \to 1$ gives the close-packed lattice limit. This substitution reduces the calculation of the fuel flux to

$$(\sigma_f + \sigma_e)\, \phi_f = (1-\beta)\, C(E)\, \sigma_e + S_\beta \ , \tag{275}$$

where S_β is the source term corresponding to a homogeneous mixture of the fuel isotopes with the isotopes from the moderator region changed by the factor $\beta\sigma_e/\sigma_m$. If the fuel and moderator each consisted of a single isotope and for isotropic scattering in the center-of-mass system, the integral equation would become

$$\begin{aligned} [\sigma_0 + \sigma_t(E)]\, \phi_f(E) &= (1-\beta)\, C(E)\, \sigma_0 \\ &+ \int_E^{E/\alpha_m} \frac{\beta\sigma_0}{(1-\alpha_m)E'}\, \phi_f(E')\, dE' \\ &+ \int_E^{E/\alpha_f} \frac{\sigma_{sf}(E')}{(1-\alpha_f)E'}\, \phi_f(E')\, dE' \ , \end{aligned} \tag{276}$$

where σ_0 is σ_e divided by the fuel density (units are barns/atom), α_m and α_f are the maximum fractional energy change in scattering for the two isotopes, and $\sigma_{sf}(E')$ is the fuel scattering cross section.

This result has a form parallel to that of Eq. 272, but the solution depends on the two parameters β and σ_0. For any given data set, β must be chosen in advance. This might not be difficult if the data are to be used for one particular system, such as pressurized water reactors. The routine also has the capability to include one more moderator integral with a different α value and a constant cross section. The full equation is

$$
\begin{aligned}
[\sigma_0 + \sigma_t(E)]\,\phi_f(E) &= (1-\beta)\,C(E)\,\sigma_0 \\
&+ \int_E^{E/\alpha_3} \frac{\beta(1-\gamma)(\sigma_0-\sigma_{am}}{(1-\alpha_3)E'}\,\phi_f(E')\,dE' \\
&+ \int_E^{E/\alpha_2} \frac{\sigma_{am}+\beta\gamma(\sigma_0-\sigma_{am}}{(1-\alpha_2)E'}\,\phi_f(E')\,dE' \\
&+ \int_E^{E/\alpha_f} \frac{\sigma_{sf}(E')}{(1-\alpha_f)E'}\,\phi_f(E')\,dE' \;, \qquad (277)
\end{aligned}
$$

where σ_{am} is the cross section of the admixed moderator (with energy loss α_2), and γ is the fraction of the admixed moderator that is mixed with the external moderator (which has energy loss α_3). This allows calculations with H_2O as the moderator and an oxide as the fuel. The flux calculator can thus obtain quite realistic flux shapes for a variety of fuel, admixed moderator and external moderator combinations. An example comparing the Bondarenko flux with a more realistic computed flux is given in Fig. 25.

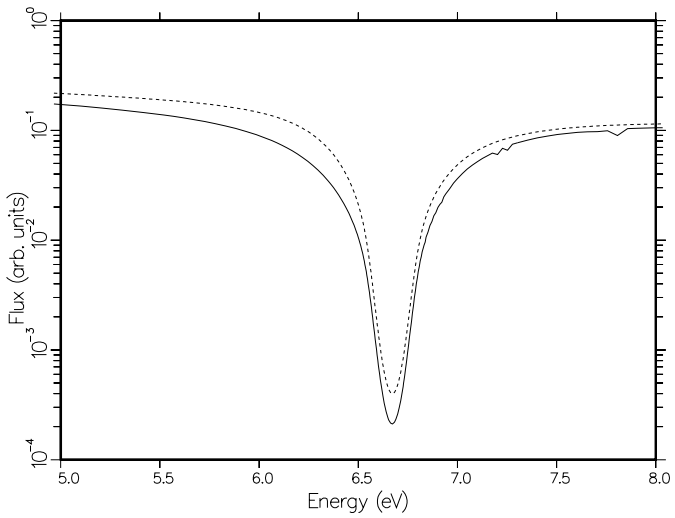

Figure 25: A comparison of the Bondarenko flux model (dashed) with a realistic computed flux (solid) for a ^{238}U oxide pin in water in the region of the 6.7 eV resonance.

In practice, a fuel rod rarely contains only one resonance isotope. As an example, consider a mixture of a few percent of ^{239}Pu with ^{238}U as the major component. There will be a strong dip in the flux associated with the 6.7 eV ^{238}U resonance that will affect the flux in the region of the 7.8 eV ^{239}Pu resonance (the interference effect), and there will also be a dip in the flux corresponding to the 7.8 eV resonance (the self-shielding effect). This additional complication

in the flux shape would be expected to change the group constants for ^{239}Pu since both features lie in the same group for typical group structures. However, the effect of the ^{239}Pu on the ^{238}U group constants should be minimal. This argument suggests that the full flux calculation be used for ^{238}U as a single resonance material. The resulting flux would then be used to estimate the flux to be used in averaging the ^{239}Pu cross sections as follows:

$$\phi_{239}(E,\sigma_0) = \frac{\phi_{238}(E,50)}{\sigma_0 + \dfrac{\sigma_{239}(E)}{1+\dfrac{\sigma_{238}(E)}{50}}} \ , \tag{278}$$

where the ^{238}U flux is characteristic of a background of 50 barns/atom, which is representative of many thermal reactor systems. This formula assumes that the effect of ^{239}Pu on the scattering source for the mixture is small, but it retains the absorption effects. The self-shielding of ^{239}Pu is treated in the narrow resonance approximation only. The GROUPR flux calculator includes an option to write out a file containing the calculated flux and cross section needed for this formula (*e.g.*, for ^{238}U) and another option to skip the flux calculation and use the formula above to obtain the weighting flux (*e.g.*, for ^{239}Pu).

The slowing-down integral equation of Eq. 272 or 276 is solved point by point (see subroutine `genflx`) using the total and elastic cross sections on the PENDF tape produced by RECONR. In order to keep this task within bounds, the flux is computed from the lower limit of the first group to a specified energy `fehi` or until `nfmax` values have been computed. The flux at higher energies is continued using the Bondarenko model described above.

8.6 Fission Source

The fission source was included in the transfer cross section σ_X in the development above. It is usually convenient to separate fission from scattering. Assuming isotropy, the source of fission neutrons into group g is given by

$$S_g = \sum_{g'} \sigma_{fg'\to g}\, \phi_{0g'} \ , \tag{279}$$

where the group-to-group matrix for fission is defined as in Eq. 251, but with ℓ equal to zero. Most existing transport codes do not use this matrix form directly because the upscatter is expensive to handle and a reasonably accurate alternative exists. 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

$$S_g = \chi_g \sum_{g'} \overline{\nu}_{g'} \, \sigma_{fg'} \, \phi_{0g'} \ , \tag{280}$$

where $\overline{\nu}_g$ 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

$$\chi_g = \frac{\sum_{g'} \sigma_{fg' \to g} \, \phi_{0g'}}{\sum_{g} \sum_{g'} \sigma_{fg' \to g} \, \phi_{0g'}} \ . \tag{281}$$

The fission neutron yield is given by

$$\overline{\nu}_g = \sum_{g'} \sigma_{fg \to g'} \, / \, \sigma_{fg} \ . \tag{282}$$

Clearly, χ_g as given by Eq. 281 depends on the flux in the system of interest. The dependence is weak except for high incident energies, and a rough guess for ϕ_{0g} usually gives an accurate spectrum. When this is not the case, the problem can be iterated, or the full matrix representation can be used.

It is possible to take advantage of the weak energy dependence of the shape of the fission spectrum at low energies to reduce the time required to process fission data, and to reduce the size of the fission data on the output file. GROUPR determines a break energy from File 5 such that the fission spectrum is constant below this energy. It only has to make a single calculation of this spectrum χ_g^{LE}. Then it computes a fission neutron production cross section σ_{fPg}^{HE} for groups below the break energy using the normal cross section processing methods. A full matrix $\sigma_{fg' \to g}^{HE}$ is computed for groups above the break point. As an example, consider using the GROUPR 187-group structure and finding that there are 130 constant groups. The 187×187 matrix is reduced to a 57×187 matrix, a 187-element spectrum vector, and a 130-group production vector, for a total reduction in size of 68%. The effective fission matrix is given by

$$\sigma_{fg' \to g} = \sigma_{fg' \to g}^{HE} + \chi_g^{LE} \sigma_{fPg'}^{LE} \ . \tag{283}$$

The fission matrix computed by GROUPR represents the prompt part of fission only. The delayed component of fission is represented by a delayed-neutron

yield $\overline{\nu}_g^D$, decay constants for six (or sometimes 8) time groups, λ_i^D, and emission probability spectra for six (or 8) time groups, χ_{ig}^D. Steady-state fission can be obtained using

$$\overline{\nu}_g^{SS} = \overline{\nu}_g + \overline{\nu}_g^D \ , \tag{284}$$

and

$$\chi_g^{SS} = \frac{\sum\limits_{g'} \sigma_{fg'\to g}\, \phi_{0g'} + \chi_g^D \sum\limits_{g'} \overline{\nu}_{g'}^D\, \sigma_{fg'}\, \phi_{0g'}}{\sum\limits_{g}\sum\limits_{g'} \sigma_{fg'\to g}\, \phi_{0g'} + \sum\limits_{g'} \overline{\nu}_{g'}^D\, \sigma_{fg'}\, \phi_{0g'}} \ , \tag{285}$$

where

$$\chi_g^D = \sum_i \chi_{ig}^D \ . \tag{286}$$

Note that χ_g^D sums to unity, but the χ for each time group sums to the fraction of the delayed neutron yield that appears in that time group.

In ENDF-format files, the total fission reaction is represented by MT=18. Important isotopes also give the partial fission reactions (n,f), (n,n′f), (n,2nf), and sometimes (n,3nf) using MT=19, 20, 21, and 38 respectively. The MT=18 representation is adequate for most fission reactor applications, but the partial reactions should be processed for applications with significant flux above 6 MeV. Caution: although the cross section for MT=18 equals the sum of its parts, the group-to-group fission matrix $\sigma_{fg\to g'}$ computed from MT=18 will not, in general, equal the sum of the partial matrices for MT=19, 20, 21, and 38 above the 6-MeV threshold for second-chance fission. The breakup into partial fission matrices has not been used in recent ENDF/B-VII evaluations. The delayed neutron data are given in MT=455. Sample input instructions for processing the various combinations of fission reactions used in ENDF/B will be found in Section 8.14. GROUPR outputs all the components of fission separately in order to give succeeding modules or codes complete flexibility.

8.7 Diffusion Cross Sections

The diffusion equation is often used in reactor physics calculations. Starting with Eq. 246, use the Legendre expansion for ϕ_g in the derivative term, and make use of the recursion relation for $\mu P_\ell(\mu)$ and the orthogonality relation for the Legendre polynomials to obtain the transport equation in spherical-harmonic

form

$$\frac{n+1}{2n+1}\frac{\partial\phi_{n+1,g}}{\partial x} + \frac{n}{2n+1}\frac{\partial\phi_{n-1,g}}{\partial x}\,\sigma_{tng}\,\phi_{ng} = \sum_{g'}\sigma_{sng'\to g}\,\phi_{ng'} + S_{fg} + Q_{ng}\ , \tag{287}$$

where the transfer term has been separated into a scattering term with cross section σ_s, and a fission source term S_f. When this set of equations is truncated at $n = \mathrm{N}$, the results are usually called the "$\mathrm{P_N}$ equations". For now, all terms with $n > 1$ are dropped, and Q is assumed to be isotropic. Thus,

$$\frac{\partial\phi_{1g}}{\partial x} + \sigma_{t0g}\,\phi_{0g} = \sum_{g'}\sigma_{s0g'\to g}\,\phi_{0g'} + S_{fg} + Q_{0g}\ , \tag{288}$$

and

$$\frac{1}{3}\frac{\partial\phi_{0g}}{\partial x} + \sigma_{t1g}\,\phi_{1g} = \sum_{g'}\sigma_{s1g'\to g}\,\phi_{1g'}\ . \tag{289}$$

The second equation can be written in the form of Fick's Law as follows:

$$\phi_{1g} = -D_g\frac{\partial\phi_{0g}}{\partial x}\ , \tag{290}$$

$$D_g = \frac{1}{3}\,\frac{1}{\sigma_{t1g} - \sum_{g'}\sigma_{s1g'\to g}\,\phi_{1g'}\,/\,\phi_{1g}}\ , \tag{291}$$

where D_g is the diffusion constant. The term in the denominator of the second factor is the transport cross section for diffusion, σ_{trD}. Unfortunately, it depends on a fairly complete knowledge of the neutron current in the system, perhaps from a previous calculation. However, for many problems, σ_{trD} can be simplified by assuming that

$$\sum_{g'}\sigma_{s1g'\to g}\,\phi_{1g'} \approx \sum_{g'}\sigma_{s1g\to g'}\,\phi_{1g}\ , \tag{292}$$

with the result that

$$\sigma_{trD,g} = \sigma_{t1g} - \sum_{g'}\sigma_{s1g\to g'}\ , \tag{293}$$

or

$$\sigma_{trD,g} = \sigma_{t1g} - \overline{\mu}_g \, \sigma_{s0g} \ , \tag{294}$$

where $\overline{\mu}_g$ is the average scattering cosine for neutrons in group g. These forms depend only on the shape of the weighting flux within the group, as usual. Substituting for ϕ_{1g} in Eq. 288 gives

$$\frac{\partial}{\partial x}(-D_g \frac{\partial \phi_{0g}}{\partial x}) + \sigma_{0tg} \, \phi_{0g} = \sum_{g'} \sigma_{s0g' \to g} \, \phi_{0g'} + S_{fg} + Q_{0g} \ , \tag{295}$$

which is the standard diffusion equation in slab geometry. Neither the diffusion coefficient nor the transport cross section for diffusion is produced directly by GROUPR. However, the components such a $\sigma_{t\ell}$ and $\sigma_{s0g' \to g}$ are made available to subsequent modules.

8.8 Cross Sections for Transport Theory

The S_N (discrete ordinates) transport codes solve the equation

$$\begin{aligned} \mu \frac{\partial}{\partial x} \phi_g(\mu, x) \ &+ \ \sigma_g^{SN} \phi_g(\mu, x) \\ &= \ \sum_{\ell=0}^{N} \frac{2\ell+1}{2} P_\ell(\mu) \sum_{g'} \sigma_{s\ell g' \to g}^{SN}(x) \, \phi_{\ell g'} \\ &+ \ S_{fg} + Q_g(\mu, x) \ , \end{aligned} \tag{296}$$

where once again one-dimensional slab geometry has been used for simplicity.[7] By comparing Eq. 296 with Eq. 246, it is seen that the S_N equations require the following cross sections:

$$\sigma_{s\ell g' \to g}^{SN} = \sigma_{s\ell g' \to g}, \qquad g' \neq g \ , \tag{297}$$

and

$$\sigma_{s\ell g \to g}^{SN} = \sigma_{s\ell g \to g} - \sigma_{t\ell g} + \sigma_g^{SN} \ , \tag{298}$$

where σ_g^{SN} is not determined and can be chosen to improve the convergence of the S_N calculation. A particular choice of σ_g^{SN} gives rise to a "transport approximation", and various recipes are in use, such as:

[7] The following development is based on the work of Bell, Hansen, and Sandmeier[56].

Consistent-P approximation:

$$\sigma_g^{SN} = \sigma_{t0g} \ . \tag{299}$$

Inconsistent-P approximation:

$$\sigma_g^{SN} = \sigma_{t,N+1,g} \ . \tag{300}$$

Diagonal transport approximation:

$$\sigma_g^{SN} = \sigma_{t,N+1,g} - \sigma_{s,N+1,g\to g} \ . \tag{301}$$

Bell-Hansen-Sandmeier (BHS) or extended transport approximation:

$$\sigma_g^{SN} = \sigma_{t,N+1,g} - \sum_{g'} \sigma_{s,N+1,g\to g'} \ . \tag{302}$$

Inflow transport approximation:

$$\sigma_g^{SN} = \sigma_{t,N+1,g} - \frac{\sum_{g'} \sigma_{s,N+1,g'\to g}\, \phi_{N+1,g'}}{\phi_{N+1,g}} \ . \tag{303}$$

The first two approximations are most appropriate when the scattering orders above N are small. The inconsistent option removes most of the delta function of forward scattering introduced by the correction for the anisotropy of the total scattering rate, and should normally be more convergent than the consistent option. The diagonal and BHS recipes make an attempt to correct for anisotropy in the scattering matrix and are especially effective for forward-peaked scattering. The BHS form is more often used, but the diagonal option can be substituted when BHS produces negative values. The inflow recipe makes the N+1 term of the P_N expansion vanish, but it requires a good knowledge of the N+1 flux moment from some previous calculation. Inflow reduces to BHS for systems in equilibrium by detail balance (*i.e.*, the thermal region). In the absence of self-shielding (that is, $\sigma_0\to\infty$), the distinction between σ_{t1} and σ_{t0} disappears, and so does the distinction between the inconsistent-P and consistent-P options. Also note that the inflow and BHS versions of σ_g are equivalent to the definitions of $\sigma_{trD,g}$ given in Eqs. 291 and 293, respectively, when N=0.

The transport-corrected cross sections are not computed directly by GROUPR, but the components needed are written to the GROUPR output file for use by subsequent modules.

8.9 Photon Production and Coupled Sets

Photon transport is treated with an equation similar to Eq. 246, except that the flux, cross sections, and groups are all referred to photon interactions and photon energies instead of to the corresponding neutron quantities. Methods of producing the photon interaction cross sections are described in GAMINR. The external photon source Q_g depends on the neutron flux and the photon production cross sections through

$$Q_g(x,\mu) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} P_\ell(\mu) \sum_{g'} \sigma_{\gamma\ell g'\to g}(x)\, \phi_{\ell g'}(x) \ , \tag{304}$$

where $\sigma_{\gamma\ell g'\to g}$ is defined by Eq. 251 with X replaced by γ. The ENDF files define σ_γ using a combination of photon production cross sections (MF=13), photon yields (MF=12) with respect to neutron cross sections (MF=3), discrete lines (MF=12 and 13), and continuous γ distributions (MF=15). Methods for working with these representations will be discussed in more detail below.

The low-energy groups for fission and capture normally have photon emission spectra whose shapes do not change with energy. The same method used for reducing the size of the fission matrix (see Section 8.6) can be used for these photon production matrices. In mathematical form,

$$\sigma_{\gamma g'\to g} = \sigma^{HE}_{\gamma g'\to g} + s^{LE}_\gamma g \sigma^{LE}_{\gamma P g'} \ , \tag{305}$$

where $s^{LE}_\gamma g$ is the normalized emission spectrum, and $\sigma^{LE}_{\gamma P g}$ is the associated production cross section.

For many practical problems, it is convenient to combine the neutron and photon transport calculations into a single application of Eq. 246 where the photons are treated as additional groups of low-energy neutrons. Since (γ,n) events are not usually very important, the downscattering structure (see Section 8.2) of the transport calculation is preserved for both n→γ and n→n events. Cross sections for this application are called "coupled sets". Coupled sets are not produced directly by GROUPR, but the n→n and n→γ components are made available to the other modules where they can be combined with γ→γ cross sections from GAMINR. As an example, the MATXSR module can format data for the TRANSX code, which can, in turn, prepare coupled sets for use by standard transport codes such as ONEDANT[14] or PARTISN[17]. The normal arrange-

ment of data in a coupled set is shown schematically in Fig. 26.

8.10 Thermal Data

GROUPR uses the thermal data written onto the PENDF tape by the THERMR module. It does not process ENDF File-7 data directly. Three different representations of thermal scattering are used in ENDF.

For crystalline materials like graphite, coherent elastic (with zero energy change) scattering can take place. The cross section for this process shows well-defined Bragg edges at energies that correspond to the various lattice-plane spacings in the crystalline powder. As shown in THERMR, the angular dependence of the coherent elastic cross section can be written as

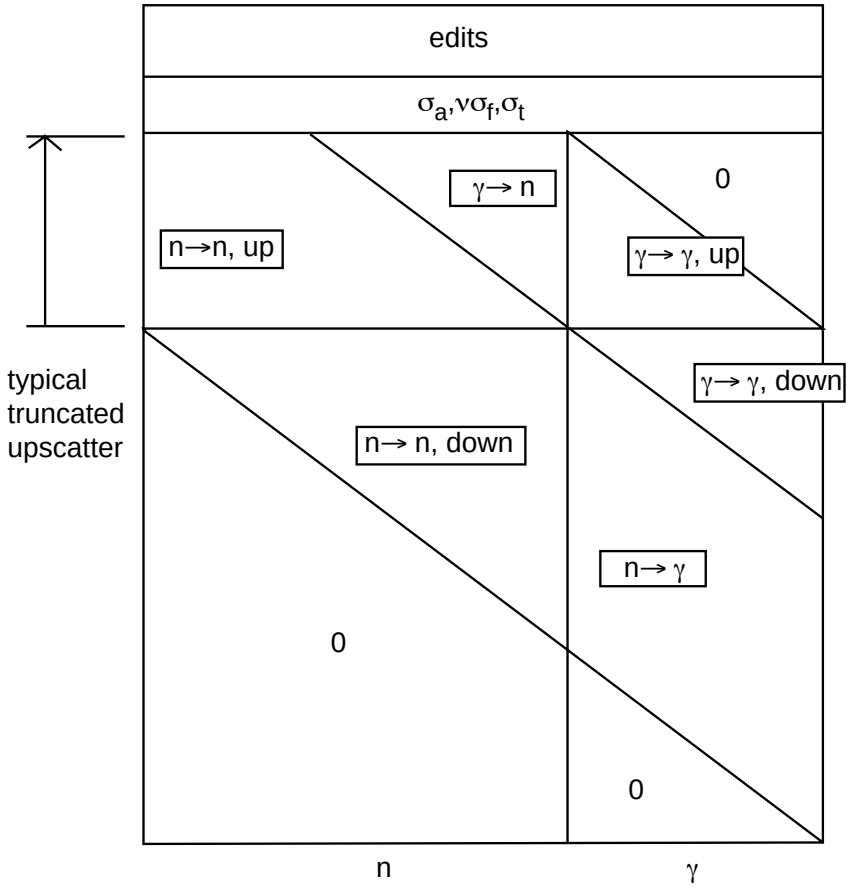

Figure 26: Arrangement of neutron (n) and photon (γ) cross sections in a coupled transport table. The group index increases from left to right, and the position index increases from top to bottom. The $\gamma \rightarrow$n and $\gamma \rightarrow \gamma$ upscatter blocks are normally empty.

$$\sigma^{coh}(E,\mu) = \sum_i \frac{b_i}{E}\,\delta(\mu-\mu_i)\ , \tag{306}$$

where

$$\mu_i = 1 - 2\frac{E_i}{E}\ , \tag{307}$$

and where the E_i are the energies of the Bragg edges. THERMR integrates Eq. 306 over all angles, and writes the result to the PENDF tape. Clearly, the b_i can be recovered from

$$E\,\sigma^{coh}(E) = {\sum_i}'\, b_i\ , \tag{308}$$

where the primed sum is over all i such that $E_i < E$. It is only necessary to locate the steps in $E\,\sigma^{coh}(E)$. The size of the step gives b_i, and the E for the step gives E_i. The Legendre cross sections become

$$\sigma_\ell^{coh}(E) = {\sum_i}'\, \frac{b_i}{E} P_\ell(\mu_i)\ , \tag{309}$$

where any terms with $\mu_i < -1$ are omitted from the primed sum. An example of a pointwise cross section for coherent elastic scattering is given in Fig. 27.

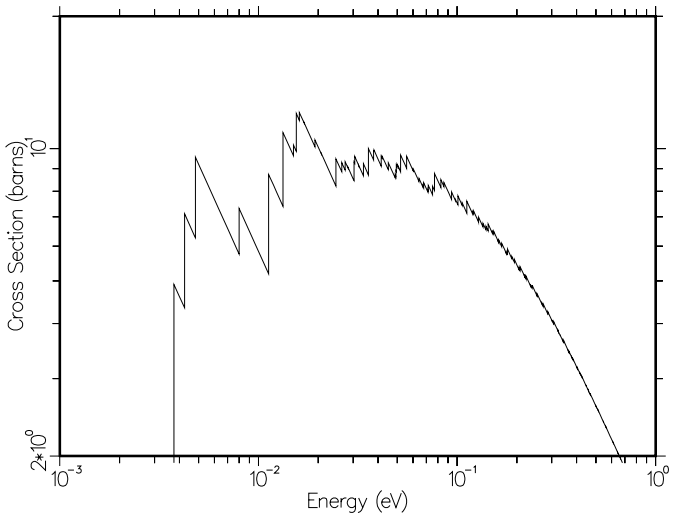

Figure 27: The coherent elastic scattering cross section for BeO showing the Bragg edges. The shape of $\sigma(E)$ between edges is $1/E$. Therefore, the function $E\sigma(E)$ is a stair-step function, where the height of each step depends on the structure factor for scattering from that set of lattice planes (see Eq. 308).

For hydrogenous solids like polyethylene and zirconium hydride, the process of incoherent elastic scattering is important. Here the angular cross section is given by

$$\sigma^{iel}(E,\mu) = \frac{\sigma_b}{2} \exp\left[-\frac{2EW}{A}(1-\mu)\right] . \tag{310}$$

THERMR converts this into an integrated cross section, $\sigma^{iel}(E)$, and a set of N equally probable emission cosines, $\overline{\mu}_i$. These angles are present in File 6 on the PENDF tape. GROUPR can easily determine the Legendre components of the scattering cross section using

$$\sigma_\ell^{iel}(E) = \sigma^{iel}(E)\frac{1}{N}\sum_{i=1}^{N} P_\ell(\mu_i) . \tag{311}$$

The third thermal process is incoherent inelastic scattering. Here the neutron energy can either increase or decrease. The data from THERMR are given as a cross section in File 3 and an energy-angle distribution using a special form of File 6. The distribution is represented by sets of secondary-energy values E' for particular incident energies E. For each $E \to E'$, a scattering probability $f^{inc}(E \to E')$ and a set of equally probable cosines $\mu_i(E \to E')$ are given. The scattering probabilities for each value of E integrate to unity. Although the thermal scattering cross section is a smooth function of incident neutron energy, this is not true for the scattering from E to one particular final energy group g', since the differential cross section tends to peak along the line $E'=E$ and at energy-transfer values corresponding to well-defined excitations in the molecule or lattice. If interpolation between adjacent values of E were to be performed along lines of constant E', the excitation peaks and the $E=E'$ feature would produce double features in the intermediate spectrum, as shown in Fig. 28. To avoid this problem, while still using a relatively sparse incident energy grid, GROUPR interpolates between E and E' along lines of constant energy transfer. Of course, this breaks down at low values of E', because one of the spectra will go to zero before the other one does. In this range, GROUPR transforms the low-energy parts of the two spectra onto a "unit base," combines them in fractions that depend on E, and scales the result back out to the interpolated value of E' corresponding to E.

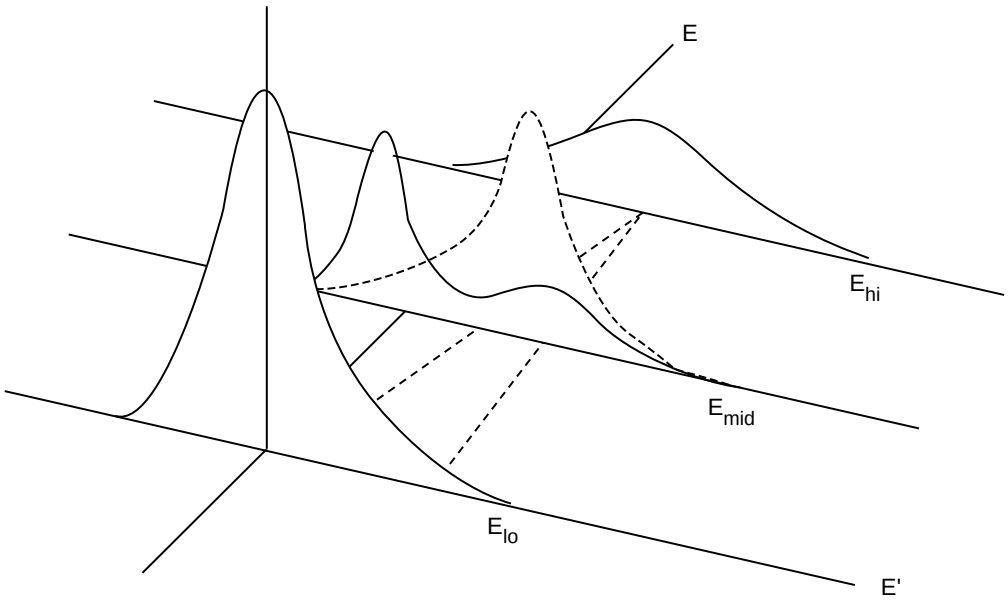

Figure 28: Illustration of thermal interpolation showing the double-humped curve resulting from simple Cartesian interpolation for a discrete excitation (solid) and the more realistic curve obtained by interpolating along lines of constant energy transfer (dashed).

8.11 Generalized Group Integrals

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

$$\sigma_g = \frac{\int_g \mathcal{F}(E)\,\sigma(E)\,\phi(E)\,dE}{\int_g \phi(E)\,dE} \ , \tag{312}$$

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. The function $\mathcal{F}(E)$ is called 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 $\overline{\mu}$ with respect to elastic scattering, $\mathcal{F}$ is set to $\overline{\mu}$. 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 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 of Eq. 312. 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 such that sigma can be obtained to within a given tolerance by linear interpolation. GROUPR 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 using the following Fortran:

```
 ...
call getsig(e,enext,...)
call getflx(e,en,...)
if (en.lt.enext) enext=en
call getff(e,en,...)
if (en.lt.enext) enext=en
 ...
```

On occasion, there will be a discontinuity at `enext` for one of the factors. In order to flag such a case, each "`get`" routine also sets a discontinuity flag `idisc`. The grid logic actually used throughout NJOY is as follows:

```
...
 call getsig(e,enext,idisc,...)
 call getflx(e,en,idis,...)
 if (en.eq.enext.and.idis.gt.idisc) idisc=idis
 if (en.lt.enext) idisc=idis
 if (en.lt.enext) enext=en
 call getff(e,en,idis,...)
 if (en.eq.enext.and.idis.gt.idisc) idisc=idis
 if (en.lt.enext) idisc=idis
 if (en.lt.enext) enext=en
 ...
```

This union grid for the integrand in the numerator is used to subdivide the generalized group integral of Eq. 312 into "panels". The main program of GROUPR carries out the integrals with the following logic:

```
 ...
 elo=egn(ig)
 ehi=egn(ig+1)
 enext=ehi
```

```
230 call panel(elo,enext,...)
    if (enext.eq.ehi) go to 240
    elo=enext
    enext=ehi
    go to 230
240 continue
    ...
```

A panel is first defined by the energy bounds of group g. Subroutine `panel` is called to sum in the contributions from this panel. However, `panel` discovers that the integrand has a grid point at `enext` less than `ehi`. It adds in the contributions for the smaller panel `elo` to `enext` and returns. GROUPR now sees that `enext` is less than `ehi`, so it tries a new panel from the top of the last panel (`elo`=`enext`) to `ehi`. This loop continues until a panel with `ehi` as its upper bound is summed in. The integral for this group is then complete.

In this simple way, the algorithm accounts for the user's group structure and for all structure in the integrand. The method used for establishing the RECONR grid makes this integration algorithm equivalent to adaptive integration as used in MINX[21]. It has the great advantage that no "stack" of intermediate results is carried along. This single-pass feature of the quadrature scheme allows many different integrals to be accumulated simultaneously within reasonable storage limits. In this way, GROUPR accumulates cross sections for all values of σ_0 simultaneously. Similarly, group-to-group cross sections are computed for all secondary energy groups and all Legendre orders simultaneously.

Any degree of complexity can be used for the integral over each subpanel. Because $\sigma(E)$ has been linearized, `panel` is based on trapezoidal integration. Both $\phi(E)$ and $R(E) = \sigma(E) \times \phi(E)$ are assumed to vary linearly across each panel. In some cases, the feed function is oscillatory over a certain energy range (see Two-Body Scattering, Section 8.12). It is then convenient to integrate inside the panel using Lobatto quadrature[57] (note the variable `nq` in `panel`). As discussed in more detail later, this method can obtain accurate results for an oscillatory function whose integral is small with respect to its magnitude. This behavior is characteristic of the higher Legendre components of two-body scattering cross sections.

The generalized integration scheme discussed here is also used by the GAMINR and ERRORR modules.

8.12 Two-Body Scattering

Elastic scattering (ENDF/B MT=2) and discrete inelastic neutron scattering (with MT=51 – 90) are both examples of two-body kinematics and are treated together by GROUPR. Some other cases that occur for charged particles or in File 6 will be discussed later. The feed function required for the group-to-group matrix calculation may be written

$$\mathcal{F}_{\ell g'}(E) = \int_{g'} dE' \int_{-1}^{+1} d\omega \, f(E{\to}E', \omega) \, P_\ell(\mu[\omega]) \ , \tag{313}$$

where $f(E{\to}E', \omega)$ 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

$$\mu = \frac{1 + R\omega}{\sqrt{1 + R^2 + 2R\omega}} \ , \tag{314}$$

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

$$\omega = \frac{E'(1+A)^2/A' - E(1+R^2)}{2RE} \ , \tag{315}$$

where A' is the ratio of the emitted particle mass to the incident particle mass (A'=1 for neutron scattering). In Eqs. 314 and 315, R is the effective mass ratio

$$R = \sqrt{\frac{A(A+1-A')}{A'}} \sqrt{1 - \frac{(A+1)(-Q)}{AE}} \ , \tag{316}$$

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 Eq. 313 over secondary energy gives

$$\mathcal{F}_{\ell g'}(E) = \int_{\omega_1}^{\omega_2} f(E, \omega) \, P_\ell(\mu[\omega]) \, d\omega \ , \tag{317}$$

where ω_1 and ω_2 are evaluated using Eq. 315 for E' equal to the upper and lower bounds of g', respectively. The scattering probability is given by

$$f(E, \omega) = \sum_{\ell=0}^{L} f_\ell(E) \, P_\ell(\omega) \ , \tag{318}$$

where the Legendre coefficients are either retrieved directly from the ENDF File 4 or computed from File 4 tabulated angular distributions (see subroutines `getfle` and `getco`).

The integration in Eq. 317 is performed (see subroutine `getdis`) simultaneously for all Legendre components using Gaussian quadrature[57]. The quadrature order is selected based on the estimated polynomial order of the integrand. A reasonable estimate is given by

$$\mathrm{ND} + \mathrm{NL} + \log(300/A) \ , \tag{319}$$

where ND is the number of Legendre components desired for the feed function, and NL is the number of components required to represent $f(E, \omega)$. The log term approximates the number of additional components required to represent the center-of-mass to lab transformation.

The two-body feed function for higher Legendre orders is a strongly oscillatory function in some energy ranges. An example is shown in Fig. 29. Furthermore, the integral of the oscillatory part is often small with respect to the magnitude of the function. Such functions are very difficult to integrate with adaptive techniques, which converge to some fraction of the integral of the absolute value. This is the reason that MINX[21] gave poor answers for small Legendre components of the scattering matrix. 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. The scheme used is also well adapted to performing many integrals in parallel. In GROUPR, all Legendre components and all final groups are accumulated simultaneously (see `panel`).

The boundaries of the various regions of the feed function are called "critical points." Between critical points, the feed function is a smooth analytic function of approximately known polynomial order. It is only necessary to add these critical points to the incident energy grid of the feed function (the `enext` variable) and to tell `panel` what quadrature order (`nq`) to use. The critical points are determined in `getff` by solving Eq. 315 for the values of E for which $\omega = +1$ and $\omega = -1$ when E' is equal to the various group boundaries. This can be done by writing

$$\frac{E_g}{E} = \frac{1}{(1+A)^2}\left[R^2 \pm 2R + 1\right] \ , \tag{320}$$

substituting for E using Eq. 316, and then solving for R. The result is

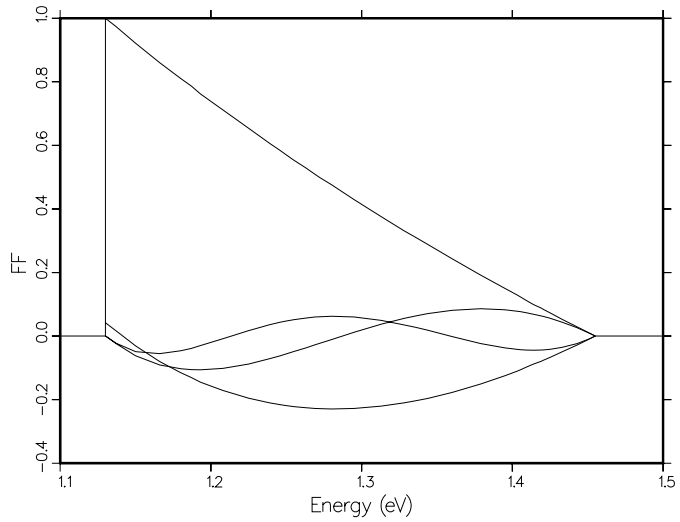

Figure 29: A typical feed function for two-body scattering showing the oscillations that must be treated correctly by the integration over incident energy.

$$E_{crit} = \frac{\dfrac{(A+1)(-Q)}{A}}{1 - \dfrac{A'F^2}{A(A+1-A')}} \ , \tag{321}$$

where

$$F = \frac{1 \pm \sqrt{D}}{1 + \dfrac{E_F}{(-Q)}} \ , \tag{322}$$

$$D = \left[\frac{A(A+1-A')}{A'}\left(1 + \frac{E_F}{(-Q)}\right) - 1\right]\frac{E_F}{(-Q)} \ , \tag{323}$$

and

$$E_F = \frac{1+A}{A+1-A'}E_g \ . \tag{324}$$

File 4 can also contain angular distributions for charged-particle emission through discrete levels (for ENDF/B-VI and later see MT=600 – 648 for protons, MT=650 – 698 for deuterons, and so on; the elastic case, MT=2, is discussed in the next section). Moreover, File 6 can contain angular distributions for discrete two-body scattering (see Law 3). It can also declare that a particular particle is the recoil particle from a two-body reaction (Law 4), in which case the appropriate angular distribution is obtained from the corresponding Law

3 subsection by complementing the angle. The representation of the angular distribution for these laws is almost identical to that in File 4, and the calculation is done in `getdis` using much of the File 4 coding. The effects on the kinematics of the difference in the mass of the emitted and incident particles are handled by the variable A' in the above equations.

8.13 Charged-Particle Elastic Scattering

Coulomb scattering only occurs in the elastic channel, and this calculation also uses the `getdis` subroutine. The problem with Coulomb scattering is that the basic Rutherford formula becomes singular at small angles. In practice, this singularity is removed by the screening effects of the atomic electrons. The forward transport of charged particles in this screening regime is usually handled by continuous-slowing-down theory by using a "stopping power". The new ENDF-6 format allows for three different ways to handle the large-angle scattering regime. First and most general is the nuclear amplitude expansion:

$$\begin{aligned} \sigma &= |\text{nucl} + \text{coul}|^2 \\ &= \sigma_{\text{nucl}} + \sigma_{\text{coul}} + \text{interference} \end{aligned} \tag{325}$$

The Coulomb term is given by the Rutherford formula, a Legendre expansion is defined for the nuclear term, and a complex Legendre expansion is defined for the interference term. This representation cannot be generated directly from experimental data; an R-matrix or phase-shift analysis is necessary.

A method very closely related to experiment (σ_{exp}) is the nuclear plus interference (NI) formula:

$$\sigma_{\text{NI}}(\mu, E) = \sigma_{\text{exp}}(\mu, E) - \sigma_{\text{coul}}(\mu, E), \tag{326}$$

where the function is only defined for angles with cosines $\mu<\mu_{\text{max}}$. The minimum angle is usually taken to be somewhere around 20 degrees (GROUPR uses μ_{max}=0.96). This function is still ill-behaved near the cutoff, and it must be tabulated. The third option is the residual cross section expansion:

$$\sigma_R(\mu, E) = (1 - \mu)\big[\sigma_{\text{exp}}(\mu, E) - \sigma_{\text{coul}}(\mu, E)\big]. \tag{327}$$

The $(1 - \mu)$ term removes the pole at the origin. The residual is uncertain, but it is usually small enough that the entire curve can be fitted with Legendre poly-

nomials without worrying about what happens at small angles. In practice, both the nuclear amplitude expansion and the nuclear plus interference representation are converted to the residual cross section representation in subroutine `conver`. As a result, `getdis` only has to cope with the one representation.

8.14 Continuum Scattering and Fission

In ENDF evaluations, scattering from many closely-spaced levels or multibody scattering such as (n,2n), (n,n′α) or fission is often represented using a separable function of scattering cosine and secondary neutron energy

$$f(E\to E',\mu) = F(E,\mu)\, g(E\to E') \ , \tag{328}$$

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 mostly identified by MT numbers of 6 – 49 and 91. Recently, previously unused MT numbers, 152 through 200, were assigned to additional continuum reactions that are beginning to appear in specialized evaluated files that extend beyond 20 MeV (*e.g.*, the International Reactor Dosimetry and Fusion File). Secondary-energy distributions, whether found in MF=5 or MF=6 are represented by "Laws" as follows:

Law	Description
1	Arbitrary tabulated function
5	General evaporation spectrum
	(Used for delayed neutrons only.)
7	Simple Maxwellian fission spectrum
9	Evaporation spectrum
11	Energy-dependent Watt fission spectrum
12	Energy-dependent spectrum of Madland and Nix

The feed functions for continuum scattering are simply

$$\mathcal{F}_{\ell g'}(E') = \int_{-1}^{+1} P_\ell(\mu)\, f(E,\mu)\, d\mu \int_{g'} g(E,E')\, dE' \ . \tag{329}$$

The first integral is returned by `getfle` ["fle" for $f_\ell(E)$] as described above, and

the second integral is returned by `getsed` ("sed" for secondary-energy distribution).

For Law 1, g is given as a tabulated function of E' for various values of E. When $E_1 \le E < E_2$, the term $\int_{g'} g(E,E')\,dE'$ is obtained by interpolating between precomputed values of $\int_{g'} g(E_1,E')\,dE'$ and $\int_{g'} g(E_2,E')\,dE'$ in subroutine `getsed`. Except in the case of fission, any apparent upscatter produced by the "stairstep" treatment near $E{=}E'$ is added to the in-group scattering term ($g'{=}g$).

For Law 5, $g(x)$ is given versus $x = E'/\theta(E)$ and $\theta(E)$ is given vs. E in File 5. This secondary neutron distribution leads to the following group integral:

$$\int_{g'} g(E,E')\,dE' = \theta(E) \int_{E_1/\theta}^{E_2/\theta} g(x)\,dx \ , \tag{330}$$

with E_1 and E_2 being the lower and upper boundary energies for group g'.

For Law 7, the secondary-energy distribution is given by

$$g(E,E') = \frac{\sqrt{E'}}{I} \exp\!\left(-\frac{E'}{\theta(E)}\right) , \tag{331}$$

where the effective temperature $\theta(E)$ is tabulated in File 5 and the normalization factor is given by

$$I = \theta^{3/2}\Big[\sqrt{\frac{\pi}{4}}\,\mathrm{erf}(x) - xe^{-x}\Big] \ , \tag{332}$$

where

$$x = \frac{E-U}{\theta} \ . \tag{333}$$

Here U is a constant used to define the upper limit of secondary neutron energy $\theta \le E' \le E{-}U$. The desired group integral is given by

$$\begin{aligned} \int_{g'} g(E,E')\,dE' &= \int_{E_1}^{E_2} g(E,E')\,dE' \\ &= \frac{X_1 - X_2 - Y_1 + Y_2}{\sqrt{\pi/4} - Y - X} \ , \end{aligned} \tag{334}$$

where

$$X = \sqrt{x}\, e^{-x} \ , \tag{335}$$

$$Y = \sqrt{\frac{\pi}{4}}\, \mathrm{rerfc}(\sqrt{x}) \ , \tag{336}$$

and where X_1, Y_1, X_2, and Y_2 refer to X and Y evaluated at E_1/θ and E_2/θ. The integral of Eq. 334 is computed in `anased` ("`ana`" for analytic). The function "rerfc" is the reduced complementary error function[57].

For Law 9, the secondary-energy distribution is given by

$$g(E, E') = \frac{E'}{I} \exp\left(-\frac{E'}{\theta(E)}\right) \ , \tag{337}$$

where

$$I = \theta^2 \left[1 - e^{-x}(1 + x)\right] \ . \tag{338}$$

Here x has the same meaning as above – see Eq. 333. The group integral is

$$\int_{g'} g(E, E')\, dE' = \frac{(1 + x_1)e^{-x_1} - (1 + x_2)e^{-x_2}}{1 - (1 + x)e^{-x}} \ , \tag{339}$$

where x_1 and x_2 refer to E_1/θ and E_2/θ, respectively. This result is also computed in `anased`.

For Law 11, the secondary-energy distribution is given by

$$g(E, E') = \frac{e^{-E'/a}}{I} \sinh\left(\sqrt{bE'}\right) \ , \tag{340}$$

where

$$I = \frac{1}{2}\sqrt{\frac{\pi a^3 b}{4}}\, e^{x_0} \left[\mathrm{erf}(\sqrt{x} - \sqrt{x_0}) + \mathrm{erf}(\sqrt{x} + \sqrt{x_0})\right] - a e^{-x \sinh(abx)} \ . \tag{341}$$

In this case, $a(E)$ and $b(E)$ are given in File 5,

$$x = \frac{E - U}{a} \ , \tag{342}$$

and

$$x_0 = \frac{ab}{4} \ . \tag{343}$$

The group integrals are given by

$$\int_{g'} g(E,E') = \frac{H(x_1,x_2,x_0)}{H(0,x,x_0)} \ , \tag{344}$$

where

$$\begin{aligned} H(x_1,x_2,x_0) &= H_2(\sqrt{x_1}-\sqrt{x_0},\sqrt{x_2}-\sqrt{x_0}) \\ &+ \sqrt{x_0}H_1(\sqrt{x_1}-\sqrt{x_0},\sqrt{x_2}-\sqrt{x_0}) \\ &- H_2(\sqrt{x_1}+\sqrt{x_0},\sqrt{x_2}+\sqrt{x_0}) \\ &+ \sqrt{x_0}H_1(\sqrt{x_1}+\sqrt{x_0},\sqrt{x_2}+\sqrt{x_0}) \ , \end{aligned} \tag{345}$$

and where

$$H_n(a,b) = \frac{1}{\pi}\int_a^b z^n e^{-z^2}\,dz \ . \tag{346}$$

The methods for computing H_n are described in BROADR. When $\sqrt{x_2} < .01$, a short-cut calculation can be used for the numerator of Eq. 344

$$\frac{4\sqrt{x_0}e^{-x_0}}{3\sqrt{\pi/4}}\left[x_2^{3/2}-x_1^{3/2}\right] \ . \tag{347}$$

For Law 12, the Madland-Nix option, the secondary-energy distribution is given by

$$g(E,E') = \frac{1}{2}\left[G(E',E_{fl})+G(E',E_{fh})\right] \ , \tag{348}$$

where

$$\begin{aligned} &G(E',E_f) \\ &\quad = \frac{1}{3\sqrt{E_f T_m}}\left[u_2^{3/2}E_1(u_2)-u_1^{3/2}E_1(u_1)+\gamma(3/2,u_2)-\gamma(3/2,u_1)\right] \ , \end{aligned} \tag{349}$$

$$u_1 = (\sqrt{E'} - \sqrt{E_f})^2/T_m \ , \text{ and} \tag{350}$$

$$u_2 = (\sqrt{E'} + \sqrt{E_f})^2/T_m \ , \tag{351}$$

and where E_{fl}, E_{fh}, and $T_m(E)$ are given in File 5. The special functions used are the first-order exponential integral, $E_1(x)$, and the incomplete gamma function, $\gamma(n,x)$. The group integrals of this function are very complex[58]. Let

$$\begin{aligned}
\alpha &= \sqrt{T_m} \ , && (352)\\
\beta &= \sqrt{E_f} \ , && (353)\\
A &= \frac{(\sqrt{E_1}+\beta)^2}{\alpha^2} \ , && (354)\\
B &= \frac{(\sqrt{E_2}+\beta)^2}{\alpha^2} \ , && (355)\\
A' &= \frac{(\sqrt{E_1}-\beta)^2}{\alpha^2} \ , \text{ and} && (356)\\
B' &= \frac{(\sqrt{E_2}-\beta)^2}{\alpha^2} \ . && (357)
\end{aligned}$$

Then the integral over the range (E_1, E_2) of G is given by one of the following three expression, depending on the region of integration in which E_1 and E_2 lie.
Region I ($E_1 \geq E_f, E_2 > E_f$)

$$\begin{aligned}
&3\sqrt{E_f T_m}\int_{E_1}^{E_2} G(E', E_f)\, dE' \\
&\quad = \big[\,(\tfrac{2}{5}\alpha^2 B^{5/2} - \tfrac{1}{2}\alpha\beta B^2)E_1(B) - (\tfrac{2}{5}\alpha^2 A^{5/2} - \tfrac{1}{2}\alpha\beta A^2)E_1(A)\,\big] \\
&\quad - \big[\,(\tfrac{2}{5}\alpha^2 B'^{5/2} + \tfrac{1}{2}\alpha\beta B'^2)E_1(B') - (\tfrac{2}{5}\alpha^2 A'^{5/2} + \tfrac{1}{2}\alpha\beta A'^2)E_1(A')\,\big] \\
&\quad + \big[\,(\alpha^2 B - 2\alpha\beta B^{1/2})\,\gamma(3/2, B) - (\alpha^2 A - 2\alpha\beta A^{1/2})\,\gamma(3/2, A)\,\big] \\
&\quad - \big[\,(\alpha^2 B' + 2\alpha\beta B'^{1/2}\,\gamma(3/2, B') - (\alpha^2 A' + 2\alpha\beta A'^{1/2})\,\gamma(3/2, A')\,\big] \\
&\quad - \frac{3}{5}\alpha^2\big[\,\gamma(5/2, B) - \gamma(5/2, A) - \gamma(5/2, B') + \gamma(5/2, A')\,\big] \\
&\quad - \frac{3}{2}\alpha\beta\big[\,\mathrm{e}^{-B}(1+B) - \mathrm{e}^{-A}(1+A) + \mathrm{e}^{-B'}(1+B') - \mathrm{e}^{-A'}(1+A')\,\big] \ .
\end{aligned} \tag{358}$$

Region II ($E_1 < E_f, E_2 \le E_f$)

$$
\begin{aligned}
&3\sqrt{(E_f T_m)} \int_{E_1}^{E_2} G(E', E_f)\, dE' \\
&\quad = \big[(\frac{2}{5}\alpha^2 B^{5/2} - \frac{1}{2}\alpha\beta B^2) E_1(B) - (\frac{2}{5}\alpha^2 A^{5/2} - \frac{1}{2}\alpha\beta A^2) E_1(A) \big] \\
&\quad - \big[(\frac{2}{5}\alpha^2 B'^2) E_1(B') - (\frac{2}{5}\alpha^2 A'^{5/2} - \frac{1}{2}\alpha\beta A'^2) E_1(A') \big] \\
&\quad + \big[(\alpha^2 B - 2\alpha\beta B^{1/2})\, \gamma(3/2, B) - (\alpha^2 A - 2\alpha\beta A^{1/2})\, \gamma(3/2, A) \big] \\
&\quad - \big[(\alpha^2 B' - 2\alpha\beta B'^{1/2}\, \gamma(3, 2, B') - (\alpha^2 A' - 2\alpha\beta A'^{1/2})\, \gamma(3/2, A') \big] \\
&\quad - \frac{3}{5}\alpha\beta \big[\gamma(5/2, B) - \gamma(5/2, A) - \gamma(5/2, B') + \gamma(5/2, A') \big] \\
&\quad - \frac{3}{2}\alpha\beta \left[\mathrm{e}^{-B}(1+B) - \mathrm{e}^{-A}(1+A) - \mathrm{e}^{-B'}(1+B') + \mathrm{e}^{-A'}(1+A') \right] .
\end{aligned}
\tag{359}
$$

Region III ($E_1 < E_f, E_2 > E_f$)

$$
\begin{aligned}
&3\sqrt{E_f T_m} \int_{E_1}^{E_2} G(E', E_f)\, dE' \\
&\quad = \big[(\frac{2}{5}\alpha^2 B^{5/2} - \frac{1}{2}\alpha\beta B^2) E_1(B) - (\frac{2}{5}\alpha^2 A^{5/2} - \frac{1}{2}\alpha\beta A^2) E_1(A) \big] \\
&\quad - \big[(\frac{2}{5}\alpha^2 B'^{5/2} + \frac{1}{2}\alpha\beta B'^2) E_1(B') - (\frac{2}{5}\alpha^2 A'^{5/2} - \frac{1}{2}\alpha\beta A'^2) E_1(A') \big] \\
&\quad + \big[(\alpha^2 B - 2\alpha\beta B^{1/2})\, \gamma(3/2, B) - (\alpha^2 A - 2\alpha\beta A^{1/2})\, \gamma(3, 2, A) \big] \\
&\quad - \big[(\alpha^2 B' + 2\alpha\beta B'^{1/2})\, \gamma(3/2, B') - (\alpha^2 A' - 2\alpha\beta A'^{1/2})\, \gamma(3/2, A_)' \big] \\
&\quad - \frac{3}{5}\alpha^2 \big[\gamma(5/2, B) - \gamma(5/2, A) - \gamma(5/2, B') + \gamma(5/2, A') \big] \\
&\quad - \frac{3}{2}\alpha\beta \left[\mathrm{e}^{-B}(1+B) - \mathrm{e}^{-A}(1+A) + \mathrm{e}^{-B'}(1+B') + \mathrm{e}^{-A'}(1+A') - 2 \right] .
\end{aligned}
\tag{360}
$$

8.15 File 6 Energy-Angle Distributions

If the File 6 data are expressed as a continuous energy-angle distribution (Law 1) in the laboratory system, it is fairly easy to generate the multigroup transfer matrix. As usual for GROUPR, the task is to calculate the "feed function" (the Legendre moments for transferring to all possible secondary-energy groups starting with incident energy E). The E' integration is controlled by the `getmf6` subroutine, which calls `f6lab` to generate the integrands. This routine expects data in Law-1 format, where the looping order is E, E', μ. The only problems here are handling the new ENDF-6 interpolation laws for two-dimensional tabulations (for example, unit base) and converting tabulated angle functions for $E \to E'$ into Legendre coefficients (which is done with a Gauss-Legendre quadrature of order ~ 8).

If the File 6 data are given in the energy-angle form of Law 7 (where the

looping order is E, μ, E'), GROUPR converts it to the Law 1 form using subroutine `ll2lab`. It does this by stepping through an E' grid that is the union of the E' grids for all the different angles given. At each of these union E' values, it calculates the Legendre coefficients using a Gauss-Legendre quadrature of order 8, and it checks back to see if the preceding point is still needed to represent the distribution to the desired degree of accuracy. Now `getmf6` and `f6lab` can be used to complete the calculation as above.

If the File 6 data are expressed in the CM system, or if the phase-space option is used, more processing is necessary to convert to the LAB system. This conversion is done for each incident energy E given in the file. The grid for laboratory secondary energy E'_L is obtained by doing an adaptive reconstruction of the emission probability $p_{L\ell}(E, E'_L)$ such that each Legendre order can be expressed as a linear-linear function of E'_L. This part is done in subroutine `cm2lab`. The values for $p_{L\ell}(E, E'_L)$ are obtained in subroutine `f6cm` by doing an adaptive integration along the contour E'_L=const in the E'_C, μ_C plane using μ_L as the variable of integration:

$$p_{L\ell}(E, E'_L) = \int_{\mu_{\min}}^{+1} p_C(E, E'_C, \mu_C)\, P_\ell(\mu_L)\, J\, d\mu_L \ , \tag{361}$$

where μ is a scattering cosine and L and C denote the laboratory and center-of-mass (CM) systems, respectively. The Jacobian is given by

$$J = \sqrt{\frac{E'_L}{E'_C}} = \frac{1}{\sqrt{1 + c^2 - 2c\mu_L}} \ , \tag{362}$$

and the cosine transformation is given by

$$\mu_C = J(\mu_L - c) \ . \tag{363}$$

The constant c is given by

$$c = \sqrt{\frac{A'}{(A+1)^2}} \sqrt{\frac{E}{E'_L}} \ , \tag{364}$$

where A is the ratio of the atomic weight of the target to the atomic weight of the projectile, and A' is the ratio of the atomic weight of the emitted particle to the atomic weight of the projectile. The lower limit of the integral depends on the maximum possible value for the center-of-mass (CM) secondary energy as follows:

$$\mu_{\min} = \frac{1}{2c}\Big(1 + c^2 - \frac{E'_{C\max}}{E'_L}\Big), \tag{365}$$

where

$$E'_{L\max} = E\Big(\sqrt{\frac{E'_{C\max}}{E}} + \sqrt{\frac{A'}{(A+1)^2}}\Big)^2. \tag{366}$$

An example of the integration contours for this coordinate transformation is given in Fig. 30.

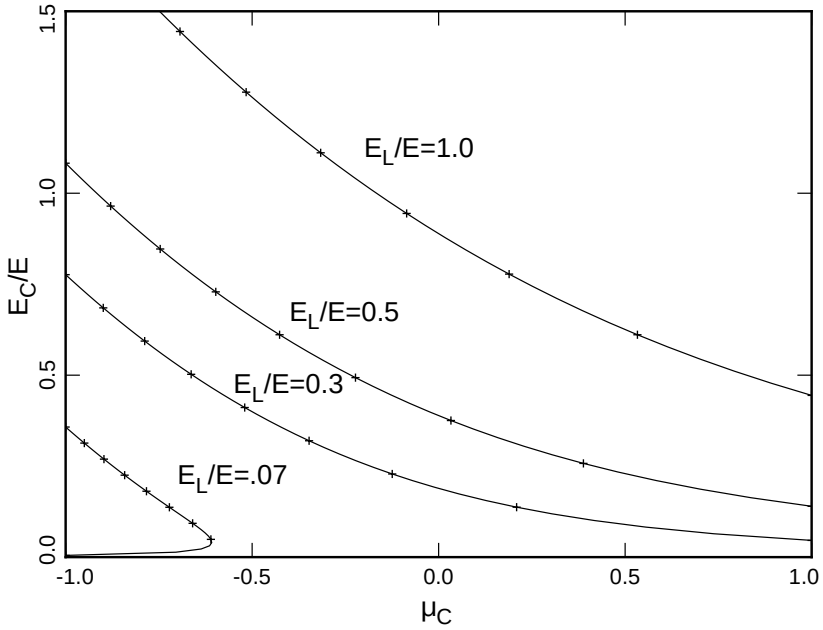

Figure 30: Coordinate mapping between CM and laboratory reference frames for A=2. The parameters E_L and E_C are the secondary energies in the Lab and CM frames. The crosses on the curves are at μ_L values of -1.0, -.75, -.50, ..., .75, and 1.0. This figure also illustrates the advantage of integrating over μ_L for the contour in the lower-left corner. The values of E_C and μ_C are single-values functions of μ_L.

The CM energy-angle distribution can be given as a set of Legendre coefficients or a tabulated angular distribution for each possible energy transfer $E \to E'$, as a "precompound fraction" $r(E, E')$ for use with the Kalbach-Mann[59] or Kalbach[60] angular distributions, or as parameters for a phase-space distribution. The first three options are processed using `f6ddx`, and the last using `f6psp`. The Kalbach option leads to a very compact representation. Kalbach and Mann

examined large number of experimental angular distributions for neutrons and charged particles. They noticed that each distribution could be divided into two parts: an equilibrium part symmetric in μ, and a forward-peaked preequilibrium part. The relative amount of the two parts depended on a parameter r, the preequilibrium fraction, that varies from zero for low E' to 1.0 for large E'. The shapes of the two parts of the distributions depended most directly on E'. This representation is very useful for preequilibrium statistical-model codes like GNASH[61], because they can compute the parameter r, and all the rest of the angular information comes from simple universal functions. More specifically, Kalbach's latest work says that

$$f(\mu) = \frac{a}{2\sinh(a)}\Big[\cosh(a\mu) + r\sinh(a\mu)\Big], \tag{367}$$

where a is a simple function of E, E', and B_b, the separation energy of the emitted particle from the liquid-drop model without pairing and shell terms. The separation energies are computed by formulas in `bach`. There is a problem for elemental evaluations, because the calculations needs an A value for the element, and it is difficult to guess which A value is most characteristic of the element. A short table is included in the routine, and an "`error in bach`" will result if the function is called for an element that doesn't appear in the table. Similar routines appear in HEATR and ACER. A better long-range solution would be desirable. Fortunately, elemental evaluations are rare in modern evaluated libraries.

The File 6 processing methods in GROUPR apply equally well to neutrons, photonuclear photons, and charged particles. The effects on the kinematics due to the difference in mass between the incident particle and the emitted particle are handled by the variable A' in the above equations.

8.16 Smoothing

For continuum CM distributions in File 6, the low-energy shape should go like $\sqrt{E'_C}$. From Eq. 362, we see that the low-energy shape for the Lab distribution would then go like $\sqrt{E'_L}$. However, ENDF/B-VII evaluations for File 6 were normally prepared using advanced nuclear model codes, such as GNASH from LANL. These codes naturally produce spectra represented with histogram bins, and these bins normally give a constant probability from zero energy to the next bin boundary. This is certainly not a $\sqrt{E}$ shape! The histogram shape can cause trouble with the CM-Lab transformation if some care is not taken. Clearly, the histogram grossly overestimates the probability of scattering to very

low energies. This apparent problem is somewhat alleviated by the fact that the total probability of that lowest histogram bin is usually quite small. However, in order to get better looking shapes, GROUPR has coding that replaces the coarse histogram with a finer histogram chosen to represent a $\sqrt{E}$ shape more closely. This smoothing coding scans up through the histogram distribution from the evaluation to find the region that behaves like $\sqrt{E}$. It then uses a recursive procedure to subdivide the region using a given factor until it reaches a fairly low energy (currently 40 eV).

A similar procedure is used to provide a $\sqrt{E}$ shape at low energies for delayed neutron spectra.

For a few of the ENDF/B-VII actinides, the energy grid used to represent the fission spectrum becomes too coarse above 10 MeV. If that situtation is found, GROUPR changes the interpolation law from the given lin-lin option to lin-log — that is, an exponential tail is assumed for the high energies. This assumption is consistent with the orginal evaluations. This modification can be important for reaction rates of high-threshold reactions.

These smoothing options are controlled by the global parameter `ismooth`, which is turned on by default (in constrast to NJOY99 where it is turned off by default).

8.17 GENDF Output

The group constants produced by GROUPR are normally written to an output file in GENDF (groupwise ENDF) format for use by other modules of NJOY. For example, DTFR can be used to convert a GENDF material to DTF (or "transport") format; CCCCR produces the standard interface files[11] ISOTXS, BRKOXS, and DLAYXS; MATXSR produces a file in MATXS format[12]; and POWR produces libraries for the Electric Power Research Institute (EPRI) codes EPRI-CELL and EPRI-CPM. Other formats can easily be produced by new modules, and some functions such as group collapse, are conveniently performed directly in GENDF format. The GENDF format is also used for GAMINR output, and both ACER and ERRORR use GENDF input for some purposes.

Depending on the sign of `ngout2` (see below), the GENDF file will be written in either coded mode (*e.g.*, ASCII) or in the special NJOY blocked-binary mode. Conversion between these modes can be performed subsequently by the MODER module.

The GENDF material begins with a header record (MF=1, MT=451), but the format of this first section is different from MT=451 on an ENDF or PENDF

tape. The section consists of a "CONT" record, containing

ZA	Standard 1000*Z+A value
AWR	Atomic weight ratio to neutron
0	Zero
NZ	Number of σ_0 values
-1	Identifies a GENDF-type data file
NTW	Number of words in title

and a single "LIST" record, containing

TEMPIN	Material temperature (Kelvin)
0.	Zero
NGN	Number of neutron groups
NGG	Number of photon groups
NW	Number of words in LIST
0	Zero
TITLE	Title from GROUPR input (NTW words)
SIGZ	Sigma-zero values (NZ words)
EGN	Neutron group boundaries, low to high (NGN+1 words)
EGG	Photon group boundaries, low to high (NGG+1 words).

For photoatomic GENDF files produced by the GAMINR module, the photon group structure is stored in ngn and egn, and the number of photon groups is given as ngg=0. The word count is NW=NTW+NZ+NGN+1+NGG+1. The LIST record is followed by a standard ENDF file-end record (FEND). The normal ENDF section-end (SEND) is omitted.

This header is followed by a series of records for reactions. The ENDF ordering requirements are relaxed, and MF and MT values can occur in any order. Each section starts with a "CONT" record.

ZA	Standard 1000*Z+A value
AWR	Standard atomic weight ratio
NL	Number of Legendre components
NZ	Number of sigma-zero values
LRFLAG	Break-up identifier flag
NGN	Number of groups

It is followed by a series of LIST records, one for every incident-energy group with nonzero result,

TEMP	Material temperature (Kelvin)
0.	Zero
NG2	Number of secondary positions
IG2LO	Index to lowest nonzero group
NW	Number of words in LIST
IG	Group index for this record
A(NW)	Data for LIST (NW words),

where NW=NL*NZ*NG2. The last LIST record in the sequence is the one with IG=NGN. It must be given even if its contents are zero. The last LIST record is followed by a SEND record.

The contents of A(NW) change for various types of data. For simple cross section "vectors" (MF=3), NG2 is 2, and A contains the two Fortran arrays

FLUX(NL,NZ), SIGMA(NL,NZ)

in that order. For ratio quantities like fission $\overline{\nu}$, NG2 is 3, and A contains

FLUX(NL,NZ), RATIO(NL,NZ), SIGMA(NL,NZ).

For transfer matrices (MF=6, 16, 21, *etc.*), A contains

FLUX(NL,NZ), MATRIX(NL,NZ,NG2-1).

The actual secondary group indices for the last index of MATRIX are usually IG2LO, IG2LO+1, *etc.*, using the GROUPR convention of labeling groups in order of increasing energy. If the low-energy part of the fission matrix (or the fission or capture photon production matrices) uses the special format described in Section 8.6, the spectrum will be found in a LIST record with IG=0 and the production cross section will be found in a series of records with IG2LO=0. The group range for the spectrum ranges from IG2LO to IG2LO+NG2-1. For IG2LO=0, NG2 will be 2 as for a normal cross section, and the two values will be the flux for group IG and the corresponding production cross section.

Finally, for delayed neutron spectra (MF=5), NL is used to index the time groups, NZ is 1, and there is only one incident energy record (IG=IGN). The array A contains

LAMBDA(NL), CHID(NL,NG2-1),

where LAMBDA contains the delayed-neutron time constants and CHID contains the spectra.

The GENDF material ends with a material-end (MEND) record, and the GENDF tape ends with a tape-end (TEND) record.

8.18 Running GROUPR

GROUPR's input instructions follow. They are reproduced from the comment cards at the beginning of the 2016.0 version of the GROUPR module. Because the code changes from time to time, it is a good idea to check these comment cards in the current version to obtain up-to-date input instructions.

```
!---input specifications (free format)--------------------------
!
! card1
!    nendf   unit for endf tape
!    npend   unit for pendf tape
!    ngout1  unit for input gout tape (default=0)
!    ngout2  unit for output gout tape (default=0)
! card2
!    matb    material to be processed
!             if ngout=0, matb<0 is an option to automatically
!              process all the mats on the endf input tape.
!             otherwise, matb<0 is a flag to add mts to and/or
!              replace individual mts on gout1.
!    ign     neutron group structure option
!    igg     gamma group structure option
!    iwt     weight function option
!    lord    legendre order
!    ntemp   number of temperatures (default=1)
!    nsigz   number of sigma zeroes (default=1)
!    iprint  long print option (0/1=minimum/maximum)
!            (default=1)
!    ismooth switch on/off smoothing operation (1/0, default=1=on)
!            set ismooth to 1 to enable sqrt(e) smoothing for
!            mf6 cm emission spectra at low energies and for
!            histogram delayed neutron spectra at low energies.
! card3
!    title   run label (up to 80 characters delimited by quotes,
!            ended with /)  (default=blank)
! card4
!    temp    temperatures in kelvin
! card5
!    sigz    sigma zero values (including infinity)
!
!          if ign=1, read neutron group structure (6a and 6b)
```

```
! card6a
!    ngn     number of groups
! card6b
!    egn     ngn+1 group breaks (ev)
!
!          if igg=1, read gamma group structure (7a and 7b)
! card7a
!    ngg     number of groups
! card7b
!    egg     ngg+1 group breaks (ev)
!
!          weight function options (8a,8b,8c,8d)
! card8a     flux calculator parameters (iwt.lt.0 only)
!    fehi    break between computed flux and bondarenko flux
!            (must be in the resolved resonance range)
!    sigpot  estimate of potential scattering cross section
!    nflmax  maximum number of computed flux points
!    ninwt   tape unit for new flux parameters (default=0)
!            note: weighting flux file is always written binary
!    jsigz   index of reference sigma zero in sigz array
!            (default=0)
!    alpha2   alpha for admixed moderator (def=o=none)
!    sam      admixed moderator xsec in barns per absorber
!             atom (def=0=none)
!    beta     heterogeneity parameter (def=0=none)
!    alpha3   alpha for external moderator (def=0=none)
!    gamma    fraction of admixed moderator cross section in
!              external moderator cross section (def=0)
! card8b     tabulated (iwt=1 or -1 only)
!    wght    read weight function as tab1 record,
!            this may span multiple lines and ends with a /.
! card8c     analytic flux parameters (iwt=4 or -4 only)
!    eb      thermal break (ev)
!    tb      thermal temperature (ev)
!    ec      fission break (ev)
!    tc      fission temperature (ev)
! card8d     input resonance flux (iwt=0 only)
!    ninwt   tape unit for flux parameters (binary)
!
! card9
!    mfd     file to be processed
!    mtd     section to be processed
```

```
!    mtname  description of section to be processed
!          repeat for all reactions desired
!          mfd=0/ terminates this temperature/material.
! card10
!    matd    next mat number to be processed
!            matd=0/ terminates groupr run.
!
!---options for input variables---------------------------------
!
!     ign          meaning
!     ---          -------
!      1           arbitrary structure (read in)
!      2           csewg 239-group structure
!      3           lanl 30-group structure
!      4           anl 27-group structure
!      5           rrd 50-group structure
!      6           gam-i 68-group structure
!      7           gam-ii 100-group structure
!      8           laser-thermos 35-group structure
!      9           epri-cpm 69-group structure
!     10           lanl 187-group structure
!     11           lanl 70-group structure
!     12           sand-ii 620-group structure
!     13           lanl 80-group structure
!     14           eurlib 100-group structure
!     15           sand-iia 640-group structure
!     16           vitamin-e 174-group structure
!     17           vitamin-j 175-group structure
!     18           xmas nea-lanl
!     all new additional group structure with 7 significant
!     decimal digits compatible with calendf
!     19           ecco  33-group structure
!     20           ecco 1968-group structure
!     21           tripoli 315-group structure
!     22           xmas lwpc 172-group structure
!     23           vit-j lwpc 175-group structure
!     24           shem cea 281-group structure
!     25           shem epm 295-group structure
!     26           shem cea/epm 361-group structure
!     27           shem epm 315-group structure
!     28           rahab aecl 89-group structure
!     29           ccfe   660-group structure  (30 MeV)
```

```
!     30          ukaea 1025-group structure  (30 MeV)
!     31          ukaea 1067-group structure (200 MeV)
!     32          ukaea 1102-group structure   (1 GeV)
!     33          ukaea  142-group structure (200 MeV)
!     34          lanl 618-group structure
!
!     igg         meaning
!     ---         -------
!      0          none
!      1          arbitrary structure (read in)
!      2          csewg 94-group structure
!      3          lanl 12-group structure
!      4          steiner 21-group gamma-ray structure
!      5          straker 22-group structure
!      6          lanl 48-group structure
!      7          lanl 24-group structure
!      8          vitamin-c 36-group structure
!      9          vitamin-e 38-group structure
!     10          vitamin-j 42-group structure
!
!     iwt         meaning
!     ---         -------
!      1          read in smooth weight function
!      2          constant
!      3          1/e
!      4          1/e + fission spectrum + thermal maxwellian
!      5          epri-cell lwr
!      6          (thermal) -- (1/e) -- (fission + fusion)
!      7          same with t-dep thermal part
!      8          thermal--1/e--fast reactor--fission + fusion
!      9          claw weight function
!     10          claw with t-dependent thermal part
!     11          vitamin-e weight function (ornl-5505)
!     12          vit-e with t-dep thermal part
!     -n          compute flux with weight n
!      0          read in resonance flux from ninwt
!
!     mfd         meaning
!     ---         -------
!      3          cross section or yield vector
!      5          fission chi by short-cut method
!      6          neutron-neutron matrix (mf4/5)
```

```
!      8           neutron-neutron matrix (mf6)
!     12           photon prod. xsec (photon yields given, mf12)
!     13           photon prod. xsec (photon xsecs given, mf13)
!     16           neutron-gamma matrix (photon yields given)
!     17           neutron-gamma matrix (photon xsecs given)
!     18           neutron-gamma matrix (mf6)
!         note: if necessary, mfd=13 will automatically change
!         to 12 and mfd=16 will automatically change to 17 or 18.
!     21           proton production matrix (mf6)
!     22           deuteron production (mf6)
!     23           triton production (mf6)
!     24           he-3 production (mf6)
!     25           alpha production (mf6)
!     26           residual nucleus (a>4) production (mf6)
!     31           proton production matrix (mf4)
!     32           deuteron production (mf4)
!     33           triton production (mf4)
!     34           he-3 production (mf4)
!     35           alpha production (mf4)
!     36           residual nucleus (a>4) production (mf4)
!          note: if necessary, mfd=21-26 will
!          automatically change to 31-36.
!    1zzzaaam       nuclide production for zzzaaam
!                     subsection from file 3
!    2zzzaaam       nuclide production for zzzaaam
!                     subsection from file 6
!    3zzzaaam       nuclide production for zzzaaam
!                     subsection from file 9
!    4zzzaaam       nuclide production for zzzaaam
!                     subsection from file 10
!    40000000       fission product production (mtd=18 only)
!                     subsection from file 10
!
!     mtd          meaning
!     ---          -------
!     -n           process all mt numbers from the previous
!                          entry to n inclusive
!     221-250      reserved for thermal scattering
!     257          average energy
!     258          average lethargy
!     259          average inverse velocity (m/sec)
!
```

```
!     automatic reaction processing options
!     -------------------------------------
!        3/        do all reactions in file3 on input pendf
!        6/        do all matrix reactions in endf dictionary
!       10/        do all isotope productions using mf8
!       13/        do all photon production cross sections
!       16/        do all photon production matrices
!       21/        do all proton production matrices
!       22/        do all deuteron production matrices
!       23/        do all triton production matrices
!       24/        do all he-3 production matrices
!       25/        do all alpha production matrices
!       26/        do all a>4 production matrices
!
!--------------------------------------------------------------------
```

In these instructions, `card1` defines the input and output units for GROUPR. The module requires both ENDF and PENDF input tapes, because the PENDF tapes produced by RECONR, BROADR *etc.*, do not contain angle (MF=4), energy (MF=5), or photon (MF=12, 15) distributions. For materials that do not use resonance parameters to represent part of the cross section, it is possible to use a copy of the ENDF tape in place of the PENDF tape. The normal mode for GROUPR is to use `ngout1`=0; however, sometimes it is convenient to add a new material or reaction to an existing GENDF tape. The old GENDF tape is then mounted on unit `ngout1`, and the revised GENDF tape will be written to `ngout2`.

Card 2 selects the first material to be processed (`matb`) and sets up the group structures, weighting option, Legendre order, and self-shielding parameters for all the materials to be processed in this run.

The names of the available group structures are given in the input instructions. Energy bounds or lethargy bounds can be found in the source code. Of course, it is always possible to read in an arbitrary group structure (see `card6a` through `card7b`). The energies must be given in increasing order (note that this is opposite from the usual convention). Here is an example of the input cards for the conventional 4-group structure historically used in some thermal reactor codes:

```
4/ card6a
```

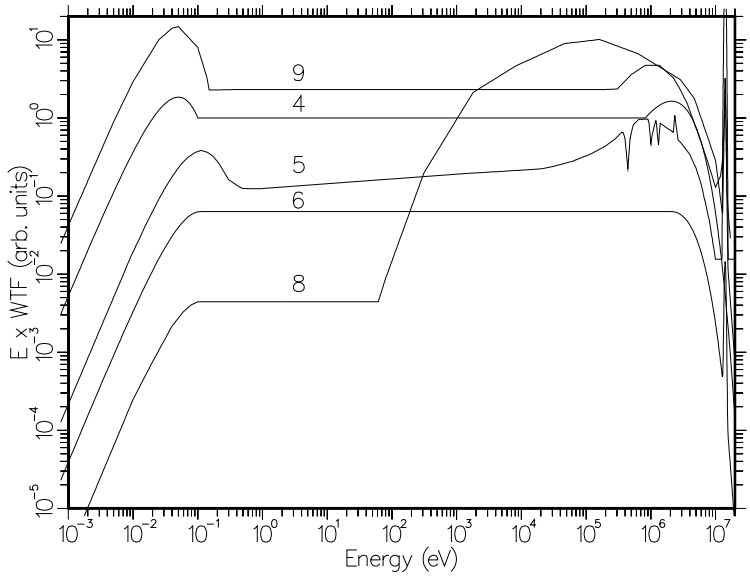

Figure 31: Built-in neutron weight functions of GROUPR on a logarithmic flux-per-unit-lethargy plot that emphasizes the low energy range.

```
1e-5 .625 5530 .821e6 10e6 / card6b
```

These cards are read by the standard Fortran READ* method. Fields are delimited by space, and "/" terminates the processing of input on a card. Anything after the slash is a comment.

The available weight function options are listed in the input instructions under `iwt`. See Fig. 31 and Fig. 32. Here are brief descriptions of the options:

IWT=2 The weight function is constant (not shown in the Figures). This option is usually chosen for very fine group structures such as the 620-group or 640-group dosimetry structures.

IWT=3 The weight function is proportional to $1/E$. The slowing-down of neutrons in water gives a $1/E$ flux from about 1 eV up to 100 keV, or so. This weight function is traditionally used for calculating resonance integrals, but it is not useful at the lower and higher energies needed for a full set of transport constants. Although not shown, the graph of this function is a flat line on a flux-per-unit-lethargy plot, such as the one in Fig. 31.

IWT=4 This weight function combines a thermal Maxwellian at low energies, a $1/E$ function at intermediate energies, and a fission spectrum at high energies to obtain a function appropriate for several different applications. The temperatures of the Maxwellian and fission parts and the energies

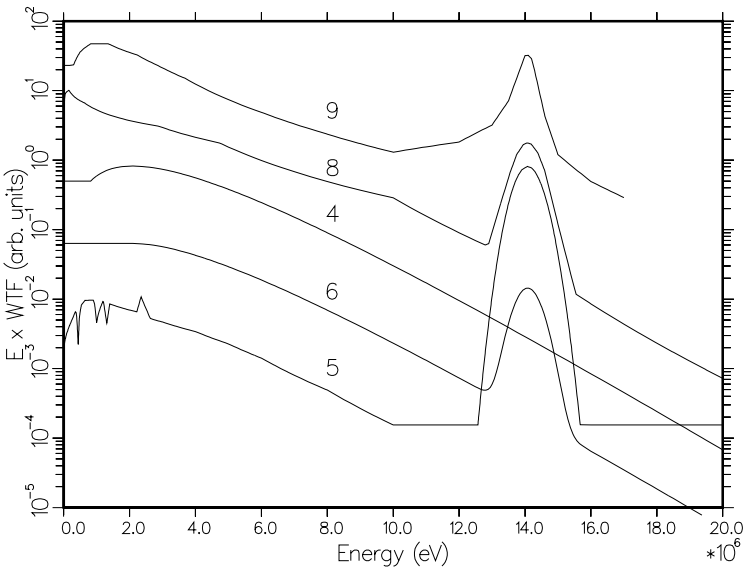

Figure 32: Built-in neutron weight functions on a linear plot that emphasizes the high-energy range.

where the components join can be chosen by the user. Therefore, option 4 can be used to produce typical thermal reactor weight functions like those shown in Figures 31 and 32, a pure fission spectrum for calculating some kinds of dosimetry cross sections, or a pure thermal spectrum for getting effective thermal average cross sections. The function for IWT=4 shown in the figure was produced using a thermal temperature of 0.025 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.

IWT=5 This is a mid-life PWR (pressurized water reactor) flux spectrum with a fusion peak added (see below for a discussion of the fusion peak used). Note the peaks and dips resulting from oxygen resonances and windows at high energies, and the hardening apparent in the epithermal region. The thermal part of the function is also hardened with respect to a simple Maxwellian shape. The dips that should show up in the eV range due to resonances and ^{238}U have been removed to allow the self-shielding method to work without risk of counting the shielding effects twice. This weight function has been used for several libraries[62] prepared for the Electric Power Research Institute (EPRI), and for the MATXS7 library used with the TRANSX code.

IWT=6 This function is similar to option 4, but the breakpoints were chosen to keep the curves continuous. The thermal Maxwellian is calculated for 300K. In this case, the fusion peak (see below) was added to the high-energy tail of the fission spectrum for smoothness.

IWT=7 This option is reserved for future use.

IWT=8 This function is intended for cross sections libraries used for fast reactor analysis (typically, fast breeder designs), but is also useful for fusion-blanket problems. It has a fusion peak at high energies, followed by a fission spectrum, a slowing-down spectrum typical of a fast reactor, and a thermal tail. The tail is provided to help give reasonable results in shields far from the core; its characteristic temperature is 300K. Note the sharp drop in the flux as energy decreases from 19 keV. This region is important for ^{238}U absorption, and this drop-off helps to give good group constants for fast reactors. Of course, it would be entirely wrong for a thermal reactor.

IWT=9 This is another typical thermal+$1/E$+fission+fusion function that has been used for many libraries at LANL in the 30-group structure. The CLAW-3 and CLAW-4 libraries are available from the Radiation Shielding Information Computational Center at ORNL.

IWT=10 This is the same as the CLAW weight function (IWT=9), but the shape of the thermal part is automatically recalculated to follow a Maxwellian law for temperature T.

IWT=11 This is the weight function used in the VITAMIN-E library. It has the following segments: a .0253-eV thermal Maxwellian below 0.414 eV, a $1/E$ law from .414 eV to 2.12 MeV, a 1.415-MeV fission spectrum from 2.12 to 10 MeV, another $1/E$ section from 10 to 12.52 MeV, a fusion peak (25 keV width) between 12.52 and 15.68 MeV, and a final $1/E$ section for all higher energies. The shape of the fusion peak is almost identical to IWT=5 (see Fig. 32). The low-energy part of this weight function is not shown.

Just as in the case of group structures, an arbitrary weight function can be read in (see `card8b`). This function is presented to GROUPR as an ENDF/B "TAB1" record. This means that a count of $E, C(E)$ pairs and one or more interpolation schemes are given.

An ENDF/B "TAB1" record consists of three distinct parts:

- two double values and four integer values of which only the last two integers (the number of interpolation ranges `NR` and the number of $E, C(E)$ pairs `NP`) are needed to read the remainder of the record
- the interpolation scheme data which is a sequence of `NR` pairs of `NBT` (index of the $E, C(E)$ pair corresponding to the end of interpolation range) and `INT` (the interpolation type)
- the tabulated data which is a sequence of `NP` $E, C(E)$ pairs

The interpolation type `INT` specifies the interpolation law to be used in the interpolation range. It can take the following values:

INT	Meaning
1	histogram
2	linear-linear
3	linear-log
4	log-linear
5	log-log

For example, INT=5 specifies that ln$C$ is a linear function of ln$E$. Similarly, INT=4 specifies that ln$C$ is a linear function of E.
In its most general form, these input cards would be

```
0.      0.      0       0       NR      NP
NBT(1)  INT(1)    ...   NBT(NR) INT(NR)
E(1)    C(1)      ...   E(NP)   C(NP)        / card8b
```

For example, a function using two interpolation ranges: the first one between 1e-5 eV and 100 eV using a histogram and the second one between 100 eV and 20 MeV using lin-lin interpolation)

```
0.      0.      0       0       2       6
3       1       6       2
1e-5    0.5     1.      0.75    100.    0.8
1e+6    0.85    1.5e+6  0.9     2.0e+7  1.0  / card8b
```

For the special case of a single interpolation scheme, the input cards are simplified as follows

```
0.      0.      0       0       1       NP
NP      INT
E(1)    C(1)      ...   E(NP)   C(NP)        / card8b
```

As many physical lines as are needed can be used for "card8b", as long as the terminating slash is included.

One of the weighting options, IWT=4, is a generalized "1/E+fission+thermal" function where the thermal temperature, fission temperature, and breakpoint en-

ergies (all in eV) are given on `card8c`. The weight function for the Los Alamos LIB-IV cross section library[63] used

```
.10 .025 820.3e3 1.4e6 / card8c
```

See Figures 31 and 32 for a plot of this function.

Several of these weight functions include a fusion peak. Because of the finite width of the distribution of ion energies in a D-T fusion plasma, the emitted 14-MeV neutrons clearly will not have a delta-function energy spectrum. In fact, owing to the presence of a cross-product term in the kinematic relations, the typical ion-energy spread of a few tens of kilovolts is magnified into a neutron-energy spread of around 1 MeV. For an assumed isotropic Maxwellian plasma, the neutron peak shape (for example, see the review article by Lehner[64]) is

$$S(E) = C \int_0^\infty \exp\{-b\left[v^2 + v_0(g)^2\right] - cg^2\} \sinh(2bvv_0) \frac{g^3\sigma(g)}{v_0(g)}\, dg \ . \qquad (368)$$

Here $S(E)$ is the number of neutrons emitted with laboratory energy between E and $E + dE$, C is a normalization factor, v is the laboratory velocity corresponding to energy E, and v_0 is the velocity of the neutron in the CM system. Both v_0 and the fusion cross section σ are determined by the relative velocity g of the reacting ions, the integration variable in Eq. 368. The coefficient b is equal to $M/2kT$, where M is the total mass of the reacting ions and kT is the plasma temperature. Similarly, c is $\mu/2kT$, where μ is the reduced mass of the ion system. The only approximation involved in Lehner's derivation of Eq. 368 is that all particles may be treated nonrelativistically. At 14 MeV, the relativistic factor

$$\gamma = \frac{1}{\sqrt{1 - v^2/c^2}} \ , \qquad (369)$$

is very close to unity (1.015), and it varies negligibly over the range of interest (say, 13 to 17 MeV). It is sufficient then to invoke relativistic mechanics in defining the location of the 14-MeV peak but not in discussing the shape. This effect moves the peak toward lower neutron energies, but only by about 20 keV. Although the expression for $S(E)$ in Eq. 368 is accurate, it has the disadvantage of requiring a numerical integration at each point E in the energy spectrum. For this reason, we consider what simplifications can be made without serious loss

of accuracy. In the energy range around the 14-MeV peak, the product $2bvv_0$ in Eq. 368 has a numerical value of about 6000. Thus, the hyperbolic sine can obviously be replaced by just the positive exponential term. If we make this change in Eq. 368, we can write the following, still nearly exact, expression for the neutron spectrum:

$$S_1(E) = C_1 \int_0^\infty \exp[-b(v - v_0)^2]\, P(v_0)\, dv_0 \ . \tag{370}$$

Here we also have inverted the function $v_0(g)$ and changed the integration variable. The spectrum then is a linear superposition of velocity exponentials with slightly different peak locations. For normal plasma temperatures, the velocity distribution $P(v_0)$ is very narrow, since the expression for v_0 is dominated by the nuclear Q-value (17.586 MeV) rather than the contribution from the ion kinetic energy (typically around 50 keV). Thus, it seems reasonable to approximate $P(v_0)$ as delta function

$$P(v_0) \approx \delta(v_0 - v_p) \ . \tag{371}$$

This gives a second approximate form,

$$S_2(E) = C_2 \exp[-b(v - v_p)^2] \ , \tag{372}$$

where v_p has the obvious meaning of the laboratory neutron velocity at the center of the peak. We shall refer to this as the velocity exponential form of the neutron energy spectrum. An expression essentially identical to Eq. 372 was given in an early paper by Nagle and coworkers[65]. In order to examine the accuracy of the velocity exponential form, we have calculated $S(E)$ from Eq. 368 and $S_2(E)$ from Eq. 372 at 20 keV, a typical plasma temperature in current fusion-reactor concepts. In performing the numerical integration over g in Eq. 368, we used numerical values for the D-T fusion cross section taken from the compilation by Jarmie and Seagrave[66]. In evaluating $S_2(E)$ using Eq. 372, a value of v_p was chosen so as to force agreement between $S_2(E)$ and $S(E)$ at 17 MeV. As discussed by Muir[67], the overall agreement is remarkably good, the maximum error over the range from 13.5 to 17 MeV being about 2%. The value of v_p thus derived corresponds to a peak-center energy E_p of 14.07 MeV. This value includes the small ($\sim$ 20 keV) relativistic correction mentioned above. If we approximate the mass of the D+T system as 5 times the neutron mass, then we obtain the recommended peak shape

$$S_{\rm rec}(E) = \exp\left[-\frac{5}{kT}(\sqrt{E} - \sqrt{E_p})^2\right] , \tag{373}$$

where E_p=14.07 MeV. The functional form in Eq. 373 was used to calculate the fusion peak shapes appearing in two of the data statements in `genwtf` (namely, those utilized for IWT=5 and 8) and is also used explicitly in `getwtf` to calculate analytically the weight function for IWT=6. In all three cases, kT=25 keV is used as an average or typical fusion-reactor plasma temperature. See Fig. 32 for a graphical display of the resulting weight functions in the 14-MeV region.

The GROUPR flux calculator is selected by a negative sign on `iwt`. The additional `card8a` is then read. The calculator option used is determined by the number of parameters given and their values. The parameters `fehi` and `nflmax` are used to select the energy range for the flux calculation, and they also determine the cost in time and storage. The actual value for `sigpot` is not very critical – a number near 10 barns is typical for fissionable materials.

Nonzero values for `ninwt` and `jsigz` will cause the computed flux for a given fissionable isotope (such as ^{238}U) to be written out onto a file. This saved flux can be used as input for a subsequent run for a fissile material (such as ^{239}Pu) with `iwt`=0 to get an approximate correction for resonance-resonance interference. See Eq. 278.

Nonzero values for some of the last five parameters on `card8a` select the extended flux calculation of Eq. 276. The simplest such calculation is for an isolated pin containing a heavy absorber with an admixed moderator. For $^{238}UO_2$, the card might be

```
400 10.6 5000 0 0 .7768 7.5 / card8a
```

where 7.5 barns is twice the oxygen cross section and α is computed from $\left[(A-1)/(A+1)\right]^2$ with $A = 15.858$. A more general case would be a PWR-like lattice of $^{238}UO_2$ fuel rods in water:

```
400 10.6 5000 0 0 .7768 7.5 .40 1.7e-7 0.086 / card8a
```

where 0.086 is computed using 3.75 barns for O and 40 barns for the two H atoms bound in water; that is,

$$\gamma = \frac{3.75}{2 * 20 + 3.75} = 0.086 \ . \tag{374}$$

A third example would compute the flux for a homogeneous mixture of ^{238}U and hydrogen

```
400 10.6 5000 0 0 0 0 1. 1.7e-7 / card8a
```

As a final example, consider a homogeneous mixture of uranium and water. This requires `beta`=1 and `sam`=0. Thus,

```
400 10.6 5000 0 0 .7768 0. 1. 1.7e-7 .086 / card8a
```

The maximum Legendre expansion order used for scattering matrices is set by `lord`. The number of tables produced is `lord`+1; that is, ℓ = 0, 1, ... `lord`. When more than 1 value of σ_0 is requested, both the ℓ=0 and ℓ=1 components of the total cross section are produced.

Card 3 contains a short descriptive title that is printed on the listing and added to the output GENDF tape. Card 4 gives the `ntemp` values of temperature for the run. They must be in ascending order, and if unresolved data are included on the PENDF tape, the temperatures in this list must match the first `ntemp` values in MF=2, MT=152 from UNRESR or PURR (see `stounr` and `getunr`). Card 5 gives the σ_0 values for the run in descending order, starting with infinity (represented by 10^{10} barns).

This completes the description of the global input parameters for GROUPR. The rest of the input cards request reactions to be processed for the various temperatures and materials desired. Because of the many types of data that it can process, GROUPR does not have a completely automatic mode for choosing reactions to be processed. On the basic level, it asks the user to request each separate cross section or group-to-group matrix using the parameters `mfd`, `mtd`, and `mtname`. However, simplified input modes are also available. For example, the one "`card9`" containing

```
3/
```

will process the cross section "vectors" for all of the reaction MT numbers found on the PENDF tape.

For completeness, the full input for `matd`, `mfd`, and `mtname` will be described first. Most readers can skip to the description of automated processing below. The value of `mfd` depends on the output desired (vector, matrix) and the form of the data on the ENDF evaluation. Simple cross section "vectors" σ_{xg} are requested using `mfd`=3 and the `mtd` numbers desired from the list of reactions available in the evaluation (check the directory in MF=1,MT=451 of the ENDF and PENDF tapes for the reactions available). A typical example would be

```
3   1 'total'/
3   2 'elastic'/
3  16 '(n,2n)'/
3  51 '(n,nprime)first'/
3 -66 '(n,nprime)next'/
3  91 '(n,nprime)continuum'/
3 102 'radiative capture'/
```

The combinations of "3 51" followed by "3 −66" means process all the reactions from 51 through 66; that is, (n,n$'_1$), (n,n$'_2$), ..., (n,n$'_{16}$). If self-shielding is requested, the following reactions will be processed using `nsigz` values of background cross section: total (MT=1), elastic (MT=2), fission (MT=18 and 19), radiative capture (MT=102), heat production (MT=301), kinematic KERMA (MT=443), and damage energy production (MT=444). The other File-3 reactions will be computed at σ_0=∞ only. This list of reactions can be altered by small changes in `init` if desired.

There are several special options for `mtd` available when processing cross section vectors:

mtd	Option
259	Average inverse neutron velocity for group in s/m.
258	Average lethargy for group.
251	Average elastic scattering cosine $\overline{\mu}$ computed from File 4.
252	Continuous-slowing-down parameter $\overline{\xi}$ (average logarithmic energy decrement for elastic scattering) computed from File 4.

253	Continuous-slowing-down parameter $\overline{\gamma}$ (the average of the square of the logarithmic energy decrement for elastic scattering, divided by twice the average logarithmic energy decrement for elastic scattering) computed from File 4.
452	$\overline{\nu}$: the average total fission yield computed from MF=1 and MF=3.
455	$\overline{\nu}^D$: the average delayed neutron yield computed from MF=1 and MF=3.
456	$\overline{\nu}^P$: the average prompt fission neutron yield computed from MF=1 and MF=3.

There are also some special options for `mfd` that can be used when processing cross sections:

mfd	Option
12	Photon production cross section computed from File 12 and File 3.
13	Photon production cross section computed from File 13. Recent versions of GROUPR will automatically shift between 12 and 13, if necessary.
1zzzaaam	nuclide production for zzzaaam from a subsection of MF=3
2zzzaaam	nuclide production for zzzaaam from a subsection of MF=6
3zzzaaam	nuclide production for zzzaaam from a subsection of MF=9
4zzzaaam	nuclide production for zzzaaam from a subsection of MF=10
40000000	fission product production from the MT=18 subsection of MF=10

An example of the isomer production capability would be the radiative capture reaction of ENDF/B-V ^{109}Ag(n,γ) from Tape 532:

```
30471090 102 '(n,g) TO g.s.'/
30471091 102 '(n,n) to isomer'/
```

Starting with ENDF/B-VIII.0, some non-fissile nuclides can have an MT=18 section (fission) in MF=10 (radioactive isotope production) to represent breakup due to high energy particles. In these cases, it is often not possible to designate a specific nuclide, which is why the `mfd` value is set to `40000000`. In such a case, the following input will make GROUPR process this part of the ENDF file:

```
40000000  18 'HE breakup' /
```

The next class of reactions usually processed is the group-to-group neutron scattering matrices. The complete list of `mtd` values is most easily found under File 4 in the MF=1,MT=451 "dictionary" section of the evaluation. An example follows:

```
6   2 'elastic matrix'/
6  16 '(n,2n) matrix'/
6  51 '(n,nprime)first matrix'/
6 -66 '(n,nprime)next matrix'/
6  91 '(n,nprime)continuum matrix'/  .
```

Using `mfd`=6 implies that File 4, or File 4 and File 5, will be used to generate the group-to-group matrix. The elastic matrix will be computed for `nsigz` values of background cross section, but the other reactions will be computed for σ_0=∞ only. The list of matrices to be self-shielded can be altered by changing `init`.

Fission is more complex. For the minor isotopes, only the total fission reaction is used, and the following input is appropriate for the prompt component:

```
3 18 'fission xsec'/
6 18 'prompt fission matrix'/
```

For the important isotopes, partial fission reactions are given. They are really not needed for most fission reactor problems, and the input above is adequate. However, for problems where high-energy neutrons are important, the following input should be used:

```
3 18 'total fission'/
3 19 '(n,f)'/
3 20 '(n,nf)'/
3 21 '(n,2nf)'/
3 38 '(n,3nf)'/
6 19 '(n,f)'/
```

```
6 20 '(n,nf)'/
6 21 '(n,2nf)'/
6 38 '(n,3nf)'/
```

Note that "6 18" is omitted because it will, in general, be different from the sum of the partial matrices (see Section 8.6). Some materials don't have data for (n,3nf); in these cases, omit the two lines with `mtd`=38 from the input. The fission matrix is not self-shielded. Since resonance-to-resonance fission-spectrum variations are not described in the ENDF format, it is sufficient to self-shield the cross section and then to use the self-shielding factor for the cross section to self-shield the fission neutron production.

Delayed fission data are available for the important actinide isotopes, and the following input to GROUPR is used to process them:

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

The line for `mfd`=5 automatically requests spectra for all time groups of delayed neutrons. The time constants are also extracted from the evaluation. As discussed in Section 8.6, formatting modules such as DTFR and CCCCR must combine the prompt and delayed fission data written onto the GENDF tape in order to obtain steady-state fission parameters for use in transport codes.

Starting with the ENDF-6 format, neutron production data may also be found in File 6, and `mfd`=8 is used to tell the code to use MF6 for this `mtd`. When using full input, the user will have to check the File 1 directory and determine what subsections occur in File 6.

Photon production reactions can be found in the ENDF dictionary under MF=12 and 13. To request a neutron-to-photon matrix, add 4 to this number.[8] For example,

```
17   3 'nonelastic photons'/
16   4 'inelastic photons'/
16  18 'fission photons'/
```

[8] In recent versions of NJOY, GROUPR will automatically shift between 16 and 17 using data read from the ENDF dictionary by the `conver` subroutine. Thus, use of `mfd`=17 is no longer necessary.

```
16 102 'capture photons'/
```

Yields (MF=12) are normally used with resonance reactions (MT=18 or MT=102), or for low-lying inelastic levels (MT=51, 52, ...). MT=3 is often used by evaluators as a catch-all reaction at high energies where it is difficult to separate the source reactions in total photon emission measurements. In these cases, photon production cross sections from other reactions like MT=102 are normally set equal to zero at high energies. The general rule for photon emission is that the total production is equal to the sum of all the partial production reactions given in the evaluation. Starting with the ENDF-6 format, photon production may also appear in File 6. Use `mfd=18` to process these contributions. Since resonance-to-resonance variations in photon spectra are not given in ENDF evaluations, GROUPR does not normally self-shield the photon production matrices (although this can be done if desired by making a small change in `init`); instead, it is assumed that only the corresponding cross section needs to be shielded. Subsequent codes can use the cross section self-shielding factor with the infinite-dilution photon production matrix to obtain self-shielded photon production numbers.

This version of GROUPR can also generate group-to-group matrices for charged-particle production from neutron reactions and for all kinds of matrices for incident charged particles. The incident particle is determined by the input tape mounted. The identity of the secondary particle is chosen by using one of the following special `mfd` values:

For distributions given in File 6 (energy-angle):

`mfd`	Meaning
21	proton production
22	deuteron production
23	triton production
24	^{3}He production
25	alpha production
26	residual nucleus (A>4) production

For distributions given in File 4 (angle only):

mfd	Meaning
31	proton production
32	deuteron production
33	triton production
34	^{3}He production
35	alpha production
36	residual nucleus (A>4) production

If necessary, mfd=21-26 will automatically change to 31-36.

The user will normally process all reactions of interest at the first temperature (for example, 300K). At higher temperatures, the threshold reactions should be omitted, because their cross sections do not change significantly with temperature except at the most extreme conditions. This means that only the following reactions should be included for the higher temperatures (if present): total (MT=1), elastic (MT=2), fission (MT=18), radiative capture (MT=102), heating (MT=301), kinematic KERMA (MT=443), damage (MT=444), and any thermal cross sections (MT=221-250). Only the elastic and thermal matrices should be included at the higher temperatures.

Warning: when using the explicit-input option, it is a fatal error to request a reaction that does not appear in the evaluation, cannot be computed from the evaluation, or was not added to the PENDF tape by a previous module. Reactions with thresholds above the upper boundary of the highest energy group will be skipped after printing a message on the output file.

Automated processing of essentially all reactions included in an ENDF/B evaluation is also available. As mentioned previously, the single card

```
3/
```

will process all the reactions found in File 3 of the input PENDF tape. However, this list excludes thermal data (MT=221-250) and special options such as mtd=251-253, 258-259, and 452-456. If any of these reactions are needed, they should be given explicitly (see example below). Similarly, the single card

```
6/
```

will process the group-to-group matrices for all reactions appearing in File 4 of the ENDF/B tape, except for MT=103-107 and thermal scattering matrices (MT=221-250). If MT=18 and 19 are both present, only MT=19 will be processed into a fission matrix. For ENDF-6 evaluations, the "`8/`" option will also process every neutron-producing subsection in File 6. Photon production cross sections are requested using

```
13/
```

and photon-production matrices are requested with the single card

```
16/
```

In both cases, all reactions in both File 12 and File 13 will be processed without the need for using `mfd`=12 or `mfd`=17. For ENDF-6 libraries, this option will also process all photon-production subsections in File 6. There is no automatic option for delayed neutron data. An example of a processing run for a fissionable isotope with thermal cross sections follows:

```
3/
3 221/ thermal xsec
3 229/ average inverse velocity
3 455/ delayed nubar
5 455/ delayed spectra
6/
6 221/ thermal matrix
16/ photon production matrix
```

An example of charged-particle processing for the incident-neutron part of a coupled n-p-γ library follows:

```
3/ cross sections
6/ neutron production matrix
16/ photon production matrix
```

```
21/ proton production matrix
```

The layout of data in a n-p-γ coupled set is shown in Figure 33.

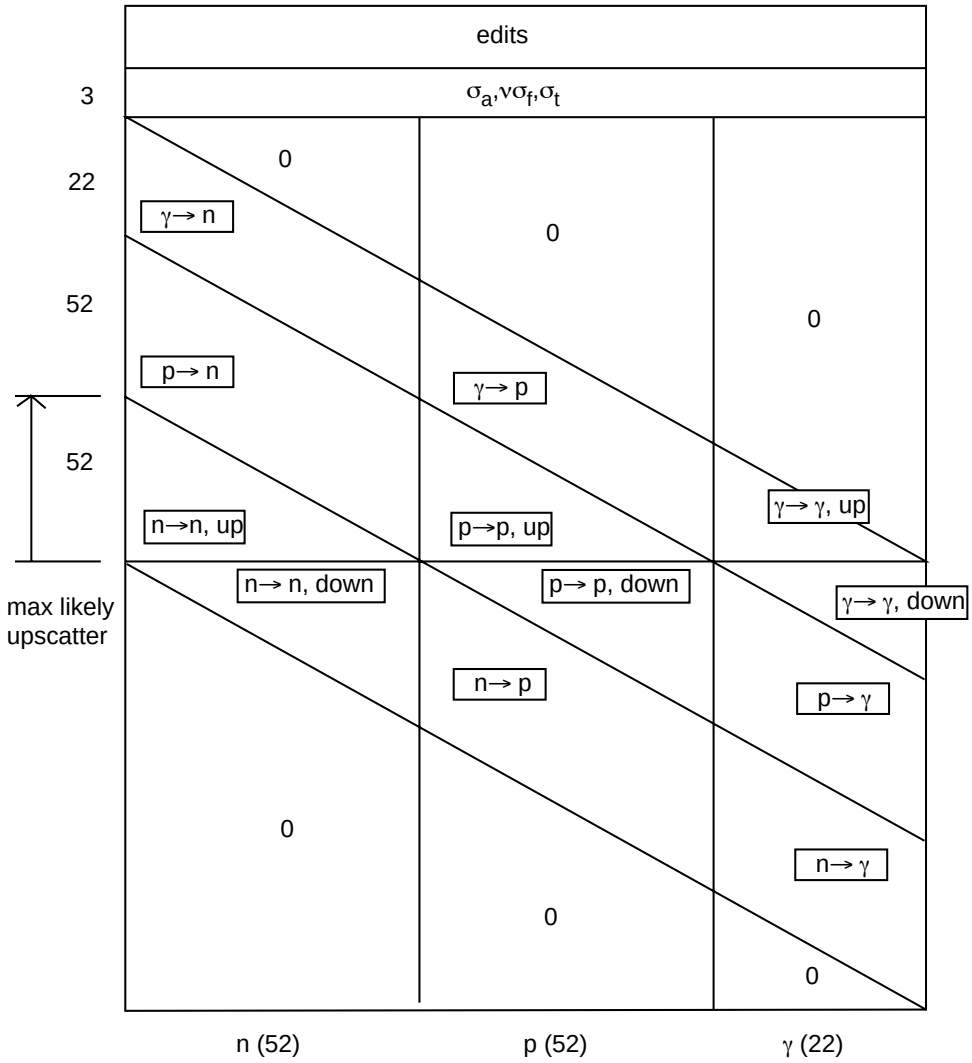

Figure 33: Layout of a coupled table for the simultaneous transport of neutrons, protons, and gamma rays. Normally, only the lower triangle of the "p to n" block would contain values for the upscatter portion of the table (the part above the line in the middle). The "group" index increases from left to right, and the "position" index increases from top to bottom.

There is a new ENDF format now becoming available that helps to describe the production of all isotopes and isomers in a given system. It uses a directory in File 8 to direct the code to productions represented by MF=3, by MF=9 times

MF=3, by MF=10, or by sections of MF=6 times MF=3. When this format is used, it is possible to give the simple command

```
10/
```

to have production cross sections generated for every nuclide that is produced. These production sections are labeled with the ZA and isomeric state of the products for use by subsequent NJOY modules.

8.19 Coding Details

The `groupr` subroutine is exported by the `groupm` module. GROUPR begins by reading most of the user's input (see `ruinb`). It then locates the desired material and temperature on the input ENDF, PENDF, and GENDF tapes, and reads in the self-shielded unresolved cross sections (if any) from the PENDF tape using `stounr`. If self-shielding was requested, `genflx` is used to compute the weighting flux as described in Section 8.4. The next step is to write the header record for this material on the output GENDF tape.

The code is now ready to begin the loop over reactions for this material and temperature. Either an input card is read to get `mfd`, `mtd` and `mtname` (the reaction name), or the next reaction in an automatic sequence is selected. First, the default Legendre order, secondary group count, and σ_0 count are selected for the reaction in `init`, and the retrieval routines are initialized. `groupr` then processes the reaction using the `panel` logic described in Section 8.11. If a "shortcut" fission spectrum was requested (`mfd=5`), for delayed fission, and for the low-energy "constant" spectra, the spectrum is calculated directly using `getff`. As the cross sections for each group are obtained, they are printed out (see `displa`) and written to the GENDF tape. When the last group has been processed, `groupr` loops back to read a new input card for a new reaction.

This loop over reactions continues until a terminating "`0/`" card is read. `groupr` then proceeds to the next temperature, if any, and repeats the loop over reactions. After the last temperature has been processed for the first material, an opportunity is provided to change to a new material, keeping all the other input parameters unchanged. A "`0/`" card at this point causes all the files to be closed, prints out the final messages, and terminates the `groupr` run.

Automatic choice of the next reaction to be processed is done in one of two ways. For a simple range of MT numbers, such as the example 51 - 66 used above,

the negative value is stored in the variable `mtdp` in the `groupr` subroutine. When `mtdp` is negative, `mtd` is incremented after each reaction until it is greater than the absolute value of `mtdp`. `mtdp` is then reset to one, and the code proceeds to the next input card. When processing photon data, `mfd`=`16` will automatically change to `17`, if necessary. Similarly, `mfd`=21, 22, ... will automatically change to `mfd`=31, 32, ... A more automatic method is triggered by `mtd`=0. In this case, a subroutine called `nextr` is called to return the next value of `mtd` to be used, and a subroutine called `namer` is called to generate the reaction name. For `mfd`=3, `nextr` finds the next reaction in File 3 on the input PENDF tape. MT=251-253 and thermal data (MT=221-250) are excluded. The MT values for the special options (258, 259, *etc.*) do not appear on the PENDF tape, and they must be requested explicitly. For matrices, GROUPR works with a set of lists loaded into global arrays by `conver`. The list `mf4` contains all the neutron-scattering MT numbers that appear in the File 4 part of the directory on the ENDF tape, and the list `mf6` contains all the MT numbers of sections of File 6 that contain subsections that produce neutrons. Therefore, reading through these two lists returns all the neutron-producing matrix reactions. Similarly, the list `mf12` contains all the File 12 entries from the directory, `mf13` contains the File 13 entries, and `mf18` contains all the MT values for sections in File 6 that contain subsections for photon production. Scanning through these three lists produces all the photon production matrix reactions. Two arrays are used for charged-particle producing reactions; the first index runs through the charged particles in the order p, d, t, ^{3}He, α, recoil. Taking proton production as an example, the list elements `mf6p(1,i)` contain the MT numbers of sections in File 6 that contain subsections that produce protons. The list elements `mf4r(1,i)` contain MT numbers from File 4 for two-body reactions that produce protons; namely, MT600 - MT648. `nextr` scans through both of these lists to return indexes to all the reactions that produce protons. The same procedure is used for the other charged particles. The arrays `mf10s` and `mf10i` are used in a similar way for nuclide production.

Subroutine `namer` generates name strings with up to 15 Hollerith words with 4 characters each (60 characters). The names depend on the "ZA" of the projectile and the MT number for the reaction. The parameter `mfd` is used to choose between the suffixes "cross section" and "matrix". Some examples of the names produced follow:

Name	Name	Name
(n,total)	(n,heat)	(p,p02)
(n,elastic)	(n,p02)	(p,n00)
(n,2n)	(p,elastic)	(g,total)
(n,n01)	(p,2n)	(g,pair)

Subroutines `mfchk` and `mfchk2` are used with the full input for reaction selection in GROUPR to check whether `mfd`=17 is needed when `mfd`=16 was requested, or whether one of the charged-particle File 4 values `mfd`=31-36 is needed when `mfd`=21-26 were requested. The lists in global variables like `mf12` are used, just as for `nextr`.

Subroutine `gengpn` generates the group bounds in the global array `egn` for the neutron group structure from input cards, from data statements, or by calculation. Some of the data statements use energies in eV and some use lethargy. Similarly, `gengpg` generates the photon group structure in global array `egg` from input cards or data statements; in this case, all bounds are in eV.

Subroutine `genwtf` sets up the weight function option requested with `iwt` by reading the input cards into `weights`, transferring numbers from data statements to `weights`, or simply reporting the analytic weight option requested. Subroutine `getwtf` returns the values of the weight function at energy E by calculation or by interpolation in the table established by `genwtf`. The current version returns the same value for all Legendre orders. Choosing `enext` is difficult for `getwtf` because the functions have not been explicitly linearized. It is important to generate extra grid points in energy regions where the weight function may vary faster than the cross section (for example, in the fusion peak).

Subroutine `genflx` computes the self-shielded weighting flux using either the Bondarenko model or the flux calculator, and it writes the result on a scratch tape using the `loada` utility routine. When fluxes are needed for the generalized group integrals, they are read from this scratch file using `finda` (see `getflx`). Subroutine `genflx` starts by checking for the weighting option. If the Bondarenko model was selected, it initializes `gety1` and `getwtf` to read the total cross section from the PENDF tape and the smooth weighting function $C(E)$ set up by `genwtf`. It then steps through the union grid of $\sigma_t(E)$ and $C(E)$ computing the flux vs. σ_0 and Legendre order by means of Eq. 253. In the unresolved energy range, `getunr` is used to retrieve the unresolved cross sections as a function of σ_0, and Eq. 261 is used to compute the weighting flux. In both cases, the data at each energy are stored as the 1+(`lord`+1)*`nsigz` components

```
e, phi(il,iz),
```

where il runs from 1 to lord+1 and iz runs from 1 to nsigz.

If the flux calculator option was requested, genflx sets up the parameters for the calculation and requests needed storage space. Next, the cross section retrieval routines gety1 and gety2 are set up to return total and elastic cross sections from the PENDF tape. A lower energy limit, felo, is chosen, and the cross sections are read into storage until a maximum energy, fehi, or a maximum number of points, nemax, is reached.

The slowing-down equation, either Eq. 272 or Eq. 276, is then solved from the break energy down to felo. The scattering source from energies above the break is based on the NR approximation. When the calculation is finished, fluxes from 10^{-5} eV to felo are written to the scratch file using loada in the same format used for the Bondarenko option. From felo to the energy break point, fluxes are transferred from memory to the loada file. Finally, above the break point, fluxes are computed and saved using the Bondarenko model.

Subroutine init is used to set up the number of σ_0 values, secondary energy groups, and Legendre components for each combination of mfd and mtd. If mfd=8, a special copy of File 6 is made for use in getaed. The list of reactions to be self-shielded can be changed if desired. The number of secondary groups ng helps determine the storage required for the accumulating group integrals (see allocatable array ans) in groupr. For simple cross section vectors, ng=2. The nz*nl flux components are stored first, followed by the nz*nl cross section components. When all the panels for one group have been processed, dividing position 2 by position 1 gives the group-averaged cross section. For ratio quantities like $\overline{\nu}$ and $\overline{\mu}$ (mtd=251-253, 452, 455, 456), ng is 3. Once again, the flux components are stored first, followed by the nl*nz components of ratio*sigma, followed by the components of the cross section. This arrangement allows for the calculation of group-averaged values of ratio*sigma, ratio, or cross section by dividing position 2 by 1, position 2 by 3, or position 3 by 1, respectively. For matrices, ng is set to one more than the number of secondary groups (ngn or ngg). The nl*nz flux components are stored first, followed by the nl*nz integrals for each secondary-energy group in turn.

Subroutine panel performs the generalized group integrals using the logic described in Section 8.11. For most calls to panel, the lower point of each "panel" was computed as the upper point of the previous panel. Therefore, panel is careful to save these previous values. However, if the bottom of the panel is just above a discontinuity, new values of cross section and flux are retrieved.

Once the values at the lower boundary of the panel are in place, new values for the top of the panel are retrieved (see `flux`, `sig`). If the top of the panel is at a discontinuity in σ or at a group boundary, the energy used is just below the nominal top of the panel. "Just below" and "just above" are determined by `rndoff` and `delta`. For maximum accuracy, these numbers should be chosen such that `rndoff`>1, `delta`<1, and `rndoff`*`delta`<1 for the precision of the machine being used.

For simple average cross sections, the integrals of $\sigma\times\phi$ and ϕ are computed for the panel using trapezoids. This is justified by the linearization of σ; the value of $\sigma\times\phi$ at the midpoint is too uncertain to justify a more complex treatment. For two-body scattering, the feed function is far from linear over the panel. In fact, it can show oscillations as described in Section 8.12. The integral of the triple product $\mathcal{F}\times\sigma\times\phi$ is obtained by Lobatto quadrature of order 6 or 10 using the quadrature points and weights given in the parameter statements (see `qp6`, `qw6`, `qp10`, `qw10`). The cross section and reaction rate are determined at each quadrature point by interpolation, and the feed function is obtained by `getff`. For many reactions, `ff` will be nonzero for only a certain range of secondary groups. The value `ig1` is the index to the first nonzero result, and `ng1` is the number of nonzero values of `ff` in the range. Subroutine `panel` maintains the two corresponding values `iglo` and `ng` to specify the nonzero range of values in array `ans`. Finally, the flux and cross section at the top of the panel are transferred to `flst` and `slst`, and control is returned to the panel loop in `groupr`.

Subroutine `displa` is used to print cross sections and group-to-group matrices on the output listing (`nsyso`). Small values are removed for efficiency. Note that different formats are used in different circumstances. Infinitely dilute data are printed without σ_0 labels. Isotropic matrices are printed with several final groups on each line. Delayed neutron spectra are printed using the Legendre order variable for time groups and with the time constants given on a heading line.

Subroutine `getflx` returns `nl`*`nz` components of the weighting flux. If `nsigz` is 1, the flux is computed using `getwtf`, and all Legendre orders are taken to be equal. When self-shielding has been requested, the flux components are obtained by interpolating between adjacent values retrieved with `finda` from the scratch tape written by `genflx`. The grid energies found on the scratch tape are used to get `enext`. The flux is taken to be continuous, so `idis` is always set to zero.

Subroutine `getyld` returns the yield needed by `getff` for fission (MT=452, 455, or 456 from File 1) or radionuclide production yields from File 9 (using

mfd=3zzzaaam). This routine also retrieves the delayed neutron time constants when mtd=455. Tabulated yields are obtained by interpolation using the utility routine terpa. Polynomial data are expanded by direct computation.

Subroutine getsig returns nl*nz components of the cross section using point data from the PENDF tape and self-shielded unresolved data, if present, from getunr. The routine starts by adjusting MF and MT for the special options (mtd=258-259, mfd=3zzzaaam, *etc.*) and locating the desired section on the PENDF tape. Subroutine getsig is then called for each desired energy value E in increasing order. For mtd=258 or 259, the appropriate velocity or lethargy is computed from E and returned. In the more general cases, gety1 is used to retrieve the pointwise cross section from the PENDF tape, and getunr is called to replace this value with self-shielded unresolved cross sections if necessary.

The unresolved cross sections are handled using stounr and getunr. Subroutine stounr locates the desired material and temperature in MF=2, MT=152 of the PENDF tape, and then it copies the data into the global allocatable storage array unr. The σ_0 grid on the PENDF tape does not have to agree with the list requested for groupr, but a diagnostic message will be printed if they are different. However, the ntemp values of temperature requested by groupr must agree with the first ntemp temperatures on the PENDF tape, or a fatal error will result. Subroutine getunr checks whether E is in the unresolved energy range and whether MT is one of the resonance reactions. If so, it locates the desired interpolation range in the unr array, and interpolates for the self-shielded cross sections. If this is an energy range where resolved and unresolved ranges overlap, the resolved part is added to the background σ_0 before interpolation. The subroutine terpu is used for interpolating in the unresolved cross section tables. The special value MT=261 is used to select the ℓ=1 component of the total cross section. Note that getsig also uses this ℓ=1 value for $\ell=2, 3, \ldots$.

Subroutine getff returns the feed function ff using different portions of the coding for different options. The first section is used for cross sections and ratio quantities. The same yield yld is returned for every ℓ-component in ff.

The second section of getff is for neutron continuum transfer matrices. The yield is either determined from MT [for example, yld=2 for MT=16, the (n,2n) reaction], or it is obtained using getyld for fission. Next, the angular distribution is obtained using getfle (see F in Eq. 328), and the secondary energy distribution is obtained using getsed (see g in Eq. 328). Finally, the product of the three factors is loaded in ff. Note that the range of groups returned extends from iglo=1 to the highest nonzero result, for a total of ng groups. Since ff for

these reactions is a smooth function of incident energy, nq is set to zero, and no additional quadrature points will be used in panel.

The third section of getff is for gamma production matrices. The photon yields are obtained using getyld. In general, there are nyl different yields, each one corresponding to a different discrete gamma ray, or to the continuum. The angular distributions for these gamma rays are obtained using getgfl (most ENDF/B photons are given as isotropic). The energy distribution for the continuum (if any) is obtained by getsed. Now, the code loops through the photon group structure placing each discrete photon in the appropriate group and adding in the continuum part. During this loop, the range of nonzero values (iglo, ighi) is determined. Finally, the nonzero values are packed into ff, and iglo and ng are returned to describe the distribution. Again, nq=0 is used.

The fourth section of getff handles two-body scattering, either elastic or discrete-level inelastic, and both neutrons and charged particles. First, subroutine parts is called to set up the particle type, A' value, and the scattering law for reactions that use File 4. Then getdis is called to finish the processing.

The fifth section handles thermal-neutron scattering. The bulk of the work is done by getaed, which is discussed below. Subroutine getff takes the output of getaed and packs it into the final result, ff.

The last two sections of getff are used to process energy-angle distributions from ENDF-6 format sections of File 6. The subroutine simply calls getmf6.

Subroutine parts is used to set up particles for reactions that use File 4. Different branches are used for ENDF-6 and earlier ENDF versions because the MT numbers used for charged-particle discrete levels have been changed. For example, proton levels use MT=600-649 in ENDF-6 libraries, but MT=700-719 in earlier libraries. If mfd=3, no action is taken. For other mfd values, the routine determines zap, aprime, and law. The result depends on whether File 4 or File 6 data are to be used.

Subroutine getmf6 is used to compute the feed function for reactions represented in File 6. As is common with NJOY subroutines, it is called with ed=0 to initialize the subroutine. The first step is to locate the desired section of File 6 on the input ENDF tape. It then sets the parameter zad (for ZA desired) based on the input value mfd and searches through the section for the desired subsection. If this subsection has law=4, it defines a recoil particle; the code backs up to the first subsection in the section, which is assumed to be the subsection describing the emitted particle. The subroutine has now arrived at statement number 140, where it decides whether to branch to special coding for two-body

reactions (statement number 500) or phase-space reactions (statement number 150). All the other options have a TAB2 record at this point that defines the incident energy grid. For `law`=1, the data for the first incident energy are read in and converted to the Lab frame by `cm2lab`. Similarly, for `law`=7, the data are read in and converted to a `law`=1 format by `ll2lab`.

For a normal entry, `getmf6` interpolates for the particle yield for the current particle. For all of the laws except 2-5 (discrete two-body laws), it then checks to see whether the energy `ed` is in the current panel (or if this is the first time in the first panel). If not, it moves the high data to the low position and reads in new high data. Of course, it also converts the new high data to the correct form with `cm2lab` or `ll2lab`. Once `ed` is in the current panel, the code reaches statement number 300, which sets up the loop over secondary energy by initializing `f6lab`. This loop uses a grid that consists of the `epnext` values returned by `f6lab` and the group bounds `eg(i)`. The actual integral over E'_L inside each group uses trapezoidal integration. The last step for laws 1, 6, and 7 is to add in the contributions to the feed function from discrete energies in File 6. For the discrete two-body laws (2-5), the routine goes through statement number 500, which simply calls `getdis`. For all laws, the last step is to check the normalization of the feed function. If the error is large, a message is printed out. In any case, the results are adjusted to preserve exact normalization.

Subroutine `cm2lab` is used to convert a CM distribution starting at `inow` in `cnow` into a Lab distribution, which will be stored starting at `jnow`. This routine generates a grid for the Lab secondary energy E'_L by adaptive reconstruction. The reconstruction stack is first primed with $p_{L\ell}(E, 0)$ as computed by `f6cm` and $p_{L\ell}(E, E'_{\rm next})$ (also from `f6cm`), where the value $E'_{\rm next}$ is the value `epnext` returned by the first call to `f6cm`. This panel is then divided in half, and the value returned by `f6cm` is compared with the linear interpolate. If they agree within 0.5% (see `tol`), this panel is converged. Otherwise, it is subdivided further. When convergence has been achieved for this panel, a new panel is chosen, and it is subdivided to convergence. When the entire range for E'_L has been processed, the final parameters are loaded into the new File 6 record for $p_{L\ell}(E, E'_L)$ (which starts at `jnow` in `cnow`), and the routine returns to `getmf6`.

Subroutine `f6cm` is used to compute the Legendre coefficients of the double-differential scattering function $p_\ell(E \to E')$ in the Lab system from data given in File 6 in the CM frame. The memory area `cnow` contains the raw data. It is necessary to call the routine `ep`=0 for each new value of E. Thereafter, values of `ep` can be requested in any order. This is required by the adaptive scheme

used to generate the Lab E' grid in `cm2lab`. The conversion uses Eqs. 361 - 366. On a normal entry with E'_L>0, the code computes the lower limit of the integral $\mu_{\min}$. It then sets up an adaptive integration over a panel starting at μ=1. The value for E'_C is computed using

$$E'_C = (1 + c^2 - 2c\mu_L)E'_L \ , \tag{375}$$

which is based on Eq. 362, and either `f6ddx` or `f6psp` is called to compute $p_C(E, E'_C, \mu_C)$ and `epnext`. The lower limit of the integration panel is then computed by setting E'_C equal to `epnext` and solving for the CM cosine using

$$\mu_C = \frac{\mu_L - c}{\sqrt{1 + c^2 - 2c\mu_L}} \ . \tag{376}$$

Of course, the larger of this result and $\mu_{\min}$ is used as the lower bound of the panel. Therefore, the adaptive integration operates on a nice continuous function. It continues subdividing the μ_C grid in this panel until convergence is achieved within 0.5% for all Legendre orders. The lower bound of this panel becomes the upper bound of a new panel, and a new lower bound is selected as above. The integration is carried out over successive panels until $\mu_{\min}$ is reached. When the calculation of the continuum part of $p_{L\ell}$ is finished for these values of E and E'_L, the routine checks for possible contributions from delta functions. Next, the routine scans through the computed coefficients and zeros out any small ones that may just be noise. Finally, it chooses a value for `epnext` and returns.

Function `f6ddx` is used to compute the double-differential scattering function $f(E \to E', \omega)$, where secondary energy E' and scattering cosine ω are in the CM system. On entry, `cnow` contains the File 6 data for a particular value of E, and `f6ddx` must be called once with `ep`=0 for each new value of E. Thereafter, `ep` values can be requested in any order (this is required by the adaptive scheme used to convert to the Lab system in `f6cm`). For a normal entry with `ep`>0, the code searches for the panel in the data that contain the requested value of E'. If `lang`=1 for this subsection of File 6, the data are already given as Legendre coefficients, and the code simply interpolates for the desired results. If `lang`=2, the data use the Kalbach-Mann scheme for representing the energy-angle distribution. This routine includes both the original Kalbach-Mann representation[59] and the newer Kalbach representation[60]. It has been set to use the latter by the parameter statement `k86`=1. The code interpolates for the model parameters at E' and computes the desired answer with the model's formulas. The

ENDF-6 format also allows CM distributions given as values tabulated versus scattering cosine (see `lang`=11-15). Note that there is a "stub" to take special action at low energies. It is currently disabled by the statement `efirst`=0.0, but it may be used sometime in the future to account for the fact that the low-energy dependence of the scattering function must vary as $f(E') = c * \sqrt{E'}$.

Function `bach` computes the Kalbach-86 a parameter, which depends on the neutron separation energy for the target `izat` using the liquid-drop model without pairing and shell terms. The formula used describes the reaction $a + B \rightarrow C + d$:

$$\begin{aligned} E_{sep} &= 15.68\big(A_C - A_B\big) \\ &- 28.07\big(\frac{(N_C - Z_C)^2}{A_C} - \frac{(N_B - Z_B)^2}{A_B}\big) \\ &- 18.56\big(A_C^{2/3} - A_B^{2/3}\big) \\ &+ 33.22\big(\frac{(N_C - Z_C)^2}{A_C^{4/3}} - \frac{(N_B - Z_B)^2}{A_B^{4/3}}\big) \\ &- 0.717\big(\frac{Z_C^2}{A_C^{1/3}} - \frac{Z_B^2}{A_B^{1/3}}\big) \\ &+ 1.211\big(\frac{Z_C^2}{A_C} - \frac{Z_B^2}{A_B}\big) \ , \end{aligned} \tag{377}$$

where A stands for atomic weight, Z for charge number, and N for neutron number. Note that, even for reactions like (n,2n), C is the residual nucleus resulting from the removal of one particle, d; it is not necessarily the real physical residual nucleus for the reaction. If the target `izat` is an element, E_{sep} has to be computed for some dominant isotope in the element. Dominant isotopes are assigned in this routine for some materials that often appear as elements in ENDF evaluations; if the particular target required does not appear, a fatal error message is issued. The user will have to add a line to the routine for the material and reassemble the code.

Subroutine `ll2lab` converts File 6 from Law-7 format to Law-1 format using the laboratory Legendre representation. Law 7 represents the double-differential scattering distribution $f(E \rightarrow E', \mu)$ by giving a series of tables of $f(E \rightarrow E')$ for series of μ values. On entry, the entire Law-7 section is stored in `c` starting at `inow`. The code loops through a set of E' values chosen to be the union of all the E' grids for all the different μ values. For each point on this union grid, the code interpolates for all the corresponding $f(\mu)$ values, and it uses them to compute

`nl` Legendre coefficients $f_\ell(E')$. After completing the calculation for a new value of E', it checks the normalization of the result, and then it checks back to see whether the previous value is still needed to represent the curves $f_\ell(E')$ within a tolerance of 0.5%. The results are written in `c` starting at `jnow`. They have the Law-1 Legendre form, namely, sets of values E', $f_0(E')$, $f_1(E')$, ..., $f_{\mathrm{NL}-1}(E')$, given for a series of E' values starting with zero.

Subroutine `f6lab` is used to return the Legendre coefficients of the double-differential scattering function $f_\ell(E{\to}E')$ for particular values of E and E' in the Lab system (see `e` and `ep`) from a part of File 6 in Law-1 format. Since Law-7 sections have been converted to Law-1 format using `ll2lab`, and since CM sections in Law-1 format have been converted to the Lab frame using `cm2lab`, the only two roles left for this subroutine are interpolation to the desired values of E and E' and preparation of Legendre coefficients for sections that use the File-6 variant with `law`=1 and `lang`=11-15 (laboratory distributions tabulated) vs. μ. Only the continuum portion of the distribution is processed here; any delta functions given in File 6 must be handled separately. The routine is initialized by calling it with `ep`=0. Data for the two incident energy values that bracket E are already present in `clo` and `chi`. Then the routine extracts various parameters from the `c` array and prepares the variables used to control interpolation. These variables are complicated, because this routine handles three different interpolation schemes: "cartesian", "unit base", and "corresponding point" (see the ENDF-6 manual[9] for more details). For a normal entry, the code searches the data in `clo` and `chi` for the intervals containing E'. If `lang`=1, it performs a two-dimensional interpolation for the `nl` coefficients at E and E'. For `lang`>1, it computes the Legendre coefficients from the File 6 data, and then does the two-dimensional interpolation. Finally, it computes `epnext` and returns.

Subroutine `getdis` is used to compute the feed function for elastic or discrete inelastic scattering of neutrons or charged particles using either File 4 or File 6 data. First, the angular distribution is retrieved with `getfle`, and an appropriate quadrature order is selected using Eq. 319. Then, a group loop from low energy to high energy is used to compute the ω_1 and ω_2 limits of Eq. 317, and the range (ω_1, ω_2) is subdivided using the appropriate Gauss-Legendre quadrature points (see `qp4`, `qp8`, `qp12`, and `qp20`). The function $f(E, \omega)$ is computed at each of these quadrature points using Eq. 318 and the angular distribution previously returned by `getfle`. The laboratory cosine at the quadrature point, $\mu\big[\omega\big]$, is computed using Eq. 314. Finally, the integrand of Eq. 317 is multiplied by the appropriate quadrature weight (`qw4`, `qw8`, `qw12`, or `qw20`) and added into the

accumulating integral. This continues until $\omega_2 = 1$. The nonzero values of `ff` and the parameters `iglo` and `ng` are then complete for this value of E.

The next step is to determine `enext` based on the next critical point as given by Eqs. 321 - 324. Special cases are used for elastic scattering to avoid numerical problems. Note that the discontinuity flag `idisc` is set for critical points. The `nq` variable is also set to force `panel` to subdivide the initial-energy integration.

Subroutine `getfle` retrieves or computes the Legendre coefficients for the angular distribution of a reaction at incident energy E. When called with `e`=0, `getfle` requests storage for the raw data, reads in the File 4 information for the first two incident energies on the file, and then uses `getco` to retrieve the corresponding coefficients. On subsequent entries with `e`>0, `getfle` simply interpolates for the desired coefficients. When `e` exceeds the upper energy in storage, the values for the upper energy are moved to the lower positions, and new upper values are read and converted to coefficients. An isotropic distribution is returned if `e` is outside the range of the angular data from File 4.

Subroutine `getaed` retrieves angle-energy data for thermal scattering reactions. For coherent elastic scattering, the routine reads through the cross section on the PENDF tape using `gety1` and locates the Bragg edges. On each subsequent call to `getaed`, the Legendre components of the cross section are computed using Eq. 309. For incoherent elastic scattering, the routine is first initialized by reading in the raw data for the first energy. On subsequent entries, a test is made to see whether `e` is in the range `elo` to `ehi`. If not, the high data are moved to the low positions, and new high data are read. The Legendre components are then computed using Eq. 311. For incoherent inelastic scattering, `getaed` is initialized by reading in the raw data for the first two incident energies. On subsequent entries, the subroutine checks to see whether `e` is between `elo` and `ehi`. If not, the data for `ehi` are moved to the low positions, and new raw data are read from File 6 and binned. Once the correct data are in place, the desired energy-angle distribution is computed by using a combination of interpolation along lines of constant energy transfer and unit-base interpolation.

Subroutine `getgfl` returns the Legendre coefficients for the angular distributions for all discrete and continuum photons for a reaction simultaneously. When called with `ed`=0, the routine reads File 14 into scratch storage and finds the starting location for the subsection describing each photon. On subsequent entries with `ed`>0, `getgfl` sets up a loop over the `ng` photons on this section of File 14. For each photon, it searches for the energy panel that contains `ed`, uses `getco` to retrieve or compute the Legendre coefficients at the upper and

lower File 14 energies, and interpolates for the desired coefficients at `ed` using `terp1`. Since most ENDF photons are represented as isotropic, a special shortcut calculation is provided for that case. Isotropic results are also returned if `ed` is outside the range of the data in File 14.

Subroutine `getco` is used by both `getfle` and `getgfl` to retrieve or compute Legendre coefficients from data in File 4 or File 14 format. The user can request output in either the Lab or CM system, and the raw data can be either Legendre coefficients or tabulated probability versus emission cosine. However, if the raw data are in the laboratory system, CM coefficients cannot be produced. If the raw data are already in the form of coefficients in the desired system, `getco` simply checks for the maximum Legendre order needed using a tolerance of `toler`=1e-6 and returns the coefficients in `fl` and the order in `nl`. If coordinate conversion is required, or if the raw data are tabulated, `getco` sets up the integral over cosine using Gauss-Legendre quadrature of order 20 (see `qp` and `qw`). The scattering probability for the quadrature point is computed from the coefficients or obtained by interpolating in the tabulation. The Legendre polynomials in the desired reference system are then computed. If the raw data are in the CM system (ω) and the result is to be in the Lab system (μ), the desired polynomials are $P_\ell(\mu[\omega])$; otherwise, the quadrature angle is used directly to compute the polynomials. Once the coefficients have been computed, they are checked using `toler` to determine the maximum order, `nl`, and the results are returned in `fl`.

Subroutine `getgyl` is used to retrieve the yields for all photons emitted in a specified reaction simultaneously. The raw data are obtained from the ENDF/B tape as either yields (MF=12) or production cross section (MF=13). In the latter case, `getgyl` actually returns the fraction of the total yield assigned to each photon. The cross section returned by `getsig` is the total photon production cross section from MF=13 on the PENDF tape, which makes the resulting integral correct. Using the normal GROUPR procedure, `getgyl` is initialized by calling it with `ed`=0. The entire File 12 or File 13 is read into scratch storage, and the starting location for each subsection is determined. On subsequent entries (`ed`>0), the routine loops over the `nyl` photons found, and uses `terpa` to compute the yield at `ed`. If this is a primary photon, a discontinuity is set up at the energy where the photon will change groups. For MF=12, the calculation is finished. For MF=13, the numbers calculated above are converted to fractions of the total yield by dividing by the total production subsection from File 13. This routine does not handle ENDF/B transition probability arrays directly, because they will have been converted to File 12 yields by `conver`.

Subroutine conver converts ENDF/B evaluations to a standard form. If transition probability arrays were used in File 12, they are converted to yields and written back into File 12. If a section with MF=1 and MT=456 is missing from the evaluation, a copy of MT=452 is added to the tape as MT=456. In addition, a second copy of the modified tape is made on unit nscr. While conver is reading through the tape, lists of the reactions in File 4, File 12, and File 13 are written to to global arrays for use by the automatic reaction selection logic in nextr. For ENDF-6 evaluations containing File 6, the routine scans through File 6 looking for sections that produce neutrons, photons, or charged particles. The MT numbers for these sections are stored into mf6, mf18, and mf6p. The routine also checks for sections of File 4 containing charged-particle angular distributions and records their MT numbers in mf4r. Finally, if the section MF=6/MT=2 contains charged-particle elastic scattering information given using the nuclear-plus-interference format, it is converted into the residual-cross-section format for getdis.

Subroutine getsed returns the secondary-energy distribution for neutrons or continuum photons for all groups simultaneously. Both tabulated and analytic functions are handled. getsed is initialized for a particular reaction by calling it with ed=0. First, scratch storage is allocated, and all the subsections are read in. Analytic subsections are left in their raw form, but tabulated subsections are averaged over outgoing energy groups for each of the given incident energies. The array loc contains pointers for each subsection. On subsequent entries (ed>0), getsed loops over the subsections for this reaction. It first retrieves the fractional probability for the subsection using terpa. If an analytic law is specified, anased is used to compute the group integral for each secondary-energy group. Each integral is multiplied by the fractional probability for the law and accumulated into sed. For tabulated data, the routine simply interpolates between the two values for the group integrals using terp1, and accumulates them into sed. Note that various restrictions on the ordering of subsections and prohibition of multiple tabulated subsections needed for earlier versions of GROUPR are no longer required. Upscatter is not allowed in secondary-energy distributions except for fission or photon production. If found, it is put into the "in-group" (g'=g).

Subroutine anased is used to calculate the integral from e1 to e2 for one of the analytic laws (see Eqs. 331 - 360). The routine uses the SLATEC version of the reduced complementary error function from the NJOY2016 math module. The resulting integral is returned in g.

Subroutine `hnab` is used to compute the special functions required for analytic law 11, the energy-dependent Watt spectrum. The method used is described in the BROADR chapter of this manual.

8.20 Error Messages

The fatal error messages and warning messages from GROUPR are listed below, along with recommended actions to recover from the problem.

`error in groupr***unable to find mat=--, t=--.`
: Check for input error or wrong PENDF tape mounted.

`error in groupr***photons not allowed with igg=0.`
: In order to produce photon data, a photon group structure must be requested. Check the input on card 2.

`error in groupr***illegal mfd.`
: Check input `mfd`; legal values are 3, 5, 6, 8, 12, 13, 16, 17, 18, 21-26, 31-36, and the isotope production values like Xzzzaaam for X = 1, 2, 3 or 4.

`message from groupr--auto finds no reactions for mf=-.`
: An automatic reaction selection card of the form "`mfd`" was given in the input, but the ENDF and PENDF tapes do not contain any sections that would produce the desired cross sections or matrices.

`error in groupr***unable to find next temp.`
: The current material ended before the requested temperature was found.

`error in ruinb***illegal ismooth.`
: Check the input, the value for the smoothing option must be either 0 or 1.

`error in gengpn***read-in group structure is out of order.`
: Group structures must be given in ascending energy order.

`error in gengpn***illegal group structure.`
: Check input; current legal values are 1 through 34.

`error in gengpg***illegal group structure.`
: Check input; current legal values are 1 through 10.

`error in genwtf***exceeded storage reading user weight function`
: See the allocatable array `tmp` with length `ntmp`=10000.

`error in genwtf***illegal weight function.`
: Check input; current legal values are −12 through +12.

`error in genflx***total not defined over energy range.`
: A complete total cross section is needed for self-shielding. This means that "dosimetry" and "activation" tapes, which normally give only a few key reactions, can only be processed using `nsigz`=1 (infinite dilution).

`error in getfwt***temperature ...`

The requested temperature could not be found while searching the tape. NJOY uses a tolerance of 1e-6 for this purpose.

`error in getfwt***e outside range of data.`

A premature end-of-file was found on the input flux tape when using the `iwt`=0 option. Check to be sure the tape was the output from a legal flux calculator run.

`error in getfwt***requested e is out of order.`

The cause could be an improper input tape. Check as above.

`error in panel***elo.gt.ehi.`

This indicates some error in the energy grid for `panel`. It usually occurs if `rndoff` and `delta` are incorrect for your machine. Make sure that `rndoff`>1, that `delta`<1, and that the product `rndoff*delta`< 1 when evaluated on your machine (for example, 1.00000001 is not greater than unity on a 32-bit machine).

`error in panel***bad nq in panel`

The nq parameter can be 2, 6, or 10 with the currently installed quadratures.

`message from panel--thermal range problem at ...`

NJOY expected scattering to a specific group but found another instead.

`error in getyld***illegal lnd.`

The maximum number of time groups is 8. See global array `dntc(8)`.

`error in getyld***unable to find nuclide for iza=... lfs=...`

Unable to find requested nuclide production yield.

`error in getsig***illegal mt.`

Check input for `mtd`.

`error in getsig***can't find mf,mt,lfs ...`

Check input for `mfd` greater than 10000000.

`message from stounr--no unresolved sigma zero data....`

This message probably means that UNRESR or PURR was never run for this material. Infinitely dilute values will be used.

`message from stounr--sigma zero grids do not match....`

The unresolved calculations will probably work best if the σ_0 grid in GROUPR matches the one in UNRESR or PURR. However, this is not necessary. `getunr` will interpolate to get values on the GROUPR grid from the UNRESR or PURR grid. A message is issued in case this isn't what the user really intended.

`error in stounr***storage exceeded.`

There is not enough storage for unresolved cross section data on PENDF tape. The allocatable array `tmp` has length `ntmp`=10000.

`error in stounr***cannot find temp=--`

The list of temperatures requested for the GROUPR run must agree with the first `ntemp` temperatures on the PENDF tape. Check your BROADR and UNRESR or PURR runs.

`message from getunr--Warning, negative URR cross sections found`

NJOY found negative cross sections for the unresolved energy range from UNRESR.

`error in getff***do not know how to handle mf,mt ...`

The `getff` routine branches to different blocks of coding for different combinations of `mfd` and `mtd`, but if no appropriate branch is found for the current values, this error message is issued. It probably indicates an error in the evaluation.

`error in getmf6***desired particle not found.`

The outgoing particle for a group-to-group matrix is implied by the value of `mfd` (for example, protons for `mfd`=21). This message means that the section of File 6 requested with `mfd` and `mtd` does not contain a subsection that produces that particle. Check the user input. This message should not occur with automatic reaction selection.

`error in getmf6***illegal law.`

The value of the `law` parameter is greater than 7. This implies an error in the evaluation.

`error in getmf6***too many subsection energy points.`

Limited by the parameter `maxss`=500.

`error in getmf6***storage exceeded.`

See the allocatable array `temp` with length `ntmp`=990000.

`message from getmf6--bad grids for corresponding point ...`

Corresponding-point interpolation won't work correctly unless the two grids above and below the point of interest have the same number of points. This message means that there is an error in the evaluation.

`error in getmf6***too many subsections for one particle.`

We currently allow for no more than three. See `iyss(3)`, `izss(3)`, and `jss(3)`.

`message from getmf6--there are multiple subsections in mf6`

This warning message is issued when a specific particle has multiple subsections in an mf6 section.

`error in cm2lab***storage exceeded.`

This means that the allocatable array `tmp` with length `ntmp`=990000 in subroutine `getmf6` has run out of space.

`message from cm2lab--lab normalization problem mt=... e=...`

This warning message is issued when the integral after normalisation to the lab system is different from 1 by more than 1%.

`error in f6ddx***illegal lang.`

The value of `lang` for tabulated angular distributions must be in the range 11-15.

`message from cm2lab--vertical segment(s) in distribution`

There appears to be a jump in the energy dependent multiplicity for the outgoing particle at a given incident energy. This message is only given when Kalbach-Mann systematics are used. This is most likely an evaluation problem.

`error in bach***dominant isotope not known.`

The calculation of the neutron separation energy needed for the Kalbach model for particle energy-angle distributions needs a value for the dominant isotope in an element. It will have to be added to the code. The same problem will occur with the parallel routines in HEATR and ACER.

`error in ll2lab***storage exceeded.`

This means that the allocatable array `tmp` with length `ntmp`=990000 in subroutine `getmf6` has run out of space.

`error in f6cm***nl>mxlg`

The current limit is `mxlg`=65.

`error in f6ddx***nl>mxlg`

The current limit is `mxlg`=65.

`error in f6lab***illegal lang.`

The value of `lang` must be equal to 1 or be in the range 11–15 for tabulated angular distributions.

`error in f6dis***illegal lang.`

The value of `lang` must be equal to 1 or 2, or be in the range 11–15 for tabulated angular distributions.

`error in getdis***illegal nqp`

The allowed quadrature orders are 4, 8, 12, and 20.

`error in getfle***desired energy above highest energy given`

Should not occur for well-constructed ENDF files. Check the evaluation to be sure File 3 and File 4 are consistent.

`message from getfle--lab distribution changed to cm for mt=...`

Angular distributions for two-body reactions are supposed to be given in the CM frame by ENDF conventions. Some old evaluations for heavy materials violate this rule; changing to the CM frame has little effect on the answers.

`error in getaed***thermal mf6/law7 not coded`

This subroutine can't handle the E-μ-E' ordering option provided by THERMR.

`error in getaed***storage exceeded.`

This refers to the allocatable array `aes` with length `maxaes`=200000.

`error in getgfl***too many gammas.`

See the parameter `maxgfl`=500.

`error in getgfl***storage exceeded.`

See the parameter `ntmp`=10000.

`error in getgfl***desired energy at highest given energy.`

This problem should not occur in a well-constructed ENDF file. Check Files 3, 12, 13, and 14 for consistency.

`error in getco***limited to 64 legendre coefficients.`

See `nlmax`=65 and `P(65)`.

`error in getco***lab to cm conversion not coded.`

The need for this type of conversion rarely occurs on the current ENDF evaluations, because CM is consistently used for two-body reactions, and the laboratory frame is consistently used for continuum reactions. There are a few exceptions for the heavy isotopes, where CM and lab are essentially equivalent, but they were errors when the files were generated.

`error in getgyl***lo=2 not coded.`

This message should not occur, because any transition probability arrays on the ENDF tape should have been converted to yields by `conver`.

`error in getgyl***too many gammas.`

The current limit is 550 photons. See the parameter `nylmax`=550.

`error in getgyl***storage exceeded.`

This refers to the allocatable array `tmp` with length `ntmp`=15000.

`message from conver--cannot do complete particle production...`

With the advent of the ENDF-6 format, it is possible to make evaluations that fully describe all the products of a nuclear reaction. Some carry-over evaluations from earlier ENDF/B versions also have this capability, but many do not. This message is intended to goad evaluators to improve things!

`message from conver--gamma production patch made for ...`

This patch is used to correct the old ENDF/B-III evaluations for MAT=1149 and MAT=1150 (chlorine and potassium).

`message from conver--mf12, mt ... may be missing`

This message indicates that the discrete photon data in mf12 for this mt number may be missing or be incomplete.

`error in conver***nnth too large`

See `mxnnth`=350.

`message from conver--skipping new mf6/mt18 multiplicity section`

Starting with ENDF/B-VIII.0, fission neutrons and photons can now be described using probability functions for emitting 0, 1, 2, ... particles per fission. This data can currently not be included in any multigroup or continuous energy format so the data is simply skipped.

`error in conver***too many lo=2 gammas.`

The LO=2 processing uses a set of allocatable arrays that are sized using `lmax`=500. That number can be increased freely, if necessary.

`error in conver***storage for fission nu exceeded.`

The storage in the allocatable array `nu` is sized using `nnu`=6000, which can be increased freely.

`error in conver***xxxx to big.`

The automatic processing of reactions is controlled by lists stored in global arrays like `mf4` or `mf6p`. This error occurs when one of the particular indexes, indicated by "xxx" exceeds `maxr1`=500. The arrays for MF=10 (see `imf10`) have a limit of `maxr2`=500.

`error in getsed***too many subsections.`

The current limit is 20. See the parameter `nkmax`=20.

`error in getsed***storage tmp exceeded.`

The input ENDF data are stored in the allocatable array `tmp` with length `ntmp`=50000. An integer will be included in this message to indicate where exactly in the source code this message was issued.

`message from getsed--corresponding point interpolation ...`

The interpolation schemes corresponding to the range 11–15 are not supported.

`message from getsed--upscatter correction....`

This reaction should not have upscatter. The error is placed into the in-group element.

`error in anased***illegal lf.`

Legal values are 5, 7, 9, 11, and 12.

`error in f6psp***3, 4, or 5 particles only.`

Phase-space formulas for 3, 4, or 5 particles are provided in this routine. Check for an error in the evaluation.