Argonne NATIONAL LABORATORY

ANL/NE-11/41 Rev.3

MC^2-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis

Nuclear Science and Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

ANL/NE-11/41 Rev.3

MC^2-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis

Changho Lee, Yeon Sang Jung, and Won Sik Yang
Nuclear Science and Engineering Division
Argonne National Laboratory

August 31, 2018

SUMMARY

The MC^2-3 code is a Multigroup Cross section generation Code for fast reactor analysis, developed by improving the resonance self-shielding and spectrum calculation methods of MC^2-2 and integrating the one-dimensional cell calculation capabilities of SDX. The code solves the consistent P_1 multigroup transport equation using basic neutron data from ENDF/B data files to determine the fundamental mode spectra for use in generating multigroup neutron cross sections. A homogeneous medium or a heterogeneous slab or cylindrical unit cell problem is solved in ultrafine (~2000) or hyperfine (~400,000) group levels. In the resolved resonance range, pointwise cross sections are reconstructed with Doppler broadening at specified isotopic temperatures. The pointwise cross sections are directly used in the hyperfine group calculation whereas for the ultrafine group calculation, self-shielded cross sections are prepared by numerical integration of the pointwise cross sections based upon the narrow resonance approximation. For both the hyperfine and ultrafine group calculations, unresolved resonances are self-shielded using the analytic resonance integral method. The ultrafine group calculation can also be performed for two-dimensional whole-core problems to generate region-dependent broad-group cross sections. Multigroup cross sections are written in the ISOTXS format for a user-specified group structure. The code is executable on UNIX, Linux, and PC Windows systems, and its library includes all isotopes of the ENDF/B-VII.0 data.

REVISION HISTORY

Revision	Description
0	Release date: September 30, 2011 Associated code version: 3.0.0 Initial release
1	Release date: January 31, 2012 Updates: - Added sample outputs (Section 6) - Basically made corrections of typos and clarified some descriptions
2	Release date: November 8, 2013 Associated code version: 3.1.2 Updates: - Added the gamma library description (Section 2.12) - Added user inputs and sample inputs (Sections 5.6, 5.10, and 5.11) for gamma library - Added Appendix E for gamma library group structure
3	Release date: August 31, 2018 Associated code version: 3.2.3 Updates: - Added the 2D/3D MOC capability description (Section 2.8) - Added user inputs (Section 4.4) and sample inputs (Section 5.12) for the MOC capability - Updated the gamma calculation description (Section 2.13) - Added the description of the ENDF/B-VII.0 gamma library (Appendix D)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

A fast reactor simulation program was initiated in 2007 as part of the Advanced Fuel Cycle Initiative (AFCI) to develop a suite of modern simulation tools for the analysis and design of fast reactors. The goal of the program is to reduce the uncertainties and biases in various areas of reactor design activities by providing enhanced prediction capabilities. Within this program, an integrated, advanced neutronics code is being developed that allows the high fidelity description of a nuclear reactor and simplifies the multi-step design process by direct coupling with thermal-hydraulics and structural mechanics calculations. This code solves the multigroup Boltzmann transport equation using three different deterministic methods (i.e., spherical harmonics, discrete ordinate, and characteristic methods), and thus requires accurate multigroup cross sections.

The ultimate goal for the cross section preparation is to generate cross sections inline within the whole-core transport code through the direct use of fine-group cross section libraries so that the approximate boundary conditions used in the calculations for cross section generation can be eliminated. Realizing that even with the current state-of-the-art computer technology it is not yet realistic to perform explicit geometry transport calculations with sufficient energy resolution to model the complicated resonance cross sections accurately, we decided to develop a modernized cross section generation code by simplifying the existing multi-step schemes and improving the computational methods and models.

As an initial step, the legacy Multigroup Cross section generation Code MC^2-2 [1], which has been extensively used for more than three decades and well verified and validated with the ENDF/B-V data, was modified to process the latest ENDF/B-VII.0 data [2]. Various improvements were then made to the resonance self-shielding and slowing-down calculation methods to enhance the solution accuracy. The improved spectrum calculation methods were integrated with the one-dimensional cell calculation capabilities of the SDX code [3]. The integrated spectrum and unit cell calculation capabilities were implemented into a new code named MC^2-3 with modern programming practices and memory structure. Extensive and systematic verification and validation tests have been performed for MC^2-3 using various benchmark problems. [4]

The MC^2-3 code solves the consistent P_1 multigroup transport equation using basic neutron data from ENDF/B data files to determine the fundamental mode spectra for use in generating multigroup neutron cross sections. A homogeneous medium, heterogeneous slab or cylindrical unit cell problem can be solved in ultrafine (~2000) or hyperfine (~400,000) group levels. In the resolved resonance range, pointwise cross sections are reconstructed with Doppler broadening at specified isotopic temperatures. The pointwise cross sections are directly used in the hyperfine group calculation whereas for the ultrafine group calculation, self-shielded cross sections are prepared by numerical integration of the pointwise cross sections. For both the hyperfine and ultrafine group calculations, unresolved resonances are self-shielded using the analytic resonance integral method. The ultrafine group calculation can also be performed for two-dimensional whole-core problems to take into account realistic leakage when generating region-dependent broad-group cross sections.

Compared with the MC^2-2 code, MC^2-3 has several new features. First, it has a one-dimensional unit cell calculation capability, whereas MC^2-2 relied upon SDX calculations to take into account the local heterogeneity effects. For the ultrafine group calculation, self-

shielded cross sections are prepared by numerical integration of the pointwise cross sections based upon the narrow resonance approximation. This removes the tailing effects of the analytic resonance integral method used in MC^2-2, and thus simplifies the procedure to prepare the MC^2 libraries using the ETOE-2 code [3] by eliminating the need for screening out wide resonances. The two-dimensional, whole-core transport calculation capability is another important feature for the region-dependent cross section generation, which was previously done in separate calculations following MC^2-2/SDX calculations. The main differences in the computational methods between MC^2-2 and MC^2-3 are illustrated in Figure 1.1, and the main improvements and capabilities of MC^2-3 are summarized below.

- Self-shielding of resolved resonances by the numerical integration of pointwise (PW) cross sections instead of the analytic resonance integral method used in MC^2-2; The reconstructed isotopic PW cross sections at user-specified temperatures are stored in a binary form for reuse.
- Self-shielding of high-order total cross sections based on the narrow resonance (NR) and B_0 approximation; In MC^2-2, only P_0 cross sections were self-shielded.
- Use of the multigroup equation for the entire energy range; The continuous slowing-down method used in the resolved resonance energy range has been replaced with the multigroup method in order to minimize the approximations involved in the slowing-down equation by light elements (especially hydrogen).

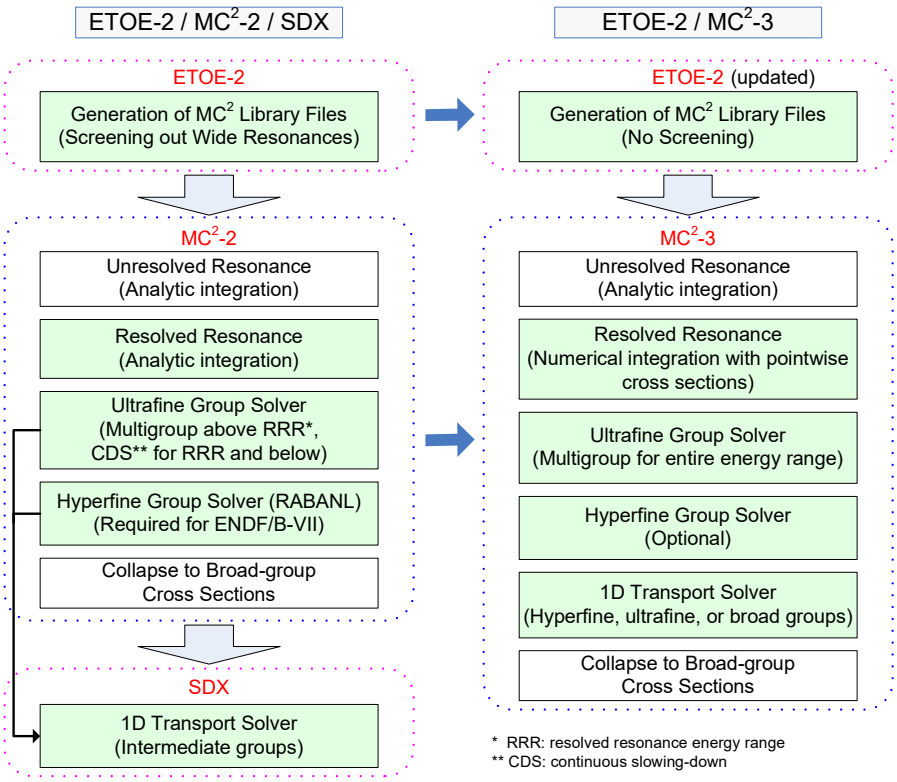


Figure 1.1 Differences in Methodologies between MC^2-2 and MC^2-3

- Introduction of anisotropic inelastic scattering (currently only P_1) which are prepared by NJOY [5] on the ultrafine group basis,
- Use of a fission spectrum matrix as a function of incident neutron energy; In MC^2-2, the inelastic scattering was assumed to be isotropic.
- Provisional use of PENDF [5] files generated by NJOY at a few different temperatures; PW cross sections are reconstructed and interpolated using the temperature-dependent PENDF files.
- Self-shielding of the resonance-like cross sections of intermediate-weight isotopes such as chromium, nickel, iron, and sodium above the resonance energy range.
- Fix-up of negative P_1 spectrum (current spectrum) for the consistent P_1 formulation by the use of the extended transport approximation.
- Self-shielding of unresolved resonances at finer effective energy grids.
- Incorporation of the one-dimensional transport capabilities of SDX; One-dimensional transport calculations are extended to the ultrafine or hyperfine group calculations, whereas they were previously done in separate SDX calculations in a fine (~230) group level.
- Improved estimation of the effective background cross sections, which are used to account for the geometry effects on resonance self-shielding, by the use of the region-to-region transmission probabilities obtained from the collision probability method (CPM).
- Incorporation of an anisotropic scattering source for the hyperfine group transport calculation; In MC^2-2, the hyperfine group calculations were performed with the assumption of an isotropic source and thus limited to the low energy resolved resonance range.
- An efficient solution scheme for hyperfine group calculations achieved by utilizing the ultrafine-group-based fission and inelastic scattering sources.
- Incorporation of the two-dimensional ultrafine group transport capabilities (currently provided by TWODANT [6]) to take into account realistic leakage when generating region-dependent cross sections.
- Computation and storage of leakage spectrum which can be used as an external source to the successive MC^2-3 calculations for the neighboring, sub-productive composition.
- Extension of the maximum energy from the previous value of 14.2 MeV to 20 MeV or higher (almost completed).
- Flexible management of output ISOTXS files [1] such as writing different cross sections (e.g., regionwise vs. homogenized or microscopic vs. macroscopic) on the output ISOTXS file, merging multiple ISOTXS files, or converting the ISOTXS file between ASCII and binary formats.
- User-friendly keyword-based (namelist) input system.
- Better programming practices and the use of data structures based on FORTRAN 95.

Recently, the MOC capability was implemented to accurately produce the self-shielded broad-group cross sections accounting for the multi-dimensional geometry effect with no

geometrical approximations as well as possibly to directly perform a whole core transport calculation. The MOC implemented in MC^2-3 is based on the methodology which combines the radial 2D MOC with the discontinuous Galerkin finite element method in the axial direction. The 2D/3D MOC methodology called was originally developed and implemented in PROTEUS [27], demonstrating a good performance even for problems including optically thin regions. Currently, the MOC capability implemented in MC^2-3 is able to deal with problems with 2D and 3D hexagonal geometry and 2D rectangular geometries. In addition, the coarse mesh finite difference (CMFD) acceleration technique was implemented to the MOC solver for 2D and 3D hexagonal geometries to improve the solution convergence. The parallel processing with OpenMP was applied to reduce the computation time as well.

Extensive and systematic verification and validation tests have been performed for MC^2-3 using various benchmark problems, including more than 30 critical benchmark problems derived from experiments. The verification test results have confirmed that the improved capabilities are functioning properly as intended. The analyses of the critical benchmark problems have also shown that the deterministic transport solutions obtained with the multi-group cross sections prepared with MC^2-3 agree well with experiments as well as Monte Carlo solutions. The results of the verification and validation tests will be discussed in a separate document along with the code performance data.

The document contains the method, user, and programmer manuals for convenience. Section 2 discusses the methodologies employed in the code. Section 3 describes the MC^2 library generation for ENDF/B-VII.0 data. Section 4 presents the code input description, followed by sample inputs and outputs for cross section generation for typical fast reactor problems in Sections 5 and 6. The program structure is briefly explained in Section 7.

2 Method

Developed at Argonne National Laboratory (ANL), MC^2-3 is a computer program for solving the neutron transport equation in homogeneous mixtures or one-dimensional geometries to determine a detailed spectrum for use in deriving multigroup cross sections for fast reactor calculations. The code is composed of eight major functional modules: input processor, unresolved resonance self-shielding, resolved resonance self-shielding, scattering matrix calculation, ultrafine group spectrum calculation, group collapsing, cross section file generation in the ISOTXS format, and optionally hyperfine group integral transport spectrum calculation. Figure 2.1 illustrates the computational flow through these modules. Different computational paths are employed, depending upon the problem geometry and the number of energy groups. The main data library required for MC^2-3 is structured in eight distinct data files which are created by ETOE-2 [3] in the file formats provided in Reference [1]. The code also needs libraries for fission spectrum matrix data and inelastic scattering data. The PENDF files are called optionally, and the pointwise cross section files are reused for computational efficiency if available. The details of each component will be discussed in the following sections.

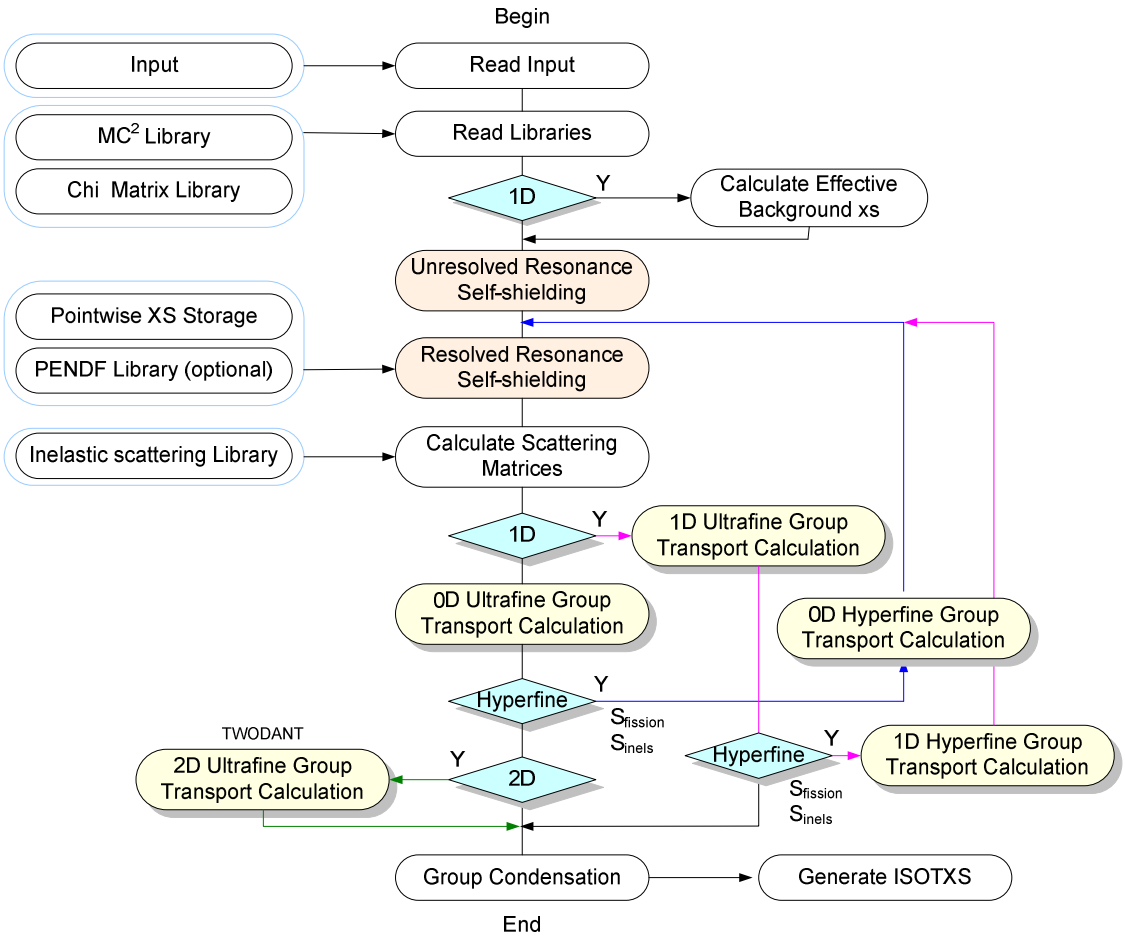


Figure 2.1 Program Flow Overview of MC^2-3

2.1 Multigroup Transport Calculation

The steady-state neutron transport equation for an isotropic fission and external source is given by

$$\nabla\cdot\Omega\psi(r,E,\Omega)+\Sigma_t(r,E)\psi(r,E,\Omega)=\iint dE'd\Omega'\psi(r,E',\Omega')\Sigma_s(r,E'\to E,\Omega\cdot\Omega') + \frac{1}{4\pi}S(r,E), \tag{2.1}$$

where $\psi(r,E,\Omega)$ is the neutron flux at location r, energy E, and angle Ω, $\Sigma_t(r,E)$ is the total cross section, $\Sigma_s(r,E'\to E,\Omega\cdot\Omega')$ is the scattering cross section from E' to E with the cosine of scattering angle $\Omega\cdot\Omega'$, and $S(r,E)$ is the isotropic fission and external neutron source.

Assuming a homogeneous mixture and making use of the fundamental mode ansatz, $\psi(r,E,\Omega)=\psi(E,\Omega)e^{iB\cdot r}$ and $S(r,E)=S(E)e^{iB\cdot r}$, Eq. (2.1) can be written as

$$\left[iB\cdot\Omega+\Sigma_t(E)\right]\psi(E,\Omega)=\iint dE'd\Omega'\psi(E',\Omega')\Sigma_s(E'\to E,\mu_s)+\frac{S(E)}{4\pi}, \tag{2.2}$$

where B^2 is the buckling, and $\mu_s=\Omega\cdot\Omega'$ is the cosine for the scattering angle in the laboratory system. For simplicity, consider the symmetry with respect to the azimuthal angle as in a one-dimensional plane geometry problem. Then the spherical harmonics expansion of angular flux and scattering transfer cross section are reduced to

$$\psi(u,\mu)=\sum_{\ell=0}^{\infty}\frac{2\ell+1}{2}\phi_\ell(u)P_\ell(\mu), \tag{2.3a}$$

$$\Sigma_s(u'\to u,\mu_s)=\sum_{\ell=0}^{\infty}\frac{2\ell+1}{2}\Sigma_{s\ell}(u'\to u)P_\ell(\mu), \tag{2.3b}$$

where $P_\ell(\mu)$ is the ℓ-th order Legendre polynomial and u is the lethargy defined by $\ln(E_0/E)$. Substituting Eq. (2.3) into Eq. (2.2) and making use of the addition theorem for spherical harmonics yields

$$\left[iB\mu+\Sigma_t(u)\right]\psi(u,\mu)=\sum_{\ell=0}^{\infty}\frac{2\ell+1}{2}P_\ell(\mu)\int du'\Sigma_{s\ell}(u'\to u)\phi_\ell(u')+\frac{S(u)}{2}\ . \tag{2.4}$$

By multiplying Eq. (2.4) by $P_n(\mu)$, integrating over μ, and using the recursion relation of the Legendre polynomials, we obtain

$$\frac{n+1}{2n+1}iB\phi_{n+1}(u)+\frac{n}{2n+1}iB\phi_{n-1}(u)+\Sigma_t(u)\phi_n(u) = \int du'\Sigma_{sn}(u'\to u)\phi_n(u')+S(u)\delta_n^0,\quad n=0,1,2,\cdots, \tag{2.5}$$

where $\phi_{-1}(u)=0$. In the P_N approximation, it is assumed that $\phi_{N+1}=0$ and thus Eq. (2.5) provides a set of $N+1$ coupled equations as

$$\frac{n+1}{2n+1}iB\phi_{n+1}(u)+\frac{n}{2n+1}iB\phi_{n-1}(u)+\Sigma_t(u)\phi_n(u) = \int du'\Sigma_{sn}(u'\to u)\phi_n(u')+S(u)\delta_n^0, \quad n=0,1,2,\cdots,N-1, \tag{2.6a}$$

$$\frac{N}{2N+1}iB\phi_{N-1}(u)+\Sigma_t(u)\phi_N(u)=\int du'\Sigma_{sN}(u'\to u)\phi_N(u'). \tag{2.6b}$$

The extended transport approximation assumes that anisotropic scattering for $n>1$ takes place without a change in energy such that

$$\int du'\Sigma_{sn}(u'\to u)\psi_n(u')=\Sigma_{sn}(u)\psi_n(u), \quad n=2,3,\cdots,N. \tag{2.7}$$

In other words, it is assumed that the number of neutrons scattered in the lethargy interval du around u is the same as the number of neutrons scattered out from this interval.

Using Eq. (2.7) in Eq. (2.6), the consistent P_1 order N extended transport equations are obtained as

$$iB\phi_1(u)+\Sigma_t(u)\phi_0(u)=\int du'\Sigma_{s0}(u'\to u)\phi_0(u')+S(u), \tag{2.8a}$$

$$\frac{2iB}{3}\phi_1(u)+\frac{iB}{3}\phi_0(u)+\Sigma_t(u)\phi_1(u)=\int du'\Sigma_{s1}(u'\to u)\phi_1(u'), \tag{2.8b}$$

$$\frac{n+1}{2n+1}iB\phi_{n+1}(u)+\frac{n}{2n+1}iB\phi_{n-1}(u)+[\Sigma_t(u)-\Sigma_{sn}(u)]\phi_n(u)=0, \quad n=2,3,\cdots,N-1, \tag{2.8c}$$

$$\frac{N}{2N+1}iB\phi_{N-1}(u)+[\Sigma_t(u)-\Sigma_{sN}(u)]\phi_N(u)=0. \tag{2.8d}$$

The last two equations in Eq. (2.8) can be solved successively from $n=N$ as

$$\phi_n(u)=-\frac{n}{2n+1}\frac{iB}{A_n(B,u,N)}\phi_{n-1}(u), \quad n=2,3,\ldots,N, \tag{2.9}$$

where $A_n(B,u,N)=b_{n-1}+\dfrac{a_n}{A_{n+1}(B,u,N)}=b_{n-1}+\dfrac{a_n}{b_n+\dfrac{a_{n+1}}{b_{n+1}+\dfrac{a_{n+2}}{\cdots+\dfrac{a_{N-1}}{b_{N-1}}}}}$,

$$a_n=\frac{n+1}{2n+1}\frac{n+1}{2(n+1)+1}B^2,$$

$$b_n=\Sigma_t(u)-\Sigma_s^{n+1}(u).$$

Substituting Eq. (2.9) for $n=2$ into Eq. (2.8b) yields

$$\frac{iB}{3}\phi_0(u)+A_1(B,u,N)\phi_1(u)=\int du'\Sigma_{s1}(u'\to u)\phi_1(u'), \tag{2.10}$$

where $b_0=\Sigma_t(u)$.

By integrating Eq. (2.8) and Eq. (2.10) over group intervals and using appropriately defined group constants, the multigroup form of the extended P_1 transport equation can be written as

$$iB\phi_1^g+\Sigma_{t0}^g\phi_0^g=\sum_{g'}[\Sigma_{s0}^{gg'}+\Sigma_{in,0}^{gg'}+2\Sigma_{n2n}^{gg'}]\phi_0^{g'}+S^g, \tag{2.11a}$$

$$\frac{iB}{3}\phi_0^g+A_1^g\phi_1^g=\sum_{g'}[\Sigma_{s1}^{gg'}+\Sigma_{in,1}^{gg'}]\phi_1^{g'}, \tag{2.11b}$$

$$\phi_l^g=-\frac{l}{(2l+1)A_l^g}iB\phi_{l-1}^g,\quad l=2,\ldots,\mathrm{N}, \tag{2.11c}$$

where $\varphi_l^g=\int_{u_{g-1}}^{u_g}\varphi_l(u)\,du$,

$$\Sigma_{tl}^g=\frac{1}{\phi_l^g}\int_{u_{g-1}}^{u_g}\Sigma_t(u)\phi_l(u)\,du,$$

$$\Sigma_{sl}^{gg'}=\frac{1}{\phi_l^{g'}}\int_{u_{g-1}}^{u_g}du\int_{u_{g'-1}}^{u_{g'}}du'\Sigma_{sl}(u'\to u)\phi_l(u'),$$

$$A_l^g=\int_{u_{g-1}}^{u_g}A_l(B,u,N)\,du,$$

$$S^g=\frac{1}{k}S_f^g+S_{ex}^g.$$

Here $\Sigma_{sl}^{gg'}$ and $\Sigma_{in,l}^{gg'}$ are respectively the elastic and inelastic scattering transfer cross sections of order l from a source group g' to a sink group g, and $\Sigma_{n2n}^{gg'}$ is the (n,2n) transfer cross section. Note that in MC^2-3 the inelastic scattering is allowed to be anisotropic whereas it was assumed to be isotropic in MC^2-2. However, the (n,2n) source is still assumed isotropic. The source term is divided into the fission source S_f^g and the external source S_{ex}^g. For an inhomogeneous source problem where $S_{ex}^g\neq 0$, the multiplication factor k is set to unity. If there is no external source and the mixture includes fissionable materials, then the eigenvalue problem is solved to determine k for a given buckling B^2.

If N is the scattering order in the center-of-mass system given in the ENDF/B data, the elastic scattering transfer cross section of order l from a source group g' to a sink group g in the laboratory system is given by

$$\sigma_{sl}^{gg'}=\frac{1}{\phi_l^{g'}}\int_{u_{g-1}}^{u_g^*}du\int_{u_{g'-1}^*}^{u_{g'}}du'\frac{\sigma_s(u')}{1-\alpha}P_l[\mu_s(u',u)]e^{-(u-u')}\phi_l(u')\sum_{n=0}^{N}(2n+1)f_n(u')P_n[\mu_c(u',u)], \tag{2.12}$$

where $P_l(\mu) = l$-th order Legendre polynomial,

$f_n(u') = n$-th order Legendre expansion coefficient of scattering cross section in the center-of-mass system,

$\alpha = \left[(A-1)/(A+1)\right]^2$, A = atomic mass,

μ_c = cosine of scattering angle in the center-of-mass system,

μ_s = cosine of scattering angle in the laboratory system.

For the hyperfine group calculation, the group structure is constructed in such a way that the cross sections and neutron spectrum within a hyperfine group can be approximated to be constant. Thus, Eq. (2.12) can be reduced to

$$\sigma_{sl}^{gg'} = \frac{\sigma_s^g}{(1-\alpha)\Delta u}\int_{u_{g'-1}^*}^{u_{g'}} du' P_l(u' \to g), \tag{2.13}$$

where

$$P_l(u' \to g) = \sum_{n=0}^{N}(2n+1)\int_{u_{g-1}}^{u_g^*} du \int_{u_{g'-1}^*}^{u_{g'}} du' f_n(u') P_n[\mu_c(u',u)] P_l[\mu_s(u',u)] e^{-(u-u')} . \tag{2.14}$$

When properly self-shielded resonance cross sections are used, the scattering cross sections and neutron spectrum can be assumed to be constant within an ultrafine group. Under this assumption, the ultrafine group scattering transfer matrices are evaluated in three different ways, depending upon the mass of the scattering nuclide: light material, heavy material, and hydrogen. The double integral in Eq. (2.14) is numerically evaluated by dividing a ultrafine group into multiple hyperfine groups. Details are presented in Section 2.5.

The ultrafine group structure is constructed by partitioning the lethargy (energy) domain into equal lethargy intervals of width Δu. The lethargy group width is selected small enough to account for the fine spectrum effects attributed to the resonances of intermediate atomic weight nuclides. The ultrafine-group lethargy interval is set to 1/120, corresponding to the MC^2 structure, giving 2082 groups from 0.414 eV to 14.2 MeV and 2123 groups from 0.414 eV to 20 MeV.

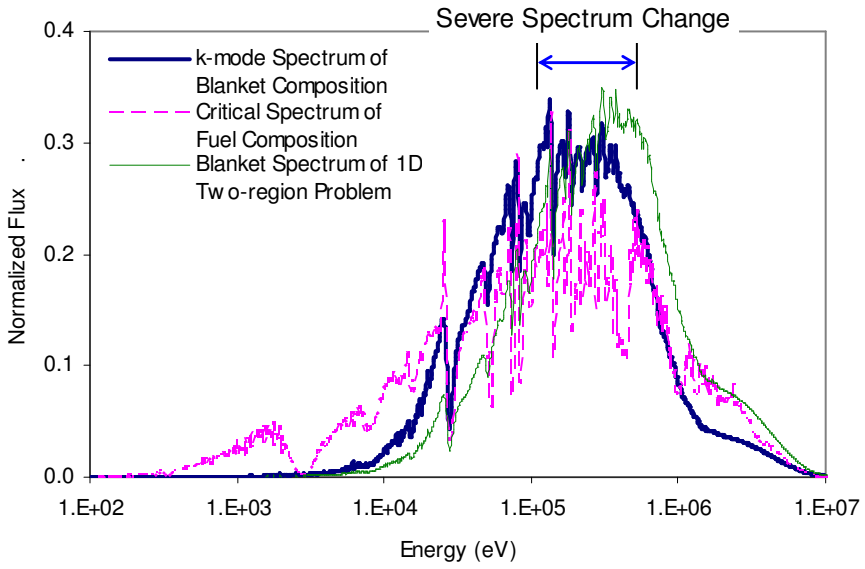
The neutron spectrum is determined by solving the multigroup equation for the entire energy range. The continuous slowing-down method used in MC^2-2 for the resolved resonance energy range has been replaced with the multigroup method in order to minimize the approximations involved in the slowing-down by light elements (especially hydrogen).

The k eigenvalue problem for a homogeneous medium is solved with a user specified buckling or a default value of 10^{-10}, which is also used as an initial guess for the critical buckling search problem. The eigenvalue problem is solved with the normal power iteration until both the ultrafine group fission source and eigenvalue are converged within a certain criteria, normally 10^{-5}. For the fundamental mode spectrum calculation, the critical buckling is iteratively determined until the multiplication factor k converges to unity. A new estimate for the critical buckling is determined by the linear interpolation of two latest buckling values.

The search procedure ends if $|k_{eff} - 1| \leq \varepsilon$ where ε is a default or user-specified convergence criterion.

The critical buckling search is possible even for a sub-productive composition, yielding a negative buckling. However, it is not permitted for a highly sub-productive composition (i.e., a blanket composition of $k < 0.5$) to prevent the flux from being negative. The fundamental mode spectrum obtained from the critical buckling search generally provides a better weighting spectrum for multigroup cross section generation, in particular when the number of broad groups is relatively small and the sub-productivity is large. However, the difference between the resulting broad group cross sections obtained with the fundamental and the k-mode eigen-spectrum diminishes when a relative large number of broad groups is used (typically more than a few tens of groups).

In case that there is no fissionable material in the mixture of interest, an inhomogeneous problem is solved using the given external source. The external source needs to be specified by the user. If not, the fission spectrum of U-238 is used as the default source spectrum. The external source is often used to approximately represent the net current from the core to the blanket and/or reflector. The leakage spectrum from the core is generally obtained using the critical spectrum of a homogeneous medium problem of fuel composition. However it is often noticed that the leakage spectrum obtained from the homogeneous medium problem of fuel composition is very different from that in the fuel region of the actual multi-dimensional configuration. As an example, Figure 2.2 compares the fundamental mode spectrum of a fuel composition, the k-mode eigen-spectrum of a blanket composition, and the blanket region spectrum of the one-dimensional two-region problem composed of the fuel and blanket compositions. As can be seen, the spectrum of the homogeneous problem of the blanket composition and that in the blanket region of the two-region problem show noticeable differences in the energy range of a few hundreds keV. The leakage spectrum obtained from the homogeneous problem of the fuel composition is also significantly different from that of the fuel region of the two-region problem, as shown in Figure 2.3. As expected, the neutron leakage from the fuel region of the two-region problem is negative around the large resonances of intermediate mass isotopes of which concentrations are much larger in the fuel region than in the blanket region. This example shows that the leakage spectrum obtained from the homogeneous problem of fuel composition can be a very poor approximation to the leakage spectrum in the actual core configuration. Therefore, to model the region-to-region leakage in actual core configurations more accurately, two-dimensional transport capabilities have been incorporated in MC^2-3. This is discussed in more detail in Section 2.11.



(Fuel k_∞ = 1.51, Blanket k_∞ = 0.33)

Figure 2.2 Neutron Spectrum Difference between Homogeneous and One-Dimensional Problems

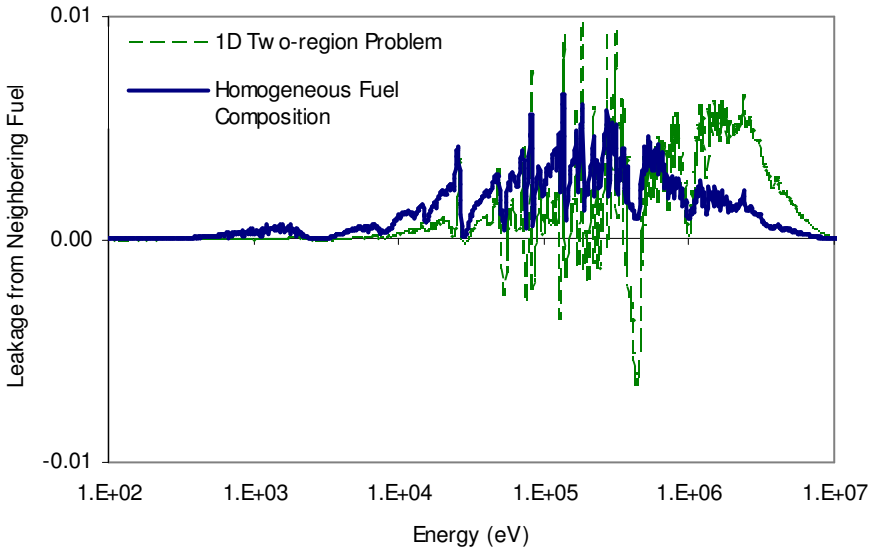


Figure 2.3 Neutron Leakage Spectrum from Neighboring Fuel Region

2.2 Resolved Resonance

2.2.1 Reconstruction of Pointwise Cross Sections

There are four resonance formalisms that are currently used in the ENDF/B format: single-level Breit-Wigner (SLBW), multi-level Breit-Wigner (MLBW), Adler-Adler (AA), and Reich-Moore (RM). The first three formalisms were extensively used in the previous ENDF/B data, while the last one has been used for many major nuclides in the latest ENDF/B-VII.0 data. [7] For each resolved resonance given in one of these formalisms, MC^2-3 reconstructs the corresponding pointwise values at a specified temperature. The RM resonance data are converted into the multi-pole formalism [8] that preserves the general features required by the traditional resonance integral concept and the Doppler-broadening algorithm in the MC^2-2 code without compromising rigor. [9]

The SLBW approach was heavily used in earlier versions of the ENDF/B data because of its easy use of analytical formulation, but it is hardly used these days except for the unresolved resonance representation because it neglects multi-level and multi-channel effects and can even produce unphysical negative cross sections for elastic scattering. Therefore, the SLBW approach becomes inadequate especially when resonances are close together such as in fissile isotopes. The SLBW resonances are represented as

$$\sigma_x = \sum_{l,J}\sum_{k}\frac{\sigma_{0xk}}{1+x_k^2} = 4\pi\lambda\!\!\!^{-2}\sum_{l,J} g_J \sum_{k}\left(\frac{\Gamma_{nk}\Gamma_{xk}}{2\Gamma_{tk}}\,\mathrm{Re}\left\{\frac{-i}{z_k - E}\right\}\right) \quad ; \quad x \in \gamma, f\,, \tag{2.15a}$$

$$\sigma_{tR} = \sum_{l,J}\sum_{k}\sigma_{0tk}\left(\frac{\cos 2\phi_l}{1+x_k^2} + \frac{x_k \sin 2\phi_l}{1+x_k^2}\right) = 4\pi\lambda\!\!\!^{-2}\sum_{l,J} g_J \sum_{k}\left(\frac{\Gamma_{nk}}{2}\,\mathrm{Re}\left\{\frac{-ie^{-i2\phi_l}}{z_k - E}\right\}\right), \tag{2.15b}$$

where Γ_α = partial width of reaction type α, Γ_n = neutron width, $\lambda\!\!\!^{-}$ = reduced neutron wave length, l = angular momentum state, J = compound nucleus spin, I = target nucleus spin, Δ_k = level shift, ϕ_l = hard-sphere phase shift, $g_J = (2J+1)/[2(2I+1)]$, $\sigma_{0xk} = 4\pi\lambda\!\!\!^{-2} g_k\left(\Gamma_{nk}\Gamma_{xk}/\Gamma_{tk}^2\right)$, $x_k = \left(E - E_k - \Delta_k\right)/\left(\Gamma_{tk}/2\right)$, and $z_k = E_k + \Delta_k - i\Gamma_{tk}/2$.

The equivalent representations of the traditional expressions in the complex domain are introduced here to signify the meromorphic nature of cross sections pertinent to discussions to follow. The quantities in parentheses in Eq. (2.15) that characterize the energy dependence of a given resonance are referred to as the Lorentzians. They are readily amenable to analytical Doppler-broadening.

Having a resonance-resonance interference term unlike the SLBW, the MLBW representation is used for some material in the ENDF/B-VII.0 data because it is positive definite. However, it is disadvantageous in that it does not handle material with multi-channel effects and does not work well for analytic reactor physics methods, either. The MLBW cross sections are represented as

$$\sigma_x = \frac{1}{E}\sum_{l,J}\sum_{\mu}\left(\mathrm{Re}\left\{\frac{-iR_{J,l,\mu}^{(x)}}{z_\mu - E}\right\}\right) \quad ; \quad x \in \gamma, f\,, \tag{2.16a}$$

$$\sigma_{tR} = \frac{1}{E}\sum_{l,J}\sum_{\mu}\left(\mathrm{Re}\left\{ \frac{-ie^{-i2\phi_l} R^{(t)}_{J,l,\mu}}{z_\mu - E} \right\} \right), \tag{2.16b}$$

where the pole z_μ is clearly identifiable with that defined in Eq. (2.15) for SLBW resonances and $R^{(x)}_{J,l,\mu}$ is related to the corresponding residue $c^{(x)}_{J,l,\mu}$ as

$$R^{(x)}_{l,J,\mu} = \left(4\pi\lambda\!\!\!^{-2} g_J E\right) c^{(x)}_{l,J,\mu}. \tag{2.17}$$

The residue $c^{(x)}_{J,l,\mu}$ is defined as

$$c^{(x)}_{l,J,\mu} = \frac{1}{4}\sum_{\nu} \frac{(2i)\sqrt{E}\left(\Gamma^{(0)}_{n\mu}\Gamma_{x\mu}\Gamma^{(0)}_{n\nu}\Gamma_{x\nu}\right)^{1/2}}{z^*_\nu - z_\mu} \quad , \tag{2.18}$$

where $\Gamma^{(0)}_{n\mu} = \Gamma_{n\mu}/\sqrt{E}$ is the reduced neutron width.

The cross sections based on the MLBW approximation account for the multilevel effects attributed to the diagonal elements of the inverse level matrix while the multi-channel effects arising from the off-diagonal elements are not accounted for. Such an approximation may be satisfactory for nuclides with relatively isolated resonances. It can become unsatisfactory when fissionable nuclides with closely-spaced resonances are considered.

The Adler-Adler approach is based on a perturbation expansion approximation to derive expressions for cross sections which have the appearance of a sum of single-level contributions with a symmetric and anti-symmetric term for each level. This makes it possible to do Doppler broadening analytically. The generic forms defined in Eq. (2.16) are equally applicable here except that z_μ must be replaced by the eigenvalue ε_μ of the complex eigenvalue problem

$$\sum_{\nu}\left[E_\mu \delta_{\mu\nu} - \sum_c \gamma_{\mu c} L^0_c \gamma_{\nu c} \right] Z^{(\lambda)}_\nu = \varepsilon_\lambda Z^{(\lambda)}_\mu, \tag{2.19}$$

and $\Gamma^{1/2}_{x\mu}$ must be replaced by the residue amplitude

$$g^{1/2}_{\mu c} = \sqrt{2P_c}\sum_{\nu} Z^{(\mu)}_\nu \gamma_{\nu c}. \tag{2.20}$$

Here $\gamma_{\mu c}$ is the reduced width amplitude, $L^0_c = L_c - B_c$, L_c is the logarithmic derivative of the outgoing wave function at the channel radius, B_c is the boundary parameter taken to be a real constant, and P_c is the penetration factor. The corresponding residue parameters are given by

$$R^{(x)}_{l,J,\mu} = \frac{(4\pi\lambda\!\!\!^{-2} g_J)E}{4}\sum_{\nu} \frac{(2i)\sqrt{E}\left(g^{(0)}_{\mu n} g_{\mu x} g^{(0)}_{\nu n} g_{\nu x}\right)^{1/2}}{\varepsilon^*_\nu - \varepsilon_\mu}, \tag{2.21a}$$

$$R^{(t)}_{l,J,\mu} = \left(4\pi\lambda\!\!\!^{-2} g_J E\right) e^{-i2\phi_l} g_{n\mu}. \tag{2.21b}$$

The Reich-Moore formalism is an approximation that best preserves the rigor of the R-matrix theory and used to be most preferred for resonance data evaluations. The only significant assumption is given below [10]

$$\sum_{c\in\gamma} \gamma_{\lambda c} L_c^0 \gamma_{\mu c} = \delta_{\lambda\mu} \sum_{c\in\gamma} \gamma_{\lambda c}^2 L_c^0 \,. \tag{2.22}$$

That is, the photon channels that appear in the inverse level matrix are taken to be diagonal. Because of the rigor of the Reich-Moore formalism in representing energy behavior of the cross sections, it has often been used in the evaluations of the high resolution for many major nuclides. However, a problem in the Reich-Moore formalism is the apparent difficulties in obtaining the Doppler broadening in the subsequent calculations. As in the POLLA code [11], the R-matrix parameters can be converted to Adler-Adler parameters via diagonalization of the level matrix, neglecting the energy dependence of the neutron width in computing the complex poles and the corresponding residues. Hence, its application is limited to resonances of fissionable isotopes with $\Gamma_n << \Gamma_\gamma + \Gamma_f$ or other narrow resonances where the energy dependence is unimportant. In other words, the Adler-Adler approach works well for s-wave resonances. However, when the cross sections of the intermediate nuclides and the higher angular momentum resonances are considered, the energy dependency of the neutron width is more complicated than that of the s-wave due to the differences in the penetration factors and shift factors.

The multi-pole representation of the R-matrix cross sections in the momentum space overcomes this problem, maintaining the Doppler broadening analytically. In addition, a special case of the multi-pole formulation for s-wave resonances becomes the same as the Adler-Adler one.

Cross sections in the Reich-Moore formalism can be expressed using the rigorous pole representation as [8]

$$\sigma_x = \frac{1}{E} \sum_{l,J} \sum_{\lambda=1}^{N} \sum_{j=1}^{2(l+1)} \mathrm{Re}\left[R_{l,J,j,\lambda}^{(x)} \cdot \frac{(-i)}{p_\lambda^{(j)*} - \sqrt{E}} \right], \tag{2.23}$$

where $p_\lambda^{(j)*}$ = pole for level λ and complex conjugate pair j,

$R_{l,J,j,\lambda}^{(x)}$ = residue for angular momentum state l, compound nucleus spin J, complex conjugate pair j, and level λ, including the phase shift factor $\exp(-i2\varphi_l)$ for the total resonance cross section,

x = reaction type,

J = compound nucleus spin,

l = angular momentum state,

N = total number of Reich-Moore resonances.

For the s-wave, the rigorous pole representation and the traditional formalism consist of an identical number of terms with the same functional form in the momentum domain. In

particular, the Adler-Alder formalism for the s-wave can be considered as a special case of the rigorous pole representation when $p_\lambda^{(1)} = -p_\lambda^{(2)}$ and $R_{l,J,1,\lambda}^{(x)} = R_{l,J,2,\lambda}^{(x)}$.

For the higher angular momentum states, the poles for a given resonance λ can be divided into two groups: two s-wavelike poles ($j \in 1,2$) and the other extremely closely spaced poles ($j \notin 1,2$). The latter poles fluctuate only slightly from the average value, and their contributions are smooth and temperature independent. As a result, by taking $p_\lambda^{(1)}$ and $p_\lambda^{(2)}$ to be poles with positive and negative real components and regrouping the pole terms, Eq. (2.23) can be simplified as

$$\sigma_x = \frac{1}{E}\sum_l \ \mathrm{Re}\left\{ \sum_J \ \sum_{\lambda=1}^{N} \ \left(R_{l,J,1,\lambda}^{(x)} \cdot \frac{2(-i)\sqrt{E}}{\left[p_\lambda^{(1)*} \right]^2 - E} \right) + s_l^{(x)}\left(\sqrt{E}\right) + q_l^{(x)}\left(\sqrt{E}\right) \cdot \delta_{l0} \right\}, \tag{2.24}$$

where $s_l^{(x)}\left(\sqrt{E}\right) = \sum_J \ \sum_{\lambda=1}^{N} \ \left\{ \frac{R_{l,J,2,\lambda}^{(x)}(-i)}{p_\lambda^{(2)*} - \sqrt{E}} + \frac{R_{l,J,1,\lambda}^{(x)}(-i)}{p_\lambda^{(1)*} + \sqrt{E}} \right\}$,

$q_\ell^{(x)}\left(\sqrt{E}\right)$ = contributions of the smooth terms consisting of the sum of the pole terms attributed to $j \notin 1, 2$ for a given angular momentum l.

Thus, the rigorous pole representation can be viewed as a combination of a fluctuating term consisting of N poles with $\mathrm{Re}\left[p_\lambda^{(1)} \right] > 0$ and two non-fluctuating terms which are respectively attributed to the tails of outlying poles with negative real component ($s_l^{(x)}$) and the poles with extremely large width for $l > 0$ states ($q_l^{(x)}$). The smooth behavior of the non-fluctuating components in the effective range of interest can be reproduced using simpler functions rather than the computation of numerous pole terms. Therefore, the terms of $s_l^{(x)}$ and $q_l^{(x)}$ are expressed in the form of pole expansion with a few number of pseudo poles (mostly 3 pseudo poles). For s-waves, the spacing of the poles in momentum space is reasonably distant as the result of Wigner repulsion. Therefore, the computation of multi-pole parameters for the s-wave resonances is trivial. For higher l states, however, the poles in question generally consist of $2N$ s-wave-like poles with distant spacing and $2l \cdot N$ poles with extremely closely spaced regardless of how well separated the input Reich-Moore resonances are.

One of the advantages of the multi-pole representation is to do Doppler-broadening analytically on the fly, which allows to use simple temperature-independent libraries. The Doppler-broadened cross section can be represented in conjunction with the Gauss kernel $G(E, E')$ as

$$\sigma_x\left(E, \Delta\right) \approx \int_0^\infty \sigma\left(E', 0\right) G\left(E, E'\right) \ dE', \tag{2.25}$$

where $G(E,E') = \frac{1}{\sqrt{\pi}\Delta}\exp\left[-\frac{(E-E')^2}{\Delta^2}\right]$, $\Delta = \left[\left(4kTE\right)/A\right]^{1/2}$ is the Doppler-width, k is the Boltzmann constant, A is the atomic weight, and T is the absolute material temperature. Using this equation, each Lorentzian term described previously can be written as

$$\left(\sqrt{\pi}\theta_k/2\right)W(z) = \left[\psi\left(x,\theta_k\right) + i\,\chi\left(x,\theta_k\right)\right], \tag{2.26}$$

where $W(z) = \frac{i}{\pi}\int_{-\infty}^{\infty}\frac{e^{-t^2}\,dt}{z-t}$ and $z = \left(E - z_k\right)/\Delta$.

For the SLBW and MLBW formalisms, $z = \left(E - E_k\right) + i\theta_k/2$ with $\theta_k = \Gamma_t/\Delta$. For the Adler-Adler formalism z_k is identifiable with ε_k described previously. The complex probability integral $W(z)$ is a well-known function while the ψ- and χ-functions are its symmetric and asymmetric components of the traditional Doppler-broadened functions (see Appendix A for calculation of Doppler-broadening line shape functions). For example, the Doppler-broadened cross sections for the case of SLBW formalism become

$$\sigma_x\left(E,T\right) = \sum_{l,J}\sum_{k}\sigma_{0xk}\psi\left(x,\theta_k\right), \tag{2.27a}$$

$$\sigma_t\left(E,T\right) = \sum_{l,J}\sum_{k}\sigma_{0tk}\left[\cos 2\phi_l\,\psi\left(x,\theta_k\right) + \sin 2\phi_l\,\chi\left(x,\theta_k\right)\right]. \tag{2.27b}$$

Other comparable approximations can also be expressed in the same general forms accordingly.

The rigor of the energy dependence of cross sections as defined by any formalism can always be preserved if it is cast into the form of a generalized pole expansion in the $\sqrt{E}$-domain. Therefore, the exact Doppler-broadening of each Lorentzian-like term can be carried out analytically as before if the rigorous Solbrig's kernel is used. Substituting Eq. (2.24) into Eq. (2.25) leads to

$$\sigma_x\left(\sqrt{E},\Delta_m\right) = \frac{1}{E}\sum_{l,J}\sum_{\lambda}^{N+l}\sum_{j=1}^{2} Re\left\{R^{(x)}_{l,J,j,\lambda}B_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right)\right\}, \tag{2.28a}$$

$$\sigma_{tR}\left(\sqrt{E},\Delta_m\right) = \frac{1}{E}\sum_{l,J}\sum_{\lambda}^{N+l}\sum_{j=1}^{2} Re\left\{R^{(t)}_{l,J,j,\lambda}\,e^{-2i\phi_l}B_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right)\right\}, \tag{2.28b}$$

where $B_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right)$ is the Doppler-broadened line-shape function of each pole given by

$$B_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right) = \frac{\sqrt{\pi}}{\Delta_{mW}}W\left(\frac{\sqrt{E}-p_{\lambda}}{\Delta_m}\right) - C_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right);\ x\in a,f, \tag{2.29a}$$

$$C_x\left(\sqrt{E},\Delta_m,p^{(j)}_{\lambda}\right) = \frac{-i\,p^{(j)}_{\lambda}}{\sqrt{\pi}\Delta_m^2}\exp\left(-\frac{E}{\Delta_m^2}\right)\int_0^{\infty}\frac{\exp\left[-t^2-\left(2\sqrt{E}/\Delta_m\right)t\right]}{\left(p^{(j)}_{\lambda}/\Delta_m\right)^2-t^2}\,dt. \tag{2.29b}$$

The quantity $C_x\left(\sqrt{E},\Delta_m,p_\lambda^{(j)}\right)$ can be viewed as the low energy correction term which vanishes exponentially as a function of $\left(E/\Delta_m^2\right)$. Typically, it becomes unimportant when $E\geq 1$ for most nuclides of practical interest. Thus, the Doppler-broadened line-shape function exhibits the same mathematical properties in $\sqrt{E}$-domain as the traditional one in *E*-domain except under the low energy limit.

2.2.2 Self-shielding

The structure of hyperfine groups is constructed in such a way that the pointwise cross sections are directly used, but self-shielded cross sections need to be prepared for the ultrafine group calculation. The effective microscopic cross section of an isotope in a group g with lethargy width Δu can be written as

$$\bar{\sigma}_{xg}=\sum_i\int_{u_g}^{u_{g-1}}\frac{\sigma_{xi}(u)F(u)}{\Sigma_t(u)}du\Big/\int_{u_g}^{u_{g-1}}\frac{F(u)}{\Sigma_t(u)}du, \tag{2.30}$$

where $F(u)$ = collision density,

$\sigma_{xi}(u)$ = cross section of resonance i for reaction type x,

$\Sigma_t(u)$ = total cross section for the mixture,

and the summation is taken over all the resonances in group g of the isotope of interest.

In MC^2-2, the analytic resonance integral approach based on the narrow resonance (NR) approximation is used to evaluate the integrals in Eq. (2.30). As an example, for notational simplicity, let us consider the case of single-level Breit-Wigner single-level resonances. By extending the integration interval to the entire range of lethargy and assuming a constant collision density over each resonance, Eq. (2.30) for capture or fission reaction can be approximated as

$$\bar{\sigma}_{xg}^i=\sum_i F_i\frac{\Sigma_p\Gamma_x^i}{\Delta u\,E_0^i f}\cdot\frac{1}{2}\int_{-\infty}^{\infty}\frac{\sigma_0^i\psi_i}{\Sigma_t(u)}dx_i=\sum_i F_i\frac{\Sigma_p\Gamma_x^i}{\Delta u E_0^i f}\cdot\frac{1}{N_i}J_i^*, \tag{2.31}$$

where $\bar{\sigma}_{xg}$ = effective cross section of reaction type x (fission or capture) for group g, F_i = constant collision density for resonance i, Σ_p = potential macroscopic scattering cross section of mixture, σ_0^i = peak cross section of resonance i, Γ_x^i = line width of reaction type x for resonance i, E_0^i = energy of resonance i, ψ_i = Doppler broadened symmetric line shape function of resonance i, $x_i = 2(E-E_0^i)/\Gamma_t^i$, N_i = atom density of the resonant isotope of interest, f = flux correction factor to account for the attenuation by the resonances, and J_i^* = resonance integral over the entire energy range (-∞,∞) for resonance i.

The resonance integral J_i^* in Eq. (2.31) that is determined by an integration over (-∞,∞) rather than the exact group interval (u_g,u_{g-1}) would yield an accurate contribution to the

group cross section only if the resonance is narrow enough to be completely contained within the lethargy interval of group g. However, for a resonance lying across a group boundary, this J_i^* integral approach overestimates the contribution of the resonance i to the group g cross section and underestimates the contribution to the adjacent group cross sections, introducing the so-called resonance tail effect. As a result, wide resonances should be screened out to avoid a large resonance tail effect. The resonance tail effect becomes more pronounced as the number of energy groups in the targeted broad group structure increases.

Furthermore, it has been observed that the resonance tail effect is more pronounced with the ENDF/B-VII.0 data than with the previous ENDF/B files because of the significantly increased number of resonances and the upper energy boundaries of resolved resonance region. Thus, the hyperfine group option RABANL [1] of MC^2-2 is recommended for accurate treatment of resolved resonances when the ENDF/B-VII.0 data is used. It should however be noted that the resonance tail effects on the ultrafine group cross sections cannot be completely removed even in the RABANL calculation since the hyperfine-group slowing-down equation is solved using the fission, inelastic scattering, and (n,2n) sources determined from the ultrafine group calculation and transferred from the ultrafine-group calculation in which the resonance tail effect remains.

As a remedy for the analytic resonance integral method, a numerical integration approach is used in MC^2-3. In this method, the self-shielded ultrafine-group cross sections are determined by the numerical integration of pointwise cross sections based on the NR approximation as

$$\bar{\sigma}_{xg}^{i} = \int_{\Delta u_g} \sigma_x^i(u) \frac{\Sigma_p(u)}{\Sigma_t(u)} du \Big/ \int_{\Delta u_g} \frac{\Sigma_p(u)}{\Sigma_t(u)} du , \tag{2.32}$$

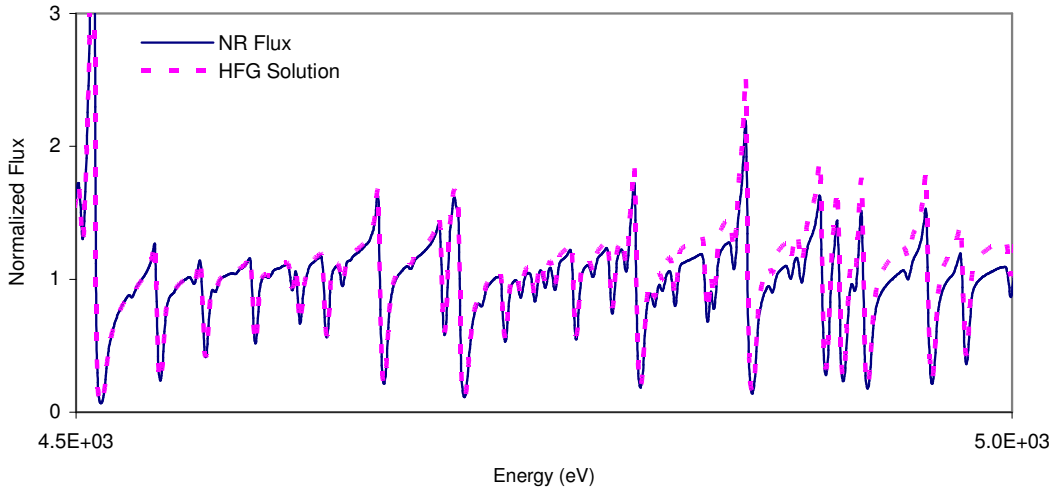
where $\bar{\sigma}_{xg}^{i}$ = effective cross section of group g for reaction type x of isotope i,

$\Sigma_p(u)$ = total macroscopic cross sections of mixture.

Since the potential cross sections are almost constant in the resonance energy range, Eq. (2.32) is simplified to

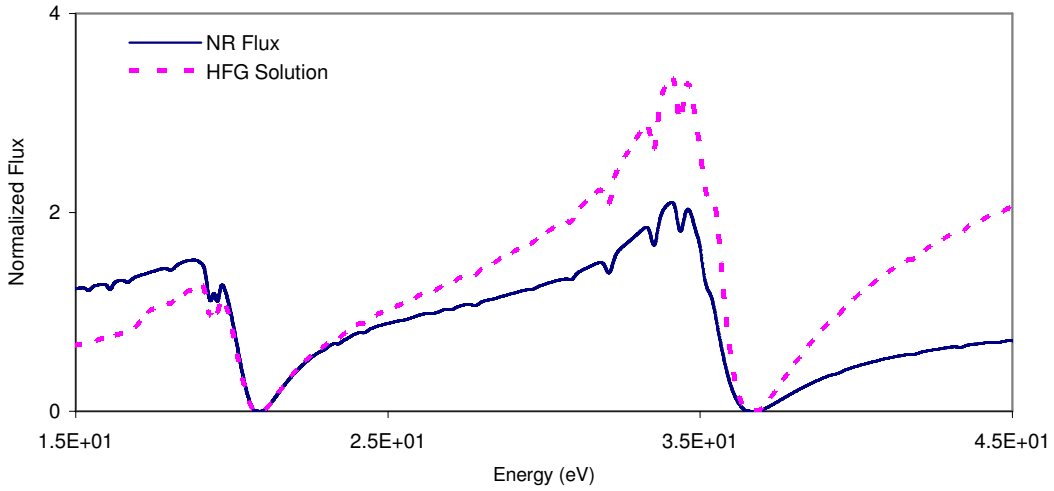
$$\bar{\sigma}_{xg}^{i} = \int_{\Delta u_g} \frac{\sigma_x^i(u)}{\Sigma_t(u)} du \Big/ \int_{\Delta u_g} \frac{1}{\Sigma_t(u)} du . \tag{2.33}$$

The NR approximation used in Eq. (2.32) is valid as far as the practical resonance width is much smaller than the average energy loss per scattering. Thus it is applicable except for the low-energy resonances of heavy isotopes below ~100 eV. As an example, Figure 2.4 and Figure 2.5 compare the neutron spectra obtained with the NR approximation and that from hyperfine group transport calculation. It can be seen that the NR approximation flux agrees well with the hyperfine group solution in the kilo-electron-volt energy ranges. On the other hand, the NR approximation shows significant deviations from the hyperfine group solution in a few tens of the electron-volt energy ranges. However, the neutron population in these low-energy ranges is very small in fast reactors, and thus the errors introduced in the ultrafine group cross sections by the NR approximation make no significant impact on global calculations. Furthermore, if needed, the hyperfine group calculation capabilities can be used.



(A mixture of U-235 and U-238)

Figure 2.4 Hyperfine Group and Narrow-Resonance Spectra at High Energy



(A mixture of U-235 and U-238)

Figure 2.5 Hyperfine Group and Narrow-Resonance Spectra at Low Energy

This numerical resonance integral approach not only improves the accuracies of the ultrafine group cross sections, but reduces the complexity of the analytic resonance integral approach of MC^2-2. With the new approach, screening out the wide resonances is no longer necessary and the complex procedure to account for the resonance overlapping effects is greatly simplified by calculating the total macroscopic cross sections of the mixture of interest directly using the pointwise cross sections of nuclides contained in the mixture. In addition, the effort to convert the Reich-Moore parameters to multi-pole parameters is reduced because the pseudo and out-range poles are no longer treated separately, unlike the previous method where those poles should be added to the smooth cross section before the resonance integration.

In order to improve the ultrafine-group high-order cross sections and consequently the leakage effect, a self-shielding method for the high-order moments of cross sections has been introduced based on the NR approximation. [12] In this method, the spectrum of the l-th flux moment within each ultrafine group is approximated as

$$\phi_l(u) \approx \frac{1}{[\Sigma_t(u)]^{l+1}}, \tag{2.34}$$

where $l \geq 0$ and Σ_t is the total macroscopic cross section of the mixture. The l dependence shown above would be appropriate for a large system with nearly isotropic scattering, i.e., the B_0 approximation.

The impact of different weighting functions on the total cross section moments of heavy nuclides is non-negligible, while the impact is relatively small for those of intermediate-weight nuclides. Due to the increased self-shielding effect, the high-order moments of total cross section become smaller than the 0^{th}-order moment (i.e., flux-weighted cross section). The reduced high-order moments of total cross section would lead to the increase of leakage.

2.2.3 Utilization of PENDF

As an alternative to reconstructing the pointwise cross sections in MC^2-3, the isotopic PENDF files generated with NJOY at multiple temperatures can be used. The PENDF file of NJOY contains capture, fission, scattering, and total cross sections at a given temperature as a function of energy, along with the interpolation law. When the PENDF files are given, MC^2-3 determines the pointwise cross sections at the temperature of interest by interpolating the PENDF data at relevant temperatures using the $T^{1/2}$ interpolation law. The available temperature table for all isotopes is stored in the file named "table_pendf." The PENDF data generated independently with NJOY are also very useful in verifying the pointwise cross sections reconstructed by MC^2-3. In addition, the PENDF files are used in self-shielding the resonance-like cross sections of intermediate weight nuclides above the resonance energy, since for this energy range, no data finer than the ultrafine group level are included in the MC^2 library.

2.3 Unresolved Resonance

For the unresolved resonance self-shielding, MC^2-3 basically uses the MC^2-2 algorithm developed by R. N. Hwang [13]. The algorithm assumes the narrow resonance approximation and accounts for the interference scattering, the effects of accidental overlap with resonances in other spin sequences, and the effects of self-overlap with resonances of the same spin sequence. The single level Breit-Wigner representation is always used for the unresolved resonance cross sections. No significant attenuation of flux in the energy interval of interest and a constant collision density are assumed.

Using the NR approximation, the effective macroscopic capture cross section for material m within an arbitrary energy interval $\Delta E = E_2 - E_1$ can be written as

$$\bar{\Sigma}_c^m(E^*) = N^m \bar{\sigma}_c^m(E^*) = \frac{\frac{1}{\Delta E}\int_{E_1}^{E_2} \frac{N^m \sigma_c^m(E)dE}{\Sigma_t(E)}}{\frac{1}{\Delta E}\int_{E_1}^{E_2} \frac{dE}{\Sigma_t(E)}}, \tag{2.35}$$

where E^* is an energy point within the interval $\Delta E = E_2 - E_1$, N^m is the atom density of material m, σ_c^m is the microscopic capture cross section for material m, and Σ_t is the total macroscopic cross section for the mixture. Here we assume no significant attenuation of flux in the energy interval ΔE and a constant collision density.

The microscopic cross section σ_c^m and the total macroscopic cross section Σ_t correspond to sums over contributing resonances belonging to various spin sequences: resonances having a particular angular momentum and channel spin. If we separate the total cross section of the equation above into a resonant part $\Sigma_r(E)$ and the remaining non-resonant part, i.e., potential part Σ_p, we can rewrite the equation as

$$\bar{\Sigma}_c^m(E^*) = \frac{\frac{1}{\Delta E}\int_{E_1}^{E_2} \frac{N^m \sum_{s,i} \sigma_{ci}^{m,s}(E)}{\sum_{m,s,i} \Sigma_{ri}^{m,s}(E) + \Sigma_p} dE}{\frac{1}{\Delta E}\int_{-\infty}^{\infty} \frac{1}{\sum_{m,s,i} \Sigma_{ri}^{m,s}(E) + \Sigma_p} dE}. \tag{2.36}$$

Here s represents a particular spin sequence and i represents the resonance in that sequence. The sum in the numerator of the upper integral ranges only over those sequences belonging to material m while the other sums are over all materials.

If we express the resonances in terms of the ψ and χ functions, perform the statistical average over the distribution functions for the resonance parameters, and factor the resulting equation, we may finally write the expectation value for a given spin sequence

$$\frac{\cos(2\phi_l)}{< D_k >} < \Gamma_\gamma^k J_k^* > \cong \left\{ \frac{1}{< D_k >} \left\langle \Gamma_\gamma^k J(\beta_k, \theta_k, a_k, 0) \right\rangle - O_\gamma^k \right\} S + \sum_{i \neq k} r_\gamma^{ki}, \tag{2.37}$$

where $J(\beta,\theta,a,b) = \frac{1}{2}\int_{-\infty}^{\infty} dx \frac{\psi(\theta,x) + b\chi(\theta,x)}{\beta + \psi(\theta,x) + a\chi(\theta,x)}$,

$\beta_k = \sigma_p / \sigma_0^k$,

$\theta_k = \frac{\Gamma_t^k \sqrt{A}}{\sqrt{4kTE_0^k}}$,

ϕ_l = the hard-sphere phase shift for angular momentum l,

$< D_k >$ = the average spacing of the spin sequence k.

In Eq. (2.37), S represents the first-order correction for the accidental overlap with the uncorrelated resonances in spin sequences $i \neq k$ given by

$$S = 1 - \sum_{i \neq k} \left[\frac{1}{< D_i >} < \Gamma_\gamma^i J(\beta_i, \theta_i, a_i, a_i) > - O_t^i \right]. \tag{2.38}$$

The r_γ^{ki} represents the higher-order corrections for the accidental overlap effect which to the second order is approximately given by

$$r_\gamma^{ki} \cong \frac{< \tau_\gamma^k >}{< D_k >} \frac{1}{< D_i >} \left\langle \Gamma_t^i \beta_i \frac{\partial J(\beta_i, \theta_i, 0, 0)}{\partial \beta_i} \right\rangle, \tag{2.39a}$$

$$< \tau_\gamma^k > = \left\langle \frac{\Gamma_\gamma^k}{2\beta_k} \int_{-\infty}^{\infty} \frac{\psi^2(\theta_k, x_k)}{\beta_k + \psi(\theta_k, x_k)} dx_k \right\rangle. \tag{2.39b}$$

As in MC^2-2, it is assumed that r_γ^{ki} is negligible. In Eq. (2.37), O_γ^k represents the capture self-overlap term for resonances of the same spin sequence k and is approximated by

$$O_\gamma^k \cong \frac{1}{< D_k >} \left\langle \frac{\Gamma_\gamma^k}{2} \int_{-\infty}^{\infty} \Omega(y) \frac{dy}{< D_k >} \int_{-\infty}^{\infty} \frac{\psi_k}{\beta_k + \psi_k} \frac{A_{k'} \psi_{k'}}{\beta_k + \psi_k + \beta_{k'} \psi_{k'}} dx_k \right\rangle_{k,k'}. \tag{2.40}$$

In Eq. (2.40), the resonances k and k' all belong to the same spin sequence k, and $A_{k'} = N_{k'} \sigma_0^{k'} / N_k \sigma_0^k$. $\Omega(y)$ is the probability of finding a resonance k' at a distance $y = E_0^k - E_0^{k'}$ from a given resonance k, and it is given by the Dyson two-level correlation function. Note that the contribution from the asymmetric line shape function χ is ignored for the self-overlap calculation. O_t^i in Eq. (2.38) is given by Eq. (2.40) where Γ_t^k replaces Γ_γ^k.

MC^2-3 neglects the higher-order corrections for the accidental overlap so that Eq. (2.37) becomes

$$\frac{1}{< D_k >} < \Gamma_\gamma^k J_k^* > \cong \frac{1}{\cos(2\phi_l)} \left[\frac{1}{< D_k >} < \Gamma_\gamma^k J_k^*(\beta_k, \theta_k, a_k, 0) > - O_\gamma^k \right] S. \tag{2.41}$$

If we define the flux correction factor f as

$$f = 1 - \sum_k \left[\frac{1}{< D_k >} < \Gamma_{\gamma_k} J_k^*(\beta_k, \theta_k, a_k, 0) > - O_\gamma^k \right], \tag{2.42}$$

we may write the effective unresolved resonance capture cross section for a given spin sequence k as

$$\bar{\sigma}_c^k = \frac{\sigma_p < \Gamma_\gamma^k J_k^* >}{< D_k > f}. \tag{2.43}$$

If we accept the first-order approximation for $B_i << 1$,

$$1-\sum_i B_i \approx \prod_i (1-B_i), \tag{2.44}$$

$\bar{\sigma}_c^k$ depends to the first order only on the resonances of sequence k since all other terms cancel in the ratio S/f. Thus, we have

$$S \approx \prod_{i\neq k}\left\{1-\left[\frac{1}{<D_i>}<\Gamma_t^i J(\beta_i,\theta_i,a_i,a_i)>-O_t^i\right]\right\}, \tag{2.45a}$$

$$f \approx \prod_{k}\left\{1-\left[\frac{1}{<D_i>}<\Gamma_t^i J(\beta_i,\theta_i,a_i,a_i)>-O_t^i\right]\right\}, \tag{2.45b}$$

so that

$$\bar{\sigma}_c^k \approx \frac{\sigma_p\left\{\frac{1}{<D_k>}<\Gamma_\gamma^k J(\beta_k,\theta_k,a_k,0)>-O_\gamma^k\right\}}{\cos(2\phi_l)\left\{1-\left[\frac{1}{<D_k>}<\Gamma_t^k J(\beta_k,\theta_k,a_k,a_k)>-O_t^k\right]\right\}}, \tag{2.46}$$

with similar expressions for the fission and total cross sections where Γ_γ^k is replaced by Γ_f^k and Γ_t^k, respectively.

The unresolved cross sections at each energy point E^* specified in the MC^2 library are finally obtained by summing the partial contributions such as given by Eq. (2.35) over all spin sequences. Thus, using Eq. (2.43), the unresolved cross section at energy point E^* for reaction type x (capture, fission, or total) of material (or isotope) m can be written as

$$\bar{\sigma}_x^m(E^*) = \sum_{k\in m}\bar{\sigma}_x^k(E^*) = \sum_{k\in m}\frac{\sigma_p <\Gamma_\gamma^k J_k^*>}{<D_k> f}, \tag{2.47}$$

where the summation ranges over all sequences belonging to material m.

The energy grid points E^* at which the average unresolved resonance parameters are supplied in the ENDF/B library vary with isotopes, and thus MC^2-3 generates a fixed energy grid on which the resonance integral, $<\Gamma_x^k J_k^*>/<D_k>$, can be interpolated linearly. That is, the self-shielded unresolved resonance cross sections are calculated at the specified energy grid points E^*, and the self-shielded resonance cross sections for the ultrafine groups in the unresolved resonance range of the particular isotopes of interest are determined by interpolating these self-shielded cross sections at E^*.

While transitioning from MC^2-2 to MC^2-3, the energy grid points E^* given in the library were used without any change. It was then noted that the energy grid points would be too coarse for the linear interpolation of self-shielded resonance cross sections when the background cross section varies significantly over the unresolved resonance range of the isotope of interest. This can happen when many resolved resonances of other isotopes in the mixture are

overlapped with the unresolved resonances of the isotope of concern. For example, a noticeable error can be observed in the unresolved resonance cross sections of U-238 around wide resonances of Fe-56 (27.8 keV), especially when a large amount of Fe-56 is present.

Sample results for a mixture of U-238 and Fe-56, in which the number density of Fe-56 is about 100 times larger than that of U-238, are shown in Figure 2.6 and Figure 2.7. Figure 2.6 shows the total cross sections of U-238 and Fe-56 in the U-238 unresolved energy range (20 keV – 150 keV), and Figure 2.7 shows percent (%) differences in the U-238 capture cross sections in the unresolved resonance energy range of U-238 between MC^2-3 and VIM [14] Monte Carlo results. It can be seen that the MC^2-3 cross section shows noticeable deviations from the VIM result around the resonance valleys of Fe-56, when the number of energy grid points E^* is small. The results also show that the deviations decrease as the number of grid points increases.

In order to enhance the accuracy of self-shielded unresolved resonance cross sections by eliminating the potential interpolation problem, the number of energy grid points for evaluating self-shielded cross sections in the unresolved energy range was increased significantly. The self-shielded unresolved resonance cross sections were evaluated at the mid-points of all the ultrafine groups within the unresolved resonance range of each isotope using the average resonance parameters interpolated from the ENDF/B data. Table 2.1 shows the number of energy grid points E^* along with the unresolved resonance range for major actinides. For example, the number of energy grid points E^* is increased from 18 to 241 for U-238; from 14 to 289 for U-235; from 70 to 296 for Pu-239. The original numbers of E^* point for all isotopes in ENDF/B-VII.0 are listed in Appendix B.

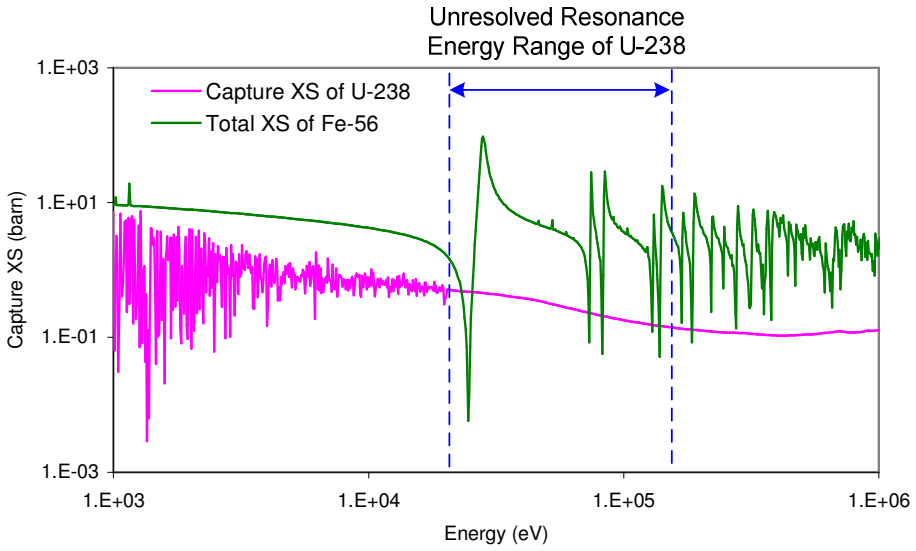
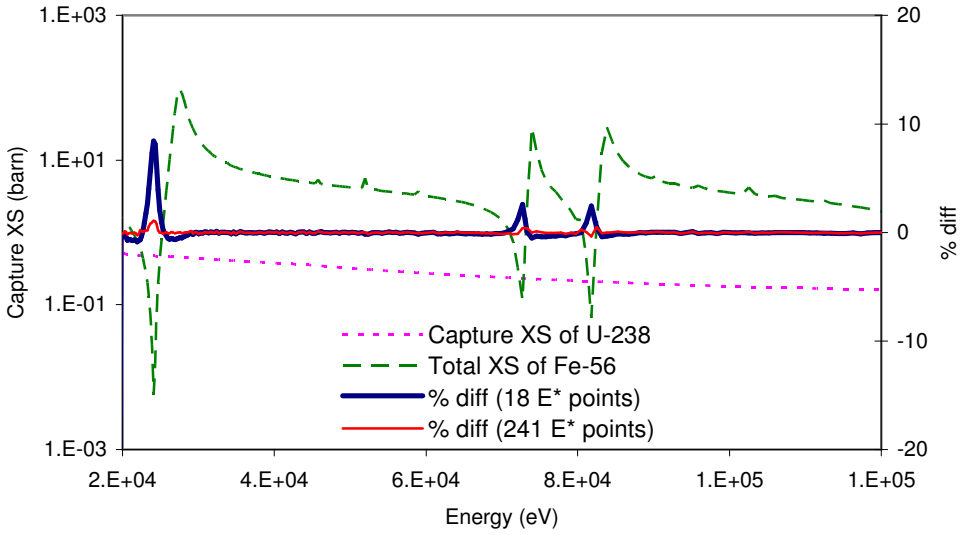


Figure 2.6 Cross Sections of U-238 and Fe-56



[% diff = (MC^2-3 – VIM) / VIM * 100]

Figure 2.7 Self-Shielded Unresolved Cross Sections of U-238 for Different Sets of E^* Points

Table 2.1 Unresolved Resonance Energy Grids for Actinides in MC^2-3

Isotope	UR Energy (keV)	E*		Isotope	UR Energy (keV)	E*	
		ENDF/B	MC^2-3			ENDF/B	MC^2-3
U-233	0.6 – 40	29	504	Pu241	0.3 – 40.2	24	588
U-234	1.5 – 100	10	504	Pu242	0.99 – 10	5	278
U-235	2.25 – 25	14	289	Am241	0.15 – 30	27	635
U-236	1.5 – 100	29	504	Am242	0.04 – 27.3	35	774
U-238	20 – 149	18	241	Am243	0.25 – 42.3	17	616
Np-237	0.5 – 35	16	510	Cm242	0.28 – 10	8	431
Pu238	0.2 – 10	3	470	Cm243	0.1 – 42.2	22	725
Pu239	2.5 – 29.5	70	296	Cm244	1 – 40	10	443
Pu240	5.7 - 40	3	234	Cm245	0.1 – 55	25	757

* UR: Unresolved Resonance.

In the ENDF/B data, the flag LSSF is used to specify the different procedures for processing the unresolved resonance cross sections. The option with LSSF = 1 is a new feature used in ENDF/B-VI, for which File 3 represents the entire dilute unresolved cross section and the parameters in File 2 are used to compute the self-shielding factors to be applied to the File 3 values. When LSSF is equal to one, the unresolved cross sections are directly derived by multiplying the shielding factors and accordingly the unresolved elastic scattering cross section is computed as:

$$\sigma_t^u = f_t^u \cdot (\sigma_t - \sigma_p) + \sigma_p , \tag{2.48a}$$

$$\sigma_c^u = f_c^u \sigma_c , \tag{2.48b}$$

$$\sigma_f^u = f_f^u \sigma_f , \tag{2.48c}$$

$$\sigma_s^u = \sigma_t^u - \sigma_c^u - \sigma_f^u - \sigma_{other}, \tag{2.48d}$$

where σ_t , σ_c , σ_f = the smooth total, capture, and fission cross sections in File 3,

σ_p = the potential cross section,

f_t^u , f_c^u , f_f^u = the shielding factors for total, capture, and fission cross sections, respectively, computed using the parameters in File 2,

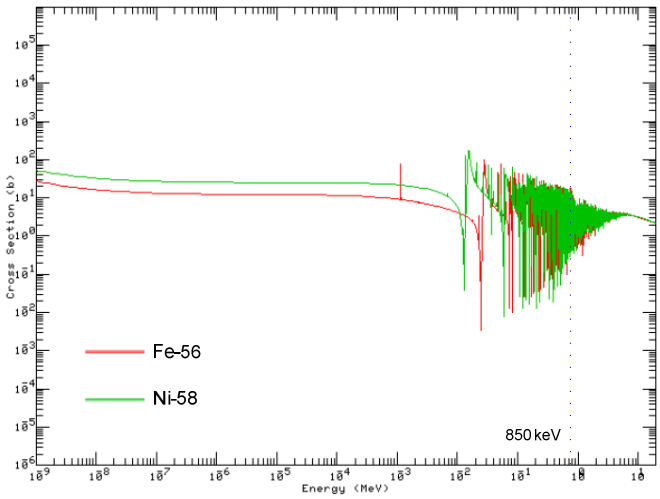
σ_{other} = the sum of cross sections other than capture, fission, and elastic scattering,

σ_s = the elastic scattering cross section.

As shown in Eq. (2.48), the shielding factor for total cross section is defined for only the portion of the total cross section excluding the potential cross section.

2.4 *Self-Shielding above Resonance Energy*

The intermediate-weight nuclides such as Fe, Cr, Ni, Mn, Co, and Cu have resonance-like fluctuating scattering cross sections above the resonance energy, as shown in Figure 2.8. When the amount of those nuclides in the mixture is small, the self-shielding effect of the cross sections above the resonance energy can normally be ignored. In the fast reactor system with a large amount of structure material, however, the self-shielding effect of the resonance-like cross sections of the medium-weight nuclides above the resonance range becomes non-negligible (up to 50 pcm).



(Resolved Resonance Cutoff Energy: 805 keV for Fe-56, 812 keV for Ni-58)

Figure 2.8 Total Cross Sections of Intermediate-Weight Nuclides

According to the previous analysis [15], it is normally required to use more than 8,000 groups in order to catch more than 90% of the self-shielding effect of those resonance-like scattering cross sections. In the previous practice, the shielded cross sections are prepared separately and incorporated into the smooth cross section file (File 5) of the MC^2 library using the SHIELD program [16]. The program directly reads the relevant ENDF/B files to reconstruct the detailed fine-group cross sections, shields those cross sections using the narrow resonance approximation, and then replaces all principal smooth cross sections by the shielded ones.

MC^2-3 provides a user option to self-shield the cross sections above the resonance energy range. Since there is no detailed cross section information finer than ultrafine groups in the given MC^2 library, PENDF files are required to reconstruct the hyperfine group cross sections in the above-resonance energy range. Once the hyperfine group cross sections for all isotopes in the mixture are reconstructed, Eq. (2.33) is applied to estimate the shielded ultrafine group cross sections for each isotope.

2.5 Elastic Scattering

For elastic scattering, the energy transfer and the deflection angle are completely correlated as

$$\frac{E'}{E} = \frac{1}{2}(1+\alpha) + \frac{1}{2}(1-\alpha)\mu_c , \tag{2.49}$$

where E and E' are the neutron energies before and after scattering, μ_c is the cosine of the scattering angle in the center-of-mass system, $\alpha = (A-1)^2/(A+1)^2$ and A is the atomic mass. The cosine of the scattering angle in the center-of-mass system is related to that in the laboratory system μ_s as

$$\mu_c = \frac{1}{A}\left[-(1-\mu_s^2) + \mu_s\sqrt{A^2-(1-\mu_s^2)}\right]. \tag{2.50}$$

Using Eq. (2.49) and Eq. (2.50), the deflection angles can be determined in terms of initial and final neutron energies as

$$\mu_c(E,E') = \frac{1}{1-\alpha}\left[2\frac{E'}{E} - (1+\alpha)\right], \tag{2.51a}$$

$$\mu_s(E,E') = \frac{1}{2}\left[(A+1)\sqrt{\frac{E'}{E}} - (A-1)\sqrt{\frac{E}{E'}}\right]. \tag{2.51b}$$

The differential scattering cross section is generally represented by a Legendre polynomial expansion in the form

$$\sigma_s(E,\mu_c) = \frac{\sigma_s(E)}{2\pi}\sum_{n=0}^{N}\frac{2n+1}{2} f_n(E) P_n(\mu_c). \tag{2.52}$$

Scattering into a particular range of energies requires scattering within an associated range of direction, and thus the differential scattering cross section can be written as a function of the final energy as

$$\sigma_s(E \to E') = 2\pi\sigma_s(E,\mu_c)\left|\frac{d\mu_c}{dE'}\right| = \frac{\sigma_s(E)}{(1-\alpha)E}\sum_{n=0}^{N}(2n+1)f_n(E)P_n\left[\mu_c(E,E')\right]. \quad (2.53)$$

Eq. (2.53) can be rewritten in terms of lethargy as

$$\sigma_s(u \to u') = \frac{\sigma_s(u)e^{-(u'-u)}}{(1-\alpha)}\sum_{n=0}^{N}(2n+1)f_n(u)P_n\left[\mu_c(u,u')\right]. \quad (2.54)$$

The Legendre expansion coefficient of the scattering transfer cross section can also be written as

$$\sigma_s^{\ell}(u \to u') = \frac{\sigma_s(u)e^{-(u'-u)}P_{\ell}[\mu_s(u,u')]}{(1-\alpha)}\sum_{n=0}^{N}(2n+1)f_n(u)P_n[\mu_c(u,u')]. \quad (2.55)$$

Integrating Eq. (2.55) over all sink energies yields

$$\sigma_s^{\ell}(u) = \int_u^{u+\varepsilon}\sigma_s^{\ell}(u \to u')du' = \sum_{n=0}^{N}\sigma_s(u)f_n(u)A_n^{\ell}(u), \quad (2.56)$$

where $\varepsilon = \ln(1/\alpha)$ and $A_n^{\ell}(u) = \frac{2n+1}{1-\alpha}\int_u^{u-\ln\alpha}du'\,P_{\ell}[\mu_s(u,u')]P_n[\mu_c(u,u')]e^{-(u'-u)}$.

The elastic scattering transfer matrix for the multigroup spectrum calculation can be obtained as

$$\begin{aligned}\sigma_s^{\ell}(g \to g') &= \frac{1}{\phi_{\ell g}}\int_{u_{g'-1}}^{u_{g'}^*}du'\int_{u_{g-1}^*}^{u_g}du\,\sigma_s^{\ell}(u \to u')\phi_{\ell} \\ &= \frac{1}{\phi_{\ell g}}\int_{u_{g'-1}}^{u_{g'}^*}du'\int_{u_{g-1}^*}^{u_g}du\frac{\phi_{\ell}(u)\sigma_s^{\ell}(u)e^{-(u'-u)}P_{\ell}[\mu_s(u,u')]}{1-\alpha}\sum_{n=0}^{N}(2n+1)f_n(u)P_n[\mu_c(u,u')],\end{aligned} \quad (2.57)$$

where $u_{g'}^*$ and u_{g-1}^* are energetically reachable boundaries as shown in Figure 2.9.

Under the assumption that the scattering cross section and the flux are constant within the ultrafine group lethargy width, which is reasonable as long as the narrow resonance approximation is valid, Eq. (2.57) can be written as

$$\sigma_s^{\ell}(g \to g') = \frac{\sigma_{sg}}{(1-\alpha)\Delta u}\sum_{n=0}^{N}(2n+1)\int_{u_{g'-1}}^{u_{g'}^*}du'\int_{u_{g-1}^*}^{u_g}du\,f_n(u)P_{\ell}[\mu_s(u,u')]P_n[\mu_c(u,u')]e^{-(u'-u)}. \quad (2.58)$$

Figure 2.9 Ultrafine Energy Transfer in Elastic Scattering

As in MC^2-3, the elastic scattering transfer matrix elements are determined in three different ways, depending on the maximum number of down-scattering ultrafine groups. As shown in Table 2.2, the scattering materials are grouped into three classes: hydrogen, light elements, and heavy elements. The heavy elements are the nuclides of which the maximum number of down-scattering groups is three (i.e., atomic mass is greater than 160).

Table 2.2 Maximum Number of Down-Scattering Ultrafine Groups

Isotope	Class	Down	Isotope	Class	Down	Isotope	Class	Down
H-1	Hydrogen	2081	Ti-46	Light	11	Xe-135	Light	4
B-10	Light	49	Cr-50		10	Sm-149		4
C-12		41	Fe-56		9	Gd-155		4
O16		31	Ni-60		9	Er-167	Heavy	3
Na-23		22	Ga-69		8	U-238		3
Cl-35		14	Zr-90		6	Pu-239		3
Ca-40		13	Mo-90		6	Cm-242		3

* Down: the maximum number of down-scattering ultrafine groups.

2.5.1 Heavy Elements

For heavy elements which scatter less than four ultrafine groups (i.e., A > 160 for Δu = 1/120), the method by Henryson [17] is used. The elastic scattering transfer matrix in Eq. (2.58) can be rewritten as

$$\sigma_s^\ell(g \to g') = \frac{1}{\phi_{\ell g}} \sum_{n=0}^{N} < \phi_\ell(u)\sigma_s(u) f_n(u) >_g A_n^\ell(g \to g'), \tag{2.59}$$

where $<>_g$ is a suitable average over the source group g, and

$$A_n^\ell(g \to g') = \frac{2n+1}{1-\alpha} \int_{u_{g'-1}}^{u_{g'}^*} du' \int_{u_{g-1}^*}^{u_g} du P_\ell(\mu_s) P_n(\mu_c) e^{-(u'-u)}. \tag{2.60}$$

Taking the group width small enough to permit a constant weight function, we have

$$< \phi_\ell(u)\sigma_s(u) f_n(u) >_g \approx \frac{\phi_{\ell g}}{\Delta u} \sigma_s \bar{f}_{ng}, \tag{2.61}$$

where $\bar{f}_{ng}$ is the group averaged Legendre coefficients which are calculated using the Legendre coefficients at group boundaries and the given interpolation law.

The summation of $A_n^\ell(g \to g')$ over all the sink groups yields

$$A_n^\ell(g) = \frac{2n+1}{2} \int_{u_{g-1}}^{u_g} du \int_0^{-\ln\alpha} dU P_\ell[\mu_s(U)] P_n[\mu_c(U)] \left(-\frac{d\mu_c}{dU} \right) = T_{\ell n}^0(\alpha) \Delta u, \tag{2.62}$$

where the function $T_{\ell n}^m$ is defined by

$$T_{\ell n}^m(\alpha) = \frac{(-1)^m}{m!} \frac{2n+1}{2} \int_0^{-\ln\alpha} dU U^m P_\ell[\mu_s(U)] P_n[\mu_c(U)] \left(-\frac{d\mu_c}{dU} \right). \tag{2.63}$$

The $T_{\ell n}^{m}(\alpha)$ function is pre-calculated for $m=0,1,2$ and saved in the File 8 of the MC^2 library.

In a similar manner, the general matrix elements $A_n^{\ell}(g \to g')$ are related to a function similar to the T-function. For example, consider the element $A_n^{\ell}(g \to g+3)$ when the group width and scattering band are related through the inequality $\varepsilon/3 \le \Delta u \le \varepsilon/2$ so that the element may scatter a neutron through three energy groups. In this case, Eq. (2.60) becomes

$$\begin{aligned} A_n^{\ell}(g \to g+3) &= \frac{2n+1}{2}\int_{2\Delta u}^{-\ln\alpha} dU\,(U-2\Delta u)P_{\ell}[\mu_s(U)]P_n[\mu_c(U)]\left(-\frac{d\mu_c}{dU}\right) \\ &= 2\Delta u\left[\overline{T}_{\ell n}^{0}(\alpha,2\Delta u)-\overline{T}_{\ell n}^{0}(\alpha,-\ln u)\right]+\left[\overline{T}_{\ell n}^{1}(\alpha,-\ln\alpha)-\overline{T}_{\ell n}^{1}(\alpha,2\Delta u)\right], \end{aligned} \tag{2.64}$$

where $\overline{T}_{\ell n}^{m}(\alpha,\beta)=\dfrac{(-1)^m}{m!}\dfrac{2n+1}{2}\displaystyle\int_0^{\beta} dU U^m P_{\ell}[\mu_s(U)]P_n[\mu_c(U)]\left(-\frac{d\mu_c}{dU}\right)$.

2.5.2 Light Elements

For light elements other than hydrogen which scatter more than three ultrafine groups, that is, A$\le$160 for Δu =1/120, an analytic integration of the elastic scattering transfer equation over the sink group is coupled with a simple average over the ends of the source group. By changing the order of integration, the elastic scattering transfer equation can be rewritten in the following form

$$\sigma_s^{\ell}(g \to g') = \frac{\sigma_{sg}}{\Delta u}\int_{u_{g-1}^{*}}^{u_g} du\, P_{\ell}(u \to g'), \tag{2.65a}$$

$$P_{\ell}(u \to g') = \sum_{n=0}^{N}\frac{2n+1}{1-\alpha}\int_{u_{g'-1}}^{u_{g'}^{*}} du'\, f_n(u)P_{\ell}(\mu_s)P_n(\mu_c)e^{-(u'-u)}. \tag{2.65b}$$

By transforming the variable from u' to r

$$r=\frac{1-e^{-(u'-u)}}{1-\alpha}\le 1, \tag{2.66}$$

and using the relations

$$dr=\frac{e^{-(u'-u)}}{1-\alpha}du', \quad \mu_c=\frac{1}{1-\alpha}[2e^{-\Delta u}-(1+\alpha)]=1-2r, \tag{2.67}$$

$P_0(u \to g')$ can be evaluated analytically as

$$P_0(u \to g') = \sum_{m=0}^{N} A_m(u)(r_{g'}^{m+1}-r_{g'-1}^{m+1}), \tag{2.68}$$

where $A_m(u)=\displaystyle\sum_{n=m}^{N}(2n+1)f_n(u)\frac{(-1)^m(n+m)!}{m!(m+1)!(n-m)!}=\sum_{n=m}^{N}c_n^m(2n+1)f_n(u)$.

Using the analytic expression in Eq. (2.68), the scattering transfer matrix is evaluated numerically by dividing each ultrafine group into H hyperfine groups, as shown in Figure 2.10, such that

$$\delta u = \Delta u / H , \tag{2.69a}$$

$$u_g^h = u_g + h\,\delta u \quad (h = 0,1,\cdots,H) . \tag{2.69b}$$

The number of hyperfine groups per ultrafine group is chosen so that (1) H does not exceed the number specified by the user; (2) $H = 1$ if scattering is isotropic or linearly anisotropic in the center-of-mass system (i.e., $N \le 1$); (3) $H = 1$ if the isotope scatters in at least 20 ultrafine groups; (4) H is calculated to ensure that the isotope scatters in at least 20 hyperfine groups, i.e., H > 20 $H > 20\Delta u / \varepsilon$.

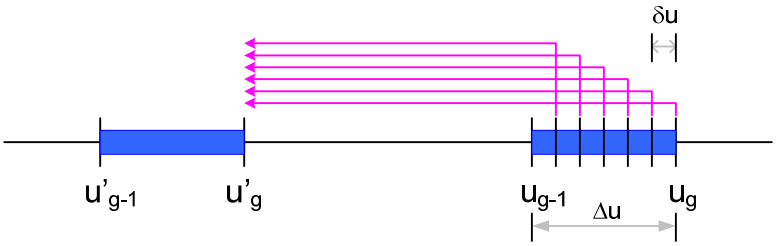


Figure 2.10 Hyperfine Energy Transfer in Elastic Scattering

Within this uniform subdivision of ultrafine groups, $P_0\,(u_h^g \to g')$ is calculated by adding the contributions for the hyperfine groups in the sink group g'. In each source ultrafine group, r and f_n are evaluated at the boundaries of all energetically reachable hyperfine groups. Using these hyperfine group boundary values of r and f_n, and the pre-calculated factorial coefficients c_n^m, the scattering probabilities $P_0\,(u_h^g \to k_h)$ from a hyperfine group boundary lethargy u_h^g in the source group g to all the energetically reachable sink hyperfine groups k_h are calculated using Eq. (2.68). The transfer matrix is then obtained using the trapezoidal rule integration,

$$P_0\,(u_g^h \to g') = \sum_{k_h \in g'} P_0\,(u_g^h \to k_h) , \tag{2.70a}$$

$$\sigma_s^0(g \to g') = \frac{\sigma_{sg}}{2H}\left[P_0\,(u_{g-1} \to g') + P_0\,(u_g \to g') + 2\sum_{h=1}^{H-1} P_0\,(u_{g-1} + h\delta u \to g') \right]. \tag{2.70b}$$

For the last group which can be reached from a ultrafine group g, the matrix element is calculated simply by neutron balance

$$\sigma_s^0(g \to g^*) = \sigma_{sg} - \sum_{g'=g}^{g^*-1} \sigma_s^0(g \to g') , \tag{2.71}$$

where $u_{g^*-1} < u_g + \varepsilon \le u_{g^*}$.

For the consistent P_1 equation, the matrix element of Eq. (2.66) for $\ell = 1$ are required. It is possible to derive $P_1(u \to g')$ analytically, but it is time consuming and difficult to evaluate numerically. Thus, the integration is approximately evaluated by taking

$$P_1(u \to \bar{k}_h) \approx \mu_s(u \to \bar{k}_h) P_0(u' \to k_h). \tag{2.72}$$

The small group size, $\delta u \le \Delta u \approx 0.0083$, makes this a good approximation. The code uses the energy midpoint of the hyperfine sink group in this calculation. μ_s can be determined by

$$\mu_s(u \to \bar{k}_h) = \frac{1}{2} \bar{X}_{k_h}^{1/2} \left[(A+1) - (A-1)(\bar{X}_{k_h})^{-1} \right], \tag{2.73}$$

where $\bar{X}_{k_h} = \frac{1}{2}[e^{-(u_{k_h-1}-u)} + e^{-(u_{k_h}-u)}] \approx \frac{1}{2} e^{-(k_h-1)\delta u}(1+e^{-\delta u})$.

The P_1 matrix is calculated in a manner analogous to the P_0 matrix,

$$P_1(u_g^h \to g') = \sum_{k_h \in g'} \mu_0(u_g^h \to \bar{k}_h) P_0(u_g^h \to k_h), \tag{2.74a}$$

$$\sigma_s^1(g \to g') = \frac{\sigma_{sg}}{2H} \left[P_1(u_{g-1} \to g') + P_1(u_g \to g') + 2\sum_{h=1}^{H-1} P_1(u_{g-1} + h\delta u \to g) \right]. \tag{2.74b}$$

For the last group which can be reached from g', the matrix element is calculated by the balance

$$\sigma_s^1(g \to g^*) = \sigma_{sg'}^1 - \sum_{g=g'}^{g^*-1} \sigma_s^1(g \to g), \tag{2.75}$$

where $\sigma_{sg}^1 = \sum_{n=0}^{N} \sigma_{sg} \bar{f}_{ng} T_{1n}^0(\alpha)$.

2.5.3 Hydrogen

For hydrogen, the cosine of the scattering angle in the laboratory system reduces to

$$\mu_s(u, u') = e^{-(u'-u)/2}. \tag{2.76}$$

Assuming the scattering is isotopic in the center-of-mass system, the elastic scattering transfer matrix in Eq. (2.58) is simplified as

$$\sigma_s^\ell(g \to g') = \frac{\sigma_{sg}}{(1-\alpha)\Delta u} \int_{u_{g'-1}}^{u_{g'}^*} du' \int_{u_{g-1}^*}^{u_g} du\, P_\ell[e^{-(u'-u)/2}] e^{-(u'-u)}. \tag{2.77}$$

The transfer matrix in Eq. (2.77) can be determined relatively easily, but a full ultrafine group scattering band needs large memory requirements. Thus, as in MC^2-2, recursive relationships are used to evaluate the scattering from hydrogen on the fly instead of storing the pre-calculated scattering transfer matrix.

Let $S_{g\ell}^H$ be the P_ℓ elastic scattering source into group g from all groups above g due to scattering from hydrogen. Then, the neutron source due to scattering from hydrogen becomes

$$S_{g\ell}^H = \int_{u_{g-1}}^{u_g} du' \int_0^{u_{g-1}} du \phi_\ell(u) \Sigma_s^H(u) e^{-(u'-u)} P_\ell[e^{-(u'-u)/2}]. \tag{2.78}$$

For $\ell = 0$ and 1, Eq. (2.78) can be rewritten as

$$S_{g\ell}^H = \int_{u_{g-1}}^{u_g} du' \int_0^{u_{g-1}} du \phi_\ell(u) \Sigma_s^H(u) e^{-(1+\ell/2)(u'-u)} = \eta_\ell(u_{g-1})[1-e^{-(1+\ell/2)\Delta u}], \tag{2.79}$$

where $\eta_\ell(u_g)$ is the hydrogen slowing down density defined as

$$\eta_l(u) = \frac{1}{1+l/2} \int_0^u du' \phi_l(u') \Sigma_s^H(u') e^{-(1+l/2)(u-u')} \quad (l = 0,1). \tag{2.80}$$

Assuming the flux and the hydrogen scattering cross section are constant within a ultrafine group, the hydrogen slowing-down density at an ultrafine group boundary can be calculated recursively as

$$\eta_\ell(u_g) \approx \eta_\ell(u_{g-1}) e^{-(1+l/2)\Delta u} + \frac{\phi_{\ell g} \Sigma_{sg}^H}{(1+\ell/2)^2 \Delta u}[1-e^{-(1+\ell/2)\Delta u}] \quad (\ell = 0,1), \tag{2.81a}$$

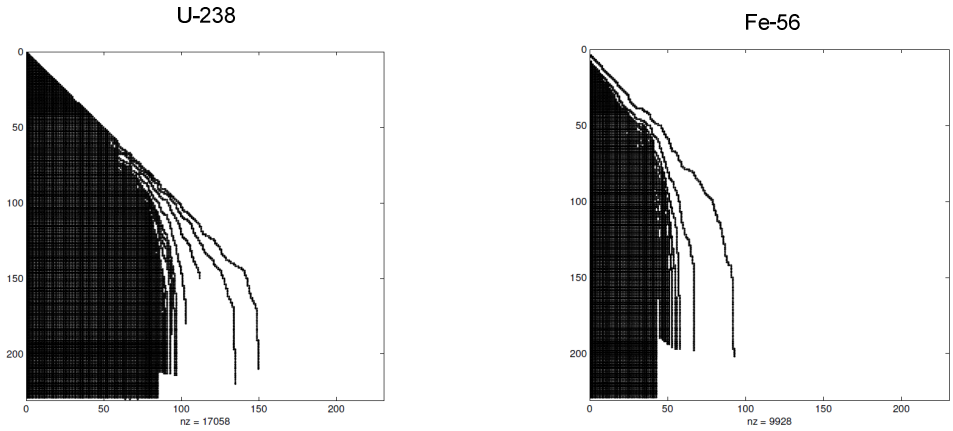
$$\eta_{\ell g} = \int_{u_{g-1}}^{u_g} \eta_\ell(u) du \approx \eta_\ell(u_{g-1}) \Delta u, \tag{2.81b}$$

where the initial condition is $\eta_\ell(0) = 0$.

In the ultrafine transport equation, the scattering source from hydrogen is counted separately from all the other nuclides.

2.6 Non-elastic Scattering

Inelastic scattering and (n,2n) reaction are threshold reactions and thus occur at relatively high neutron energies where resonances are small and self-shielding has little effect. Because these reactions are endothermic, the neutron energy after reaction can be very low, as shown in Figure 2.11. Indeed, non-elastic scattering is the dominant mechanism for energy loss in the high energy range. In major nuclides of fast reactors such as U-238 and Fe-56, the portion of inelastic scattering to the total cross sections is up to 40-50% at high energies greater than 1 MeV, as shown in Figure 2.12. Therefore, non-elastic scattering plays an important role in fast reactor systems.



U-238 (Q = 45 keV), F-56 (Q = 84 keV)

Figure 2.11 Ultrafine Group Inelastic Scattering Matrices for U-238 and Fe-56

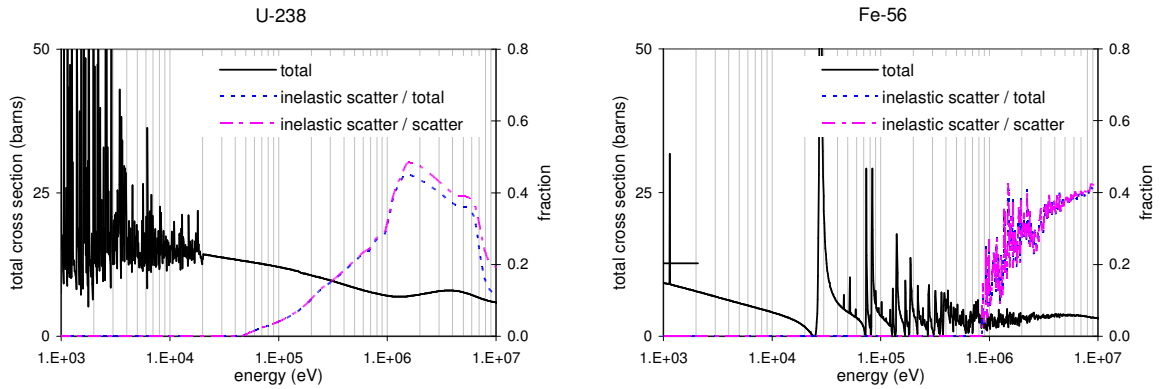


Figure 2.12 Group-Total Inelastic Scattering Cross Sections of U-238 and Fe-56

In MC^2-2, in order to save the storage due to the almost full scattering band, inelastic scattering and (n,2n) reactions are directly included in the source term in the ultrafine group calculation. Both the inelastic scattering and (n,2n) sources are assumed isotropic in the laboratory system although the anisotropy of inelastic scattering is not negligible at high energies. On the other hand, MC^2-3 prepares the energy transfer matrices of inelastic scattering and (n,2n) reaction in the same manner as for elastic scattering. In addition, the inelastic scattering is allowed to be anisotropic whereas (n,2n) reaction is still treated as isotropic. For both the inelastic scattering and (n,2n) reaction, MC^2-3 permits three descriptions of the secondary energy distributions: tabulated function, evaporation spectrum, and discrete levels.

2.6.1 Tabulated Function

For the tabulated function, the following data are provided on the File 6 of the MC^2 library,

$P_x(g \rightarrow E_{tab})$ = probability that a neutron is scattered by reaction type x (inelastic or (n,2n)) from group g to energy point E_{tab},

E_{tab} = an array of sink energy points,

KT = an interpolation law on the sink energies E_{tab},

σ_{xg} = cross section in group g for reaction x multiplied by the fraction of scattering events described by the P_x law.

The inelastic and (n,2n) scattering source into group g described by the tabulated law is given by

$$S_g^{tab} = \sum_{g'} \phi_{g'} \left[\sigma_{in,g'} \int_{E_{g-1}}^{E_g} dE\, P_{in}(g' \to E) + 2\sigma_{n2n,g'} \int_{E_{g-1}}^{E_g} dE\, P_{n2n}(g' \to E) \right], \tag{2.82}$$

The integration of Eq. (2.82) is evaluated analytically using the interpolation law defined by the KT data.

2.6.2 Evaporation Spectrum

The MC^2 library provides evaporation temperatures and fractional probabilities such that the source into group g by reaction type x is given by

$$S_{xg}^{evap} = \sum_{g'} \sigma_{xg'} \phi_{g'} \sum_n \frac{W_{ng'}^x}{I_{ng'}^x} \int_{E_{g-1}}^{E_g} dE\, P_n^x(g' \to E), \tag{2.83}$$

The evaporation spectra $P_n^x(g' \to E)$ are defined with source group dependent evaporation temperatures $\theta_{ng'}^x$ as

$$P_n^x(g' \to E) = \begin{cases} E\, e^{-E/\theta_{ng'}^x} & \text{for } E \le E_{g'-1} - U_n^x \\ 0 & \text{for } E > E_{g'-1} - U_n^x \end{cases}, \tag{2.84}$$

and $I_{ng'}^x$ is a normalization factor,

$$I_{ng'}^x = \int_0^{E_{g'-1}-U_n^x} dE P_n^x(g' \to E). \tag{2.85}$$

In Eq. (2.84), U_n^x is a constant introduced to define the range of final energies allowed, and it is assumed to be zero for (n,2n) reaction.

The evaluation of Eq. (2.83) requires an exponential for each ultrafine sink group, which takes substantial time. Therefore, a fast exponential function [1] is used to evaluate the required exponentials. In addition, the calculation over sink groups g is terminated once the following criterion is met:

$$\frac{\int_{E_{g-1}}^{E_g} dE\, P_n^x(g' \to E)}{\int_{E_{g-1}}^{E_{g'-1}-U_n^x} dE\, P_n^x(g' \to E)} \le 10^{-4}. \tag{2.86}$$

2.6.3 Discrete Levels

The MC^2 library provides the interaction Q values and ultrafine group cross sections for all inelastic and (n,2n) discrete scattering levels. It also gives the average cosine of the scattering angle in the center-of-mass system for discrete inelastic scattering by level and group.

2.6.3.1 Approximate Treatment

Using the energy and momentum conservation equations, the neutron energy after collision in the laboratory system can be written as

$$E = \frac{1+A^2}{(1+A)^2}E' - \frac{A}{A+1}Q_\lambda + \frac{2A}{(1+A)^2}\mu E'\sqrt{1-\frac{A+1}{A}\frac{Q_\lambda}{E'}}, \tag{2.87}$$

where E' is the energy before collision in the laboratory system, μ is the cosine of the scattering angle in the center-of-mass system, Q_λ is $-Q$ value of reaction for level λ, and A is the atomic mass of scattering isotope. The threshold energy of the reaction is

$$E_\lambda = \frac{A+1}{A}Q_\lambda. \tag{2.88}$$

A rigorous evaluation of the group-to-group transfer probability accounting for the energy-angle correlation of Eq. (2.87) is quite complicated. In the approximate treatment, the following two assumptions are made to simplify the situation without much loss in accuracy: (1) angle of scattering is fixed at the angle corresponding to the average cosine and (2)

$$\left(1-\frac{E_\lambda}{E'}\right)^{1/2} \approx 1-\frac{1}{2}\frac{E_\lambda}{E'}. \tag{2.89}$$

With these assumptions, Eq. (2.87) can be written as

$$E = \frac{1+A^2+2A\bar{\mu}}{(1+A)^2}E' - \frac{A+\bar{\mu}}{A+1}Q_\lambda, \tag{2.90a}$$

$$E' = \frac{(1+A)^2E + (A+\bar{\mu})(A+1)Q_\lambda}{1+A^2+2A\bar{\mu}}E. \tag{2.90b}$$

where $\bar{\mu}$ is the average cosine of the scattering angle. From Eq. (2.90), it is clear that for a given group j, a part of which lies above the threshold (i.e., $E_{j-1} > E_\lambda$), the probability of scattering from group j to group k for a discrete level λ is identically zero, unless group k lies between the energy boundaries

$$\frac{1+A^2+2A\bar{\mu}}{(1+A)^2}E_{j-1} - \frac{A+\bar{\mu}}{A+1}Q_\lambda, \tag{2.91a}$$

$$\frac{1+A^2+2A\bar{\mu}}{(1+A^2)}\max(E_j, E_\lambda) - \frac{A+\bar{\mu}}{A+1}Q_\lambda. \tag{2.91b}$$

For group k which falls partially or totally within this range, the probability of scattering from group j to group k is given by the fractional part of group j which scatters into group k.

$$P_\lambda(j \to k) = \frac{E^*_{j-1} - E^*_j}{E_{j-1} - E_j}, \tag{2.92}$$

where

$$E^*_{j-1} = \min\left[E_{j-1}, \frac{(1+A^2)E_{k-1} + (A+1)(A+\bar{\mu})Q_\lambda}{1+A^2+2A\bar{\mu}}\right], \tag{2.93a}$$

$$E^*_j = \max\left[E_j, E_\lambda, \frac{(1+A^2)E_k + (A+1)(A+\bar{\mu})Q_\lambda}{1+A^2+2A\bar{\mu}}\right]. \tag{2.93b}$$

Equations (2.95), (2.96) and (2.98) are used to calculate the discrete scattering source

$$S^{disc}_g = \sum_{g'} \phi_{g'} \left[\sum_\lambda \sigma_{in,g'} P^{in}_\lambda(g' \to g) + \sum_{\lambda'} \sigma_{n2n,g'} P^{n2n}_{\lambda'}(g' \to g)\right]. \tag{2.94}$$

The average cosine of the scattering angle is taken to be zero, i.e., isotropic in the center-of-mass system, for scattering from all (n,2n) levels.

This approximation is good for source energies far above the threshold where the level cross sections σ^λ_g are large. For standard fast reactor configurations, it has been found that the discrete inelastic calculation using the approximate method is quite accurate for heavy nuclides as shown in Figure 2.13 and significantly faster than the rigorous one which accounts explicitly for the energy-angle correlation of discrete level scattering. As shown in Figure 2.14, however, it is not so accurate for the intermediate mass nuclides such as iron.

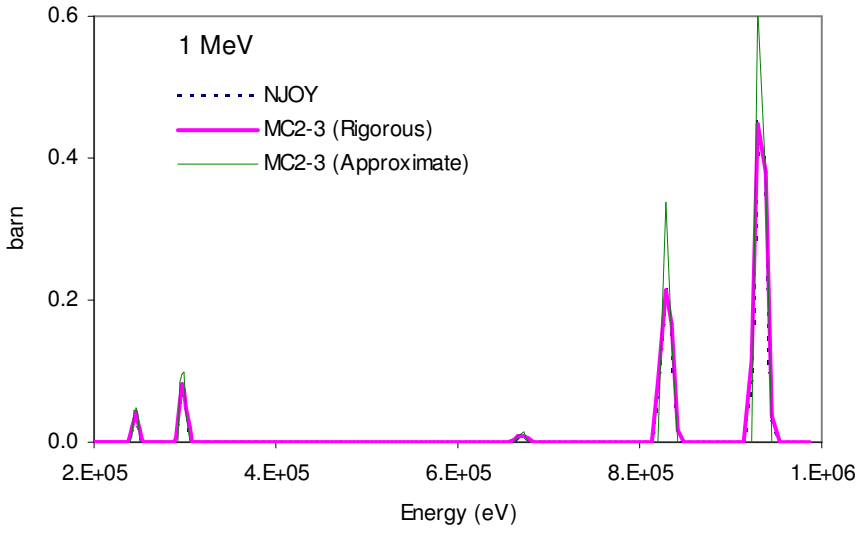


Figure 2.13 Inelastic Scattering Matrices of U-238 from NJOY and MC^2-3

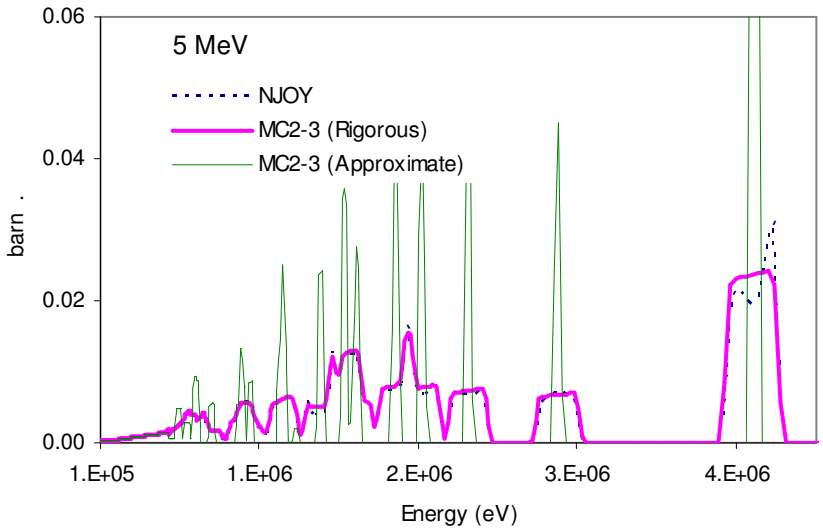


Figure 2.14 Inelastic Scattering Matrices of Fe-56 from NJOY and MC^2-3

2.6.3.2 Rigorous Treatment

The rigorous treatment accounts explicitly for the fact that a neutron of an incident energy E' scatters into a band of energies defined by Eq. (2.87) as

$$\left[\frac{1\pm A\sqrt{1-E_\lambda/E'}}{1+A}\right]^2 E'. \tag{2.95}$$

This band of energies must be used to define the possible sink groups. Similarly, a neutron into energy E' may be scattered from a range of source energies defined by

$$\left[\frac{1+A\sqrt{1+\frac{A-1}{A+1}\frac{E_\lambda}{E'}}}{1-A}\right]^2 E \quad \text{and} \quad \max\left\{E_\lambda^{(c)}, \left[\frac{1-A\sqrt{1+\frac{A-1}{A+1}\frac{E_\lambda}{E'}}}{1-A}\right]^2 E\right\}, \tag{2.96}$$

where $E_\lambda^{(c)} = E_\lambda A^2/(A^2-1)$ is a pseudo-threshold energy defined to avoid the need to consider the double valued nature of E' for a given E. Accounting for the energy bands in Eqs. (2.95) and (2.96) leads to four domains of integration in evaluating the probability of scattering from group j to group k as

$$P_\lambda(j\to k) = \frac{1}{E_{j-1}-E_j}\int_j dE'\int_k dE\sum_n (2n+1)\frac{(1-E_\lambda/E')^{-1/2}}{(1-\alpha)E'} f_n(E')P_n(\mu). \tag{2.97}$$

Eq. (2.97) is evaluated analytically taking proper account of the four domains of integration [18]. Because of the time consuming nature of the calculation, the equation is at the moment solved assuming isotropic scattering in the center-of-mass system. As compared in Figure

2.13 and Figure 2.14, the rigorous method produces group-to-group transfer matrices very accurately.

Recent efforts have been made to produce P_N inelastic scattering matrices using Eq. (2.97) considering anisotropic scattering both in the laboratory and center-of-mass systems such that external inelastic scattering cross section files are not necessary. However, those routines are not fully ready at this time.

2.7 One-Dimensional Transport Calculation

2.7.1 Collision Probability Method

The collision probability methods used in MC^2-3 are based on the RABANL algorithms of MC^2-2. The RABANL algorithms are improvements over earlier codes: RABID [19] for slab geometry and RABBLE [20] for cylindrical geometry.

2.7.1.1 Slab Geometry

In the slab geometry configuration in which the cross section and neutron sources are a function of x only, the current at x for a given energy group can be written in terms of the exponential integral E_2 as

$$J(x)=\frac{1}{2}\int dx'S(x')E_2[\Sigma_t(x')|x-x'|], \tag{2.98}$$

where $S(x')$ is the source at x' and $\Sigma_t(x')$ is the total macroscopic cross section at x'.

The neutron source S includes the source due to slowing down from all other groups and the within-group source. Thus the current at τ_m mean free paths beyond a source slab of optical thickness τ_1 (e.g., the current at the left surface of slab 2 in Figure 2.15 due to the source in slab 1) can be represented as

$$J(\tau_m,\tau_1)=\frac{1}{2}\int_0^{t_1} dx'S(x')E_2[\tau_m+\Sigma_t x'], \tag{2.99}$$

where τ_m is the optical thickness from the right surface of slab 1 to the left surface of slab 2 and t_1 is the thickness of slab 1.

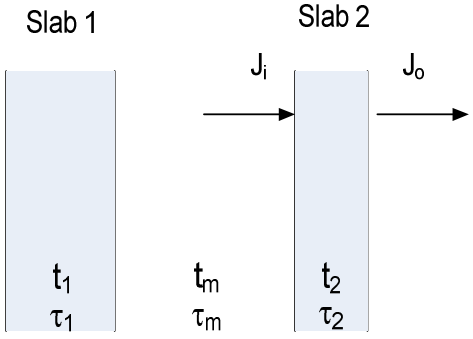


Figure 2.15 Slab Geometry

The collision rate in a slab is the difference between the un-collided current into the slab and the un-collided current out of the slab. Thus, when slab 2 lies to the right of slab 1 and they are separated by τ_m mean free paths, the collision rate in slab 2 of optical thickness τ_2 due to the source in slab 1 of optical thickness τ_1 is given as

$$C_R(1\to 2)_{right} = J(\tau_m,\tau_1) - J(\tau_m+\tau_2,\tau_1). \tag{2.100}$$

Here the subscript *right* implies that slab 2 lies to the right of slab 1. Assuming that the spatial source per unit length in the slab is given by a linear function

$$S(x') = \frac{1}{t}\left[\bar{S} + \frac{\Delta S}{t}\left(\frac{t}{2} - x'\right)\right], \tag{2.101}$$

Eq. (2.99) becomes

$$J(\tau_m,\tau_1) = \frac{1}{2}\int_0^{t_1} dx' \frac{1}{t_1}\left[\bar{S}_1 + \frac{\Delta S_1}{t_1}\left(\frac{t_1}{2} - x'\right)\right] E_2[\tau_m + \Sigma_t x']. \tag{2.102}$$

By evaluating the integrals analytically, Eq. (2.102) can be written as

$$\begin{aligned} J(\tau_m,\tau_1) = {} & \frac{\bar{S}_1}{2\tau_1}\left[E_3(\tau_m) - E_3(\tau_m+\tau_1)\right] \\ & + \frac{\Delta S_1}{2\tau_1}\left\{\frac{1}{2}\left[E_3(\tau_m) + E_3(\tau_m+\tau_1)\right] - \frac{1}{\tau_1}\left[E_4(\tau_m) - E_4(\tau_m+\tau_1)\right]\right\}. \end{aligned} \tag{2.103}$$

If there is an infinite array of unit cells, then the collision rate in all type 2 slabs which lie to the right of type 1 slabs is given by the infinite sum

$$C_R^{\infty}(1\to 2)_{right} = \sum_{m=0}^{\infty}\left[J(\tau_m + m\Delta,\tau_1) - J(\tau_m+\tau_2+m\Delta,\tau_1)\right], \tag{2.104}$$

where Δ is the optical thickness of the unit cell. Since neutrons proceed both to the right and to the left directions, the total collision rate in type 2 slabs due to sources in type 1 slabs is given by the sum of those two directional collision rates

$$C_R^{\infty}(1\to 2) = C_R^{\infty}(1\to 2)_{right} + C_R^{\infty}(1\to 2)_{left}. \tag{2.105}$$

Thus, for the neutron source S_i in a slab mesh interval i, the collision probability from slab i to slab j is given by

$$C_{i\to j} = C_R^{\infty}(i\to j)/S_i. \tag{2.106}$$

The unit cell may have either periodic or reflective boundary conditions as specified by the user. If the unit cell consists of N slab mesh intervals, then the collision probability can be represented by an $N\times N$ square matrix. Similarly, the collision rate, the total group source, and the within-group or self-scattering sources are all N dimensional vectors. Using matrix notation, the following neutron balance equation can be written as

$$\mathbf{C}_R = \mathbf{C}\cdot\left[\mathbf{S}_t + \mathbf{S}_s\right], \tag{2.107}$$

where $\mathbf{S}_t$ is the total group source including fission, slowing-down, and external sources, and $\mathbf{S}_s$ are the within-group scattering source. The within-group scattering source $\mathbf{S}_s$ can be represented by $\mathbf{R}\cdot\mathbf{C}_R$ where $\mathbf{R}$ is an $N\times N$ diagonal matrix. Thus, Eq. (2.107) can be rewritten as

$$\mathbf{C}_R = \left[\mathbf{I} - \mathbf{C}\cdot\mathbf{R}\right]^{-1}\cdot\mathbf{C}\cdot\mathbf{S}_t, \tag{2.108}$$

where $\mathbf{I}$ is the $N\times N$ identity matrix.

The solution of Eq. (2.108) requires the inversion of an $N\times N$ matrix. If self-scattering is ignored so that $\mathbf{R}$ is the null matrix, then it reduces to

$$\mathbf{C}_R = \mathbf{C}\cdot\mathbf{S}_t. \tag{2.109}$$

The neutron scalar flux for each mesh interval at given energy group can now be calculated by

$$\phi_i = C_{R,i}/\Sigma_i, \tag{2.110}$$

where Σ_i is the total macroscopic cross section of a slab region i.

2.7.1.2 Cylindrical Geometry

The flux at the surface of any cylindrical mesh interval is assumed to be isotropic so that neutron currents at the interface vary as the cosine of the incident angle (i.e., the cosine current approximation is used). In this case, the collision rate in a mesh interval needs to be related only to the neutron currents impinging on its inner and outer surfaces so that the collision rate is dependent only upon the adjacent mesh intervals.

If mesh interval 1 is the center interval and isotropic return is assumed as the outer boundary condition for the cylindrical unit cell (i.e., with the white boundary condition), then for an N cylindrical mesh interval unit cell shown in Figure 2.16, we have neutron balance equations for each group as

$$J_i^+ = \begin{cases} T_i^{OO}J_i^- + P_i^+S_i & i=1 \\ T_i^{OI}J_{i-1}^+ + T_i^{OO}J_i^- + P_i^+S_i & i=2,...,N \end{cases} \tag{2.111a}$$

$$J_i^- = \begin{cases} T_i^{OI}J_{i+1}^- + P_{i+1}^-S_{i+1} & i=1,...,N-1 \\ J_i^+ & i=N \end{cases} \tag{2.111b}$$

where

J_i^+ = the current impinging on the inner surface of the $(i+1)$-th cylindrical interval (in the increasing radial direction),

J_i^- = the current impinging on the outer surface of the i-th cylindrical interval (in the decreasing radial direction),

P_i^+ = the probability of escape through the outer surface of interval i due to a flat volume source,

P_i^- = the probability of escape through the inner surface of interval i due to a flat volume source,

T_i^{OI} = the transmission probability from the inner to the outer surface of interval i,

T_i^{IO} = the transmission probability from the outer to the inner surface of interval i,

T_i^{OO} = the transmission probability from the outer to the outer surface of interval i,

S_i = the flat volumetric source in cylindrical mesh i.

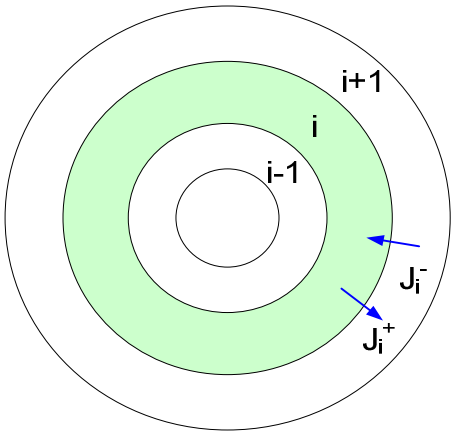


Figure 2.16 Cylindrical Geometry

If we define the column vector $\mathbf{J}$ having $2N$ elements $\{J_1^+, J_1^-, J_2^+, J_2^-, J_3^+, \ldots, J_N^-\}$ and the column vector $\mathbf{P}_S$ having $2N$ elements $\{P_1^+S_1, P_2^-S_2, P_2^+S_2, P_3^-S_3, \ldots, P_N^-S_N, P_N^+S_N, 0\}$, then Eq. (2.111) may be represented in matrix notation as

$$\mathbf{T}\,\mathbf{J} = \mathbf{P}_S, \tag{2.112}$$

where $\mathbf{T}$ is a $2N \times 2N$ matrix whose elements involve T_i^{OO}, T_i^{OO}, and T_i^{OO}.

For a cylindrical region i having outer radius r_i, inner radius r_{i-1}, and macroscopic cross section Σ_i, the transmission probabilities T_i^{OI} and T_i^{OO} are given by

$$T_i^{OI} = \frac{4}{\pi}\int_0^{\pi/2} d\phi \cos\phi\, K_{i3}[\Sigma_i f_i\ (\phi)]\,, \tag{2.113a}$$

$$T_i^{OO} = \frac{4}{\pi}\int_0^{\pi/2} d\phi \cos\phi \left\{ K_{i3}[\Sigma_i g_i\ (\phi)] - \frac{r_{i-1}}{r_i} K_{i3}[\Sigma_i h_i\ (\phi)] \right\}, \tag{2.113b}$$

where $K_{i3}[\xi] = \int_0^{\pi/2} e^{-\xi \csc\theta} \sin^2\theta d\theta$,

$$f_i\ (\phi) = -r_{i-1}\cos\phi + \left[r_i^2 - r_{i-1}^2 \sin^2\phi \right]^{1/2},$$

$$g_i\ (\phi) = 2r_i\ \cos\phi,$$

$$h_i\ (\phi) = 2\left[r_i^2 - r_{i-1}^2 \sin^2\phi \right]^{1/2}.$$

The Bickley function $K_{i3}[\xi]$ is evaluated for $0 < \xi \le 39$ using the rational Chebyshev approximation given by Gargantine and Pomentale [21]. When ξ is greater than 39, the function returns zero. The other quantities in the $\mathbf{T}$ matrix are determined as

$$T_i^{IO} = r_{i-1} T_i^{OI} / r_i\,, \tag{2.114a}$$

$$P_i^- = \frac{r_{i-1}(1 - T_i^{OI})}{2(r_i^2 - r_{i-1}^2)\Sigma_i}, \tag{2.114b}$$

$$P_i^+ = \frac{r_i\ (1 - T_i^{OO} - T_i^{OI})}{2(r_i^2 - r_{i-1}^2)\Sigma_i}. \tag{2.114c}$$

Since $\mathbf{T}$ is a tri-diagonal matrix, Eq. (2.112) may be solved for J_i^+ and J_i^- by the method of forward elimination and backward substitution. Having solved for J_i^+ and J_i^-, the collision rates for each interval can be obtained from neutron balance as

$$C_{R,i} = \begin{cases} S_i + J_i^- - J_i^+, & i = 1 \\ S_i + J_i^- - J_i^+ + J_{i-1}^+ - J_{i-1}^-, & i = N \end{cases}. \tag{2.115}$$

If the volumetric source S_i in Eq. (2.111) does not include the within-group scattering source, Eq. (2.115) provides the collision rate in each mesh i. When the within-group source is included, the collision probabilities must be calculated to determine the collision rate in each mesh. The collision probabilities $C_{i\to j}$ can be determined by solving Eq. (2.111) for $S_i = 1$ and $S_j = 0$ for all $j \ne i$, that is,

$$C_{i\to j} = C_{R,j} \quad (\text{for } S_j = \delta_{ji}). \tag{2.116}$$

This procedure is repeated for each interval to complete the evaluation of $C_{i\to j}$. Having obtained $C_{i\to j}$, the collision rates are calculated by

$$\mathbf{C}_R = \left[\mathbf{I} - \mathbf{C}\cdot\mathbf{R}\right]^{-1}\cdot\mathbf{C}\cdot\mathbf{S}_t \,. \tag{2.117}$$

Here $\mathbf{R}$ is the matrix that yields the within-group scattering source by operating on the collision rate as in the case of slab geometry.

Similarly to the slab geometry, the neutron scalar flux for each group is calculated by

$$\phi_i = C_{R,i} / \Sigma_i \,, \tag{2.118}$$

where Σ_i is the total macroscopic cross section of a cylindrical region i.

2.7.2 Geometry Effect of Resonance Self-Shielding

When the one-dimensional transport calculation is performed based on ultrafine groups, the heterogeneity effect on resonance self-shielding is taken into account in terms of the effective background cross sections (i.e., using equivalence theory). For the hyperfine group calculation, pointwise cross sections are directly used, and thus there is no need to account for the self-shielding effect except for the unresolved resonance region, where it is not possible to define precise values for the cross sections of resonance reactions. As discussed in Section 2.3, the cross sections in the unresolved resonance range are self-shielded on the ultrafine group basis. Therefore, it is important to estimate the effective background cross section accurately for generating multigroup cross sections in both the ultrafine and hyperfine group calculations.

Equivalence theory is the basis for the Bondarenko method. Equivalence theory originated from a heterogeneous two-region cell composed of a moderator and a fuel region, and shows that the heterogeneous cell can be made equivalent to a homogeneous mixture under the narrow resonance approximation and the rational approximation for the collision probabilities. Since the absorber region is limited geometrically to the fuel region and energetically to the resonance region, a neutron can escape resonance absorption leaving this region either geometrically or energetically (scattering collision in the narrow resonance approximation). The equivalence relation consists in simulating the geometrical escape by the addition of a fictitious scattering cross section (energetic escape).

In a general heterogeneous problem, the escape probability of a region i can be written in one-term rational function form as [22]

$$P_{e,i}(E) = \sum_{j\neq i} P_{ij}(E) = \frac{\Sigma_{ei}^*(E)}{\Sigma_{ti}(E) + \Sigma_{ei}^*(E)}, \tag{2.119}$$

where P_{ij} is the collision probability that a neutron originating in region i makes its next collision in region j, Σ_{ti} is the total cross section of region i, and Σ_{ei}^* is the effective escape cross section of region i. In terms of the conventional Dancoff and Bell-Levine factors, the effective escape cross section can be written as

$$\Sigma_{ei}^*(E) = \Sigma_{ei}\frac{a_i(E)[1-C_i(E)]}{1+[a_i(E)-1]C_i(E)}, \tag{2.120}$$

where Σ_{ei} is the escape cross section defined by the inverse of average chord length $\bar{\ell}_i$, of region i, i.e., $\Sigma_{ei} = 1/\bar{\ell}_i = A_i/4V_i$ with volume V_i and surface area A_i. The Dancoff factor C_i is the probability that the neutrons escaped from region i to the other regions would come back to region i without suffering their first collision in the other regions. The Bell-Levine factor a_i is used to improve the accuracy of Wigner's rational approximation for the escape probability.

Once the collision probabilities are known as a function of neutron energy, the effective escape cross section can be determined using Eq. (2.120) as

$$\Sigma_{ei}^{*}(E) = \Sigma_{ti}(E)\left[\frac{1}{1-P_{e,i}(E)} - 1\right] = \Sigma_{ti}(E)\left[\frac{1}{P_{ii}(E)} - 1\right]. \tag{2.121}$$

Accurate evaluation of energy-dependent collision probabilities in the resonance region requires hyperfine group calculations with heterogeneous geometries, which negate the ultrafine group calculation and associate resonance self-shielding. For the purpose of resonance self-shielding, therefore, an alternative way was developed based on ultrafine group CPM calculations. The collision probabilities were approximately determined in the ultrafine group level by solving a unit cell problem using infinite-dilute ultrafine group total cross sections, and the effective escape cross sections were estimated by

$$\Sigma_{ei,g}^{*} = \Sigma_{ti,g}\left(\frac{1}{P_{ii,g}} - 1\right), \tag{2.122}$$

where the subscripts g denotes a ultrafine group. Test results for the multi-region slab and cylindrical problems indicated that the resulting self-shielded ultrafine group cross sections are accurate when each resonant isotope is included only in a single region. However, when the same resonant isotope is present in more than one region, the results were no longer satisfactory because the infinite-dilute ultrafine group cross section of a resonant isotope yield overestimated values for the collision probability that a neutron originating in a region of interest, containing the resonant isotope, makes its next collision in another region that includes the same resonant isotope.

The flux in a region i can be written in terms of collision probabilities by applying the NR approximation

$$\phi_i(E) = \frac{1}{E}\frac{\sum_j P_{ji}(E)\Sigma_{pj}V_j}{\sum_j P_{ji}(E)\Sigma_{tj}(E)V_j}, \tag{2.123}$$

Assuming that $P_{ji}(E)/\Sigma_{ti}(E)$ can be represented as a function of energy only in a region i, i.e., $P_{ji}(E)/\Sigma_{ti}(E) = \alpha_i(E)P_{ji}^{g}/\Sigma_{ti}^{g}$ [23] and separating out the resonant isotope r, the above equation is simplified as

$$\phi_i(E) = \frac{1}{E}\frac{\sum_j P_{ji}^{g}\Sigma_{pj}V_j}{\sum_j P_{ji}^{g}\Sigma_{tj}(E)V_j} = \frac{1}{E}\frac{C_i^{r}}{\Sigma_{ti}(E)+\Sigma_{ei}^{r}(E)}, \tag{2.124}$$

where $\Sigma_{ei}^{r}(E)=\dfrac{\sum_j\sum_{k\neq r}P_{ji}^{g}\Sigma_{tj}^{k}(E)V_j}{\sum_j P_{ji}^{g}N_j^{r}V_j/N_i^{r}}-\sum_{k\neq r}\Sigma_{ti}^{k}(E)$ and $C_i^{r}=\dfrac{\sum_j P_{ji}^{g}\Sigma_{pj}V_j}{\sum_j P_{ji}^{g}N_j^{r}V_j/N_i^{r}}$.

As shown in the equation above, the escape cross section $\Sigma_{ei}^{r}(E)$ varies with resonant isotopes in each region. The equation also indicates that the resonant isotopes which are present only in a region i have the same escape cross section because it is determined independently of the resonant isotopes of interest. In the code, the escape cross section $\Sigma_{ei}^{r}(E)$ is approximated as a constant Σ_{ei}^{rg} within an ultrafine group.

Although the hyperfine group calculation can be used for accurate resonance self-shielding, the new self-shielding approach is still beneficial since the unresolved resonance self-shielding is carried out on the ultrafine group basis.

2.8 2D and 3D Method of Characteristics Transport Calculations

2.8.1 2D/3D MOC Formulation

A full 3D MOC method requires significant memory and computational resources. The 2D/1D method [26], which uses a 2D MOC for the radial direction and a pin-based homogenization in the axial direction, can inherently have numerical instability as well as accuracy loss as the axial node size becomes small or a region becomes optically very thin (almost void). To overcome the drawbacks of the full 3D MOC and 2D/1D methods, the 2D/3D MOC method, which combines the radial 2D MOC with the discontinuous Galerkin finite element method in the axial direction, was developed and implemented in PROTEUS-MOC [27] under the DOE NEAMS program. The 2D/3D method has been implemented in MC^2-3 as well. This 2-D/3-D MOC solver can be used in solving 2-D problems by imposing the reflective boundary conditions at the top and bottom of the 3-D domain. However, solving the 2-D/3-D MOC equation is much more expensive than solving the 2-D MOC equation directly. Therefore, along with the 2-D/3-D solver, a dedicated 2-D MOC solver was developed separately and implemented in MC^2-3.

In the 2D/3D MOC formulation, the derivation of 2D/3D MOC formulation starts from the one-group neutron transport equation for a discretized angle:

$$\left(\mu_m\frac{\partial}{\partial x}+\eta_m\frac{\partial}{\partial y}+\xi_m\frac{\partial}{\partial z}\right)\varphi_i^{m}(x,y,z)+\Sigma_i\varphi_i^{m}(x,y,z)=Q_i^{m}(x,y,z), \tag{2.125}$$

where m and i are indices for angle and mesh, respectively. The solution of each 3D mesh is represented with the polynomial basis functions. The 3D geometry is represented as multiple 2D planar planes extruded axially so that every 2D plane has the same radial meshes. On the axial interface, the angular flux is constructed from its polynomial representation and thus the neutron transmission can be incorporated due to the conformal mesh property of the 3D extruded geometry. Therefore, the 2D/3D MOC method is able to account for the mesh-to-mesh leakage in the axial direction with almost no accuracy loss.

The angular flux and associated source term are expressed using the linear basis functions as:

$$\varphi_i(x,y,z,\Omega) \approx \sum_{j=1}^{2} \varphi_i^j(x,y,\Omega) u_i^j(z), \tag{2.126}$$

where $u_i^j(z)$ is the *j*-th basis function. In the 2D/3D MOC solver implemented in MC^2-3, the following orthogonal basis functions are used:

$$u_i^1(z) = 1, \tag{2.127}$$

$$u_i^2(z) = \frac{2}{\Delta_i}\left(z - \frac{(z_i^- + z_i^+)}{2}\right), \tag{2.128}$$

By inserting Eqs. (2.127) thru (2.129) into Eq. (2.126) and applying the discontinuous Galerkin method, the neutron transport can be casted into an analogy form of the 2D MOC equation with matrix-vector representation. For a polar direction in the upper hemisphere of angular domain, the 2D/3D MOC equation is written as:

$$\left(\Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y}\right)\Psi_i(x,y,\Omega) + \boldsymbol{\Sigma}_i \Psi_i(\mathrm{x},\mathrm{y},\Omega) = S_i, \tag{2.129}$$

where Ψ_i is the angular flux vector that consists of the φ_i^1 and φ_i^2 appearing in Eq. (2.127). The cross section matrix, $\boldsymbol{\Sigma}_i$, and the source vector, $\mathbf{S}_i$, can be obtained as:

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma_i + \Omega_z/\Delta_i & \Omega_z/\Delta_i \\ -3\Omega_z/\Delta_i & \Sigma_i + 3\Omega_z/\Delta_i \end{bmatrix}, \tag{2.130}$$

$$\mathbf{S}_i = \begin{bmatrix} q_i^1(x,y,\Omega_z) + \Omega_z \varphi_i^-(x,y,\Omega_z)/\Delta_i \\ q_i^2(x,y,\Omega_z) - 3\Omega_z \varphi_i^-(x,y,\Omega_z)/\Delta_i \end{bmatrix}, \tag{2.131}$$

where φ_i^- is the incoming angular flux at the bottom of the mesh and q_i^j is the *j*-th basis function contribution to the fission and scattering source.

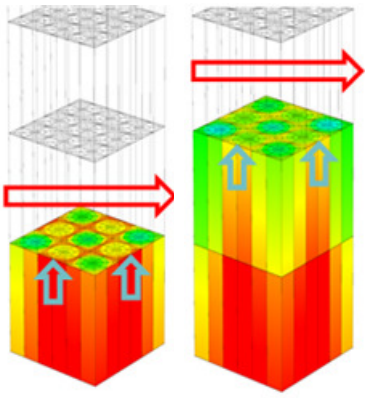


Figure 2.17 Solution Flow of the 2D/3D MOC Method

The 2D MOC solver solves Eq. (2.130) for each 2D plane with the incoming angular fluxes at the axial interfaces shown in Eq. (2.132). Since these incoming angular fluxes are available from the solution of adjacent planes, the 2D planar problems can be sequentially solved from bottom to top or top to bottom, depending on the angular flux direction, as illustrated in Figure 2.17.

2.8.2 Ray Tracing Module for Hexagonal Geometries

The 2D/3D MOC solver employs the 2D ray-tracing module to model a heterogeneous configuration in the radial direction. The primary application target of the 3D transport calculation is a typical sodium-cooled fast reactor (SFR) fuel pin or assembly. The 2D ray-tracing module was implemented to handle a 2D domain consisting of hexagonal cells in triangular lattice. It was further extended to the BFS-type critical assembly illustrated in Figure 2.18. The 2D/3D MOC solver is able to handle rectangular geometries as well, including the ZPR/ZPPR unit cells which are composed of many plates with irregular sizes.

Figure 2.19 shows the hexagonal cells supported by the 2D ray-tracing module. Each hexagonal cell is divided into hexagonal or annular meshes. The steel wire inserted in the interstitial space can be modeled explicitly for the BFS [28] unit cell calculation. The 2D ray-tracing module can handle hexagonal assemblies in the triangular lattice as well. The fuel rod and duct structure and inter-assembly gap can be represented explicitly as shown in Figure 2.20. The modular ray tracing technique was adopted for an effective memory usage. All of the supporting geometries can be easily constructed by assembling the hierarchical input keywords so that a user can create the hexagonal geometry core of interest with ease.

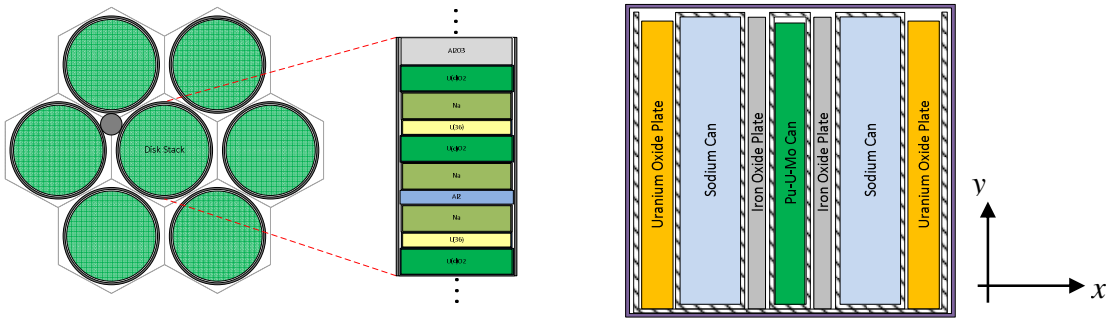


Figure 2.18 Illustration of BFS Unit Assembly and ZPR/ZPPR Fuel Drawer

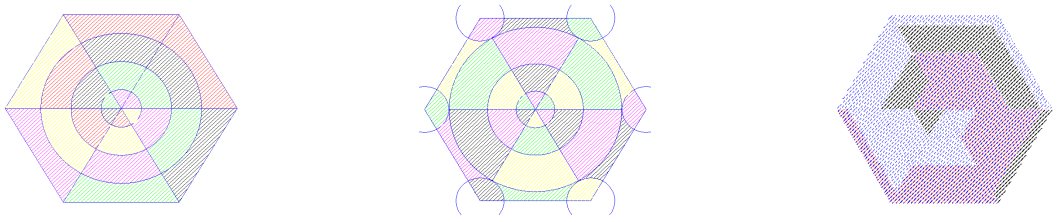


Figure 2.19 Hexagonal Cell Models in 2D Ray Tracing Module

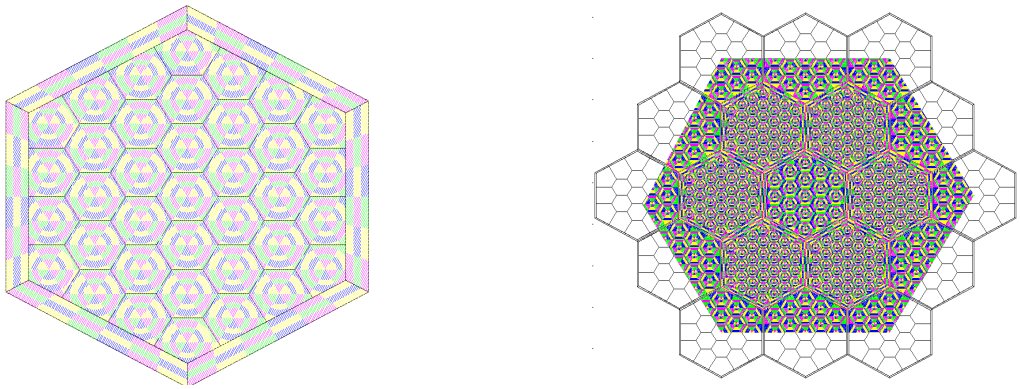


Figure 2.20 Illustration of Typical SFR Assembly Modeling

2.8.3 CMFD Acceleration

The 2D/3D MOC solver of MC^2-3 was initially developed for performing a unit cell calculation for hexagonal unit cells in the triangular lattice, being later extended to handle typical SFR core configurations. Since the 2D/3D MOC solver requires significant computational time compared to the CPM solver, the CMFD acceleration technique was implemented to the 2D/3D MOC to improve the computational performance.

The CMFD acceleration performs the diffusion-type calculations based on the coarse mesh structure. In typical SFR core configurations, hexagonal assemblies or mostly hexagonal pin cells can be used as the coarse meshes. A module for handling the coarse mesh geometry was implemented to the 3D transport solver to provide the geometric data, such as mesh connectivity, which are needed to construct the linear systems for the CMFD acceleration. Along with the coarse mesh geometry, the neutron current information at the coarse mesh interfaces is required to determinate the CMFD coupling coefficients. A radial surface current at the assembly or pin interfaces can be computed during the MOC calculation by using the following equation:

$$J_i^s = \sum_{k\in s}\sum_m \varphi_{m,\mathrm{i}}^{k,out} \sin\theta_m w_m \Delta_m - \sum_{k\in s}\sum_m \varphi_{m,i}^{k,in} \sin\theta_m w_m \Delta_m , \tag{2.132}$$

where m, k, i and s are the indices for angle, ray, coarse mesh and surface respectively, φ is the angular flux, w is the weight of angular quadrature, and Δ is the ray spacing. An axial surface current can be easily computed using the mesh-averaged angular flux distribution, which are obtained as an output of the MOC calculation for each plane. The neutron current tally routines were added to the MOC module.

In order to have the CMFD solution accelerate the 2D/3D MOC calculation, the following quantities are needed: the homogenized group constants for coarse meshes and the coupling coefficients that specify interface current relations between the two adjacent coarse meshes. From the 2D/3D MOC solution, the homogenized group constants can be generated by the flux-volume weighting over the fine meshes that belong to the coarse mesh. The coupling coefficients can be determined using the following CMFD relation:

$$J_{i\to k} = -\tilde{D}_{i\to k}(\bar{\phi}_k - \bar{\phi}_i) - \hat{D}_{i\to k}(\bar{\phi}_k + \bar{\phi}_i), \tag{2.133}$$

where i and j are indices for the coarse meshes that are connected through a common interface. $\bar{\phi}_i$ and $\bar{\phi}_j$ are the averaged fluxes of the coarse meshes i and j, respectively, and $J_{i\to k}$ is the surface current from the coarse mesh i to the coarse mesh j. $\tilde{D}$ is the coupling coefficient obtained from the conventional finite difference diffusion formulation, and $\hat{D}$ are the current correction factor to preserve the interface current obtained from the MOC solution.

The average flux and surface current can be readily obtained from the 2D/3D MOC solution. The CMFD relation in Eq. (2.134) can be solved for the current correction factor as:

$$\hat{D}_{i\to k} = -\frac{J_{i\to k} + \tilde{D}_{i\to k}(\bar{\phi}_k - \bar{\phi}_i)}{(\bar{\phi}_k + \bar{\phi}_i)}. \tag{2.134}$$

Using the surface currents obtained from the 2D/3D MOC solution in Eq. (2.135), the current correction factors can be determined. Under the current continuity condition, the current correction factors satisfy the following relation:

$$\hat{D}_{i\to k} = -\hat{D}_{k\to i}, \tag{2.135}$$

In the 2D/3D MOC method, the angular flux and current can be discontinuous at the axial mesh interfaces due to the discontinuous Galerkin formulation for the treatment of axial variable. In order to ensure the CMFD current relations consistent with the 2D/3D MOC solution, the discontinuity of axial current is inevitably allowed, and consequently, Eq. (2.136) does not hold for the axial current correction factors. Therefore, when the axial current correction factor of the coarse mesh i toward the coarse mesh k, $\hat{D}_{i\to k}$, is determined, the current value yielded from the axial angular flux distribution of the coarse mesh i should be used. By applying the neutron balance equation for each coarse mesh together with the CMFD relation given in Eq. (2.135) and the homogenized group constants, a set of linear equations can be formed in terms of averaged fluxes and neutron sources.

Since the weighted residual technique applied to the axial variable in the 2D/3D MOC method yields the non-conservative discretization, the solution of 2D/3D MOC method does not satisfy the neutron balance equation for fine meshes and coarse meshes as well. In order to resolve the inconsistency of the neutron balance between the CMFD and 2D/3D MOC solutions, a fictitious cross section, referred to as a pseudo absorption cross section, is introduced in the CMFD neutron balance equation as: [29]

$$\sum_k J_{i\to k} + (\Sigma_r^i + \Sigma_p^i)\bar{\phi}_i V_i = \bar{q}_i V_i, \tag{2.136}$$

where Σ_p is the pseudo absorption cross section. The pseudo absorption cross section of each coarse mesh can be determined using the 2D/3D MOC solution as:

$$\Sigma_p^i = \frac{\bar{q}_i V_i - \sum_k J_{i\to k}}{\bar{\phi}_i V_i} - \Sigma_r^i, \tag{2.137}$$

The homogenized group constants, current correction factors and pseudo absorption cross sections are iteratively updated during the iterative solution process because those parameters are a function of the 2D/3D MOC solution.

In the 2D/3D MOC calculation with CMFD acceleration, the CMFD problem is formed with the partially converged transport solution. The CMFD solution is obtained via the conventional fission source iteration and each group equation can be effectively solved using the well-established iterative method. Based on a coarse mesh flux ratio of the previous MOC and newly obtained CMFD solutions, fine-mesh fluxes within the coarse mesh are updated. The updated fine-mesh solution together with the eigenvalue of CMFD result is put to the subsequent 2D/3D MOC calculation. In the CMFD calculation, the global flux distribution can be effectively obtained since the boundary information can be promptly propagated to the interior of problem domain. By incorporating the CMFD solution, therefore, the number of fission source iterations in the 2D/3D MOC calculation can be considerably reduced.

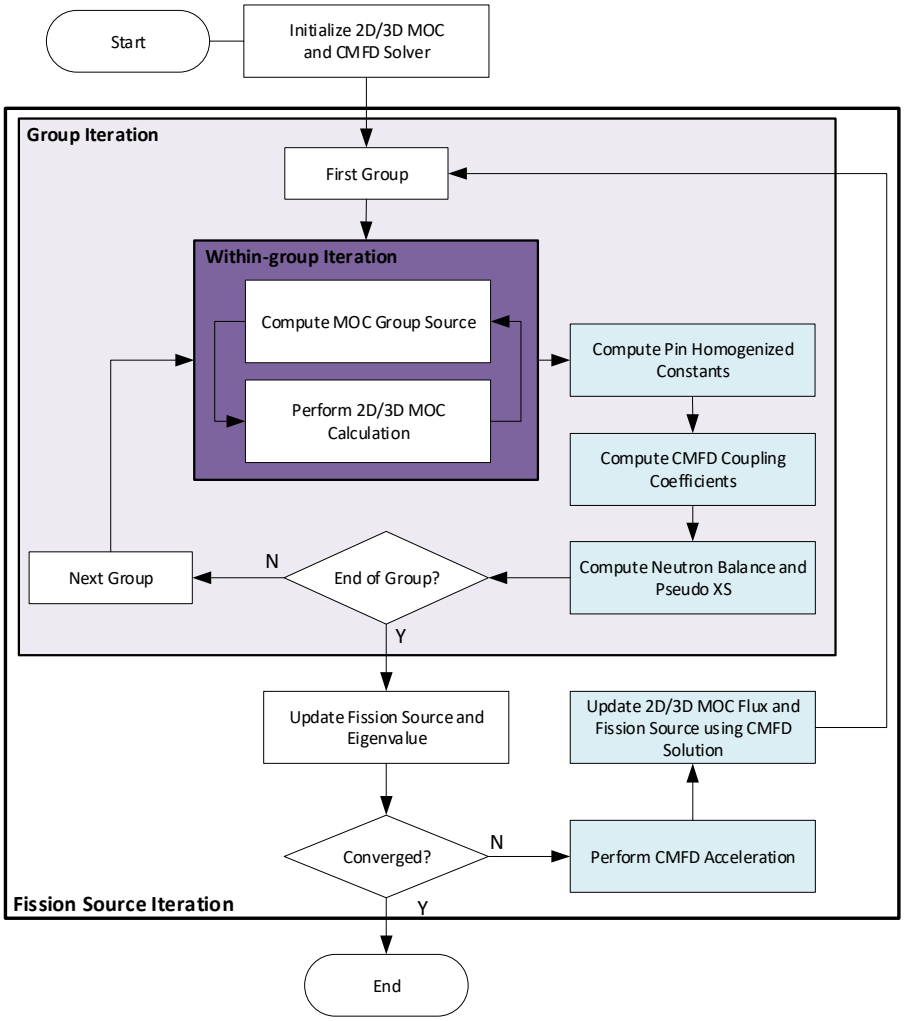


Figure 2.21 Computational Flow of the 2D/3D MOC Solver with the CMFD Acceleration

The overall calculation procedure is illustrated in Figure 2.21. The three layers of nested iterations are performed in the calculation: fission source, group and within-group iterations. At the end of each fission source iteration which is the outermost iteration, the CMFD calculation is performed and then the fine-mesh flux and fission source distributions are updated using the resulting CMFD solution. Using this, the fully converged coarse mesh solution is obtained at each CMFD calculation since its computational cost is negligible compared to the single 2D/3D MOC calculation. In the group iteration, the coarse mesh parameters required for constructing the CMFD problem are computed using the updated group solutions. In the innermost iteration which is a within-group iteration, the 2D/3D MOC calculation is performed repeatedly with the within-group scattering source update.

2.8.4 Self-shielding Calculation

The 2D/3D MOC solver was implemented into the existing framework of MC^2-3 to perform the UFG transport calculation for a 3D hexagonal geometry. If the UFG cross sections for all regions are prepared, the transport calculation for the 3D domain is performed with the source iteration. The transport source iteration, which is to update the fission source distribution as well as eigenvalue, was implemented in the 3D transport solver. For given the within-group source and cross sections, the 2D/3D MOC solver comes into play to solve the with-in group transport problem. To account for the anisotropic scattering effect, the anisotropic scattering source up to P_1 is incorporated in the 3D transport solver.

In the multi-dimensional transport calculation of MC^2-3, the equivalence theory is employed to account for the heterogeneity effect to the UFG cross sections. The existing procedure for the escape cross sections was extended to a 3D domain. The two fixed-source transport problems presented in Section 2.7.3 are solved using the 3D transport solver.

The UFG cross sections are determined by self-shielding the pointwise cross sections using the NR approximation (material background cross sections) and the escape cross sections. In addition, the interface routines that connect the data structures of MC^2-3 and the 3D solver were implemented so that the existing routines for processing the cross sections, i.e. homogenization and group condensation, can be utilized without modification.

2.9 Fission Source Data

For each ultrafine group, the number of neutrons per fission is derived from the expression

$$\nu_i^g = \frac{1}{E_{g-1} - E_g} \int_{E_g}^{E_{g-1}} \nu_i(E) dE, \tag{2.138}$$

with

$$\nu_i(E) = A_o^i + A_1^i E + A_2^i E^2 + A_3^i E^3. \tag{2.139}$$

In MC^2-2, the dependency of the fission spectrum on the incident energy is neglected and the fission spectrum vector is determined for a given incident neutron energy by

$$\chi_i^g = \frac{\int_{E_g}^{E_{g-1}} dE \chi_i(E)}{\int_{E_{\min}}^{E_{\max}} dE \chi_i(E)} = \frac{\hat{\chi}_i^g}{\sum_{g'} \hat{\chi}_i^{g'}}, \tag{2.140}$$

with

$$\chi_i(E) = \alpha_i \frac{E}{\tau_i^2} e^{-E/\tau_i} + (1-\alpha_i)\sqrt{\frac{4E}{\pi\beta_i^3}} e^{-E/\beta_i} \quad . \tag{2.141}$$

This combination of a Maxwellian and an evaporation spectrum suffices to describe most of the fission spectra of the ENDF/B data files. Integration of Eq. (2.128) gives

$$\begin{aligned} \hat{\chi}_i^g = {} & \alpha_i \left[\left(1+\frac{E_g}{\tau_i}\right) e^{-E_g/\tau_i} - \left(1+\frac{E_{g-1}}{\tau_i}\right) e^{-E_{g-1}/\tau_i} \right] \\ & + (1-\alpha_i) \left[\operatorname{erf}\sqrt{\frac{E_{g-1}}{\beta_i}} - \operatorname{erf}\sqrt{\frac{E_g}{\beta_i}} - \sqrt{\frac{4E_{g-1}}{\pi\beta_i}} e^{-E_{g-1}/\beta_i} + \sqrt{\frac{4E_g}{\pi\beta_i}} e^{-E_g/\beta_i} \right]. \end{aligned} \tag{2.142}$$

The temperature parameters τ_i and β_i are energy dependent, but they are not energy dependent in the MC^2-2 formulation.

In MC^2-3, the ultrafine group fission spectrum matrix is processed directly from the ENDF/B data and a separate file is given for each isotope. As an example, Figure 2.22 shows the fission spectra of U-238 at different incident neutron energies. A noticeable dependency on the incident neutron energy is observed. Another option is to produce the ultrafine group fission spectrum vector using a typical fast reactor spectrum in the GROUPR module of NJOY. In fact, both options do not make big differences, but MC^2-3 primarily uses the fission spectrum matrix.

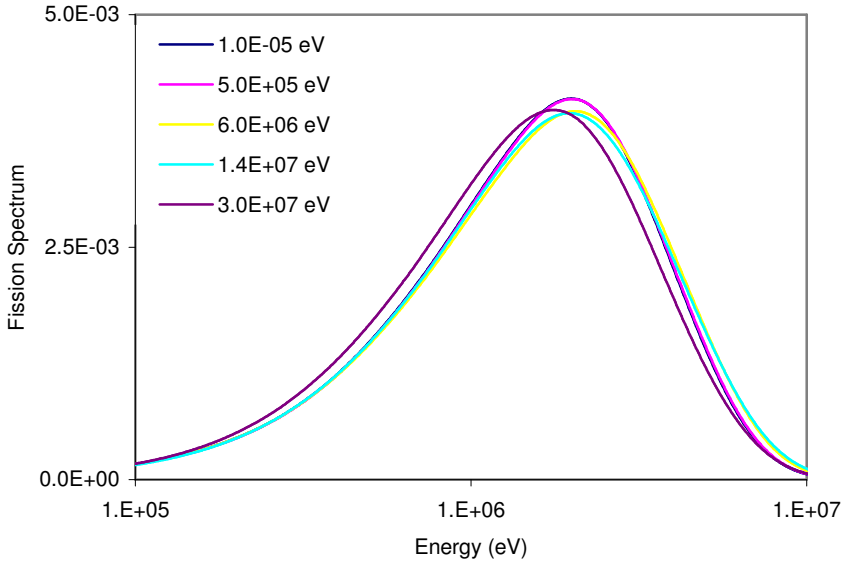


Figure 2.22 Fission Spectrum Data of U-238 at Different Incident Neutron Energies

2.10 Group Condensation

The isotopic microscopic broad group cross sections are obtained by averaging the ultrafine group data with the flux and current spectra determined by solving the ultrafine group equations. The broad group structure is specified by the user, but the broad group energy boundaries must be defined as a subset of the ultrafine group boundaries. The broad group cross section data are written in the ISOTXS format.

The ultrafine group data of absorption cross sections are averaged over the flux spectrum as

$$\sigma_G^x = \frac{\sum_{g\in G} \sigma_g^x \phi_g}{\phi_G}, \quad \phi_G = \sum_{g\in G} \phi_g , \tag{2.143}$$

where x includes the six reactions (n,p), (n,d), (n,t), (n, f), (n, γ), and (n, α). The (n, α) cross section calculated from Eq. (2.131) is actually a sum of (n, He^3) and (n, He^4) cross sections since the ISOTXS format does not allow editing of both the partials.

The group condensation of the ultrafine group scattering transfer matrices are given as

$$\sigma_\ell (G' \to G) = \frac{\sum_{g'\in G'} \sum_{g\in G} \sigma_\ell (g' \to g) \phi_{\ell g'}}{\phi_{\ell G'}}, \quad \phi_{\ell G} = \sum_g \phi_{\ell g} . \tag{2.144}$$

The P_1 calculation is performed for the consistent P_1 spectrum option, and thus the ultrafine group current is calculated directly. The higher-order flux moments $\phi_{\ell g}$ for the extended transport approximation are calculated using the recursive relation

$$\phi_{\ell g} = \frac{\ell}{2\ell+1} \frac{|B|}{A_{\ell g}} \phi_{\ell-1,g}, \quad \ell > 1, \tag{2.145}$$

where $A_{\ell g}$ is the extended transport cross sections defined in the consistent P_N equation discussed in Section 2.1.

The flux moment weighted total cross sections are obtained by summing all of the partials discussed above,

$$\sigma_{\ell G}^{total} = \sigma_{\ell G}^{absorption} + \sigma_{\ell G}^{elastic} + \sigma_{\ell G}^{inelastic} + \sigma_{\ell G}^{n2n}, \quad l = 0,1,\cdots,N . \tag{2.146}$$

In analogy with the total cross section, moments of the transport cross sections are calculated if the order of the extended transport approximation is greater than unity. The transport cross section algorithms are defined to conserve the relation,

$$\Sigma_{\ell G}^{tr} = \frac{\ell}{2\ell+1} |B| \frac{\phi_{\ell-1,G}}{\phi_{\ell G}} = \sum_i N^i \sigma_{\ell G}^{i,tr}, \quad \ell = 1,2,\cdots,N , \tag{2.147}$$

where i is an isotope in the composition of concern. The microscopic transport cross section is calculated as

$$\sigma_{\ell G}^{tr} = \sigma_{\ell G}^{t} - \sigma_{\ell G}^{s} . \tag{2.148}$$

Note that the Legendre expansion coefficient factor, $2\ell+1$, is not included in any of $\sigma_{\ell G}^{total}$, $\sigma_{\ell G}^{tr}$, and $\sigma_{\ell GG'}^{s}$.

The average number of neutrons per fission, ν, is given by

$$\nu_G = \frac{\sum_{g\in G} \nu_g \sigma_g^f \phi_g}{\sigma_G^f \phi_G}. \tag{2.149}$$

The isotopic fission spectrum distributions are derived by summing the ultrafine group vectors

$$\chi_G = \sum_{g\in G} \chi_g . \tag{2.150}$$

The MC^2-3 code provides the composition-wide fission spectrum vector which is calculated as

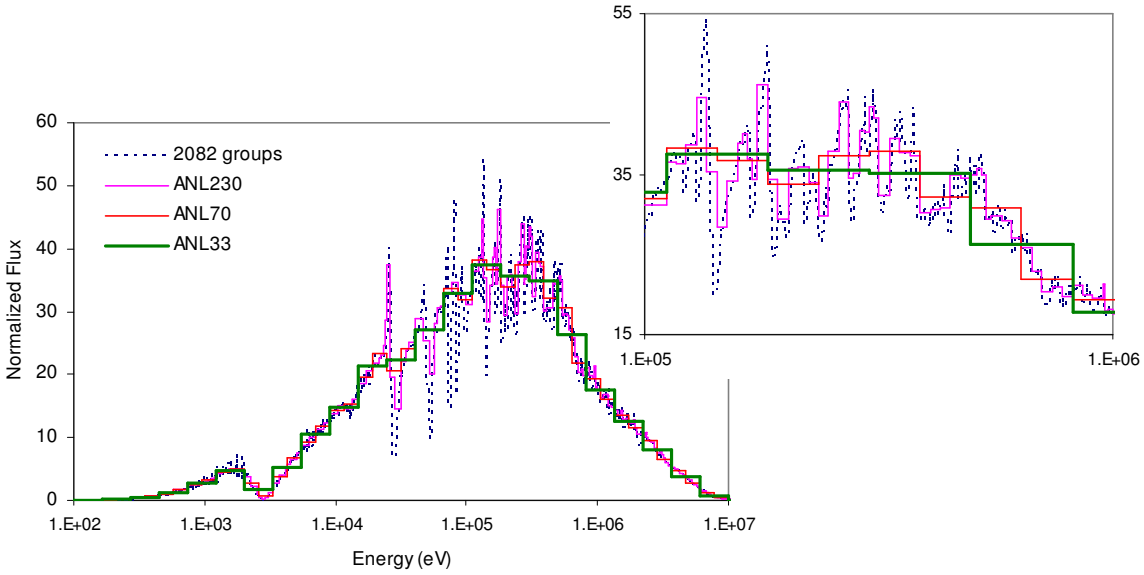
$$\chi_G^{comp} = \frac{\sum_{g\in G}\sum_{i\in comp} \chi_g^i S_f^i}{\sum_{i\in comp} S_f^i}, \tag{2.151}$$

where S_f^i is the fission source of isotope i which belongs to the composition, $comp$.

For the consistent P_N approximation which is used in many transport codes, the MC^2-3 code provides the corrected within-group scattering cross sections by default.

$$\sigma_{\ell GG}^{s'} = \sigma_{\ell GG}^{s} + (\sigma_{0G}^{t} - \sigma_{\ell G}^{t}) . \tag{2.152}$$

The code provides built-in broad group structures such as 33, 70, and 230 groups, as illustrated in Figure 2.23. More detailed data are given in Appendix D.



2082	1424	1148	871	595	319	43
230	215	196	168	141	88	15
70	48	38	29	20	11	2
33	25	20	15	11	6	2

Figure 2.23 Group Condensation Based on the Built-in Group Structures

2.11 Hyperfine Group Transport Calculation

The MC^2-3 code solves the one-dimensional hyperfine group transport equation for slab and cylindrical geometries using the collision probability method (CPM) and the method of characteristics (MOC). A MOC solver has been implemented for slab geometry only, and it is able to handle anisotropic scattering sources.

2.11.1 Solution Scheme

The one-dimensional hyperfine group calculation with anisotropic scattering source requires an impractically large computation time when all of the scattering and fission sources are explicitly evaluated for the hyperfine group level. The computation time is increased by several order of magnitude compared to the case of isotropic scattering where the scattering transfer matrix can be evaluated with only four pre-calculated probabilities. Although the number of spatial meshes is not substantial for one-dimensional problems, the floating point operations required for evaluating scattering transfer matrices with more than 350,000 energy groups (Δu = 1/24000, below 1 MeV) are substantial. Previously, the hyperfine group scattering transfer matrices are calculated on the fly during iterative slowing-down calculations to avoid the huge storage requirements. This increases the computational burden associated with the scattering transfer matrix evaluation further, because the increase in the number of groups generally slows down the convergence of the iterative solution of the underlying eigenvalue problem.

In order to make the hyperfine group calculation more practical, a hybrid approach was devised with iteration between ultrafine and hyperfine group calculations, as illustrated in

Figure 2.24. In this approach, a ultrafine group eigenvalue problem is first solved using the self-shielded ultrafine group cross sections based on the narrow resonance and equivalence theory approximations, and the converged ultrafine group fission and non-elastic scattering sources are saved. Then, by interpolating these ultrafine group sources, hyperfine group sources are determined. With the fission and non-elastic sources, a hyperfine group calculation is performed by explicitly evaluating elastic scattering sources, including anisotropic scattering sources. An eigenvalue problem is thus solved for the ultrafine group calculation in the first step, followed by the hyperfine-group fixed-source calculation as

$$\nabla \cdot \Omega \psi(r,E,\Omega) + \Sigma_t \psi(r,E,\Omega) = S_s^e(r,E,\Omega) + S_f(r,E) + S_s^{ne}(r,E,\Omega), \tag{2. 153}$$

where $S_s^e(r,E,\Omega)$, $S_f(r,E)$, and $S_s^{ne}(r,E,\Omega)$ are sources due to elastic scattering, fission, and non-elastic scattering. The multigroup formulation of the equation above becomes

$$iB\phi_1^g + \Sigma_{t0}^g \phi_0^g = \sum_{g' \neq g} \Sigma_{s0}^{gg'} \phi_0^{g'} + S_f^g + S_{ne,0}^g, \tag{2.154a}$$

$$\frac{1}{3} iB\phi_0^g + A_1^g \phi_1^g = \sum_{g' \neq g} \Sigma_{s1}^{gg'} \phi_1^{g'} + S_{ne,1}^g. \tag{2.142b}$$

Since this fixed source problem does not require the fission and slowing-down source iteration, the required computational time is much shorter than that for the corresponding eigenvalue problem. Finally, self-shielded ultrafine group cross sections are determined using the hyperfine group fluxes, and the ultrafine group calculation in the first step is repeated. The iteration between ultrafine and hyperfine group calculations can be repeated until the ultrafine group solution converges within the specified criterion, but various test problems indicated that the second ultrafine group calculation is generally adequate for convergence of the ultrafine group solution.

This iterative approach is based on the assumption that anisotropy of inelastic scattering does not make a significant impact on the local spectrum within an ultrafine group which is used in resonance self-shielding. In this approach, the hyperfine group solution is used to determine the self-shielded ultrafine group cross sections from the second ultrafine group calculation and thus the equivalence theory approximation is no longer needed.

In MC^2-2, the hyperfine group calculation module named RABNAL is used for correcting the targeted broad group cross sections in resolved resonance energy range where the NR approximation is not valid. Because of the isotropic scattering approximation in the center-of-mass system, the application of RABANL is restricted to the resolved resonance energy range of heavy isotopes (below 20 keV) where the scattering is almost isotropic in the center-of-mass system. The increase of upper energy for the RABANL calculation to the resonance range of intermediate weight isotopes causes a significant error in the spectrum solution. The error is mainly due to the increased anisotropy of elastic scattering with increasing neutron energy.

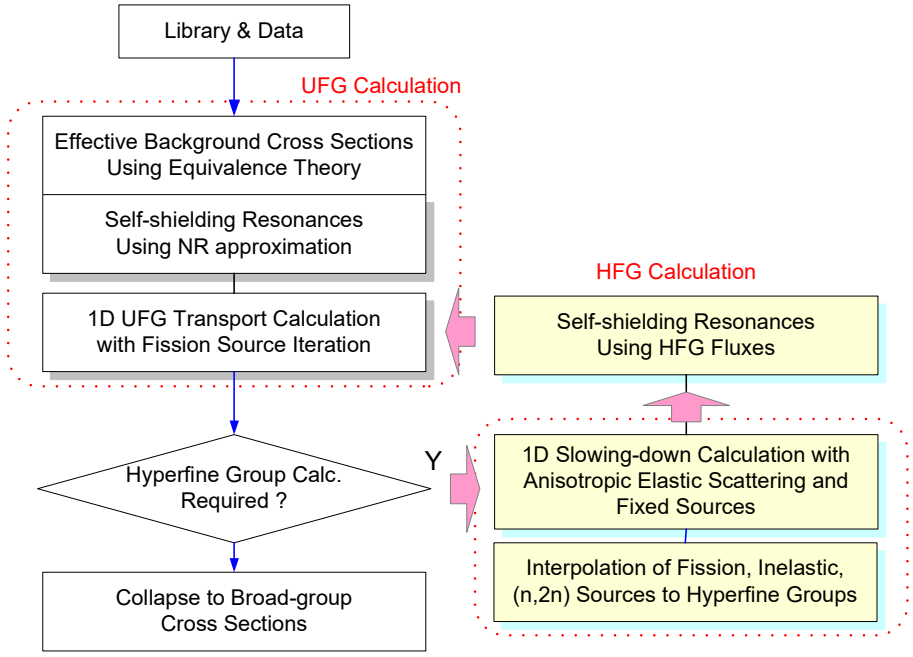


Figure 2.24 One-dimensional Hyperfine Group Calculation Scheme of MC^2-3

As an example to show the increased anisotropy with increasing energy, the energy transfer probabilities of carbon scattering for different incident neutron energies are compared in Figure 2.25. This problem is eliminated in the hyperfine-group module of MC^2-3 by considering the anisotropic elastic scattering in the center-of-mass system and solving the consistent P_1 equation with the extended transport approximation in the same manner as in the ultrafine-group transport calculation. As shown in Figure 2.26, the neutron spectra obtained with and without anisotropic elastic scattering show noticeable differences, especially in the high energy range over a few keV.

The cross sections and neutron spectrum within a hyperfine group can be approximated to be constant, and thus the transfer matrix of scattering order l from a source group g to a sink group g' can be written as

$$\sigma_{sl}(g \to g') \cong \int_{u_{g'-1}}^{u_{g'}^*} du' \int_{u_{g-1}^*}^{u_g} du \frac{\sigma_s(u)\, e^{-(u'-u)} P_l(\mu_s)}{(1-\alpha)} \sum_{n=1}^{N} (2n+1) f_n(u) P_n(\mu_c), \tag{2.155}$$

where f_n = Legendre expansion coefficients of scattering cross section in the ENDF/B data, μ_s = cosine of the scattering angle in the laboratory system, μ_c = cosine of the scattering angle in the center-of-mass system, P_l = Legendre polynomials of order l, σ_s = point-wise cross section reconstructed from the resonance parameters, and $u_{g'}^*$ = energetically reachable boundary.

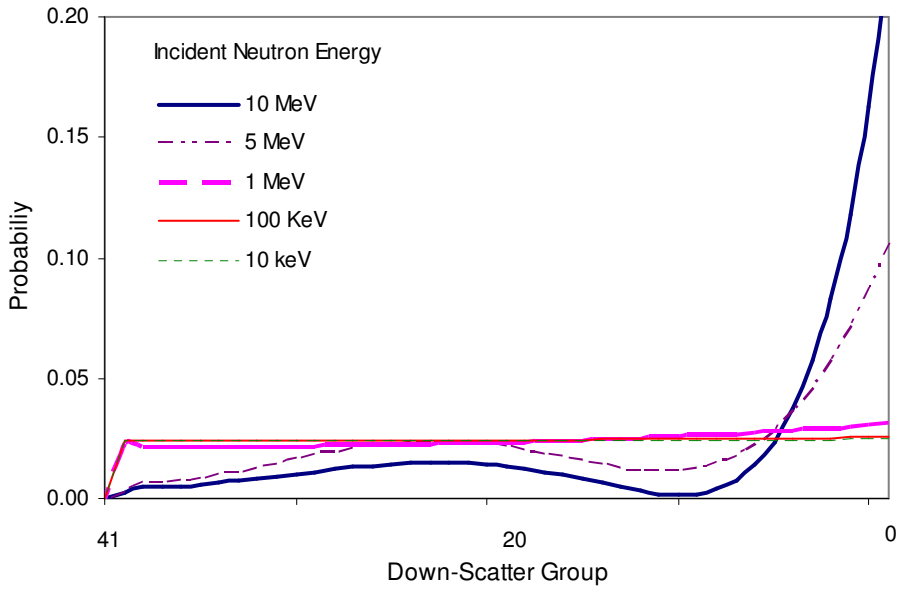


Figure 2.25 Elastic Scattering Probabilities of Neutron with Various Incident Energies at Carbon

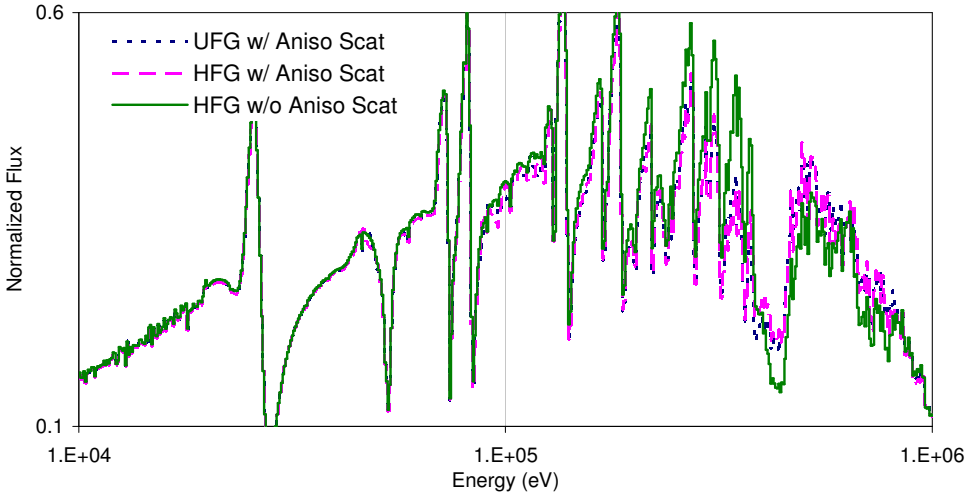


Figure 2.26 Ultrafine Group Spectra Obtained from Different Hyperfine Group Spectra

In MC^2-2, the hyperfine group width is determined relative to the resolved resonance Doppler width $\Delta = (4kTE/A)^{1/2}$ for temperature T, energy E, and mass A, where k is the Boltzmann constant. On the other hand, in MC^2-3, the hyperfine group width is set to be constant (typically a lethargy width of 1/24000) rather than evaluating the resolved resonance Doppler width for a mixture.

Ignoring the anisotropic scattering in the center-of-mass system, as in the RABANL module of MC^2-2, the evaluation of the scattering source in a hyperfine group k requires only four terms such as the source in the previous hyperfine group $S_{0,k-1}$ and the scattering rates $\Sigma_s\phi$ for hyperfine groups, $k-1$, $k-L$, and $k-L-1$, where L is the maximum number of down-scattering hyperfine groups. This makes it possible to save computational time and storage substantially. However, this simple approach is not applicable when anisotropic scattering in the center-of-mass system is introduced. In particular, for hydrogen which has an almost full scattering band, it is impractical to hold the huge scattering transfer matrix in memory. Therefore, during the flux calculation for a group g, the scattering transfer vector from the group g to all down-scattering groups g' is generated, and then the contributions of the scattering in group g to all the down-scattering group sources are computed. The scattering source of each group is accumulated, as the slowing-down calculation proceeds from the highest energy group to the lowest energy group. The other neutron sources from fission and non-elastic scattering are currently evaluated by interpolating the ultrafine group sources obtained from the ultrafine group slowing-down calculation.

2.11.2 Scattering Transfer Matrix

Considering a hyperfine group structure with a uniform lethargy width Δu, a neutron in an arbitrary group with lethargy interval $(u_0, u_0+\Delta u)$ can be scattered into the lethargy interval $(u_0, u_0+\Delta u+\varepsilon)$, where $\varepsilon = \ln(1/\alpha)$. Denoting the largest integer equal to or less than a real number a by $\text{int}(a)$, the number of next hyperfine groups into which a neutron in a hyperfine group is scattered down is given by

$$L = \begin{cases} \varepsilon/\Delta u, & f = 0 \\ \text{int}(\varepsilon/\Delta u)+1, & f > 0 \end{cases}, \tag{2.156}$$

where $f = \varepsilon/\Delta u - \text{int}(\varepsilon/\Delta u)$. Assuming the isotropic scattering in the center-of-mass system, the probability per unit lethargy for within-group scattering is given by

$$P_0\Delta u = \frac{1}{1-\alpha}\int_{u_0}^{u_0+\Delta u} du \int_{u}^{u_0+\Delta u} du' e^{-(u'-u)} = \frac{1}{1-\alpha}(\Delta u - 1 + e^{-\Delta u}). \tag{2.157}$$

The probability per unit lethargy of scattering down ℓ hyperfine groups for $\ell = 1, 2, \cdots, L-1$ is given by

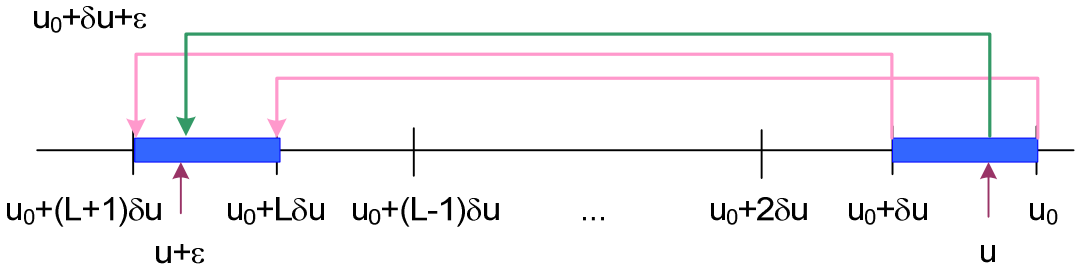
$$P_\ell\Delta u = \frac{1}{1-\alpha}\int_{u_0}^{u_0+\Delta u} du \int_{u_0+l\Delta u}^{u_0+(l+1)\Delta u} du' e^{-(u'-u)} = \frac{1}{1-\alpha}(1-e^{-\Delta u})^2 e^{-(\ell-1)\Delta u} = (P_1\Delta u)e^{-(\ell-1)\Delta u}. \tag{2.158}$$

When $f = 0$ and then $\varepsilon = L\Delta u$ (see Figure 2.27 (a)). Thus the probability of scattering into the lowest energy group can be determined as

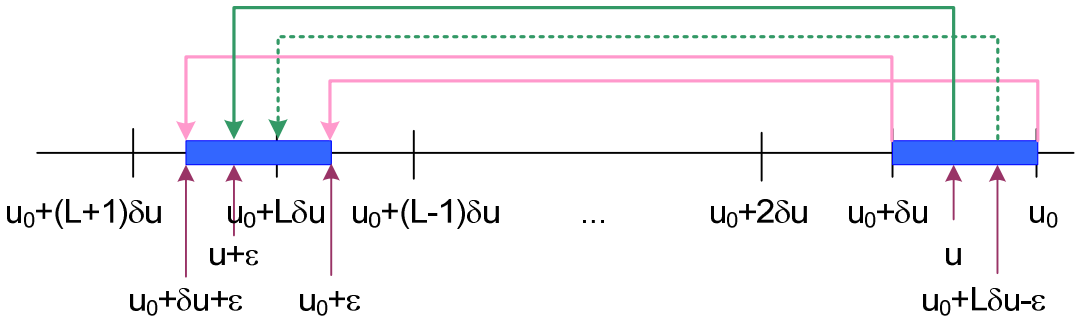
$$P_L \Delta u = \frac{1}{1-\alpha}\int_{u_0}^{u_0+\Delta u} du \int_{u_0+L\Delta u}^{u+\varepsilon} du' e^{-(u'-u)} = \frac{\alpha}{1-\alpha}(e^{\Delta u} - 1 - \Delta u). \tag{2.159}$$

Using the results above, it can easily be shown that the sum of probabilities is unity as

$$\sum_{\ell=0}^{L} P_\ell \, \Delta u = \frac{1}{1-\alpha}(\Delta u - 1 + e^{-\Delta u}) + \frac{1}{1-\alpha}(1-e^{-\Delta u})^2 \sum_{\ell=1}^{L-1} e^{-(\ell-1)\Delta u} + \frac{\alpha}{1-\alpha}(e^{\Delta u} - 1 - \Delta u) = \Delta u. \tag{2.160}$$



(a) $L = \varepsilon / \delta u$



(b) $L = \varepsilon / \delta u + (1 - f_h)$

Figure 2.27 Slowing Down by Elastic Scattering

If $f > 0$, then $\varepsilon = L\Delta u - (1-f)\Delta u$ (see Figure 2.27 (b)). In this case, the probabilities of scattering into the last two lowest energy groups are adjusted as

$$\begin{aligned} P_{L-1}\Delta u &= \frac{1}{1-\alpha}\left[\int_{u_0}^{u_0+L\Delta u-\varepsilon} du \int_{u_0+(L-1)\Delta u}^{u+\varepsilon} du' e^{-(u'-u)} + \int_{u_0+L\Delta u-\varepsilon}^{u_0+\Delta u} du \int_{u_0+(L-1)\Delta u}^{u+L\Delta u} du' e^{-(u'-u)}\right] \\ &= \frac{\alpha}{1-\alpha}\left[e^{(1+f)\Delta u} - 2e^{f\Delta u} + 1 - (1-f)\Delta u\right], \end{aligned} \tag{2.161a}$$

$$P_L \Delta u = \frac{1}{1-\alpha}\int_{u_0+L\Delta u-\varepsilon}^{u_0+\Delta u} du \int_{u_0+L\Delta u}^{u+\varepsilon} du' e^{-(u'-u)} = \frac{\alpha}{1-\alpha}\left[e^{f\Delta u} - 1 - f\Delta u\right]. \tag{2.149b}$$

Thus, the sum of probabilities becomes

$$\begin{aligned} \sum_{\ell=0}^{L} P_\ell \, \Delta u &= \frac{1}{1-\alpha}(\Delta u - 1 + e^{-\Delta u}) + \frac{1}{1-\alpha}(1-e^{-\Delta u})^2 \sum_{\ell=1}^{L-2} e^{-(\ell-1)\Delta u} \\ &+ \frac{\alpha}{1-\alpha}\left[\left\{e^{(1+f)\Delta u} - 2e^{f\Delta u} + 1 - (1-f)\Delta u\right\} + \left(e^{f\Delta u} - 1 - f\Delta u\right)\right] = \Delta u. \end{aligned} \tag{2.162}$$

It is noted that for $f = 0$, Eq. (2.149b) becomes zero and Eq. (2.149a) reduces to Eq. (2.147).

The scattering probability depends on the scattering isotopes, but not on the incident neutron energy as far as the hyperfine group width is constant. As a result, the source of neutron per unit lethargy into group g' due to the isotropic elastic scattering of isotope i in group g can be determined as

$$\begin{aligned} S^i_{g\to g'} &= \frac{N_i}{\Delta u}\int_{u_{g'-1}}^{u_{g'-1}+\Delta u} du' \int_{u_{g-1}}^{u_{g-1}+\Delta u} du\, P_i\ (u \to u')\, \sigma^i_s(u)\, \phi(u) \\ &\approx \frac{N_i}{\Delta u}\sigma^i_{sg}\ \phi_g \int_{u_{g'-1}}^{u_{g'-1}+\Delta u} du' \int_{u_{g-1}}^{u_{g-1}+\Delta u} du\, P_i\ (u \to u') = P^i_{g'\to g}(\Sigma^i_s\phi)_g, \end{aligned} \tag{2.163}$$

where N_i is the atom number density of isotope i, σ^i_{sg} is the elastic scattering cross section of isotope i in group g, and ϕ_g is the neutron flux per unit lethargy in group g.

The scattering-in source per unit lethargy into group g due to elastic scattering from all energetically possible lower lethargy hyperfine groups is given by

$$S^i_g = N_i \sum_{\ell=1}^{L} \sigma^i_{s,g-\ell}\phi_{g-\ell}P^i_\ell = P^i_1 \sum_{\ell=1}^{L-2} e^{-(\ell-1)\Delta u}(\Sigma^i_s\phi)_{g-\ell} + P^i_{L-1}(\Sigma^i_s\phi)_{g-L+1} + P^i_L(\Sigma^i_s\phi)_{g-L}. \tag{2.164}$$

Similarly, the within-group scattering source is given by

$$S^i_{og} = N_i\, \sigma^i_{sg}\phi_g P^i_0 = P^i_0(\Sigma^i_s\phi)_g. \tag{2.165}$$

The scattering-in source into group g can be written as

$$\begin{aligned} S^i_g &= e^{-\Delta u}S^i_{g-1} + P^i_1(\Sigma^i_s\phi)_{k-1} + (P^i_{L-1} - P^i_{L-2}e^{-\Delta u})(\Sigma^i_s\phi)_{g-L+1} + (P^i_L - P^i_{L-1}e^{-\Delta u})(\Sigma^i_s\phi)_{g-L} \\ &\quad - P^i_L e^{-\Delta u}(\Sigma^i_s\phi)_{g-L-1}. \end{aligned} \tag{2.166}$$

For a mixture of various mass nuclides, the scattering sources are obtained by summing up the contributions from the constituents of the composition in question as

$$S_g = \sum_i S^i_g, \tag{2.167a}$$

$$S_{0g} = \sum_i P^i_0 \Sigma^i_{sg}\phi_g = P_0\, \Sigma_{sg}\phi_g. \tag{2.155b}$$

Thus, in a homogeneous infinite medium without leakage, the neutron flux in group g is simply determined as

$$\phi_g = \frac{S_g}{\Sigma_{tg} - \sum_i P^i_0 \Sigma^i_{sg}}. \tag{2.168}$$

When the anisotropic scattering in the center-of-mass system is included, the elastic scattering transfer cross sections are evaluated using the methods described in Section 2.5. As discussed in Section 2.10.1, the group-to-group scattering probability is calculated on the fly as the slowing-down calculation proceeds from the highest energy group to the lowest energy group,

and the scattering source of each group is accumulated by adding the contribution from the scattering in the current group.

2.12 Two-Dimensional Transport Calculation

In general, a homogeneous or one-dimensional unit cell calculation with critical buckling search is adequate for determining the broad group cross sections of fuel assemblies. However, the neutron spectrum in non-fueled assemblies (e.g., reflector and shield) is mainly determined by slowing-down of the neutrons leaking out of neighboring assemblies, and thus the neutron spectrum obtained from a unit cell calculation results in non-negligible errors in the broad group cross sections, in particular when a small number of broad groups is used. As a result, a large number of broad groups need to be used for accurate whole-core calculations. For example, more than a thousand groups are required to model the severe spectral transition at the core and reflector interface accurately using the broad group cross sections generated with unit cell calculations. As shown in Figure 2.28, the effective multiplication factor converges very slowly with an increasing number of energy groups.

In order to improve modeling of the region-to-region leakage in actual core configurations in the generation of broad group cross sections, MC^2-3 provides two-dimensional, ultrafine group transport calculation capability by using the TWODANT code [6]. As shown in Figure 2.29, MC^2-3 calculates the ultrafine group cross sections for each region and provides the region-wise, ultrafine group macroscopic cross sections for TWODANT which performs a two-dimensional transport calculation with a user-specified core model in RZ geometry. TWODANT sends the transport solution to MC^2-3 in the form of RZMFLX, which is a non-standard CCCC interface file for regular zone-averaged moment fluxes. Using the ultrafine group spectra of flux moments in RZMFLX, MC^2-3 calculates the microscopic broad group cross sections for each region in the user-specified broad group structure.

For a given scattering order n, there are $(2n+1)$ components of spherical harmonics. Thus, using the n-th moment of angular flux, the n-th moment of scattering cross section from a broad group G' to G can be defined to preserve the scattering source moment as

$$\Sigma^n_{G'\to G} = \frac{\sum_{g\in G}\sum_{g'\in G'} \Sigma^n_{g'\to g} \sum_{m=-n}^{n} \phi^{g'}_{n,m} Y_{n,m}(\Omega)}{\sum_{m=-n}^{n} \phi^{G'}_{n,m} Y_{n,m}(\Omega)}, \tag{2.169}$$

where n is a scattering order and $-n \le m \le n$. This definition yields direction-dependent scattering cross sections, which is undesirable for core calculations.

To avoid this problem, group condensation can be carried out with individual components of the angular flux moments as

$$\Sigma^n_{G'\to G} = \frac{\sum_{g\in G}\sum_{g'\in G'} \Sigma^n_{g'\to g} \phi^{g'}_{n,m}}{\phi^{G'}_{n,m}}. \tag{2.170}$$

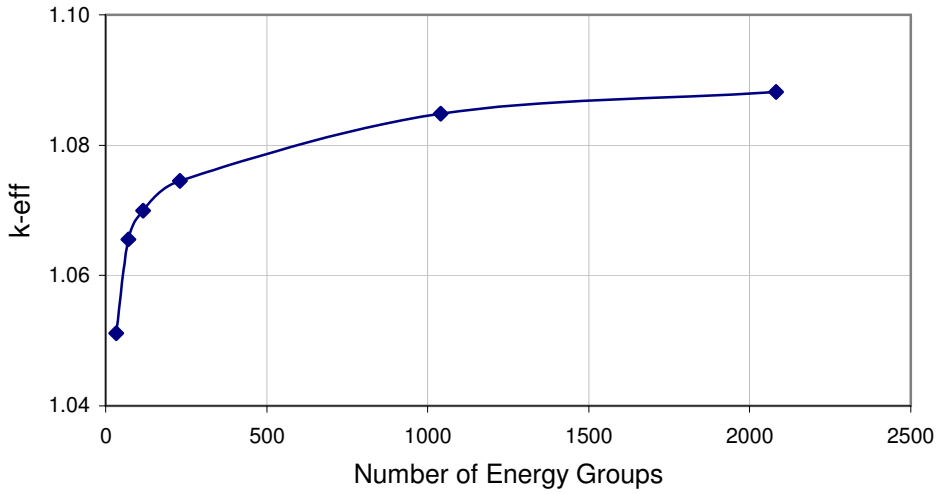


Figure 2.28 Convergence of Effective Multiplication Factor of the Core/Reflector Problem

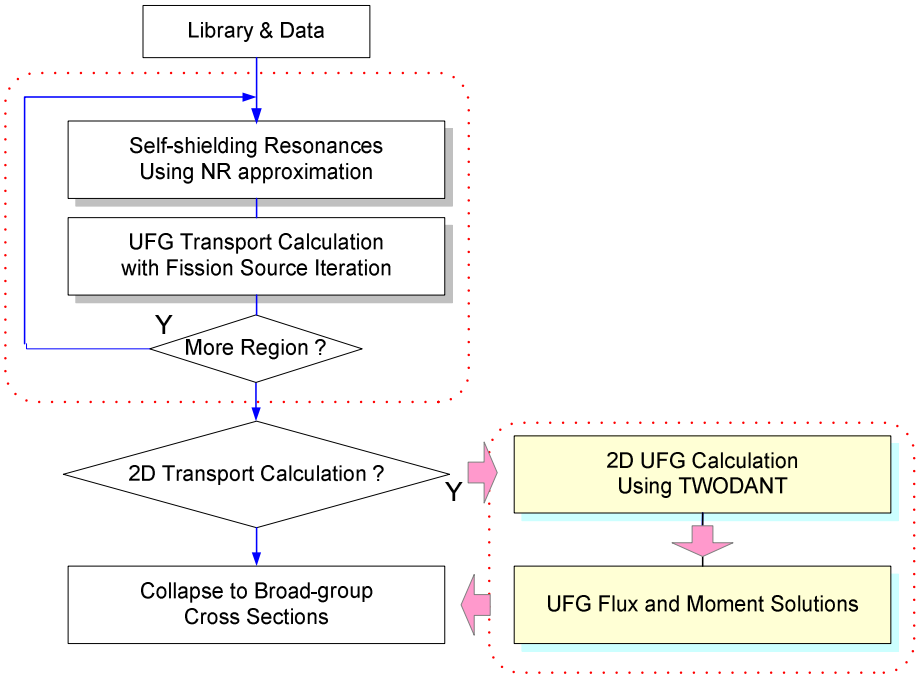


Figure 2.29 Group Condensation of MC^2-3 Using Two-Dimensional Transport Solutions

However, there are $2n+1$ components, and thus the n-th order scattering cross section moment cannot be defined uniquely. Therefore, in MC^2-3, higher-order scattering moments are condensed using the sum of $\left|\phi_{g'}^{n,m}\right|$ as

$$\sigma_{G'\to G}^{n}=\frac{\sum_{g\in G}\sum_{g'\in G'}\sigma_{g'\to g}^{n}\left[\sum_{m=-n}^{n}\left|\phi_{g'}^{n,m}\right|\right]}{\sum_{g'\in G'}\left[\sum_{m=-n}^{n}\left|\phi_{g'}^{n,m}\right|\right]}. \tag{2.171}$$

2.13 Gamma Data Generation

In the conventional procedure, the SDX code reads gamma production and yield matrices from LAPGAS and delayed fractional fission yield matrices from DLYFIS, and generates cell-averaged or region-averaged multigroup gamma production matrices to store them in PMATRX. The code produces gamma interaction matrices to write them on GAMISO as well, which is based on the ISOTXS format. For neutron group G and gamma group GG, the cell average gamma production and yield matrices can be defined as

$$P_r^i(G\to GG)=\frac{1}{\phi_{r,G}}\sum_{g\in G}\phi_{rg}\sigma_g^i[Y_i^P(g\to GG)+Y_i^D(g\to GG)], \tag{2.172}$$

where σ_g^i is the intermediate-group self-shielded cross section (absorption and inelastic scattering cross sections); $Y_i\ (g\to GG)$ is the prompt fractional yield matrix for a neutron in energy group g to produce gamma rays in energy group GG; $Y_i^D(g\to GG)$ is the delayed fractional fission yield matrix; $\phi_{r,G}$ are the flux of region r for broad group G. The gamma production yields vary smoothly over energy group and thus the change of the gamma production cross sections over group are mainly dependent upon that of the self-shielded cross sections.

The MC^2-3 code (Version 3.1 and higher) is now able to process the gamma library to produce PMATRX and GAMISO for the user-specified composition. In the new procedure using NJOY/MC^2-3, as illustrated in Figure 2.30, the gamma libraries are generated based on the ultrafine neutron group structure and thus the intermediate groups required in the old procedure no longer needs to be introduced.

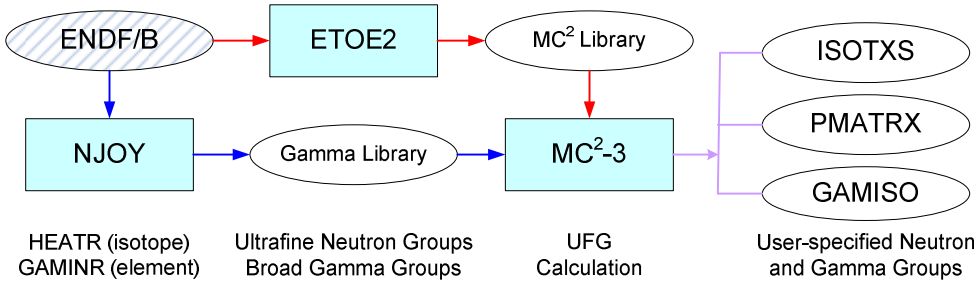


Figure 2.30 Neutron and Gamma Data Generation Procedure

In NJOY, the gamma production matrices and neutron interaction cross sections are processed at the infinite dilute condition and 300K. Therefore, those production matrices for the capture and fission reactions are converted to yield matrices by dividing the gamma production cross section by the neutron cross section. In MC^2-3, those yield matrices are multiplied with the self-shielded neutron interaction cross section to form the self-shielded gamma production matrices. Since, however, no resonance self-shielding is necessary for the inelastic scattering reaction, the gamma production matrices are not converted to yield matrices. In short, in MC^2-3, the gamma yield matrices are used for the absorption (capture and fission) reactions and the gamma production matrices are directly employed for the inelastic reaction. The spectrum effect is accounted for based on the ultrafine group calculation in the code.

The neutron and gamma heating data are stored in PMATRX. A KERMA (Kinetic Energy Release in Materials) factor is defined as an averaged kinetic energy release per reaction times a cross section:

$$k_{ij}(E) = \sum_l \overline{E}_{ijl}(E)\sigma_{ij}(E), \tag{2.173}$$

where $\overline{E_{ijl}}$ is the total kinetic energy carried away by the l^{th} species of secondary particles, and σ_{ij} is the cross section for material i and reaction j. The neutron KERMA factors are calculated in the HEATR module of NJOY as a pointwise cross section. However, since only a few materials have the exact total kinetic energy data carried away by the secondary particles in their libraries, NJOY computes KERMA factors by the energy-balance method:

$$k_{ij}(E) = (E + Q_{ij} - \overline{E}_{ijn} - \overline{E}_{ijr})\sigma_{ij}(E), \tag{2.174}$$

where $\overline{E}_{ijn}$ and $\overline{E}_{ij\gamma}$ are the total energy of secondary neutron including multiplicity and that of photon including multiplicity, respectively. In the energy-balance method, the negative neutron heating may appear in some cases due to the inconsistent data set of neutron and photon. Even though the energy-balance method could induce unphysical negative values in the process, it is applicable to the heat calculation since the negative KERMA factors cancel the excess heating by photons and guarantee the conservation of total energy in large homogeneous system. Those pointwise KERMA factors are collapsed in the module GROUPR.

The module GROUPR produces the neutron KERMA factors at infinite dilute condition. In order to account for the self-shielding in the neutron KERMA factor, the partial kinetic energy release per reaction is determined by dividing the partial KERMA factor by the neutron cross section. In a MC^2-3 calculation, the neutron KERMA factors for a given composition are computed by multiplying the energy releases per reaction and self-shielded cross sections.

In the HEATR module of NJOY, the effective Q-value is determined by subtracting the portion of delayed photons and neutrinos from the mass difference Q-value. In order to obtain the exact KERMA factors and photon production matrix, the contribution of delayed photons should be considered after the process of NJOY. Main sources of delayed photons are the decays of fission products and their daughter isotopes. The contribution of delayed photons and betas resulting from the decay of fission products can be easily obtained from the ENDF data File 1, MT 458. Under the assumption that beta particle energies are deposited locally at the site of fission occurred, the contribution of delayed betas to heat generation rates can be

considered by adding the delayed beta KERMA factor to the total KERMA factor. The delayed beta KERMA factor can be determined by multiplying the total energy released by delayed betas with the self-shielded fission cross section.

To calculate the delayed gamma production matrix, the decay chains of fission products are constructed by reading the fission yield and radioactive decay sublibraries. The gamma production matrix is then constructed by aggregating all the decay gamma spectra. The ENDF-6 format sublibraries of decay and fission yield provide the information for decay heat analysis and fission products. The accuracy of delayed gamma yield calculation depends on the data in these libraries. Although the ENDF/B VII sublibraries provide the decay data for all fission products, it is noted that the accuracy of decay data are not guaranteed due to the incompleteness of experimental data for short-lived nuclides and complex decay scheme. Moreover, when the Q-value of a fission product increases, the gamma rays with weak energy are not observed in the gamma ray measurement. Therefore, the JENDL decay data compiled in 2011 with the fission-product decay data file are used to generate the delayed gamma and beta libraries. The isotopes including delayed gamma and beta data for ENDF/B-VII.0 are listed in Appendix D.

2.14 Assumptions and Limitations

In the previous sections, the methods and capabilities of MC^2-3 have been described in details. Assumptions and approximations used in developing each method have also been discussed. Although the verification and validation test results for various fast system benchmark problems have demonstrated that MC^2-3 generates accurate cross sections for fast reactor applications [4], the assumptions and limitations of MC^2-3 are summarized below to remind the user.

- Up-scattering is not considered, and no thermal scattering law is implemented. Thus MC^2-3 is not applicable to the thermal energy range.
- The narrow resonance approximation is used for self-shielding unresolved and resolved resonances. (The resolved resonances are self-shielded for the ultrafine group calculation only.)
- For scattering transfer matrix calculation, $\sigma_s(E)\phi_\ell(E)$ is assumed to be constant within a ultrafine group (lethargy width of 1/120).
- Unresolved resonance cross sections are self-shielded at the ultrafine group level using equivalence theory, even for the hyperfine group calculation.
- In the hyperfine group calculation, the fission and non-elastic scattering sources are obtained by interpolating the corresponding ultrafine group sources.
- Interaction between resolved and unresolved resonances is not taken into account.
- Flux moments higher than one are calculated using the recursive relation based on the extended transport approximation.
- The (n,3n) reaction is not included.

- The maximum energy can easily be extended from 14.2 MeV to 20 MeV. However, because of no (n,3n) reaction treated, the total cross sections are noticeably underestimated in the high energy range above 10 MeV.
- Inelastic scattering moments higher than one are not provided; the P_1 moments are currently supplied by the external files which are generated by NJOY.
- Variable lethargy intervals in the ultrafine and hyperfine group levels are not allowed.
- The Legendre expansion coefficient factor $2\ell+1$ is not included in the resulting cross sections.
- The (n,2n) cross sections are provided in the reaction basis.
- Only homogeneous and one-dimensional slab and cylinder geometries are currently supported for unit cell calculations.
- Multigroup gamma production matrices, KERMA (Kinetic Energy Released to Material) factors, and kinetic parameters are not generated yet.
- The MC^2 libraries have been tested and verified only for ENDF/B-VII.0 data, and those for ENDF/B-V are being tested because of methodological changes in MC^2-3, although they were used successfully in MC^2-2 for a long time. The other libraries such as ENDF/B-VI, JEF, and JENDL have not been tested yet.

Note that the Legendre expansion coefficient factor, $2\ell+1$, is not included in any of $\sigma_{\ell G}^{total}$, $\sigma_{\ell G}^{tr}$, and $\sigma_{\ell GG'}^{s}$.

3 MC^2 Library

3.1 ETOE-2

A suite of RIGEL/ETOE-2 codes [3] processes the ENDF/B data and prepares the MC^2 libraries. An original ENDF/B file contains evaluated data in the form of six data records. The file is first processed by the RIGEL code and transformed from the ENDF/B format into the binary format which is made readable by ETOE-2. At this step no change is introduced into the original structure of the ENDF/B file. The ETOE-2 code processes the fundamental nuclear data written in the RIGEL binary format to prepare eight binary library files for use in MC^2-2 and MC^2-3.

The ETOE-2 code performs the following five basic functions which are diagrammed in Figure 3.1: (1) reformat the data, (2) preprocess resonance cross sections by converting Reich-Moore parameters to multi-pole parameters and by screening extremely wide and weak resolved resonances (the latter for MC^2-2 only), (3) generate ultrafine-group smooth cross sections, (4) calculate isotope-independent function tables, and (5) convert all ENDF/B formats to the laws which are allowed in MC^2-3. A brief code structure of ETOE-2 is presented in Appendix E.

The ETOE-2 code is able to treat single level, the multilevel Breit-Wigner formulas, the Adler-Adler description, and the Reich-Moore formalism. It also has capabilities to convert the Reich-Moore parameters into the multi-pole parameters or Adler-Adler parameters so that the cross section values at any energy and temperature can be represented as a linear combination of the Doppler-broadened line shape functions. While the ENDF/B data permit six secondary energy distribution laws for inelastic and (n,2n) scattering, MC^2-3 permits only three of those: tabulated function, evaporation spectrum, and discrete levels, and thus ETOE-2 generates a tabulated function if data are provided with other than the three laws.

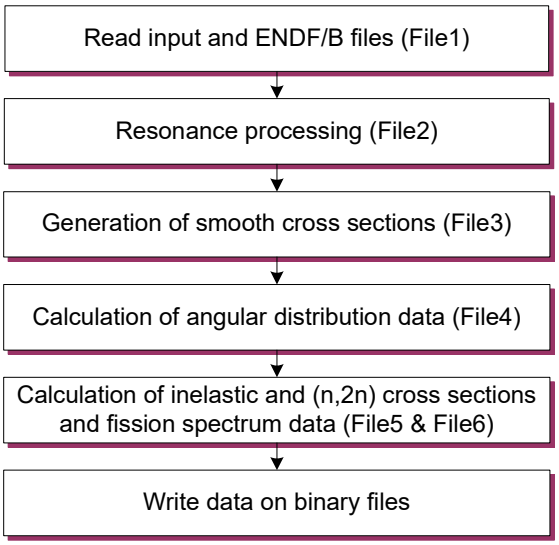


Figure 3.1 Generation of Isotopic MC^2 Library in ETOE-2

The binary data files provided by the ETOE-2 code contains resolved resonance parameters, unresolved resonance parameters, ultrafine group smooth cross sections, inelastic and (n,2n) scattering data, fission spectrum parameters, elastic scattering distributions, isotope-independent function tables, and an administrative file. Those data are saved to the MC^2 library files [1], as listed in

Table 3.1.

ETOE-2 has been changed significantly in order to correctly process the ENDF/B-VII.0 data released in December 2006. Even though the code had been updated to process the new formats of ENDF/B-VI from 1990s to early 2000s, problems and inaccuracies were still detected while processing the ENDF/B-VII.0 files and verifying the resulting MC^2 libraries. For example, one of the important options available in ETOE-2 is to screen out very wide and weak resolved resonances into composition- and temperature-independent ultrafine group smooth cross sections. Screening-out resolved resonances wider or narrower than certain criteria is required for the ultrafine group calculation in the MC^2-2 methodology. In fact, the screening process needs iterations between ETOE-2 and MC^2-2 to ensure that the screening criteria used in ETOE-2 are adequate for generating accurate cross sections in MC^2-2. This cumbersome process for each isotope was one of the motivations to develop the new self-shielding method in MC^2-3, which does not need the screening-out process.

The resolved resonance data of Pu-239 is given for three separate energy regions (0~1 keV, 1~2 keV, and 2~2.5 keV) in the ENDF/B-VII.0 data. The three energy regions respectively have 405, 441, and 224 sets of resolved resonance data. The resonance data in each energy region are supposed to have contributions to the corresponding region only. Therefore, the ETOE-2 and MC^2-3 codes have been updated to process the resolved resonance data given in multiple energy regions. When the Reich-Moore formalism given in ENDF/B data is converted to the multi-pole representation, a reduced form of multi-pole representation produces a few pseudo poles (typically, three pseudo poles for each cross section type) which represent smoothly varying components. The current version of ETOE-2/MC^2-3 assigns different indices to in-range, out-range, and pseudo poles so that out-range and pseudo poles are identified and added to the smooth cross sections without explicit treatment. Currently, the indices are extended to cover up to three separate energy ranges.

Table 3.1 MC^2 Library Files

Library	Description
MCCF1	Maximum data sizes, control information, material IDs, etc.
MCCF2	Isotope-independent function table
MCCF3	Unresolved resonance data
MCCF4	Resolved resonance data
MCCF5	Ultrafine-group smooth (non-resonant) cross section data
MCCF6	Inelastic and (n,2n) P_0 scattering data
MCCF7	Fission spectrum, neutron-per-fission data
MCCF8	T-function matrices, anisotropic elastic scattering Legendre data
MCCF9	Chi matrix, infinite-dilute ultrafine-group total cross sections
MCCF10	Ultrafine-group Inelastic scattering matrices

3.2 ENDF/B-VII Library

In addition to the eight standard MC^2 library files, MC^2-3 requires the chi matrix data files of fissionable isotopes and the inelastic scattering data files. The PENDF files of NJOY are also used as an alternative to the reconstruction of resolved resonance cross sections (in particular, for verifying the reconstructed pointwise cross sections) or to self-shield the resonance-like cross sections of intermediate mass isotopes above the resonance energy. As discussed in Section 3.1, the eight MC^2 library files are generated using the ETOE-2 code. Detailed information on those files is available in separate input manuals.

The MC^2 libraries are prepared for almost all 393 isotopes of ENDF/B-VII.0. The resolved and unresolved resonance maximum energies of all isotopes are summarized in Figure 3.2. Note that there are no unresolved resonances in most of the isotopes whose atomic masses are less than 70. The resonance cutoff energies as well as identifications of isotopes in the ENDF/B-VII.0 MC^2 library are listed in Appendix B.

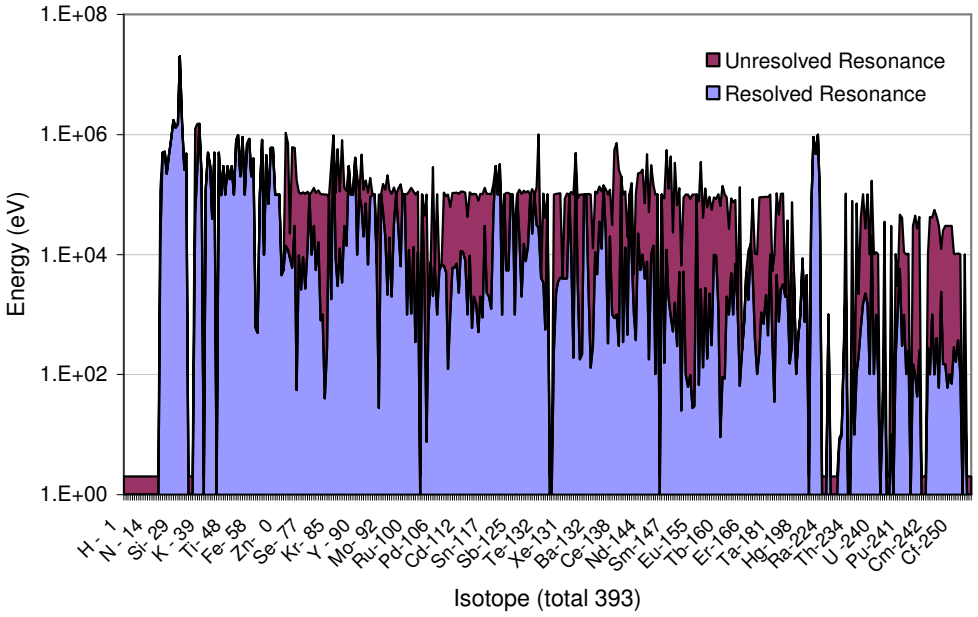


Figure 3.2 Upper Energy Boundaries of Resolved and Unresolved Resonances (ENDF/B-VII.0)

The ENDF/B-V MC^2 libraries are also available, whose isotopic identifications are summarized in Appendix C. Since they were originally generated for MC^2-2 that requires screening out very wide and weak resolved resonances, some isotopes have more than one set of cross sections which are characterized by a suffix at the 5th or 6th character position in the isotope names. The suffix "S" means an isotope with screened resolved resonances, which is normally used in MC^2-2. The ENDF/B-VI libraries were also prepared in early 2000s, but they are not supported any more since they have not been validated extensively.

The use of the existing ENDF/B-V libraries in MC^2-3 have not been fully verified, but the preliminary test results using the ENDF/B-V libraries indicated that the iron library

represented with "FE SV" appeared to be incompatible with the new methodologies. Therefore, it is recommended to use the PENDF file at least for iron when using the existing ENDF/B-V libraries for MC^2-3.

The chi matrix data file contains the ultrafine group chi matrices as a function of incident and secondary neutron energies, and it is provided as a separate file for each fissionable isotope. The inelastic scattering data file contains the ultrafine group matrices of P_0 and P_1 inelastic scattering, and it is also provided as a separate file for each isotope. Since the P_0 inelastic scattering data are included in the standard library file MCCF6, the P_0 data in this file are used only for normalizing the P_1 data to be consistent with the P_0 data in MCCF6. Once the ETOE-2 code is modified to produce the chi matrix and anisotropic inelastic scattering data, these external files would no longer be necessary.

The ENDF/B-VII.0 gamma libraries are generated using the HEATR and GAMINR modules of NJOY. The libraries include 391 isotopic gamma heating data and 100 element-wise gamma production and interaction data, based on 21 gamma groups. One additional file, named "table_gammalib," is necessary to map the files and isotopes.

4 INPUT DESCRIPTION

The MC^2-3 code needs many input files and generates several output files, as shown in Figure 4.1. Each file has its own fixed file name or naming rules. The standard input and output are named "input" and "output," respectively. The eight MC^2 libraries are called MCCF[] in which [] ranges from 1 to 10. Each of these files contains the data described in Table 3.1 for all isotopes, except for MCCF2 which is composed of the isotope-independent function tables. Each chi matrix data file includes data for a single isotope and thus the file name is created with the prefix "chimatrix" and the corresponding isotope identification (e.g., chimatrix.U238_7). Similarly, each inelastic scattering data file is named with the prefix "inelastic" and the corresponding isotope name (e.g., inelastic.U238_7). MC^2-3 optionally uses PENDF files generated with NJOY. Each PENDF file name is also dependent upon its isotope identification. In addition, it is tagged with its temperature condition in Kelvin (e.g., pendf.U238_7.0300 for U-238 with 300K). The pointwise cross section file generated by a MC^2-3 job can be saved for future use, and it is named in the similar way to the PENDF file name (e.g., pwxs.U238_7.0300).

In addition to the standard output file, the code produces an ISOTXS file by default and many other output files depending upon user-specified edit options for flux and moment, fission spectrum, microscopic or macroscopic cross section, and leakage. Those output files are named with the prefix "output." (e.g., output.flux_bg, output.flux_ufg, output.chi_bg, output.microxs_bg, output.macroxs_bg, and output.leakage_ufg.r01)

The input data for MC^2-3 are based on the namelist format provided in the Fortran language. The namelist block should begin with "&" (or "$") and end with "/". Currently, the namelist blocks available include the following eight blocks:

&control
&control1d
&geometry
&groupstructure
&library
&material
&output
&twodant

A whole line can be commented out by adding "!" at the first column. An exclamation mark can also be placed in any of the lines in order to write comments. If the same block appears more than once in the input file, the last one will overwrite the previous ones. Any input or statement outside a namelist block will be ignored. The space between blocks is a good place for comments.

The type of input cards can be identified by the first prefix character of each input card

c_ : character or string
i_ : integer
r_ : real
l_ : logical
t_ : type

Basically, no quotation or dot is necessary for character or logical data. As an exception, directory names with "/" or isotope names with blanks in the middle should be enclosed with double quotations because the slash in the directory name must be distinguished from the block end and the blank in the isotope name is differentiated from a word separator. Mostly the use of lower case input is recommended, excluding a few exceptions such as the built-in group structures (e.g., ANL230).

At least three namelist blocks are necessary: &library, &control, and &material for a homogeneous mixture problem, and additionally &geometry for a one-dimensional unit cell problem. For the libraries, four cards are required: c_mcclibdir, c_chilibdir, c_inelslibdir, and c_pwlibdir, among which c_pwlibdir is necessary to make use of the existing pointwise cross sections or to save those for the future use.

The hyperfine group calculation requires two sequential jobs: the first job is to perform the hyperfine group transport calculation to create the hyperfine group flux solutions, and the second job is to use those solutions for self-shielding ultrafine group cross sections. Therefore, the hyperfine group option activated in the first job (i.e., l_hyperfine_transport = True) needs to be turned off in the second job. In addition, as the hyperfine group calculation normally takes a long time, the flux solution file that is created in the first job is recommended to be saved just in case that the second job fails.

Similarly to the hyperfine group calculation, the two-dimensional transport calculation also needs two sequential jobs: the first job is to call TWODANT to perform the two-dimensional ultrafine group transport calculation which produces regionwise flux and moment solutions, and the second job is to use the TWODANT solutions for group condensation. In a similar manner to the hyperfine group calculation, the two-dimensional calculation option (l_twodant = True, c_twodant_group = UFG, and l_isotxs_macroxs = True) activated in the first job needs to be turned off in the second job. It is advisable to save the flux solution file created at the first job in case that the second job fails. Note that for this capability we used TWODANT version 02-05-90 which produces the *rzmflx* file for regular zone-averaged moment fluxes.

Figure 4.2 shows the calculation flow of the code. As shown in the figure, a major difference in the iteration between the hyperfine group calculation and the two-dimensional transport calculation using TWODANT is that the former is to update the energy self-shielding (resolved resonance self-shielding) and the latter is used to revise the energy spectrum (group condensation).

As the reconstruction of pointwise cross sections is costly, they are stored for future use. As aforementioned, the file name convention is pw.{isotope name}.{temperature}. As the Doppler broadening effect is quite small for a certain small variation of temperature, the temperature window, which is set to ±15K by default, can be given by the user so that the reconstruction of the pointwise cross sections are avoided. More details of the input cards will be discussed in the following sections. The isotopic pointwise cross section files are not provided with the code package but can be generated for specific temperatures by the user.

As an output, at least one ISOTXS file is always generated. When more than one mixture is defined, multiple ISOTXS files for individual compositions (e.g., ISOTXS.r01, … , ISOTXS.r99) and an additional ISOTXS file which is generated by merging all composition-wise ISOTXS files (ISOTXS.merged) are produced.

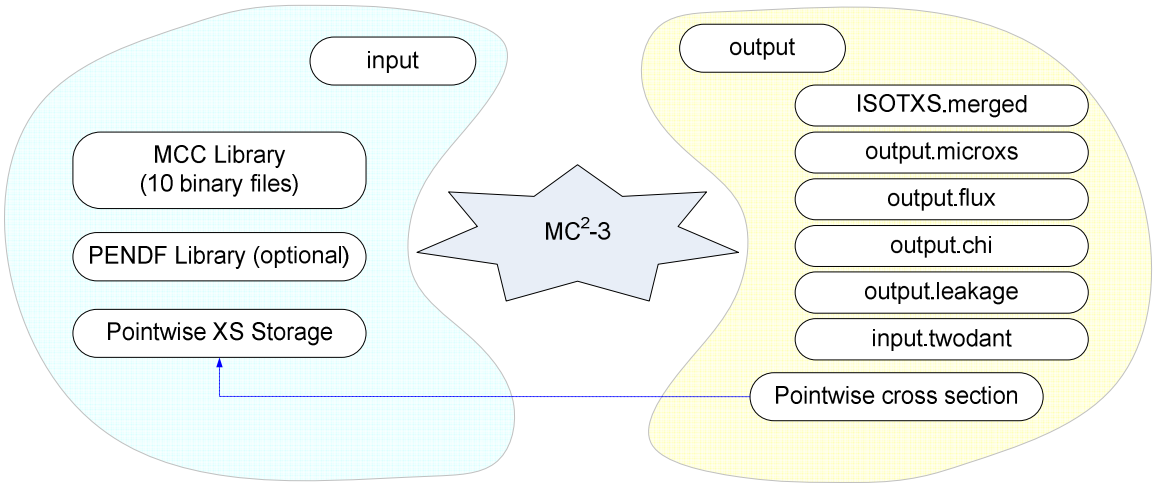


Figure 4.1 Inputs and Outputs of MC^2-3

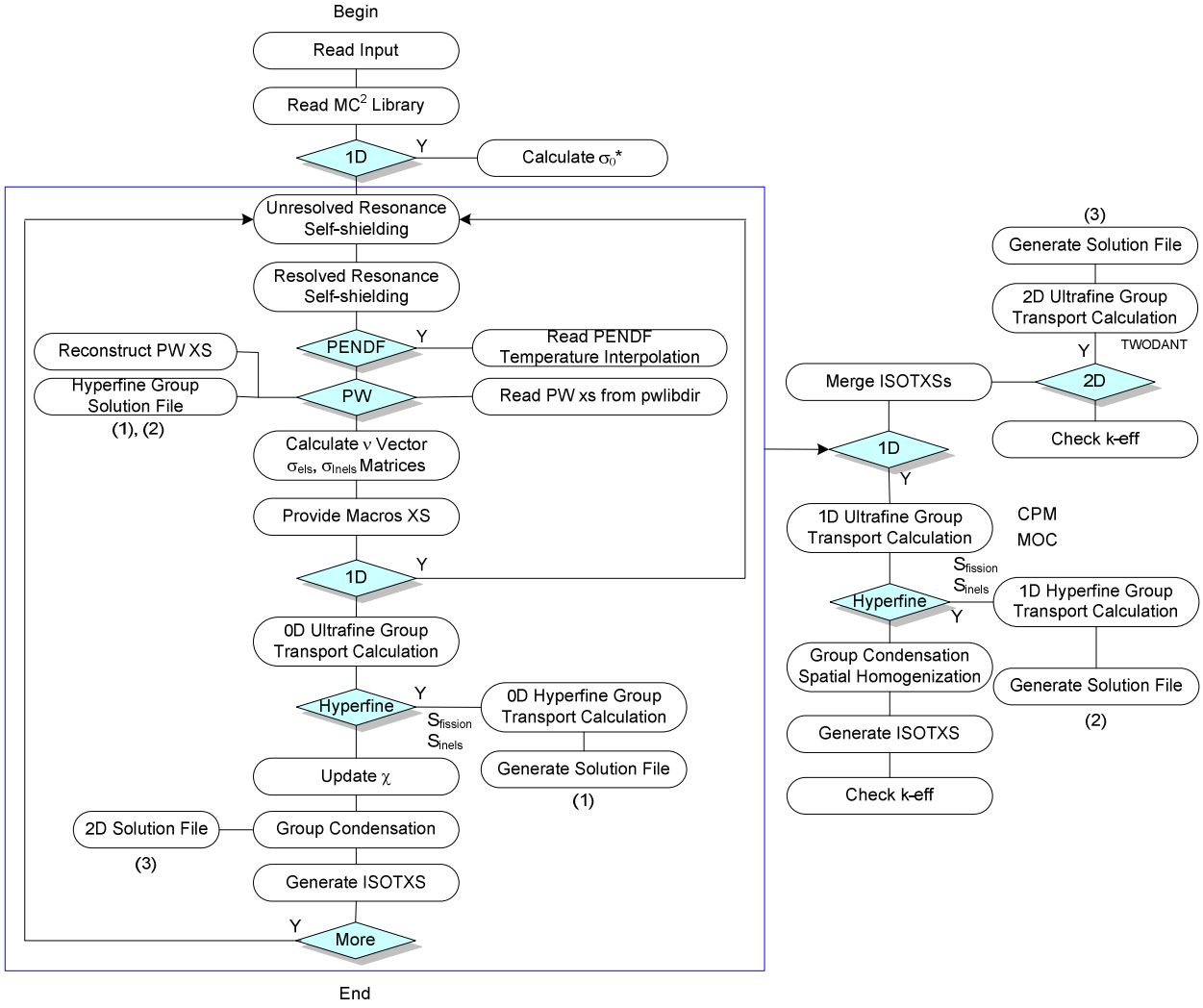


Figure 4.2 Detailed Calculation Flow of MC^2-3

The MC^2-3 code possesses many capabilities that used be performed by other utility programs. For example, the data conversion of ISOTXS from ASCII to binary and vice versa can be done, by simply activating a few input cards (l_isotxs_conversion = bin2asc, c_isotoxs_file = ISOTXS.r01). If more than an ISOTXS file is defined, then both conversion and integration occur and thus the input cards can be used to merging multiple ISOTXS files. The conversion and/or integration of other types of files, i.e., pointwise cross sections and MC^2 libraries, are also available. Details of the input blocks and cards will be discussed in the following sections.

4.1 Main Control Block

&control

Name	Type	Option	Description
c_dlayxs_conversion	character (len=8)	asc2bin bin2asc	Convert DLAYXS from ASCII to binary or vice versa. This card requires c_dlayxs_file in &output. bin2asc: input files: {user input} output files: {user input}.ascii asc2bin: input files: {user input} output files: {user input}.binary (Relevant blocks and cards) c_dlayxs_file (&output) (Example) &control c_dlayxs_conversion = asc2bin / &output c_dlayxs_file = dlayxs.asc / This job produces dlayxs.asc.binary
c_externalspectrum_ufg	character (len=50)		A file name of the ultrafine-group spectrum solution generated from TWODANT. This input is used after obtaining the zone-wise spectrum solutions from the TWODANT calculation.

			(Relevant blocks and cards) l_twodant c_twodant_group
c_externalspectrum_hfg	character (len=50)		A file name of the hyperfine-group spectrum solution. This input is used after obtaining the hyperfine-group spectrum solutions from the previous MC^2 job. (Relevant blocks and cards) l_hyperfine_transport
c_geometry_type	character (len=8)	mixture (D) slab cylinder	Choice of mixture or one-dimensional geometry (slab or cylinder). The choice of slab or cylinder requires defining the &geometry block. (Relevant blocks and cards) &geometry
c_group_structure	character (len=8)	ANL230 (D) USER	Broad group structure. Built-in group structures: ANL2082, ANL1041, ANL703, ANL425, ANL230, ANL116, ANL70, ANL33, ANL9, ANL4, ANL2 (see Appendix D for details of the built-in group structures). The choice of USER requires defining the &groupstructure block. (Relevant blocks and cards) &groupstructure
c_inelastic_treatment	character (len=12)	approximate(D) rigorous	Inelastic scattering cross section calculation method.
c_isotxs_conversion	character (len=8)	asc2bin bin2asc	Convert ISOTXS from ASCII to binary or vice versa. If more than one file is defined, merge and convert them. This card requires c_isotxs_file in &output. bin2asc: input files: {user input}

			output files: {user input}.ascii asc2bin: input files: {user input} output files: {user input}.binary (Relevant blocks and cards) c_isotoxs_file (&output) (Example) &control c_isotxs_conversion = asc2bin / $outout c_isotxs_file = isotxs.a1 isotxs.a2 / This job produces ISOTXS.merged and ISOTXS.merged.ascii, instead of {user file name}.binary
c_mcclib_conversion	character (len=8)	asc2bin bin2asc	Convert the MC^2 libraries from ASCII to binary or vice versa. bin2asc: input files: MCCF[] output files: MCCF[].ascii asc2bin: input files: MCCF[].ascii output files: MCCF[].binary (Example) &control c_mcclib_conversion = asc2bin /
c_pmatrx_conversion	character (len=8)	asc2bin bin2asc	Convert PMATRX from ASCII to binary or vice versa. bin2asc: input files: {user input} output files: {user input}.ascii asc2bin: input files: {user input}.ascii output files: {user input}.binary (Example) &control

			c_pmatrx_conversion = asc2bin /
c_pw_conversion	character (len=8)	asc2bin bin2asc	Convert a pointwise (PW) file from ASCII to binary or vice versa. bin2asc: input files: {user input} output files: {user input}.ascii asc2bin: input files: {user input} output files: {user input}.binary (Relevant blocks and cards) c_pw_file (&output) (Example) &control c_pw_conversion = asc2bin / &control c_pw_file = pw_U235_7.0300 / This job produces pw_U235_7.0300.ascii.
c_transport_solver	character (len=8)	cpm (D) moc	The one-dimensional transport solver. MOC is available only for slab (not available for the current version).
c_twodant_group	character (len=8)	UFG (D) BG	Energy group for TWODANT calculation. This input makes sense only when l_twodant = True. For group condensation using the 2D flux solutions from TWODANT, two sequential jobs are required. In the first job, c_twodant_group should be set so that the ultrafine group 2D calculation is performed. The second job does not require running TWODANT. If c_twodant_group = BG, the TWODANT calculation is conducted with broad group cross sections,

			which is useful to check broad group cross sections using TWODANT. (Relevant blocks and cards) &twodant c_twodantexe (&library) l_twodant (&control) l_isotxs_macroxs (&output)
i_debug_mode	integer	0 (D) 1	Debugging mode. 1: print debugging data
i_external_chi	integer	3 (D) 0, 1	Fission spectrum (chi) calculation option: 0 : use the original chi from the MC^2 lib. 1 : use the chi vector which is generated by NJOY using the typical fast system spectrum 2 : (unused) 3 : use the chi matrix MCCF9 includes all fission matrix data (Relevant blocks and cards) c_chilibdir (&library)
i_max_inneriteration	integer	100 (D)	Maximum inner iteration.
i_max_outeriteration	integer	200 (D)	Maximum outer iteration for the eigenvalue calculation.
i_number_region	integer	0 (D)	Number of regions, up to 99 as long as the system memory is allowed. (Relevant blocks and cards) t_composition (&material)
i_scattering_order	integer	3 (D) 0 – 9	Order of Legendre moments for scattering, total and transport cross sections. Recommend to use up to 5.
i_temperature_window	integer	15 (D)	Check the existing PW files with the input temperature ± window so that no PW reconstruction is performed. (Example) i_temperature_window = 0
i_use_mcclib	integer	10 (D)	The library file number to read.

			For example, if 8 is input, MCCF9 and MCCF10 would not be read.
l_aboveresonance	logical	false (D)	Self-shielding above resonance. Since this requires PENDF files, the c_pendflibdir card in the &library block should be defined. (Relevant blocks and cards) c_pendflibdir (&library)
l_broadgroup_transport	logical	false (D)	Transport calculation using the broad group cross sections. This works only for the one-dimensional calculation.
l_buckling_search	logical	false (D)	Buckling search for critical spectrum.
l_external_inelasticpn	logical	true (D)	P_N ($N > 0$) for inelastic scattering are obtained from the external file which is normally generated by NJOY. Currently only P_1 is prepared. P_N moments ($N > 0$) are normalized by the factor P_0 (MC^2-3)/P_0 (NJOY). (Relevant blocks and cards) c_inelslibdir (&library)
l_external_source	logical	false (D)	Use external leakage source which is generated when l_edit_leakage = True in the &output block. This card requires setting the l_edit_leakage to true or providing output.leakage_ufg.r[][] (where [][] is a two-digit region number from 01 to 99) before running the executable. The user may define the i_source_region(:) of the &material block. If nothing is defined, i_source_region(i) is set to i-1 by default. (Relevant blocks and cards) l_edit_leakage (&output) i_source_region (&material)
l_gamma	logical	false (D)	Gamma cross section calculation

			This generates PMATRX (gamma production data) and GAMISO (gamma interaction data) in a binary form. The output files "pmatrx.out" and "gamiso.out" are generated as well for the user information. Those binary files can be converted to the ASCII format using the input cards "l_pmatrx_ascii" and "l_gamiso_ascii" in the &output block in the same job or using the input cards "c_pmatrx_conversion" and "c_isotxs_conversion" in a separate job. (Relevant blocks and cards) l_pmatrx_ascii (&output) l_gamiso_ascii (&output)
l_hyperfine_transport	logical	false (D)	Hyperfine group calculation for mixture or one-dimensional calculation. This option requires two sequential jobs: the first job is to generate hyperfine group solutions and the second job is to use those solutions to self-shield ultrafine group cross sections.
l_pendf	logical	false (D)	Use PENDF if available for resolved resonance self-shielding. PENDF is normally provided by NJOY with several temperatures so that temperature interpolation is possible at input temperature. The temperature table, called "table_pendf" should be available in the directory defined at the "c_pendflibdir" card of the &control block. This option is very useful for verifying the pointwise resolved resonances reconstructed in the code. (Relevant blocks and cards)

			c_pendflibdir (&library)
l_pn_correction	logical	true (D)	P_N correction for within-group scattering cross section $\sigma_{s,gg}^{n'} = \sigma_{s,gg}^{n} + (\sigma_{tg}^{0} - \sigma_{tg}^{n})$, $n \geq 1$
l_twodant	logical	false (D)	TWODANT calculation. This requires defining the &twodant block. If setting the "l_isotxs_macroxs" card to "True" in the &output block, then the smaller size of ISOTXS is created with regionwise macroscopic cross sections and so less memory is required for TWODANT. (Relevant blocks and cards) c_twodantexe (&library) l_twodant (&control) c_twodant_group(&control) l_isotxs_macroxs (&output)
r_eps_buckling	real	1.E-3 (D)	Conversion criterion for buckling. (Relevant blocks and cards) l_buckling_search (&control)
r_eps_rho	real	1.E-5 (D)	Conversion criterion for eigenvalue
r_eps_source	real	1.E-5 (D)	Conversion criterion for source. L2 norm is checked.

(Example)

```
&control
     c_group_structure   =ANL33              ! 33 groups
     i_number_region     =2                  ! 2 regions
     i_scattering_order  =5                  ! P5
     c_geometry_type     =mixture
/
```

```
&control
     c_group_structure   =USER               ! requires &groupstructure
     i_number_region     =2                  ! 2 regions
     i_scattering_order  =5                  ! P5
     c_geometry_type     =slab               ! requires &geometry
     l_twodant           =T
     c_twodant_group     =UFG
/
```

4.2 Control Block for One-Dimensional Geometry

(Precondition)

```
&control
     i_number_region = 10             ! greater than 1
     c_geometry_type = cylinder       ! or slab
/
```

&control1d

Name	Type	Option	Description
i_homogenized_region	integer	i_number_region (D)	Regions to be homogenized i_homogenized region ≤ i_number_region (in &control). (Relevant blocks and cards) c_geometry_type (&control) &geometry (Example) i_homogenized_region = 4 i_number_region = 5 Then, the first four regions out of total five regions will be homogenized.
l_spatial_homogenization	logical	true (D)	Spatial homogenization for one-dimensional geometry. If false, regionwise cross sections are generated. (Relevant blocks and cards) c_geometry_type (&control) &geometry

This block is activated when c_geometry_type is set to other than "mixture" (currently, only cylinder or slab is valid) in the &control block.

Even though a one-dimensional calculation is requested, this block is not always required.

(Example)

```
&control
     i_number_region     =5                ! 5 regions
     c_geometry_type     =cylinder
/
&control1d
     i_homogenized_region=3                ! homogenize the first 3 regions
/
```

```
&control1d
     l_spatial_homogenization = F           ! produce regionwise cross
                                            ! sections
/
```

4.3 Geometry Block

(Precondition)

```
&control
     i_number_region = 10          ! greater than 1
     c_geometry_type = cylinder    ! or slab
/
```

&geometry

Name	Type	Option	Description
c_boundary_condition	character (len=12)	reflective (D) periodic	Boundary condition. Only reflective boundary condition available for the cylindrical geometry.
i_composition	integer array	same as region number (D)	Assignment of compositions to regions. The number should not exceed the number of regions. (Example) i_composition = 1 2 1 2 1 * This expects two compositions defined in the &material block t_composition(:, 1) = t_composition(:, 2) =
i_mesh	integer array	0 (D)	Number of meshes for regions. (Example) i_mesh = 2 8 2 6 2
r_gfactor	real array	1 (D)	Multiplication factors to regionwise cross sections.
r_location	real array	0 (D)	Location of right boundary of regions from zero. (Example) i_mesh = 0.5 2.5 3.0 4.5 5.0

This block is activated when c_geometry_type is set to other than “mixture” (currently, only cylinder or slab is valid) in the &control block.

At least one of the regions should have fissionable material.

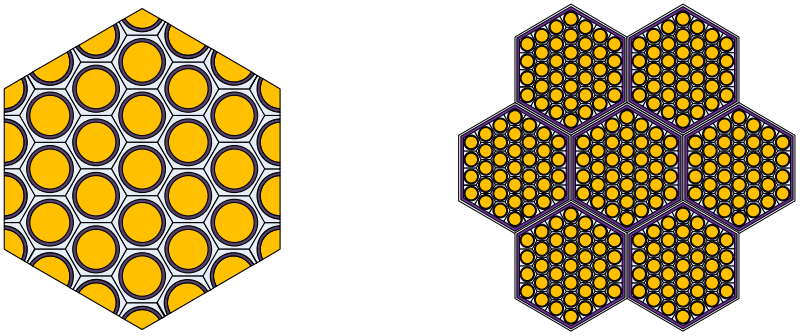
(Example)

```
&geometry
     c_boundary_condition = reflective
     i_mesh      =          2          2          6          2         12
                            6          2          6          2          2
     r_location =       0.22       0.27       0.81       0.86       1.17
                        1.50       1.70       1.80       2.10       2.25
     i_composition =       1          2          3          4          5
                           5          4          3          2          1
/
```

4.4 Geometry Block for MOC Calculations

4.4.1 Hexagonal Geometry Inputs

The MOC module is capable to model two types of hexagonal geometries in the triangular lattice: hexagonal cells or hexagonal assemblies the triangular lattice as illustrated in Figure 4.3. A geometry with hexagonal cells can be utilized for a unit cell calculation, which can be defined using the "T_HEXCELL_LATTICE" input card. To impose reflective boundary conditions, the cells on the periphery ring are split by boundary lines as shown in Figure 4.3 and the portions that are located inside of the boundary lines are used in the calculation. The hexagonal assembly model can be built using the "T_HEXASSY_LATTICE" and "T_HEXASSY_TYPE" input cards. The assembly duct and inter-assembly gap can be explicitly modeled and each assembly can have different fuel rod layouts.



(a) Hexagonal cells in triangular lattice (b) Hexagonal assemblies in triangular lattice

Figure 4.3 Hexagonal Geometries for MOC Calculations

The hexagonal cells, the basic constituent of hexagonal geometry input, can have the hexagonal or annular intra-cell structure as illustrated in Figure 4.4. The hexagonal and annular ring are further divided into 6 sectors to account for the azimuthal variations of angular fluxes.

(a) Hexagonal mesh structure (b) Annular mesh structure

Figure 4.4 Hexagonal cell structure

The 3D geometry is constructed based on the 2D planar geometry which is extruded in the axial direction. Therefore, all axial planes have the same ray structure and each may have different material assignment. In the input deck, the mesh structure is defined only for the radial direction and the material assignment for each axial plane is specified via user inputs to construct a 3D problem.

A group of meshes (or flat source regions) that shares the same cross section data is referred to as a region which has its own composition. In the MOC module, the cross section data is assigned to meshes in the two steps: defining a region for each mesh and choosing a corresponding composition to the region. The region-wise indices should be tracked instead of composition-wise indices.

Note that the current version supports the CMFD acceleration for 2D and 3D hexagonal geometry only. The number of energy groups for CMFD can be determined by an input card.

Triangular Lattice Properties and Boundary Conditions for Hexagonal Cells

T_HEXCELL_LATTICE%I_NUMPLANE

Description: Number of axial planes
If it is 1, the problem is the 2D case
(Default) 1
Type: Single integer
Associated Input:
T_HEXCELL_LATTICE%R_HZ
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_RING_REGION(:, :)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEX_REGION(:, :)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_GAP_REGION(:, :)
Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMPLANE = 10
/
```

T_HEXCELL_LATTICE%I_NUMRING

Description: Number of rings of triangular lattice filled with cells
Type: Single integer
Associated Input:
T_HEXCELL_LATTICE% I_CELLTYPE_MAP(:)
Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 5
/
```

T_HEXCELL_LATTICE%R_HZ (:)
Description: Axial mesh sizes of each plane
From bottom to top planes
Type: 1D integer array
(Array Size) T_HEXCELL_LATTICE%I_NUMPLANE
(1st Array Index) Plane index
Associated Input
T_HEXCELL_LATTICE% I_NUM_PLANE
Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMPLANE = 3
 T_HEXCELL_LATTICE%I_R_HZ(1:3) = 5.0 5.0 5.0
/
```

T_HEXCELL_LATTICE%R_CELL_FLAT2FLAT
Description: Flat to flat distance (pitch) of hexagonal cell
Used to specify triangular lattice structure
Type: Single real
Example

```
&geometry
   T_HEXCELL_LATTICE%R_CELL_FLAT2FLAT = 2.0
/
```

T_HEXCELL_LATTICE%C_TOP_BOUNDARY_CONDITION
T_HEXCELL_LATTICE%C_BOTTOM_BOUNDARY_CONDITION
Description: Boundary condition at the top boundary surface
(Default) reflective
Type: Character
Option: reflective / vacuum
Example

```
&geometry
   T_HEXCELL_LATTICE%C_TOP_BOUNDARY_CONDITION = reflective
   T_HEXCELL_LATTICE%C_BOTTOM_CONDITION = vacuum
/
```

Cross Section Mapping for Hexagonal Cells

T_HEXCELL_LATTICE%I_NUMREGION
Description: Number of regions in domain
Type: Single integer
Associated Input
T_HEXCELL_LATTICE% I_REGION_COMPOSITION(:)
Example

```
Given in 'T_HEXCELL_LATTICE%I_REGION_COMPOSITION'
```

T_HEXCELL_LATTICE%I_REGION_COMPOSITION (:)
Description: Arrangement of composition to regions
Type: 1D integer array
(Array Size) T_HEXCELL_LATTICE%I_NUMREGION
(1st Array Index) Region index
Associated Input
T_HEXCELL_LATTICE% I_NUMREGION
Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMREGION = 5
 T_HEXCELL_LATTICE%I_REGION_COMPOSITION(1:5) = 1 2 3 4 5
/
```

Cell Structure and Mapping for Hexagonal Cells

T_HEXCELL_LATTICE%I_NUMCELLTYPE
Description: Number of cell types defined in input deck
Type: Single integer
Associated Input
T_HEXCELL_LATTICE%T_CELLTYPE(:)
Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)
Description: Derived type for defining individual cell structures
Detailed description is given in below
Type: 1D derived type array
(Array Size) T_HEXCELL_LATTICE%I_NUMCELLTYPE
(1st Array Index) Cell type index
Associated Input
T_HEXCELL_LATTICE% I_NUMCELLTYPE
Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMPLANE              = 5
 T_HEXCELL_LATTICE%I_NUMCELLTYPE           = 2
 T_HEXCELL_LATTICE%R_CELL_FLAT2FLAT        = 1.2
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%c_type    = 'annular'
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%I_NUMRING = 3
 ! Annular Cell
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%R_RADIUS(1:3)         = 0.25 0.5 0.55
 !    Axial region arrangement of 1st ring
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%I_RING_REGION(1, 1:5) = 1 1 1 1 1
 !    Axial region arrangement of 2nd ring
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%I_RING_REGION(2, 1:5) = 1 1 1 1 1
 !    Axial region arrangement of 3rd ring
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%I_RING_REGION(3, 1:5) = 2 2 2 2 2
 !    Axial region arrangement of 4th ring
 T_HEXCELL_LATTICE%T_CELLTYPE(1)%I_RING_REGION(4, 1:5) = 3 3 3 3 3

 ! Hexagonal Mesh
```

```
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%c_type   = 'hex'
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%I_NUMHEX = 3
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%R_CENTER2SIDE(1:3)   = 0.25 0.5 0.55
 !    Axial region arrangement of 1st hexagonal shape ring
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%I_HEX_REGION(1, 1:5) = 1 1 1 1 1
 !    Axial region arrangement of 2nd hexagonal shape ring
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%I_HEX_REGION(2, 1:5) = 1 1 1 1 1
 !    Axial region arrangement of 3nd hexagonal shape ring
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%I_HEX_REGION(3, 1:5) = 2 2 2 2 2
 ! Axial region arrangement of 4th hexagonal shape ring
 T_HEXCELL_LATTICE%T_CELLTYPE(2)%I_HEX_REGION(4, 1:5)
/
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE

Description: Type of intra-cell structure
(Pre-defined Input Arguments) 'annular', 'hex'
Annual mesh structure : 'annular'
Hexagonal mesh structure : 'hex'

Type: Character

Associated Input

(For annular mesh structure)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_RADIUS(:)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_RING_REGION(:, :)
(For hexagonal mesh structure)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMHEX
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_CENTER2SIDE(:)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEX_REGION(:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING

Description: Number of annular rings in cell
Required to define annular meshes

Type: Single integer

Associated Input

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_RADIUS(:)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_RING_REGION(:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_RADIUS(:)

Description: Radii of annular rings
Required to define annular meshes
From center to outer rings

Type: 1D real array
(Array Size) T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING
(1st Array Index) Ring index

Associated Input

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE

T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_RING_REGION(:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_RING_REGION (:, :)

Description: Region Index Mapping
Required to define annular meshes
From center to outer rings and from bottom to top planes

Type: 2D integer array
(Array Size) (T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING+1)
× T_HEXCELL_LATICE%I_NUMPLANE
(1st Array Index) Ring index
(2nd Array Index) Axial mesh (plane) index

Associated Input

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMRING
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_RADIUS (:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMHEX

Description: Number of hexagonal shape rings in cell
Required to define hexagonal meshes

Type: Single integer

Associated Input

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_CENTER2SIDE(:)
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEX_REGION(:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_CENTER2SIDE(:)

Description: Center to side lengths (half of hexagonal pitch) of hexagonal shape rings
Required to define hexagonal meshes
From center to outer hexagonal shape rings

Type: 1D real array
(Array Size) T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMHEX
(1st Array Index) Ring index

Associated Input

T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMHEX
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEX_REGION(:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEX_REGION (:, :)

Description: Region index mapping

Required to define hexagonal meshes
From center to outer rings and from bottom to top planes

Type: 2D integer array
(Array Size) (T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_HEXRING+1)
× T_HEXCELL_LATICE%I_NUMPLANE
(1st Array Index) Hexagonal shape ring index
(2nd Array Index) Axial mesh (plane) index

Associated Input
T_HEXCELL_LATTICE%T_CELLTYPE(:)%C_TYPE
T_HEXCELL_LATTICE%T_CELLTYPE(:)%I_NUMHEX
T_HEXCELL_LATTICE%T_CELLTYPE(:)%R_CENTE2SIDE (:, :)

Example

```
Given in 'T_HEXCELL_LATTICE%T_CELLTYPE' description
```

T_HEXCELL_LATTICE%I_CELLTYPE_ MAP (:, :)

Description: Cell type arrangement for triangular lattice
Given for 2D array
Mapping for 2D array to triangular lattice is given in Figure 4.5
Illustration of mapping is given in Figure 4.6

Type: 2D integer array
(Array Size) (2*T_HEXCELL_LATTICE%I_NUMRING-1)
× (2*T_HEXCELL_LATTICE%I_NUMRING-1)
(1st Array Index) Index for x-direction
(2nd Array Index) Index for y-direction

Option: Positive integer
Null cell type : '0'

Associated Input
T_HEXCELL_LATTICE% I_NUMRING
T_HEXVELL_LATTICE%T_CELLTYPE(:)

Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 5
 T_HEXCELL_LATTICE%I_CELLTYPE_MAP(1:9, 1:9) =
                              1  1  1  1  1  0  0  0  0
                              1  1  1  1  1  1  0  0  0
                              1  1  1  1  1  1  1  0  0
                              1  1  1  1  1  1  1  1  0
                              1  1  1  1  1  1  1  1  1
                              0  1  1  1  1  1  1  1  1
                              0  0  1  1  1  1  1  1  1
                              0  0  0  1  1  1  1  1  1
                              0  0  0  0  1  1  1  1  1
/
```

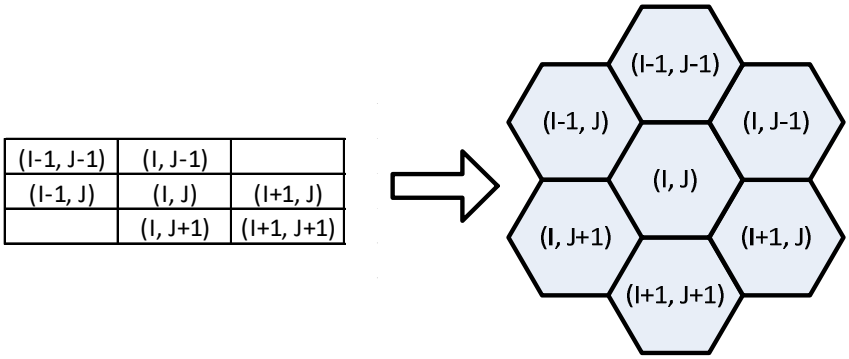


Figure 4.5 Mapping of Hexagonal Cells

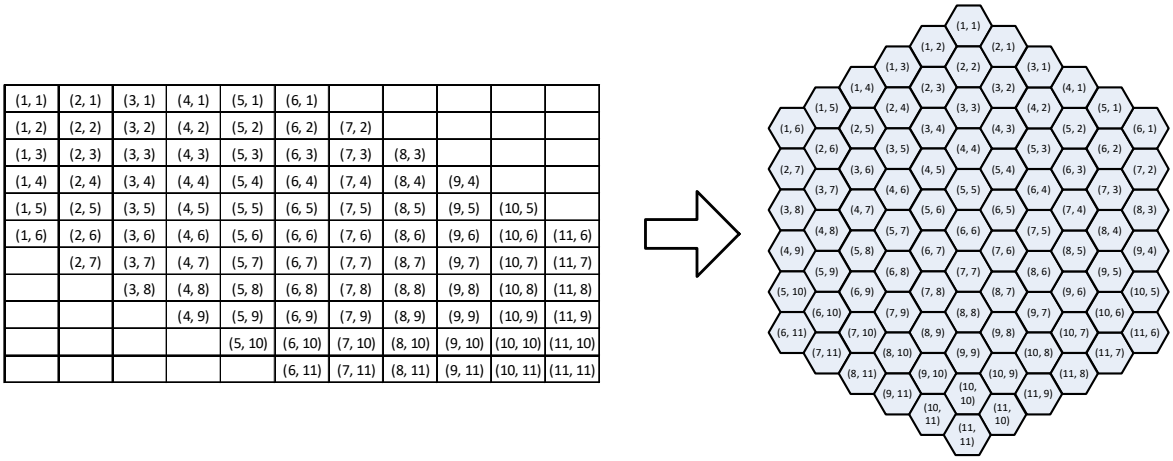


Figure 4.6 An example of 2D Mapping of Hexagonal Cells

Triangular Lattice Properties and Boundary Conditions for Hexagonal Assemblies

T_HEXASSY_LATTICE%I_NUMPLANE

Description: Number of axial planes
If this is 1, the problem is a 2D case
(Default) 1

Type: Single integer

Associated Input
T_HEXASSY_LATTICE%R_HZ(:)
T_HEXASSY_TYPE(:)%T_CELLTYPE(:)%I_RING_REGION(:, :)
T_HEXASSY_TYPE(:)%T_CELLTYPE(:)%I_HEX_REGION(:, :)
T_HEXASSY_TYPE(:)%T_CELLTYPE(:)%I_GAP_REGION(:, :)

Example

```
&geometry
 T_HEXASSY_LATTICE%I_NUMPLANE = 10
/
```

T_HEXASSY_LATTICE%I_NUMRING

Description: Number of rings of triangular lattice filled with assemblies

Type: Single integer
Associated Input Card
T_HEXASSY_LATTICE% I_CELLTYPE_MAP(:)
Example

```
&geometry
 T_HEXASSY_LATTICE%I_NUMRING = 5
/
```

T_HEXASSY_LATTICE%R_HZ (:)
Description: Axial mesh sizes of each plane
From bottom to top planes
Type: 1D integer array
(Array Size) T_HEXASSY_LATTICE%I_NUMPLANE
(1st Array Index) Plane index
Associated Input
T_HEXASSY_LATTICE% I_NUM_PLANE
Example

```
&geometry
 T_HEXASSY_LATTICE%I_NUMPLANE = 3
 T_HEXASSY_LATTICE%I_R_HZ(1:3) = 5.0 5.0 5.0
/
```

T_HEXASSY_LATTICE%R_ASSEMBLY_OUTER_FLAT2FLAT
Description: Flat to flat distance (pitch) of hexagonal assembly
Used to specify triangular lattice structure
Include dimensions of inter-assembly gap and duct structure
Type: Single real
Associated Input
T_HEXASSY_TYPE(:)%R_ASSEMBLY_INNER_FLAT2FLAT
Example

```
&geometry
   T_HEXASSY_LATTICE%R_OUTER_ASSEMBLY_FLAT2FLAT = 14.6850
/
```

T_HEXASSY_LATTICE%C_OUTER_BOUNDARY_CONDITION
Description: Boundary condition at outer boundary of plane
(Default) reflective
Type: Character
Option: reflective / vacuum
Example

```
&geometry
   T_HEXASSY_LATTICE%C_OUTER_BOUNDARY_CONDITION = reflective
   T_HEXASSY_LATTICE%C_TOP_BOUNDARY_CONDITION   = vacuum
/
```

T_HEXASSY_LATTICE%C_BOTTOM_BOUNDARY_CONDITION
T_HEXASSY_LATTICE%C_TOP_BOUNDARY_CONDITION
Description: Boundary condition at top and bottom boundaries of plane
(Default) reflective
Type: Character
Option: reflective / vacuum

Example

```
&geometry
   T_HEXASSY_LATTICE%C_BOTTOM_BOUNDARY_CONDITION = reflective
   T_HEXASSY_LATTICE%C_TOP_BOUNDARY_CONDITION    = vacuum
/
```

Cross Section Mapping for Hexagonal Assemblies

T_HEXASSY_LATTICE%I_NUMREGION

Description: Number of regions in the domain
Type: Single integer
Associated Input
T_HEXASSY_LATTICE%I_REGION_COMPOSITION(:)
Example

```
Given in 'T_HEXASSY_LATTICE%I_REGION_COMPOSITION'
```

T_HEXASSY _LATTICE%I_REGION_COMPOSITION (:)

Description: Assignment of compositions to regions
Type: 1D integer array
(Array Size) T_HEXASSY_LATTICE%I_NUMREGION
(1st Array Index) Region index
Associated Input
T_HEXASSY_LATTICE%I_NUMREGION
Example

```
&geometry
 T_HEXASSY_LATTICE%I_NUMREGION = 5
 T_HEXASST_LATTICE%I_REGION_COMPOSITION(1:5) = 1 2 3 1 2
/
```

Assembly Structure for Hexagonal Assemblies

T_HEXASSY_LATTICE%I_NUMASSYTYPE

Description: Number of assembly type defined in input deck
Type: Single integer
Associated Input
T_HEXASSY_TYPE(:)
T_HEXASSY_LATTICE%I_ASSYTYPE_MAP(:, :)
Example

```
Given in 'T_HEXASSY_TYPE' description
```

T_HEXASSY_TYPE(:)

Description: Derived type for defining individual assembly structure
Detailed description is given in below
Type: 1D derived type array
(Array Size) T_HEXASSY_LATTICE%I_NUMASSYTYPE
(1st Array Index) Assembly type index
Associated Input

T_HEXASSY_LATTICE%I_NUMASSYTYPE
T_HEXASST_LATTICE%I_ASSYTYPE_MAP(:, :)

Example

```
&geometry
  T_HEXASSY_LATTICE%I_NUMASSYTYPE = 2
  T_HEXASSY_LATTICE%R_ASSEMBLY_OUTER_FLAT2FLAT = 14.6850

  T_HEXASSY_TYPE(1)%C_NAME = 'Fuel Assembly'
  T_HEXASSY_TYPE(1)%I_NUMRING = 9
  T_HEXASSY_TYPE(1)%R_CELL_FLAT2FLAT = 0.9134
  T_HEXASSY_TYPE(1)%R_ASSEMLBY_INNER_FLAT2FLAT = 13.6790
  T_HEXASSY_TYPE(1)%I_NUMCELLTYPE = 2
  T_HEXASSY_TYPE(1)%L_GAP_STRUCTURE = T
  T_HEXASSY_TYPE(1)%I_CELLTYPE_MAP(1:17, 1:17) =
                                    ! Omitted for brevity
                                    ! Should be given
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1) ! Omitted for brevity
                                    ! Should in given
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2) ! Omitted for brevity
                                    ! Should in given
  T_HEXASSY_TYPE(1)%T_GAPTYPE     ! Omitted for brevity
                                    ! Should in given

  T_HEXASSY_TYPE(2)%C_NAME = 'Control Assembly'
  T_HEXASSY_TYPE(2)%I_NUMRING = 6
  T_HEXASSY_TYPE(2)%R_CELL_FLAT2FLAT = 1.2558
  T_HEXASSY_TYPE(2)%R_ASSEMLBY_INNER_FLAT2FLAT = 12.2707
  T_HEXASSY_TYPE(2)%I_NUMCELLTYPE = 1
  T_HEXASSY_TYPE(2)%L_GAP_STRUCTURE = T
  T_HEXASSY_TYPE(2)%I_CELLTYPE_MAP(1:11, 1:11) =
                                    ! Omitted for brevity
                                    ! Should be given
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1) ! Omitted for brevity
                                    ! Should in given
  T_HEXASSY_TYPE(2)%T_GAPTYPE     ! Omitted for brevity
                                    ! Should in given
/
! Detailed description of each entity are given in below
```

T_HEXASSY_TYPE(:)%C_NAME

Description: Name of assembly type
Type: Character

Example

```
Given in 'T_HEXASSY_TYPE' description
```

T_HEXASSY_TYPE(:)%I_NUMRING

Description: Number of ring for hexagonal cells in hexagonal assembly
Type: Single integer
Associated Input
T_HEXASSY_TYPE(:)%I_CELLTYPE_MAP(:, :)

Example

```
Given in 'T_HEXASSY_TYPE' description
```

T_HEXASSY_TYPE(:)%R_CELL_FLAT2FLAT
Description: Flat to flat distance (pitch) of hexagonal cell
Used to specify triangular lattice structure of assembly
Type: Single real
Example

```
Given in 'T_HEXASSY_TYPE' description
```

T_HEXASSY_TYPE(:)%R_ASSEMBLY_INNER_FLAT2FLAT
Description: Flat to flat distance (pitch) of hexagonal assembly excluding intra-assembly gap and duct structure
Used to specify triangular lattice structure of assembly
Type: Single real
Associated Input
T_HEXASSY_LATTICE%R_ASSEBLY_OUTER_FLAT2FLAT
Example

```
Given in 'T_HEXASSY_TYPE' description
```

T_HEXASSY_TYPE(:)%I_NUMCELL_TYPE
Description: Number of cell type defined in assembly
Type: Single integer
Associated Input
T_HEXASSY_TYPE(:)%T_CELLTYPE(:)
Example

```
Given in 'T_HEXASSY_TYPE(:)%T_CELLTYPE' description
```

T_HEXASSY_TYPE(:)%T_CELLTYPE(:)
Description: Derived type for defining individual cell structure
Detailed description of its entities
Define cell types for each assembly
Type: 1D derived type array
(Array Size) T_HEXCELL_LATTICE%I_NUMCELLTYPE
(1st Array Index) Cell type index
Associated Input
T_HEXASSY_TYPE(:)%I_NUMCELLTYPE
T_HEXASSY_TYPE(:)%I_CELLTYPE_MAP(:, :)
Example

```
&geometry
  ! Assembly Type 1
  T_HEXASSY_TYPE(1)%I_NUMCELLTYPE = 2
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%c_type = 'annular'
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%I_NUMRING = 3
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%R_RADIUS(1:3) = 0.25 0.5 0.55
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%I_RING_REGION(1, 1:5) = 1 1 1 1 1
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%I_RING_REGION(2, 1:5) = 1 1 1 1 1
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%I_RING_REGION(3, 1:5) = 2 2 2 2 2
  T_HEXASSY_TYPE(1)%T_CELLTYPE(1)%I_RING_REGION(4, 1:5) = 3 3 3 3 3
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%c_type = 'hex'
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%I_NUMHEX = 3
```

```
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%R_CENTER2SIDE(1:3)   = 0.25 0.5 0.55
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%I_HEX_REGION(1, 1:5) = 1 1 1 1 1
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%I_HEX_REGION(2, 1:5) = 1 1 1 1 1
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%I_HEX_REGION(3, 1:5) = 2 2 2 2 2
  T_HEXASSY_TYPE(1)%T_CELLTYPE(2)%I_HEX_REGION(4, 1:5) = 3 3 3 3 3
  ! Assembly Type 2
  T_HEXASSY_TYPE(2)%I_NUMCELLTYPE = 1
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%c_type = 'annular'
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%I_NUMRING = 2
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%R_RADIUS(1:2) = 0.5 0.51
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%I_RING_REGION(1, 1:5) = 1 1 1 1 1
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%I_RING_REGION(2, 1:5) = 2 2 2 2 2
  T_HEXASSY_TYPE(2)%T_CELLTYPE(1)%I_RING_REGION(3, 1:5) = 3 3 3 3 3
/
```

T_HEXASSY_TYPE(:)%I_CELLTYPE_MAP (:, :)

Description: Cell type arrangement for triangular lattice
Given for 2D array
Mapping for 2D array to a triangular lattice of assemblies is identical to that of 'T_HEXCELL_LATICE%I_CELLTYPE_MAP'
Null cell type : '0'

Type: 2D integer array
(Array Size) (2*T_HEXASSY_TYPE(:)%I_NUMRING-1) × (2*T_HEXASSY_TYPE(:)%I_NUMRING-1)
(1st Array Index) Index for x direction
(2nd Array Index) Index for y direction

Associated Input
T_HEXASSY_TYPE(:)%I_NUMRING
T_HEXASSY_TYPE(:)%I_NUMCELLTYPE
T_HEXASSY_TYPE(:)%T_CELLTYPE(:)

Example

```
&geometry
 T_HEXASSY_TYPE(1)%I_NUMRING = 8
 T_HEXASSY_TYPE(1)%I_CELLTYPE_MAP(1:15, 1:15) =
                             1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
                           3 1 1 1 1 1 1 1 1 0 0 0 0 0 0
                         1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
                       2 1 1 1 1 1 1 1 1 1 1 0 0 0 0
                     3 1 1 1 1 1 1 1 1 1 1 1 0 0 0
                   1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
                 2 1 1 1 1 1 1 1 1 1 1 1 1 1 0
               1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
             0 3 1 1 1 1 1 1 1 1 1 1 1 1 1
           0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
         0 0 0 2 1 1 1 1 1 1 1 1 1 1 1
       0 0 0 0 3 1 1 1 1 1 1 1 1 1 1
     0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
   0 0 0 0 0 0 3 1 1 1 1 1 1 1 1
 0 0 0 0 0 0 0 1 2 1 3 1 2 1 1
/
```

T_HEXASSY_TYPE(:)%L_GAP_STRUCTURE

Description: Flag for gap structure

(Important) Gap refers region (domain) between inner and outer assembly flat to flat distance and each side of assembly has gap (total 6 gaps in assembly)
Use gap structure for modeling the inter-assembly gap and duct structure

Type: Single logical

Option: True / False
W/ inter-assembly gap and duct structure : T
W/o inter-assembly gap and duct structure : F
(Default) T

Associated Input
T_HEXASSY_LATTICE%R_ASSEMBLY_OUTER_FLAT2FLAT
T_HEXASSY_TYPE(:)%R_ASSEMBLY_INNER_FLAT2FLAT
T_HEXASSY_TYPE(:)%T_GAPTYPE

Example

```
Given in 'T_HEXASSY_TYPE(:)%T_GAPTYPE' description
```

T_HEXASSY_TYPE(:)%T_GAPTYPE

Description: Derived type for defining gap structure of assembly
Define single gap structure (type) for assembly and each gap on hexagonal sides has same mesh structure
Define gap structure for each assembly

Type: Single derived type

Associated Input
T_HEXASSY_TYPE(:)%L_GAP_STRUCTURE

Example

```
&geometry
  T_HEXASSY_LATTICE%I_NUMPLANE = 4

  T_HEXASSY_LATTICE%R_ASSEMBLY_OUTER_FLAT2FLAT = 14.6860

  T_HEXASSY_TYPE(1)%R_ASSEMBLY_INNER_FLAT2FLAT = 12.2707
  T_HEXASSY_TYPE(1)%L_GAP_STRUCTURE = T
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_NUMGAP = 4
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_NUMGAP_DIV = 10
  T_HEXASSY_TYPE(1)%T_GAPTYPE%R_GAP_DIMENSION(1:4) = 0.301 0.4023 0.301 0.201
  ! sum(R_GAP_DIMENSION(1:4)) =
  !                   0.5 * (R_ASSEMBLY_OUTER_FLAT2FLAT
  !                        - R_ASSEMBLY_INNER_FLAT2FLAT)
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_GAP_REGION(1:4, 1) = 1 1 2 1 ! 1st axial mesh
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_GAP_REGION(1:4, 2) = 1 1 2 1 ! 2nd axial mesh
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_GAP_REGION(1:4, 3) = 1 1 2 1 ! 3rd axial mesh
  T_HEXASSY_TYPE(1)%T_GAPTYPE%I_GAP_REGION(1:4, 4) = 1 1 2 1 ! 4th axial mesh
/
```

T_HEXASSY_TYPE(:)%T_GAPTYPE%I_NUMGAP

Description: Number of meshes for gap structure in vertical direction (Vertical direction refers that a direction perpendicular to hexagon side; see Figure 4.7)

Type: Single integer

Associated Input
T_HEXASSY_TYPE(:)%T_GAPTYPE%R_GAP_DIMENSION(:, :)

Example

```
Given in 'T_HEXASSY_TYPE(:)%T_GAPTYPE' description
```

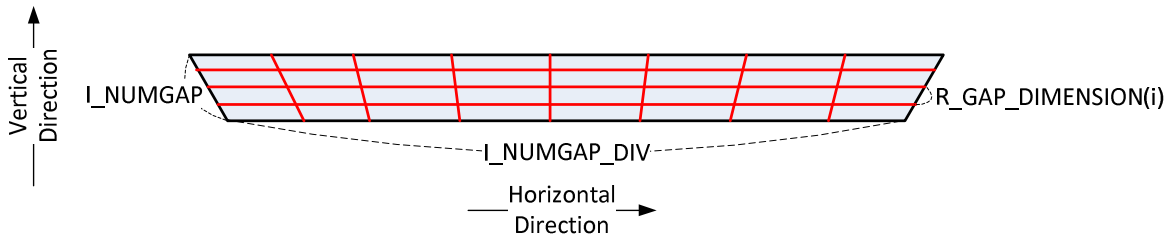


Figure 4.7 Gap Structure Specification

T_HEXASSY_TYPE(:)%T_GAPTYPE%I_NUMGAP_DIV
Description: Number of mesh for gap structure in horizontal direction (Parallel direction refers to a direction parallel to hexagon side; see Figure 4.7)
Used to determine dimension of mesh in horizontal direction
(Important) Use even number only
Type: Single Integer
Example

```
Given in 'T_HEXASSY_TYPE(:)%T_GAPTYPE' description
```

T_HEXASSY_TYPE(:)%T_GAPTYPE%R_GAP_DIMESION(:)
Description: Dimension of meshes for gap structure in vertical direction
From inner to outer direction with respect to assembly center (see Figure 4.7)
Type: 1D real array
(Array Size) T_HEXASSY_TYPE(:)%T_GAPTYPE%I_NUMGAP
(1st Array Index) Vertical direction index
Example

```
Given in 'T_HEXASSY_TYPE(:)%T_GAPTYPE' description
```

T_HEXASSY_TYPE(:)%T_GAPTYPE%I_GAP_REGION(:, :)
Description: Region index mapping for meshes
From inner to outer direction with respect to assembly center and from bottom to top (See Figure 4.7)
Type: 2D integer array
(Array Size) T_HEXASSY_TYPE(:)%T_GAPTYPE%I_NUMGAP
(1st Array Index) Vertical direction index (from inner to outer)
(2nd Array Index) Axial mesh (plane) index (from bottom to top)
Example

```
Given in 'T_HEXASSY_TYPE(:)%T_GAPTYPE' description
```

Assembly Configuration in Triangular Lattice Structure for Hexagonal Assembly

T_HEXASSY_LATTICE%C_GEOMETRY_TYPE
Description: Lattice type
(Pre-defined Input Arguments) 'lattice_360'

'reflective_120', 'reflective_60'
'rotational_120', 'rotational_60'
Lattice defined for 360 deg.→ 'lattice_360'
Lattice defined in 120 deg. with reflective symmetry → 'reflective_120'
Lattice defined in 60 deg. with reflective symmetry → 'reflectuve_60'
Lattice defined in 120 deg. with rotational symmetry → 'rotional_120'
Lattice defined in 60 deg. with rotational symmetry → 'rotional_60'
(Default) 'lattice_360'

Type: Character

Associated Input

T_HEXASSY_LATTICE%I_ASSYTYPE_MAP(:, :)

Example

```
Given in 'T_HEXASSY_LATTICE%I_ASSYTYPE_MAP' description
```

T_HEXASSY_LATTICE%I_ASSYTYPE_MAP (:, :)

If T_HEXASSY_LATTICE%C_GEOMETRY_TYPE = 'lattice_360'

Description: Assembly type arrangement for triangular lattice defined for 360 deg.
Fill every possible locations of triangular lattice
(Important) Not allow dummy assembly in current version
Given in 2D array form
Mapping for 2D array and triangular is given in Figure 4.8
Illustration of mapping is given in Figure 4.9

Type: 2D integer array
(Array Size) (2*T_HEXASSY_LATTICE%I_NUMRING-1)
× (2*T_HEXASSY_LATTICE%I_NUMRING-1)
(1st Array Index) Index for x-direction
(2nd Array Index) Index for y-direction

Option: Positive integer
Null cell type : '0'

Associated Input

T_HEXASSY_LATTICE% I_NUMRING
T_HEXASSY_LATTICE_C_GEOMETRY_TYPE
T_HEXASSY_LATTICE% I_NUMASSYTYPE
T_HEXASSY_TYPE(:)

Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 3
 T_HEXCELL_LATTICE%I_ASSYTYPE_MAP(1:5, 1:5) =
                                    2   2   2   0   0
                                  2   1   1   2   0
                                2   1   3   1   2
                              0   2   1   1   2
                            0   0   2   2   2
/
```

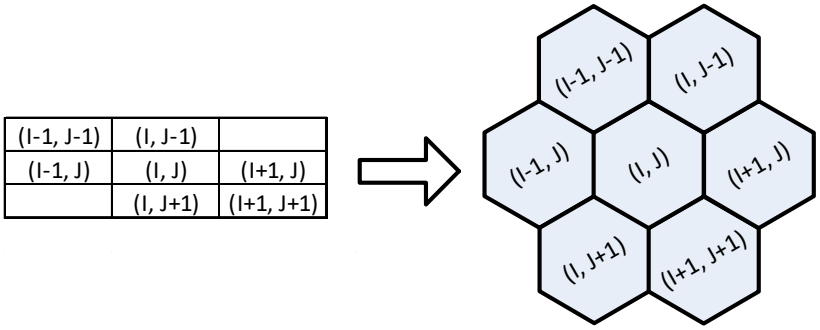


Figure 4.8 Mapping of Hexagonal Assemblies

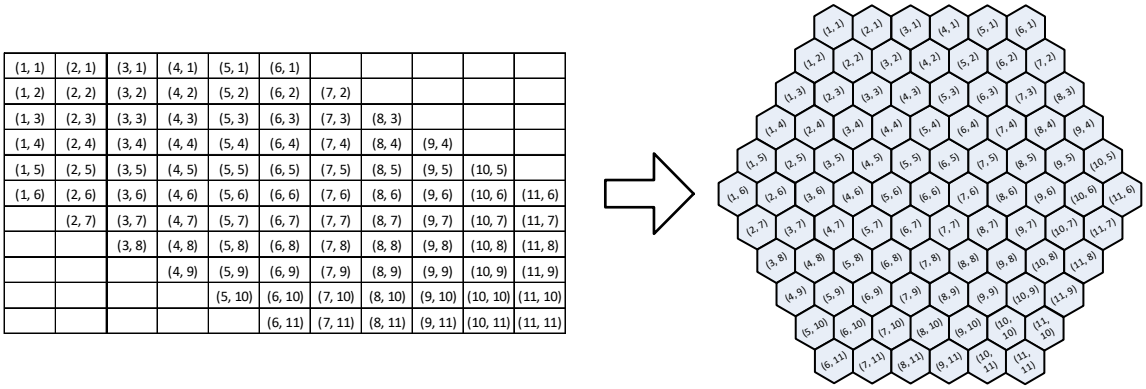


Figure 4.9 An example of 2D Mapping of Hexagonal Assemblies with 'lattice_360'

If T_HEXASSY_LATTICE%C_GEOMETRY_TYPE = 'reflective_120' or 'rotational_120'

Description: Assembly type arrangement for triangular lattice defined for 120 deg.
Fill every possible locations of triangular lattice
(Important) Not allow dummy assembly in current version
Given in 2D array form
Mapping for 2D array and triangular is given in Figure 4.8
Illustration of mapping is given in Figure 4.10
(Default) N/A

Type: 2D integer array
(Array Size) (T_HEXASSY_LATTICE%I_NUMRING)
(T_HEXASSY_LATTICE%I_NUMRING)
(1st Array Index) Index for x-direction
(2nd Array Index) Index for y-direction

Option: Positive integer
Null cell type : '0'

Associated Input
T_HEXASSY_LATTICE%I_NUMRING
T_HEXASSY_LATTICE%C_GEOMETRY_TYPE
T_HEXASSY_LATTICE%I_NUMASSYTYPE

T_HEXASSY_TYPE(:)

Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 5
 T_HEXCELL_LATTICE%I_REGION_ASSYTYPE_MAP(1:5, 1:5) =
                                      1   1   1   1   2
                                    1   1   3   1   2
                                  1   3   1   1   2
                                1   1   1   1   2
                              2   2   2   2   2
/
! Following input is incorrect
! (1,6), (6, 1) and (6, 6) are not defined
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 5
 T_HEXCELL_LATTICE%I_REGION_ASSYTYPE_MAP(1:5, 1:5) =
                                      1   1   1   1   0
                                    1   1   3   1   2
                                  1   3   1   1   2
                                1   1   1   1   2
                              0   2   2   2   0
/
```

(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)
(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)
(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)
(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)
(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)
(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)

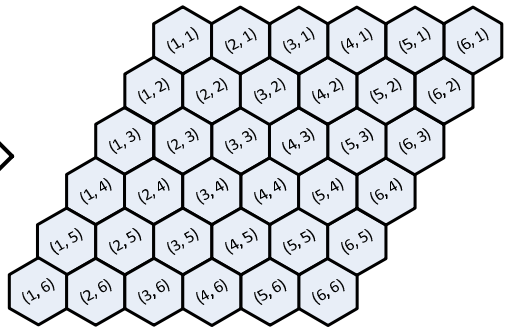


Figure 4.10 An example of 2D Mapping of Hexagonal Assemblies with ‘reflective_120’ or ‘rotational_120’

If T_HEXASSY_LATTICE%C_GEOMTRTY_TYPE = ‘reflective_60’ or ‘rotational_60’

Description: Assembly type arrangement for triangular lattice defined for 60 deg.
Fill every possible locations of triangular lattice
(Important) Not allow dummy assembly in current version
Given in 2D array form
Mapping for 2D array and triangular is given in Figure 4.8
Illustration of mapping is given in Figure 4.11

Type: 2D integer array
(Array Size) (T_HEXASSY_LATTICE%I_NUMRING)
× (T_HEXASSY_LATTICE%I_NUMRING)
(1st Array Index) Index for x direction
(2nd Array Index) Index for y direction

Option: Positive integer

Null cell type : ‘0’

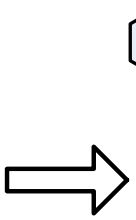
Associated Input

T_HEXASSY_LATTICE%I_NUMRING
T_HEXASSY_LATTICE%C_GEOMETRY_TYPE
T_HEXASSY_LATTICE%I_NUMASSYTYPE
T_HEXASSY_TYPE(:)

Example

```
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 5
 T_HEXCELL_LATTICE%I_ASSYTYPE_MAP(1:5, 1:5) =
                                      1  1  1  1  2
                                   0  1  3  1  2
                                0  0  1  1  2
                             0  0  0  1  2
                          0  0  0  0  2
/
! Following input is incorrect
! (6, 1) and (6, 6) are not defined
&geometry
 T_HEXCELL_LATTICE%I_NUMRING = 6
 T_HEXCELL_LATTICE%I_ASSYTYPE_MAP(1:6, 1:6) =
                                      1  1  1  1  2  0
                                   0  1  3  1  2  2
                                0  0  1  1  2  2
                             0  0  0  1  2  2
                          0  0  0  0  2  0
/
```

(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)
	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)
		(3, 3)	(4, 3)	(5, 3)	(6, 3)
			(4, 4)	(5, 4)	(6, 4)
				(5, 5)	(6, 5)
					(6, 6)



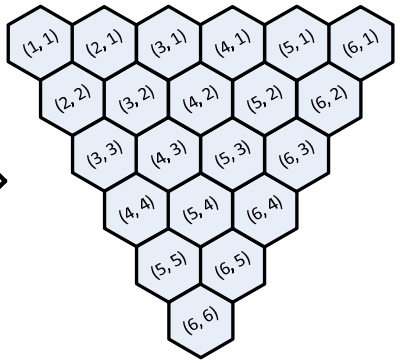


Figure 4.11 An example of 2D Mapping for Hexagonal Assembly with ‘reflective_60’ or ‘rotational_60’

Ray Tracing Parameters

T_CONTROL_RAY%R_DEL

Description: Ray spacing
(Default) 0.05

Type: Single real

Example
```
&geometry
 T_CONTROL_RAY%R_DEL = 0.01
/
```

T_CONTROL_RAY%I_AZIMUTHAL
Description: Number of azimuthal angles between 0 to 180 deg.
(Important) mod(I_AZIMUTHAL, 3) = 0
(Default) 6
Type: Single integer
Example
```
&geometry
 T_CONTROL_RAY%I_AZIMUTHAL = 12
/
```

T_CONTROL_RAY%I_POLAR
Description: Number of polar angles between 0 to 90 deg.
(Default) 4
Type: Single integer
Example
```
&geometry
 T_CONTROL_RAY%I_POLAR = 0.01
/
```

T_CONTROL_OMP%I_NTHREAD
Description: Number of threads for parallel processing with OpenMP
(Default) 1
Type: Single integer
Example
```
&geometry
 T_CONTROL_RAY%I_NTHREAD = 24
/
```

T_CONTROL_OPT%L_CMFDACC
Description: Use of the CMFD acceleration
(Default) False
Type: Logical
Example
```
&geometry
 T_CONTROL_OPT%L_CMFDACC = T
/
```

T_CONTROL_OPT%I_CMFDACC_NUMGROUP
Description: Number of energy groups for the CMFD acceleration (≤ the number of UFGs)
(Default) 200
Type: Single integer
Example
```
&geometry
 T_CONTROL_OPT%I_CMFDACC_NUMGROUP = 400
/
```

4.4.2 Rectangular Geometry Inputs

The MOC solver for rectangular geometry provides a flexible modeling capability based on a Cartesian grid structure. It can model typical square lattice structures such as fuel assemblies of lattice water reactors. It also can be utilized for modeling a drawer of ZPR and ZPPR critical assemblies. Currently, this module only supports a 2D domain and it is planned to extend for a 3D domain.

T_MOC_GEOMETRY%I_REGION_COMPOSITION (:)
Description: Arrangement of composition to regions
Type: 1D integer array
(Array Size) I_NUMBER_REGION
Associated Input
I_NUMBER_REGION (in the $control block)
Example

```
&control
 I_NUMBER_REGION = 5
/
&geometry
 T_MOC_GEOMETRY%I_REGION_COMPOSITION(1:5) = 1 2 3 4 5
/
```

T_MOC_GEOMETRY%I_NX
Description: Number of nodes in the x-direction
Type: Single integer
Example

```
&geometry
 T_MOC_GEOMETRY%I_NX = 3
/
```

T_MOC_GEOMETRY%I_NY
Description: Number of nodes in the y-direction
Type: Single integer
Example

```
&geometry
 T_MOC_GEOMETRY%I_NY = 3
/
```

T_MOC_GEOMETRY%I_NCELL
Description: Number of cells in the cell map
Type: Single integer
Example

```
&geometry
 T_MOC_GEOMETRY%I_NCELL = 2
/
```

T_MOC_GEOMETRY%I_NCELLTYPE
Description: Number of cell types in the cell map
The number of cell types should be equal to or less than the number of cells
This allows a user to define a cell type and use it repeatedly in different regions
Type: Single integer

Example

```
&geometry
 T_MOC_GEOMETRY%I_NCELLTYPE = 2
/
```

T_MOC_GEOMETRY%R_X(:)

Description: Location of grids in the x-direction
Type: Real 1D array
Example

```
&geometry
 T_MOC_GEOMETRY%R_X(1:3) = -1.26 0.0 1.26
/
```

T_MOC_GEOMETRY%R_Y(:)

Description: Location of grids in the y-direction
Type: Real 1D array
Example

```
&geometry
 T_MOC_GEOMETRY%R_Y(1:3) = -1.26 0.0 1.26
/
```

T_MOC_GEOMETRY%I_CELL_MAP(:,:)

Description: Loading of cells in the problem
Type: Integer 2D array
Associated Input

T_MOC_GEOMETRY%I_NX
T_MOC_GEOMETRY%I_NY

Example

```
&geometry
 t_moc_geometry%i_cell_map    (1:4,1:4)    =
                1 2 2 2
                1 2 2 2
                3 4 4 3
                3 4 4 3
/
```

T_MOC_GEOMETRY%I_CELLTYPE_MAP(:,:)

Description: Loading of cell types in the problem
The area in a rectangular shape occupied by the same cell type number is considered as a cell type region. The example below can produce the geometry shown in Figure 4.12 by defining cell types using the T_MOC_GEOMETRY%T_CELLTYPE(:) card with ‘annular’ and ‘rectangular’
Type: Integer 2D array
Associated Input

T_MOC_GEOMETRY%I_NX
T_MOC_GEOMETRY%I_NY
T_MOC_GEOMETRY%T_CELLTYPE(:)

Example

```
&geometry
 t_moc_geometry%i_celltype_map    (1:6,1:6)    =
                1 1 2 2 2 2
```

```
                  1 1 2 2 2 2
                  3 4 4 4 4 3
                  3 4 4 4 4 3
                  3 4 4 4 4 3
                  3 4 4 4 4 3
/
```

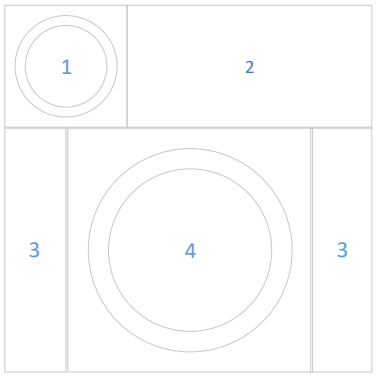


Figure 4.12 An Example of a Rectangular Geometry Problem for MOC

T_MOC_GEOMETRY%T_CELLTYPE(:)%C_TYPE
Description: Name of cell type
Type: Character
Option: annular / rectangular
Associated Input
T_MOC_GEOMETRY%I_NCELLTYPE
T_MOC_GEOMETRY%T_CELLTYPE(:)%R_RADIUS(:)
T_MOC_GEOMETRY%T_CELLTYPE(:)%I_REGION(:)
Example

```
&geometry
  t_moc_geometry%t_celltype(1)%c_type       = 'annular'
  t_moc_geometry%t_celltype(1)%r_radius     =   .418 .475
  t_moc_geometry%t_celltype(1)%i_region     =      1    2   3
/
```

T_MOC_GEOMETRY%T_CELLTYPE(:)%R_RADIUS(:)
Description: Radii of annual rings
Type: Real 1D array
Associated Input
T_MOC_GEOMETRY%I_NCELLTYPE
T_MOC_GEOMETRY%T_CELLTYPE(:)%R_RADIUS(:)
T_MOC_GEOMETRY%T_CELLTYPE(:)%I_REGION(:)
Example

```
See T_MOC_GEOMETRY%T_CELLTYPE(:)%C_TYPE
```

T_MOC_GEOMETRY%T_CELLTYPE(:)%I_REGION(:)
Description: Region assignment of annual rings
Type: Integer 1D array
Associated Input

T_MOC_GEOMETRY%I_REGION_COMPOSITION(:)
T_MOC_GEOMETRY%I_NCELLTYPE
T_MOC_GEOMETRY%T_CELLTYPE(:)%R_RADIUS(:)
T_MOC_GEOMETRY%T_CELLTYPE(:)%I_REGION(:)

Example

See T_MOC_GEOMETRY%T_CELLTYPE(:)%C_TYPE

4.5 Group Structure Block

(Precondition)

```
&control
     c_group_structure = user
/
```

&groupstructure

Name	Type	Option	Description
i_nufg_in_bg	integer array	0 (D)	Number of ultrafine groups in each broad group. The sum of the number of ultrafine groups in broad group should be the total number of ultrafine groups (currently 2082).
i_number_group	integer	0 (D)	Number of broad groups.
r_energy	real array	0 (D)	Upper energy boundary of broad groups. The energy should be one of the ultrafine groups: the broad group boundaries must fall on ultrafine group boundaries. If there is no ultrafine group energy that matches the energy given, the nearest ultrafine group is taken.

This block is activated when c_group_structure is set to “user” in the &control block.

Either i_nufg_in_bg or r_energy should be input.

Inputs of i_nufg_in_bg(:) and r_energy(:) beyond i_number_group are ignored.

(Example)

```
&groupstructure
     i_number_group = 4
     i_nufg_in_bg   = 3*500  582
/
```

```
&groupstructure
     i_number_group = 4
     r_energy       = 14.19e6   1.e5   1e3   4.
/
```

```
&groupstructure
     i_number_group = 1041
     i_nufg_in_bg   = 1041*2
/
```

4.6 Library Block

&library

Name	Type	Option	Description
c_gammalibdir	character (len=200)	"." (D)	Directory of the gamma libraries (optional) (Example) c_gammalibdir = "/home/user/mcclib/gamma"
c_mcclibdir	character (len=200)	"." (D)	Directory of the MC^2 libraries (**required**) (Example) c_mccdir = "/home/user/mcclib/mcc"
c_pendflibdir	character (len=200)	"." (D)	Directory of the PENFD libraries (optional: unnecessary for a normal calculation) (Example) c_pendflibdir = "/home/user/mcclib/pendf"
c_pwlibdir	character (len=200)	Blank (D)	Directory of the pointwise (PW) cross sections. (**required**) Up to 5 directories can be input to search for existing PW cross sections. If no PW file exists, a new file is created at the last directory. The working directory or current directory is recommended to be input as the last one for which the user has write permission. (Example) c_pwlibdir = "/home/user/mcclib/lib.pw.200" "/home/clee/mcc3_lib.pw.200" "."
c_twodantexe	character (len=200)	"." (D)	A file name of the TWODANT executable together with its directory path. (optional) (Example) c_twodnatexe= "/home/user/TWODANT/twodant.x"

The libraries required for a MC^2-3 job are the MC^2 libraries discussed in Section 3. The fission spectrum (chi) matrix and inelastic scattering data libraries are also necessary for improved solutions. Since these data are already included in the MC^2 MCCF9 and MCCF10, the directories for them do not need to be additionally defined. The gamma directory should be defined using "c_gammalibdir" when the gamma cross section generation is requested. The PENDF libraries are necessary only when PENDF files are requested by the user for self-shielding resolved resonance or cross sections above the resonance energy range. The executable file of TWODANT with a full directory path is needed only when the TWODANT calculation is requested with the "l_twodant" card of the &control block. MC^2-3 runs without the pointwise cross section libraries, but they are very useful to make it run fast by making use of the existing isotopic pointwise cross sections at the same temperature condition without reconstructing them again.

In a namelist, a slash "/" is interpreted by the end of a namelist section, and thus a directory name including "/" should be enclosed with double quotations.

(Example)

```
&library
     c_mcclibdir   ="/mcc3_directory/lib.mcc.e70"
! Up to 5 directory inputs are allowed for c_pwlibdir
! The last one may need write permission
! in case that no PW files exist in the directories
     c_pwlibdir    ="/mcc3_directory/lib.pw.200.e70", "."
     c_gammalibdir ="/mcc3_directory/lib.gamma.e70"
     c_pendflibdir ="/mcc3_directory/lib.pendf"
     c_twodantexe  ="/twodant_directory/twodant.x"
/
```

4.7 Material Block

(Precondition)

```
&control
     i_number_region = 5     ! 5 homogeneous regions or
                             ! 1D problem with 5 regions
/
```

(Possible precondition)

```
&output
     l_edit_leakage = T                ! i_source_region
     l_edit_macroxs = T                ! c_composition_name
/
```

&material

Name	Type	Option	Description
c_composition_name	character (len=8)	blank (D)	Name of mixture or region. If no name is defined, a name is automatically created as REG_[][] where [][]

			represents a two-digit region number from 01 to 99. This card is needed for naming macroscopic cross sections. When microscopic cross sections are requested as usual, this card would not be used since the isotope names of each region are given by the user through the "t_composition" card. (Relevant blocks and cards) t_composition (&material) (Example) c_composition_name(1) = INCORE
i_externalspectrum	integer array	0 (D)	The data set number in the external spectrum file. This would not be used for the hyperfine-group external spectrum file. (Relevant blocks and cards) c_externalspectrum_ufg (Example) i_externalspectrum(1) = 3
i_source_region	integer array	0 (D)	External source region. If any number greater than 0 is input, the code searches for "output.leakage_ufg.r[][] in the working directory. If the source file is not found, a warning message is issued. Up to 99 is allowed. The user can generate leakages from each region and elect to use them in the following regions. To do this, define l_edit_leakage = T in the &output block and l_external_source = T in the &control block. (Relevant blocks and cards) l_edit_leakage (&output) l_external_source (&control) (Example)

			When the leakage from Composition 1 is the external source of Compositions 3 and 4, i_source_region (3) = 1 i_source_region (4) = 1
t_composition	type (char*6, char*6, real, real)	blank, 0 (D)	Composition of a mixture including: - isotope name in the MC^2 library (see Appendices C and D), - user-specified isotope name, - number density, - temperature in K An isotope name including blanks should be enclosed in double quotation marks. Those blanks given in the user-defined isotope name are replaced by "_" in ISOTXS. Up to 99 isotopes are allowed in a mixture or region and maximum 99 mixtures or regions can be defined in an input as long as memory is allowed. (Relevant blocks and cards) i_number_region (&control)

This block is required all the time except for ISOTXS management cases.

The order of composition cards as well as other cards does not need to be sequential.

(Example)

```
&material
     c_composition_name(1)= ICORE
     t_composition(:,1)=  U235_7    U-235I   4.4482E-02   500.
                          U238_7    U-238I   2.7038E-03   300.
                         "NA23 7"  "NA23 I"  4.8869E-04   600.

     C_composition_name(2) = OCORE
     t_composition(:,2)=  U235_7    U-235°   3.4610E-04   300.
                          U238_7    U-238°   4.7721E-02   500.
                          U234_7    U-234°   2.6438E-06   800.
/
```

```
&output
     l_edit_leakage    = T
/
&material
     t_composition( :,1)=  U235_7    U-235I   4.4482E-02   500.
                           U238_7    U-238I   2.7038E-03   300.

     T_composition( :,2)=  PU2397    PU239O   4.4482E-02   500.
                           U238_7    U-238O   2.7038E-03   300.
```

```
     I_source_region(3)= 1                  ! leakage from Composition 1
     t_composition( :,3)=  FE56_7    FE56AR   1.4610E-04   300.

     I_source_region(4)= 2                  ! leakage from Composition 2
     t_composition( :,4)=  FE56_7    FE56RR   1.4610E-04   300.
/
```

When using the existing leakage files (l_edit_leakage = F) instead of the leakage files to be created during the calculation, the user needs to rename those leakage files as follows (see the example below):

mv leakage1 output.leakage_ufg.r01
mv leakage2 output.leakage_ufg.r02

```
&output
     l_edit_leakage    = F              ! default
/

&material
     i_source_region(1) = 1             ! leakage from the existing file 1
     t_composition( :,1)=  FE56_7    FE56AR   1.4610E-04   300.

     I_source_region(2) = 2             ! leakage from the existing file 2
     t_composition( :,2)=  FE56_7    FE56RR   1.4610E-04   300.
/
```

4.8 Output Block

&output

Name	Type	Option	Description
c_check_memory	character (len=8)	short(D) long	Edit memory information. Short: print out total memory requirements long : print out the detailed memory information.
c_dlayxs_file	character (len=50)	DLAYXS (D)	File name for DLAYXS (Relevant blocks and cards) l_delayedneutron (&control) c_dlayxs_conversion (&control)
c_homogenizationbase	character (len=8)	user(D) library	Spatial homogenization is performed by the user-specified name of isotope (user) or the isotope name given from the library (library). If c_homogenizationbase is set to “library”, then the first-coming user-specified name of the same isotopes over

			regions will be selected as a representative name of the isotope. (Relevant blocks and cards) t_composition
c_isotxs_file	character array (len=50)	blank (D)	Additional ISOTXS files to be merged together with those which are to be generated in the current job. Up to 99 files can be handled. (Relevant blocks and cards) c_isotxs_conversion (&control)
c_isotxs_isotopeid	character (len=2)	blank (D)	Two-character composition ID to be written in the 5th and 6th location of all the isotope names belonging to a composition. Any characters placed at the locations are overwritten. Therefore, when this card is used, make sure that all isotopes in the composition should be able to be distinguished by the first four characters. Otherwise, more than one isotope with the same ID will be created. (Relevant blocks and cards) t_composition (&material) (Example) &output c_isotxs_isotopeid = IC / &material t_composition(:,1)= U238_7 U238 0.1 300 / Then, the ID of U238 to be written in the ISOTXS file becomes U238IC.
c_lump_name	character array (len=50)	blank (D)	Name of macroscopic cross sections that are merged with the isotopes listed in the "t_lump_isotope" card. This is normally used to generate the lumped fission products. (Relevant blocks and cards) t_composition (&material) t_lump_isotope (&output)

			(Example) c_lump_name(1) = LFP_A1
c_pmatrx_file	character array (len=50)	blank (D)	Additional PMATRX files to be merged together with those which are to be generated in the current job. Up to 99 files can be handled. (Relevant blocks and cards) c_pmatrx_conversion (&control)
c_pw_file	character array (len=50)	blank (D)	Name of the PW file to be converted by the direction given from the "c_pw_conversion" of the &control block. (Relevant blocks and cards) c_pw_conversion (control) c_pwlibdir (&library)
i_reactionrate_group	integer	0 (D)	Number of groups for editing reaction rates. This requires defining the "i_reactionrate_ufg" card. If this is undefined or set to zero, then the code automatically defines the energy boundaries of reaction rate edits based on unresolved and resolved resonance energy boundaries of the isotopes in the composition. (Relevant blocks and cards) i_reactionrate_ufg (&output) l_edit_reactionrate (&output)
i_reactionrate_ufg	integer array	0 (D)	Starting ultrafine group number corresponding to the number of reaction rate groups defined in the "i_reactionrate_group" card. (example) i_reactionrate_group = 5 i_reactionrate_ufg = 1, 200, 500, 1000, 1500
l_edit_chi	logical	false (D)	Create fission spectrum distributions for ultrafine groups (output.chi_ufg) and broad groups (output.chi_bg).
l_edit_flux	logical	true (D)	Create ultrafine group fluxes (output.flux_ufg) and broad group fluxes

			(output.flux_bg)
l_edit_leakage	logical	false (D)	Create ultrafine group leakages (output.leakage_ufg.r[][]) where [][] represents a two-digit region number from 01 to 99 (Relevant blocks and cards) i_source_region (&material) l_external_source (&control)
l_edit_reactionrate	logical	false (D)	Create total reaction rates as well as partial ones for capture, fission, elastic scattering, inelastic scattering, and (n,2n). The data is written in the standard output file. (Relevant blocks and cards) i_reactionrate_ufg (&output) l_reactionrate_group (&output)
l_edit_rzmflx	logical	false (D)	Create output.rzmflx in an ASCII format
l_edit_xs	logical	true (D)	Create microscopic cross sections with ultrafine groups (output.microxs_ufg.r[][]) and broad groups (output.microxs_bg.r[][]) where [][] represents a two-digit region number from 01 to 99.
l_isotxs_ascii	logical	false (D)	Create ISOTXS files with ASCII format: ISOTXS.r[][].ascii, ISOTXS.merged.ascii, where [][] represents a two-digit region number from 01 to 99.
l_isotxs_macroxs	logical	false (D)	Create ISOTXS with macroscopic cross section. A composition name is determined as REG_[][] where [][] represent a region number from 01 to 99.
l_gamiso_ascii	local	false (D)	Create GAMISO files with ASCII format: GAMISO.r[][].ascii, GAMISO.merged.ascii, where [][] represents a two-digit region number from 01 to 99. Note that the format of GAMISO is the same as ISOTXS. To merge GAMISO files, use the input cards for ISOTXS (i.e., c_isotxs_conversion & c_isotxs_file)
l_pmatrx_ascii	local	false (D)	Create PMATRX files with ASCII format: PMATRX.r[][].ascii, PMATRX.merged.ascii, where [][] represents a two-digit region number from 01 to 99.

t_isotope_namechange	type (char*6, char*6)	blank (D)	Change of isotope names. (Example) (old) (new) t_isotope_namechange = U235I1 U235I2 PU39I1 PU39I2
t_lump_isotope	type (char*6, real)	blank, 0 (D)	A set of {isotope, number density} to be lumped to macroscopic cross sections. The name of the lumped cross section is defined in the “c_lump_name” card. Up to 99 lumped compositions can be defined. Up to 500 isotopes can be defined for a lumped composition. The isotopes listed in the “t_lump_isotope” card should exist in the “t_composition” card. To avoid confusion, only a single region is allowed in a job (i.e., i_number_region=1). Note that the isotope name should be found in either the library or the user-specified. If a library name of isotope is used more than once with different user-specified names, you should use the user-specified name of the isotope for t_lump_isotope. Accordingly, to avoid confusion, it is recommended to use the user-specified isotope name for t_lump_isotope rather than the isotope name in the library. (Relevant blocks and cards) t_composition (&material) c_lump_name (&output) (Example) &material t_composition(:,1)= GE73_7 GE73_1 1.0 300. GE73_7 GE73_2 0.5 300. GE74_7 GE74_1 0.1 300. GE76_7 GE76_1 0.2 300. /

			&output c_lump_name(2)= LUMP2 t_lump_isotope(:,2)= GE73_2 4.51000E-06 GE74_1 8.63000E-06 GE76_1 3.01000E-05 /

(Example)

```
&output
     l_edit_flux           = F
     l_edit_leakage        = T
     l_edit_xs             = F
     l_isotope_ascii       = T
     l_isotope_isotopename = T
     l_isotope_macroxs     = T
     c_isotxs_file         = ISOTXS.blanket  ! additional ISOTXS to merge
                             ISOTXS.shield
                             ISOTXS.crod_A
/
```

For generating multiple lumped fission products,

```
&control
     i_number_region    = 1
     c_geometry_type    = mixture    ! default
/
&material
     t_composition(:, 1)=  U234_7  U-234A      2.18633E-07   850.0
                       U235_7  U-235A      5.41647E-04   850.0
                       U238_7  U-238A      8.59044E-03   850.0
                       PU2397  PU239A      4.59984E-04   850.0
                       PU2407  PU240A      2.25392E-05   850.0
                       PU2417  PU241A      7.17045E-07   850.0
                       PU2427  PU242A      2.06030E-08   850.0
                       GE73_7  GE73_5      1.00000E-08   850.0  ! infinite dilute
                       GE74_7  GE74_5      1.00000E-08   850.0
                       GE76_7  GE76_5      1.00000E-08   850.0
                       AS75_7  AS75_5      1.00000E-08   850.0
                       SE76_7  SE76_5      1.00000E-08   850.0
                       SE77_7  SE77_5      1.00000E-08   850.0
/
&output
     c_lump_name     (1)   = LFU235                 ! macroscopic xs name
     t_lump_isotope(;,1)   = GE73_7   5.23995E-06    ! fission yield from U-235
                             GE74_7   1.18999E-05
                             GE76_7   1.75758E-04
                             AS75_7   3.52997E-05
                             SE76_7   5.47995E-09
                             SE77_7   3.18077E-04

     c_lump_name     (2)   = LFU238                 ! macroscopic xs name
     t_lump_isotope(:,2)   = GE73_7   2.06001E-06    ! fission yield from U-238
                             GE74_7   2.77002E-06
                             GE76_7   8.00005E-06
                             AS75_7   4.62003E-06
                             SE76_7   5.60004E-11
```

```
                              SE77_7    3.30102E-05
/
```

4.9 TWODANT Block

Except for a few cards, the whole block is input to TWODANT. Therefore, refer to the TWODANT manual [6] for detailed input descriptions. Since the cards "niso", "ngroup", "matls", "assign", and "chi" are dependent upon cross sections to be generated by MC^2-3, each of them should be written in a separate line, as shown in the example. MC^2-3 will replace these lines with appropriate input data. Especially, the region-dependent fission spectrum will be written at the "chi" location.

The card "isct" is recommended to be consistent with the scattering order specified in the "i_scattering_order" card of the &control block so that all high-order moments are condensed with the correct spectrum.

Since the purpose of running TWODANT is to generate the zone-wise flux and moment solutions, it is important to ensure that both rmflux and rzmflx cards are turned on: i.e., rmflux=1 of Block V and rzmflx=1 of Block VI.

&twodant

(Possible precondition)

```
&library
     c_twodantexe    = "&twodantdir/twodant.exe"
/
&control
     l_twodant       = T
     l_twodant_group = UFG               ! default
/
&output
     l_edit_macroxs  = T                 ! c_composition_name
/
```

(Example)

```
&twodant
     1

niso=,
ngroup=,
mt=2,nzone=2,im=2,
igeom=3,isn=16,it=130,maxlcm=18000000,maxscm=35000000,
t
 xmesh=0.0 6.1156  24.1242 ,
 xints=30 100,
 zones=1 2;
t
lib=isotxs,balxs=0,
t
matls=
assign=
t
isct=3,ievt=1,ibl=1,ibr=0,iquad=-2,
```

```
ith=0,fluxp=0,xsectp=0,fissrp=0,sourcp=0,angp=0,geomp=0,
influx=0,norm=1.000e+00,epsi=0.0010,epso=.000001,
oitm=100,iitm=100,iitl=100,insors=0,raflux=0,rmflux=1
chi=,
t
rzmflx=1
t
/
```

5 SAMPLE INPUTS

5.1 Ultrafine Group Calculation for Mixture

Multiple homogeneous regions (i.e., compositions) can be defined in a single job as long as the computer memory permits, so that the cross sections for all regions can be generated in a single job and then stored in a single ISOTXS file. The user can also generate an ISOTXS file for each single region and then merge the ISOTXS files.

The blocks essential to the mixture calculation are “library”, “control”, and “material.” If “l_isotxs_ascii” is set to true, an additional ISOTXS file with ASCII format is produced.

```
lib={MC2-3 directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc.e70"
     c_pwlibdir   ="$lib/lib.pw.200.e70","."
!    c_pendflibdir="$lib/lib.pendf"
/
&control
     c_group_structure       =ANL33
     i_number_region         =2
     c_geometry_type         =mixture
/
&output
     l_edit_flux             =T
!    l_isotxs_macroxs        =F
!    l_isotxs_ascii          =T
/
&material
     t_composition(:,1)= U235_7     U-235I    1.1530E-03    300.
                         U236_7     U-236I    5.6000E-06    300.
                         U238_7     U-238I    5.8010E-03    300.
                         U234_7     U-234I    1.1000E-05    300.

     t_composition(:,2)= U235_7     U-235O    1.1490E-03    300.
                         U236_7     U-236O    5.6000E-06    300.
                         U238_7     U-238O    5.7840E-03    300.
                         U234_7     U-234O    1.1000E-05    300.
                         NA23_7    "NA23 O"   9.2020E-03    300.  ! NA23_O
                         O16__7    "O-16 O"   1.4740E-02    300.  ! O-16_O
                         FE54_7    "FE54 O"   8.1142E-04    300.  ! FE54_O
                         FE56_7    "FE56 O"   1.2831E-02    300.  ! FE56_O
                         FE57_7    "FE57 O"   3.0778E-04    300.  ! FE57_O
                         FE58_7    "FE58 O"   3.9172E-05    300.  ! FE58_O
/
EOF

$lib/mcc3.x

mv output           $0.out
mv output.flux_bg   $0.flux_bg
mv output.flux_ufg  $0.flux_ufg
mv ISOTXS.merged    $0.ISOTXS
```

5.2 Ultrafine Group Calculation for Mixture with TWODANT

For this mixture calculation with TWODANT, two sequential jobs are required. The first one is to provide region-wise ultrafine group macroscopic cross sections for use in TWODANT and the second job is to use the TWODANT flux and moment solutions for group condensation to produce an ISOTXS file with region-wise broad group microscopic or macroscopic cross sections. Therefore, the first job must include the following options in the &control block:

i_number_group = ANL2082 (or ANL1041, whatever needed)
l_twodant = T,
c_twodant_group = BG (default).

The macroscopic cross sections are recommended to reduce the memory requirement in the TWODANT calculation using the following option in the &output block:

l_isotxs_macroxs = T.

With l_twodant on, the TWODANT calculation will automatically be performed at the end of the first job. If the TWODANT job fails (the failure sometimes happens due to insufficient memory assignment with *maxlcm* and *maxscm*), the user can rerun TWODANT separately until it successfully produces *rzmflx* which includes regular zone-averaged moment fluxes. For the separate execution of the TWODANT job, its input and ISOTXS that are produced from the first job are recommended to be saved as shown in the example below.

(the 1st job)

```
# ------------------------------------------------------------------------------
# the 1st job to generate TWODANT fluxes using sphere geo (igeom=3) and 1041groups
# 1) 0D calculation for 2 regions
# 2) TWODANT calculation for sphere geometry to create RZMFLX
# ------------------------------------------------------------------------------
# (main input)

lib={MC2-3 directory}
twodant.x={TWODANT executable with a full path}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
     c_twodantexe ="&twodant.x"
/
&control
     c_group_structure       =ANL1042
     i_number_region         =2
     l_external_inelasticpn  =T
     c_geometry_type         =mixture
     l_twodant               =T
     c_twodant_group         =BG      !(default)
/
&output
     l_isotxs_macroxs        =T       !(not necessary but good to save memory)
!    l_isotxs_ascii          =T
!    l_edit_flux             =T
/
&material
     t_composition(:,1)=  U235_7    U-235I   4.4482E-02        300.
                          U238_7    U-238I   2.7038E-03        300.
                          U234_7    U-234I   4.8869E-04        300.
     t_composition(:,2)=  U235_7    U-235O   3.4610E-04        300.
                          U238_7    U-238O   4.7721E-02        300.
```

```
                          U234_7    U-234O   2.6438E-06        300.
/
&twodant
     1

niso=,
ngroup=,
mt=2,nzone=2,im=2,
igeom=3,isn=16,it=130,maxlcm=18000000,maxscm=35000000,
t
 xmesh=0.0 6.1156  24.1242 ,
 xints=30 100,
 zones=1 2;
t
lib=isotxs,balxs=0,
t
matls=
assign=
t
isct=3,ievt=1,ibl=1,ibr=0,iquad=-2,
ith=0,fluxp=0,xsectp=0,fissrp=0,sourcp=0,angp=0,geomp=0,
influx=0,norm=1.000e+00,epsi=0.0010,epso=.000001,
oitm=100,iitm=100,iitl=100,insors=0,raflux=0,rmflux=1
chi=,
t
rzmflx=1
t
/
EOF

$lib/mcc3.x
mv output                 $0.out
mv ISOTXS.merged          $0.ISOTXS.1041
mv input.twodant          $0.twodantinp
cp rzmflx                 $0.rzmflx        # TWODANT output
```

In the second job, the following card should be added in the &control block:

c_externalspectrum_ufg = rzmflx (or user-specified file name)

Then, MC^2-3 will search for the *rzmflx* file during group collapsing. Other than this card, the following input cards should be turned off or changed as:

i_number_group = ANL33 (or whatever needed)
l_twodant = F

In order to store microscopic cross sections in ISOTXS, the macroscopic cross section option in the &output block should be turned off, too:

l_isotxs_macroxs = F.

The &twodant block should remain even though no TWODANT calculation is expected in the second job because the *rzmflx* file itself has no geometry information and thus the block is used to identify the geometry option that TWODANT used.

If the number of groups less than 2082 is selected in the first job and consequently in the TWODANT calculation, the neutron spectra between ultrafine group and the user-specified group will remain unadjusted in the second job where the final group condensation is conducted.

(the 2nd job)

```
# ------------------------------------------------------------------------------
# the 2nd job to collapse UFG to BG
# 1) 0d calculation for 2 region (repeated)
# 2) condense UFG to BG using TWODANT fluxes
# ------------------------------------------------------------------------------

lib={MC2-3 directory}

ln -sf {rzmflx file} rzmflx

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
     c_twodantexe ="&twodant.x"
/
&control
     c_group_structure       =ANL33
     i_number_region         =2
     c_geometry_type         =mixture
     c_externalspectrum_ufg  =rzmflx
!    l_twodant               =T
!    c_twodant_group         =BG      !(default)
/
&output
!    l_isotxs_macroxs        =T
!    l_isotxs_ascii          =T
!    l_edit_flux             =T
/
&material
     t_composition(:,1)= U235_7  U-235I  4.4482E-02        300.
                         U238_7  U-238I  2.7038E-03        300.
                         U234_7  U-234I  4.8869E-04        300.
     t_composition(:,2)= U235_7  U-235O  3.4610E-04        300.
                         U238_7  U-238O  4.7721E-02        300.
                         U234_7  U-234O  2.6438E-06        300.
/
&twodant
     1

niso=,
ngroup=,
mt=2,nzone=2,im=2,
igeom=3,isn=16,it=130,maxlcm=18000000,maxscm=35000000,
t
 xmesh=0.0 6.1156  24.1242 ,
 xints=30 100,
 zones=1 2;
t
lib=isotxs,balxs=0,
t
matls=
assign=
t
isct=3,ievt=1,ibl=1,ibr=0,iquad=-2,
ith=0,fluxp=0,xsectp=0,fissrp=0,sourcp=0,angp=0,geomp=0,
influx=0,norm=1.000e+00,epsi=0.0010,epso=.000001,
oitm=100,iitm=100,iitl=100,insors=0,raflux=0,rmflux=1
chi=,
t
rzmflx=1
t
/
EOF

$dir/mcc3.x
mv output                 $0.out
```

```
mv ISOTXS.merged          $0.ISOTXS
```

5.3 Hyperfine Group Calculation for Mixture

In a similar manner to the spectrum calculation with TWODANT, the hyperfine group calculation requires two sequential jobs. The first job should include l_hyperfine_transport in the &control block to produce the hyperfine group solutions,

l_hyperfine_transport = T,

and the second job needs an additional input card, c_externalspectrum to input the spectrum solution file,

c_externalspectrum = output.flux_hfg (or user-specified file name)

(the 1st job)

```
# ------------------------------------------------------------------------------
# the 1st job to generate hyperfine group flux solutions
# ------------------------------------------------------------------------------

lib={MC2-3 directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL33
     i_number_region         =1
     c_geometry_type         =mixture
     l_hyperfine_transport   =T
/
&material
     t_composition(:,1)= U235_7    U-235I   1.1530E-03    300.
                         U236_7    U-236I   5.6000E-06    300.
                         U238_7    U-238I   5.8010E-03    300.
                         U234_7    U-234I   1.1000E-05    300.
/
EOF

$lib/mcc3.x

mv output            $0.out
mv output.flux_hfg   $0.output.flux_hfg
```

(the 2nd job)

```
# ------------------------------------------------------------------------------
# the 2nd job to self-shield resonances using the hyperfine group flux solutions
# that were generated in the 1st job
# ------------------------------------------------------------------------------

lib={MC2-3 directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL33
     i_number_region         =1
     c_geometry_type         =mixture
```

```
     c_externalspectrum_hfg  ={$0.output.flux_hfg}
/
&material
     t_composition(:,1)= U235_7    U-235I   1.1530E-03    300.
                         U236_7    U-236I   5.6000E-06    300.
                         U238_7    U-238I   5.8010E-03    300.
                         U234_7    U-234I   1.1000E-05    300.
/
EOF

$lib/mcc3.x

mv output            $0.out
mv ISOTXS.merged     $0.ISOTXS
```

5.4 Ultrafine Group Calculation for One-dimensional Geometry

The following example is a one-dimensional, five-region slab geometry problem. The number of composition input data can be equal to or less than the number of regions. When the number of compositions are less than the number of regions, the composition-to-region assignment should be given using the i_composition card of the &geometry block.

```
lib={MC2-3 library directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL230
     i_number_region         =5
     i_external_chi          =3
     i_scattering_order      =5
     c_geometry_type         =slab
/
&geometry
     i_mesh        =   1   4   2   1   4
     r_location    = 0.5 2.5 3.5 4.0 6.0
     i_composition =   1   2   3   1   3       ! 3 compositions to 5 regions
     c_boundary_condition = reflective
/
&material
     t_composition(:,1)= U235_7     U235   3.50000E-03    300.
                         U238_7     U238   2.50000E-02    300.
     t_composition(:,2)= FE56 7     FE56   5.00000E-02    300.
                         C12__7    "C   "  2.50000E-04    300.  ! C___
     t_composition(:,3)= NA23_7     NA23   9.20200E-03    300.
                         O16__7     O-16   1.47400E-02    300.
/
EOF

$lib/mcc3.x

mv output           $0.out
mv ISOTXS.merged    $0.ISOTXS
```

5.5 Hyperfine Group Calculation for One-dimensional Geometry

The one-dimensional hyperfine group calculation requires two sequential jobs. The first job is basically the same as that for the one-dimensional ultrafine group calculation, except for the additional input l_hyperfine_transport = T. This job will produce *output.flux_hfg1d* which contains the hyperfine-group flux solutions.

In the second job, the l_hyperfine_transport card should be turned off and the new card c_externalspectrum_hfg needs to be added with the file name of the hyperfine group flux solutions that were generated from the first job. Then, prior to the one-dimensional transport calculation, the ultrafine group cross sections are self-shielded using the region-wise hyperfine group flux solutions instead of using the NR-based pointwise fluxes.

(the 1st job)

```
# ------------------------------------------------------------------------------
# the 1st job to generate one-dimensional hyperfine group flux solutions
# ------------------------------------------------------------------------------


lib={MC2-3 library directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL230
     i_number_region         =5
     i_external_chi          =3
     c_geometry_type         =slab
     l_hyperfine_transport   =T
/
&geometry
     i_mesh        =   1   4   2   1   4
     r_location    = 0.5 2.5 3.5 4.0 6.0
     i_composition =   1   2   3   1   3       ! 3 compositions to 5 regions
     c_boundary_condition = reflective
/
&material
     t_composition(:,1)= U235_7     U235   3.50000E-03    300.
                         U238_7     U238   2.50000E-02    300.
     t_composition(:,2)= FE56 7     FE56   5.00000E-02    300.
                         C12__7    "C   "  2.50000E-04    300.  ! C___
     t_composition(:,3)= NA23_7     NA23   9.20200E-03    300.
                         O16__7     O-16   1.47400E-02    300.
/
EOF

$xdir/mcc3.x
mv output              $0.out
mv output.flux_hfg1d   $0.output.flux_hfg1d
```

(the 2nd job)

```
# ------------------------------------------------------------------------------
# the 2nd job to self-shield resonances using the one-dimensional hyperfine group
# flux solutions that were generated in the 1st job
# ------------------------------------------------------------------------------
```

```
lib={MC2-3 library directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL230
     i_number_region         =5
     i_external_chi          =3
     c_geometry_type         =slab
     c_externalspectrum_hfg  ={$0.output.flux_hfg1d}
/
&geometry
     i_mesh        =   1   4   2   1   4
     r_location    = 0.5 2.5 3.5 4.0 6.0
     i_composition =   1   2   3   1   3       ! 3 compositions to 5 regions
     c_boundary_condition = reflective
/
&material
     t_composition(:,1)= U235_7     U235   3.50000E-03    300.
                         U238_7     U238   2.50000E-02    300.
     t_composition(:,2)= FE56 7     FE56   5.00000E-02    300.
                         C12__7    "C   "  2.50000E-04    300.
     t_composition(:,3)= NA23_7     NA23   9.20200E-03    300.
                         O16__7     O-16   1.47400E-02    300.
/
EOF

$xdir/mcc3.x
mv output           $0.out
mv ISOTXS.merged    $0.ISOTXS
```

5.6 Generation of Gamma Cross Sections

For generation of gamma data: PMATRX (gamma production) and GAMISO (gamma interaction), the l_gamma card in the &control block needs to be set to true. This generates PMATRX and GAMISO in a binary form, which can be converted to an ASCII form using the input cards “l_pmatrx_ascii” and “l_gamiso_ascii” in the &output block in the same job or using the input cards “c_pmatrx_conversion” and “c_isotxs_conversion” in a separate job. The output files “pmatrx.out” and “gamiso.out” are also generated for the user information. test11.1.out,

```
lib={MC2-3 library directory}


cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
     c_gammalibdir="$lib/lib.gamma"
/
&control
     c_group_structure       =ANL33
     i_number_region         =3
     l_gamma                 =T
/
&material
    t_composition(:, 1)= U235_7    U23501   3.31549E-05   300.00
                         U238_7    U23801   1.64625E-02   300.00
                         PU2397    P23901   2.72982E-03   300.00
                         O16__7    O-1601   4.12135E-02   300.00

    t_composition(:, 2)= CR52_7    CR5202   1.19417E-02   300.00
```

```
                          FE56_7    FE5602   4.67666E-02   300.00

    t_composition(:, 3)= NA23_7    NA2303   2.25353E-02   300.00
                          FE56_7    FE5603   2.44363E-03   300.00
/
EOF

$lib/mcc3.x

mv output               $0.out
mv PMATRIX.merged       $0.PMATRX
mv GAMISO.merged        $0.GAMISO
```

5.7 Conversion of ISOTXS

The ISOTXS conversion job requires two blocks: &control and &output.

For conversion of ISOTXS from binary to ASCII format, use c_isotoxs_conversion = bin2asc.

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = bin2asc
/
&output
   c_isotxs_file = ISOTXS.bin
/
EOF

$lib/mcc3.x

mv output             $0.out
mv ISOTXS.bin.ascii   $0.ISOTXS.ascii
```

For conversion of ISOTXS from ASCII to binary format, use c_isotoxs_conversion = asc2bin.

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = asc2bin
/
&output
   c_isotxs_file = ISOTXS.asc
/
EOF

$lib/mcc3.x

mv output              $0.out
mv ISOTXS.asc.binary   $0.ISOTXS
```

For merging multiple ISOTXS files, the user can use the same option as for the format conversion: c_isotxs_conversion = bin2asc or asc2bin which first merges files if more than a file are input at c_isotxs_file, and then creates two files: ISOTXS.merged and ISOTXS.merged.ascii.

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = bin2asc
/
&output
   c_isotxs_file = ISOTXS.file1
                   ISOTXS.file2
                   ISOTXS.file2
/
EOF

$lib/mcc3.x

mv output              $0.out
mv ISOTXS.merged       $0.ISOTXS
```

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = asc2bin
/
&output
   c_isotxs_file = ISOTXS.file1.ascii
                   ISOTXS.file2.ascii
                   ISOTXS.file2.ascii
/
EOF

$lib/mcc3.x

mv output               $0.out
mv ISOTXS.merged.ascii  $0.ISOTXS.ascii
```

5.8 Generation of DLAYXS

For generation of DLAYXS, the l_delayedneutron card in the &control block needs to be set to true. If c_dlayxs_file is not specified, the file name becomes DLAYXS by default. In the t_composition(:,:), the user-specified name, number density, and temperature do not affect the final results since DLAYXS takes the isotope names given in the library. Note that the number of compositions should be one and if more than one composition is defined only the first composition is used for generating DLAYXS.

```
lib={MC2-3 library directory}

cat > input << EOF
&library
     c_mcclibdir  ="$lib/lib.mcc"
     c_pwlibdir   ="$lib/lib.pw.200","."
/
&control
     c_group_structure       =ANL33
     i_number_region         =1
     l_delayedneutron        =T
/
&output
```

```
     l_dlayxs_ascii          =T
/
&material
!                                 any      any           any
t_composition(:, 1)=  TH2327   TH2321   1.00000E+00   300.
                      U232_7   U232_1   1.00000E+00   300.
                      U233_7   U233_1   1.00000E+00   300.
                      U234_7   U234_1   1.00000E+00   300.
                      U235_7   U235_1   1.00000E+00   300.
                      U236_7   U236_1   1.00000E+00   300.
                      U237_7   U237_1   1.00000E+00   300.
                      U238_7   U238_1   1.00000E+00   300.
                      NP2377   NP2371   1.00000E+00   300.
                      PU2387   PU2381   1.00000E+00   300.
                      PU2397   PU2391   1.00000E+00   300.
                      PU2407   PU2401   1.00000E+00   300.
                      PU2417   PU2411   1.00000E+00   300.
                      PU2427   PU2421   1.00000E+00   300.
                      AM2417   AM2411   1.00000E+00   300.
/
EOF

$lib/mcc3.x

mv output               $0.out
mv DLAYXS               $0.DLAYXS
mv DLAYXS.ascii         $0.DLAYXS.ascii  # optional
```

5.9 Conversion of DLAYXS

The DLAYXS conversion job requires two blocks: &control and &output.

For conversion of DLAYXS from binary to ASCII format, use c_isotoxs_conversion = bin2asc. The file name is given in the c_dlayxs_file card in the &output block. Conversion of multiple files like ISOTXS is not allowed in this option.

```
lib={MC2-3 library directory}


cat > input << EOF
&control
   c_dlayxs_conversion = bin2asc
/
&output
   c_dlayxs_file = dlayxs.bin
/
EOF

$lib/mcc3.x

mv output               $0.out
ls -l dlayxs.bin.ascii
```

For conversion of ISOTXS from ASCII to binary format, use c_isotoxs_conversion = asc2bin.

```
lib={MC2-3 library directory}


cat > input << EOF
&control
```

```
   c_dlayxs_conversion = asc2bin
/
&output
   c_dlayxs_file = dlayxs.asc
/
EOF

$lib/mcc3.x

mv output               $0.out
ls -l dlayxs.asc.binary
```

5.10 Conversion of PMATRX

The PMATRX conversion job requires two blocks: &control and &output.

For conversion of PMATRX from binary to ASCII format, use c_pmatrx_conversion = bin2asc, as shown in the following example. To convert the PMATRX file from ASCII to binary format, set c_pmatrx_conversion to asc2bin in the &control block.

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = bin2asc
/
&output
   c_isotxs_file = PMATRX.bin
/
EOF

$lib/mcc3.x

mv output                $0.out
mv PMATRX.bin.ascii      $0.PMATRX.ascii
```

5.11 Conversion of GAMISO

The format of GAMISO is in the same as ISOTXS and therefore its conversion from binary to ASCII or vice versa can be done in the same manner as for ISOTXS using the two input cards: c_isotoxs_conversion (&control) and c_isotxs_file (&output).

```
lib={MC2-3 library directory}

cat > input << EOF
&control
   c_isotxs_conversion = bin2asc
/
&output
   c_isotxs_file = GAMISO.bin
/
EOF

$lib/mcc3.x

mv output              $0.out
mv GAMISO.bin.ascii    $0.GAMISO.ascii
```

5.12 2D/3D MOC inputs

5.12.1 Hexagonal Cells in Triangular Lattice

```
&geometry
! Lattice with 1 cell type
! General condition
  T_HEXCELL_LATTICE.I_NUMREGION               = 3
  T_HEXCELL_LATTICE.I_REGION_COMPOSITION(1:3) = 1 2 3

! Axial configuration
  T_HEXCELL_LATTICE.I_NUMPLANE                = 5
  T_HEXCELL_LATTICE.R_HZ(1:5)                 = 5.0 5.0 5.0 5.0 5.0
  T_HEXCELL_LATTICE.C_TOP_CONDITION           = 'reflective'
  T_HEXCELL_LATTICE.C_BOTTOM_CONDITION        = 'reflective'

! Radial configuration
  T_HEXCELL_LATTICE.I_NUMRING                 = 2
  T_HEXCELL_LATTICE.I_NUMCELLTYPE             = 1
  T_HEXCELL_LATTICE.R_CELL_FLAT2FLAT          = 1.1
  T_HEXCELL_LATTICE.I_CELLTYPE_MAP(1:3, 1:3)  =
                   1   1   0
                   1   1   1
                   0   1   1

! (Cell 1) Define cell type 1
  T_HEXCELL_LATTICE.T_CELLTYPE(1).C_TYPE        = 'annular'
  T_HEXCELL_LATTICE.T_CELLTYPE(1).I_NUMRING     = 4
  T_HEXCELL_LATTICE.T_CELLTYPE(1).R_RADIUS(1:4) = 0.15 0.30 0.45 0.50
  T_HEXCELL_LATTICE.T_CELLTYPE(1).I_RING_REGION(1:3, 1:5) =  2 1 3
                                                             2 1 3
                                                             2 1 3
                                                             2 1 3
                                                             2 1 3
/
```

5.12.2 Hexagonal Assemblies in Triangular Lattices with 360 Degree

```
&geometry
! Core with 2 assembly types
! General condition
   T_HEXASSY_LATTICE.I_NUMREGION               = 4
   T_HEXASSY_LATTICE.I_REGION_COMPOSITION(1:4)  = 1 2 3 4

! Axial configuration
   T_HEXASSY_LATTICE.I_NUMPLANE                = 1
   T_HEXASSY_LATTICE.R_HZ(1)                   = 1

! Radial configuration
```

```
   T_HEXASSY_LATTICE.I_NUMRING                  = 2
   T_HEXASSY_LATTICE.I_NUMASSYTYPE              = 2
   T_HEXASSY_LATTICE.R_ASSEMBLY_OUTER_FLAT2FLAT = 14.6850
   T_HEXASSY_LATTICE.I_ASSYTYPE_MAP(1:3, 1:3)   =
                           1  1  0
                           1  2  1
                           0  1  1
   T_HEXASSY_LATTICE.C_GEOMETRY_TYPE            = 'lattice_360'
   T_HEXASSY_LATTICE.C_OUTER_BOUNDARY_CONDITION = 'reflective'

!  Assembly type 1
   T_HEXASSY_TYPE(1).C_NAME                     = 'FA'
   T_HEXASSY_TYPE(1).R_CELL_FLAT2FLAT           =  0.9134
   T_HEXASSY_TYPE(1).R_ASSEMBLY_INNER_FLAT2FLAT =  13.6790
   T_HEXASSY_TYPE(1).I_NUMRING                  = 9   ! 17x17
   T_HEXASSY_TYPE(1).L_GAP_STUCTURE             = T
   T_HEXASSY_TYPE(1).I_NUMCELLTYPE              = 1
   T_HEXASSY_TYPE(1).I_CELLTYPE_MAP(1:17, 1:17) =
                  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
                  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0
                  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0
                  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0
                  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
                  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
                  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0
                  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0
                  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
                  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
                  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
                  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1
                  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1
                  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1
                  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
                  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1
                  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1

! (Cell 1) Define cell type 1 for assembly 1
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).C_TYPE           = 'annular'
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).I_NUMRING        = 2
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).R_RADIUS(1:2)    = 0.3501  0.4057
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).I_RING_REGION(1:3, 1) =  1 2 3
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_NUNGAP             = 2
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_NUMGAP_DIV         = 8
   T_HEXASSY_TYPE(1).T_GAPTYPE.R_GAP_DIMENSION(1:2) = 0.3018 0.2012
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_GAP_REGION(1:2, 1) = 4 3

!  Assembly type 2
   T_HEXASSY_TYPE(2).C_NAME                     ='CA'
   T_HEXASSY_TYPE(2).R_CELL_FLAT2FLAT           =  1.2558
   T_HEXASSY_TYPE(2).R_ASSEMBLY_INNER_FLAT2FLAT =  12.2707
   T_HEXASSY_TYPE(2).I_NUMRING                  = 6   ! 11x11
   T_HEXASSY_TYPE(2).L_GAP_STUCTURE             = T
   T_HEXASSY_TYPE(2).I_NUMCELLTYPE              = 1
   T_HEXASSY_TYPE(2).I_CELLTYPE_MAP(1:11, 1:11) =
                        1   1   1   1   1   1   0   0   0   0   0
```

```
                          1   1   1   1   1   1   1   0   0   0   0
                          1   1   1   1   1   1   1   1   0   0   0
                          1   1   1   1   1   1   1   1   1   0   0
                          1   1   1   1   1   1   1   1   1   1   0
                          1   1   1   1   1   1   1   1   1   1   1
                          0   1   1   1   1   1   1   1   1   1   1
                          0   0   1   1   1   1   1   1   1   1   1
                          0   0   0   1   1   1   1   1   1   1   1
                          0   0   0   0   1   1   1   1   1   1   1
                          0   0   0   0   0   1   1   1   1   1   1

! (Cell 1) Define cell type 1 for assembly 1
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).C_TYPE           = 'annular'
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).I_NUMRING        = 2
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).R_RADIUS(1:2)    = 0.4881  0.5625
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).I_RING_REGION(1:3, 1) =  1 2 3
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_NUMGAP             = 4
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_NUMGAP_DIV         = 2
   T_HEXASSY_TYPE(2).T_GAPTYPE.R_GAP_DIMENSION(1:4) =
                                        0.3018 0.40235 0.3018 0.2012
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_GAP_REGION(1:4, 1) = 4 3 4 3
/
```

5.12.3 Hexagonal Assemblies in Triangular Lattices with 120 Degree

```
&geometry
! Core with 3 assembly types
! General condition
   T_HEXASSY_LATTICE.I_NUMREGION                = 1
   T_HEXASSY_LATTICE.I_REGION_COMPOSITION(1)    = 1

   T_HEXASSY_LATTICE.I_NUMPLANE                 = 1
   T_HEXASSY_LATTICE.R_HZ(1)                    = 1

   T_HEXASSY_LATTICE.I_NUMRING                  = 2
   T_HEXASSY_LATTICE.I_NUMASSYTYPE              = 3
   T_HEXASSY_LATTICE.R_ASSEMBLY_OUTER_FLAT2FLAT = 9.53760
   T_HEXASSY_LATTICE.I_ASSYTYPE_MAP(1:5, 1:5)   =
                       3   1   3   1   2
                       1   1   1   3   2
                       3   1   3   3   2
                       1   3   3   1   2
                       2   2   2   2   2
   T_HEXASSY_LATTICE.C_GEOMETRY_TYPE            = 'rotational_120'
   T_HEXASSY_LATTICE.C_OUTER_BOUNDARY_CONDITION = 'vacuum'

!  Assembly type 1
   T_HEXASSY_TYPE(1).C_NAME                     ='ASSEMBLY 1'
   T_HEXASSY_TYPE(1).R_CELL_FLAT2FLAT           = 2.0
   T_HEXASSY_TYPE(1).R_ASSEMBLY_INNER_FLAT2FLAT = 8.63760
   T_HEXASSY_TYPE(1).I_NUMRING                  = 3   ! 5x5
   T_HEXASSY_TYPE(1).L_GAP_STUCTURE             = T
   T_HEXASSY_TYPE(1).I_NUMCELLTYPE              = 1
```

```
   T_HEXASSY_TYPE(1).I_CELLTYPE_MAP(1:5, 1:5)   =
                1   1   1   0   0
                1   1   1   1   0
                1   1   1   1   1
                0   1   1   1   1
                0   0   1   1   1

! (Cell 1) Define cell type 1 for assembly 1
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).C_TYPE              = 'hexagon'
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).I_NUMHEX            = 3
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).R_CENTER2SIDE(1:3)  = 0.3 0.7 0.95
   T_HEXASSY_TYPE(1).T_CELLTYPE(1).I_HEX_REGION(1:4, 1) = 1 1 1 1
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_NUMGAP                = 3
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_NUMGAP_DIV            = 6
   T_HEXASSY_TYPE(1).T_GAPTYPE.R_GAP_DIMENSION(1:3)    = 0.15 0.15 0.15
   T_HEXASSY_TYPE(1).T_GAPTYPE.I_GAP_REGION(1:3, 1)    = 1 1 1

!  Assembly type 2
   T_HEXASSY_TYPE(2).C_NAME                    ='ASSEMBLY 2'
   T_HEXASSY_TYPE(2).R_CELL_FLAT2FLAT          = 2.0
   T_HEXASSY_TYPE(2).R_ASSEMBLY_INNER_FLAT2FLAT = 9.23760
   T_HEXASSY_TYPE(2).I_NUMRING                 = 3   ! 5x5
   T_HEXASSY_TYPE(2).L_GAP_STUCTURE            = T
   T_HEXASSY_TYPE(2).I_NUMCELLTYPE             = 1
   T_HEXASSY_TYPE(2).I_CELLTYPE_MAP(1:5, 1:5)   =
                1   1   1   0   0
                1   1   1   1   0
                1   1   1   1   1
                0   1   1   1   1
                0   0   1   1   1

! (Cell 1) Define cell type 1 for assembly 2
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).C_TYPE              = 'hexagon'
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).I_NUMHEX            = 3
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).R_CENTER2SIDE(1:3)  = 0.3 0.6 0.8
   T_HEXASSY_TYPE(2).T_CELLTYPE(1).I_HEX_REGION(1:4, 1) = 1 1 1 1
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_NUMGAP                = 3
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_NUMGAP_DIV            = 6
   T_HEXASSY_TYPE(2).T_GAPTYPE.R_GAP_DIMENSION(1:3)    = 0.05 0.05 0.05
   T_HEXASSY_TYPE(2).T_GAPTYPE.I_GAP_REGION(1:3, 1)    = 1 1 1

!  Assembly type 3
   T_HEXASSY_TYPE(3).C_NAME                    ='ASSEMBLY 3'
   T_HEXASSY_TYPE(3).R_CELL_FLAT2FLAT          = 1.4
   T_HEXASSY_TYPE(3).R_ASSEMBLY_INNER_FLAT2FLAT = 8.33760
   T_HEXASSY_TYPE(3).I_NUMRING                 = 4
   T_HEXASSY_TYPE(3).L_GAP_STUCTURE            = T
   T_HEXASSY_TYPE(3).I_NUMCELLTYPE             = 1
   T_HEXASSY_TYPE(3).I_CELLTYPE_MAP(1:7, 1:7)   =
                1   1   1   1   0   0   0
                1   1   1   1   1   0   0
                1   1   1   1   1   1   0
                1   1   1   1   1   1   1
                0   1   1   1   1   1   1
                0   0   1   1   1   1   1
```

```
                0   0   0   1   1   1   1

! (Cell 1) Define cell type 1 for assembly 3
   T_HEXASSY_TYPE(3).T_CELLTYPE(1).C_TYPE                = 'annular'
   T_HEXASSY_TYPE(3).T_CELLTYPE(1).I_NUMRING             = 3
   T_HEXASSY_TYPE(3).T_CELLTYPE(1).R_RADIUS(1:3)         = 0.3 0.45 0.65
   T_HEXASSY_TYPE(3).T_CELLTYPE(1).I_RING_REGION(1:4, 1) = 1 1 1 1
   T_HEXASSY_TYPE(3).T_GAPTYPE.I_NUMGAP                  = 3
   T_HEXASSY_TYPE(3).T_GAPTYPE.I_NUMGAP_DIV              = 6
   T_HEXASSY_TYPE(3).T_GAPTYPE.R_GAP_DIMENSION(1:3)      = 0.2 0.2 0.2
   T_HEXASSY_TYPE(3).T_GAPTYPE.I_GAP_REGION(1:3, 1)      = 1 1 1
/
```

6 SAMPLE OUTPUTS

6.1 Main Output File

The main output is saved in the *output* file which contains the following information:

```
Copy of the input
Brief isotope information in the library
Broad-group energy structure
Isotope information in terms of resonances
Isotopic scattering matrix pointers
Region-wise data preparation and transport calculation status
Iteration table for the transport calculation
Result summary (including eigenvalue, elapsed time, and allocated memory)
```

Warning and error messages will be stated with the subroutine name when the code is terminated abnormally. In the current version, those might not fully cover all possible combinations of warning and error situations.

6.2 Cross Section File

The ultrafine- and broad-group principal cross sections for all isotopes are saved in separate files, output.microxs_ufg and output.microxs_bg, respectively, when the l_edit_xs card of &output is turned on. Those files contain capture, scattering, total, transport, fission, nu (the number of neutrons per fission), and chi (fission spectrum) for each isotope, as shown below:

```
region          1
 U238_7
      energy(eV)     capture  scattering       total   transport     fission          nu         chi
    1  1.4191E+07  8.5740E-04  2.8114E+00  5.4364E+00  2.9177E+00  1.1542E+00  4.4671E+00  4.2192E-06
    2  1.4073E+07  8.7727E-04  2.7918E+00  5.4507E+00  2.9537E+00  1.1442E+00  4.4493E+00  4.6010E-06
    3  1.3956E+07  8.9450E-04  2.7762E+00  5.4666E+00  2.9878E+00  1.1304E+00  4.4316E+00  4.9988E-06
    4  1.3840E+07  9.1093E-04  2.7618E+00  5.4828E+00  3.0212E+00  1.1158E+00  4.4141E+00  5.4286E-06
    5  1.3725E+07  9.2722E-04  2.7479E+00  5.4992E+00  3.0543E+00  1.1013E+00  4.3968E+00  5.8976E-06
.
.
.
 2080  4.2447E-01  1.5830E+01  1.4133E+01  1.3556E+02  1.3552E+02  1.0559E+02  2.4367E+00  2.7472E-13
 2081  4.2095E-01  1.6198E+01  1.4141E+01  1.3747E+02  1.3743E+02  1.0713E+02  2.4367E+00  2.7018E-13
 2082  4.1746E-01  1.6533E+01  1.4149E+01  1.3919E+02  1.3914E+02  1.0850E+02  2.4367E+00  2.6571E-13
 FE56_7
      energy(eV)     capture  scattering       total   transport
    1  1.4191E+07  7.4582E-04  1.1962E+00  2.5312E+00  1.5911E+00
    2  1.4073E+07  7.3133E-04  1.2073E+00  2.5494E+00  1.5972E+00
    3  1.3956E+07  7.1270E-04  1.2164E+00  2.5656E+00  1.6029E+00
    4  1.3840E+07  6.9309E-04  1.2251E+00  2.5811E+00  1.6082E+00
.
.
.
```

6.3 Flux and Moment File

The ultrafine- and broad-group fluxes for all regions or mixtures are saved in separate files, output.flux_ufg and output.flux_bg, respectively, when the l_edit_flux card of &output is turned on. Those files contain energy, lethargy, average flux (L_0) and Legendre moments (L_n, $n>0$) for all regions or mixtures. The following example shows the ultrafine-group flux file for the case with two mixtures:

```
       energy(eV)    lethargy    avg flux
 moment=           0
    1  1.4191E+07  8.3333E-03  5.8843E-05  8.4621E-05
    2  1.4073E+07  8.3333E-03  7.2380E-05  1.0454E-04
    3  1.3956E+07  8.3333E-03  8.1826E-05  1.1876E-04
    4  1.3840E+07  8.3333E-03  9.1438E-05  1.3246E-04
    5  1.3725E+07  8.3333E-03  1.0149E-04  1.4630E-04
.
.
.
 2080  4.2447E-01  8.3333E-03  1.3352E-09  2.0988E-10
 2081  4.2095E-01  8.3333E-03  1.3060E-09  1.9538E-10
 2082  4.1746E-01  8.3333E-03  1.2773E-09  1.8163E-10
       energy(eV)    lethargy    avg flux
 moment=           0
    1  1.4191E+07  8.3333E-03  5.8843E-05  8.4621E-05
    2  1.4073E+07  8.3333E-03  7.2380E-05  1.0454E-04
    3  1.3956E+07  8.3333E-03  8.1826E-05  1.1876E-04
    4  1.3840E+07  8.3333E-03  9.1438E-05  1.3246E-04
    5  1.3725E+07  8.3333E-03  1.0149E-04  1.4630E-04
.
.
.
```

6.4 Fission Spectrum File

The fission spectrum information is edited in the cross section file described above. For the convenience in the TWODANT input, for example, it is saved in a separate file: output.chi_ufg and output.chi_bg. In fact, the TWODANT input is generated automatically when the l_twodant card of &control and the &twodant block are defined, so the fission spectrum file will rarely be used. The following example shows the broad-group (33 groups) fission spectrum file for the case with a mixture:

```
  1.84781E-03  2.85726E-02  1.18911E-01  2.20214E-01  2.32471E-01
  1.72442E-01  1.07434E-01  5.93330E-02  3.02931E-02  1.48958E-02
  7.17869E-03  3.41500E-03  1.59780E-03  7.43089E-04  3.46439E-04
  1.61928E-04  7.58333E-05  3.55745E-05  1.67115E-05  7.85975E-06
  3.70029E-06  1.74316E-06  8.21625E-07  3.87417E-07  1.82780E-07
  8.62479E-08  4.07057E-08  1.92020E-08  8.94699E-09  4.30825E-09
  1.21474E-09  8.68648E-12  2.31025E-13
 ;
```

6.5 Leakage File

The ultrafine-group leakage information is edited to the output.leakage_[][] where [][] is the two-digit region number. This leakage data, $D_g B^2 \phi_g$, generated from a mixture will be used as an external source of other mixtures using l_external_source (&control) and i_source_region(:) (&material). The file contains one set of leakage whose region is identified by the file name for user convenience. The following example is part of the leakage output in which the energy (eV) and leakage are listed in the 2nd and 3rd columns in turn:

```
    1  1.41907E+07  1.88358E-06
    2  1.40729E+07  2.32098E-06
    3  1.39561E+07  2.61973E-06
    4  1.38403E+07  2.90099E-06
    5  1.37254E+07  3.19132E-06
.
.
.
 2078  4.31608E-01  5.22191E-10
 2079  4.28026E-01  4.82024E-10
 2080  4.24474E-01  4.38711E-10
```

```
 2081  4.20951E-01  4.08238E-10
 2082  4.17458E-01  3.73451E-10
```

6.6 Pointwise Cross Section File

The pointwise cross section file contains the hyperfine-group cross section data for each isotope and given temperature condition. Thus, the filename indicates an isotope name and temperature. For example, pw_U238_7.0300 should be for U238_7 at 300K. The MC^2-3 code reads it in a binary form and is able to convert it to an ASCII format, which is shown below. The starting hyperfine group is determined based on the highest resolved resonance energy in the library.

```
(starting UFG) (starting HFG) (ending HFG)
(HFG capture cross sections, i=(starting HFG),(ending HFG))
(HFG fission cross sections, i=(starting HFG),(ending HFG))
(HFG scattering cross sections, i=(starting HFG),(ending HFG))
(HFG total cross sections, i=(starting HFG),(ending HFG))
```

6.7 RZMFLX File

The *rzmflx* file contains zone-averaged fluxes and moments in which the format uses the standard CCCC form as described below. The code reads it in a binary form (l_externalspectrum_ufg = rzmflx) and is able to convert it to an ASCII format (l_edit_rzmflx = T) to see the fluxes and moments. In case that *rzmflx* is generated from TWODANT, the moments are defined differently depending upon the geometry selected:

$N = L_0 + 1$ for standard plane and spherical geometry

$N = (L_0 + 1)^2 / 4$ for cylindrical geometry

$N = (L_0 + 1)^2$ for two-angle plane geometry

$N = (L_0 + 1)(L_0 + 2) / 2$ for RZ geometry

where L_0 is the Legendre order and N is the number of moments.

```
C*********************************************************************
CF            RZMFLX                                                 -
CE            REGULAR ZONE-AVERAGED MOMENTS FLUX                     -
C*********************************************************************
C---------------------------------------------------------------------
CR            FILE IDENTIFICATION                                    -
C                                                                    -
CL    HNAME,(HUSE(I),I=1,2),IVERS                                    -
C                                                                    -
CW    1+3*MULT                                                       -
C                                                                    -
CD    HNAME          HOLLERITH FILE NAME  - RZMFLX -    (A6)         -
CD    HUSE           HOLLERITH USER IDENTIFICATION      (A6)         -
CD    IVERS          FILE VERSION NUMBER                             -
CD    MULT           DOUBLE PRECISION PARAMETER                      -
CD                        1- A8 WORD IS SINGLE WORD                  -
```

```
CD                        2- A8 WORD IS DOUBLE PRECISION WORD           -
C-------------------------------------------------------------------------
CR           SPECIFICATIONS                                               -
C                                                                         -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,NORD,EFFK,POWER                       -
C                                                                         -
CW    8=NUMBER OF WORDS                                                   -
C                                                                         -
CD    NDIM        NUMBER OF DIMENSIONS                                    -
CD    NGROUP      NUMBER OF GROUPS                                        -
CD    NINTI       NUMBER OF FIRST DIMENSION INTERVALS                     -
CD    NINTJ       NUMBER OF SECOND DIMENSION INTERVALS                    -
CD                NINTJ.EQ.1 IF NDIM.EQ.1                                 -
CD    NINTK       NUMBER OF THIRD DIMENSION INTERVALS                     -
CD                NINTK.EQ.1 IF NDIM.LE.2                                 -
CD    NORD        NUMBER OF LEGENDRE MOMENTS                              -
CD    EFFK        EFFECTIVE MULTIPLICATION FACTOR                         -
CD    POWER       POWER IN WATTS TO WHICH FLUX IS NORMALIZED               -
C-------------------------------------------------------------------------
CR          REGULAR MOMENTS FLUXES ON MULTIDIMENSIONAL INTERVALS          -
C                                                                         -
CL    ((FLUX(M,I),M=1,NORD,I=1,NINTI)----NOTE STRUCTURE BELOW----         -
C                                                                         -
CW    NORD*NINTI=NUMBER OF WORDS                                          -
C-------------------------------------------------------------------------
```

7 PROGRAM STRUCTURE

The MC^2-3 code is largely composed of the eight calculation parts: input processing, library processing, resonance self-shielding, scattering matrix generation, spectrum calculation, spatial homogenization and group condensation, ISOTXS generation, output processing. The resonances to be self-shielded are unresolved and resolved resonances and resonance-like cross sections above the resonance energy range. For resonance self-shielding, resolved resonances are reconstructed using resonance parameters or optionally provided through PENDF files. The transport calculation routines solve the broad, ultrafine, or hyperfine group transport equation for a homogeneous mixture or one-dimensional (slab or cylindrical) geometry. In addition to the standard output information, the code produces ultrafine and broad group flux moments, ultrafine and broad group fission spectra, ultrafine group leakage spectrum, and broad group cross sections according to the user-specified edit options.

Programmed in Fortran 95 and developed on PC Windows using the Compaq Visual Fortran environment, MC^2-3 should work on any Linux and UNIX systems, with trivial changes if needed, as well as Windows as long as the system memory permits.

7.1 Subroutine

The code includes the following 17 modules, whose names begin with "m_" so that they are easily identified as modules (see Appendix F for the subroutine tree of the code) :

Subroutine		Description
m_allocation	:	Allocate and deallocate arrays (including 20 subroutines)
m_database	:	Define default data such as built-in group structures, fission spectrum of U-238, elastic scattering coefficients, etc.
m_debug	:	Debugging routines
m_dimension	:	Common arrays
m_errorwarn	:	Error and warning messages (including 4 subroutines)
m_globaldata	:	Common variables other than arrays
m_inout	:	Input and output file units
m_isotxs	:	ISOTXS relevant routines (including 6 subroutines)
m_kind	:	Data kinds such as integer, real, double precision
m_localdata	:	Local data which are frequently defined in the subroutines
m_mcclib	:	Read the MC^2 libraries (including 8 subroutines)
m_memorymap	:	Keep track of memory allocation (including 1 subroutine)
m_outputs	:	Write outputs such as standard outputs, flux moments, microscopic and macroscopic cross sections
m_parameter	:	Define parameters

m_resolvedres : Routines for reconstructing the resolved resonances such as single-level Breit-Wigner, multi-level Breit-Wigner, Adler-Adler, and multi-pole resonances (including 7 subroutines)

m_time : Define data and functions for timing

m_types : Define types for the structured data

m_utility : Tools for copying variables and arrays and operating matrices (including 8 subroutines)

The code contains the following 96 subroutines listed in an alphabetical order:

avg_fn : Calculate the group average values of Legendre coefficients for the anisotropic scattering in the center-of-mass system using the Legendre coefficients at the group boundaries and the interpolation law

Called by cal_elasticscat

avg_sig : Calculate an average cross section value for the ultrafine group interval using the interpolation law

Called by chi_chimatrix, chi_nuchi

cal_0dtransport : Solve the ultrafine group transport equation for homogeneous mixture

Called by mcc3

cal_0dtransport_hfg : Solve the hyperfine group transport equation for homogeneous mixture

Called by mcc3

cal_1dtransport : Solve the ultrafine group transport equation for one-dimensional geometry

Called by mcc3

cal_1dtransport_bg : Solve the broad group transport equation for one-dimensional geometry

Called by mcc3

cal_1dtransport_hfg : Solve the hyperfine group transport equation for one-dimensional geometry

Called by mcc3

cal_chimatrix : Construct fission spectrum matrix for an isotope

Called by cal_chimatrix

cal_elasticscat : Generate ultrafine group elastic scattering matrices for P_N, $N \leq 1$

Called by mcc3

cal_elasticscat_hfg : Generate hyperfine group elastic scattering matrices for P_N, $N \leq 1$

Called by cal_0dtransport_hfg, cal_1dtransport_hfg

cal_elasticscatpl : Generate ultrafine group elastic scattering matrices for P_N, N > 1

Called by mcc3

cal_nonelasticscat : Generate inelastic and (n,2n) scattering matrices

Called by mcc3

cal_nuchi : Generate the average total number of neutrons per fission (ν) and the fission spectrum (χ) for an isotope

Called by mcc3

cal_resolvedres : Calculate self-shielded resolved resonances by numerical integration of the pointwise cross sections reconstructed using resonance parameters based on the narrow resonance approximation

Called by mcc3

cal_unresolvedres : Calculate self-shield unresolved resonances using the analytic integration method

Called by mcc3

check_balance : Check the neutron balance

Called by homogenize_region

check_keff : Calculate the k-eff value

Called by homogenize_region, postprocess

check_reactionrate : Calculate reaction rates for given coarse groups

Called by mcc3

chi_allocatematrix : Generate the chi matrix for a fissionable isotope

Called by mcc3

chi_update : Update the chi vector for an isotope to be equivalent to the chi matrix and solution spectrum

Called by cal_0dtransport

condense_group : Condense the ultrafine group cross sections to the broad group ones based on the flux moment solutions from the transport calculation

Called by homogenize_region, mcc3

condense_group1d : Condense the ultrafine group cross sections to the broad group ones for each region of the one-dimensional problem in which more than one compositions are defined

Called by mcc3

convert_pw : Convert a data type of pointwise cross sections from binary to ASCII or vice versa

Called by mcc3

cpm_cylinder : Solve the collision probability method equation for the cylindrical geometry (ultrafine group)

Called by cal_1dtransport

cpm_cylinder1g : Solve the collision probability method equation for a group for the cylindrical geometry

Called by cal_1dtransport_hfg

cpm_cylinder_bg : Solve the collision probability method equation for the cylindrical geometry (broad group)

Called by cal_1dtransport_bg

cpm_slab : Solve the collision probability method equation for the slab geometry (ultrafine group)

Called by cal_1dtransport

cpm_slab1g : Solve the collision probability method equation for a group for the slab geometry

Called by cal_1dtransport_hfg

cpm_slab_bg : Solve the collision probability method equation for the slab geometry (broad group)

Called by cal_1dtransport_bg

cpmpij_cylinder : Calculate the region-to-region transmission probability (P_{ij}) for the cylindrical geometry

Called by escapexs

cpmpij_slab : Calculate the region-to-region transmission probability (P_{ij}) for the slab geometry

Called by escapexs, cpm_slab, cpm_slab1g, cpm_slab_bg

energygrid : Calculate upper energy boundaries for ultrafine and broad groups

Called by mcc3

escapexs : Calculate the effective background cross sections taking into account the heterogeneity effect based on equivalence theory

Called by mcc3

f_deone : (function) Exponential integral

Called by cal_nuchi

f_derf : (function) Error function, Erf(x)

called by cal_nuchi

f_exp2 : (function) Exponential function, Exp2(x)

Called by f_ez3, f_yz33

f_ez3 : (function) Calculate the sum of Exp3(x) for k, $0 \le k \le \infty$, used for optically thin unit cells

Called by f_yz33

f_gamin : (function) Incomplete gamma function

Called by cal_nuchi

f_get_chi : (function) Obtain chi for a given incident neutron energy

Called by cal_0dtransport, cal_update, cpm_slab, cpm_slab_bg

f_pe3 : (function) Exp3 integral

Called by ez3

f_pfunc : (function) Obtain the collision escape probability for slab: (0.5 – Exp3(x)) / x

Called by yz33

f_yz33 : (function) Obtain the difference in the sum of Exp3(x) and Exp3(x+del) for k, $0 \le k \le \infty$

Called by cpmpij_slab

fill_region : Fill in the region array

Called by preprocess

gen_datafile : Generate the output files requested by the user, such as leakage and chi

Called by mcc3

gen_isotxs_macro : Generate a ISOTXS file with macroscopic cross sections

Called by mcc3

gen_isotxs_micro : Generate a ISOTXS file with microscopic cross sections

Called by mcc3

gen_tdinput : Generate the TWODANT input based on the inputs in the &twodant block

Called by cal_2dtransport

get_fn : Calculate $(2N+1)f_N$ for all Legendre coefficients at the hyperfine group boundaries

Called by cal_elasticscat, cal_elasticscatpl, cal_elasticscat_hfg

get_lumpedisotope : Produce the lumped isotope with the macroscopic cross sections

Called by isotoxs_merge

get_macroxs : Generated the ultrafine group macroscopic cross sections required for the transport calculation

Called by condense_group, mcc3

get_macroxs_bg : Generated the broad group macroscopic cross sections required for the transport calculation

Called by homogenize_region

get_scatteringindex : Assign the scattering indices: 100s for elastic scattering, 200s for inelastic scattering, and 300s for (n,2n) reaction

Called by isotxs_merge

get_totalscat : Calculate the total scattering cross section by summing up all partial cross sections

Called by condense_group

get_twonorm : Calculate the two-norm

Called by cal_0dtransport

homogenize_region : Homogenize all the regions in the one-dimensional geometry

Called by mcc3

integrate_pwxs : Integrate the pointwise cross sections numerically for the given energy range

Called by cal_resovedres, selfshield_aboveres

interpolate_table : Calculate the probability of inelastic or (n,2n) scattering from a group into all possible sink groups by interpolation of the tabulated probabilities given from ENDF/B

Called by cal_nonelasticscat

inversematrix : Invert the square matrix

Called by unres_integral

isotxs_convert : Convert a data type of ISOTXS from binary to ASCII or vice versa

Called by mcc3

isotxs_header_asc : Read or write isotope-independent data in an ASCII form

Called by gen_isotxs_macro, gen_isotxs_micro, isotxs_convert, isotxs_merge

isotxs_isotope_asc : Read or write isotope-dependent in an ASCII form

Called by gen_isotxs_macro, gen_isotxs_micro, isotxs_convert, isotxs_merge

isotxs_read_header : Read isotope-independent data in a binary form

Called by isotxs_convert, isotxs_merge

isotxs_read_isotope : Read isotope-dependent data in a binary form

Called by isotxs_convert, isotxs_merge

isotxs_write_header : Write isotope-independent data in a binary form

Called by gen_isotxs_macro, gen_isotxs_macro_ufg, gen_isotxs_micro, isotxs_convert, isotxs_merge

isotxs_write_isotope : Write isotope-dependent data in a binary form

Called by gen_isotxs_macro, gen_isotxs_macro_ufg, gen_isotxs_micro, isotxs_convert, isotxs_merge

jintegral : Perform analytic integration of a resonance

Called by unres_integral

legendrepoly : Calculate Lengendre polynomial using a recursive formula

Called by cal_elasticscatpl

manual : Quick input manual

Called by read_input

mcc3 : The main program

moc_slab : Solve the method-of-characteristic transport equation for homogeneous mixture and one-dimensional geometry

Called by cal_1dtransport, cal_1dtransport_hfg

nonelasticscat : Calculate non-elastic scattering cross sections for discrete level, evaporation, and continuum formats

Called by cal_nonelasticscat

openfile : Create or open files with a unit number available

Called by cal_0dtransport, cal_0dtransport_hfg, cal_elasticscat, cal_nonelasticscat, cal_nuchi, cal_resolvedres, chi_allocatematrix, condense_group, convert_pw, genn_datafile, gen_isotxs_macro, gen_isotxs_macro_ufg, gen_isotxs_micro, gen_tdinput, isotxs_convert, isotxs_merge, output_2var(m_output), mcc3, read_input, read_pendf, read_pwfile, search_pendf, write_pwfile

packer : Determine which ultrafine group an unresolved resonance energy point falls, compute the phase shifts for $\ell = 0, 1, 2, 3$, calculate the center-of-correction factor for use with the effective background cross section, compute the penetration factors for $\ell = 1, 2, 3$

Called by unres_integral

postprocess : Check k-effective, etc.

Called by mcc3

preprocess : Get problem-independent data

Called by mcc3

prob_elasticscat : Calculate group-to-group probabilities for elastic scattering

Called by cal_0dtransport_hfg, cal_1dtransport_hfg

prob_inelasticscat : Calculate group-to-group probabilities for inelastic scattering

Called by nonelasticscat

read_inelasticscat : Read data from the "MCCF10" file

Called by cal_nonelasticscat, condense_group, condense_group1d

read_input : Read inputs from the "input" file

Called by mcc3

read_mcclib : Read data from the eight MC^2 files

Called by mcc3

read_pendf : Read data from PENDF and interpolate values at the ultrafine group points

Called by cal_resolvedres, search_pendf, selfshield_aboveres

read_pwfile : Read data from a pointwise file

Called by cal_resolvedres, reconstruct_basexs

read_rzmflx : Read data from the "rzmflx" file

Called by mcc3, read_spectrum

read_spectrum : Read data from the file defined by c_externalspectrum_ufg or c_externalspectrum_hfg

Called by cal_resolvedres, condense_group

reconstruct_basexs : Construct the smooth cross sections based on the ultrafine group cross section data from File 5 of the MC^2 library or the pointwise cross section if available

Called by escapexs, mcc3

reconstruct_pwxs : Reconstruct a pointwise file using the interpolation law for the smooth cross section and the resonance parameters from the MC^2 library

Called by cal_resolvedres

scat2dto1d : Convert a two-dimensional scattering array to an one-dimensional array for the debugging purpose

scat1dto2d : Convert an one-dimensional scattering array to a two-dimensional array for the debugging purpose

search_pendf : Search for PENDF files relevant to the temperature given by the user

Called by cal_resolvedres

search_pwfile : Search the pointwise cross section for a given temperature using the interpolation law

Called by cal_resolvedres, reconstruct_basexs, search_pwfile

select_mcclib : Select the MC^2 libraries only for the isotopes of the problem

Called by read_mcclib

selfshield_aboveres : Self-shield cross sections above the resonance energy range. This routine requires reconstructing the pointwise cross sections over the energy range of concern using PENDF files.

Called by cal_resolvedres

solve_wxy : Calculate the real and imaginary parts of $w(x, y)$

Called by jintegral, slbw (m_resolvedres), mlbw (m_resolvedres), adleradler (m_resolvedres), multipole (m_resolvedres)

unres_allocate : Allocate unresolved resonance array

Called by cal_unresolvedres

unres_deallocate : Deallocate unresolved resonance array

Called by cal_unresolvedres

unres_integral : Calculate the single-level Breit-Wigner unresolved resonance integral including interference scattering and self-overlap

Called by cal_unresolvedres

unres_interpolate : Interpolate E* points for a given energy

Called by cal_unresolvedres

wfunction : Calculate the real part of $w(0, x)$ using a rational approximation for $Exp(x^2) * Erfc(x)$

Called by jintegral, unres_integral

write_pwfile : Write the pointwise file on a file named pw_{isotope name}.{temperature in K}

Called by cal_resolvedres

7.2 Main Data Structure

Many data in the code are stored in the structure form so that the data management is efficient. Major arrays in the structure form are listed as follows. Note that (C), (R), (I), (L), and (T) denote character, real, integer, logical, and type, respectively.

```
Major variables and arrays

nisotope        :(I) number of isotopes
nhfg            :(I) number of hyperfine group
nufg            :(I) number of ultrafine group
nbg             :(I) number of broad group
nregion         :(I) number of regions for one-dimensional problem
                     number of compositions for mixture problems
```

```
nregion1        :(I) nregion+1 for one-dimensional problem in which the
                     nregion1-th mixture includes homogeneous composition
                     nregion for mixture problems
norder          :(I) number of scattering order
energy_hfg(:)   :(R) energies(eV) for hyperfine groups (2082 groups),
                     currently constant lethargy interval
energy_ufg(:)   :(R) energies(eV) for ultrafine groups,
                     currently constant lethargy interval
energy_bg(:)    :(R) energies(eV) for broad groups
exp_du          :(R) Exp(-Δu) where Δu is the unit lethargy of ultrafine
                     group
exp_dh          :(R) Exp(-Δh) where Δh is the unit lethargy of hyperfine
                     group
fission_source :(L) index for fission source in the current composition
```

Composition

```
Composition(:)%isotopename_lib       :(C) Isotope name in the library
               isotopename_user      :(C) Isotope name defined by the user
               pwfilename            :(C) name of pointwise file
               masstype              :(C) mass type (i.e., heavy, light,
                                           and hydrogen)
               numden                :(R) number density (#/barn-cm)
               temperature           :(R) temperature in K
               energy_unresolvedres  :(R) highest upper energy (eV) of
                                           unresolved resonance
               energy_resolvedres    :(R) highest upper energy (eV) of
                                           resolved resonance
               alpha                 :(R) [(A-1)/(A+1)]^2
               amass                 :(R) atomic mass (amu)
               nlib_smooth           :(I) address of the smooth cross
                                           section in the library file 5
               nlib_unresolvedres    :(I) address of the unresolved
                                           resonance in the library file 3
               nlib_resolvedres      :(I) address of the resolved resonance
                                           in the library file 4
               iufg_unresolvedres    :(I) beginning ufg for unresolved
                                           resonance
               iufg_resolvedres      :(I) beginning ufg for resolved
                                           resonance
               ndownscat_ufg         :(I) number of down-scattering groups
                                           in ufg
               ndownscat_hfg         :(I) number of down-scattering groups
                                           in hfg
               fissionable           :(L) true if fissionable isotope
```

Ultrafine group cross sections

```
    ufgroup(:)%cap(:)                :(R) capture cross section
               fis(:)                :(R) fission cross section
               nufis(:)              :(R) nu-fission cross section
               chi(:)                :(R) fission spectrum
               scat(:)               :(R) total scattering cross section
```

```
 sigetar(:)             :(R) effective escape cross section
 sig_np(:)              :(R) (n,p) cross section
 sig_nd(:)              :(R) (n,d) cross section
 sig_nh3(:)             :(R) (n,h3) cross section
 sig_nhe3(:)            :(R) (n,he3) cross section
 sig_nalpha(:)          :(R) (n,alpha) cross section
 scatn2n(:)             :(R) (n,2n) cross section
 scatn3n(:)             :(R) (n,3n) cross section
 scatggn2n(:)           :(R) (n,2n) cross section matrix
%moment(:)%tot(:)       :(R) P_N total cross section
           trn(:)       :(R) P_N transport cross section
           scatgg(:)    :(R) P_N total scattering cross section
                             matrix
           scatggels(:):(R) P_N elastic scattering cross
                             section matrix
           scatggine(:):(R) P_N inelastic scattering cross
                             section matrix
 ip_scatgg(:)           :(I) address of the highest group
                             scattering into group g of total
                             scattering cross section
 ip_scatggels(:)        :(I) address of the highest group
                             scattering into group g of
                             elastic scattering cross section
 ip_scatggine(:)        :(I) address of the highest group
                             scattering into group g of
                             inelastic scattering cross
                             section
 ip_scatggn2n(:)        :(I) address of the highest group
                             scattering into group g of (n,2n)
                             cross section
 ip_scatggn3n(:)        :(I) address of the highest group
                             scattering into group g of (n,3n)
                             cross section
 igg_scatgg(:)          :(I) address of the self-group
                             scattering of total scattering
                             cross section
 igg_scatggels(:)       :(I) address of the self-group
                             scattering of elastic scattering
                             cross section
 igg_scatggine(:)       :(I) address of the self-group
                             scattering of inelastic
                             scattering cross section
 igg_scatggn2n(:)       :(I) address of the self-group
                             scattering of (n,2n) cross
                             section
 igg_scatggn3n(:)       :(I) address of the self-group
                             scattering of (n,3n) cross
                             section
```

The scattering matrix elements are stored in the one-dimensional array in a scattering source production based order: i.e., 1→1, 1→2, 2→2, 2→3, 3→3, etc. For example, the beginning and ending addresses of group 3 are found at ip_scatggels(3)=address(2→3) and igg_scatggels(3)=address(3→3), respectively. Therefore, the scattering band of group 3 should be igg_scatggels(3) - ip_scatggels(3) + 1 when up-scattering is not accounted for or ip_scatggels(4) – ip_scatggels(3) in general even when up-scattering is considered. The following scheme can be used for calculating scattering sources for each group, given an isotope m and a scattering moment order L:

```
Do ig = 1, nufg
   Band = ip_scatgg(ig+1)- ip_scatgg(ig)
   Gbeg = ig - (igg_scatgg(ig) - ip_scatgg(ig))
   Gend = ig + (ip_scatgg(ig+1) - igg_scatgg(ig) - 1)
   Ip = ip_scatgg(ig) - 1
   Do jg = Gbeg, Gend
      Ip = Ip + 1
      Scat_source(ig) = Scat_source(ig) + ufgroup(m)%moment(L)%scatgg(ip)
                                        * composition(m)%numden
                                        * Solution_ufg%phi(jg,L)
   Enddo
Enddo
```

Broad group cross sections

```
    bgroup(:) - the same structure as ufgroup(:)
```

Hyperfine group cross sections

```
    hfgroup(:)%cap(:)              :(R) capture cross section
              fis(:)               :(R) fission cross section
              scat(:)              :(R) scattering cross section
              tot(:)               :(R) total cross section
```

Tranport solution

```
solution_hfg%phi(:,:)              :(R) flux and moment solutions for
                                        hyperfine groups
solution_ufg%phi(:,:)              :(R) flux and moment solutions for
                                        ultrafine groups
solution_bg%phi(:,:)               :(R) flux and moment solutions for
                                        broad groups
```

Region

```
     region(:)%composition(:)      :(T) composition-dependent data
              hfgroup(:)           :(T) hyperfine group data
              ufgroup(:)           :(T) ultrafine group data
              bgroup(:)            :(T) broad group data
              macro_hfg(:)         :(T) hyperfine group data
                                        (macroscopic)
              macro_ufg(:)         :(T) ultrafine group data
                                        (macroscopic)
              macro_bg(:)          :(T) broad group data (macroscopic)
              solution_hfg(:)      :(T) hyperfine group solution
              solution_ufg(:)      :(T) ultrafine group solution
              solution_bg(:)       :(T) broad group solution
              xmesh                :(R) location from zero
              thick                :(R) thickness or interval
              volfrac              :(R) volume fraction
              pot                  :(R) potential cross section
              keff                 :(R) k∞
              nisotope             :(I) number of isotopes
```

```
                nmesh                 :(I) number of meshes
                sourceregion          :(I) external source region
                reg2comp              :(I) composition number for the region
                fissionsource         :(L) True if fission source exists
                exist_hydrogen        :(L) True if hydrogen exists
```

```
Fission spectrum
  chimatrix(:)%ngroup_in              :(I) number of incident neutron group
               ngroup_out             :(I) number of outgoing neutron group
               group_in(:)            :(R) energies (eV) of incident neutron
                                           groups
               group_out(:)           :(R) energies (eV) of outgoing neutron
                                           groups
               chi(:,:)               :(R) ultrafine group fission spectrum
                                           matrix
```

References

1. H. Henryson II, B. J. Toppel, and C. G. Stenberg, "MC^2-2: A Code to Calculate Fast Neutron Spectra and Multi-group Cross Sections," ANL-8144, Argonne National Laboratory (1976).
2. P. Oblozinsky and M. Herman, "Special Issues on Evaluated Nuclear Data File ENDF/B-VII.0," *Nuclear Data Sheets*, **107**, 2931 (2006).
3. B. J. Toppel, H. Henryson II, and C. G. Stenberg, "ETOE-2/MC^2-2/SDX Multi-group Cross-Section Processing," Conf-780334-5, Presentation at RSICC Seminar Workshop on Multi-group Cross Sections, Oak Ridge, TN, March (1978).
4. C. H. Lee and W. S. Yang, "Development of Multigroup Cross Section Generation Code MC^2-3 for Fast Reactor Analysis," International Conference on Fast Rectors and Related Fuel Cycles, Kyoto, Japan, Dec. 7-11 (2009).
5. R. E. MacFarlane, "NJOY 99/2001: New Capabilities in Data Processing," Presentation at the Workshop on Reactor Physics and Analysis Capabilities for Generation IV Nuclear Energy Systems, Argonne National Laboratory, Argonne, Illinois, February 18-19 (2003).
6. R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," LA-10049-M, Los Alamos National Laboratory (1990).
7. Cross Section Evaluation Working Group, "ENDF-6 Formats Manual: Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII," National Nuclear Data Center, Brookhaven National Laboratory, BNL-NCS-44945-05-Rev, June (2005).
8. R. N. Hwang, "A Rigorous Pole Representation of Multilevel Cross Sections and Its Practical Applications," *Nucl. Sci. Eng.,* **96,** 192-209 (1987).
9. R. N. Hwang, "Resonance Theory for Reactor Applications," *Nuclear Computational Science: A Century in Review*, Y. Azmy & E. Sartori (eds.) Springer, Dordrecht (2010).
10. C. W. Reich and M. S. Moore, *Phys. Rev.,* **111**, 929 (1958).
11. G. de Saussure and R. B. Perez, "POLLA: A FORTRAN Program to Convert R-Matrix-Type Multilevel Resonance Parameters for Fissile Nuclei into Equivalent Kapur-Peierls-Type Parameters," ORNL-TM-2599 (1969).
12. R. E. MacFarlane, "TRANSX 2: A Code for Interfacing MATXS Cross Section Libraries to Nuclear Transport Codes," LA-12312-MS (1992).
13. R. N. Hwang, "Efficient Methods for the Treatment of Resonance Integrals," *Nucl. Sci. Eng*., **52**, 157 (1973).
14. R. N. Blomquist, "VIM Continuous Energy Monte Carlo Transport Code," Proc. Intl. Conf. on Mathematics, Computations, Reactor Physics and Environmental Analysis, Portland, OR, April 30-May 4 (1995).
15. J. R. Liaw, Private Communication, Argonne National Laboratory. September (2002).
16. R. N. Hwang, Private Communication, Argonne National Laboratory. August (2002).

17. H. Henryson II, "Multigroup Elastic Scattering for Cross Sections for Heavy Elements," *Nucl. Sci. Eng*., **43**, 235 (1971).

18. M. Segev, "Group Transfer Matrices and Their Relation to Basic Cross Section Data," *Nucl. Sci. Eng*., **45**, 269 (1971).

19. A. P. Olson, "RABID: An Integral Transport Theory Code for Neutron Slowing Down in Slab Cells," ANL-7654 (1970).

20. P. H. Kier and A. A. Robba, "RABBLE: A Program for Computation of Resonance Absorption in Multigroup Reactor Cells," ANL-7326 (1967).

21. I. Gargantini and T. Pomentale, "Rational Chebyshev Approximation to the Bessel Function Integrals K_i (x)," communication to the ACM, 7, 727 (1964).

22. R. N. Hwang, Private Communication, Argonne National Laboratory. August (1989) and October (1991).

23. Tone, "A Numerical Study of Heterogeniety Effects in Fast Reactor Critical Assemblies," *J. Nucl. Sci. Technol*., **12**, 467 (1975).

24. D. M. O'Shea and H. C. Thacher, "Computation of Resonance Line Shape Functions," *Tran. Am. Nucl.*, **6**, 36 (1963).

25. M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, U.S. Dept. of Commerce, National Bureau of Standards, Applied Mathematics Series 65, 328 (1964).

26. H. G. Joo et al., "Method and Performance of a Three-Dimensional Whole-Core Transport Code DeCART," Proceeding of PHYSOR2004, Chicago, IL, Apr. 25-29 (2004) (CD-ROM).

27. A. Marin-Lafleche, M. A. Smith, and C. H. Lee, "PROTEUS-MOC: A 3D Deterministic Solver Incorporating 2D Method of Characteristics," *Proc. M&C 2013*, Sun Valley, ID, May 5-9 (2013).

28. A. Kochetkov et al., "Experimental program Performed at the BFS-31 Assemblies – K-infinity Experiments for 238U in Fast Neutron Spectra: Measurements with Plutonium Mixed with Depleted Uranium Dioxide," OECD/NEA, NEA/NSC/DOC(2006)1 (2006).

29. Y. S. Jung and W. S. Yang, "A Consistent CMFD Formulation for the Acceleration of Neutron Transport Calculations Based on the Finite Element Method," *Nucl. Sci. Eng.,* **185**, 307-324 (2018).

Appendix A. Calculation of Doppler-Broadened Line Shape Functions

The symmetric and anti-symmetric line shape functions are used in the computation of Doppler-broadened resonance cross sections. These functions are defined in terms of the real and imaginary parts of the error function for complex arguments as

$$\psi(a,b) = \frac{a\sqrt{\pi}}{2}\,\mathrm{Re}\left[W\left(\frac{ab}{2},\frac{a}{2}\right)\right], \tag{A.1a}$$

$$\chi(a,b) = a\sqrt{\pi}\,\mathrm{Im}\left[W\left(\frac{ab}{2},\frac{a}{2}\right)\right], \tag{A.1b}$$

where $W(z) = W(x,y) = \exp(-z^2)\,\mathrm{erfc}(-iz)$ and $z = x + iy$.

$\mathrm{Re}[W(x,y)]$ and $\mathrm{Im}[W(x,y)]$ are pre-calculated using the methods described in Reference [24] and stored in coarse and fine mesh tables as indicated below. In the fine mesh tables, y ranges between -0.02 and 0.5 with a mesh interval of 0.02, while in the coarse mesh tables y ranges between 0.4 and 3.0 with a mesh interval of 0.1. In both tables, x ranges between -0.1 and 3.9 with a mesh interval of 0.1.

If $|x| \le 3.9$ and $y \le 3.0$, the $\mathrm{Re}[W(z)]$ and $\mathrm{Im}[W(z)]$ are obtained using the six point bivariate interpolation formula of Reference [25] in either the fine or coarse mesh tables as appropriate to the value of y.

If $|x| > 3.9$ or $y > 3.0$ but $|x| \le 6.0$ and $y \le 6.0$, $W(z)$ is approximated by

$$W(z) = iz\sum_{i=1}^{3}\frac{a_i}{z^2 - b_i}, \tag{A.2}$$

where $a_1 = 0.4613135$, $a_2 = 0.09999216$, $a_3 = 0.002883894$, $b_1 = 0.1901635$, $b_2 = 1.7844927$, $b_3 = 5.5253437$.

Setting $z = x + iy$, Eq. (A.2) is transformed to

$$\mathrm{Re}[W(x,y)] = \sum_{i=1}^{3}\frac{a_i\left[-y(x^2 - y^2 - b_i) + 2x^2y\right]}{(x^2 - y^2 - b_i)^2 + 4x^2y^2}, \tag{A.3a}$$

$$\mathrm{Im}[W(x,y)] = \sum_{i=1}^{3}\frac{a_i\left[x(x^2 - y^2 - b_i) + 2xy^2\right]}{(x^2 - y^2 - b_i)^2 + 4x^2y^2}. \tag{A.3b}$$

If $|x| > 6.0$ or $y > 6.0$ but $|x| \le 100.0$ and $y \le 100.0$, $W(z)$ is approximated by

$$W(z) = iz\sum_{i=1}^{2}\frac{c_i}{z^2 - d_i}, \tag{A.4}$$

where $c_1 = 0.5124242$, $c_2 = 0.05176536$, $d_1 = 0.2752551$, $d_2 = 2.724745$.

Thus,

$$\mathrm{Re}\left[W(x,y)\right]=\sum_{i=1}^{2}\frac{c_i\left[-y(x^2-y^2-d_i)+2x^2y\right]}{(x^2-y^2-d_i)^2+4x^2y^2}, \tag{A.5a}$$

$$\mathrm{Im}\left[W(x,y)\right]=\sum_{i=1}^{2}\frac{c_i\left[x(x^2-y^2-d_i)+2xy^2\right]}{(x^2-y^2-d_i)^2+4x^2y^2}. \tag{A.5b}$$

If $|x|$ > 100.0 or y > 100.0, the code uses an asymptotic approximation. The integral representation of $W(z)$ is given as

$$W(Z)=\frac{i}{\pi}\int_{-\infty}^{\infty}\frac{e^{-t^2}}{z-t}dt\approx\frac{i}{\pi}\sum_{j=1}^{N}\frac{w_j}{z-t_j}, \tag{A.6}$$

where w_j and t_j are the weights and abscissae for the Hermite quadrature. In particular, for very large x or y, we set $N=2$ and ignore t_j relative to z so that

$$W(Z)\approx\frac{2i}{\pi}\frac{w}{z}, \tag{A.7}$$

where $w=\sqrt{\pi}/2$. Setting $z=x+iy$ again, the final equation becomes

$$\mathrm{Re}\left[W(x,y)\right]=\frac{y}{\sqrt{\pi}(x^2+y^2)}, \tag{A.8a}$$

$$\mathrm{Im}\left[W(x,y)\right]=\frac{x}{\sqrt{\pi}(x^2+y^2)}. \tag{A.8b}$$

For the special case of $x=0$, $W(z)$ becomes

$$W(0,y)=e^{y^2}\mathrm{erfc}(y). \tag{A.9}$$

If y < 2.0, Eq. (A.9) is evaluated using the rational approximation of Reference [25].

Appendix B. Major Data of ENDF/B-VII.0

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
H - 1	H1___7			0		0	20.48	0	2.22
H - 2	H2___7			0		0	3.39	0	6.26
H - 3	H3___7			0		0	1.66	0	0
He- 3	HE3__7			0		0	3.31	0	20.58
He- 4	HE4__7			0		0	0.76	0	0
Li- 6	LI6__7			0		0	0.67	0	7.25
Li- 7	LI7__7			0		0	0.97	0	2.03
Be- 7	BE7__7			0		0	19.06	0	0
Be- 9	BE9__7			0		0	6.15	0	6.81
B - 10	B10__7			0		0	2.14	0	11.46
B - 11	B11__7			0		0	4.84	0	3.37
C - 0	C_____7			0		0	4.74	0	4.95
N - 14	N14__7			0		0	9.87	0	10.83
N - 15	N15__7			0		0	4.42	0	2.49
O - 16	O16__7			0		0	3.89	0	4.14
O - 17	O17__7			0		0	4.2	0	8.05
F - 19	F19__7			0		0	3.61	0	6.6
Na- 22	NA22_7	MLBW	1.50E+04	1	1.00E+05	10	4.08	0	12.42
Na- 23	NA23_7	MLBW	5.00E+05	23		0	3.68	0	6.96
Mg- 24	MG24_7	MLBW	5.20E+05	28		0	3.66	0	7.33
Mg- 25	MG25_7	MLBW	2.20E+05	10		0	2.01	0	11.09
Mg- 26	MG26_7	MLBW	4.50E+05	6		0	2.32	0	6.44
Al- 27	AL27_7	RM	8.45E+05	79		0	2.35	0	7.73
Si- 28	SI28_7	RM	1.75E+06	58		0	2.15	0	8.47
Si- 29	SI29_7	RM	1.30E+06	29		0	2.43	0	10.61
Si- 30	SI30_7	RM	1.50E+06	29		0	2.22	0	6.59
P - 31	P31__7			0		0	4.1	0	7.93
S - 32	S32__7	MLBW	1.57E+06	83		0	1.93	0	8.64
S - 33	S33__7	MLBW	2.60E+05	9		0	1.86	0	11.42
S - 34	S34__7	MLBW	4.80E+05	9		0	1.63	0	6.99
S - 36	S36__7			0		0	2.16	0	4.31
Cl- 35	CL35_7			0		0	2.92	0	8.58
Cl- 37	CL37_7			0		0	3	0	6.11
Ar- 36	AR36_7	MLBW	4.65E+04	2	1.20E+06	17	0.85	0	8.79
Ar- 38	AR38_7	MLBW	3.00E+05	2	1.20E+06	9	0.85	0	6.6
Ar- 40	AR40_7	RM	1.50E+06	209		0	1.54	0	6.1
K - 39	K39__7	MLBW	2.00E+05	33		0	0.41	0	7.8
K - 40	K40__7			0		0	2.32	0	10.1
K - 41	K41__7	MLBW	1.25E+05	25		0	0.5	0	7.53
Ca- 40	CA40_7	MLBW	5.00E+05	133		0	1.63	0	8.36
Ca- 42	CA42_7	MLBW	3.00E+05	54		0	1.63	0	7.93
Ca- 43	CA43_7	MLBW	4.00E+04	26		0	1.63	0	11.13
Ca- 44	CA44_7	MLBW	5.00E+05	39		0	1.63	0	7.41
Ca- 46	CA46_7			0		0	1.15	0	7.28
Ca- 48	CA48_7	MLBW	5.00E+05	6		0	1.63	0	5.15
Sc- 45	SC45_7	MLBW	9.69E+04	191		0	2.71	0	8.76

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Ti- 46	TI46_7	MLBW	3.00E+05	97		0	2.54	0	8.88
Ti- 47	TI47_7	MLBW	1.00E+05	48		0	2.54	0	11.63
Ti- 48	TI48_7	MLBW	3.00E+05	62		0	2.22	0	8.14
Ti- 49	TI49_7	MLBW	1.80E+05	48		0	2.54	0	10.94
Ti- 50	TI50_7	MLBW	3.00E+05	32		0	2.54	0	6.37
V - 0	V____7	MLBW	1.00E+05	102		0	4.8	0	7.31
Cr- 50	CR50_7	RM	7.92E+05	386		0	3.63	0	9.26
Cr- 52	CR52_7	RM	9.80E+05	223		0	3.73	0	7.94
Cr- 53	CR53_7	RM	2.00E+05	131		0	3.77	0	9.72
Cr- 54	CR54_7	RM	9.00E+05	116		0	3.82	0	6.25
Mn- 55	MN55_7	MLBW	1.00E+05	149		0	3.33	0	7.27
Fe- 54	FE54_7	RM	7.00E+05	380		0	3.77	0	9.3
Fe- 56	FE56_7	RM	8.50E+05	311		0	3.72	0	7.65
Fe- 57	FE57_7	RM	2.00E+05	75		0	4.37	0	10.04
Fe- 58	FE58_7	RM	4.00E+05	68		0	6.51	0	6.58
Co- 58	CO58_7	MLBW	5.96E+02	1		0	5.31	0	10.45
Co- 58m	CO58M7	MLBW	5.00E+02	1		0	5.31	0	10.45
Co- 59	CO59_7	RM	1.00E+05	165		0	5.59	0	7.49
Ni- 58	NI58_7	RM	8.12E+05	482		0	5.15	0	9
Ni- 59	NI59_7	MLBW	1.00E+04	9		0	5.31	0	11.24
Ni- 60	NI60_7	RM	4.50E+05	272		0	4.52	0	7.82
Ni- 61	NI61_7	RM	7.00E+04	32		0	5.31	0	10.6
Ni- 62	NI62_7	RM	6.00E+05	57		0	4.83	0	6.84
Ni- 64	NI64_7	RM	6.00E+05	52		0	7.16	0	6.1
Cu- 63	CU63_7	RM	9.95E+04	254		0	5.64	0	7.92
Cu- 65	CU65_7	RM	9.95E+04	178		0	5.64	0	7.07
Zn- 0	ZN___7	MLBW	1.00E+05	827		0	3.94	0	10.2
Ga- 69	GA69_7	MLBW	4.50E+03	27		0	5.64	0	7.65
Ga- 71	GA71_7	MLBW	5.60E+03	30		0	5.64	0	6.52
Ge- 70	GE70_7	MLBW	1.40E+04	21	1.05E+06	26	6.25	0	7.42
Ge- 72	GE72_7	MLBW	1.20E+04	17	7.01E+05	15	5.31	0	6.78
Ge- 73	GE73_7	MLBW	8.63E+03	49	1.37E+04	13	5.31	0	10.2
Ge- 74	GE74_7	MLBW	6.00E+03	11	6.04E+05	18	5.31	0	6.51
Ge- 76	GE76_7	MLBW	3.00E+04	10	5.70E+05	9	5.31	0	6.07
As- 74	AS74_7	MLBW	5.50E+01	3	1.74E+05	25	5.64	0	10.24
As- 75	AS75_7	MLBW	9.70E+03	250	1.00E+05	11	5.64	0	7.33
Se- 74	SE74_7	MLBW	2.60E+03	9	1.00E+05	17	7.07	0	8.03
Se- 76	SE76_7	MLBW	9.00E+03	22	1.00E+05	11	7.07	0	7.42
Se- 77	SE77_7	MLBW	2.70E+03	38	1.00E+05	17	7.07	0	10.5
Se- 78	SE78_7	MLBW	1.20E+04	21	1.00E+05	10	8.66	0	6.96
Se- 79	SE79_7			0	1.00E+05	27	7.17	0	9.91
Se- 80	SE80_7	MLBW	1.00E+04	16	1.00E+05	10	7.07	0	6.7
Se- 82	SE82_7	MLBW	3.00E+04	7	1.00E+05	7	5.64	0	5.9
Br- 79	BR79_7	MLBW	5.50E+03	342	1.00E+05	15	5.49	0	7.89
Br- 81	BR81_7	MLBW	1.60E+04	209	1.00E+05	11	5.98	0	7.59
Kr- 78	KR78_7	MLBW	8.00E+02	4	1.00E+05	20	6.16	0	8.36
Kr- 80	KR80_7	MLBW	1.00E+03	11	1.00E+05	19	5.49	0	7.88
Kr- 82	KR82_7	MLBW	4.00E+01	2	1.00E+05	24	6.88	0	7.47

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Kr- 83	KR83_7	MLBW	2.72E+02	3	1.00E+05	19	6.21	0	10.52
Kr- 84	KR84_7	MLBW	9.00E+04	68		0	5.98	0	7.11
Kr- 85	KR85_7	MLBW	1.80E+03	8	3.08E+05	39	6.16	0	9.86
Kr- 86	KR86_7	MLBW	9.50E+05	204		0	6.33	0	5.52
Rb- 85	RB85_7	MLBW	8.47E+03	291	1.00E+05	11	6.51	0	8.65
Rb- 86	RB86_7	MLBW	3.00E+03	65	5.62E+05	22	5.47	0	9.92
Rb- 87	RB87_7	MLBW	1.25E+04	30	1.00E+05	10	6.33	0	6.08
Sr- 84	SR84_7	MLBW	3.40E+03	11	8.03E+05	34	6.26	0	8.53
Sr- 86	SR86_7	MLBW	3.00E+04	32	1.00E+05	7	4.87	0	8.43
Sr- 87	SR87_7	MLBW	1.41E+04	116	1.00E+05	10	6.35	0	11.11
Sr- 88	SR88_7	MLBW	3.00E+05	88		0	6.33	0	6.37
Sr- 89	SR89_7			0	1.00E+05	19	6.24	0	7.8
Sr- 90	SR90_7			0	1.00E+05	13	5.8	0	5.8
Y - 89	Y89__7	MLBW	4.10E+05	401		0	5.71	0	6.86
Y - 90	Y90__7	MLBW	1.00E+04	34	2.05E+05	12	6.16	0	7.93
Y - 91	Y91__7			0	1.00E+05	24	5.72	0	6.54
Zr- 90	ZR90_7	MLBW	6.00E+04	89	4.00E+05	9	6.51	0	7.19
Zr- 91	ZR91_7	MLBW	2.00E+04	95	1.00E+05	8	6.51	0	8.64
Zr- 92	ZR92_7	MLBW	7.10E+04	101	1.00E+05	3	6.51	0	6.73
Zr- 93	ZR93_7	MLBW	6.80E+03	51	1.00E+05	12	6.51	0	8.22
Zr- 94	ZR94_7	MLBW	9.00E+04	73	1.00E+05	2	6.51	0	6.47
Zr- 95	ZR95_7			0	1.00E+05	26	5.7	0	7.85
Zr- 96	ZR96_7	MLBW	1.00E+05	30		0	5.81	0	5.58
Nb- 93	NB93_7	SLBW	7.35E+03	194		0	5.98	0	7.21
Nb- 94	NB94_7	MLBW	2.80E+01	3	1.00E+05	27	6.16	0	8.49
Nb- 95	NB95_7			0	1.00E+05	27	5.64	0	6.89
Mo- 92	MO92_7	MLBW	5.00E+04	77	1.00E+05	5	6.16	0	8.07
Mo- 94	MO94_7	MLBW	2.00E+04	55	1.00E+05	9	5.98	0	7.37
Mo- 95	MO95_7	MLBW	2.14E+03	56	2.06E+05	19	6.16	0	9.15
Mo- 96	MO96_7	MLBW	1.90E+04	75	1.00E+05	9	6.16	0	6.82
Mo- 97	MO97_7	MLBW	2.00E+03	66	1.00E+05	17	5.98	0	8.64
Mo- 98	MO98_7	MLBW	3.20E+04	158	1.00E+05	7	5.98	0	5.93
Mo- 99	MO99_7			0	1.00E+05	27	5.64	0	8.29
Mo-100	MO1007	MLBW	2.60E+04	124	1.00E+05	5	5.98	0	5.4
Tc- 99	TC99_7	MLBW	6.37E+03	538	1.42E+05	36	6.51	0	6.76
Ru- 96	RU96_7			0	1.00E+05	25	4.85	0	8.07
Ru- 98	RU98_7			0	1.00E+05	26	4.81	0	7.47
Ru- 99	RU99_7	MLBW	1.00E+03	40	1.00E+05	19	4.68	0	9.67
Ru-100	RU1007	MLBW	1.20E+04	88	1.00E+05	10	4.99	0	6.8
Ru-101	RU1017	MLBW	1.04E+03	49	1.28E+05	17	5.35	0	9.22
Ru-102	RU1027	MLBW	1.34E+04	145	1.00E+05	8	4.99	0	6.23
Ru-103	RU1037	MLBW	3.50E+02	8	1.00E+05	14	5.42	0	8.9
Ru-104	RU1047	MLBW	1.11E+04	114	1.00E+05	10	4.68	0	5.91
Ru-105	RU1057			0		0	5.48	0	8.47
Ru-106	RU1067			0	1.00E+05	22	3.34	0	5.45
Rh-103	RH1037	MLBW	4.12E+03	280	4.01E+04	13	5.41	0	7
Rh-105	RH1057	SLBW	7.50E+00	2	1.00E+05	28	4.83	0	6.59
Pd-102	PD1027	MLBW	8.20E+02	4		0	5.64	0	7.62

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Pd-104	PD1047	MLBW	7.50E+03	122		0	5.45	0	7.09
Pd-105	PD1057	MLBW	2.05E+03	200	2.83E+05	23	5.47	0	9.56
Pd-106	PD1067	MLBW	6.00E+03	86		0	5.31	0	6.54
Pd-107	PD1077	MLBW	1.00E+03	138	1.00E+05	19	5.31	0	9.22
Pd-108	PD1087	MLBW	5.50E+03	77		0	5.98	0	6.15
Pd-110	PD1107	MLBW	6.80E+03	80		0	5.64	0	5.76
Ag-107	AG1077	MLBW	6.50E+03	400	1.00E+05	13	5.47	0	7.27
Ag-109	AG1097	MLBW	5.00E+03	308	8.88E+04	19	5.47	0	6.81
Ag-110m	AG10M7	MLBW	1.25E+02	11	1.00E+05	26	5.47	0	8.84
Ag-111	AG1117	MLBW	9.99E+02	83	6.04E+04	17	5.47	0	6.48
Cd-106	CD1067	MLBW	6.00E+03	56	1.00E+05	13	4.52	0	7.93
Cd-108	CD1087	MLBW	6.00E+03	61	1.00E+05	9	4.21	0	7.33
Cd-110	CD1107	MLBW	7.00E+03	102	1.00E+05	8	4.21	0	6.98
Cd-111	CD1117	MLBW	2.30E+03	156	1.00E+05	16	4.37	0	9.4
Cd-112	CD1127	MLBW	1.15E+04	118	1.00E+05	7	4.32	0	6.54
Cd-113	CD1137	MLBW	1.10E+04	400	1.00E+05	11	4.99	0	9.04
Cd-114	CD1147	MLBW	8.00E+03	85	1.00E+05	8	5	0	6.15
Cd-115m	CD15M7	MLBW	1.00E+03	25	4.09E+04	14	4.83	0	8.88
Cd-116	CD1167	MLBW	9.50E+03	49	1.00E+05	8	4.68	0	5.76
In-113	IN1137	MLBW	6.00E+02	76	1.00E+05	19	4.71	0	7.28
In-115	IN1157	MLBW	2.00E+03	253	1.00E+05	17	5.36	0	6.78
Sn-112	SN1127	MLBW	1.50E+03	15	1.00E+05	18	4.83	0	7.75
Sn-113	SN1137	MLBW	5.10E+02	29	7.81E+04	28	4.99	0	10.3
Sn-114	SN1147	MLBW	2.00E+03	13	1.00E+05	17	4.99	0	7.55
Sn-115	SN1157	MLBW	9.00E+02	5	1.00E+05	20	4.99	0	9.56
Sn-116	SN1167	MLBW	3.00E+04	213	1.00E+05	7	4.83	0	6.94
Sn-117	SN1177	MLBW	2.30E+03	77	1.00E+05	18	4.83	0	9.33
Sn-118	SN1187	MLBW	2.00E+03	13	1.00E+05	15	4.52	0	6.48
Sn-119	SN1197	MLBW	1.26E+03	23	1.00E+05	19	4.96	0	9.11
Sn-120	SN1207	MLBW	5.00E+04	260	1.00E+05	4	4.83	0	6.17
Sn-122	SN1227	MLBW	3.00E+05	355		0	4.54	0	5.95
Sn-123	SN1237			0	1.00E+05	27	3.87	0	8.49
Sn-124	SN1247	MLBW	3.15E+05	191		0	4.52	0	5.73
Sn-125	SN1257	MLBW	9.90E+02	9	2.77E+04	26	4.52	0	8.19
Sn-126	SN1267	MLBW		0	1.00E+05	17	3.92	0	5.65
Sb-121	SB1217	MLBW	5.40E+03	213	1.00E+05	15	4.99	0	6.81
Sb-123	SB1237	MLBW	5.40E+03	205	1.00E+05	14	4.37	0	6.47
Sb-124	SB1247			0	1.00E+05	39	4.08	0	8.71
Sb-125	SB1257			0	1.00E+05	27	4.39	0	6.22
Sb-126	SB1267	MLBW	9.99E+02	100	1.78E+04	19	4.37	0	8.38
Te-120	TE1207			0	1.00E+05	27	3.63	0	7.18
Te-122	TE1227	MLBW	2.00E+04	260	1.00E+05	8	3.8	0	6.93
Te-123	TE1237	MLBW	2.00E+03	43	1.00E+05	17	4.37	0	9.43
Te-124	TE1247	MLBW	1.50E+04	187	1.00E+05	8	4.23	0	6.57
Te-125	TE1257	MLBW	7.75E+03	294	1.00E+05	12	3.94	0	9.12
Te-126	TE1267	MLBW	1.36E+04	58	1.00E+05	8	4.08	0	6.29
Te-127m	TE27M7			0	1.00E+05	28	3.6	0	8.87
Te-128	TE1287	MLBW	2.25E+04	39	1.00E+05	9	3.8	0	6.09

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Te-129m	TE29M7			0	1.00E+05	28	3.67	0	8.52
Te-130	TE1307	MLBW	3.10E+04	23	1.00E+05	7	3.66	0	5.93
Te-132	TE1327	MLBW	2.78E+04	27	9.81E+05	20	3.8	0	5.82
I -127	I127_7	MLBW	4.05E+03	340		0	4.52	0	6.83
I -129	I129_7	MLBW	3.40E+03	125	1.00E+05	12	6.18	0	6.46
I -130	I130_7	MLBW	5.60E+02	64	4.03E+04	15	3.4	0	8.58
I -131	I131_7			0	1.00E+05	27	3.66	0	6.33
I -135	I135_7			0		0	4.81	0	3.8
Xe-123	XE1237			0		0	6.01	0	10.47
Xe-124	XE1247	MLBW	2.60E+02	4	1.00E+05	25	6.04	0	7.6
Xe-126	XE1267	MLBW	2.33E+03	5	1.00E+05	17	4.23	0	7.23
Xe-128	XE1287	MLBW	3.50E+03	15	1.00E+05	16	6.15	0	6.91
Xe-129	XE1297	MLBW	4.10E+03	70	1.00E+05	15	3.14	0	9.26
Xe-130	XE1307	MLBW	4.00E+03	19		0	6.21	0	6.61
Xe-131	XE1317	MLBW	3.95E+03	48	8.08E+04	16	3.66	0	8.94
Xe-132	XE1327	MLBW	4.00E+03	7	1.00E+05	15	3.66	0	6.44
Xe-133	XE1337			0	1.00E+05	27	3.57	0	8.53
Xe-134	XE1347	MLBW	1.00E+04	6	1.00E+05	10	4.23	0	6.45
Xe-135	XE1357	SLBW	1.90E+02	1	1.00E+05	15	3.54	0	7.99
Xe-136	XE1367	MLBW	4.90E+05	37		0	6.38	0	4.02
Cs-133	CS1337	MLBW	3.99E+03	189	8.16E+04	15	3.53	0	6.89
Cs-134	CS1347	MLBW	1.80E+02	8	1.00E+05	26	3.53	0	8.83
Cs-135	CS1357	MLBW	2.20E+02	7	1.00E+05	25	6.35	0	6.76
Cs-136	CS1367			0	1.00E+05	27	3.4	0	8.27
Cs-137	CS1377			0	1.00E+05	18	3.27	0	4.28
Ba-130	BA1307	MLBW	2.80E+03	41	1.00E+05	17	3.53	0	7.49
Ba-132	BA1327	MLBW	1.30E+02	3	1.00E+05	26	3.54	0	7.19
Ba-133	BA1337	MLBW	3.06E+02	6	1.24E+04	9	3.53	0	9.47
Ba-134	BA1347	MLBW	1.10E+04	87	1.00E+05	11	3.53	0	6.97
Ba-135	BA1357	MLBW	4.70E+03	111	1.00E+05	17	3.53	0	9.11
Ba-136	BA1367	MLBW	3.50E+04	104	1.00E+05	8	3.53	0	6.9
Ba-137	BA1377	MLBW	1.25E+04	78	1.00E+05	9	3.53	0	8.61
Ba-138	BA1387	MLBW	1.40E+05	34		0	3.53	0	4.72
Ba-140	BA1407	MLBW	2.30E+04	8	1.00E+05	2	4.88	0	4.83
La-138	LA1387	MLBW	3.30E+02	11	1.00E+05	24	3.53	0	8.78
La-139	LA1397	MLBW	2.00E+04	150	1.00E+05	9	3.02	0	5.16
La-140	LA1407	MLBW	9.99E+02	15	3.02E+04	14	3.14	0	6.69
Ce-136	CE1367	MLBW	8.80E+02	10	5.56E+05	37	3.41	0	7.48
Ce-138	CE1387	MLBW	9.99E+02	9	7.19E+05	48	3.02	0	7.46
Ce-139	CE1397	MLBW	3.01E+02	11	2.57E+05	22	3.02	0	9.2
Ce-140	CE1407	MLBW	2.00E+05	198		0	3.94	0	5.43
Ce-141	CE1417	MLBW	3.50E+02	7	1.00E+05	24	3.8	0	7.17
Ce-142	CE1427	MLBW	1.30E+04	50	1.00E+05	9	4.08	0	5.14
Ce-143	CE1437	MLBW	4.60E+02	12	1.90E+04	22	8.45	0	6.9
Ce-144	CE1447			0	1.00E+05	27	2.99	0	4.76
Pr-141	PR1417	MLBW	1.01E+04	190	1.46E+05	48	3.02	0	5.84
Pr-142	PR1427	MLBW	9.99E+02	24	3.69E+03	7	3.94	0	7.35
Pr-143	PR1437	MLBW	3.75E+02	6	1.00E+05	24	3.14	0	5.76

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Nd-142	ND1427	MLBW	2.25E+04	72	2.00E+05	17	3.94	0	6.12
Nd-143	ND1437	MLBW	5.50E+03	150	2.25E+05	61	3.94	0	7.82
Nd-144	ND1447	MLBW	9.98E+03	57	2.50E+05	14	5.15	0	5.76
Nd-145	ND1457	MLBW	3.98E+03	195	6.77E+04	13	4.52	0	7.57
Nd-146	ND1467	MLBW	7.60E+03	57	4.57E+05	20	6.51	0	5.29
Nd-147	ND1477	MLBW	1.80E+02	12	5.03E+04	8	8.66	0	7.33
Nd-148	ND1487	MLBW	9.99E+03	115	3.00E+05	15	6.16	0	5.04
Nd-150	ND1507	MLBW	1.39E+04	78	1.31E+05	19	6.16	0	5.33
Pm-147	PM1477	MLBW	1.02E+02	44	1.00E+05	26	8.45	0	5.9
Pm-148	PM1487			0	1.00E+05	28	2.03	0	7.26
Pm-148m	PM48M7	SLBW	1.00E+00	1		0	5.12	0	7.41
Pm-149	PM1497			0	1.00E+05	28	1.98	0	5.56
Pm-151	PM1517	MLBW	1.56E+02	38	8.57E+04	19	6.7	0	5.94
Sm-144	SM1447	MLBW	1.19E+04	75	5.36E+05	52	3.64	0	6.76
Sm-147	SM1477	MLBW	1.99E+03	212	1.22E+05	46	8.66	0	8.14
Sm-148	SM1487	MLBW	9.14E+02	11	4.27E+05	53	8.66	0	5.87
Sm-149	SM1497	MLBW	5.20E+02	159	2.27E+04	28	8.66	0	7.99
Sm-150	SM1507	MLBW	1.57E+03	23	3.36E+05	51	8.04	0	5.6
Sm-151	SM1517	MLBW	2.96E+02	121	6.63E+04	32	8.66	0	8.26
Sm-152	SM1527	MLBW	5.15E+03	92	1.23E+05	48	8.66	0	5.87
Sm-153	SM1537	MLBW	2.50E+01	12	6.50E+03	13	7.65	0	7.97
Sm-154	SM1547	MLBW	5.20E+03	36	8.25E+04	45	8.04	0	5.81
Eu-151	EU1517	MLBW	1.00E+02	92	1.00E+05	31	7.84	0	6.31
Eu-152	EU1527	MLBW	6.20E+01	84	1.00E+05	22	9.73	0	8.55
Eu-153	EU1537	MLBW	9.78E+01	72	8.39E+04	16	8.45	0	6.44
Eu-154	EU1547	MLBW	2.75E+01	20	1.00E+05	27	8.04	0	8.15
Eu-155	EU1557	MLBW	2.97E+01	8	1.00E+05	27	9.36	0	6.34
Eu-156	EU1567			0	1.00E+05	28	6.29	0	7.45
Eu-157	EU1577	MLBW	6.72E+01	22	7.72E+04	27	7.07	0	5.82
Gd-152	GD1527	RM	2.66E+03	130	3.46E+05	47	8.04	0	6.25
Gd-153	GD1537	RM	1.32E+02	10	4.10E+04	48	9.08	0	8.89
Gd-154	GD1547	RM	2.76E+03	164	1.23E+05	14	6.88	0	6.43
Gd-155	GD1557	RM	1.83E+02	92	6.04E+04	32	7.84	0	8.54
Gd-156	GD1567	RM	2.23E+03	88	9.00E+04	17	7.84	0	6.36
Gd-157	GD1577	RM	3.07E+02	60	5.49E+04	12	7.84	0	7.94
Gd-158	GD1587	RM	9.98E+03	96	8.00E+04	37	7.65	0	5.94
Gd-160	GD1607	RM	9.66E+03	58	7.50E+04	33	7.02	0	5.63
Tb-159	TB1597	MLBW	1.25E+03	224	1.00E+05	17	7.33	0	6.38
Tb-160	TB1607	MLBW	9.00E+00	4	6.40E+04	30	6.7	0	7.7
Dy-156	DY1567	MLBW	9.10E+01	19	1.39E+05	60	7.07	0	6.97
Dy-158	DY1587	MLBW	8.62E+01	4	9.96E+04	35	6.88	0	6.83
Dy-160	DY1607	MLBW	2.01E+03	66	8.73E+04	15	6.99	0	6.45
Dy-161	DY1617	MLBW	9.96E+02	254	2.58E+04	18	7.01	0	8.2
Dy-162	DY1627	MLBW	4.85E+03	75	8.12E+04	19	4.37	0	6.27
Dy-163	DY1637	MLBW	9.97E+02	115	7.39E+04	15	7.06	0	7.66
Dy-164	DY1647	MLBW	7.00E+03	70	7.38E+04	12	7.09	0	5.71
Ho-165	HO1657	MLBW	1.25E+03	250		0	7.45	0	6.24
Ho-166m	HO66M7	MLBW	6.50E+01	3	1.33E+05	36	7.26	0	7.29

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Er-162	ER1627	MLBW	2.50E+02	18		0	8.24	0	6.9
Er-164	ER1647	MLBW	8.00E+02	19		0	8.24	0	6.65
Er-166	ER1667	MLBW	5.00E+03	174		0	8.24	0	6.44
Er-167	ER1677	MLBW	1.75E+03	270	1.00E+04	0	7.84	0	7.77
Er-168	ER1687	MLBW	1.50E+04	130		0	7.65	0	6
Er-170	ER1707	MLBW	5.00E+03	126	7.91E+04	14	8.45	0	5.68
Lu-175	LU1757	MLBW	4.11E+02	115	1.00E+04	13	7.45	0	6.19
Lu-176	LU1767	MLBW	1.02E+02	59	1.00E+04	17	7.26	0	7.07
Hf-174	HF1747	MLBW	2.30E+02	11	9.00E+04	17	7.45	0	6.82
Hf-176	HF1767	MLBW	1.08E+03	24	9.00E+04	13	7.26	0	6.4
Hf-177	HF1777	MLBW	7.00E+02	180	9.00E+04	14	8.04	0	7.67
Hf-178	HF1787	MLBW	2.10E+03	25	9.00E+04	11	7.59	0	6.12
Hf-179	HF1797	MLBW	4.50E+02	71	9.00E+04	15	7.7	0	7.18
Hf-180	HF1807	MLBW	1.00E+04	155	9.00E+04	7	8.04	0	5.65
Ta-181	TA1817	MLBW	3.30E+02	76	5.00E+03	7	8.3	0	6.07
Ta-182	TA1827	MLBW	3.50E+01	10	1.00E+04	14	8.3	0	6.93
W -182	W182_7	MLBW	4.50E+03	69	1.00E+05	5	8	0	6.2
W -183	W183_7	MLBW	7.65E+02	50	4.50E+04	7	8	0	7.4
W -184	W184_7	MLBW	2.65E+03	38	1.00E+05	6	8	0	5.8
W -186	W186_7	MLBW	3.20E+03	40	1.00E+05	6	8	0	5.5
Re-185	RE1857	MLBW	2.00E+03	479		0	7.84	0	6.18
Re-187	RE1877	MLBW	2.00E+03	375	3.50E+04	9	7.84	0	5.87
Ir-191	IR1917	MLBW	1.53E+02	46		0	11.82	0	6.2
Ir-193	IR1937	MLBW	3.10E+02	40	7.34E+04	32	12.32	0	6.07
Au-197	AU1977	MLBW	5.00E+03	265		0	11.75	0	6.51
Hg-196	HG1967	MLBW	1.03E+02	2		0	11.34	0	6.98
Hg-198	HG1987	MLBW	4.59E+02	5		0	14.39	0	6.65
Hg-199	HG1997	MLBW	9.68E+02	10		0	13.33	0	8.03
Hg-200	HG2007	MLBW	8.58E+03	5		0	14.39	0	6.22
Hg-201	HG2017	MLBW	7.54E+02	8		0	14.39	0	7.76
Hg-202	HG2027	MLBW	4.52E+03	3		0	11.34	0	5.99
Hg-204	HG2047			0		0	11.34	0	5.67
Pb-204	PB2047	MLBW	5.00E+04	80		0	9.08	0	6.73
Pb-206	PB2067	RM	9.00E+05	381		0	11.34	0	6.74
Pb-207	PB2077	RM	4.75E+05	139		0	11.34	0	7.37
Pb-208	PB2087	RM	1.00E+06	80		0	11.8	0	3.94
Bi-209	BI2097	MLBW	1.00E+05	102		0	11.77	0	4.6
Ra-223	RA2237			0		0	12.39	200.00	6.5
Ra-224	RA2247			0		0	12.49	0	4.89
Ra-225	RA2257			0		0	12.39	0	6.39
Ra-226	RA2267	MLBW	1.00E+03	33		0	11.58	200.00	4.56
Ac-225	AC2257			0		0	12.39	0	5.38
Ac-226	AC2267			0		0	7.96	0	6.53
Ac-227	AC2277			0		0	12.39	200.00	5.04
Th-227	TH2277			0		0	12.39	200.00	7.13
Th-228	TH2287	MLBW	7.80E+00	5		0	15	185.00	5.25
Th-229	TH2297	MLBW	9.50E+00	15		0	12.57	200.00	6.79
Th-230	TH2307	SLBW	2.51E+02	22		0	8.7	190.60	5.12

Isotope	MC² ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Th-232	TH2327	RM	4.00E+03	927	1.00E+05	78	11.79	188.47	4.79
Th-233	TH2337			0		0	8.12	185.00	6.19
Th-234	TH2347			0		0	8.14	185.00	4.53
Pa-231	PA2317	RM	1.17E+02	136	7.80E+04	59	13.59	200.00	5.55
Pa-232	PA2327	MLBW	1.00E+01	14		0	12.29	200.00	6.52
Pa-233	PA2337	RM	1.07E+02	93	7.09E+04	56	14.66	200.00	5.22
U -232	U232_7	RM	1.94E+02	43	2.00E+03	10	12.07	189.00	5.74
U -233	U233_7	RM	6.00E+02	770	4.00E+04	29	11.63	191.04	6.84
U -234	U234_7	MLBW	1.50E+03	119	1.00E+05	10	10.02	191.84	5.3
U -235	U235_7	RM	2.25E+03	3193	2.50E+04	14	11.59	193.48	6.55
U -236	U236_7	MLBW	1.50E+03	117	1.00E+05	29	11	194.49	5.13
U -237	U237_7	SLBW	1.03E+02	27	1.00E+04	9	10.5	180.00	6.15
U -238	U238_7	RM	2.00E+04	3343	1.49E+05	18	11.29	198.03	4.81
U -239	U239_7	SLBW	1.03E+02	27	1.00E+04	9	10.5	180.00	5.93
U -240	U240_7	SLBW	9.86E+02	68	1.00E+04	5	10.69	198.00	4.58
U -241	U241_7	SLBW	1.03E+02	27	1.00E+04	9	10.5	180.00	5.68
Np-235	NP2357			0		0	11.39	200.00	5.74
Np-236	NP2367	MLBW	2.19E+01	32		0	12.14	200.00	6.57
Np-237	NP2377	MLBW	5.00E+02	760	3.50E+04	16	13.85	196.37	5.48
Np-238	NP2387			0		0	12.07	200.00	6.22
Np-239	NP2397			0		0	10.5	190.00	5.17
Pu-236	PU2367	MLBW	1.00E+01	4	3.00E+04	12	11.25	195.00	5.88
Pu-237	PU2377			0		0	10.52	195.90	7
Pu-238	PU2387	SLBW	2.00E+02	16	1.00E+04	3	10.89	197.38	4.81
Pu-239	PU2397	RM	2.50E+03	1070	3.00E+04	70	11.13	198.84	6.53
Pu-240	PU2407	MLBW	5.70E+03	268	4.00E+04	3	9.91	199.47	5.24
Pu-241	PU2417	RM	3.00E+02	244	4.02E+04	24	11.44	201.98	6.3
Pu-242	PU2427	SLBW	9.86E+02	68	1.00E+04	5	10.69	201.58	5.07
Pu-243	PU2437	SLBW	1.02E+02	41	1.00E+04	9	10.2	200.00	6.02
Pu-244	PU2447	SLBW	2.49E+02	21	1.00E+04	8	10.1	202.85	4.72
Pu-246	PU2467			0		0	10.8	200.00	5.97
Am-241	AM2417	SLBW	1.50E+02	195	3.00E+04	27	11.03	201.96	5.5
Am-242m	AM42M7	MLBW	1.00E+02	122	4.43E+04	15	11.15	200.00	6.4
Am-242	AM2427	MLBW	4.30E+01	106	2.73E+04	35	10.56	200.00	6.4
Am-243	AM2437	SLBW	2.50E+02	220	4.24E+04	17	11.82	203.62	5.4
Am-244	AM2447			0		0	8.02	200.00	6.1
Am-244m	AM44M7			0		0	8.02	200.00	6.1
Cm-241	CM2417			0		0	10.18	202.35	7.0
Cm-242	CM2427	SLBW	2.76E+02	13	1.00E+04	8	10.2	202.55	5.7
Cm-243	CM2437	SLBW	1.00E+02	105	4.22E+04	22	11.28	200.00	6.8
Cm-244	CM2447	MLBW	1.00E+03	68	4.00E+04	10	15.76	200.00	5.5
Cm-245	CM2457	MLBW	1.00E+02	91	5.50E+04	25	11.39	200.00	6.5
Cm-246	CM2467	MLBW	4.00E+02	17	4.30E+04	32	10.28	200.00	5.2
Cm-247	CM2477	SLBW	6.00E+01	44	3.00E+04	23	10.5	200.00	6.2
Cm-248	CM2487	SLBW	2.40E+03	46	1.00E+04	4	10.4	208.74	4.7
Cm-249	CM2497	MLBW	1.50E+02	11	2.50E+04	11	10.71	200.00	5.8
Cm-250	CM2507	MLBW	1.50E+02	4	3.00E+04	12	10.66	200.00	4.4
Bk-249	BK2497	MLBW	6.00E+01	40	3.00E+04	22	12.57	200.00	4.97

Isotope	MC^2 ID	Resolved Resonance (RR)			URR		Potential XS	Fission Energy	Capture Energy
		Type	Max E (eV)	#	Max E (eV)	#			
Bk-250	BK2507	MLBW	1.00E+02	57	3.00E+04	23	10.78	190.00	0.86
Cf-249	CF2497	MLBW	7.00E+01	64	3.00E+04	25	12.57	200.00	6.62
Cf-250	CF2507	SLBW	2.86E+02	20	1.00E+04	8	9.88	200.00	5.11
Cf-251	CF2517	SLBW	1.64E+02	20	1.00E+04	9	9.88	200.00	6.17
Cf-252	CF2527	SLBW	3.67E+02	21	1.00E+04	8	9.8	200.00	4.79
Cf-253	CF2537	SLBW	1.00E+02	120	1.00E+04	9	9.76	200.00	5.98
Cf-254	CF2547			0		0	8.24	200.00	4.45
Es-253	ES2537	SLBW	1.01E+02	27	1.00E+04	9	9.76	0	5.09
Es-254	ES2547			0		0	10.59	200.00	5.98
Es-255	ES2557			0		0	8.26	200.00	4.9
Fm-255	FM2557			0		0	10.59	200.00	6.39
DUMMY	DUMMY								

* SLBW: Single-Level Breit Wigner, MLBW: Multi-Level Breit Wigner, RM: Reich Moore, AA: Adler-Adler,
* Fission and capture energies in MeV

Appendix C. MC^2 Library for ENDF/B-V

V.0	V.2	V.0	V.2	V.0	V.2	V.0	V.2
HYDRGN		ZR95 5		TE127M		GD1585	
H-2 5		ZR96 5	ZR96SV	TE1285		GD1605	
H-3 5		NB93 5		TE129M		TB1595	
		NB93 S					
HE3 5		NB94 5		TE1305		TB1605	
HE4 5		NB95 5		TE1325		DY1605	
LI-6 5		MO92 5		I-1275		DY1615	
LI-7 5	LI-7 V	MO94 5		I-1295		DY1625	
BE-9 3		MO95 5		I-1305		DY1635	
B-11 5		MO96 5		I-1315		DY1645	
B-10 5		MO97 5		I-1355		HO1655	
C 5		MO98 5		XE1245		ER1665	
N-14 5		MO99 5		XE1265		ER1675	
N-15 5		MO1005		XE1285		LU1755	
O-16 5		MO 5		XE1295		LU1765	
		MO S					
O-17 5		TC99 5		XE1305		HF 5	
		TC99 S					
F-19 5		RU96 5		XE1315		HF1745	
NA23 5		RU98 5		XE1325		HF1765	
NA23 S							
MG 5		RU99 5		XE1335		HF1775	
AL27 5		RU1005		XE1345		HF1785	
SI 5		RU1015		XE1365		HF1795	
P-31 5		RU1025		CS1335		HF1805	
				CS133S			
S-32 5		RH1035		CS1345		TA1815	
S 5		RU1035		CS1355		TA1825	
CL 5		RU1045		CS1365		W-1825	W-182V
K 5		RU1055		CS1375		W-1835	W-183V
CA 5		RU1065		BA1345		W-1845	W-184V
TI 5		RH103S		BA1355		W-1865	W-186V
V 5		RH1055		BA1365		RE1855	
CR 5		PD1025		BA1375		RE1875	
CR S							
MN55 S		PD1045		BA1385		AU1975	
						AU197S	
MN55 5		PD1055		BA1405		PB 5	
FE 5	FE V	PD1065		LA1395		BI2095	
FE S	FE SV						
CO59 5		PD1075		LA1405		TH2305	
NI S		PD1085		CE1405		TH2325	
						TH232S	
CU 5		PD1105		CE1415		PA2315	
CU S							
GA 5		AG1075		CE1425		PA2335	
		AG107S					
GE72 5		AG1095		CE1435		U-2325	
		AG109S					
GE73 5		AG1115		CE1445		U-2335	
						U-233S	
GE74 5		CD 5		PR1415		U-2345	
GE76 5		CD1065		PR1425		U-2355	
						U-235S	
AS75 5		CD1085		PR1435		U-2365	
SE74 5		CD1105		ND1425		U-2375	
SE76 5		CD1115		ND1435		U-2385	
						U-238S	

SE77 5		CD1125		ND1445		NP2375	NP237V
						NP237S	
SE78 5		CD113S		ND1455		NP2385	
SE80 5		CD1135		ND1465		PU2365	
SE82 5		CD1145		ND1475		PU2375	
BR79 5		CD115M		ND1485		PU2385	
BR81 5		CD1165		ND1505		PU2395	PU239V
						PU239S	
V.0	**V.2**	**V.0**	**V.2**	**V.0**	**V.2**	**V.0**	**V.2**
KR85 5		IN1135		PM1475		PU2405	
						PU240S	
KR78 5		IN1155		PM1485		PU2415	
						PU241S	
KR80 5		SN1125		PM148M		PU2425	
KR82 5		SN1145		PM1495		PU2435	
KR83 5		SN1155		PM1515		PU2445	
KR84 5		SN1165		SM1445		AM2415	
KR86 5		SN1175		SM1475		AM2425	
RB85 5		SN1185		SM1485		AM242M	
RB86 5		SN1195		SM1495		AM2435	AM243V
				SM149S		AM243S	
RB87 5		SN1205		SM1505		CM2415	
SR84 5		SN1225		SM1515		CM2425	
SR86 5		SN1235		SM1525		CM2435	
						CM243S	
SR87 5		SN1245		SM1535		CM2445	
SR88 5		SN1255		SM1545		CM2455	
SR89 5		SN1265		EU1555		CM2465	
SR90 5		SB1215		EU1565		CM2475	
Y89 5		SB1235		EU1575		CM2485	
Y90 5		SB1245		EU1525		BK2495	
Y91 5		SB1255		EU1545		CF2495	
ZIRCSV		SB1265		EU1515		CF2505	
ZR 5	ZR V	TE1205		EU1535		CF2515	
ZR S	ZR SV						
ZR90 5	ZR90SV	TE1225		GD1525		CF2525	
ZR91 5	ZR91SV	TE1235		GD1545		CF2535	
ZR92 5	ZR92SV	TE1245		GD1555		ES2535	
ZR93 5		TE1255		GD1565			
ZR94 5	ZR94SV	TE1265		GD1575			

* ID ending with S: screened wide and narrow resonances.

Appendix D. MC^2 Gamma Library Table for ENDF/B-VII.0 Data

table_gammalib

```
393
1    2 H1___7   heatr_H_001      gaminr_H
2    2 H2___7   heatr_H_002      gaminr_H
3    2 H3___7   heatr_H_003      gaminr_H
4    2 HE3__7   heatr_He_003     gaminr_He
5    2 HE4__7   heatr_He_004     gaminr_He
6    2 LI6__7   heatr_Li_006     gaminr_Li
7    2 LI7__7   heatr_Li_007     gaminr_Li
8    2 BE7__7   heatr_Be_007     gaminr_Be
9    2 BE9__7   heatr_Be_009     gaminr_Be
10   2 B10__7   heatr_B_010      gaminr_B
11   2 B11__7   heatr_B_011      gaminr_B
12   2 C____7   heatr_C_000      gaminr_C
13   2 N14__7   heatr_N_014      gaminr_N
14   2 N15__7   heatr_N_015      gaminr_N
15   2 O16__7   heatr_O_016      gaminr_O
16   2 O17__7   heatr_O_017      gaminr_O
17   2 F19__7   heatr_F_019      gaminr_F
18   2 NA22_7   heatr_Na_022     gaminr_Na
19   2 NA23_7   heatr_Na_023     gaminr_Na
20   2 MG24_7   heatr_Mg_024     gaminr_Mg
21   2 MG25_7   heatr_Mg_025     gaminr_Mg
22   2 MG26_7   heatr_Mg_026     gaminr_Mg
23   2 AL27_7   heatr_Al_027     gaminr_Al
24   2 SI28_7   heatr_Si_028     gaminr_Si
25   2 SI29_7   heatr_Si_029     gaminr_Si
26   2 SI30_7   heatr_Si_030     gaminr_Si
27   2 P31__7   heatr_P_031      gaminr_P
28   2 S32__7   heatr_S_032      gaminr_S
29   2 S33__7   heatr_S_033      gaminr_S
30   2 S34__7   heatr_S_034      gaminr_S
31   2 S36__7   heatr_S_036      gaminr_S
32   2 CL35_7   heatr_Cl_035     gaminr_Cl
33   2 CL37_7   heatr_Cl_037     gaminr_Cl
34   2 AR36_7   heatr_Ar_036     gaminr_Ar
35   2 AR38_7   heatr_Ar_038     gaminr_Ar
36   2 AR40_7   heatr_Ar_040     gaminr_Ar
37   2 K39__7   heatr_K_039      gaminr_K
38   2 K40__7   heatr_K_040      gaminr_K
39   2 K41__7   heatr_K_041      gaminr_K
40   2 CA40_7   heatr_Ca_040     gaminr_Ca
41   2 CA42_7   heatr_Ca_042     gaminr_Ca
42   2 CA43_7   heatr_Ca_043     gaminr_Ca
43   2 CA44_7   heatr_Ca_044     gaminr_Ca
44   2 CA46_7   heatr_Ca_046     gaminr_Ca
45   2 CA48_7   heatr_Ca_048     gaminr_Ca
46   2 SC45_7   heatr_Sc_045     gaminr_Sc
47   2 TI46_7   heatr_Ti_046     gaminr_Ti
48   2 TI47_7   heatr_Ti_047     gaminr_Ti
49   2 TI48_7   heatr_Ti_048     gaminr_Ti
50   2 TI49_7   heatr_Ti_049     gaminr_Ti
51   2 TI50_7   heatr_Ti_050     gaminr_Ti
52   2 V____7   heatr_V_000      gaminr_V
53   2 CR50_7   heatr_Cr_050     gaminr_Cr
54   2 CR52_7   heatr_Cr_052     gaminr_Cr
55   2 CR53_7   heatr_Cr_053     gaminr_Cr
56   2 CR54_7   heatr_Cr_054     gaminr_Cr
57   2 MN55_7   heatr_Mn_055     gaminr_Mn
58   2 FE54_7   heatr_Fe_054     gaminr_Fe
59   2 FE56_7   heatr_Fe_056     gaminr_Fe
60   2 FE57_7   heatr_Fe_057     gaminr_Fe
61   2 FE58_7   heatr_Fe_058     gaminr_Fe
62   2 CO58_7   heatr_Co_058     gaminr_Co
63   2 CO58M7   heatr_Co_058m1   gaminr_Co
64   2 CO59_7   heatr_Co_059     gaminr_Co
65   2 NI58_7   heatr_Ni_058     gaminr_Ni
```

```
66  2 NI59_7   heatr_Ni_059     gaminr_Ni
67  2 NI60_7   heatr_Ni_060     gaminr_Ni
68  2 NI61_7   heatr_Ni_061     gaminr_Ni
69  2 NI62_7   heatr_Ni_062     gaminr_Ni
70  2 NI64_7   heatr_Ni_064     gaminr_Ni
71  2 CU63_7   heatr_Cu_063     gaminr_Cu
72  2 CU65_7   heatr_Cu_065     gaminr_Cu
73  2 ZN___7   heatr_Zn_000     gaminr_Zn
74  2 GA69_7   heatr_Ga_069     gaminr_Ga
75  2 GA71_7   heatr_Ga_071     gaminr_Ga
76  2 GE70_7   heatr_Ge_070     gaminr_Ge
77  2 GE72_7   heatr_Ge_072     gaminr_Ge
78  2 GE73_7   heatr_Ge_073     gaminr_Ge
79  2 GE74_7   heatr_Ge_074     gaminr_Ge
80  2 GE76_7   heatr_Ge_076     gaminr_Ge
81  2 AS74_7   heatr_As_074     gaminr_As
82  2 AS75_7   heatr_As_075     gaminr_As
83  2 SE74_7   heatr_Se_074     gaminr_Se
84  2 SE76_7   heatr_Se_076     gaminr_Se
85  2 SE77_7   heatr_Se_077     gaminr_Se
86  2 SE78_7   heatr_Se_078     gaminr_Se
87  2 SE79_7   heatr_Se_079     gaminr_Se
88  2 SE80_7   heatr_Se_080     gaminr_Se
89  2 SE82_7   heatr_Se_082     gaminr_Se
90  2 BR79_7   heatr_Br_079     gaminr_Br
91  2 BR81_7   heatr_Br_081     gaminr_Br
92  2 KR78_7   heatr_Kr_078     gaminr_Kr
93  2 KR80_7   heatr_Kr_080     gaminr_Kr
94  2 KR82_7   heatr_Kr_082     gaminr_Kr
95  2 KR83_7   heatr_Kr_083     gaminr_Kr
96  2 KR84_7   heatr_Kr_084     gaminr_Kr
97  2 KR85_7   heatr_Kr_085     gaminr_Kr
98  2 KR86_7   heatr_Kr_086     gaminr_Kr
99  2 RB85_7   heatr_Rb_085     gaminr_Rb
100 2 RB86_7   heatr_Rb_086     gaminr_Rb
101 2 RB87_7   heatr_Rb_087     gaminr_Rb
102 2 SR84_7   heatr_Sr_084     gaminr_Sr
103 2 SR86_7   heatr_Sr_086     gaminr_Sr
104 2 SR87_7   heatr_Sr_087     gaminr_Sr
105 2 SR88_7   heatr_Sr_088     gaminr_Sr
106 2 SR89_7   heatr_Sr_089     gaminr_Sr
107 2 SR90_7   heatr_Sr_090     gaminr_Sr
108 2 Y89__7   heatr_Y_089      gaminr_Y
109 2 Y90__7   heatr_Y_090      gaminr_Y
110 2 Y91__7   heatr_Y_091      gaminr_Y
111 2 ZR90_7   heatr_Zr_090     gaminr_Zr
112 2 ZR91_7   heatr_Zr_091     gaminr_Zr
113 2 ZR92_7   heatr_Zr_092     gaminr_Zr
114 2 ZR93_7   heatr_Zr_093     gaminr_Zr
115 2 ZR94_7   heatr_Zr_094     gaminr_Zr
116 2 ZR95_7   heatr_Zr_095     gaminr_Zr
117 2 ZR96_7   heatr_Zr_096     gaminr_Zr
118 2 NB93_7   heatr_Nb_093     gaminr_Nb
119 2 NB94_7   heatr_Nb_094     gaminr_Nb
120 2 NB95_7   heatr_Nb_095     gaminr_Nb
121 2 MO92_7   heatr_Mo_092     gaminr_Mo
122 2 MO94_7   heatr_Mo_094     gaminr_Mo
123 2 MO95_7   heatr_Mo_095     gaminr_Mo
124 2 MO96_7   heatr_Mo_096     gaminr_Mo
125 2 MO97_7   heatr_Mo_097     gaminr_Mo
126 2 MO98_7   heatr_Mo_098     gaminr_Mo
127 2 MO99_7   heatr_Mo_099     gaminr_Mo
128 2 MO1007   heatr_Mo_100     gaminr_Mo
129 2 TC99_7   heatr_Tc_099     gaminr_Tc
130 2 RU96_7   heatr_Ru_096     gaminr_Ru
131 2 RU98_7   heatr_Ru_098     gaminr_Ru
132 2 RU99_7   heatr_Ru_099     gaminr_Ru
133 2 RU1007   heatr_Ru_100     gaminr_Ru
134 2 RU1017   heatr_Ru_101     gaminr_Ru
135 2 RU1027   heatr_Ru_102     gaminr_Ru
136 2 RU1037   heatr_Ru_103     gaminr_Ru
```

```
137 2 RU1047   heatr_Ru_104     gaminr_Ru
138 2 RU1057   heatr_Ru_105     gaminr_Ru
139 2 RU1067   heatr_Ru_106     gaminr_Ru
140 2 RH1037   heatr_Rh_103     gaminr_Rh
141 2 RH1057   heatr_Rh_105     gaminr_Rh
142 2 PD1027   heatr_Pd_102     gaminr_Pd
143 2 PD1047   heatr_Pd_104     gaminr_Pd
144 2 PD1057   heatr_Pd_105     gaminr_Pd
145 2 PD1067   heatr_Pd_106     gaminr_Pd
146 2 PD1077   heatr_Pd_107     gaminr_Pd
147 2 PD1087   heatr_Pd_108     gaminr_Pd
148 2 PD1107   heatr_Pd_110     gaminr_Pd
149 2 AG1077   heatr_Ag_107     gaminr_Ag
150 2 AG1097   heatr_Ag_109     gaminr_Ag
151 2 AG10M7   heatr_Ag_110m1   gaminr_Ag
152 2 AG1117   heatr_Ag_111     gaminr_Ag
153 2 CD1067   heatr_Cd_106     gaminr_Cd
154 2 CD1087   heatr_Cd_108     gaminr_Cd
155 2 CD1107   heatr_Cd_110     gaminr_Cd
156 2 CD1117   heatr_Cd_111     gaminr_Cd
157 2 CD1127   heatr_Cd_112     gaminr_Cd
158 2 CD1137   heatr_Cd_113     gaminr_Cd
159 2 CD1147   heatr_Cd_114     gaminr_Cd
160 2 CD15M7   heatr_Cd_115m1   gaminr_Cd
161 2 CD1167   heatr_Cd_116     gaminr_Cd
162 2 IN1137   heatr_In_113     gaminr_In
163 2 IN1157   heatr_In_115     gaminr_In
164 2 SN1127   heatr_Sn_112     gaminr_Sn
165 2 SN1137   heatr_Sn_113     gaminr_Sn
166 2 SN1147   heatr_Sn_114     gaminr_Sn
167 2 SN1157   heatr_Sn_115     gaminr_Sn
168 2 SN1167   heatr_Sn_116     gaminr_Sn
169 2 SN1177   heatr_Sn_117     gaminr_Sn
170 2 SN1187   heatr_Sn_118     gaminr_Sn
171 2 SN1197   heatr_Sn_119     gaminr_Sn
172 2 SN1207   heatr_Sn_120     gaminr_Sn
173 2 SN1227   heatr_Sn_122     gaminr_Sn
174 2 SN1237   heatr_Sn_123     gaminr_Sn
175 2 SN1247   heatr_Sn_124     gaminr_Sn
176 2 SN1257   heatr_Sn_125     gaminr_Sn
177 2 SN1267   heatr_Sn_126     gaminr_Sn
178 2 SB1217   heatr_Sb_121     gaminr_Sb
179 2 SB1237   heatr_Sb_123     gaminr_Sb
180 2 SB1247   heatr_Sb_124     gaminr_Sb
181 2 SB1257   heatr_Sb_125     gaminr_Sb
182 2 SB1267   heatr_Sb_126     gaminr_Sb
183 2 TE1207   heatr_Te_120     gaminr_Te
184 2 TE1227   heatr_Te_122     gaminr_Te
185 2 TE1237   heatr_Te_123     gaminr_Te
186 2 TE1247   heatr_Te_124     gaminr_Te
187 2 TE1257   heatr_Te_125     gaminr_Te
188 2 TE1267   heatr_Te_126     gaminr_Te
189 2 TE27M7   heatr_Te_127m1   gaminr_Te
190 2 TE1287   heatr_Te_128     gaminr_Te
191 2 TE29M7   heatr_Te_129m1   gaminr_Te
192 2 TE1307   heatr_Te_130     gaminr_Te
193 2 TE1327   heatr_Te_132     gaminr_Te
194 2 I127_7   heatr_I_127      gaminr_I
195 2 I129_7   heatr_I_129      gaminr_I
196 2 I130_7   heatr_I_130      gaminr_I
197 2 I131_7   heatr_I_131      gaminr_I
198 2 I135_7   heatr_I_135      gaminr_I
199 2 XE1237   heatr_Xe_123     gaminr_Xe
200 2 XE1247   heatr_Xe_124     gaminr_Xe
201 2 XE1267   heatr_Xe_126     gaminr_Xe
202 2 XE1287   heatr_Xe_128     gaminr_Xe
203 2 XE1297   heatr_Xe_129     gaminr_Xe
204 2 XE1307   heatr_Xe_130     gaminr_Xe
205 2 XE1317   heatr_Xe_131     gaminr_Xe
206 2 XE1327   heatr_Xe_132     gaminr_Xe
207 2 XE1337   heatr_Xe_133     gaminr_Xe
```

```
208 2 XE1347   heatr_Xe_134     gaminr_Xe
209 2 XE1357   heatr_Xe_135     gaminr_Xe
210 2 XE1367   heatr_Xe_136     gaminr_Xe
211 2 CS1337   heatr_Cs_133     gaminr_Cs
212 2 CS1347   heatr_Cs_134     gaminr_Cs
213 2 CS1357   heatr_Cs_135     gaminr_Cs
214 2 CS1367   heatr_Cs_136     gaminr_Cs
215 2 CS1377   heatr_Cs_137     gaminr_Cs
216 2 BA1307   heatr_Ba_130     gaminr_Ba
217 2 BA1327   heatr_Ba_132     gaminr_Ba
218 2 BA1337   heatr_Ba_133     gaminr_Ba
219 2 BA1347   heatr_Ba_134     gaminr_Ba
220 2 BA1357   heatr_Ba_135     gaminr_Ba
221 2 BA1367   heatr_Ba_136     gaminr_Ba
222 2 BA1377   heatr_Ba_137     gaminr_Ba
223 2 BA1387   heatr_Ba_138     gaminr_Ba
224 2 BA1407   heatr_Ba_140     gaminr_Ba
225 2 LA1387   heatr_La_138     gaminr_La
226 2 LA1397   heatr_La_139     gaminr_La
227 2 LA1407   heatr_La_140     gaminr_La
228 2 CE1367   heatr_Ce_136     gaminr_Ce
229 2 CE1387   heatr_Ce_138     gaminr_Ce
230 2 CE1397   heatr_Ce_139     gaminr_Ce
231 2 CE1407   heatr_Ce_140     gaminr_Ce
232 2 CE1417   heatr_Ce_141     gaminr_Ce
233 2 CE1427   heatr_Ce_142     gaminr_Ce
234 2 CE1437   heatr_Ce_143     gaminr_Ce
235 2 CE1447   heatr_Ce_144     gaminr_Ce
236 2 PR1417   heatr_Pr_141     gaminr_Pr
237 2 PR1427   heatr_Pr_142     gaminr_Pr
238 2 PR1437   heatr_Pr_143     gaminr_Pr
239 2 ND1427   heatr_Nd_142     gaminr_Nd
240 2 ND1437   heatr_Nd_143     gaminr_Nd
241 2 ND1447   heatr_Nd_144     gaminr_Nd
242 2 ND1457   heatr_Nd_145     gaminr_Nd
243 2 ND1467   heatr_Nd_146     gaminr_Nd
244 2 ND1477   heatr_Nd_147     gaminr_Nd
245 2 ND1487   heatr_Nd_148     gaminr_Nd
246 2 ND1507   heatr_Nd_150     gaminr_Nd
247 2 PM1477   heatr_Pm_147     gaminr_Pm
248 2 PM1487   heatr_Pm_148     gaminr_Pm
249 2 PM48M7   heatr_Pm_148m1   gaminr_Pm
250 2 PM1497   heatr_Pm_149     gaminr_Pm
251 2 PM1517   heatr_Pm_151     gaminr_Pm
252 2 SM1447   heatr_Sm_144     gaminr_Sm
253 2 SM1477   heatr_Sm_147     gaminr_Sm
254 2 SM1487   heatr_Sm_148     gaminr_Sm
255 2 SM1497   heatr_Sm_149     gaminr_Sm
256 2 SM1507   heatr_Sm_150     gaminr_Sm
257 2 SM1517   heatr_Sm_151     gaminr_Sm
258 2 SM1527   heatr_Sm_152     gaminr_Sm
259 2 SM1537   heatr_Sm_153     gaminr_Sm
260 2 SM1547   heatr_Sm_154     gaminr_Sm
261 2 EU1517   heatr_Eu_151     gaminr_Eu
262 2 EU1527   heatr_Eu_152     gaminr_Eu
263 2 EU1537   heatr_Eu_153     gaminr_Eu
264 2 EU1547   heatr_Eu_154     gaminr_Eu
265 2 EU1557   heatr_Eu_155     gaminr_Eu
266 2 EU1567   heatr_Eu_156     gaminr_Eu
267 2 EU1577   heatr_Eu_157     gaminr_Eu
268 2 GD1527   heatr_Gd_152     gaminr_Gd
269 2 GD1537   heatr_Gd_153     gaminr_Gd
270 2 GD1547   heatr_Gd_154     gaminr_Gd
271 2 GD1557   heatr_Gd_155     gaminr_Gd
272 2 GD1567   heatr_Gd_156     gaminr_Gd
273 2 GD1577   heatr_Gd_157     gaminr_Gd
274 2 GD1587   heatr_Gd_158     gaminr_Gd
275 2 GD1607   heatr_Gd_160     gaminr_Gd
276 2 TB1597   heatr_Tb_159     gaminr_Tb
277 2 TB1607   heatr_Tb_160     gaminr_Tb
278 2 DY1567   heatr_Dy_156     gaminr_Dy
```

```
279 2 DY1587   heatr_Dy_158     gaminr_Dy
280 2 DY1607   heatr_Dy_160     gaminr_Dy
281 2 DY1617   heatr_Dy_161     gaminr_Dy
282 2 DY1627   heatr_Dy_162     gaminr_Dy
283 2 DY1637   heatr_Dy_163     gaminr_Dy
284 2 DY1647   heatr_Dy_164     gaminr_Dy
285 2 HO1657   heatr_Ho_165     gaminr_Ho
286 2 HO66M7   heatr_Ho_166m1   gaminr_Ho
287 2 ER1627   heatr_Er_162     gaminr_Er
288 2 ER1647   heatr_Er_164     gaminr_Er
289 2 ER1667   heatr_Er_166     gaminr_Er
290 2 ER1677   heatr_Er_167     gaminr_Er
291 2 ER1687   heatr_Er_168     gaminr_Er
292 2 ER1707   heatr_Er_170     gaminr_Er
293 2 LU1757   heatr_Lu_175     gaminr_Lu
294 2 LU1767   heatr_Lu_176     gaminr_Lu
295 2 HF1747   heatr_Hf_174     gaminr_Hf
296 2 HF1767   heatr_Hf_176     gaminr_Hf
297 2 HF1777   heatr_Hf_177     gaminr_Hf
298 2 HF1787   heatr_Hf_178     gaminr_Hf
299 2 HF1797   heatr_Hf_179     gaminr_Hf
300 2 HF1807   heatr_Hf_180     gaminr_Hf
301 2 TA1817   heatr_Ta_181     gaminr_Ta
302 2 TA1827   heatr_Ta_182     gaminr_Ta
303 2 W182_7   heatr_W_182      gaminr_W
304 2 W183_7   heatr_W_183      gaminr_W
305 2 W184_7   heatr_W_184      gaminr_W
306 2 W186_7   heatr_W_186      gaminr_W
307 2 RE1857   heatr_Re_185     gaminr_Re
308 2 RE1877   heatr_Re_187     gaminr_Re
309 2 IR1917   heatr_Ir_191     gaminr_Ir
310 2 IR1937   heatr_Ir_193     gaminr_Ir
311 2 AU1977   heatr_Au_197     gaminr_Au
312 2 HG1967   heatr_Hg_196     gaminr_Hg
313 2 HG1987   heatr_Hg_198     gaminr_Hg
314 2 HG1997   heatr_Hg_199     gaminr_Hg
315 2 HG2007   heatr_Hg_200     gaminr_Hg
316 2 HG2017   heatr_Hg_201     gaminr_Hg
317 2 HG2027   heatr_Hg_202     gaminr_Hg
318 2 HG2047   heatr_Hg_204     gaminr_Hg
319 2 PB2047   heatr_Pb_204     gaminr_Pb
320 2 PB2067   heatr_Pb_206     gaminr_Pb
321 2 PB2077   heatr_Pb_207     gaminr_Pb
322 2 PB2087   heatr_Pb_208     gaminr_Pb
323 2 BI2097   heatr_Bi_209     gaminr_Bi
324 2 RA2237   heatr_Ra_223     gaminr_Ra
325 2 RA2247   heatr_Ra_224     gaminr_Ra
326 2 RA2257   heatr_Ra_225     gaminr_Ra
327 2 RA2267   heatr_Ra_226     gaminr_Ra
328 3 AC2257   heatr_Ac_225     gaminr_Ac    dbeta_89225
329 3 AC2267   heatr_Ac_226     gaminr_Ac    dbeta_89226
330 3 AC2277   heatr_Ac_227     gaminr_Ac    dbeta_89227
331 3 TH2277   heatr_Th_227     gaminr_Th    dbeta_90227    dgamma_90227
332 3 TH2287   heatr_Th_228     gaminr_Th    dbeta_90228
333 3 TH2297   heatr_Th_229     gaminr_Th    dbeta_90229    dgamma_90229
334 3 TH2307   heatr_Th_230     gaminr_Th    dbeta_90230
335 3 TH2327   heatr_Th_232     gaminr_Th    dbeta_90232    dgamma_90232
336 3 TH2337   heatr_Th_233     gaminr_Th    dbeta_90233
337 3 TH2347   heatr_Th_234     gaminr_Th    dbeta_90234
338 4 PA2317   heatr_Pa_231     gaminr_Pa    dbeta_91231    dgamma_91231
339 3 PA2327   heatr_Pa_232     gaminr_Pa    dbeta_91232
340 3 PA2337   heatr_Pa_233     gaminr_Pa    dbeta_91233
341 4 U232_7   heatr_U_232      gaminr_U     dbeta_92232    dgamma_92232
342 4 U233_7   heatr_U_233      gaminr_U     dbeta_92233    dgamma_92233
343 4 U234_7   heatr_U_234      gaminr_U     dbeta_92234    dgamma_92234
344 4 U235_7   heatr_U_235      gaminr_U     dbeta_92235    dgamma_92235
345 4 U236_7   heatr_U_236      gaminr_U     dbeta_92236    dgamma_92236
346 4 U237_7   heatr_U_237      gaminr_U     dbeta_92237    dgamma_92237
347 4 U238_7   heatr_U_238      gaminr_U     dbeta_92238    dgamma_92238
348 3 U239_7   heatr_U_239      gaminr_U     dbeta_92239
349 3 U240_7   heatr_U_240      gaminr_U     dbeta_92240
```

```
350 3 U241_7   heatr_U_241      gaminr_U     dbeta_92241
351 3 NP2357   heatr_Np_235     gaminr_Np    dbeta_93235
352 3 NP2367   heatr_Np_236     gaminr_Np    dbeta_93236
353 4 NP2377   heatr_Np_237     gaminr_Np    dbeta_93237    dgamma_93237
354 4 NP2387   heatr_Np_238     gaminr_Np    dbeta_93238    dgamma_93238
355 3 NP2397   heatr_Np_239     gaminr_Np    dbeta_93239
356 3 PU2367   heatr_Pu_236     gaminr_Pu    dbeta_94236
357 3 PU2377   heatr_Pu_237     gaminr_Pu    dbeta_94237
358 4 PU2387   heatr_Pu_238     gaminr_Pu    dbeta_94238    dgamma_94238
359 4 PU2397   heatr_Pu_239     gaminr_Pu    dbeta_94239    dgamma_94239
360 4 PU2407   heatr_Pu_240     gaminr_Pu    dbeta_94240    dgamma_94240
361 4 PU2417   heatr_Pu_241     gaminr_Pu    dbeta_94241    dgamma_94241
362 4 PU2427   heatr_Pu_242     gaminr_Pu    dbeta_94242    dgamma_94242
363 3 PU2437   heatr_Pu_243     gaminr_Pu    dbeta_94243
364 3 PU2447   heatr_Pu_244     gaminr_Pu    dbeta_94244
365 3 PU2467   heatr_Pu_246     gaminr_Pu    dbeta_94246
366 4 AM2417   heatr_Am_241     gaminr_Am    dbeta_95241    dgamma_95241
367 4 AM2427   heatr_Am_242     gaminr_Am    dbeta_95242    dgamma_95242
368 4 AM42M7   heatr_Am_242m1   gaminr_Am    dbeta_95342    dgamma_95342
369 4 AM2437   heatr_Am_243     gaminr_Am    dbeta_95243    dgamma_95243
370 3 AM2447   heatr_Am_244     gaminr_Am    dbeta_95244
371 3 AM44M7   heatr_Am_244m1   gaminr_Am    dbeta_95344
372 3 CM2417   heatr_Cm_241     gaminr_Cm    dbeta_96241
373 4 CM2427   heatr_Cm_242     gaminr_Cm    dbeta_96242    dgamma_96242
374 4 CM2437   heatr_Cm_243     gaminr_Cm    dbeta_96243    dgamma_96243
375 4 CM2447   heatr_Cm_244     gaminr_Cm    dbeta_96244    dgamma_96244
376 4 CM2457   heatr_Cm_245     gaminr_Cm    dbeta_96245    dgamma_96245
377 4 CM2467   heatr_Cm_246     gaminr_Cm    dbeta_96246    dgamma_96246
378 3 CM2477   heatr_Cm_247     gaminr_Cm    dbeta_96247
379 4 CM2487   heatr_Cm_248     gaminr_Cm    dbeta_96248    dgamma_96248
380 3 CM2497   heatr_Cm_249     gaminr_Cm    dbeta_96249
381 3 CM2507   heatr_Cm_250     gaminr_Cm    dbeta_96250
382 3 BK2497   heatr_Bk_249     gaminr_Bk    dbeta_97249
383 3 BK2507   heatr_Bk_250     gaminr_Bk    dbeta_97250
384 4 CF2497   heatr_Cf_249     gaminr_Cf    dbeta_98249    dgamma_98249
385 4 CF2507   heatr_Cf_250     gaminr_Cf    dbeta_98250    dgamma_98251
386 3 CF2517   heatr_Cf_251     gaminr_Cf    dbeta_98251
387 3 CF2527   heatr_Cf_252     gaminr_Cf    dbeta_98252
388 3 CF2537   heatr_Cf_253     gaminr_Cf    dbeta_98253
389 3 CF2547   heatr_Cf_254     gaminr_Cf    dbeta_98254
390 2 ES2537   heatr_Es_253     gaminr_Es
391 4 ES2547   heatr_Es_254     gaminr_Es    dbeta_99254    dgamma_99254
392 2 ES2557   heatr_Es_255     gaminr_Es
393 3 FM2557   heatr_Fm_255     gaminr_Fm    dbeta_100255
```

Appendix E. Built-in Multigroup Structures

Δu = 1/120: ultrafine group lethargy width.

Upper energy boundaries in eV are shown in the tables.

The total number of all Δu should be 2082.

ANL4

582 Δu , 3*(500 Δu)

	1	2	3	4
1	1.4191E+07	1.1109E+05	1.7223E+03	2.6703E+01

ANL9

222 Δu , 120 Δu , 5*(180 Δu), 540 Δu , 300 Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	2.2313E+06	8.2085E+05	1.8316E+05	4.0868E+04	9.1188E+03	2.0347E+03	4.5400E+02
9	5.0435E+00							

ANL33

42 Δu , 28*(60 Δu), 90 Δu , 240 Δu , 29 Δu , Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	1.0000E+07	6.0653E+06	3.6788E+06	2.2313E+06	1.3534E+06	8.2085E+05	4.9787E+05
9	3.0197E+05	1.8316E+05	1.1109E+05	6.7379E+04	4.0868E+04	2.4787E+04	1.5034E+04	9.1188E+03
17	5.5308E+03	3.3546E+03	2.0347E+03	1.2341E+03	7.4852E+02	4.5400E+02	2.7536E+02	1.6702E+02
25	1.0130E+02	6.1442E+01	3.7267E+01	2.2603E+01	1.3710E+01	8.3153E+00	3.9279E+00	5.3158E-01
33	4.1746E-01							

ANL70

42 Δu , 67*(30 Δu), 29 Δu , Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	1.0000E+07	7.7880E+06	6.0653E+06	4.7237E+06	3.6788E+06	2.8650E+06	2.2313E+06
9	1.7377E+06	1.3534E+06	1.0540E+06	8.2085E+05	6.3928E+05	4.9787E+05	3.8774E+05	3.0197E+05
17	2.3518E+05	1.8316E+05	1.4264E+05	1.1109E+05	8.6517E+04	6.7379E+04	5.2475E+04	4.0868E+04
25	3.1828E+04	2.4787E+04	1.9305E+04	1.5034E+04	1.1709E+04	9.1188E+03	7.1017E+03	5.5308E+03
33	4.3074E+03	3.3546E+03	2.6126E+03	2.0347E+03	1.5846E+03	1.2341E+03	9.6112E+02	7.4852E+02
41	5.8295E+02	4.5400E+02	3.5357E+02	2.7536E+02	2.1445E+02	1.6702E+02	1.3007E+02	1.0130E+02
49	7.8893E+01	6.1442E+01	4.7851E+01	3.7267E+01	2.9023E+01	2.2603E+01	1.7603E+01	1.3710E+01
57	1.0677E+01	8.3153E+00	6.4759E+00	5.0435E+00	3.9279E+00	3.0590E+00	2.3824E+00	1.8554E+00
65	1.4450E+00	1.1254E+00	8.7642E-01	6.8256E-01	5.3158E-01	4.1746E-01		

ANL116

15*(6 Δu), 3 Δu , 2*(6 Δu), 3 Δu , 12 Δu , 3*(6 Δu), 3*(12 Δu), 2*(6 Δu), 2*(12 Δu), 4 Δu , 6 Δu , 2 Δu , 12 Δu , 2*(6 Δu), 12 Δu , 2*(6 Δu),2*(12 Δu), 6 Δu , 12 Δu , 2*(6 Δu), 6*(12 Δu), 6 Δu , 4*(12 Δu), 4*(6 Δu), 5*(12 Δu), 6 Δu , 3*(12 Δu), 6 Δu , 2*(30 Δu), 2*(15 Δu), 30 Δu , 4*(15 Δu), 18 Δu , 12 Δu , 5*(30 Δu), 24 Δu , 12 Δu , 24 Δu , 19 Δu , 11 Δu , 18 Δu , 24 Δu , 3*(18 Δu), 2*(12 Δu), 14*(60 Δu), 2*(30 Δu), 29 Δu , Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	1.3499E+07	1.2840E+07	1.2214E+07	1.1618E+07	1.1052E+07	1.0513E+07	1.0000E+07
9	9.5123E+06	9.0484E+06	8.6071E+06	8.1873E+06	7.7880E+06	7.4082E+06	7.0469E+06	6.7032E+06
17	6.5377E+06	6.2189E+06	5.9156E+06	5.7695E+06	5.2205E+06	4.9659E+06	4.7237E+06	4.4933E+06
25	4.0657E+06	3.6788E+06	3.3287E+06	3.1664E+06	3.0119E+06	2.7253E+06	2.4660E+06	2.3851E+06
33	2.2688E+06	2.2313E+06	2.0190E+06	1.9205E+06	1.8268E+06	1.6530E+06	1.5724E+06	1.4957E+06
41	1.3534E+06	1.2246E+06	1.1648E+06	1.0540E+06	1.0026E+06	9.5369E+05	8.6294E+05	7.8082E+05
49	7.0651E+05	6.3928E+05	5.7844E+05	5.2340E+05	4.9787E+05	4.5049E+05	4.0762E+05	3.6883E+05
57	3.3373E+05	3.1746E+05	3.0197E+05	2.8725E+05	2.7324E+05	2.4724E+05	2.2371E+05	2.0242E+05
65	1.8316E+05	1.6573E+05	1.5764E+05	1.4264E+05	1.2907E+05	1.1679E+05	1.1109E+05	8.6517E+04
73	6.7379E+04	5.9462E+04	5.2475E+04	4.0868E+04	3.6066E+04	3.1828E+04	2.8088E+04	2.4787E+04
81	2.1335E+04	1.9305E+04	1.5034E+04	1.1709E+04	9.1188E+03	7.1017E+03	5.5308E+03	4.5283E+03
89	4.0973E+03	3.3546E+03	2.8634E+03	2.6126E+03	2.2487E+03	1.8411E+03	1.5846E+03	1.3639E+03
97	1.1739E+03	1.0622E+03	9.6112E+02	5.8295E+02	3.5357E+02	2.1445E+02	1.3007E+02	7.8893E+01
105	4.7851E+01	2.9023E+01	1.7603E+01	1.0677E+01	6.4759E+00	3.9279E+00	2.3824E+00	1.4450E+00
113	8.7642E-01	6.8256E-01	5.3158E-01	4.1746E-01				

ANL230

30*(3 Δu), 3*(Δu), 5*(3 Δu), 3*(6 Δu), 4*(3 Δu), 6*(6 Δu), 4*(3 Δu), 4*(6 Δu), 2*(2 Δu), 2*(Δu), 3*(2 Δu), 2*(6 Δu), 4*(3 Δu), 2*(6 Δu), 4*(3 Δu), 4*(6 Δu), 2*(3 Δu), 3*(6 Δu), 3 Δu , 2 Δu , Δu , 11*(6 Δu), 4*(3 Δu), 9*(6 Δu), 6*(3 Δu), 9*(6 Δu), 4*(3 Δu), 7*(6 Δu), 4*(15 Δu), 9 Δu , 2*(6 Δu), 9 Δu , 3*(15 Δu), 2*(3 Δu), 9 Δu , 15 Δu , 9 Δu , 6 Δu , 2*(3 Δu), 9 Δu , 3 Δu , 12 Δu , 10*(15 Δu), 2*(12 Δu), 2*(6 Δu), 3*(12 Δu), 7 Δu , 5 Δu , 2*(6 Δu), 4*(12 Δu), 2*(6 Δu), 2*(12 Δu), 5*(6 Δu), 28*(30 Δu), 6 Δu , 24 Δu , 10 Δu , 20 Δu , 29 Δu , Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	1.3840E+07	1.3499E+07	1.3165E+07	1.2840E+07	1.2523E+07	1.2214E+07	1.1912E+07
9	1.1618E+07	1.1331E+07	1.1052E+07	1.0779E+07	1.0513E+07	1.0253E+07	1.0000E+07	9.7531E+06
17	9.5123E+06	9.2774E+06	9.0484E+06	8.8250E+06	8.6071E+06	8.3946E+06	8.1873E+06	7.9852E+06
25	7.7880E+06	7.5957E+06	7.4082E+06	7.2253E+06	7.0469E+06	6.8729E+06	6.7032E+06	6.6476E+06
33	6.5924E+06	6.5377E+06	6.3763E+06	6.2189E+06	6.0653E+06	5.9156E+06	5.7695E+06	5.4881E+06
41	5.2205E+06	4.9659E+06	4.8432E+06	4.7237E+06	4.6070E+06	4.4933E+06	4.2741E+06	4.0657E+06
49	3.8674E+06	3.6788E+06	3.4994E+06	3.3287E+06	3.2465E+06	3.1664E+06	3.0882E+06	3.0119E+06
57	2.8650E+06	2.7253E+06	2.5924E+06	2.4660E+06	2.4252E+06	2.3851E+06	2.3653E+06	2.3457E+06
65	2.3069E+06	2.2688E+06	2.2313E+06	2.1225E+06	2.0190E+06	1.9691E+06	1.9205E+06	1.8731E+06
73	1.8268E+06	1.7377E+06	1.6530E+06	1.6122E+06	1.5724E+06	1.5336E+06	1.4957E+06	1.4227E+06
81	1.3534E+06	1.2873E+06	1.2246E+06	1.1943E+06	1.1648E+06	1.1080E+06	1.0540E+06	1.0026E+06
89	9.7783E+05	9.6167E+05	9.5369E+05	9.0718E+05	8.6294E+05	8.2085E+05	7.8082E+05	7.4274E+05
97	7.0651E+05	6.7206E+05	6.3928E+05	6.0810E+05	5.7844E+05	5.5023E+05	5.3665E+05	5.2340E+05

105	5.1047E+05	4.9787E+05	4.7359E+05	4.5049E+05	4.2852E+05	4.0762E+05	3.8774E+05	3.6883E+05
113	3.5084E+05	3.3373E+05	3.1746E+05	3.0962E+05	3.0197E+05	2.9452E+05	2.8725E+05	2.8015E+05
121	2.7324E+05	2.5991E+05	2.4724E+05	2.3518E+05	2.2371E+05	2.1280E+05	2.0242E+05	1.9255E+05
129	1.8316E+05	1.7422E+05	1.6992E+05	1.6573E+05	1.6163E+05	1.5764E+05	1.4996E+05	1.4264E+05
137	1.3569E+05	1.2907E+05	1.2277E+05	1.1679E+05	1.1109E+05	9.8036E+04	8.6517E+04	7.6351E+04
145	6.7379E+04	6.2511E+04	5.9462E+04	5.6562E+04	5.2475E+04	4.6309E+04	4.0868E+04	3.6066E+04
153	3.5175E+04	3.4307E+04	3.1828E+04	2.8088E+04	2.6058E+04	2.4787E+04	2.4175E+04	2.3579E+04
161	2.1875E+04	2.1335E+04	1.9305E+04	1.7036E+04	1.5034E+04	1.3268E+04	1.1709E+04	1.0333E+04
169	9.1188E+03	8.0473E+03	7.1017E+03	6.2673E+03	5.5308E+03	5.0045E+03	4.5283E+03	4.3074E+03
177	4.0973E+03	3.7074E+03	3.3546E+03	3.0354E+03	2.8634E+03	2.7465E+03	2.6126E+03	2.4852E+03
185	2.2487E+03	2.0347E+03	1.8411E+03	1.6659E+03	1.5846E+03	1.5073E+03	1.3639E+03	1.2341E+03
193	1.1739E+03	1.1167E+03	1.0622E+03	1.0104E+03	9.6112E+02	7.4852E+02	5.8295E+02	4.5400E+02
201	3.5357E+02	2.7536E+02	2.1445E+02	1.6702E+02	1.3007E+02	1.0130E+02	7.8893E+01	6.1442E+01
209	4.7851E+01	3.7267E+01	2.9023E+01	2.2603E+01	1.7603E+01	1.3710E+01	1.0677E+01	8.3153E+00
217	6.4759E+00	5.0435E+00	3.9279E+00	3.0590E+00	2.3824E+00	1.8554E+00	1.4450E+00	1.1254E+00
225	8.7642E-01	8.3368E-01	6.8256E-01	6.2799E-01	5.3158E-01	4.1746E-01		

ANL425

30*(3 Δu), 3* Δu , 39*(3 Δu), 2*(2 Δu), 2* Δu , 3*(2 Δu), 31*(3 Δu), 2 Δu , Δu , 222*(3 Δu), Δu , 3 Δu , 2 Δu , 42*(3 Δu), 28*(30 Δu), 6 Δu , 24 Δu , 10 Δu , 20 Δu , 29 Δu , Δu

	1	2	3	4	5	6	7	8
1	1.4191E+07	1.3840E+07	1.3499E+07	1.3165E+07	1.2840E+07	1.2523E+07	1.2214E+07	1.1912E+07
9	1.1618E+07	1.1331E+07	1.1052E+07	1.0779E+07	1.0513E+07	1.0253E+07	1.0000E+07	9.7531E+06
17	9.5123E+06	9.2774E+06	9.0484E+06	8.8250E+06	8.6071E+06	8.3946E+06	8.1873E+06	7.9852E+06
25	7.7880E+06	7.5957E+06	7.4082E+06	7.2253E+06	7.0469E+06	6.8729E+06	6.7032E+06	6.6476E+06
33	6.5924E+06	6.5377E+06	6.3763E+06	6.2189E+06	6.0653E+06	5.9156E+06	5.7695E+06	5.6270E+06
41	5.4881E+06	5.3526E+06	5.2205E+06	5.0916E+06	4.9659E+06	4.8432E+06	4.7237E+06	4.6070E+06
49	4.4933E+06	4.3824E+06	4.2741E+06	4.1686E+06	4.0657E+06	3.9653E+06	3.8674E+06	3.7719E+06
57	3.6788E+06	3.5880E+06	3.4994E+06	3.4130E+06	3.3287E+06	3.2465E+06	3.1664E+06	3.0882E+06
65	3.0119E+06	2.9376E+06	2.8650E+06	2.7943E+06	2.7253E+06	2.6580E+06	2.5924E+06	2.5284E+06
73	2.4660E+06	2.4252E+06	2.3851E+06	2.3653E+06	2.3457E+06	2.3069E+06	2.2688E+06	2.2313E+06
81	2.1762E+06	2.1225E+06	2.0701E+06	2.0190E+06	1.9691E+06	1.9205E+06	1.8731E+06	1.8268E+06
89	1.7817E+06	1.7377E+06	1.6948E+06	1.6530E+06	1.6122E+06	1.5724E+06	1.5336E+06	1.4957E+06
97	1.4588E+06	1.4227E+06	1.3876E+06	1.3534E+06	1.3199E+06	1.2873E+06	1.2556E+06	1.2246E+06
105	1.1943E+06	1.1648E+06	1.1361E+06	1.1080E+06	1.0807E+06	1.0540E+06	1.0280E+06	1.0026E+06
113	9.7783E+05	9.6167E+05	9.5369E+05	9.3014E+05	9.0718E+05	8.8478E+05	8.6294E+05	8.4163E+05
121	8.2085E+05	8.0058E+05	7.8082E+05	7.6154E+05	7.4274E+05	7.2440E+05	7.0651E+05	6.8907E+05
129	6.7206E+05	6.5546E+05	6.3928E+05	6.2349E+05	6.0810E+05	5.9309E+05	5.7844E+05	5.6416E+05
137	5.5023E+05	5.3665E+05	5.2340E+05	5.1047E+05	4.9787E+05	4.8558E+05	4.7359E+05	4.6190E+05
145	4.5049E+05	4.3937E+05	4.2852E+05	4.1794E+05	4.0762E+05	3.9756E+05	3.8774E+05	3.7817E+05
153	3.6883E+05	3.5973E+05	3.5084E+05	3.4218E+05	3.3373E+05	3.2549E+05	3.1746E+05	3.0962E+05
161	3.0197E+05	2.9452E+05	2.8725E+05	2.8015E+05	2.7324E+05	2.6649E+05	2.5991E+05	2.5349E+05
169	2.4724E+05	2.4113E+05	2.3518E+05	2.2937E+05	2.2371E+05	2.1818E+05	2.1280E+05	2.0754E+05
177	2.0242E+05	1.9742E+05	1.9255E+05	1.8779E+05	1.8316E+05	1.7863E+05	1.7422E+05	1.6992E+05
185	1.6573E+05	1.6163E+05	1.5764E+05	1.5375E+05	1.4996E+05	1.4625E+05	1.4264E+05	1.3912E+05

193	1.3569E+05	1.3234E+05	1.2907E+05	1.2588E+05	1.2277E+05	1.1974E+05	1.1679E+05	1.1390E+05
201	1.1109E+05	1.0835E+05	1.0567E+05	1.0306E+05	1.0052E+05	9.8036E+04	9.5616E+04	9.3255E+04
209	9.0953E+04	8.8707E+04	8.6517E+04	8.4381E+04	8.2297E+04	8.0265E+04	7.8284E+04	7.6351E+04
217	7.4466E+04	7.2627E+04	7.0834E+04	6.9085E+04	6.7379E+04	6.5716E+04	6.4093E+04	6.2511E+04
225	6.0967E+04	5.9462E+04	5.7994E+04	5.6562E+04	5.5166E+04	5.3804E+04	5.2475E+04	5.1180E+04
233	4.9916E+04	4.8683E+04	4.7481E+04	4.6309E+04	4.5166E+04	4.4051E+04	4.2963E+04	4.1902E+04
241	4.0868E+04	3.9859E+04	3.8875E+04	3.7915E+04	3.6979E+04	3.6066E+04	3.5175E+04	3.4307E+04
249	3.3460E+04	3.2634E+04	3.1828E+04	3.1042E+04	3.0276E+04	2.9528E+04	2.8799E+04	2.8088E+04
257	2.7394E+04	2.6718E+04	2.6058E+04	2.5415E+04	2.4787E+04	2.4175E+04	2.3579E+04	2.2996E+04
265	2.2429E+04	2.1875E+04	2.1335E+04	2.0808E+04	2.0294E+04	1.9793E+04	1.9305E+04	1.8828E+04
273	1.8363E+04	1.7910E+04	1.7467E+04	1.7036E+04	1.6616E+04	1.6205E+04	1.5805E+04	1.5415E+04
281	1.5034E+04	1.4663E+04	1.4301E+04	1.3948E+04	1.3604E+04	1.3268E+04	1.2940E+04	1.2621E+04
289	1.2309E+04	1.2005E+04	1.1709E+04	1.1420E+04	1.1138E+04	1.0863E+04	1.0595E+04	1.0333E+04
297	1.0078E+04	9.8290E+03	9.5863E+03	9.3497E+03	9.1188E+03	8.8937E+03	8.6741E+03	8.4599E+03
305	8.2510E+03	8.0473E+03	7.8486E+03	7.6548E+03	7.4659E+03	7.2815E+03	7.1017E+03	6.9264E+03
313	6.7554E+03	6.5886E+03	6.4259E+03	6.2673E+03	6.1125E+03	5.9616E+03	5.8144E+03	5.6709E+03
321	5.5308E+03	5.3943E+03	5.2611E+03	5.1312E+03	5.0045E+03	4.8809E+03	4.7604E+03	4.6429E+03
329	4.5283E+03	4.4165E+03	4.3074E+03	4.2011E+03	4.0973E+03	3.9962E+03	3.8975E+03	3.8013E+03
337	3.7074E+03	3.6159E+03	3.5266E+03	3.4395E+03	3.3546E+03	3.2718E+03	3.1910E+03	3.1122E+03
345	3.0354E+03	2.9604E+03	2.8874E+03	2.8634E+03	2.7927E+03	2.7465E+03	2.6787E+03	2.6126E+03
353	2.5481E+03	2.4852E+03	2.4238E+03	2.3640E+03	2.3056E+03	2.2487E+03	2.1932E+03	2.1390E+03
361	2.0862E+03	2.0347E+03	1.9844E+03	1.9354E+03	1.8877E+03	1.8411E+03	1.7956E+03	1.7513E+03
369	1.7080E+03	1.6659E+03	1.6247E+03	1.5846E+03	1.5455E+03	1.5073E+03	1.4701E+03	1.4338E+03
377	1.3984E+03	1.3639E+03	1.3302E+03	1.2974E+03	1.2653E+03	1.2341E+03	1.2036E+03	1.1739E+03
385	1.1449E+03	1.1167E+03	1.0891E+03	1.0622E+03	1.0360E+03	1.0104E+03	9.8545E+02	9.6112E+02
393	7.4852E+02	5.8295E+02	4.5400E+02	3.5357E+02	2.7536E+02	2.1445E+02	1.6702E+02	1.3007E+02
401	1.0130E+02	7.8893E+01	6.1442E+01	4.7851E+01	3.7267E+01	2.9023E+01	2.2603E+01	1.7603E+01
409	1.3710E+01	1.0677E+01	8.3153E+00	6.4759E+00	5.0435E+00	3.9279E+00	3.0590E+00	2.3824E+00
417	1.8554E+00	1.4450E+00	1.1254E+00	8.7642E-01	8.3368E-01	6.8256E-01	6.2799E-01	5.3158E-01
425	4.1746E-01							

ANL703

701*(3 Δu), 2 Δu , Δu

ANL1041

1041*(2 Δu)

ANL2082

2082*(Δu)

Appendix F. Gamma Library Group Structures

21 group structures (ANL21G)

	1	2	3	4	5	6	7	8
1	2.0000E+07	1.0000E+07	8.0000E+06	7.0000E+06	6.0000E+06	5.0000E+06	4.0000E+06	3.0000E+06
9	2.5000E+06	2.0000E+06	1.5000E+06	1.0000E+06	7.0000E+05	4.5000E+05	3.0000E+05	1.5000E+05
17	1.0000E+05	7.5000E+04	4.5000E+04	3.0000E+04	2.0000E+04			

94 group structures (ANL94G)

	1	2	3	4	5	6	7	8
1	2.0000E+07	1.4000E+07	1.2000E+07	1.1000E+07	1.0600E+07	1.0000E+07	9.5000E+06	9.0000E+06
9	8.5000E+06	8.0000E+06	7.7500E+06	7.5000E+06	7.2500E+06	7.0000E+06	6.7500E+06	6.5000E+06
17	6.2500E+06	6.0000E+06	5.7500E+06	5.5000E+06	5.4000E+06	5.2000E+06	5.0000E+06	4.7000E+06
25	4.5000E+06	4.4000E+06	4.2000E+06	4.0000E+06	3.9000E+06	3.8000E+06	3.6500E+06	3.5000E+06
33	3.3330E+06	3.1660E+06	3.0000E+06	2.8330E+06	2.6660E+06	2.5000E+06	2.3330E+06	2.1660E+06
41	2.0000E+06	1.8750E+06	1.7500E+06	1.6600E+06	1.6000E+06	1.5000E+06	1.4200E+06	1.3300E+06
49	1.2500E+06	1.2000E+06	1.1250E+06	1.0000E+06	9.0000E+05	8.6500E+05	8.2500E+05	8.0000E+05
57	7.5000E+05	7.0000E+05	6.7500E+05	6.5000E+05	6.2500E+05	6.0000E+05	5.7500E+05	5.5000E+05
65	5.2500E+05	5.0000E+05	4.5000E+05	4.2500E+05	4.0000E+05	3.7500E+05	3.5000E+05	3.2500E+05
73	3.0000E+05	2.6000E+05	2.2000E+05	1.9000E+05	1.6000E+05	1.5000E+05	1.4000E+05	1.2000E+05
81	1.0000E+05	9.0000E+04	8.0000E+04	7.5000E+04	6.5000E+04	6.0000E+04	5.5000E+04	4.5000E+04
89	4.0000E+04	3.5000E+04	3.0000E+04	2.0000E+04	1.5000E+04	1.0000E+04		

* Upper energy boundaries in eV

Appendix G. Program Structure of ETOE-2 with Major Subroutines

ADMIN : control the writing of the administrative MCC1 file.

DRIVF1, FILE1 : read and process the first record of the RIGEL tape which contains general information such as descriptive data, number of neutrons per fission, delayed neutron data, components of energy release due to fission, etc.

DRIVF2, FILE2 : control the calculation and writing of the resonance parameters and cross sections into the MC^2 library (files MCC3 and MCC4). As aforementioned, the MC^2 code can only handle Breit-Wigner (BW, both single and multilevel), Adler-Adler (AA) or Multi-pole (MP) resonance parameters. Therefore, ETOE-2 employs different options for the treatment of the resolved resonance energy range as a function of the parameter type provided on the ENDF/B tape. First, if the ENDF/B tape contains single level or multilevel BW parameters, or AA parameters, the code treats these parameters accordingly. Second, if the Reich-Moore resonance parameters are given, two options are available: 1) create corresponding multi-pole resonance parameters (subroutine POLLY); 2) create equivalent Adler-Adler parameters (subroutine POLLA). The subroutine RESKAT calculates point-wise resolved resonance scattering, capture and fission cross sections from resonance parameters.

DRIVF3, FILE3 : control the calculation (fitting) of the smooth non-resonant cross sections from the record 3 of the RIGEL tape and writes the results into the MCC5 file of the MC^2 library.

DRIVF4, FILE4 : read forth record from the RIGEL tape and is responsible for preparation and recording of the Legendre data on the MCC8 file.

DRIVF5, FILE5 : read RIGEL record 5 and writes file MCC7. It also calculates inelastic and (n,2n) data for the MCC6 file of the MCC^2-2 library.

FTABLE : control the calculation and writing of the function table file MCC2 of the MC^2 library.

GNUFT and GNUFT1 : help to fit evaluation data points with polynomials by the least square method. The DRIVF2 is the driver for the treatment of the resolved and unresolved resonance energy ranges (record 2 on the RIGEL tape).

ISCTRR, MATRNS, ISCTUR, UNSPIN, and WIPUTC : help to allocate resulting resonance parameters into arrays that will be written into the MCC libraries. They also perform reordering of the data.

POLLY : compute poles and residuals in the momentum space and generate Adler-Adler type parameters using the Reich-Moore parameters.

REICH : generate pointwise cross section from Reich-Moore data.

SRCHFA, FL6DA, and FL6MC2 : search and write data.

SRFILE : control the search through each of the six data records on the RIGEL tape by using the SRCHF1 to SRCHF6 subroutines. The treatment of the six data records on the RIGEL tape is governed by subroutines DRIVF1 to DRIVF5.

UNSCAT : calculate unresolved resonance scattering and capture cross sections from the unresolved resonance parameters.

XILE2 : reorder all (resolved and unresolved) resonance parameters from high energy to low energy.

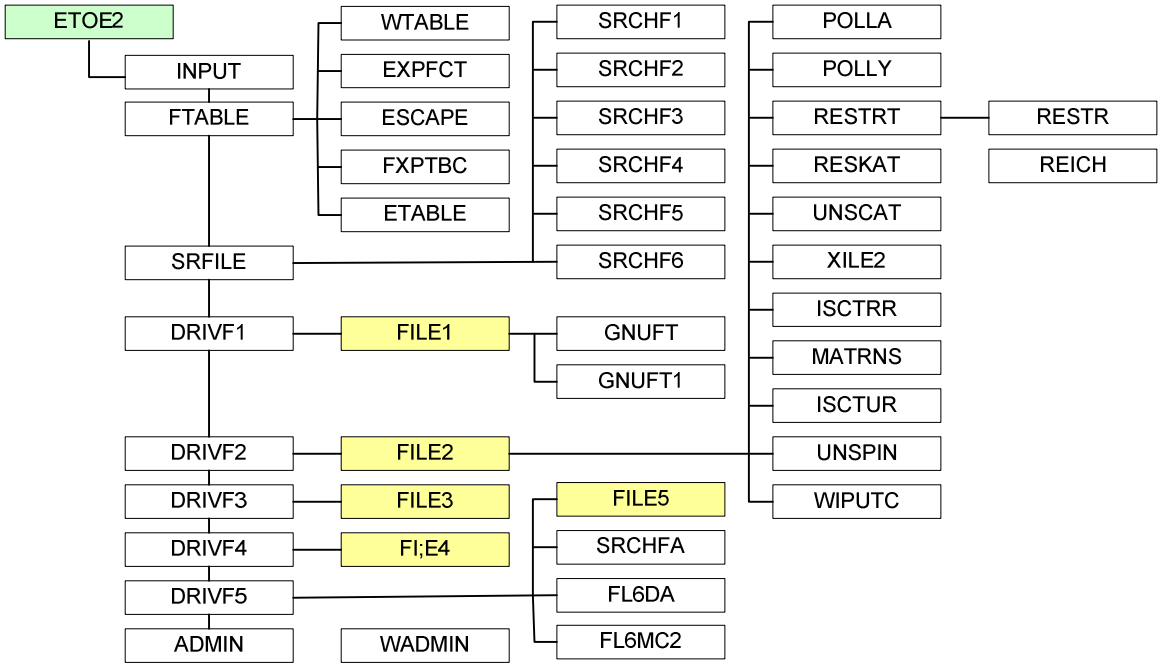


Figure F.1 Subroutine Flow Diagram of ETOE-2

Appendix H. Program Structure of MC^2-3 with Major Subroutines

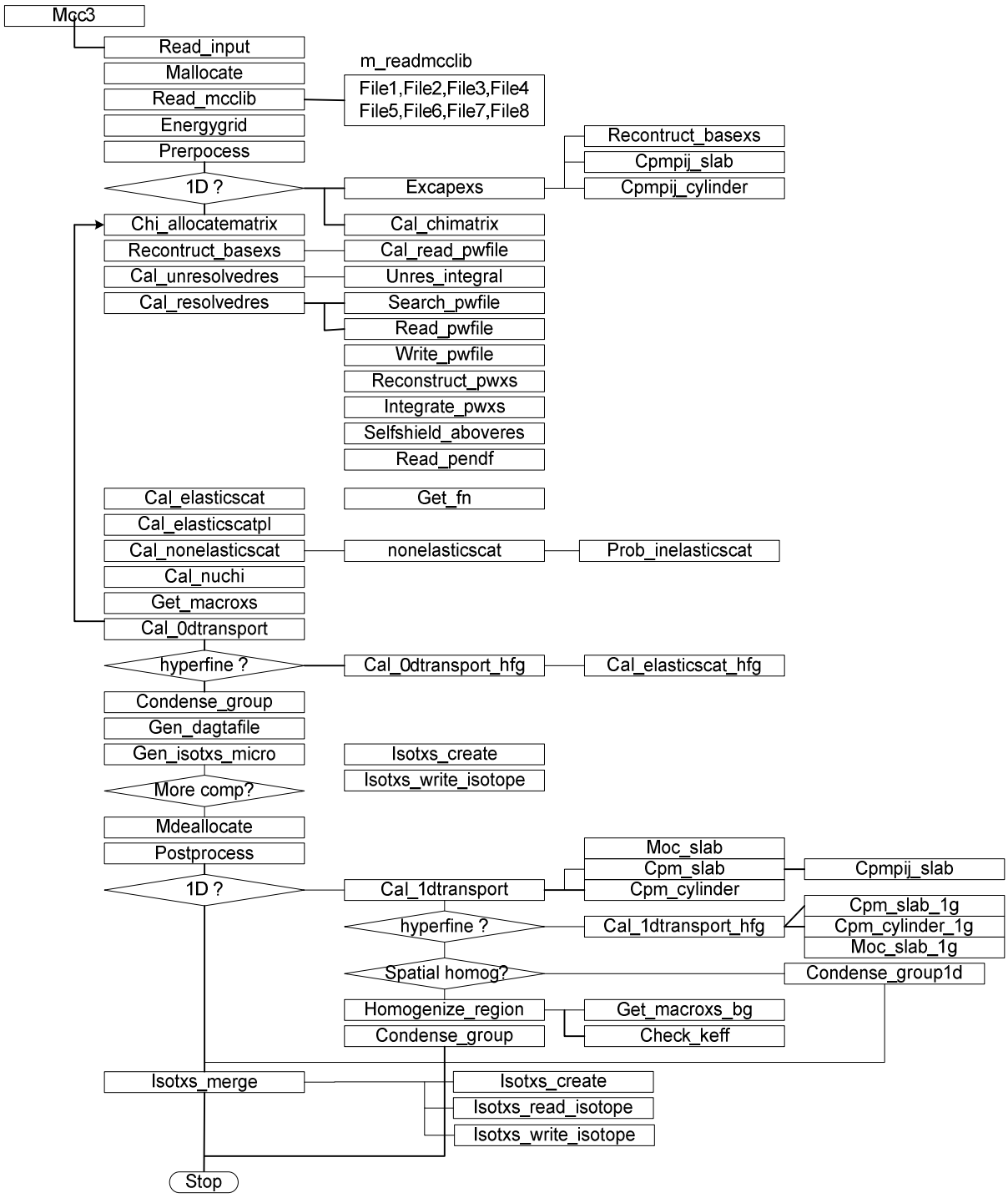


Figure G.1 Subroutine Flow Diagram of MC^2-3

Appendix I. Brief Software Requirement Specification (SRS) of MC^2-3

Type	Subject	Status
Cross section	Self-shielding resolved resonance: SLBW, MLBW, AA, multi-pole (Reich-Moore)	O
	Self-shielding unresolved resonance: LSSF = 0 (addition), 1 (shielding factor)	O
	Analytic Doppler broadening	O
	Self-shielding cross sections above resonance	O
	Elastic scattering: anisotropic source, P_N	O
	Inelastic scattering: anisotropic source, P_N	O Up to P_1
	(n,2n) reaction	O
	(n,3n) reaction	×
	Consistent P_1 multigroup transport calculation	O
	Group condensation: built-in group or user-specified structures	O
	Incident neutron energy dependent fission spectrum	O
	Scattering order up to P_9	O Practically up to P_5
	Thermal energy models	Δ
	Upscattering	Δ
Transport Solution	Consistent P_1 multigroup transport	O
	Collision probability method (CPM)	O Main solver
	Method of characteristics (MOC)	O Only for slab, not included in this version
	Ultrafine group calculation	O
	Hyperfine group calculation	O
	Buckling search	O
	External source: fixed source problem	O
Library data	ENDF/B-V	Δ Under verification
	ENDF/B-VI	× Not tested
	ENDF/B-VII.0	O
	Other libraries: JEF, JENDL	×
Geometry	Homogeneous mixture	O
	One-dimensional geometry: slab, cylinder	O
	Two-dimensional geometry: RZ, XY	O TWODANT is used
	Three-dimensional geometry	O
	Effective background cross section: equivalence theory	O
	Boundary condition: reflective, periodic	O
Input	Standard input	O
	PENDF file	O
	MC^2 library	O
	External fission spectrum: chi matrix	O
	External inelastic scattering matrix	O
	PENDF: tabulated with temperature	O Temp. interpolation
	Pointwise cross section	O
	Standard input: keyword, comment	O
Output	Standard output	O
	ISOTXS	O
	Broad group flux, moments	O

	Broad group microscopic cross sections	O
	User-defined group reaction rates	O
	Chi matrix	Δ Vector form only
	Pointwise cross section	O
	Macroscopic cross section	O
	Leakage	O
	Partially spatial homogenization	O
	Regionwise cross sections	O
	Region-homogenized cross sections	O
	Lumped fission product	O
	Consistent P_N correction	O
	TWODANT input generation	O
	Delayed neutron processing	O
	Gamma production yield	O
	KERMA (heating) factor	O
	Kinetic parameters	O
Energy	0.4 eV – 14.2 MeV up to 20 MeV	O
	Ultrafine group Δu = 1/120	O
	Hyperfine group: limited by the system memory	O $\Delta h = \Delta u$ / 200
Data Management	Pointwise cross section	O
	MC^2 library	O
	ISOTXS (data conversion, etc.)	O
	Isotope name change	O
Accuracy	Targeted accuracy	< 200 pcm
Computer Platform	Linux, Sun OS, PC Windows	O
	Macintosh	Δ (not tested)
	Memory required	Problem-dependent
	Computation time	Reasonable
Program	Fortran 95 or higher	O
	Test benchmark problems: inputs and outputs	O
	Code manual: user, methodology, programmer	O Brief programmer manual
	On-line user support: website	http://internal.ne.anl.gov/~clee/mcc3.html (available only from the Argonne intra-net)

* O : complete, Δ : incomplete, × : not implemented

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Case Avenue
Argonne, IL 60439

www.anl.gov