NUREG/CR-6694
ORNL/TM-2000/035

POLIDENT: A Module for Generating Continuous-Energy Cross Sections from ENDF Resonance Data

Prepared by
M. E. Dunn and N. M. Greene

Oak Ridge National Laboratory

U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555-0001

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at www.nrc.gov.NRC/ADAMS/index.html.
Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, of the Code of *Federal Regulations*, may also be purchased from one of these two sources:

1. The Superintendent of Documents
   U.S. Government Printing Office
   P.O. Box 37082
   Washington, DC 20402–9328
   www.access.gpo.gov/su_docs
   202–512–1800
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: Office of the Chief Information Officer,
Reproduction and Distribution
Services Section
U.S. Nuclear Regulatory Commission
Washington, DC 20555–0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address www.nrc.gov/NRC/NUREGS/indexnum.html are updated regularly and may differ from the last printed version.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG/XXXX) or agency contractors (NUREG/CR-XXXX), (2) proceedings of conferences (NUREG/CP-XXXX), (3) reports resulting from international agreements (NUREG/IA-XXXX), (4) brochures (NUREG/BR-XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG-0750).

DISCLAIMER: This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed in this publication, or represents that its use by such third party would not infringe privately owned rights.

NUREG/CR-6694
ORNL/TM-2000/035

POLIDENT: A Module for Generating Continuous-Energy Cross Sections from ENDF Resonance Data

Manuscript Completed: October 2000
Date Published: December 2000

Prepared by
M. E. Dunn and N. M. Greene, ORNL

Oak Ridge National Laboratory
Managed by UT-Battelle, LLC
Oak Ridge, TN 37831-6370

D. D. Ebert, NRC Project Manager

Prepared for
Division of Systems Analysis and Regulatory Effectiveness
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC Job Code W6479

ABSTRACT

POLIDENT (POint LIbraries of Data from ENDF/B Tapes) is an AMPX module that accesses the resonance parameters from File 2 of an ENDF/B library and constructs the continuous-energy cross sections in the resonance energy region. The cross sections in the resonance range are subsequently combined with the File 3 background data to construct the cross-section representation over the complete energy range. POLIDENT has the capability to process all resonance reactions that are identified in File 2 of the ENDF/B library. In addition, the code has the capability to process the single- and multi-level Breit-Wigner, Reich-Moore and Adler-Adler resonance formalisms that are identified in File 2. POLIDENT uses a robust energy-mesh-generation scheme that determines the minimum, maximum and points of inflection in the cross-section function in the resolved-resonance region. Furthermore, POLIDENT processes all continuous-energy cross-section reactions that are identified in File 3 of the ENDF/B library and outputs all reactions in an ENDF/B TAB1 format that can be accessed by other AMPX modules.

CONTENTS

Page

ABSTRACT . . . iii

LIST OF FIGURES . . . vii

LIST OF TABLES . . . ix

ACKNOWLEDGMENTS . . . xi

1 INTRODUCTION . . . 1
- 1.1 Background . . . 1
- 1.2 Purpose and Scope . . . 2
- 1.3 New Features . . . 2

2 THEORY . . . 3
- 2.1 Resonance Representations . . . 3
  - 2.1.1 Resolved-Resonance Region . . . 3
    - 2.1.1.1 Single- and Multi-Level Breit-Wigner . . . 4
    - 2.1.1.2 Reich-Moore . . . 8
    - 2.1.1.3 Adler-Adler . . . 12
  - 2.1.2 Unresolved-Resonance Region . . . 14
- 2.2 Energy-Mesh Generation . . . 25
  - 2.2.1 Resolved-Resonance Region . . . 25
  - 2.2.2 Unresolved-Resonance Region . . . 35
- 2.3 Cross-Section-Data Construction . . . 35
  - 2.3.1 Combining Functions without Discontinuities . . . 37
  - 2.3.2 Combining Functions with Discontinuities . . . 44

3 LOGICAL PROGRAM FLOW . . . 49
- 3.1 Program Initiation . . . 49
- 3.2 Overall Program Flow . . . 51
- 3.3 Resonance Processing . . . 54
  - 3.3.1 Mesh Generation . . . 56
  - 3.3.2 Resolved Region . . . 58
  - 3.3.3 Unresolved Region . . . 60
- 3.4 Post-Resonance Processing . . . 62

4 INPUT DATA GUIDE . . . 67
- 4.1 FIDO Input Structure . . . 67
- 4.2 Logical Unit Parameters . . . 69

5 DESCRIPTION OF OUTPUT . . . 71
- 5.1 Header Page . . . 71
- 5.2 Program-Verification Information . . . 72
- 5.3 Problem-Verification Information . . . 73
- 5.4 File 1 Processing . . . 74
- 5.5 File 2 Processing . . . 75

5.5.1 Resolved-Resonance Data 75
5.5.2 Unresolved-Resonance Data 77
5.6 Post-Resonance Processing 78
5.7 Termination of Output File 82

6 MESSAGES 83
6.1 Warning Messages 83
6.2 Error Messages 85

7 REFERENCES 89

APPENDIX A — ALPHABETICAL INDEX OF SUBROUTINES 91

APPENDIX B — SAMPLE PROBLEMS 101

APPENDIX C — NUMERICAL EXPRESSIONS FOR REICH-MOORE
RESOLVED-RESONANCE EQUATIONS 105

APPENDIX D — FIDO INPUT 115
D.1 INTRODUCTION 117
D.2 FIXED-FIELD INPUT 117
D.3 FREE-FIELD INPUT 121
D.4 USER-FIELD INPUT 122
D.5 CHARACTER INPUT 122

LIST OF FIGURES

Figure **Page**

2.1 Mesh generation for resolved-resonance region 26

2.2 Example resonance with critical points 27

2.3 Slope calculation using critical points 28

2.4 Halving-iteration scheme 30

2.5 Fine-grid structure with critical points 31

2.6 Transfer critical points to auxiliary grid 33

2.7 Collapsing fine energy grid to auxiliary grid 34

2.8 Two arbitrary energy-dependent functions to be combined 38

2.9 Arbitrary example for combining two functions that overlap at a single energy point 39

2.10 Arbitrary example for combining two functions with overlapping energy regions 40

2.11 Detail for processing the interpolation region during the combining of two arbitrary functions with overlapping energy regions 43

2.12 Schematic diagrams of discontinuities for cases 1 through 4 of Table 2.6 47

2.13 Schematic diagrams of discontinuities for cases 5 through 7 of Table 2.6 48

3.1 Flowchart for program initiation 49

3.2 Flowchart for overall program flow 51

3.3 Flowchart for resonance processing 54

3.4 Flowchart for energy-mesh generation 56

3.5 Flowchart for processing single- and multi-level Breit-Wigner resonance data 58

3.6 Flowchart for processing Reich-Moore resonance data 59

3.7 Flowchart for processing unresolved-resonance data 60

3.8 Flowchart for processing cross-section data from resonance region 62

3.9 Flowchart for combining File 3 data with cross sections from the resonance region 64

LIST OF FIGURES (continued)

Figure **Page**

5.1 Example header-page output 71

5.2 Example program-verification output 72

5.3 Example problem-verification-information output 73

5.4 Example File 1 processing output 74

5.5 Example resolved-resonance-processing output 76

5.6 Example unresolved-resonance-processing output 77

5.7 Example output for processing data blocks from resonance region 79

5.8 Example output for combining resonance-region cross sections with background data 81

LIST OF TABLES

Table **Page**

2.1 Equations for penetrability, level shift and phase shift factors for different angular momentum states 5

2.2 Categorization of unresolved resonance parameter formats 35

2.3 ENDF/B interpolation laws 36

2.4 Example ENDF/B function 36

2.5 Combining operations available in AMPX 41

2.6 Possible cases for processing discontinuities in energy-dependent functions 46

3.1 Summary of subroutines called by SPARKY 52

4.1 File parameters for logical units 69

A.1 Index of subroutines 93

C.1. Definitions of the elements of the *K* matrix 110

D.1 General example of FIDO input 119

ACKNOWLEDGMENTS

Sincere appreciation is expressed to D. D. Ebert, C. W. Nilsen and D. E. Carlson of the Nuclear Regulatory Commission for their support of this effort. Special appreciation is also expressed to L. C. Leal for performing the initial work on the energy-mesh-generation scheme for the resolved-resonance region, as well as providing the initial FORTRAN coding for the energy-meshing scheme. His early efforts and guidance established the framework for the subsequent development of the meshing scheme for the current version of POLIDENT.

1 INTRODUCTION

1.1 Background

The evaluation of nuclear systems typically requires a numerical solution of the Boltzmann transport equation. Consequently, a numerical-transport calculation requires the use of energy-dependent cross-section data. For neutron-induced reactions, the Evaluated Nuclear Data File (ENDF) system has procedures and formats that are used to describe the complex structure of cross-section data for specific materials of interest.[1] The ENDF system is divided into forty separate files that are used to describe the underlying physics of the nuclear data for a specific material evaluation. Prior to describing the various ENDF files that are pertinent for this manual, an overview of the energy-dependent cross-section information is provided for neutron-induced reactions.

The entire cross-section energy range for neutron-induced reactions can typically be divided into four separate regions. In particular, the different energy ranges include the low-energy region (LER), followed by the resolved-resonance region (RRR), unresolved-resonance region (URR) and the high-energy region (HER). The boundaries of the different energy ranges are extremely material dependent. For instance, the resonances of some materials may extend into the low-eV range, while the resolved resonances of other materials are not present at low energies.

In the LER, the cross sections can be represented as smooth functions of energy. Moreover, the data in the LER must be characterized such that the Doppler effects are negligible and the cross sections undergo negligible broadening. For some light elements, the cross-section data can be represented as a smooth function of energy throughout the entire energy range. Typically, the LER is used for the energy range below the lowest resolved resonances.

The RRR is an energy region that is characterized by well-defined resonances. In other words, the experimental resolution is sufficient to permit the determination of the resolved-resonance parameters by area or shape analysis. In contrast, the URR is an energy region in which the experimental resolution is inadequate for determining the resonance parameters of individual resonances. As a result, the unresolved-resonance parameters are averages of resolved-resonance parameters over energy. Moreover, the unresolved-resonance parameters are constant throughout a specified energy interval, but the values of the parameters vary as a function of the different energy intervals. At energies above the URR in the HER, the cross-section resonances overlap such that the data can be represented as a smooth function of energy.

In every ENDF evaluation, File 1 provides an overview of the available cross-section information in the section defined by MT = 451. Additional sections that describe the number of neutrons produced from fission and corresponding energy release are also provided in File 1 for fissionable isotopes. File 2, which must be present for all neutron-induced reactions, provides the data for the resolved- and unresolved-resonance parameters. Subsequently, File 3 provides cross-section data as a function of energy with a corresponding ENDF interpolation law that is used to obtain cross-section values between the grid points. In the resonance range, the data in File 3 are "background" cross-section values that compensate for inadequacies in the resonance formula representation, missed resonances, competing cross sections or the effects of resonances outside the energy range. To construct the cross-section representation in the RRR, the contribution to the cross-section values from the resonances is computed from the resonance parameters in File 2 and summed with the background values from File 3. In the URR, the cross-section values are obtained by either summing the contributions from Files 2 and 3 as in the RRR or by multiplying the File 3 cross-section values by self-shielding factors that are computed from the File 2 parameters.

1.2 Purpose and Scope

POLIDENT was developed from the earlier NPTXS module and released with AMPX-77.[2] Since the initial development of POLIDENT, the ENDF/B cross-section formats have been enhanced over the years to the current release that is denoted as Version VI.[1] Unfortunately, the previous version of POLIDENT could only process evaluations through ENDF/B-V. This manual documents the latest version of POLIDENT, which has the capability to process all ENDF formats through Version VI.

With regard to processing capabilities, POLIDENT accesses the resonance parameters in File 2 of an ENDF/B library and reconstructs the resonance cross sections as a function of energy based on the appropriate formalism (e.g., single-level Breit-Wigner, Reich-Moore, etc.). In the resolved range, POLIDENT combines the appropriate background cross sections from File 3 with the File 2 contribution. POLIDENT also constructs the cross section as a function of energy in the unresolved range. The resulting output consists of continuous-energy cross sections for all the neutron-induced reactions that are defined in File 3. The output cross sections are stored in an ENDF/B TAB1 format that can be read and processed by other modules in the AMPX code system.

1.3 New Features

One of the latest features includes a more robust energy-mesh-generation scheme that is based largely on the numerical determination of a very-fine energy mesh that is collapsed to a desired auxiliary grid. The auxiliary grid is determined such that the fine grid can be reproduced within a user-specified tolerance. As a result, POLIDENT is able to construct an accurate and efficient (i.e., number of points) energy grid for a specific material in the resolved-resonance region.

2 THEORY

The primary objective of POLIDENT is to reconstruct the resonance region cross-section representations from resonance parameters that are specified by the evaluator in File 2 of the ENDF system. In the following sections, the different resonance formalisms that are available in the ENDF files are discussed. As part of the resonance-reconstruction process, POLIDENT must generate a suitable energy mesh for the resonance region. As a result, a description of the mesh-generation scheme is also provided in the following discussion.

2.1 Resonance Representations

Resonance theory is concerned with the mathematical description of neutron-nucleus interactions. If a neutron penetrates and is absorbed by a target nucleus, a new compound nucleus is formed, and the compound nucleus subsequently decays by emitting an energetic particle. Many neutron-nucleus reactions such as radiative capture, fission, etc., are characterized by a compound-nucleus formation. In terms of collision kinematics, the energy available for a compound-nucleus interaction can be denoted as E_c in the center-of-mass system and is a function of the neutron and nucleus masses, as well as the neutron kinetic energy in the laboratory system. When the compound nucleus is formed, the excited nucleus is characterized by an energy level that is higher than E_c because of the binding energy of the interacting neutron, E_b. If the summation of E_c and E_b is on the order of an energy level of the compound nucleus, the probability for the formation of a compound nucleus increases significantly relative to other energies. Since neutron cross sections are a measure of the probability of neutron-nucleus interactions, the increased probability for the formation of a compound nucleus at energy levels of the compound nucleus are defined as "resonances" in the cross-section data. Moreover, the corresponding energy levels for the formation of a compound nucleus are defined as "resonance energies" in the cross-section data. As noted previously, there are a variety of neutron-nucleus interactions that lead to the formation of a compound nucleus, and these reactions are typically defined as "resonance reactions."

The resonance representations in the subsequent sections are essentially interaction models that describe the interaction between a neutron and a target nucleus. Because of the complexity of the internal structure of a nucleus, the resonance representations do not model the nuclear effects within the nucleus.[3] However, the resonance parameters in File 2 are strongly correlated to the internal properties of a nucleus and are obtained by the evaluation of measured cross-section data. As noted in Section 1.1, the resonance region is divided into the resolved- and unresolved-resonance region (RRR) and (URR), respectively. Both the RRR and URR are discussed in the following sections.

2.1.1 Resolved-Resonance Region

Within the ENDF file system, six different cross-section representations are permitted in the resolved-resonance region:

1. Single-level Breit-Wigner (SLBW)
2. Multi-level Breit-Wigner (MLBW)
3. Reich-Moore (RM)
4. Adler-Adler (AA)
5. General R-matrix (GRM)
6. Hybrid R-function (HRF)

Currently, all ENDF/B-VI evaluations use the first four resonance representations. As a result, POLIDENT does not process the General R-matrix or Hybrid R-function representations. The following discussion is limited to the single- and multi-level Breit-Wigner, Reich-Moore and Adler-Adler formalisms.

2.1.1.1 Single- and Multi-Level Breit-Wigner

In File 2 of the ENDF data, the SLBW and MLBW data consist of six parameters for each resonance that include the resonance energy, E_r, resonance spin, J, and four channel widths. The channel widths that are provided in the Breit-Wigner data include the total width, Γ_r, neutron width, Γ_n, radiative-capture width, Γ_γ, and fission width, Γ_f. Using the resolved-resonance parameters from File 2, the SLBW representation for the elastic-scattering cross section as a function of energy is obtained from the following equation:[1]

$$\sigma_s(E) = \sum_{\ell=0}^{NLS-1} \frac{4\pi}{k^2}(2\ell+1)\sin^2\phi_\ell + \frac{\pi}{k^2}\sum_J g_J \sum_{r=1}^{NR_J} \frac{\Gamma_{nr}^2 - 2\Gamma_{nr}\Gamma_r \sin^2\phi_\ell + 2(E-E'_r)\Gamma_{nr}\sin(2\phi_\ell)}{(E-E'_r)^2 + \frac{\Gamma_r^2}{4}}, \quad (2.1)$$

where

ℓ	=	relative neutron-nucleus angular momentum (provided in File 2),
NLS	=	total number of ℓ-states (provided in File 2),
k	=	neutron-wave number at energy E,
ϕ_ℓ	=	angular momentum hard-sphere-phase shift at energy E,
J	=	spin of the resonance (provided in File 2),
g_J	=	statistical spin factor,
r	=	resonance index,
NR_J	=	number of resonances with spin J,
Γ_{nr}	=	neutron line width at energy E,
Γ_r	=	total resonance width at energy E,
E'_r	=	shifted resonance energy.

In Equation (2.1), various quantities are introduced that need further clarification. Depending on the relative neutron-nucleus angular momentum (i.e., ℓ-state), the hard-sphere-phase shift, ϕ_ℓ, can have different values, and the equations for the phase shift factors for $\ell = 0$ through $\ell = 4$ resonances are provided in Table 2.1.

Table 2.1 Equations for penetrability, level-shift and phase-shift factors for different angular momentum states[a]

ℓ	Penetrability		Level shift	Phase shift
	P_ℓ	q_ℓ	S_ℓ	ϕ_ℓ
0	ρ	$q_0 = \frac{\rho\sqrt{E + Q_l}}{\sqrt{E}}$	0	ρ
1	$\frac{\rho^3}{1 + \rho^2}$	$\frac{q_0^3}{1 + q_0^2}$	$\frac{-1}{1 + \rho^2}$	$\rho - \tan^{-1}\rho$
2	$\frac{\rho^5}{(9 + 3\rho^2 + \rho^4)}$	$\frac{q_0^5}{(9 + 3q_0^2 + q_0^4)}$	$\frac{-(18 + 3\rho^2)}{(9 + 3\rho^2 + \rho^4)}$	$\rho - \tan^{-1}\frac{3\rho}{3 - \rho^2}$
3	$\frac{\rho^7}{(225 + 45\rho^2 + 6\rho^4 + \rho^6)}$	$\frac{q_0^7}{(225 + 45q_0^2 + 6q_0^4 + q_0^6)}$	$\frac{-(675 + 90\rho^2 + 6\rho^4)}{(225 + 45\rho^2 + 6\rho^4 + \rho^6)}$	$\rho - \tan^{-1}\frac{\rho(15 - \rho^2)}{15 - 6\rho^2}$
4	$\frac{\rho^9}{(11025 + 1575\rho^2 + 135\rho^4 + 10\rho^6 + \rho^8}$	$\frac{q_0^9}{(11025 + 1575q_0^2 + 135q_0^4 + 10q_0^6 + q_0^8)}$	$\frac{-(44100 + 4725\rho^2 + 270\rho^4 + 10\rho^6)}{(11025 + 1575\rho^2 + 135\rho^4 + 10\rho^6 + \rho^8)}$	$\rho - \tan^{-1}\frac{\rho(105 - 10\rho^2)}{105 - 45\rho^2 + \rho^4}$

[a]Note: $\rho = ka_c$, where a_c is the channel radius and k is the neutron-wave number or momentum in the center-of-mass system. Q_ℓ = reaction energy or Q-value for compound nucleus formation.

The statistical spin factor, g_J, for neutrons is given by the following equation:

$$g_J = \frac{(2J + 1)}{2(2SPI + 1)} , \tag{2.2}$$

where

SPI	=	spin of the target nucleus (provided in File 2).

The neutron-wave number, k, in Equation (2.1) is implicitly denoted as a function of energy and is calculated with the following relation:

$$k = 2.196771 \; x \; 10^{-3} \frac{AWRI}{AWRI + 1.0} \sqrt{E} , \tag{2.3}$$

where

E	=	energy in the laboratory system,
$AWRI$	=	ratio of the target isotope mass to the neutron mass.

In Equation (2.1), the neutron-line width, Γ_{nr}, and the total width, Γ_r, are also a function of energy and are calculated with Equations (2.4) and (2.5), respectively:

$$\Gamma_{nr}(E) = \frac{P_\ell(E) \; \Gamma_{nr}(|E_r|)}{P_\ell(|E_r|)} , \tag{2.4}$$

and

$$\Gamma_r(E) = \Gamma_{nr}(E) + \Gamma_\gamma + \Gamma_f + \Gamma_x(E) , \tag{2.5}$$

where

E_r	=	resonance energy (provided in File 2),
P_ℓ	=	penetrability or penetration shift factor,
$\Gamma_{nr}(\|E_r\|)$	=	neutron-line width at the resonance energy (provided in File 2),
$\Gamma_{nr}(E)$	=	neutron-line width at energy E,
Γ_γ	=	gamma width at the resonance energy (provided in File 2),
Γ_f	=	fission width at the resonance energy (provided in File 2),
$\Gamma_x(E)$	=	competitive width at energy E.

In order to calculate the total width at energy E in Equation (2.5), the competitive width at energy E can be calculated as follows:

$$\Gamma_x(E) = \frac{q_\ell(E)\ \Gamma_x(|E_r|)}{q_\ell(|E_r|)} , \tag{2.6}$$

where

q_ℓ = penetrability factor that is dependent on the reaction energy for compound-nucleus formation (i.e., Q-value). The penetrability factors are presented in Table 2.1 as a function of angular momentum.

Note that $\Gamma_r(|E_r|)$ is the total width at the resonance energy and is provided in File 2. The competitive-resonance width, $\Gamma_x(|E_r|)$, is not given explicitly for the SLBW and MLBW representations in File 2; however, $\Gamma_x(|E_r|)$ can be calculated as follows:

$$\Gamma_x(|E_r|) = \Gamma_r(|E_r|) - \Gamma_{nr}(|E_r|) - \Gamma_\gamma - \Gamma_f . \tag{2.7}$$

Associated with the resonance energy, E_r, is a shift in the resonance energy, E'_r, that is calculated as follows:

$$E'_r = E_r + \frac{S_\ell(|E_r|) - S_\ell(E_r)}{2P_\ell(|E_r|)}\ \Gamma_{nr}(|E_r|) , \tag{2.8}$$

where

S_ℓ = level-shift factor. The shift factors are presented in Table 2.1 as a function of angular momentum.

Radiative-capture reactions are characterized by a compound-nucleus formation that occurs following the absorption of an incident neutron. Subsequently, the excited nucleus decays by emitting high-energy gamma particles. Using the resolved-resonance parameters from File 2, the SLBW representation for the capture cross section as a function of energy is obtained with the following equation:[1]

$$\sigma_\gamma(E) = \sum_{\ell=0}^{NLS-1} \frac{\pi}{k^2} \sum_J g_J \sum_{r=1}^{NR_J} \frac{\Gamma_{nr}\ \Gamma_\gamma}{(E - E'_r)^2 + \frac{1}{4}\Gamma_r^2} . \tag{2.9}$$

Regarding fission, a compound nucleus is formed when a neutron is absorbed by the nucleus. Following absorption, the compound nucleus decays by fissioning into two lighter nuclei and a certain number of neutrons. The SLBW representation for the fission cross section as a function of energy is obtained with an equation that is similar to the equation for radiative capture:[1]

$$\sigma_f(E) = \sum_{\ell=0}^{NLS-1} \frac{\pi}{k^2} \sum_J g_J \sum_{r=1}^{NR_J} \frac{\Gamma_{nr}\,\Gamma_f}{(E - E'_r)^2 + \frac{1}{4}\Gamma_r^2} \,. \tag{2.10}$$

As observed for elastic scattering, the neutron-line width, Γ_{nr}, and the total width, Γ_r in Equations (2.9) and (2.10) are evaluated at energy E, and the gamma and fission widths are evaluated at the resonance energy, E_r.

The equations for the MLBW representation are the same as the SLBW, except for the elastic-scattering cross section. In particular, there is a resonance-resonance interference term that is included in the equation for elastic scattering:

$$\sigma_s(E) = \sum_{\ell=0}^{NLS-1} \frac{4\pi}{k^2}(2\ell+1)\sin^2\phi_\ell + \frac{\pi}{k^2}\sum_J g_J \left\{ \sum_{r=1}^{NR_J} \frac{(E - E'_r)\Gamma_{nr}\sin(2\phi_\ell) - \Gamma_{nr}\Gamma_r \sin^2\phi_\ell}{2\left((E - E'_r)^2 + \frac{1}{4}\Gamma_r^2\right)} \right.$$
$$\left. + \left[\sum_{r=1}^{NR_J} \frac{\Gamma_r\,\Gamma_{nr}}{2\left((E - E'_r)^2 + \frac{1}{4}\Gamma_r^2\right)} \right]^2 + \left[\sum_{r=1}^{NR_J} \frac{\Gamma_{nr}\,(E - E'_r)}{(E - E'_r)^2 + \frac{1}{4}\Gamma_r^2} \right]^2 \right\} . \tag{2.11}$$

2.1.1.2 Reich-Moore

The following discussion is provided to highlight the Reich-Moore (R-M) approximation to general R-matrix theory for describing neutron-nucleus interactions. A derivation of the R-M formulae is not provided in the following discussion; rather, the defining equations for the approximation are presented. In the R-M approximation, the off-diagonal contribution to the R-matrix from the photon channels is neglected.[3] The resulting formalism is especially suited for representing isotopes that are characterized by a limited number of channels and many resonances. Since the R-M formulation has two fission channels, the R-M representation is well suited for fissile isotopes because the fission process occurs through a small number of channels.[3] For neutron-induced reactions, the general expression for a cross section as a function of energy with an exit channel c is given by the following equation:[1]

$$\sigma_{nc}(E) = \frac{\pi}{k^2}\sum_J g_J \,|\, \delta_{nc} - U^J_{nc} \,|^2 \,, \tag{2.12}$$

where

δ_{nc} = Kronecker delta function,
U^J_{nc} = collision or scattering matrix for neutron-induced reactions with exit channel c.

Note that the subscripts in Equation (2.12) correspond to entrance and exit channels, and the angular momentum subscripts, ℓ, are implied. In the subsequent equations, the angular momentum subscripts are also implied.
In Equation (2.12), the exit channel c can be scattering (n), capture (γ) or either of the two partial-fission widths (f_1 or f_2). The collision matrix is a complex matrix that is defined as follows:

$$U_{nc}^{J} = e^{-i(\phi_n + \phi_c)} \left[2 \left((I - K) \right)_{nc}^{-1} - \delta_{nc} \right] , \tag{2.13}$$

where

$$(I - K)_{nc} = \delta_{nc} - \frac{i}{2} \sum_{r} \frac{\Gamma_{nr}^{1/2} \, \Gamma_{cr}^{1/2}}{(E_r - E) - i\frac{\Gamma_{\gamma r}}{2}} . \tag{2.14}$$

The total and elastic-scattering cross sections can be expressed as a function of energy in terms of the collision matrix:

$$\sigma_t(E) = \frac{2\pi}{k^2} \sum_{J} g_J \, Re(1 - U_{nn}^{J}) , \tag{2.15}$$

and

$$\sigma_{n,n}(E) = \frac{\pi}{k^2} \sum_{J} g_J \, | 1 - U_{nn}^{J} |^2 , \tag{2.16}$$

where

Re = the real component of the complex quality.

Likewise, the fission cross section can be calculated in terms of the collision matrix and corresponding exiting fission channels:

$$\sigma_f(E) = \frac{\pi}{k^2} \sum_{J} g_J \sum_{c} |U_{nc}^{J}|^2 . \tag{2.17}$$

The absorption cross section can be calculated by subtracting the elastic-scattering cross section from the total at a specific energy. Subsequently, the capture cross section is obtained by subtracting the fission from the absorption cross section. Experience has shown that calculating the absorption cross section by subtracting cross-section quantities at low energies leads to a computation of the form 1- (1 - ϵ), where ϵ is a small number. Taking the difference of large numbers that are nearly equal can lead to numerical instabilities in the calculation. In particular, the instability occurs if the matrix K_{nc} is small with respect to 1. To circumvent the numerical problem, an expression is derived for the absorption cross section. In the subsequent derivation, the matrix ρ can be defined as follows:

$$\rho_{nc} = \left(I - (I - K)^{-1} \right)_{nc} = - \left((I - K)^{-1} K \right)_{nc} . \tag{2.18}$$

Using Equation (2.18), the collision matrix can be expressed in the following format:

$$U_{nc}^{J} = e^{-i(\phi_n + \phi_c)} \left[2 \, (I - \rho) - I \right]_{nc} = e^{-i(\phi_n + \phi_c)} \left[I - 2\rho \right]_{nc} . \tag{2.19}$$

Using Equation (2.16) and expanding the squared quantity, the equation for the elastic-scattering cross section can be expressed in the following form:

$$\sigma_{1,1}(E) = \sigma_{el}(E) = \frac{\pi}{k^2} \sum_{J} g_J \, [1 - 2ReU_{11} + |U_{11}|^2] . \tag{2.20}$$

Note that the neutron subscripts are replaced with the value of 1 corresponding to the neutron channel in the R-M approximation. In the subsequent equations, the first- and second-fission channels are denoted with a 2 and 3, respectively. Using the equation for the collision matrix, the elastic-scattering cross section can be expressed as follows:

$$\sigma_{el}(E) = \frac{\pi}{k^2} \sum_J g_J \left\{ 1 - 2Re\left[e^{-2i\phi_1}(1 - 2\rho_{11})\right] + |1 - 2\rho_{11}|^2 \right\}, \tag{2.21}$$

$$\sigma_{el}(E) = \frac{\pi}{k^2} \sum_J g_J \left\{ 1 - 2\cos 2\phi_1 + 4\, Re[\, e^{-i2\phi_1}\rho_{11}\,] + 1 - 4\, Re\rho_{11} + 4\, |\rho_{11}|^2 \right\}, \tag{2.22}$$

$$\sigma_{el}(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ \sin^2\phi_1(1 - 2\, Re\rho_{11}) + \sin(2\phi_1)\, Im\rho_{11} + |\rho_{11}|^2 \right\}, \tag{2.23}$$

where

Im = the imaginary component of the complex quantity.

In Equations (2.21) through (2.23), the phase-shift subscripts correspond to channel number while the angular-momentum subscripts are implied.

Following a similar procedure, the total cross section as a function of energy is obtained by expanding Equation (2.15) in terms of the ρ matrix:

$$\sigma_t(E) = \frac{2\pi}{k^2} \sum_J g_J \left\{ 1 - Re\left[\, e^{-2i\phi_1}(1 - 2\rho_{11})\,\right] \right\}, \tag{2.24}$$

$$\sigma_t(E) = \frac{2\pi}{k^2} \sum_J g_J \left\{ 1 - \cos 2\phi_1 + 2\, Re[\, e^{-i2\phi_1}\rho_{11}\,] \right\}, \tag{2.25}$$

$$\sigma_t(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ \sin^2\phi_1 + (1 - 2\sin^2\phi_1)\, Re\rho_{11} + \sin(2\phi_1)\, Im\rho_{11} \right\}. \tag{2.26}$$

By subtracting Equation (2.23) from Equation (2.26), an expression is obtained for the absorption cross section as a function of energy:

$$\sigma_a(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ Re\rho_{11} - |\rho_{11}|^2 \right\}. \tag{2.27}$$

Equation (2.27) can be used to calculate the absorption cross section and avoid the numerical instability associated with subtracting large numbers with small differences (i.e., outside precision limits). Although Equation (2.27) does avoid the numerical instability that has been encountered with the subtraction of large cross-section quantities that

are nearly equal, an additional numerical problem can occur if the real component of the resonance quantity ρ_{11} is approximately equal to $|\rho_{11}|^2$. Experience has revealed that a numerical instability can occur for isotopes that have a very small absorption or capture cross section (e.g., ~ 10^{-5}), coupled with a detailed resonance structure in the absorption or capture cross section. Since ρ_{11} is a complex quantity, ρ_{11} has the form a + *i*b. Consequently, the quantity $|\rho_{11}|^2$ is equal to $a^2 + b^2$. If a and b are less than 1 and the quantity b is approximately equal to $a^{1/2}$, the difference calculation in Equation (2.27) involves the subtraction of two small numbers that are approximately equal. If the calculation in Equation (2.27) is performed using the double-precision complex functions that are available in FORTRAN on a very fine energy grid, experience has revealed that numerical instabilities occur for isotopes/nuclides with very small absorption cross-section values. Therefore, an alternative approach must be used to calculate the absorption cross section. In Equation (2.27), the quantity $Re\rho_{11} - |\rho_{11}|^2$ can be expanded, and a numerical expression can be developed that avoids the subtraction of two numbers that are approximately equal. An expression that avoids the numerical instability problem is provided in Appendix C for calculating the quantity $Re\rho_{11} - |\rho_{11}|^2$.

Before calculating the elastic-scattering cross section, Equation (2.23) must be revisited. The ENDF formats only provide the energy, E_r, and spin of the resonance, *J*; however, in the measurement process, a neutron with intrinsic spin *i* (i.e., *i* = 1/2) strikes a target with spin *SPI*. For the neutron-nucleus interaction, there is an associated channel spin *j* that is the vector summation of the neutron and target spins (i.e., $\vec{j} = \vec{i} + \overrightarrow{SPI}$).[4] The relative-orbital-angular momentum of the channel is denoted as ℓ. The total spin, *J*, associated with the interaction is obtained by a vector summation of the channel spin and angular momentum (i.e., $\vec{J} = \vec{j} + \vec{\ell}$). As a result, the total spin associated with the compound nucleus can have values between $J = |\ell - j|$ and $J = \ell + j$. In the calculation of the elastic-scattering cross section, three quantum numbers (E_r, j, J) must be considered to account for all of the potential scattering. For a non-zero spin target nuclide, the present form of Equation (2.23) only accounts for one of the two possible channel-spin terms in the potential scattering. For example, consider a spin-1 isotope (i.e., *SPI* = 1) with relative-angular momentum of 2 (i.e., $\ell = 2$). Six combinations of channel spin and total spin will contribute to the potential scattering:

j	*J*
1/2	3/2
1/2	5/2
3/2	1/2
3/2	3/2
3/2	5/2
3/2	7/2

If the present form of Equation (2.23) is used to calculate the elastic-scattering cross section, only four of the combinations of channel spin and total spin will be included in the contribution to the potential scattering. In particular, the calculation will only include *J* values of 3/2 and 5/2 for channel spin 1/2 and *J* values of 1/2 and 7/2 for channel spin 3/2. As a result, *J* values of 3/2 and 5/2 would be omitted for a channel spin of 3/2. To circumvent this problem, Equation (2.23) can be recast in a different form that includes an explicit summation over the channel spin, *j*, and total spin *J* for the potential-scattering contribution:

$$\sigma_{el}(E) = \frac{4\pi}{k^2} \left\{ \sum_J g_J \left\{ -2\sin^2\phi_1 \; Re\rho_{11} + \sin(2\phi_1) \; Im\rho_{11} + |\rho_{11}|^2 \right\} + \sum_{j=\left|SPI-\frac{1}{2}\right|}^{SPI+\frac{1}{2}} \sum_{J=|\ell-j|}^{\ell+j} g_J \sin^2\phi_1 \right\} . \quad (2.28)$$

Equations (2.27) and (2.28) are used to calculate the absorption and elastic-scattering cross sections, respectively. As noted for the absorption cross section, numerical instabilities can occur in the elastic-scattering cross-section

calculation if the FORTRAN single- or double-precision complex functions are used to calculate small elastic-scattering cross sections (e.g., ~10^{-5}) on a very fine energy grid. Appendix C provides a discussion for the calculation of $\sigma_{el}(E)$ in POLIDENT using Equation (2.28).

Regarding the fission cross section, the phase shift for channel c, ϕ_c, is zero, and the off-diagonal elements of I_{nc} are also zero. As a result, the collision matrix can be expressed as follows:

$$U_{nc}^{J} = e^{-i\phi_n}\, 2\rho_{nc} \ . \tag{2.29}$$

In Equation (2.17), the fission cross section is obtained by taking the square of the absolute value of the collision matrix and summing over the possible exit channels. Since the collision matrix is a complex quantity, the quantity $|U^J_{nc}|^2 = U^J_{nc}\,(U^J_{nc})^*$. Consequently, the fission cross section can be expressed in the form:

$$\sigma_f(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ |\rho_{12}|^2 + |\rho_{13}|^2 \right\} . \tag{2.30}$$

The capture cross section as a function of energy is obtained by subtracting Equation (2.30) from Equation (2.27):

$$\sigma_\gamma(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ Re\rho_{11} - |\rho_{11}|^2 - |\rho_{12}|^2 - |\rho_{13}|^2 \right\} . \tag{2.31}$$

As noted for the absorption cross section, a similar numerical instability problem can occur in the capture cross-section calculation using Equation (2.31) with the complex functions that are available in FORTRAN. To avoid the numerical instability, the quantity within braces in Equation (2.31) can be expanded, and a numerical expression can be developed to avoid any numerical instabilities. An expression is provided in Appendix C for calculating the quantity $Re\rho_{11} - |\rho_{11}|^2 - |\rho_{12}|^2 - |\rho_{13}|^2$.

The total cross section is calculated by summing the partial reactions. By summing Equations (2.28), (2.30) and (2.31), an expression is obtained for the total cross section as a function of energy:

$$\sigma_t(E) = \frac{4\pi}{k^2} \left\{ \sum_J g_J \left\{ (1 - 2\sin^2\phi_1)\, Re\rho_{11} + \sin(2\phi_1)\, Im\rho_{11} \right\} + \sum_{j=\left|SPI-\frac{1}{2}\right|}^{SPI+\frac{1}{2}} \sum_{J=|\ell-j|}^{\ell+j} g_J \sin^2\phi_1 \right\} . \tag{2.32}$$

POLIDENT uses Equations (2.28), (2.30), (2.31) and (2.32) to calculate the elastic, fission, capture and total cross sections as a function of energy for evaluations that specify the R-M resonance representation.

2.1.1.3 Adler-Adler

In addition to the Reich-Moore approximation, the Adler-Adler (AA) formalism is another approximation to the general R-matrix theory for describing neutron-nucleus interactions. Unlike the Reich-Moore approximation, the AA approach consists of an expansion of the collision matrix, U_{nc}, in terms of complex residues and poles.[5] The approach leads to a cross-section representation that can be expressed in terms of partial fractions involving the neutron energy and resonance parameters. One of the fundamental assumptions in the AA approximation involves neglecting the energy dependence of the neutron width, Γ_n. For low-energy s-wave neutrons (i.e., $\ell = 0$), the neutron width is much smaller than the gamma and fission widths (i.e., $\Gamma_\gamma + \Gamma_f$), and Γ_n has a negligible impact on the computed cross-section value.[4,6] For fissionable isotopes, the neutron width is much smaller than the gamma and fission widths at low energies, and the AA formalism is a suitable approximation for describing the neutron-nucleus interactions. As the energy dependence of the neutron width becomes important relative to the gamma and fission widths, the AA method is not suitable for representing the energy-dependent cross-section values.

Based on the underlying assumptions, the AA formalism is only applicable for $\ell = 0$ resonances. A detailed derivation of the AA formulae is not provided in the following discussion; however, the defining equations for the AA approximation are presented. For neutron-induced reactions, the total cross section as a function of energy is calculated with the following expression:[1]

$$\sigma_t(E) = \frac{4\pi}{k^2}\sin^2\phi_0 + \frac{\pi\sqrt{E}}{k^2}\left\{\sum_{r=1}^{NRS}\frac{\nu_r\left(G_r^T\cos 2\phi_0 + H_r^T\sin 2\phi_0\right) + (\mu_r - E)\left(H_r^T\cos 2\phi_0 - G_r^T\sin 2\phi_0\right)}{(\mu_r - E)^2 + \nu_r^2} + AT_1 + \frac{AT_2}{E} + \frac{AT_3}{E^2} + \frac{AT_4}{E^3} + BT_1E + BT_2E^2\right\}, \tag{2.33}$$

where

k	=	neutron-wave number as defined by Equation (2.3),
ϕ_0	=	phase shift for $\ell = 0$,
r	=	resonance index,
NRS	=	number of resonances,
ν_r	=	resonance half-width ($\Gamma/2$),
μ_r	=	resonance energy,
G_r^T	=	symmetrical total cross-section parameter corresponding to μ_r,
H_r^T	=	asymmetrical total cross-section parameter corresponding to μ_r,
AT_i	=	background constants for the total cross section,
BT_i	=	background constants for the total cross section.

Similar expressions are provided for the capture and fission cross sections as a function of energy:[1]

$$\sigma_\gamma(E) = \frac{\pi\sqrt{E}}{k^2}\left\{\sum_{r=1}^{NRS}\frac{\nu_r G_r^\gamma + (\mu_r - E)H_r^\gamma}{(\mu_r - E)^2 + \nu_r^2} + AC_1 + \frac{AC_2}{E} + \frac{AC_3}{E^2} + \frac{AC_4}{E^3} + BC_1E + BC_2E^2\right\}, \tag{2.34}$$

and

$$\sigma_f(E) = \frac{\pi\sqrt{E}}{k^2}\left\{\sum_{r=1}^{NRS}\frac{\nu_r G_r^f + (\mu_r - E)H_r^f}{(\mu_r - E)^2 + \nu_r^2} + AF_1 + \frac{AF_2}{E} + \frac{AF_3}{E^2} + \frac{AF_4}{E^3} + BF_1E + BF_2E^2\right\}, \tag{2.35}$$

where

G_r^γ	=	symmetrical capture cross-section parameter corresponding to μ_r,
H_r^γ	=	asymmetrical capture cross-section parameter corresponding to μ_r,
AC_i	=	background constants for the capture cross section,
BC_i	=	background constants for the capture cross section,
G_r^f	=	symmetrical fission cross-section parameter corresponding to μ_r,

H_r^f = asymmetrical fission cross-section parameter corresponding to μ_r,
AC_i = background constants for the fission cross section,
BC_i = background constants for the fission cross section.

POLIDENT calculates the total, capture and fission cross sections using Equations (2.33), (2.34) and (2.35), respectively. The AA representation does not provide an equation for the elastic-scattering cross section; rather, σ_{el} is calculated by subtracting the capture and fission cross sections from the total:

$$\sigma_{el}(E) = \sigma_t(E) - \sigma_\gamma(E) - \sigma_f(E) \ . \tag{2.36}$$

2.1.2 Unresolved-Resonance Region

As noted in Section 1.1, the URR is a region in which the measurement of individual resonances is very difficult. The unresolved-resonance data are averages of resolved-resonance parameters with respect to energy. Moreover, the resonance parameters are for the SLBW formalism, and each width is distributed according to a chi-square distribution with a specified number of degrees of freedom.[1] Since the unresolved-resonance parameters are averaged over a specified energy interval, ΔE, the approach is to calculate the effective cross sections at an energy point within ΔE using the average parameters. The method employed in POLIDENT for calculating cross sections in the URR originated from the ETOX and MC^2-II codes.[7,8] Note that the NJOY[9] module UNRESR calculational approach in the URR is also based upon the methods from ETOX and MC^2-II, and the following development is analogous to the methods presented for NJOY. An average cross section for reaction i at some energy E^* within ΔE is given by the following equation:

$$\bar{\sigma}_i(E^*) = \frac{\dfrac{1}{\Delta E}\displaystyle\int_{\Delta E} \sigma_i(E)\phi(E)dE}{\dfrac{1}{\Delta E}\displaystyle\int_{\Delta E} \phi(E)dE} \ , \tag{2.37}$$

where

$\phi(E)$ = scalar flux at energy E,
ΔE = energy range for specified unresolved-resonance parameters (E_2 - E_1).

In Equation (2.37), the index i can represent capture, fission, elastic or total. Also, the energy interval ΔE is assumed to contain a large number of resonances; however, the flux over the interval must be slowly varying with respect to E. The major difficulty associated with calculating the average cross section at E^* is the determination of the weighting function or flux. The approach is to assume that the weighting function can be estimated using the narrow resonance approximation (NR). The assumption is valid provided that the width of a resonance is narrow relative to the average-scattering width. In other words, a neutron scattering through a resonance will experience, at most, one collision in the resonance. This assumption is valid at typical energies within the unresolved range. Based on the NR approximation, the flux has the asymptotic form:

$$\phi(E) = \frac{1}{\Sigma_t(E)} \ , \tag{2.38}$$

where

$\Sigma_t(E)$ = macroscopic total cross section for the mixture.

With regard to the isotope of interest, the macroscopic total cross section is assumed to be separable such that the contributions from the other isotopes in the mixture are characterized by a constant σ_0:

$$\Sigma_t(E) = N_m\sigma_{m_t}(E) + \sum_{n\neq m} N_n\sigma_{n_t}(E) = N_m\left[\sigma_{m_t} + \sum_{n\neq m} \frac{N_n}{N_m}\,\sigma_{n_t}(E)\right] = N_m\left[\sigma_{m_t} + \sigma_0\right] . \tag{2.39}$$

Using Equations (2.38) and (2.39) and dropping the material subscript *m*, the effective cross section for reaction *i* at energy E^* can be expressed as follows:

$$\bar{\sigma}_i(E^*) = \frac{\displaystyle\int_{\Delta E} \frac{\sigma_i(E)}{\sigma_t + \sigma_0}\,dE}{\displaystyle\int_{\Delta E} \frac{1}{\sigma_t + \sigma_0}\,dE} . \tag{2.40}$$

Since a background component may be provided in File 3 for the cross sections in the unresolved-resonance range, it is convenient to separate the cross-section quantities into a resonant component and a background component:

$$\sigma_i(E) = \sigma_{b_i} + \sigma_{\lambda_i} , \tag{2.41}$$

where

σ_{bi} = cross-section background component for reaction *i* (constant),
$\sigma_{\lambda i}$ = cross-section resonant component for reaction *i*.

The background quantity in Equation (2.41) must also include the potential-scattering contribution for the elastic scattering and total cross sections. Using Equation (2.41), the effective cross section for reaction *i* in the unresolved resonance range becomes

$$\bar{\sigma}_i(E^*) = \frac{\displaystyle\int_{\Delta E} \frac{\sigma_{b_i} + \sigma_{\lambda_i}(E)}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE}{\displaystyle\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE} , \tag{2.42}$$

$$\bar{\sigma}_i(E^*) = \frac{\sigma_{b_i}\displaystyle\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE}{\displaystyle\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE} + \frac{\displaystyle\int_{\Delta E} \frac{\sigma_{\lambda_i}(E)}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE}{\displaystyle\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{b_t} + \sigma_0}\,dE} . \tag{2.43}$$

In Equation (2.43), the constant quantities σ_{bt} and σ_0 can be grouped together and defined in terms of a constant σ_{0b}:

$$\sigma_{0b} = \sigma_{b_t} + \sigma_0 \ . \tag{2.44}$$

Using the definition from Equation (2.44), the average cross section at energy E^* can be defined as follows:

$$\bar{\sigma}_i(E^*) = \sigma_{b_i} + \frac{\displaystyle\int_{\Delta E} \frac{\sigma_{\lambda_i}(E)}{\sigma_{\lambda_t} + \sigma_{0b}}\, dE}{\displaystyle\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{0b}}\, dE} \ . \tag{2.45}$$

Prior to evaluating the cross-section quantity in Equation (2.45), some additional algebraic manipulations can be performed on the integral in the denominator. In particular, the integrand in the denominator can be expressed as follows:

$$\frac{1}{\sigma_{0b} + \sigma_{\lambda_t}} = \frac{1}{\sigma_{0b}} \left(1 - \frac{\sigma_{\lambda_t}}{\sigma_{0b} + \sigma_{\lambda_t}} \right) \ . \tag{2.46}$$

The denominator in Equation (2.45) can be evaluated by substituting the expression from Equation (2.46) as follows:

$$\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{0b}}\, dE = \frac{1}{\sigma_{0b}} \int_{\Delta E} dE - \frac{1}{\sigma_{0b}} \int_{\Delta E} \frac{\sigma_{\lambda_t}}{\sigma_{0b} + \sigma_{\lambda_t}}\, dE \ , \tag{2.47}$$

and

$$\int_{\Delta E} \frac{1}{\sigma_{\lambda_t} + \sigma_{0b}}\, dE = \frac{\Delta E}{\sigma_{0b}} \left\{ 1 - \frac{1}{\Delta E} \int_{\Delta E} \frac{\sigma_{\lambda_t}}{\sigma_{0b} + \sigma_{\lambda_t}}\, dE \right\} \ . \tag{2.48}$$

By substituting the relation from Equation (2.48) into Equation (2.45), the average cross section at energy E^* can be redefined as follows:

$$\bar{\sigma}_i(E^*) = \sigma_{b_i} + \frac{\displaystyle\int_{\Delta E} \frac{\sigma_{\lambda_i}(E)}{\sigma_{\lambda_t} + \sigma_{0b}}\, dE}{\dfrac{\Delta E}{\sigma_{0b}} \left\{ 1 - \dfrac{1}{\Delta E} \displaystyle\int_{\Delta E} \frac{\sigma_{\lambda_t}}{\sigma_{0b} + \sigma_{\lambda_t}}\, dE \right\}} \ . \tag{2.49}$$

Note that the integrals in Equation (2.49) have a similar form, and further simplification of the equation can be obtained by defining the resonance-fluctuation integral:

$$I_{0_i} = \frac{1}{\Delta E} \int_{\Delta E} \frac{\sigma_{\lambda_i}}{\sigma_{0b} + \sigma_{\lambda_t}}\, dE \ . \tag{2.50}$$

The average cross section at energy E^* can be expressed in terms of the resonance-fluctuation integral from Equation (2.50):

$$\bar{\sigma}_i(E^*) = \sigma_{b_i} + \frac{\sigma_{0b}\, I_{0_i}}{1 - I_{0_t}} \ . \tag{2.51}$$

Based on the expression in Equation (2.51), the calculation of the average cross section is reduced to evaluating the appropriate resonance-fluctuation integrals.

The following discussion outlines the steps for calculating the fluctuation integrals and identifies the appropriate assumptions that are used in the numerical evaluation of the integrals. Ultimately, the fluctuation integral will be expressed in terms of the appropriate resonance parameters. Therefore, the resonance-dependent cross-section quantity in Equation (2.50) can be expressed in terms of a summation-over-spin sequence (s) and resonance index (r) corresponding to the appropriate spin sequence:

$$\sigma_i(E) = \sigma_{b_i} + \sum_s \sum_r \sigma_{isr}(E - E_{sr}) = \sigma_{b_i} + \sum_{sr} \sigma_{isr}(E - E_{sr}) \ . \tag{2.52}$$

The resonant cross-section component in Equation (2.52) is a function of energy E as well as the center energy for the resonance, E_{sr}. The resonant cross-section quantity can be inserted into the equation for the fluctuation integral that is defined by Equation (2.50):

$$I_{0_i} = \frac{1}{\Delta E} \int_{\Delta E} \frac{\sum_{sr} \sigma_{isr}(E - E_{sr})}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}(E - E_{sr})} \, dE \ . \tag{2.53}$$

The integrand in Equation (2.53) can be reexpressed in the following form (i.e., temporarily drop cross-section functional dependence):

$$\frac{\sum_{sr} \sigma_{isr}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} = \sum_{sr} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} \left\{ 1 - \sum_{r' \neq r} \frac{\sigma_{tsr'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} \right\} . \tag{2.54}$$

The second term in Equation (2.54) can be expanded as follows:

$$\sum_{r' \neq r} \frac{\sigma_{tsr'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} = \sum_{r' \neq r} \frac{\sigma_{tsr'}}{\sigma_{0b} + \sigma_{tsr} + \sigma_{tsr'}} \left(1 - \sum_{\substack{r'' \neq r \\ r'' \neq r'}} \frac{\sigma_{tsr''}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} \right) . \tag{2.55}$$

By substituting Equation (2.55) into Equation (2.54) and neglecting the product of three different resonances in a sequence, the following expression is obtained for the integrand of the fluctuation integral:

$$\frac{\sum_{sr} \sigma_{isr}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} = \sum_{sr} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} \left\{ 1 - \sum_{r' \neq r} \frac{\sigma_{tsr'}}{\sigma_{0b} + \sigma_{tsr} + \sigma_{tsr'}} \right\} \left\{ 1 - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} \right\} . \tag{2.56}$$

Note that the first factor in Equation (2.56) represents the isolated-resonance contribution, and the first and second quantities in brackets represent the in-sequence overlap correction and the sequence-sequence overlap correction, respectively.

As noted previously, the method of calculating cross sections in the URR originated from the preceding ETOX and MC^2-II codes. The method employed by these earlier codes assumes that the dominant resonance-overlap contribution comes from resonances in different spin sequences. Consequently, the method neglects the in-sequence overlap contribution. Note that the right-hand side of Equation (2.56) has the following form:

$$a(1 - b)(1 - c) = a[1 - c - b + bc] = a[(1 - c) - (b - bc)]. \tag{2.57}$$

By neglecting the in-sequence contribution, the integrand of the fluctuation integral reduces to the following:

$$\frac{\sum_{sr} \sigma_{isr}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} = \sum_{sr} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} \left\{ 1 - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} \right\} . \tag{2.58}$$

Note that Equation (2.58) is a recursion relation, and the ETOX/MC^2-II approximation stops the recursion after one iteration. Consequently, the integrand of the fluctuation integral is expressed as follows:

$$\frac{\sum_{sr} \sigma_{isr}}{\sigma_{0b} + \sum_{sr} \sigma_{tsr}} = \sum_{sr} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} \left\{ 1 - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sigma_{ts'r'}} \right\} . \tag{2.59}$$

After substituting Equation (2.59) into Equation (2.53), the fluctuation integral for reaction *i* becomes:

$$I_{0_i} = \frac{1}{\Delta E} \int_{\Delta E} \sum_{sr} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} \left\{ 1 - \sum_{s' \neq s} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sigma_{ts'r'}} \right\} dE . \tag{2.60}$$

The method further assumes that the resonances in different spin sequences are uncorrelated. As a result, the integral of the product of the two functions in Equation (2.60) reduces to the product of two integrals of the following form:

$$I_{0_i} = \sum_{s} \frac{1}{\Delta E} \int_{\Delta E} \sum_{r} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} dE \left\{ 1 - \sum_{s' \neq s} \frac{1}{\Delta E} \int_{\Delta E} \sum_{r'} \frac{\sigma_{ts'r'}}{\sigma_{0b} + \sigma_{ts'r'}} dE \right\} . \tag{2.61}$$

In order to evaluate the integral of Equation (2.61), the following integral quantity is defined for a particular spin sequence *s*:

$$B_{is} = \frac{1}{\Delta E} \int_{\Delta E} \sum_{r} \frac{\sigma_{isr}}{\sigma_{0b} + \sigma_{tsr}} dE . \tag{2.62}$$

As noted previously, one of the underlying assumptions of the unresolved-resonance treatment is that the energy interval, ΔE, is wide enough to contain a large number of resonances. The integration-interval assumption also implies that the range of integration is large relative to the width of any resonance in the interval. Based on the integration-interval assumption, the variable of integration can be defined as $\xi = E - E_{sr}$ and the range of integration can be extended to infinity:

$$B_{is} = \frac{1}{\Delta E} \int_{-\infty}^{\infty} \sum_{r} \frac{\sigma_{isr}(\xi)}{\sigma_{0b} + \sigma_{isr}(\xi)} \, d\xi \ . \tag{2.63}$$

For a specific spin sequence, the energy interval ΔE is equal to the product of the average-resonance spacing in the sequence, D_s, and the number of resonances within the energy interval, N_s. Using the resonance information, B_{is} has the following form:

$$B_{is} = \frac{1}{D_s \, N_s} \int_{-\infty}^{\infty} \sum_{r} \frac{\sigma_{isr}(\xi)}{\sigma_{0b} + \sigma_{isr}(\xi)} \, d\xi \ . \tag{2.64}$$

Using Equation (2.64), a more compact expression is obtained for the fluctuation integral I_0:

$$I_{0_i} = \sum_{s} B_{is} \left(1 - \sum_{s' \neq s} B_{ts'} \right) . \tag{2.65}$$

Before Equation (2.65) can be used, a numerical expression must be developed for the integral B_{is}. In the unresolved region, average-resonance widths are provided that are distributed according to a chi-square distribution. As noted in Equation (2.64), B_{is} is dependent upon a resonance-dependent-cross section quantity. Furthermore, each energy interval is assumed to be comprised of many resonances. As a result, the summation over resonances in Equation (2.64) can be changed to a multiple integration of the resonance widths times the probability of finding a resonance with a particular value of the resonance widths:

$$\frac{1}{N_s} \sum_{r \in s} f_r = \int_{0}^{\infty} d\Gamma_c \, P_s^{\mu}(\Gamma_c) \int_{0}^{\infty} d\Gamma_{c'} \, P_s^{\nu}(\Gamma_{c'}) \ .... \ f(\Gamma_c \ , \Gamma_{c'} \ , ...) \ , \tag{2.66}$$

where

f_r	=	function that is defined for a specific resonance index r,
Γ_c	=	resonance width for some exit channel c,
$\Gamma_{c'}$	=	resonance width for some exit channel c',
P_c^{μ}	=	probability distribution for Γ_c with μ degrees of freedom,
$P_{c'}^{\nu}$	=	probability distribution for $\Gamma_{c'}$ with ν degrees of freedom.

By using the unresolved-resonance-parameter information, the expression for B_{is} becomes:

$$B_{is} = \frac{1}{D_s} \int_0^{\infty} P(\alpha) \int_{-\infty}^{\infty} \frac{\sigma_{is\alpha}(\xi)}{\sigma_{0b} + \sigma_{ts\alpha}(\xi)} \, d\xi \, d\alpha \ . \tag{2.67}$$

Note that the multiple integration over the different resonance-parameter widths is denoted by a single integration over a single parameter α in Equation (2.67). The abbreviated integral representation is used to simplify the expressions in the subsequent discussion.

For a reaction *i*, the resonant component of the cross section is calculated as follows:

$$\sigma_{isr}(E - E_{sr}) = \sigma_{isr}(\xi) = \left(\frac{4\pi g_J}{k^2} \frac{\Gamma_n \Gamma_i}{\Gamma_t^2} \psi(\theta, X) \right) , \tag{2.68}$$

where

$$\psi(\theta, X) = \frac{\theta\sqrt{\pi}}{2} \, Re\, W\left(\frac{\theta X}{2}, \frac{\theta}{2} \right) , \tag{2.69}$$

$$W(x,y) = e^{-(x + iy)^2} erfc[\ -i(x + iy)\ ] \ , \tag{2.70}$$

$$\theta = \sqrt{\frac{A}{4kTE_{sr}}} \Gamma_t \ , \tag{2.71}$$

and

$$X = \frac{2(E - E_{sr})}{\Gamma_t} \ . \tag{2.72}$$

With regard to the total, the resonant component of the total cross section is calculated with the following equation:

$$\sigma_{tsr}(E - E_{sr}) = \sigma_{tsr}(\xi) = \left(\frac{4\pi g_J}{k^2} \frac{\Gamma_n}{\Gamma_t} \right) \left(\cos(2\phi_\ell) \psi(\theta, X) + \sin(2\phi_\ell) \chi(\theta, X) \right)_{sr} , \tag{2.73}$$

where

$$\chi(\theta,X) = \theta\sqrt{\pi}\ Im\,W\left(\frac{\theta X}{2},\frac{\theta}{2}\right) . \tag{2.74}$$

In Equations (2.69) and (2.74), the functions ψ and χ are the symmetric- and antisymmetric-line-shape functions that are documented throughout the reactor physics literature. The function W in Equations (2.69) and (2.74) is the complex-probability integral.

A numerical expression for the isolated-resonance integral can be defined based on the cross-section expressions that are defined by Equations (2.68) through (2.74). The equation for B_{is} can be redefined as follows:

$$B_{is} = \frac{1}{D_s}\int_0^{\infty} P(\alpha)\int_{-\infty}^{\infty}\frac{\sigma_m\Gamma_i}{\Gamma_t}\,\frac{\psi(\theta,X)}{\sigma_{0b} + \sigma_m\,[\,\cos(2\phi_\ell)\psi(\theta,X) + \sin(2\phi_\ell)\chi(\theta,X)\,]}\,d\xi\ d\alpha\ , \tag{2.75}$$

where

$$\sigma_m = \frac{4\pi g_J}{k^2}\frac{\Gamma_n}{\Gamma_t} . \tag{2.76}$$

Equation (2.75) can be simplified further by changing the variable of integration from ξ to X and defining a quantity β as follows:

$$\beta = \frac{\sigma_{0b}}{\sigma_m\cos(2\phi_\ell)} . \tag{2.77}$$

As a result, the equation for B_{is} is defined with the following relation:

$$B_{is} = \frac{1}{D_s}\int_0^{\infty} P(\alpha)\int_{-\infty}^{\infty}\frac{\Gamma_i}{2\,\cos(2\phi_\ell)}\left[\frac{\psi(\theta,X)}{\beta + \psi(\theta,X) + \tan(2\phi_\ell)\chi(\theta,X)}\right]dX\ d\alpha\ . \tag{2.78}$$

The MC^2-II manual defines the integral over X as the general-resonance integral of the form:[8]

$$J(\beta,\theta,a,b) = \frac{1}{2}\int_{-\infty}^{\infty}\left[\frac{\psi(\theta,X) + b\chi(\theta,X)}{\beta + \psi(\theta,X) + a\chi(\theta,X)}\right]dX\ . \tag{2.79}$$

Using the general-resonance-integral definition for J, the equation for B_{is} becomes

$$B_{is} = \frac{1}{D_s} \int_0^\infty P(\alpha) \frac{\Gamma_i}{\cos(2\phi_\ell)} J(\beta,\theta,\tan 2\phi_\ell,0)\ d\alpha \ . \tag{2.80}$$

With regard to the total cross section, the equation for B_{ts} is defined in a similar manner:

$$B_{ts} = \frac{1}{D_s} \int_0^\infty P(\alpha) \int_{-\infty}^\infty \frac{\Gamma_t}{2} \left[\frac{\psi(\theta,X) + \tan(2\phi_\ell)\chi(\theta,X)}{\beta + \psi(\theta,X) + \tan(2\phi_\ell)\chi(\theta,X)} \right] dX\ d\alpha \ , \tag{2.81}$$

$$B_{ts} = \frac{1}{D_s} \int_0^\infty P(\alpha)\ \Gamma_t\ J(\beta,\theta,\tan 2\phi_\ell,\tan 2\phi_\ell)\ d\alpha \ . \tag{2.82}$$

Note that POLIDENT uses the subroutine AJKU to calculate the J integrals that are present in the equations for B_{is} and B_{ts}. Moreover, AJKU only computes $J(\beta,\theta,0,0)$, thereby neglecting interference effects. Since the tangent of the phase shift is assumed to be zero (i.e., $\tan 2\phi_\ell = 0$), the cosine of the phase shift must be 1 (i.e., $\cos 2\phi_\ell = 1$). Consequently, the equations for B_{is} and B_{ts} reduce to the following:

$$B_{is} = \frac{1}{D_s} \int_0^\infty P(\alpha)\ \Gamma_i\ J(\beta,\theta,0,0)\ d\alpha \ , \tag{2.83}$$

and

$$B_{ts} = \frac{1}{D_s} \int_0^\infty P(\alpha)\ \Gamma_t\ J(\beta,\theta,0,0)\ d\alpha \ . \tag{2.84}$$

In order to complete the numerical evaluation for B_{is} and B_{ts}, the remaining integrals over the probability distributions for the resonance widths must be addressed. The multiple integration is evaluated using a ten-point-quadrature scheme that was developed for MC^2-II. The integral over the different chi-squared probability distributions has the following form:[8]

$$\langle f(x,y)\rangle = \int_0^\infty P_\mu(x)\,dx \int_0^\infty P_\nu(y)\ f(x,y)\ dy \ , \tag{2.85}$$

where

P_μ = probability distribution for x with μ degrees of freedom,
P_ν = probability distribution for y with ν degrees of freedom.

The statistical integration approach that was developed for MC^2-II evaluates the integrals in Equation (2.85) using the following ten-point-quadrature scheme:

$$<f(x,y)> = \sum_{j=1}^{10} \sum_{k=1}^{10} A_j A_k f(X_j, Y_k) \ . \tag{2.86}$$

The A_j and X_j values in Equation (2.86) are the quadrature weights and values, respectively.

In the URR, the ENDF formats provide an average-reduced-neutron-line width (Γ_n), an average-radiation width (Γ_γ), an average-fission width (Γ_f) and an average-competitive-reaction width (Γ_x). The average widths can be combined to provide the total width:

$$\Gamma_t = \Gamma_n + \Gamma_f + \Gamma_\gamma + \Gamma_x \ . \tag{2.87}$$

The ten-point-quadrature scheme can be used to complete the numerical evaluation for B for a reaction i and spin sequence s:

$$B_{is} = \frac{1}{D_s} \sum_{nn=1}^{10} \sum_{nf=1}^{10} \sum_{nx=1}^{10} A_{nn} A_{nf} A_{nx} \Gamma_i Q_{ni} J(\beta,\theta,0,0) \ , \tag{2.88}$$

where

Γ_i = average width for reaction i,
nn = index corresponding to reduced neutron-line width,
nf = index corresponding to fission width,
nx = index corresponding to competitive width,
A_{nn} = quadrature weight corresponding to nn index,
A_{nf} = quadrature weight corresponding to nf index,
A_{nx} = quadrature weight corresponding to nx index,
Q_{ni} = quadrature weight corresponding to ni index where i is either n, f, x or γ. Note that $Q_{n\gamma}$ is 1.0.

The approach for the URR assumes that Γ_γ is constant, and there is no integration over the radiation-width distribution. A similar expression can be derived for B_{ts} as follows:

$$B_{ts} = \frac{1}{D_s} \sum_{nn=1}^{10} \sum_{nf=1}^{10} \sum_{nx=1}^{10} A_{nn} A_{nf} A_{nx} \left[\Gamma_n Q_{nn} + \Gamma_f Q_{nf} + \Gamma_\gamma + \Gamma_x Q_{nx} \right] J(\beta,\theta,0,0) \ . \tag{2.89}$$

The fluctuation integral can be calculated for a reaction *i*, but substituting Equations (2.88) and (2.89) into the equation for I_0:

$$I_{0_i} = \sum_s B_{is} \left(1 - \sum_{s' \neq s} B_{ts'} \right) . \tag{2.90}$$

Subsequently, the average cross section at energy E^* for reaction *i* is obtained by substituting the calculated fluctuation integrals into Equation (2.51)

$$\bar{\sigma}_i(E^*) = \sigma_{b_i} + \frac{\sigma_{0b} \left[\sum_s B_{is} \left(1 - \sum_{s' \neq s} B_{ts'} \right) \right]}{1 - \left[\sum_s B_{ts} \left(1 - \sum_{s' \neq s} B_{ts'} \right) \right]} . \tag{2.91}$$

Equation (2.91) is the basis for the calculation of unresolved cross sections in POLIDENT.

2.2 Energy-Mesh Generation

Although the ENDF formats provide resonance-parameter information for the resonances in File 2, the formats do not provide an energy mesh for calculating the cross-section values based on the resonance parameters. In the resonance region, the cross-section values can vary rapidly as a function of energy. Consequently, the cross-section function can be extremely dependent on the energy grid. If certain energy values or grid points are omitted, the structure of the resonances may not be represented correctly during reconstruction from the resonance parameters. Therefore, the precision of the continuous-energy cross-section data that are calculated from the File 2 parameters is strongly dependent upon the energy mesh. The subsequent sections describe the energy-mesh-generation schemes that are currently used in POLIDENT.

2.2.1 Resolved-Resonance Region

Historically, energy-mesh-generation schemes begin with a very coarse grid, and points are added to the coarse grid using a "halving-iteration" scheme until a desired convergence tolerance is achieved. Based on experience, the "halving" approach, coupled with a coarse intial energy grid, can lead to inadequate representations of the resonance structure particularly near inflection points. In an effort to avoid potential problems associated with previous energy-grid-generation schemes, a new adaptive-meshing scheme has been developed for POLIDENT for the resolved-resonance region (RRR). In the RRR, the energy-mesh-generation scheme is based largely on the determination of a very-fine energy mesh that is subsequently collapsed to an auxiliary- or user-grid structure using a user-specified convergence tolerance (see Figure 2.1). In the new mesh-generation scheme, a significant amount of effort is devoted to the construction of a fine-energy grid to ensure that the detailed resonance structure is represented faithfully. As an overview of the energy-grid construction, the following steps are used in the mesh-generation scheme:

Step	Description
1.	Estimate a fine ΔE based upon the resonance parameters
2.	Calculate cross sections with appropriate resonance formalism
3.	Calculate minimum, maximum and inflection points (critical points)
4.	Add points between inflection and maximum grid points
5.	Use halving-iteration scheme to add points to fine grid
6.	Collapse fine grid to auxiliary grid

A description of each step in the energy-mesh generation scheme is provided in the subsequent discussion.

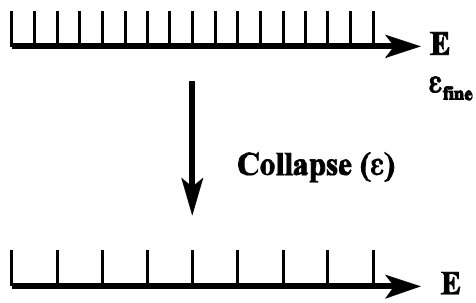


Figure 2.1 Mesh generation for resolved-resonance region

Step 1: Estimate fine ΔE from resonance parameters

The fine-energy mesh construction is a very significant component of the meshing scheme. Initially, POLIDENT reads the File 2 resonance parameters and subdivides the RRR into decade (i.e., 10-eV) intervals. Within each decade, the mean level spacing, $<D>$, and the mean neutron-line width, $<\Gamma_n>$, are calculated. The initial estimate of the energy increment, ΔE, for each decade is considered to be directly proportional to the mean neutron-line width and mean level spacing as follows:

$$\Delta E \propto \sqrt{\frac{<\Gamma_n>}{<D>}} \ , \tag{2.92}$$

or

$$\Delta E = C \sqrt{\frac{<\Gamma_n>}{<D>}} \ . \tag{2.93}$$

In Equation (2.93), C is a constant multiplier and has an empirical value of 0.1. The initial estimate of the energy grid provides a very fine initial mesh that is suitable for calculating cross-section values.

Step 2: Calculate cross sections

Based on the initial estimate of ΔE, POLIDENT calculates the absorption, capture, fission, scattering and total cross-section values using the ENDF-specified functional representation (i.e., Reich-Moore, single- and multi-level Breit Wigner). Use of the ENDF-specified format provides a rigorous treatment of the cross-section data and accounts for interference effects that are induced by other resonance reactions.

Step 3: Calculate critical points

Once the initial fine-grid cross-section values are calculated, the minimum, maximum and inflection points (i.e., critical points) are determined numerically for each reaction. A schematic diagram of an arbitrary resonance constructed from the fine-grid values is provided in Figure 2.2. Following the calculation of the critical points, each resonance is evaluated to ensure each resonance peak corresponding to the File 2 resonance energy is present in the fine energy grid, and the resonance energy points from File 2 are added to the fine grid as needed.

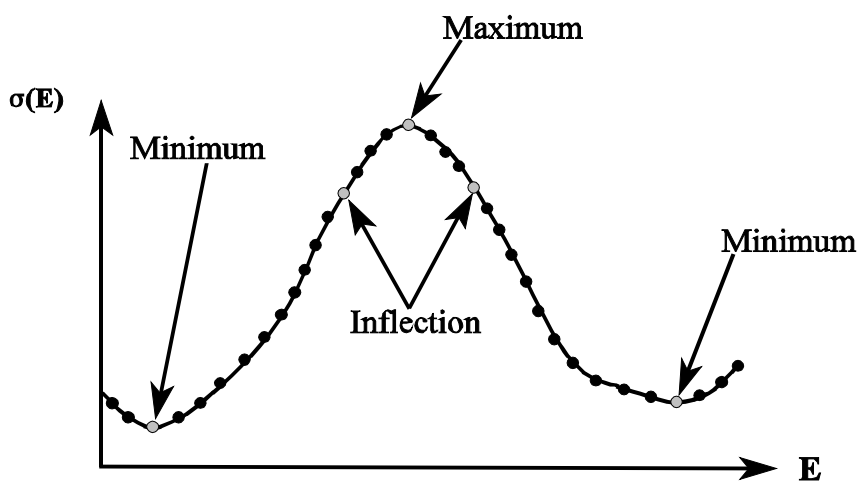


Figure 2.2 Example resonance with critical points

Step 4: Add points between inflection and maximum points

Additional points are added between the resonance peak and the points of inflection based on the slope between the two points, as illustrated in Figure 2.3. The number of points, N, that are added between the inflection point and resonance peak is based on the following empirical criteria:

If $\tan^{-1}(|m|) < \pi/6$, N = 10.
If $\pi/6 \leq \tan^{-1}(|m|) < \pi/3$, N = 15.
If $\pi/3 \leq \tan^{-1}(|m|) < \pi/2$, N = 20.

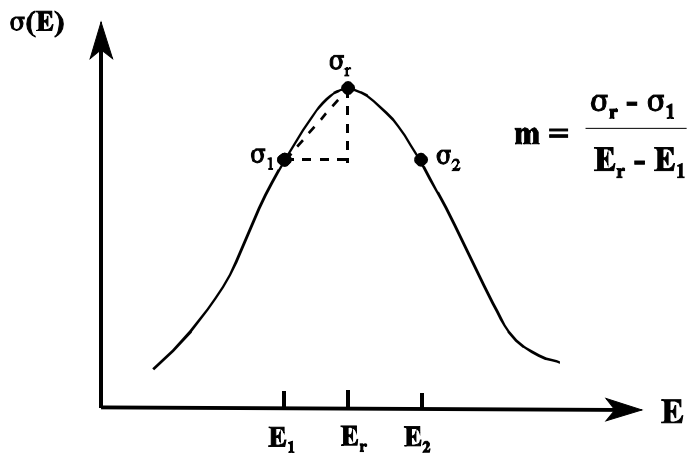


Figure 2.3 Slope calculation using critical points

Step 5: Halving-iteration scheme

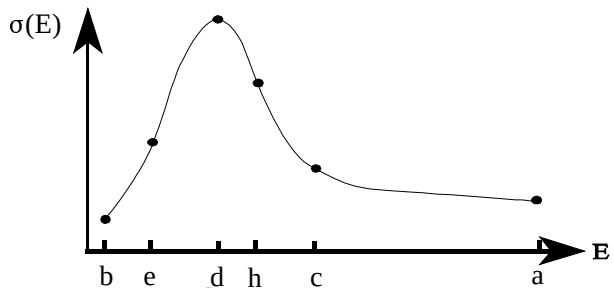
To complete the fine-grid determination, a halving-iteration scheme is also used to add points until the fine-grid cross sections are within a tolerance, ε_{fine}, of the actual cross-section data using linear interpolation. The fine-grid tolerance is less than the desired auxiliary-grid tolerance, ε_{aux}, (i.e., $0.1\varepsilon_{aux} \leq \varepsilon_{fine} < \varepsilon_{aux}$). Prior to the halving-iteration scheme, the fine grid may have more than enough points to satisfy the ε_{fine} criterion in all or part of the RRR; however, the iteration scheme ensures enough points are present in the fine grid such that the actual cross-section data can be reproduced to at least ε_{fine}. In other words, the maximum difference between the actual cross-section data and the fine grid is ε_{fine}.

The halving-iteration scheme that is used in the final step of the fine-grid construction is similar to the traditional halving scheme that is used in other resonance-processing codes such as RECONR.[9] Typically, in the traditional halving approach, an energy mesh is constructed from a coarse grid that is based on the ENDF File 3 point data. During resonance reconstruction, two adjacent energy points from the coarse grid are used to define a panel. In the traditional halving approach, the cross-section value is calculated at the midpoint of the panel using the appropriate resonance formalism (e.g., Reich-Moore, Adler-Adler, etc.). In addition, the cross-section value is also estimated at the midpoint by linearly interpolating between the endpoints of the panel. If the estimated cross-section value is within a desired convergence tolerance relative to the value calculated with the resonance equations, the panel is converged, and the endpoints of the panel are stored in the energy grid. However, if the panel is not converged, the panel is divided again, and the linear-interpolation test is repeated for the smaller panel. The panel is divided into smaller and smaller panels until the desired convergence tolerance is achieved. As a result, points are added to the initial grid by halving each panel and inserting points until a desired convergence tolerance is achieved. Moreover, the traditional halving scheme builds the energy grid from the initial coarse-grid structure (i.e., a "bottom-up"-type approach). Historically, the halving approach works well if the maximum error occurs at the midpoint of the panel. However, if the maximum error occurs at another point within an interpolation panel (e.g., inflection point), the halving approach could miss an important energy point in the grid construction.

A slightly different approach is used in POLIDENT. As opposed to using a very coarse mesh as the initial grid, a very-fine energy mesh is used as the initial grid for the halving-iteration scheme. An example illustration of the halving scheme is presented in Figure 2.4. A description of the halving scheme in Figure 2.4 is provided in the following discussion:

> Initially, an interpolation panel has two fine-grid points **(b and a)**, and the panel is divided into two equal parts, as shown in Figure 2.4 (i.e., iteration i = 2, with midpoint at point **c**). In the halving approach, the "true" cross-section value is calculated at the midpoint **(c)** of the panel (i.e. σ**(c)**) using the appropriate resonance formalism. The cross-section value at **c** is also estimated by linearly interpolating between the endpoints of the panel. As shown in Figure 2.4 for the second iteration, the estimated cross-section value at **c** is not within ε_{fine} of the "true" cross-section value at **c**, and the **b-a** panel is not converged. Therefore, the lower panel that is defined by points **b** and **c** is divided into two equal parts at point **d** (i.e., iteration i = 3), and the linear-interpolation test is repeated at the midpoint. Based on the test results in the third iteration, the estimated cross-section value at **d** does not satisfy the convergence criterion, and the **b-d** panel must be halved at point **e**.
>
> In the fourth iteration, the intermediate value at **e** is estimated by linearly interpolating between points **b** and **d**; however, the **b-d** panel is not converged at the midpoint, and the **b-e** panel is divided at point **f** in iteration 5. As shown in Figure 2.4 for the fifth iteration, the estimated cross-section value at point **f** is within ε_{fine} of σ**(f)**, and the convergence criterion is satisfied for the **b-e** panel. Therefore, point **b** is added to the energy grid. Since the convergence criterion is satisfied at point **f**, no other points are tested in the **b-e** panel. As a result, the entire panel is accepted, and the next interpolation panel that is defined by points **e-d** is evaluated. In the sixth iteration, the **e-d** panel is divided at point **g**, and the interpolation test is repeated at the midpoint. Since the convergence criterion is satisfied at point **g**, point **e** is added to the fine energy grid.
>
> The next interpolation panel that must be considered is panel **d-c,** which has a midpoint at **h**. For the seventh iteration, the **d-c** panel does not satisfy the convergence criterion at the midpoint. Consequently, the **d-h** panel must be divided at point **i**. Based on the test results in the eighth iteration, the **d-h** panel is converged at point **i**, and point **d** is added to the fine energy grid. The next interpolation panel that must be evaluated is panel **h-c,** which has a midpoint at **j**. For the ninth iteration, the convergence criterion is satisfied at point **j**, and point **h** is added to the grid. In the final iteration, the **c-a** panel is tested at the midpoint **k**. As shown in Figure 2.4 for the tenth iteration, the convergence criterion is satisfied at the midpoint, and both points **c** and **a** are added to the fine energy grid.
>
> Once the evaluation of the fine-grid panel that was originally defined by points **b** and **a** is complete, the next fine-grid panel is processed. In particular, point **a** will define the lower boundary of the next fine-grid panel, and the halving-iteration scheme is repeated in the next panel. The halving procedure is repeated for each fine-grid panel throughout the RRR.

Although POLIDENT uses the halving-iteration scheme, the method is different from other methods because a fine energy grid is largely determined prior to the implementation of the halving-iteration scheme. In addition, the critical points, which include the inflection points, are calculated numerically prior to using the halving-iteration scheme. As a result, POLIDENT uses the halving-iteration scheme as a consistency check to ensure that the desired fine grid is constructed. Moreover, any errors in the energy-grid construction that are attributed to the halving-iteration scheme are considered to be a second-order effect. Once the iterations are complete, a very-fine energy-grid structure is created. An example diagram of an arbitrary fine-grid structure is provided in Figure 2.5.



$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Iteration (i)		Test Panel	Result	Action
i = 1	b a			Halve b-a panel
2	b c a	$x = c$, $x_1 = b$, $x_2 = a$, $y_1 = \sigma(b)$, $y_2 = \sigma(a)$; $\lvert y - \sigma(c) \rvert > \varepsilon_{fine}\, \sigma(c)$?	Yes	Halve b-c panel
3	b d c a	$\mathbf{x = d}$, $x_1 = b$, $\mathbf{x_2 = c}$, $y_1 = \sigma(b)$, $\mathbf{y_2 = \sigma(c)}$; $\lvert y - \sigma(d) \rvert > \varepsilon_{fine}\, \sigma(d)$?	Yes	Halve b-d panel
4	b e d c a	$\mathbf{x = e}$, $x_1 = b$, $\mathbf{x_2 = d}$, $y_1 = \sigma(b)$, $\mathbf{y_2 = \sigma(d)}$; $\lvert y - \sigma(e) \rvert > \varepsilon_{fine}\, \sigma(e)$?	Yes	Halve b-e panel
5	b f e d c a	$\mathbf{x = f}$, $x_1 = b$, $\mathbf{x_2 = e}$, $y_1 = \sigma(b)$, $\mathbf{y_2 = \sigma(e)}$; $\lvert y - \sigma(f) \rvert > \varepsilon_{fine}\, \sigma(f)$?	No	Add b to grid Halve e-d panel
6	e g d c a	$\mathbf{x = g}$, $\mathbf{x_1 = e}$, $\mathbf{x_2 = d}$, $\mathbf{y_1 = \sigma(e)}$, $\mathbf{y_2 = \sigma(d)}$; $\lvert y - \sigma(g) \rvert > \varepsilon_{fine}\, \sigma(g)$?	No	Add e to grid Halve d-c panel
7	d h c a	$\mathbf{x = h}$, $\mathbf{x_1 = d}$, $\mathbf{x_2 = c}$, $\mathbf{y_1 = \sigma(d)}$, $\mathbf{y_2 = \sigma(c)}$; $\lvert y - \sigma(h) \rvert > \varepsilon_{fine}\, \sigma(h)$?	Yes	Halve d-h panel
8	d i h c a	$\mathbf{x = i}$, $x_1 = d$, $\mathbf{x_2 = h}$, $y_1 = \sigma(d)$, $\mathbf{y_2 = \sigma(h)}$; $\lvert y - \sigma(i) \rvert > \varepsilon_{fine}\, \sigma(i)$?	No	Add d to grid Halve h-c panel
9	h j c a	$\mathbf{x = j}$, $\mathbf{x_1 = h}$, $\mathbf{x_2 = c}$, $\mathbf{y_1 = \sigma(h)}$, $\mathbf{y_2 = \sigma(c)}$; $\lvert y - \sigma(j) \rvert > \varepsilon_{fine}\, \sigma(j)$?	No	Add h to grid Halve c-a panel
10	c k a	$\mathbf{x = k}$, $\mathbf{x_1 = c}$, $\mathbf{x_2 = a}$, $\mathbf{y_1 = \sigma(c)}$, $\mathbf{y_2 = \sigma(a)}$;$\lvert y - \sigma(k) \rvert > \varepsilon_{fine}\, \sigma(k)$?	No	Add c and a to grid

Figure 2.4 Halving-iteration scheme

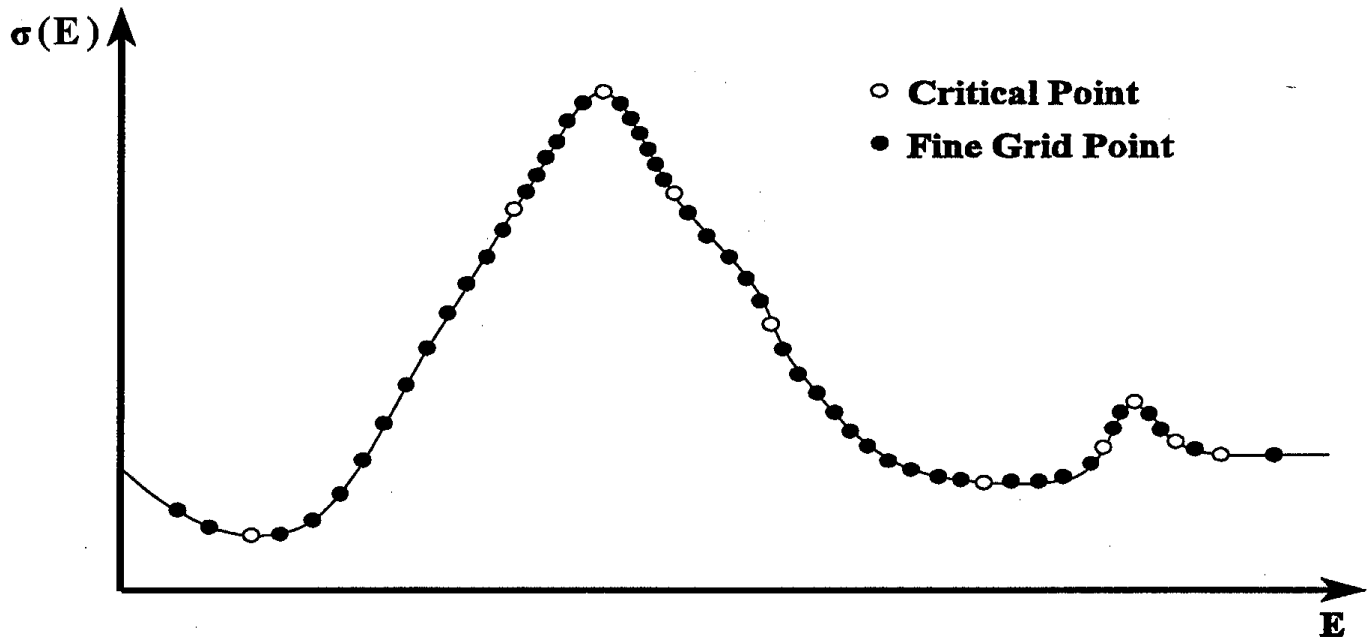


Figure 2.5 Fine-grid structure with critical points

Step 6: Collapse fine grid to auxiliary grid

Once the halving-iteration scheme is complete, the auxiliary grid is collapsed from the fine energy mesh. Note that the fine grid for the entire resolved-resonance region can become very large for certain isotopes/nuclides. As a result, a fine grid is only used during the energy-mesh construction for a decade interval. After the fine grid is generated for a decade, the user or auxiliary grid is collapsed from the fine energy mesh and stored for later use.

Initially, the critical points from the fine-mesh calculation are transferred to the auxiliary grid as shown in Figure 2.6. To complete the auxiliary-grid construction, **each** fine-grid cross-section value, $\sigma^{fine}(E)$, between critical points is linearly estimated using the critical points in the auxiliary grid. Fine-grid points are added to the auxiliary grid between critical points until **each** fine-grid cross-section value can be calculated within the desired convergence tolerance, ε_{aux}. Figure 2.7 illustrates the auxiliary-grid construction between two critical points for an arbitrary cross-section function. As shown in Figure 2.7, the cross-section value at energy E_1 is linearly interpolated using the critical points E_{c1} and E_{c2}. Since the interpolated value is not within the auxiliary-grid convergence tolerance, ε_{aux}, the panel between points E_{c1} and E_{c2} is divided in two, and the energy closest to the midpoint is selected (i.e., energy E_6). Subsequently, the cross-section value at energy E_1 is linearly interpolated using the points E_{c1} and E_6. As shown in Figure 2.7, the interpolated cross-section value at energy E_1 is not within the convergence criterion, and the panel is subdivided again. Following the panel division, energy point E_3 is selected as the top value in the interpolation panel. Using energy points E_{c1} and E_3, the cross-section values at energies E_1 and E_2 are successfully interpolated within the auxiliary-grid convergence tolerance. As a result, energy point E_3 is added to the auxiliary grid. The cross-section values that correspond to the remaining energies between E_3 and E_{c2} are linearly estimated with these two grid points. As shown in Figure 2.7, the cross-section values that correspond to energies E_5 and E_{11} can be estimated by linear interpolation within the ε_{aux} convergence tolerance using the points E_3 and E_{c2}. Therefore, the resulting auxiliary grid is defined by energies E_1, E_3 and E_{c2}.

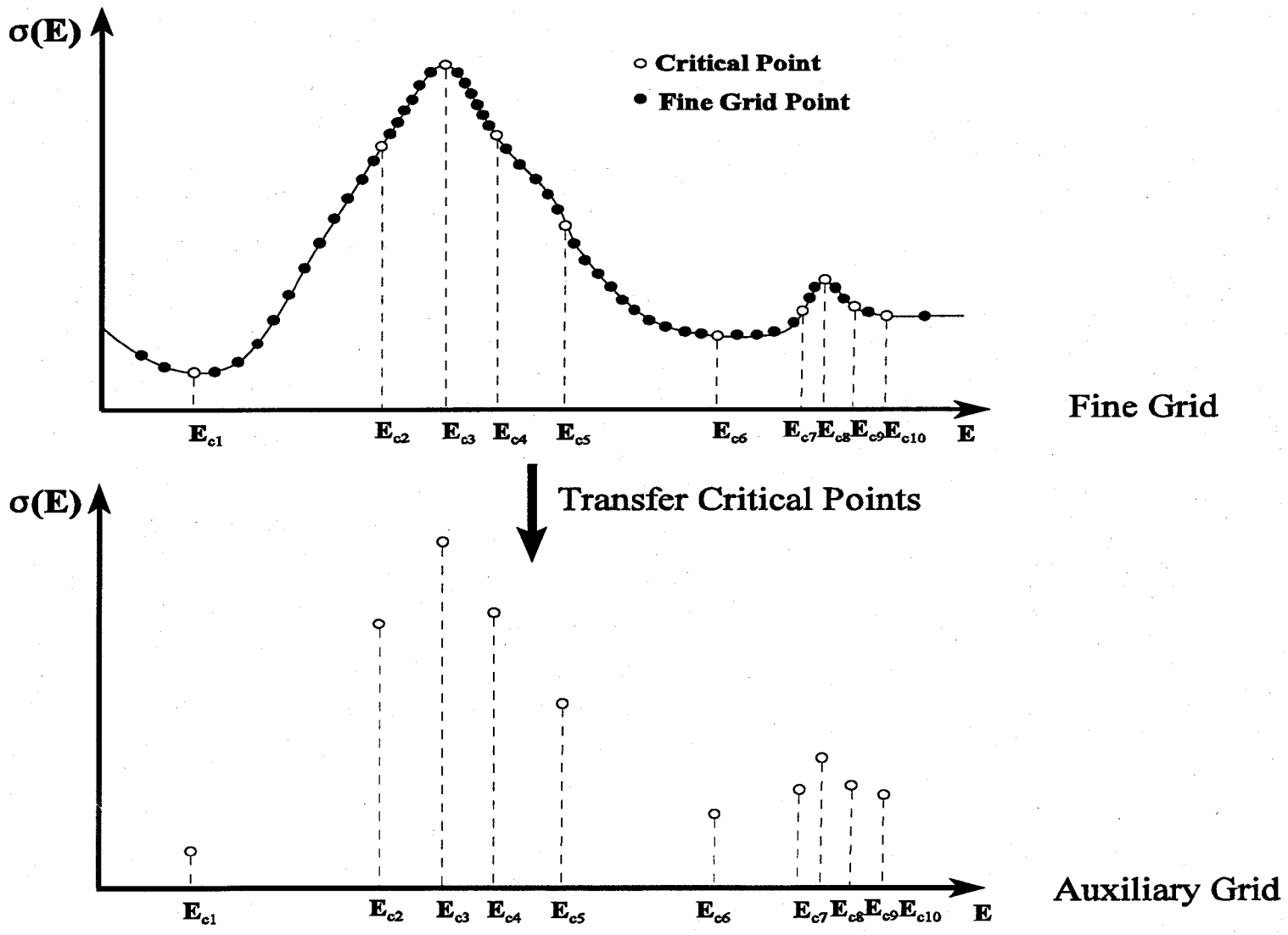


Figure 2.6 Transfer critical points to auxiliary grid

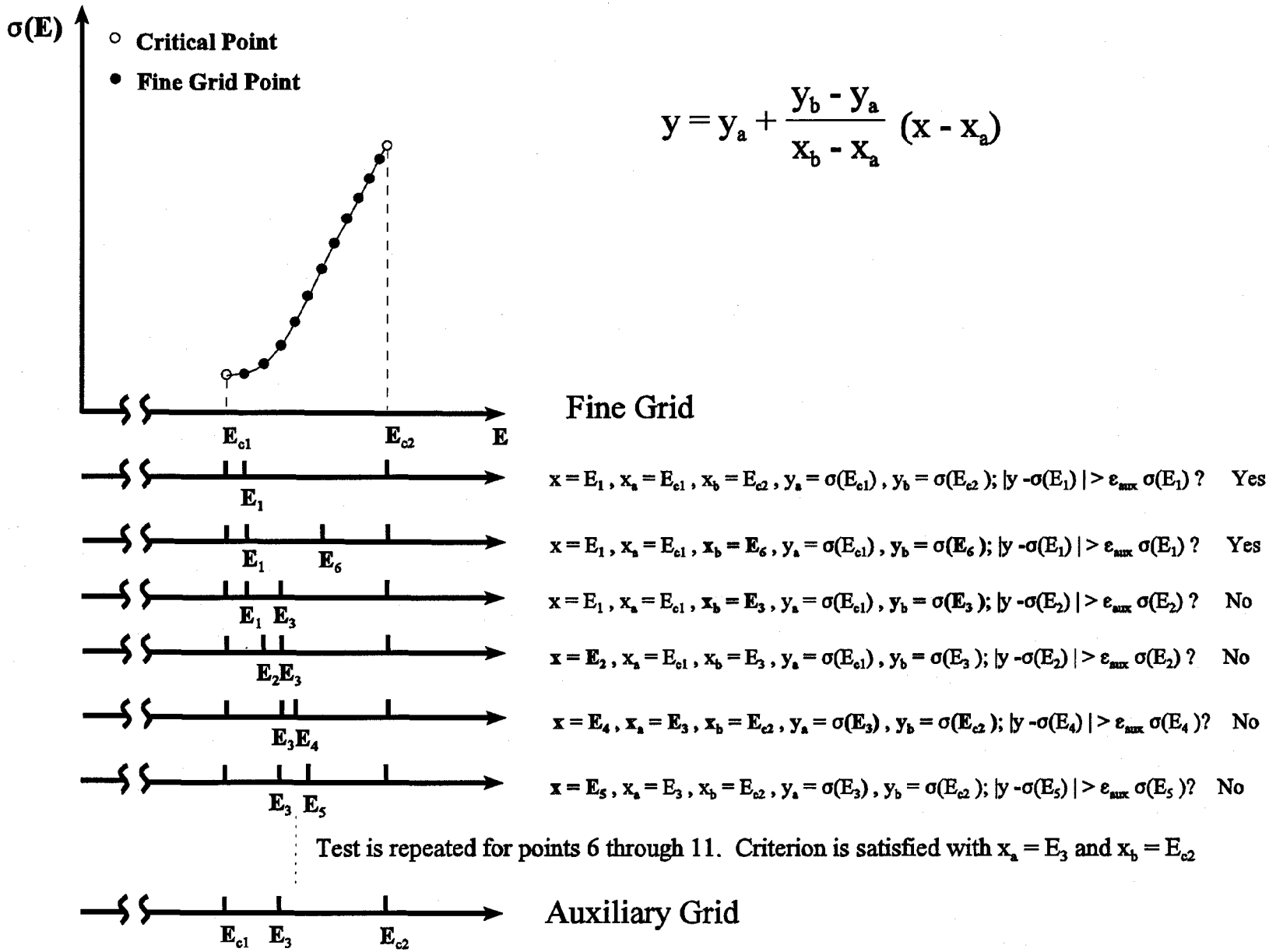


Figure 2.7 Collapsing fine-energy grid to auxiliary grid

2.2.2 Unresolved-Resonance Region

In the unresolved-resonance region, the energy mesh is determined based on the unresolved-resonance parameters that are provided in the evaluation of interest. In particular, the data can be categorized as follows:

Table 2.2 Categorization of unresolved resonance parameter formats

Case	Resonance flags	Description
1.	LFW = 0	No fission widths are provided
	LRF = 1	All other parameters are constant
2.	LFW = 1	Fission widths are provided as a function of energy
	LRF =1	All other parameters are constant
3.	LFW = 0/1	No fission widths are provided / fission widths are provided as a function of energy
	LRF = 2	All other parameters vary as a function of energy

For the second and third types of data, the ENDF formats provide energies for tabulating the energy-dependent widths, and POLIDENT simply uses the tabulated data points for the energy grid in the unresolved-resonance region. With regard to the first category of data (i.e., LFW = 0 and LRF = 1), all the resonance parameters are energy-independent. As a result, no energy-grid values are provided in the ENDF data. Since POLIDENT needs to calculate an energy-dependent cross-section function in the unresolved-resonance region, an energy grid is generated as follows:

$$E(i) = E_\ell e^{(i-1)\Delta E} , \tag{2.94}$$

where

$$\Delta E = \frac{\ln\left(\frac{E_h}{E_\ell}\right)}{N-1} , \tag{2.95}$$

and

i = energy-grid index,
N = number of energy intervals (default value = 100),
E_ℓ = lower energy value for unresolved-resonance region,
E_h = upper boundary for unresolved-resonance region,
ΔE = energy increment.

2.3 Cross-Section-Data Construction

The preceding sections provide the mathematical basis for generating the cross-section data as a function of energy in the resolved- and unresolved-resonance regions. Before the data can be used by another AMPX module, the different functional representations for each energy range must be combined to form a single cross-section function.

In particular, the cross-section data from the resolved and unresolved resonance regions must be combined to form a single function in the resonance region. Subsequently, the background cross-section data from File 3 are combined with the cross-section information from the resonance region to form a single cross-section function that is defined throughout the energy range for the evaluation of interest. In order to combine different energy-dependent functions, the processing code must treat the discontinuities at the energy boundaries as well as the functional overlap that may exist for specific energy ranges.

Prior to describing the combining procedures that are used in POLIDENT, the different ENDF/B interpolation codes must be discussed. A complete description of the ENDF/B interpolation laws is provided in Ref. 1., and the five ENDF/B interpolation laws are summarized in Table 2.3.

Table 2.3 ENDF/B interpolation laws

Interpolation code	Interpolation type	
1	Histogram	y is constant in x
2	Linear-linear	y is linear in x
3	Linear-log	y is linear in ln(x)
4	Log-linear	ln(y) is linear in x
5	Log-log	ln(y) is linear in ln(x)

An ENDF/B energy-dependent function may be divided into one or more regions with a different interpolation law for each region. In particular, an ENDF/B function is defined with N2 points and N1 interpolation regions. Four arrays are associated with each function. The energy and cross-section values are provided in two arrays with a length of N2. The remaining two arrays identify the location of each region (i.e., position index within the energy array) and type of interpolation law to use for each region. In particular, for each function, there is an NBT array and INT array that have a length of N1. The INT array provides the interpolation code that is to be used in the mth interpolation region (i.e., $1 \leq m \leq N1$), and the NBT array provides the position index in the energy array separating the mth and $(m+1)^{th}$ interpolation region (i.e., NBT(m) = n; $1 < n \leq N2$). For example, consider an arbitrary function with 100 points (N2 = 100) and 3 interpolation regions (N1 = 3) as described in Table 2.4. For the arbitrary ENDF/B function, the first interpolation region is defined for energy point 1 through energy point 27, and the corresponding interpolation law as defined in Table 2.3 for the first region is log-log interpolation (i.e., INT(1) = 5). Likewise, log-linear interpolation is specified for the second region, and linear-linear interpolation is specified for the third region.

Table 2.4 Example ENDF/B function

Energy and cross-section arrays			Interpolation arrays		
Point index (n)	E(n) (eV)	σ(n) (barns)	Region index (m)	NBT(m)	INT(m)
27	5.5×10^2	70.4	1	27	5
67	1.0×10^4	25.2	2	67	4
100	2.0×10^7	3.5	3	100	2

Regarding POLIDENT, The task of combining two ENDF/B arrays or functions to form a final function is controlled by the AMPX library subroutine COMB2, which is identified in the logical program flow in Section 3.4. The following discussion provides a description of the procedures for combining different energy-dependent functions.

2.3.1 Combining Functions without Discontinuities

Consider two functions *A* and *B* that are defined over the energy intervals [$E_{A,1}$, $E_{A,NA}$] and [$E_{B,1}$, $E_{B,NB}$], respectively. The objective is to combine *A* and *B* to form a new function *C* between energies E_{lo} and E_{hi} ($E_{lo} < E_{hi}$). For example, *A* could represent the cross-section function that is constructed from the resonance parameters (i.e., from ENDF File 2) in the resonance region, and *B* could represent the background cross-section data (i.e., from ENDF File 3) that are to be combined with function *A*. With regard to the combining routines in the AMPX library, the intervals for *A* and *B* can lie completely within E_{lo} and E_{hi}, or one or both functions may only be partially within [E_{lo}, E_{hi}]. For simplicity, assume that the two functions do not have discontinuities. Note that a discussion for the treatment of discontinuities is provided in Section 2.3.2. A schematic diagram of the arbitrary functions *A* and *B* is provided in Figure 2.8.

For the simplest case, the last energy point of one function may correspond to the first energy point of the second function. For example, $E_{A,NA}$ may be equal to $E_{B,1}$, or $E_{B,NB}$ may be equal to $E_{A,1}$. In either case, the two functions overlap at a single point, and the new function is created by adding the two functions together. If $E_{A,NA}$ is equal to $E_{B,1}$, the final function will be defined over the interval [$E_{A,1}$, $E_{B,NB}$], and the point of overlap is treated by discarding the first point of *B*, as shown in Figure 2.9. If $E_{B,NB}$ is equal to $E_{A,1}$, the final function will be defined over the interval [$E_{B,1}$, $E_{A,NA}$], and the point of overlap is treated by discarding the first point of *A*.

If *A* and *B* overlap at more than one point within [E_{lo}, E_{hi}], the combining procedures become more complex. Initially, the two functions are examined for discontinuities; however, as noted previously, *A* and *B* are assumed to be free of discontinuities for the present discussion. Subsequently, the interpolation regions are examined to ensure that the two functions conform to the ENDF/B laws. If problems are encountered with either function, POLIDENT prints a warning message but does not stop execution. Therefore, the output file should always be examined for possible problems in the cross-section function.

In order to combine *A* and *B* between E_{lo} and E_{hi}, a search is performed in the energy arrays of both functions to find the first energy point within [E_{lo}, E_{hi}]. The procedure for combining two functions is best demonstrated by an arbitrary example, and the following discussion demonstrates the combining procedures for constructing the new function *C* by combing *A* and *B*. A schematic diagram of the arbitrary functions *A* and *B*, which overlap within [E_{lo}, E_{hi}], is provided in Figure 2.10.

For the present discussion, *i* and *j* are the indices of the energy points in *A* and *B*, respectively. As shown in Figure 2.10, the first energy points within [E_{lo}, E_{hi}] for *A* and *B* are $E_{A,i}$ and $E_{B,j}$, respectively. Therefore, the first interpolation panel for *A* is ($E_{A,i-1}$, $E_{A,i}$), and the first interpolation panel for *B* is ($E_{B,j-1}$, $E_{B,j}$). Typically, E_{lo} and E_{hi} will correspond to energy points in one or both functions to be combined; however, for illustrative purposes, E_{lo} does not correspond to an energy value within either function. As shown in Figure 2.10, E_{lo} is within the first interpolation panel for both functions, and $E_{A,i}$ is less than $E_{B,j}$. Initially, the first panel of the new function *C* will be constructed between E_{lo} and $E_{A,i}$.

As noted previously, the AMPX library routine COMB2 controls the combining process for the two functions. Once the first panel for the new function is identified, COMB2 calls the AMPX library routine COMBI2 to

construct the new function within the first panel. Initially, COMBI2 is called with the panel between E_{lo} and $E_{A,i}$ and the first interpolation panels for *A* and *B*. Also COMBI2 is called with the AMPX library function that defines how to combine the *A* and *B*.

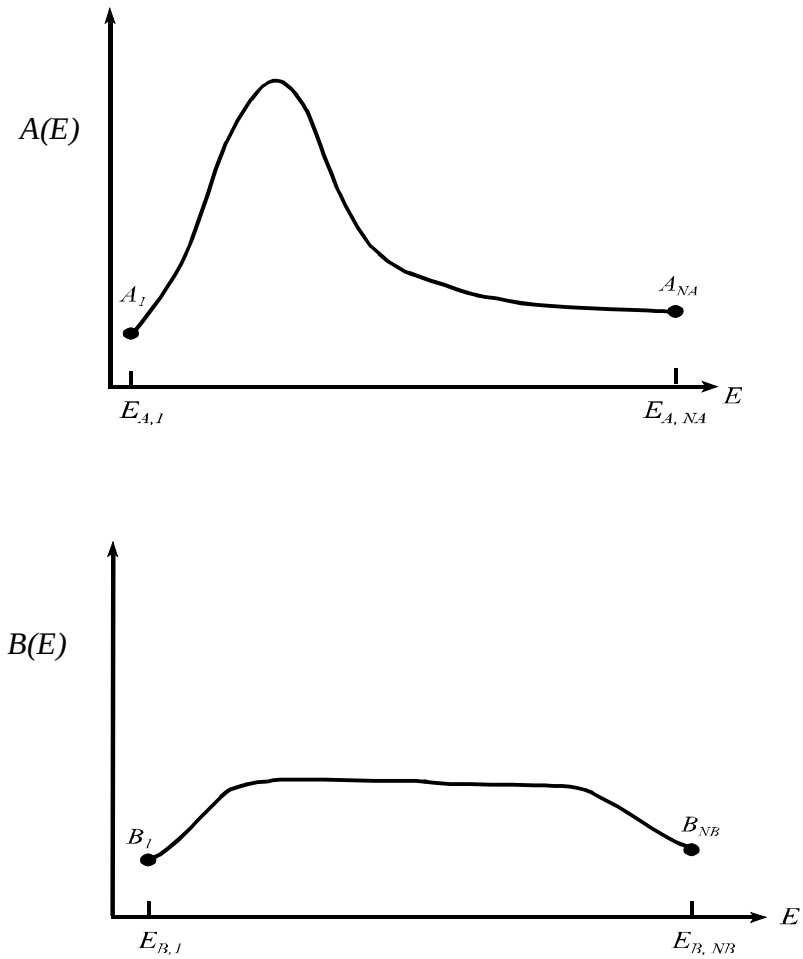


Figure 2.8 Two arbitrary energy-dependent functions to be combined

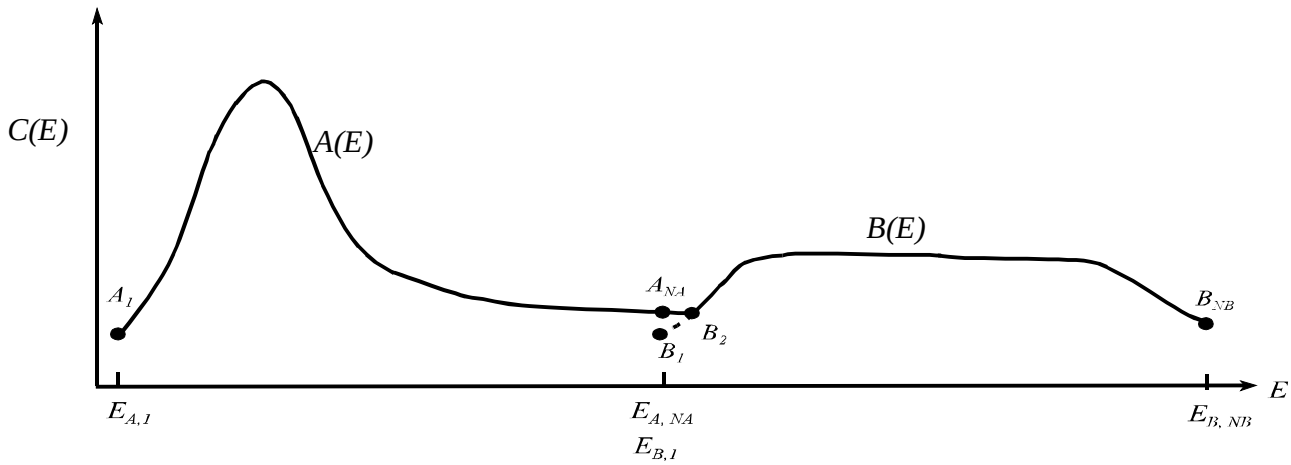


Figure 2.9 Arbitrary example for combining two functions that overlap at a single energy point

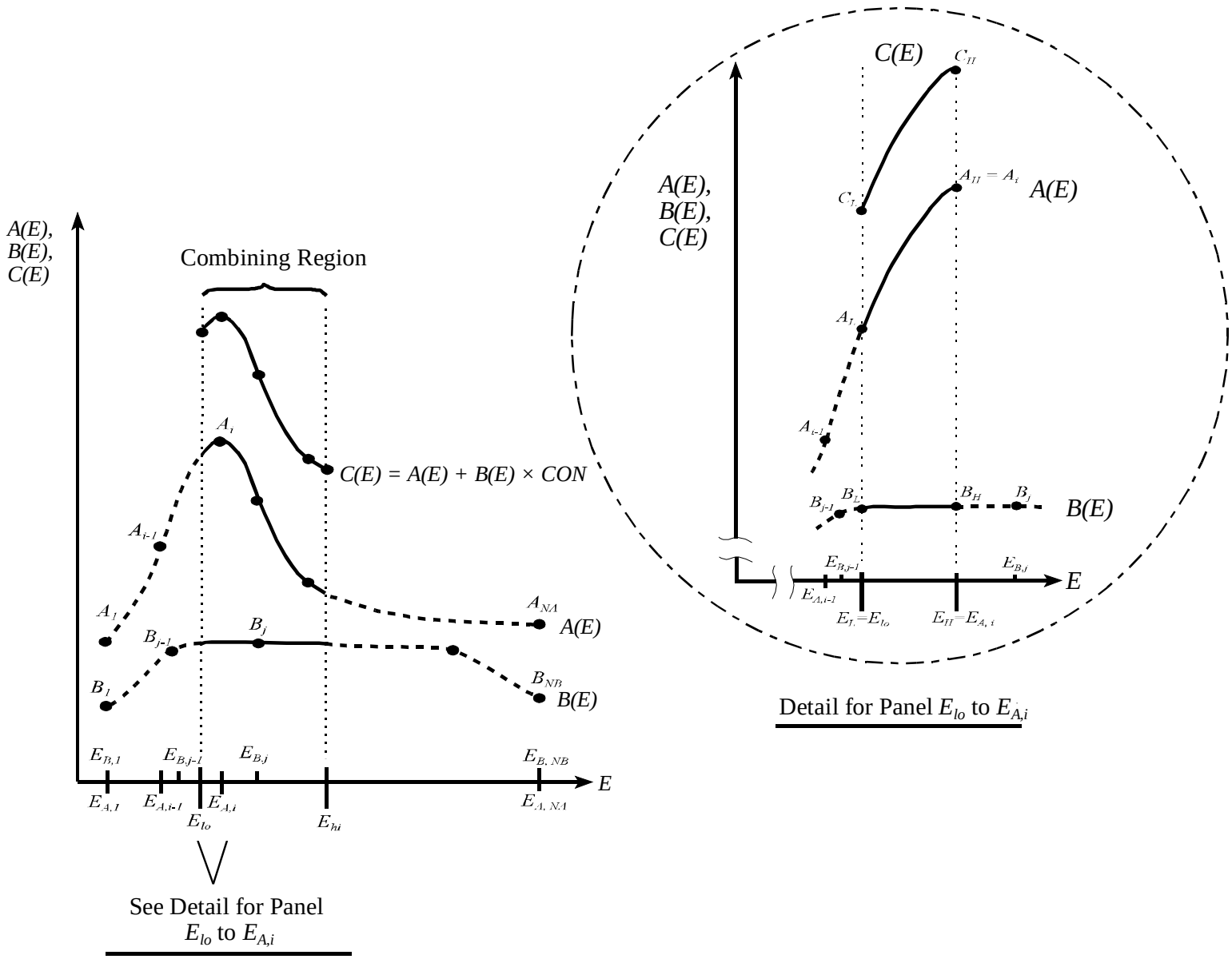


Figure 2.10 Arbitrary example for combining two functions with overlapping energy regions

Within the AMPX library, there are four procedures that can be used to combine two energy-dependent functions, and the available operations are identified in Table 2.5. The operation name in Table 2.5 is the AMPX library name specified in the call to COMBI2. In POLIDENT, the functions are combined using the ADDIT operation.

Table 2.5 Combining operations available in AMPX

Operation name	Description	Operation (*CON* = constant)
ADDIT	Add functions *A* and *B*	$C = A + B \times CON$
SUBIT	Subtract functions *A* and *B*	$C = A - B \times CON$
MULTIT	Multiply functions *A* and *B*	$C = A \times B \times CON$
DIVIT	Divide functions *A* and *B*	$C = A / (B \times CON)$

In the call to COMBI2, the endpoints of the interpolation panel for the new function are passed as E_L and E_H. For the first interpolation panel in the new function, E_L is set equal to E_{lo}, and E_H is set equal to $E_{A,i}$. A schematic diagram for the interpolation panel [E_{lo}, $E_{A,i}$] is provided in the detailed portion of Figure 2.10. Prior to combining *A* and *B*, the values at E_L and E_H are obtained by interpolating in *A* and *B* in accordance with the appropriate ENDF/B interpolation law. As shown in Figure 2.10, A_L and A_H are the interpolated values of *A* at E_L and E_H, respectively. If the interpolation law is linear-linear, the values of A_L and A_H are obtained as follows:

$$A_L = A_{i-1} + \frac{A_i - A_{i-1}}{E_{A,i} - E_{A,i-1}} (E_L - E_{A,i-1}) , \tag{2.96}$$

$$A_H = A_{i-1} + \frac{A_i - A_{i-1}}{E_{A,i} - E_{A,i-1}} (E_H - E_{A,i-1}) . \tag{2.97}$$

Since E_H is equal to $E_{A,i}$, the value for A_H is equal to A_i. Likewise, if the interpolation law is linear-linear, B_L and B_H are the interpolated values of *B* at E_L and E_H, respectively:

$$B_L = B_{j-1} + \frac{B_j - B_{j-1}}{E_{B,j} - E_{B,j-1}} (E_L - E_{B,j-1}) , \tag{2.98}$$

$$B_H = B_{j-1} + \frac{B_j - B_{j-1}}{E_{B,j} - E_{B,j-1}} (E_H - E_{B,j-1}) . \tag{2.99}$$

The values of the new function *C* at E_L and E_H, are obtained using ADDIT:

$$C_L = A_L + B_L \times CON , \tag{2.100}$$

$$C_H = A_H + B_H \times CON . \tag{2.101}$$

In POLIDENT, the combining procedure is performed with *CON* set to 1.

Before the interpolation panel [E_L, E_H] is accepted for the new function, the panel is tested to ensure that intermediate values within the panel can be interpolated to within a specified tolerance *EPS*. In particular, COMBI2 is called with the *EPS* tolerance for combining the two functions. For example, a value of 0.0001 is used to produce a function that is represented to a precision of 0.01%. To perform the test, the [E_L, E_H] panel is halved to obtain the midpoint E_M:

$$E_M = \frac{1}{2}(E_L + E_H) \ . \tag{2.102}$$

A detailed diagram of the [E_L, E_H] panel is provided in Figure 2.11. The values at the panel midpoint, E_M, are obtained for *A*, *B* and *C*. If the interpolation law is linear-linear, Equations (2.103) and (2.104) are used to calculate $A(E_M)$ and $B(E_M)$, respectively:

$$A_M = A(E_M) = A_{i-1} + \frac{A_i - A_{i-1}}{E_{A,i} - E_{A,i-1}}(E_M - E_{A,i-1}) \tag{2.103}$$

and

$$B_M = \ = B(E_M) = B_{j-1} + \frac{B_j - B_{j-1}}{E_{B,j} - E_{B,j-1}}(E_M - E_{B,j-1}) \ . \tag{2.104}$$

The value of *C* at E_M is C_M is given by the following equation:

$$C_M = A_M + B_M \times CON \ . \tag{2.105}$$

The testing procedure assumes that the "exact" functional value of *C* at E_M is C_M. In order to test the interpolation panel, the value at E_M is also obtained by interpolating in *C* between E_L and E_H. If linear-linear interpolation is used, the value of *C* at E_M is estimated as follows:

$$C_{test} = C_{test}(E_M) = C_L + \frac{C_H - C_L}{E_H - E_L}(E_M - E_L) \ . \tag{2.106}$$

The interpolation panel is accepted if the following criterion is satisfied:

$$\left| \frac{C_{test} - C_M}{C_M} \right| < EPS \ . \tag{2.107}$$

If the convergence test as defined by Equation (2.107) is not satisfied at E_M, the combining routines have the capability to try different ENDF/B interpolation procedures; however, by default, POLIDENT only tries to construct the function using linear-linear interpolation. If the convergence test is not satisfied, the [E_L , E_H], and the testing procedure as defined in Equations (2.100) through (2.107) is repeated for the smaller panel (i.e., [E_L , E_M]) as shown in Figure 2.11. Each time the convergence criterion in Equation (2.107) is not satisfied, the panel is halved and the testing procedure is repeated. For the [E_L , E_M] panel in Figure 2.11, the convergence test is satisfied, and the energy point E_M is added to the grid. Once the [E_L , E_M] panel is successfully converged, the testing procedure is repeated for the [E_M , E_H] panel. As shown in Figure 2.11, the convergence criterion is satisfied at the midpoint of the [E_M , E_H] panel, and the points that are needed to define the original panel (i.e., [E_L , E_H]) are E_L , E_M and E_H.

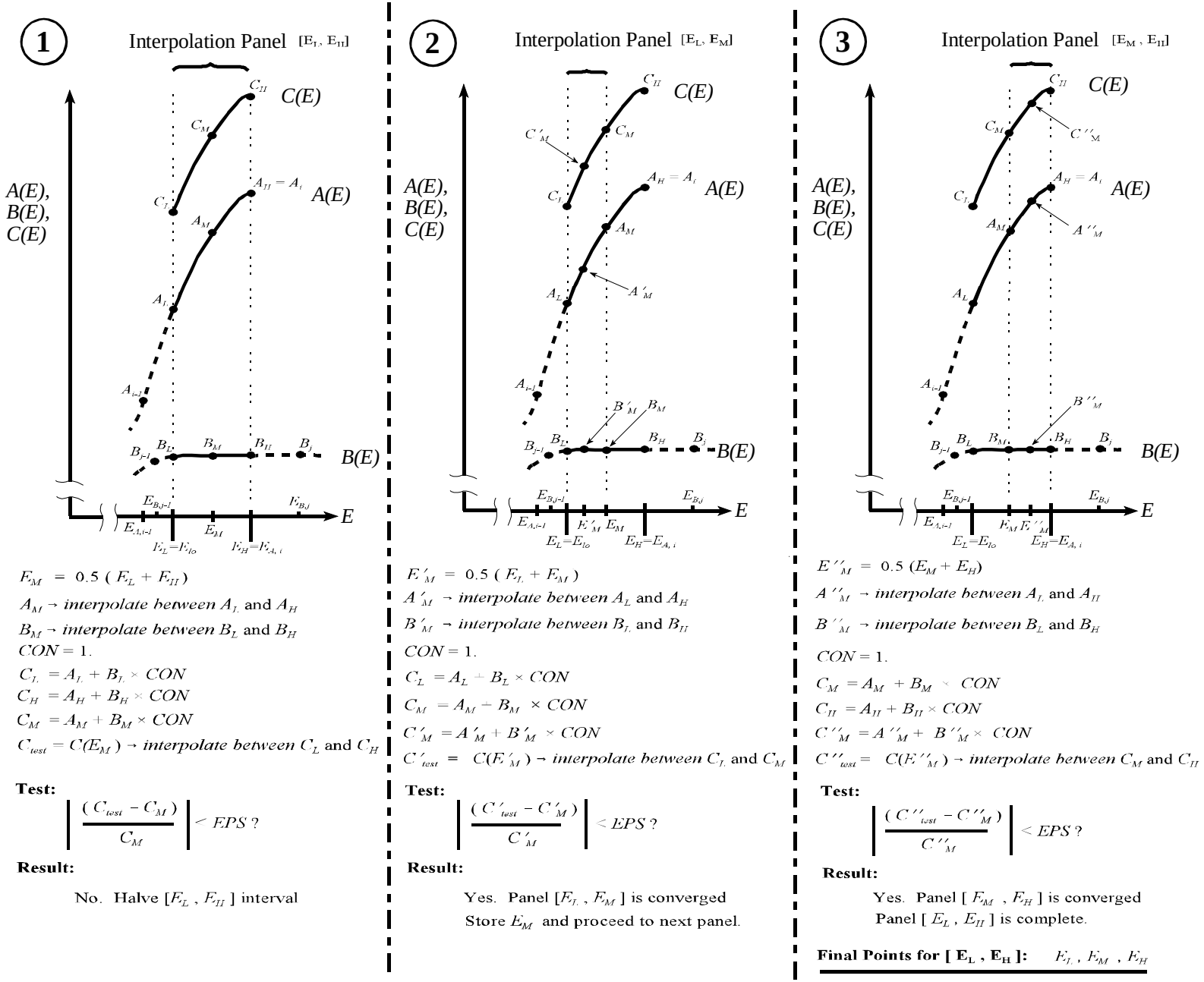


Figure 2.11 Detail for processing the interpolation region during the combining of two arbitrary functions with overlapping energy regions

2.3.2 Combining Functions with Discontinuities

The procedures for combining two functions, *A* and *B*, without discontinuities are provided in the previous section. If *A* and *B* overlap at more than one point in the combining region, [E_{lo} , E_{hi}], the two functions are examined for discontinuities, and the locations of the discontinuities within the energy array of each function are stored for each function. For example, if $E_{A,i-1}$ is equal to $E_{A,i}$ and $A(E_{A,\,i-1})$ is not equal to $A(E_{A,\,i})$, the value of *i* is stored as the location of the discontinuity in the *A* function. The maximum number of discontinuities in a function is limited to 25 by the AMPX library routine COMB2. If the discontinuity limit is exceeded, an error message is printed and the program stops execution. Once the discontinuities are identified, the interpolation regions are examined to ensure that the two functions conform to the ENDF/B laws. As noted in Section 2.3.1, if problems are encountered with either function, POLIDENT prints a warning message but does not stop execution.

As in the previous section, *i* and *j* are the indices of the energy points in *A* and *B*, respectively. In order to combine *A* and *B* between E_{lo} and E_{hi}, a search is performed in the energy arrays of both functions to find the first energy point within [E_{lo}, E_{hi}]. For the present discussion, *A* and *B* are energy-dependent functions, and either one or both functions can have one or more discontinuities. With the exception of the points of discontinuity, the procedures for combining *A* and *B* are analogous to the procedures that are presented in Section 2.3.1. In other words, *A* and *B* are combined using the procedures that are presented in Section 2.3.1 until a point of discontinuity is reached in one or both functions. If a panel in either function has a discontinuity, the procedures that are presented in this section are used to process the discontinuity. For discussion purposes in this section, the energy points with a possible discontinuity in the combining region for *A* and *B* are $E_{A,i}$ and $E_{B,j}$, respectively. Therefore, the corresponding interpolation panel with a discontinuity in *A* is [$E_{A,i-1}$, $E_{A,i}$], and the corresponding interpolation panel for *B* is [$E_{B,j-1}$, $E_{B,j}$]. After processing the discontinuity, the combining procedures from Section 2.3.1 are used to combine *A* and *B* until the next point of discontinuity is reached. Since the procedures for combining functions without discontinuities is addressed in Section 2.3.1, this section only describes the specific steps for processing the discontinuity that may be present in one or both functions.

Recall that the locations of discontinuities within each function are identified and stored prior to combining the two functions. Consequently, $E_{A,i}$ is examined to see if the energy point corresponds to a point of discontinuity in *A*. Likewise, $E_{B,j}$ is examined to see if the energy point corresponds to a point of discontinuity in *B*. There are seven different cases that the AMPX combining routines must consider before processing a discontinuity. Each possible case is identified in Table 2.6. For each case in Table 2.6, there is a corresponding schematic diagram presented in Figures 2.12 and 2.13 to aide in the visualization of the possible discontinuities.

If both the ith and j^{th} panels of *A* and *B*, respectively, correspond to a discontinuity in *A* and *B*, the case corresponds to Case 1 of Table 2.6 and Figure 2.12. The AMPX library routine DISCON is used to process the discontinuity. For Case 1, both *A* and *B* have a discontinuity at $E_{A,i}$ and $E_{B,j}$, respectively, and $E_{A,i}$ is equal to $E_{B,j}$. Note that the KTYPE flag in Table 2.6 is equal to 0 for Case 1. The KTYPE=0 flag indicates that both functions have a discontinuity at the same energy point. As a result, the new function must be obtained by combining *A* and *B* at the point of discontinuity (i.e., $E_{A,i} = E_{B,j}$), which is the second point in the panel for *A* and *B*. The combined functional value at $E_{A,i}$ is obtained using the AMPX library function ADDIT:

$$C(E_{A,i}) = A(E_{A,i}) + B(E_{A,i}) \times CON \ , \tag{2.108}$$

In POLIDENT, the combining procedure is performed with *CON* set to 1.

For Case 2 of Table 2.6 and Figure 2.12, the first point of *B* is not a discontinuity in *B*; however, $E_{B,1}$ corresponds to the point of discontinuity in *A* (i.e., $E_{A,i} = E_{B,1}$). As noted in the last column of Table 2.6, a pseudo point, ($E_{B,1}$, 0.), is defined for the panel in *B*. As a result, DISCON is called with the set of points that are denoted in the last

column of Table 2.6 for Case 2. In addition, the KTYPE=0 flag indicates that both *A* and *B* have a "discontinuity" at $E_{A,i} = E_{B,1}$. Note that the pseudo point for Case 2 is passed to DISCON as the first point in the panel for the *B* function. Therefore, DISCON does not use the pseudo point in *B*, and *A* and *B* are combined at the second point in the panel (i.e., $(E_{A,i}, A_i)$ and $(E_{B,1}, B_1)$) using Equation 2.108.

Regarding Case 3 of Table 2.6 and Figure 2.12, the last point of *B* is not a discontinuity in *B*; however, $E_{B,NB}$ corresponds to the point of discontinuity in *A* (i.e., $E_{A,i} = E_{B,NB}$). The procedure for Case 3 is analogous to Case 2; however, the pseudo point is passed to DISCON as the second point in the panel for the *B* function. As a result, *A* and *B* are combined at the second point in the panel using Equation 2.108. Note that pseudo point corresponds to $B(E_{B,NB} = E_{A,i}) = 0.$, and the value of *C* at $E_{A,i}$ is $A(E_{A,i})$.

In Case 4 of Table 2.6 and Figure 2.12, a discontinuity is only present in the ith panel of *A* at $E_{A,i}$, and the jth panel of *B* is free of any discontinuity. In order to combine *A* and *B*, DISCON is called with the set of points that are denoted in the last column of Table 2.6 for Case 4 and the points of the jth panel in *B*. In addition, the KTYPE=1 indicates that the discontinuity is in the first function (i.e., *A*). When the discontinuity is only in *A*, the value of *B* at $E_{A,i}$ is obtained by interpolating in *B* using the appropriate ENDF/B interpolation law. Subsequently, the functional value of *C* at $E_{A,i}$ is obtained using Equation 2.108.

The remaining discussion in this section addresses Cases 5 through 7 of Table 2.6 and Figure 2.13. The procedure for Case 5 in Table 2.6 is analogous to the procedure for Case 4; however, the discontinuity is only present in the jth panel of *B* at $E_{B,j}$, and the ith panel of *A* is free of any discontinuity. In order to combine *A* and *B*, DISCON is called with the set of points that are denoted in the last column of Table 2.6 for Case 5 and the points of the ith panel in *A*. In addition, the KTYPE=2 indicates that the discontinuity is in the second function (i.e., *B*). When the discontinuity is only in *B*, the value of *A* at $E_{B,j}$ is obtained by interpolating in *A* using the appropriate ENDF/B interpolation law. Subsequently, the functional value of *C* at $E_{B,j}$ is obtained as follows:

$$C(E_{B,j}) = A(E_{B,j}) + B(E_{B,j}) \times CON \ . \tag{2.109}$$

The remaining cases in Table 2.6 are similar to Cases 2 and 3, except that the discontinuity is in *B*. For Case 6, the first point of *A* is not a discontinuity in *A*; however, $E_{A,1}$ corresponds to the point of discontinuity in *B* (i.e., $E_{B,j} = E_{A,1}$). As noted in the last column of Table 2.6 for Case 6, a pseudo point, $(E_{A,1}, 0.)$, is defined for the panel in *A*, and DISCON is called with the set of points that are denoted in the last column of Table 2.6 for Case 6. As noted previously, the KTYPE=0 flag indicates that both *A* and *B* have a "discontinuity" at $E_{B,j} = E_{A,1}$. In order to obtain the final functional value at the point of discontinuity, *A* and *B* are combined at the second point in the panel (i.e., $(E_{A,1}, A_1)$ and $(E_{B,j}, B_j)$) using Equation 2.109.

Regarding Case 7, the last point of *A* is not a discontinuity in *A*; however, $E_{A,NA}$ corresponds to the point of discontinuity in *B* (i.e., $E_{B,j} = E_{A,NA}$). The procedure for Case 7 is analogous to Case 3; however, a pseudo point, $(E_{A,NA}, 0.)$, is defined for the panel in *A*, and the pseudo point is passed to DISCON as the second point in the panel for the *A* function. As a result, *A* and *B* are combined at the second point in the panel using Equation 2.109. Note that the pseudo point corresponds to $A(E_{A,NA} = E_{B,j}) = 0.$, and the value of *C* at $E_{B,j}$ is $B(E_{B,j})$.

Table 2.6 Possible cases for processing discontinuities in energy-dependent functions

Case	Description		Flag (KTYPE)[a]	Points of discontinuity
1	Both functions in their respective panels have a discontinuity.	$E_{A,\,i-1} = E_{A,\,i}$; $A_{i-1} \neq A_i$ $E_{B,\,j-1} = E_{B,\,j}$; $B_{j-1} \neq B_j$	0	$(E_{A,\,i-1}, A_{i-1})$ $(E_{A,\,i}, A_i)$ $(E_{B,\,j-1}, B_{j-1})$ $(E_{B,\,j}, B_j)$
2	Discontinuity in *A*. First energy point in *B* corresponds to discontinuity in *A*. No discontinuity in first panel of *B*.	$E_{A,\,i-1} = E_{A,\,i}$; $A_{i-1} \neq A_i$ $E_{B,1} = E_{A,\,i}$	0	$(E_{A,\,i-1}, A_{i-1})$ $(E_{A,\,i}, A_i)$ $(E_{B,\,1},\ 0.)$[b] $(E_{B,\,1}, B_1)$
3	Discontinuity in *A*. Last energy point in *B* corresponds to discontinuity in *A*. No discontinuity in last panel of *B*.	$E_{A,\,i-1} = E_{A,\,i}$; $A_{i-1} \neq A_i$ $E_{B,NB} = E_{A,\,i}$	0	$(E_{A,\,i-1}, A_{i-1})$ $(E_{A,\,i}, A_i)$ $(E_{B,\,NB}, B_{NB})$ $(E_{B,\,NB},\ 0.)$[b]
4	Discontinuity only in panel for *A*.	$E_{A,\,i-1} = E_{A,\,i}$; $A_{i-1} \neq A_i$	1	$(E_{A,\,i-1}, A_{i-1})$ $(E_{A,\,i}, A_i)$
5	Discontinuity only in panel for *B*.	$E_{B,\,j-1} = E_{B,\,j}$; $B_{j-1} \neq B_j$	2	$(E_{B,\,j-1}, B_{j-1})$ $(E_{B,\,j}, B_j)$
6	Discontinuity in *B*. First energy point in *A* corresponds to discontinuity in *B*. No discontinuity in first panel of *A*.	$E_{B,\,j-1} = E_{B,\,j}$; $B_{j-1} \neq B_j$ $E_{A,1} = E_{B,\,j}$	0	$(E_{A,\,1},\ 0.)$[b] $(E_{A,\,1}, A_1)$ $(E_{B,\,j-1}, B_{j-1})$ $(E_{B,\,j}, B_j)$
7	Discontinuity in *B*. Last energy point in *A* corresponds to discontinuity in *B*. No discontinuity in last panel of *A*.	$E_{B,\,j-1} = E_{B,\,j}$; $B_{j-1} \neq B_j$ $E_{A,\,NA} = E_{B,\,j}$	0	$(E_{A,\,NA}, A_{NA})$ $(E_{A,\,NA},\ 0.)$[b] $(E_{B,\,j-1}, B_{j-1})$ $(E_{B,\,j}, B_j)$

[a]Flag passed to AMPX library routine DISCON to identify the type of discontinuity to process. KTYPE=0 (discontinuity in panel in both functions); KTYPE=1 (discontinuity in panel of first function); and KTYPE=2 (discontinuity in panel of second function).

[b]Pseudo point defined for processing discontinuity with KTYPE=0.

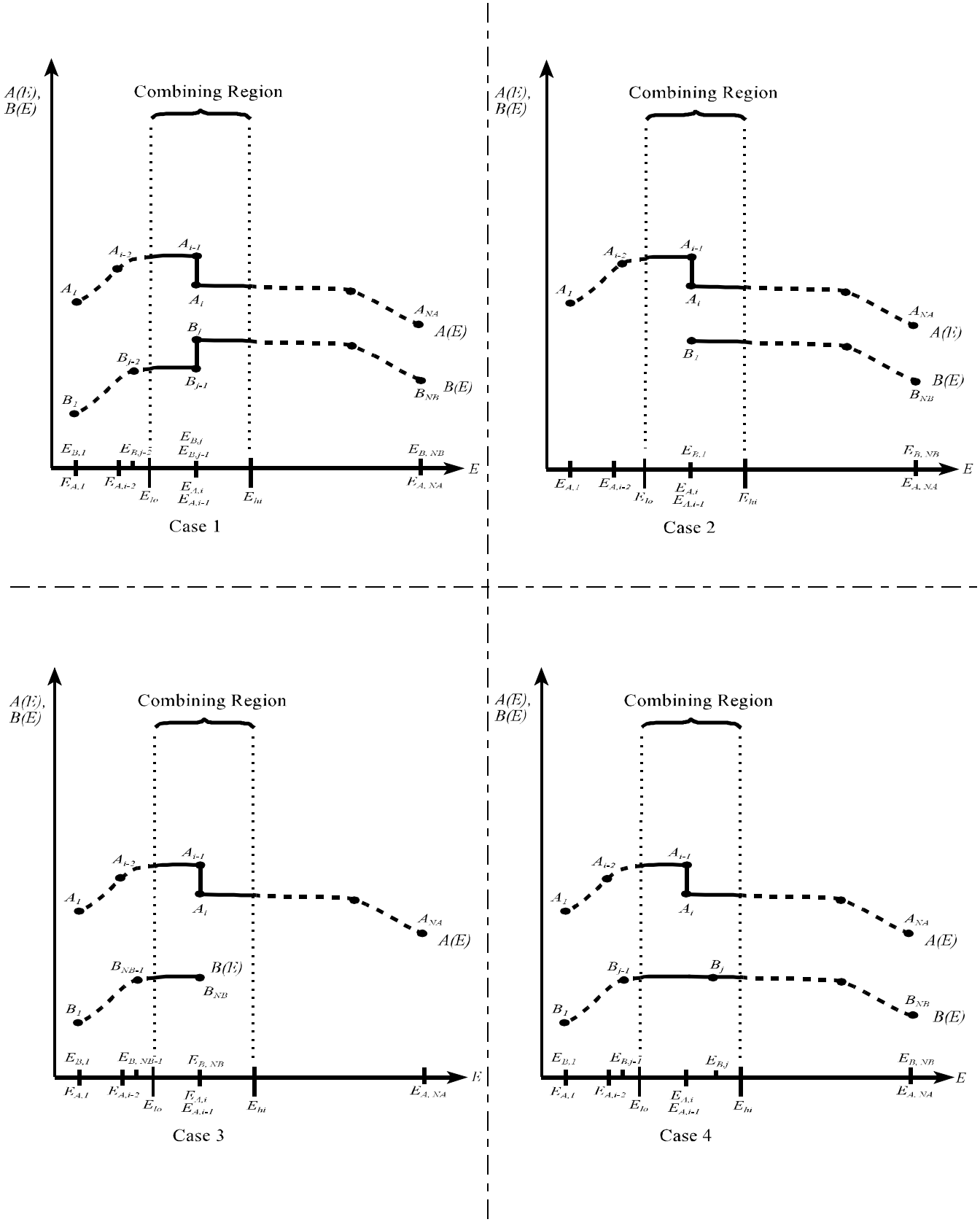


Figure 2.12 Schematic diagrams of discontinuities for cases 1 through 4 of Table 2.6

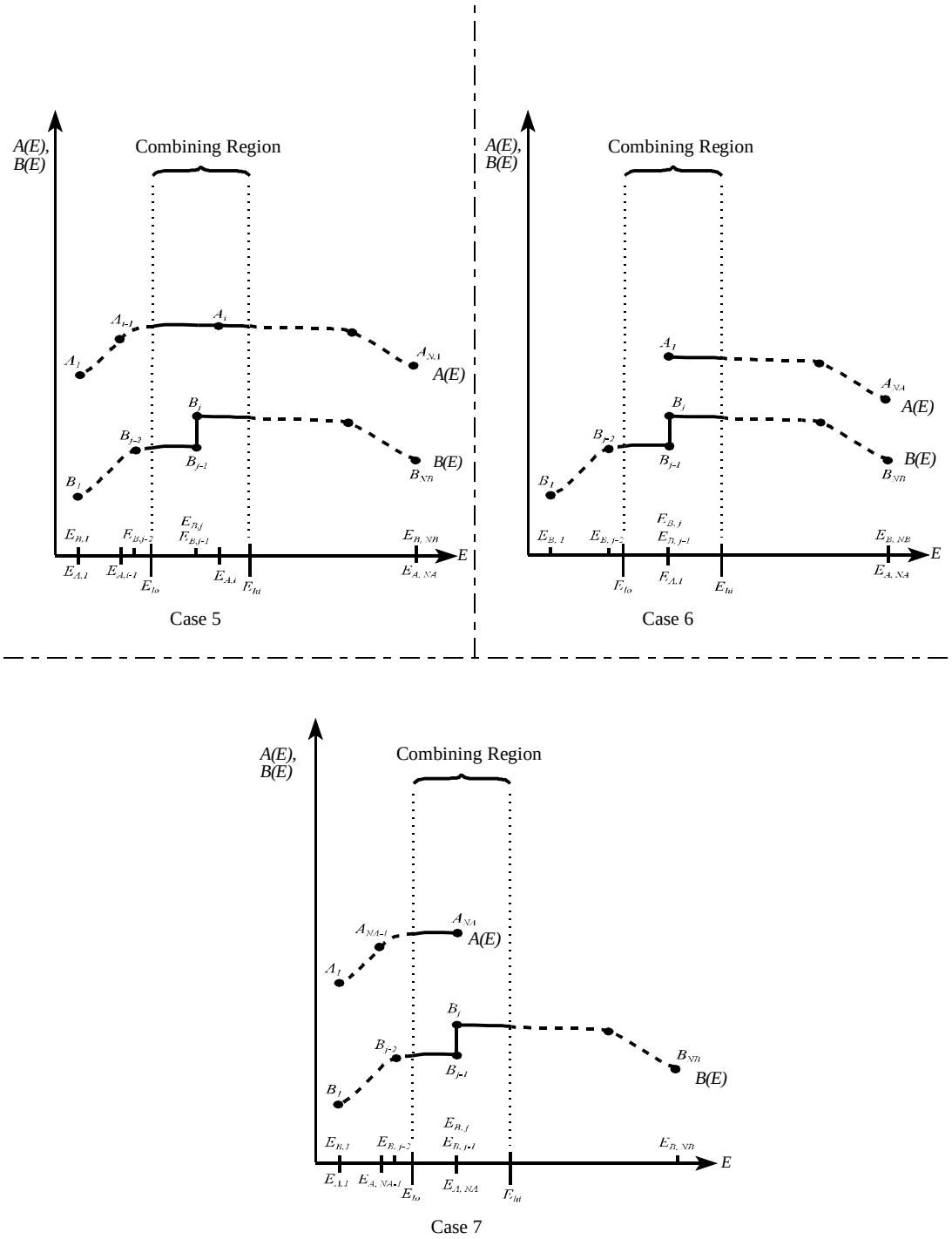


Figure 2.13 Schematic diagrams of discontinuities for cases 5 through 7 of Table 2.6

3 LOGICAL PROGRAM FLOW

The following sections outline the logical program flow of POLIDENT. Because of the number of subroutines involved in the program, the logical flow description is divided into separate sections. Consequently, an overall flow diagram for the program is not provided. Each section describes a portion of the program flow with an abbreviated flow diagram. In the subsequent flow diagrams, the POLIDENT subroutines are shown with the names enclosed by boxes, and the AMPX library routines are depicted without boxes. A brief description of each subroutine is provided with the accompanying flowchart. In each abbreviated flowchart, a POLIDENT subroutine box may also have an arrow indicating additional information. The additional flow logic associated with each arrow is described in greater detail in another section.

3.1 Program Initiation

Figure 3.1 provides a flow diagram for the initiation of POLIDENT.

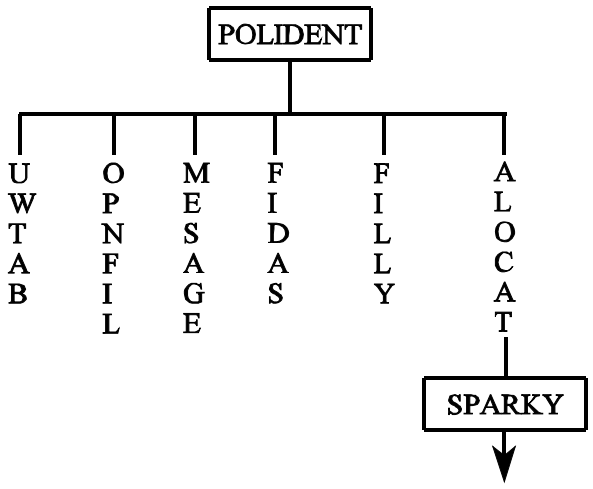


Figure 3.1 Flowchart for program initiation

The function of this portion of the code is to perform the initialization tasks for program execution. In particular, the program reads the first data block of the input file, determines the type of problem to execute and the amount of core needed for the problem. Subsequently, POLIDENT calls the AMPX library routine ALOCAT, which allocates the requested storage space for the problem and calls the controlling subroutine SPARKY.

POLIDENT: The main program that opens the user input file and reads the first block of data for the problem. In addition, POLIDENT calls MESAGE to print the banner page, and SPARKY is called after the required storage space is obtained by ALOCAT.

UWTAB: Library routine that is called to generate a table of the complex-probability integral that will be used to calculate the line-shape functions for the unresolved-resonance treatment.

OPNFIL: Library routine that initializes the input and output logical units as well as the scratch devices. In addition, OPNFIL initializes the cross-section output units LOGP and LOGP1.

MESAGE: Library routine that prints the banner page on the logical output device.

FIDAS: Library routine that reads the FIDO input structure.

FILLY: Library routine that writes a message to the logical output file describing the cross-section output file (i.e., descriptive title, data set name and volume).

ALOCAT: Assembly language program that is called with three arguments. The first argument is a subroutine name, and the second argument is the maximum number of words of storage to be allocated. ALOCAT calls subroutine SPARKY with two arguments, an array name and the corresponding array length. The third argument is prefixed by an asterisk and specifies the statement number to return to if there is insufficient space to execute the problem.

SPARKY: Controlling subroutine for POLIDENT. SPARKY, which is called by ALOCAT, controls the overall flow of POLIDENT and is described in more detail in Section 3.2.

3.2 Overall Program Flow

SPARKY is designed to direct the primary flow of POLIDENT (Figure 3.2). Moreover, SPARKY reads the remaining user-input data for each requested nuclide, sets up the problem, reads the ENDF File 1 information, calls the appropriate subroutine (i.e., RESN) to read and process the ENDF resonance data and combines the File 3 cross-section data with the cross sections from the resonance region. The continuous-energy cross sections are stored in a binary TAB1 format that can be read by other AMPX modules.

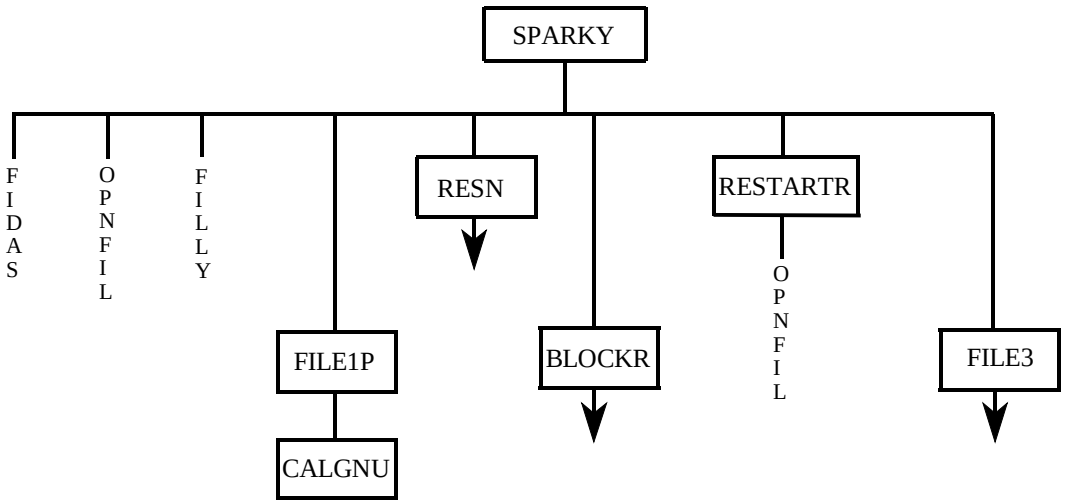


Figure 3.2 Flowchart for overall program flow

SPARKY: Reads the user-input data and prints the appropriate information on the logical-output device. Subsequently, SPARKY calls FILE1P to read the ENDF File 1 directory information. Subroutine RESN is called to read the resonance data from File 2 and to process the data accordingly. The cross sections that are produced from RESN are stored on a scratch device for later retrieval. If the resonance data from a previous problem were processed, the resonance calculations do not need to be repeated. As a result, a restart problem may be specified, and subroutine RESTARTR is called to read the cross sections from the resonance region. Once the cross sections are generated for the resonance region, subroutine BLOCKR constructs the cross sections in the resonance region by combining the cross-section arrays from the resolved- and unresolved-resonance region. Subsequently, subroutine FILE3 combines the cross sections from File 3 with the resonance-region cross sections and treats the discontinuities that may be present at the boundaries of the different regions. Upon completion, the continuous-energy cross sections are stored on LOGP in a binary TAB1 format. A summary of the subroutines called by SPARKY is provided in Table 3.1, and a complete description of each subroutine in Table 3.1 is provided in the following discussion.

Table 3.1 Summary of subroutines called by SPARKY

Subroutine	Function	Condition
FIDAS	Reads the user-input data specified for each nuclide	Always
OPNFIL	Initializes ENDF cross-section files	Always
FILLY	Prints message describing ENDF cross-section file	Always
FILE1P	Reads the ENDF File 1 information	Always
RESN	Processes resonance data from ENDF File 2	If restart problem is not specified
BLOCKR	Reads cross sections from different regions of the resonance region and combines the data arrays to construct continuous-energy cross-section data for resonance region	Always
RESTARTR	Reads cross sections from resonance region	If restart unit is specified
FILE3	Reads and processes cross-section data from ENDF File 3	Always

FIDAS: POLIDENT can processes NNUC nuclides as specified by the user. Within SPARKY, there is a loop over NNUC, and the library routine FIDAS reads the user-specified input for each nuclide to be processed.

OPNFIL: Library routine used to open ENDF data files. If restart information is specified, OPNFIL initializes the restart file.

FILLY: Library routine that writes a message to the logical-output file describing the input ENDF cross-section file (i.e., descriptive title, data set name and volume).

FILE1P: Subroutine that processes the information in File 1 of the ENDF cross-section file.

CALGNU: If $\bar{\nu}$ data are provided, the multiplicity data may be given as a polynomial that is to be constructed from a set of coefficients. This subroutine constructs the appropriate polynomial representation for $\bar{\nu}$ from the coefficients specified in File 1.

RESN: Subroutine that generates continuous-energy cross sections from the parameters that are specified in File 2 of the ENDF file system.

BLOCKR: Subroutine that reads the blocks of cross-section data that are generated by RESN for the resonance regions. BLOCKR combines the blocks of data and treats the corresponding discontinuities that may be present. The complete continuous-energy cross-section representation for each reaction in the resonance region is stored on a scratch device for further processing in subroutine FILE3.

RESTARTR: If the resonance parameters from a previous case were processed completely and the corresponding cross-section data are available, the problem can be restarted without repeating the resonance calculations. Subroutine RESTARTR is used to read the blocks of cross-section data from the resonance region and prepare the data for processing by BLOCKR.

OPNFIL: Library routine that is used to initialize the restart data file.

FILE3: Subroutine that reads the continuous-energy cross-section data from File 3 of the ENDF file system. Subsequently, the data from File 3 are combined with the continuous-energy cross sections in the resonance region. The complete continuous-energy cross-section array is stored on LOGP in a TAB1 format for further processing by other AMPX modules.

3.3 Resonance Processing

Figure 3.3 provides a flow diagram for the resonance processing in POLIDENT.

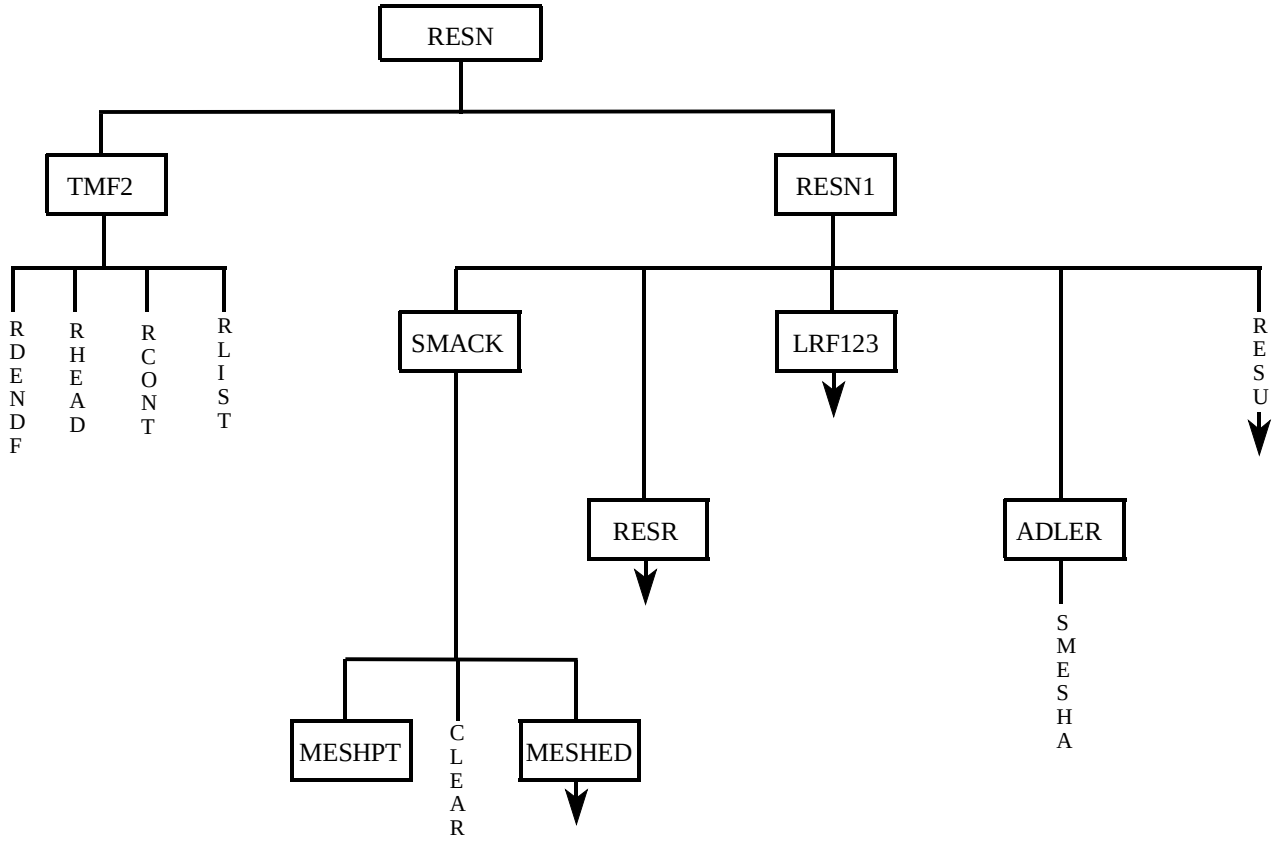


Figure 3.3 Flowchart for resonance processing

TMF2: Subroutine that reads the File 2 resonance information for a specific nuclide and stores the data for later retrieval.

RDENDF: Library routine that searches for a specific set of MAT, MF and MT values on an ENDF tape. Additional subroutine entries are provided to read specific ENDF records.

RHEAD: Entry in RDENDF that searches an ENDF tape for a HEAD record corresponding to a specific MAT, MF and MT.

RCONT: Entry in RDENDF that searches an ENDF tape for a CONT record corresponding to a specific MAT, MF and MT.

RLIST: Entry in RDENDF that searches an ENDF tape for a LIST record corresponding to a specific MAT, MF and MT.

RESN1: Subroutine that processes the resonance information and generates continuous-energy cross sections for the resonance region.

SMACK: Subroutine that is used in the resolved-resonance region to construct a problem-dependent energy mesh for calculating cross sections using the single-level, multi-level Breit-Wigner or Reich-Moore representations.

MESHPT: Subroutine that determines array pointers for the energy-mesh calculation.

CLEAR: Library routine that is called to zero the values of a specific array.

MESHED: Subroutine that calculates the problem-dependent energy mesh.

RESR: Subroutine that processes the single- or multi-level Breit-Wigner parameters and calculates the cross sections on the energy grid that is generated in SMACK.

LRF123: Subroutine that processes the Reich-Moore parameters and calculates the cross sections on the energy grid that is generated in SMACK.

ADLER: Subroutine that processes the Adler-Adler parameters and calculates the cross sections on the energy grid that is generated by SMESHA.

SMESHA: Subroutine that generates a lethargy mesh by ensuring that a constant cross-section ratio is maintained between lethargy points. The lethargy-grid points are converted to energy in subroutine ADLER for calculating cross-section values.

RESU: Library routine that reads and processes the unresolved-resonance data.

3.3.1 Mesh Generation

Figure 3.4 provides a flow diagram for the energy-mesh generation in POLIDENT.

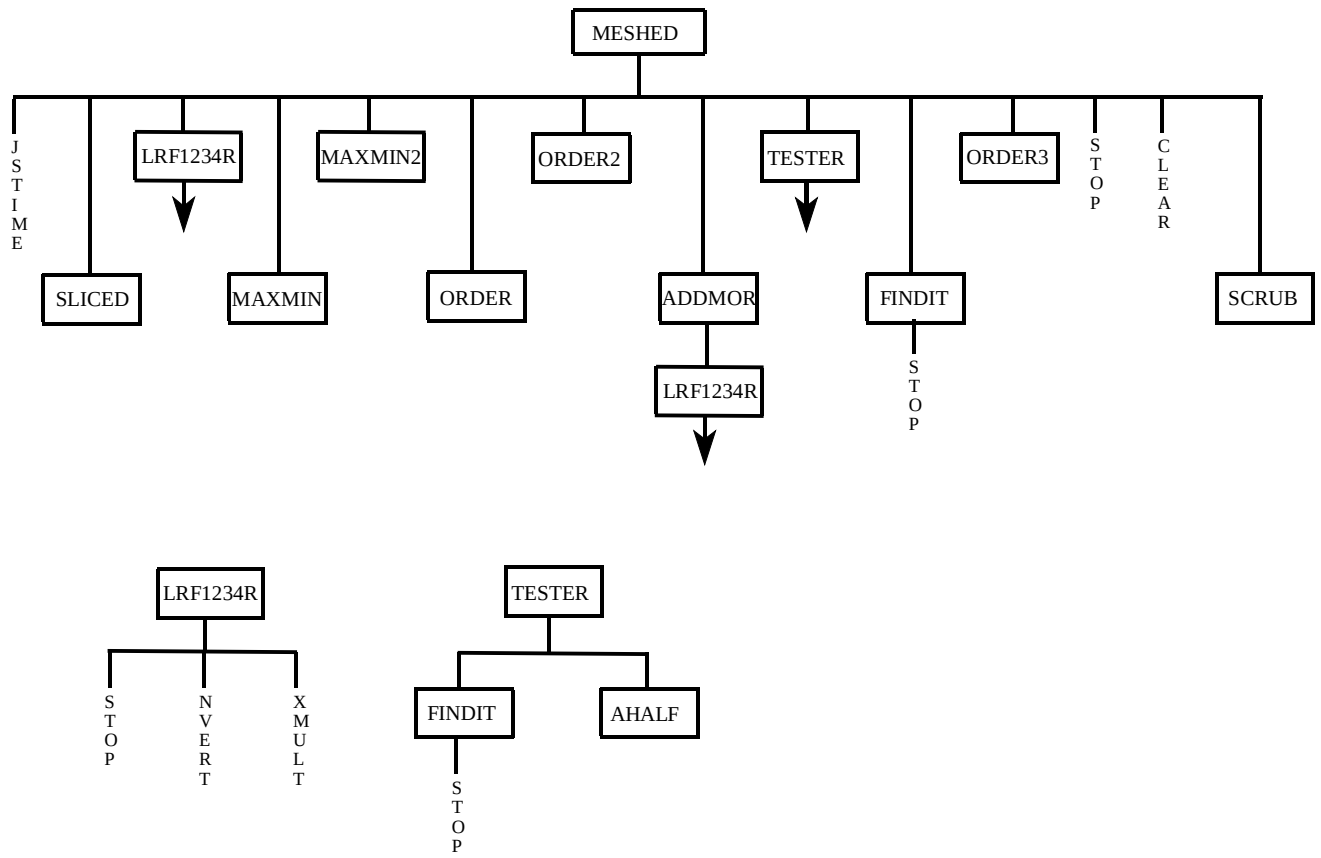


Figure 3.4 Flowchart for energy-mesh generation

JSTIME: Library routine used for timing the energy-mesh calculation.

SLICED: Subroutine that divides the resolved-resonance region and estimates the initial energy increment based on the resonance parameters.

LRF1234R: Subroutine that calculates the cross-section value at a specific energy using the single-level, multi-level Breit-Wigner or Reich-Moore representation.

- STOP: Library routine that is used to stop program execution if problems are encountered with the resonance parameters.
- NVERT: Library routine that inverts a complex matrix.

XMULT: Library routine that forms the product of two complex matrices.

MAXMIN: Subroutine that calculates the critical points (i.e., maximum, minimum and inflection) for the total cross section as a function of energy.

MAXMIN2: Subroutine that calculates the critical points for the capture and fission cross sections.

ORDER: Subroutine that places an energy grid and corresponding cross-section value in ascending order.

ORDER2: Subroutine that places an energy grid and five corresponding cross-section values in ascending order.

ADDMOR: Subroutine that adds points between fine grid energy values using a halving-iteration scheme.

LRF1234R: Refer to previous description.

TESTER: Subroutine that tests each fine-grid point between critical points and determines the energy points that are needed to satisfy the user-specified-convergence tolerance.

FINDIT: Subroutine that locates a specific energy point in an array.

STOP: Library routine that stops program execution if problems are encountered with the energy grid.

AHALF: Subroutine that locates the nearest midpoint between two critical points on the fine grid.

FINDIT: Refer to previous description.

ORDER3: Subroutine that performs the same function as ORDER2, except the subroutine also accommodates power-interpolation parameters.

STOP: Library routine that stops program execution if an error is encountered in calculating the energy mesh.

CLEAR: Library routine that is called to zero the values of a specific array.

SCRUB: Subroutine that removes duplicate values in an array.

3.3.2 Resolved Region

Figures 3.5 and 3.6 provide flow diagrams for processing Breit-Wigner and Reich-Moore resonance data, respectively, in POLIDENT.

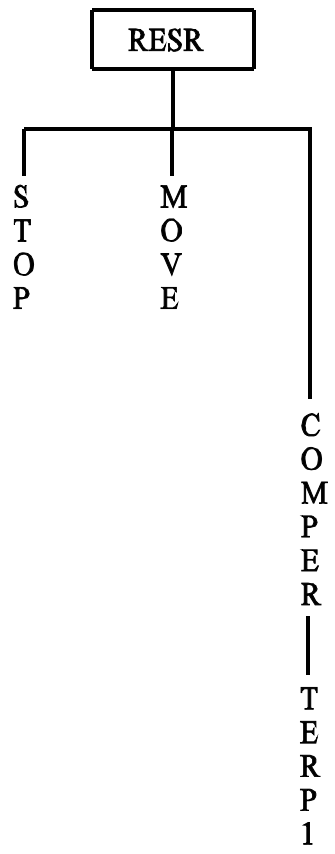


Figure 3.5 Flowchart for processing single- and multi-level Breit-Wigner resonance data

If the data in File 2 have Breit-Wigner resonance parameters in the resolved energy range, subroutine RESR is used to calculate the cross-section values on the energy grid from subroutine SMACK using the appropriate resonance parameters.

STOP: Library routine that stops program execution if problems are encountered with the calculation of cross-section values from the Breit-Wigner parameters.

MOVE: Library routine that moves one array to another array.

COMPER: Library routine that compresses a function to a specified tolerance.

TERP1: Library routine that considers two points (x_1, y_1) and (x_2, y_2) and interpolates a value y given a value x that is between x_1 and x_2.

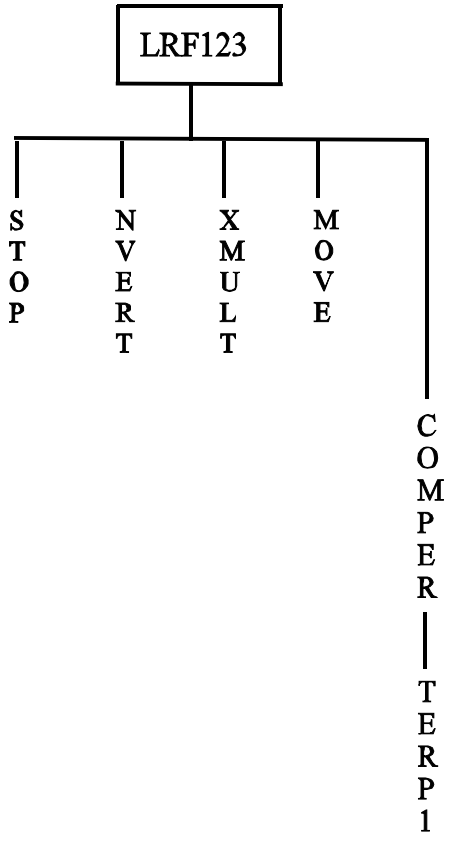


Figure 3.6 Flowchart for processing Reich-Moore resonance data

If the resonance parameters in File 2 require the Reich-Moore representation in the resolved region, subroutine LRF123 is used to calculate the cross-section values on the energy grid from subroutine SMACK using the appropriate resonance parameters.

STOP: Library routine that stops program execution if problems are encountered with the calculation of cross-section values from the Reich-Moore parameters.

NVERT: Library routine that inverts a complex matrix.

XMULT: Library routine that forms the product of two complex matrices.

MOVE: Library routine that moves one array to another array.

COMPER: Library routine that compresses a function to a specified tolerance.

TERP1: Library routine that considers two points (x_1, y_1) and (x_2, y_2) and interpolates a value y given a value x that is between x_1 and x_2.

3.3.3 Unresolved Region

Figure 3.7 provides a flow diagram for processing unresolved-resonance data in POLIDENT.

RESU1: Library routine that determines the energy grid for calculating cross sections in the unresolved resonance region. Three possible cases are considered:

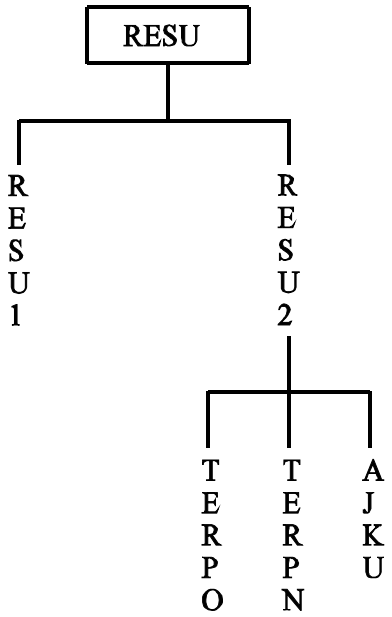


Figure 3.7 Flowchart for processing unresolved-resonance data

1. No fission widths are given and all other parameters are constant (i.e., LFW = 0 and LRF = 1). The unresolved-resonance region is divided into equally spaced intervals.

2. Fission widths are provided as a function of energy and all other parameters are constant (i.e., LFW = 1 and LRF = 1). The energy-grid points are provided with the unresolved-resonance data.

3. All parameters vary as a function of energy (i.e., LFW = 0 or 1 and LRF = 2). The energy-grid points are provided with the unresolved-resonance data.

RESU2: Library routine that calculates unresolved cross sections based on the resonance-parameter data from File 2. Cross-section values are calculated at grid points that are provided by RESU1.

TERPO: Library routine that considers two vectors x and y, and interpolates a value yy given an argument xx that is in x.

TERPN: Library routine that considers the two-dimensional arrays x and y, and interpolates a value yy given an argument xx that is in x.

AJKU: Library routine that calculates the *J* and *K* integrals that are used to compute the isolated-resonance-fluctuation integrals.

3.4 Post-Resonance Processing

Subroutine BLOCKR is used to process the blocks of cross-section data that are generated for the resonance region (Figure 3.8). For example, the resolved- and unresolved-resonance data are processed separately. Consequently, POLIDENT must combine the cross-section data from the resolved region with the corresponding cross-section data from the unresolved-resonance region. In addition, the resolved-resonance region could be divided into different energy ranges. As a result, the cross-section data from the resolved region may consist of multiple cross-section arrays that must be combined to form a single cross-section array. The combining procedure must consider any discontinuities that may exist between the two arrays or functions of data.

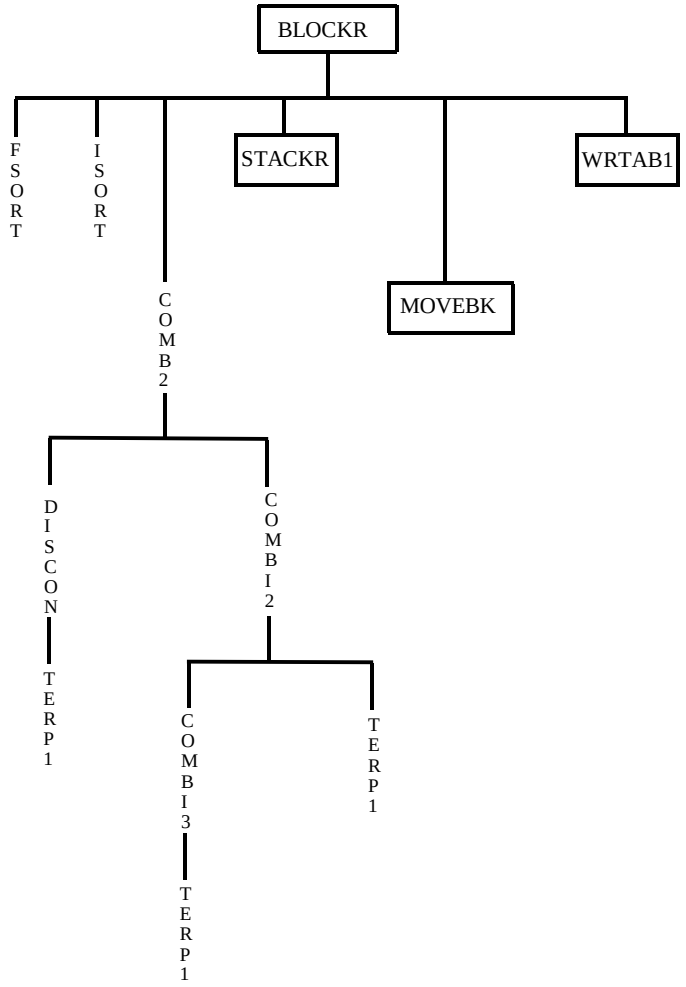


Figure 3.8 Flowchart for processing cross-section data from resonance region

FSORT: Subroutine that sorts the blocks of cross-section data from the resonance region based on the lower or upper energy value of the data range (i.e., E_L or E_H).

ISORT: Subroutine that sorts the blocks of cross-section data from the resonance calculation based on the MT number.

COMB2: Library routine that considers two functions and combines the two functions by addressing the different interpolation schemes and/or discontinuities at the boundary of each function.

- DISCON: Library routine that evaluates the points of discontinuity in two functions a and b.
  - TERP1: Library routine that considers two points (x_1, y_1) and (x_2, y_2) and interpolates a value y given a value x that is between x_1 and x_2.
- COMBI2: Library routine that controls the combining of panels from one function (x_{a1}, y_{a1}), (x_{a2}, y_{a2}) with the panel from a second function (x_{b1}, y_{b1}), (x_{b2}, y_{b2}).
  - COMBI3: Library routine that controls the combining of panels from two functions if the range of one of the functions has been exceeded.
  - TERP1: Refer to previous description.

STACKR: Subroutine that combines two functions that do not overlap into a single function.

MOVEBK: Subroutine that moves a one-dimensional (1-D) array x to a 1-D array y, beginning with the last element and proceeding to the first element.

WRTAB1: Subroutine that writes the cross-section data on a scratch device in an ENDF TAB1 format.

Subroutine FILE3 completes the continuous-energy cross-section construction by combining the cross-section data from the resonance region with the background cross-section values from File 3 of the ENDF file system (Figure 3.9).

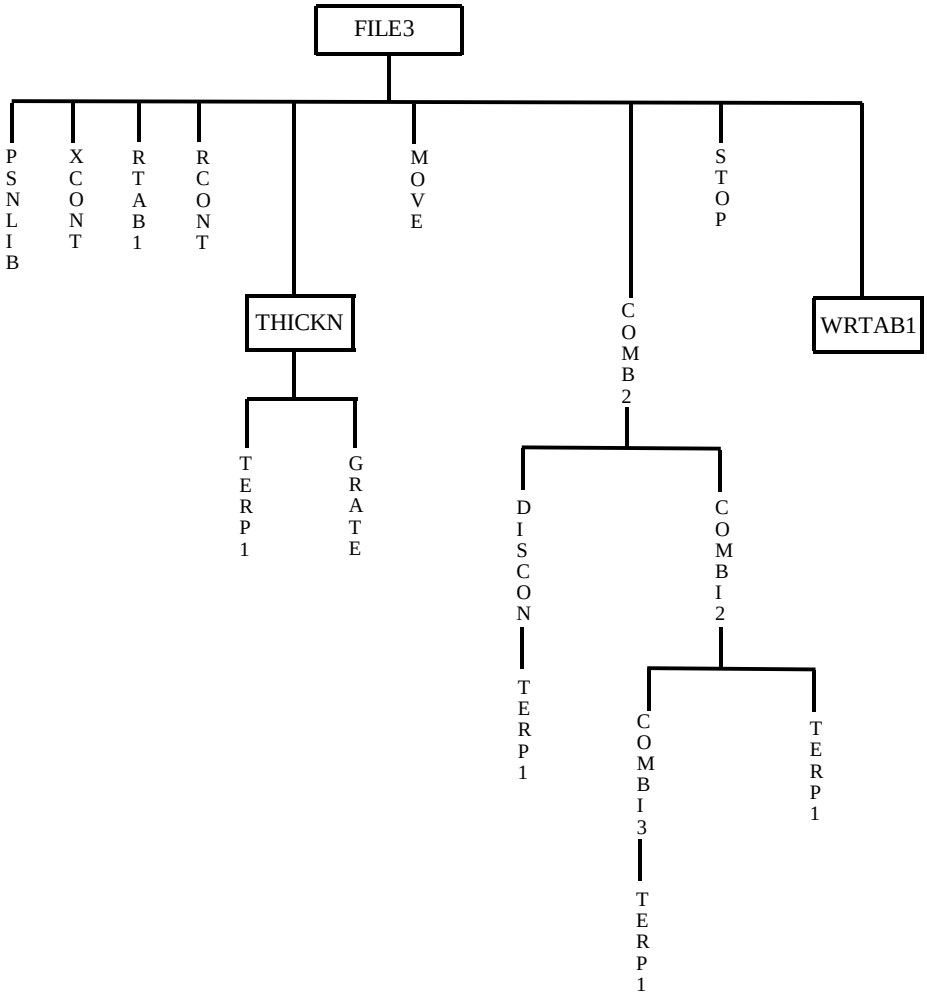


Figure 3.9 Flowchart for combining File 3 data with cross sections from the resonance region

PSNLIB: Library routine that searches for a specific set of MAT, MF and MT values on an ENDF tape.

XCONT: Entry in library routine RDENDF that reads an ENDF CONT record.

RTAB1: Entry in library routine RDENDF that searches an ENDF tape for a TAB1 record corresponding to a specific MAT, MF and MT.

RCONT: Entry in RDENDF that searches an ENDF tape for a CONT record corresponding to a specific MAT, MF and MT.

THICKN: Subroutine that adds points to the background energy grid by ensuring that a constant cross-section ratio is maintained between energy points.

- TERP1: Library routine that considers two points (x_1, y_1) and (x_2, y_2) and interpolates a value y given a value x that is between x_1 and x_2.
- GRATE: Library routine that integrates the cross-section function over a specified energy range.

MOVE: Library routine that moves one array to another array.

COMB2: Library routine that considers two functions and combines the two functions by addressing the different interpolation schemes and/or discontinuities at the boundary of each function.

- DISCON: Library routine that evaluates the points of discontinuity in two functions a and b.
  - TERP1: Library routine that considers two points (x_1, y_1) and (x_2, y_2) and interpolates a value y given a value x that is between x_1 and x_2.
- COMBI2: Library routine that controls the combining of panels from one function (x_{a1}, y_{a1}), (x_{a2}, y_{a2}) with the panel from a second function (x_{b1}, y_{b1}), (x_{b2}, y_{b2}).
  - COMBI3: Library routine that controls the combining of panels from two functions if the range of one of the functions has been exceeded.
  - TERP1: Refer to previous description.

STOP: Library routine that stops program execution if problems are encountered while combining the cross-section data from the resonance region with the background cross-section values.

WRTAB1: Subroutine that writes the cross-section data on the logical unit LOGP in an ENDF TAB1 format.

4 INPUT DATA GUIDE

POLIDENT uses the FIDO input method, which is described in Appendix D, for specifying the input data for a problem. The following section provides a detailed description of the POLIDENT input options.

4.1 FIDO Input Structure

Block 1.

−1$ Core Allocation [1]

1. NWORD - Number of words to allocate **(1000000)**

0$ Output Library [3]

1. LOGP - Output unit with cross sections in TAB1 format **(31)**.
2. LOGP1 - Output unit with File 1 and abbreviated File 2 information **(32)**.
3. LOGRES - Restart unit **(0)**.

Note: When the resonance calculation is complete, the cross sections from the resonance calculation are typically stored on scratch device 18. While processing an evaluation in BLOCKR or FILE3, a problem could be encountered following the resonance calculation (e.g., storage limit exceeded, parameter index exceeded, etc.). Therefore, the user can rename logical unit 18 to a restart unit (LOGRES). After correcting the input problem, the cross sections from the resonance region can be read from the restart unit as opposed to repeating the resonance calculation.

1$ Number of Cases [1]

1. NNUC - Number of nuclides to process

T Terminate Block 1.

Repeat Block 2 NNUC times.

Block 2.

2$ ENDF/B Data Source [4]

1. MATNO - ENDF Material identifier for nuclide to be processed.
2. NDFB - Logical unit number for ENDF library **(11)**.
3. MODE - ENDF library format (1 - binary/ 2 - BCD) **(1)**.
4. NVERS - Version of ENDF data **(6)**.

4* Floating-Point Parameters [14]

1. EPS - To construct a final function, POLIDENT must combine strings of point data from File 3 with resolved- and unresolved-data arrays. The final function contains enough points such that any intermediate value is accurate to EPS **(0.001 or 0.1%)**.

Note: Options 2 through 4 are only used for processing nuclides with the Adler-Adler formalism. Option 5 is used for specifying the fractional-convergence tolerance for an energy mesh with all resonance formalisms except Adler-Adler.

2. R - The ratio factor used in a cross-section energy mesh.
3. XNP - The number of points taken equally spaced in lethargy between "resonance bodies" **(50.0)**.
4. XGT - The multiplier on the total width above and below a resonance over which the ratio mesh scheme is used **(50.0)**.

5. OPT(1) - Convergence tolerance for energy-mesh generation **(0.01)**.
6. OPT(2) - Not used.

.
.
.

14. OPT(10) - Not used.

5$ Options [8]

1. IOPT1
2. IOPT2

These are used in conjunction with IOPT4 through IOPT8 to select the ENDF/B interpolation options to be tried. These are both set to 1 by default.

3. IOPT3

Maximum number of interpolation regions allowed in the output ENDF/B function.

4. IOPT4
5. IOPT5
6. IOPT6
7. IOPT7
8. IOPT8

Interpolation types to be tried. IOPT4 is 2 by default, and IOPT5 through IOPT8 are zero by default.

IOPT4 through IOPT8 are used to specify the interpolation types and their order, which will be tried in combining two or more ENDF/B functions. The types are as follows:

Value	Type of interpolation
1	Histogram
2	Linear x, linear y
3	Linear x, log y
4	Log x, linear y
5	Log x, log y

IOPT1 and IOPT2 specify which entries in the five-position table are to be used (e.g., the default values of 1 say to use only the first entry in the table, or linear-linear interpolation by default).

6$ Function Parameters [4]

1. N1MAX - Maximum number of interpolation regions in any "input" ENDF/B function **(20)**.
2. N2MAX - Maximum number of points in any input ENDF/B TAB1 array **(10000)**.
3. MLBW - If MLBW > 0, force single-level Breit-Wigner evaluations to multi-level Breit-Wigner format **(0)**.
4. IPOINTS - Maximum number of points per 10 eV interval **(5000)**.

T Terminate Block 2.

4.2 Logical Unit Parameters

The logical units that are used by POLIDENT are provided in Table 4.1.

Table 4.1 File parameters for logical units

Var.	Unit No.	Type	Description
N5	5	BCD	FIDO-formatted input file
N6	6	BCD	Formatted output file
NDFB	User defined	BCD or Binary	ENDF data file
LOGP	User defined	Binary	Cross-section output in TAB1 format
LOGP1	User defined	Binary	ENDF File 1 and abbreviated File 2 data
LOGRES	User defined	Binary	Restart information
N14	14	Binary	Scratch
N15	15	Binary	Scratch
N16	16	Binary	Scratch
N17	17	Binary	Scratch
N18	18	Binary	Scratch

5 DESCRIPTION OF OUTPUT

The following section provides a brief description of the POLIDENT output. An entire output file is not provided in this section. However, portions of the output are printed and discussed.

5.1 Header Page

A sample header page is provided in Figure 5.1. The program title is printed in the first line of block letters. Within the second line of block letters, the problem identification is provided. The third line provides the date the job was executed in terms of the month, day and year. The last line provides the time of execution in an hour, minute and second format. Note that the time printout is in terms of a 24-hour clock with midnight denoted as 2400 hours. The header page is printed by subroutine MESAGE, which is discussed in Section 3.

```
1 pppppppppppp    oooooooooooo   ll            iiiiiiiiiiii   ddddddddddddd   ##########e   nn          nn  ttttttttttttt
  pppppppppppp    oooooooooooo   ll            iiiiiiiiiiii   ddddddddddddd   ##########e   nnn         nn  ttttt
  pp        pp    oo        oo   ll                   ii      dd        dd    ee            nnnn        nn
  pp        pp    oo        oo   ll                   ii      dd        dd    ee            nn nn       nn      tt
  pp        pp    oo        oo   ll                   ii      dd        dd    ee            nn  nn      nn
  pppppppppppp    oo        oo   ll                   ii      dd        dd    ########      nn   nn     nn
  pppppppppppp    oo        oo   ll                   ii      dd        dd    ########      nn    nn    nn      tt
  pp              oo        oo   ll                   ii      dd        dd    ee            nn     nn   nn
  pp              oo        oo   ll                   ii      dd        dd    ee            nn      nn  nn
  pp              oo        oo   ll                   ii      dd        dd    ee            nn       nn nn
  pp              oo        oo   ll                   ii      dd        dd    ee            nn        nnnn      tt
  pp               oooooooooooo   llllllllllll  iiiiiiiiiiii   ddddddddddddd   ##########e   nn         nnn
  pp               oooooooooooo   llllllllllll  iiiiiiiiiiii   ddddddddddddd   ##########e   nn          nn     tt
0

  ffffffffffff   99999999999      qqqqqqqq
  ffffffffffff   99999999999    qqqqqqqqqq
  ff             99        99   qq
  ff             99        99   qq        qq
  ff             99        99   qq        qq
  ffffffff       99999999999    qq        qq
  ffffffff       99999999999    qq        qq
  ff                       99   qq        qq
  ff                       99   qq        qq
  ff                       99   qq      qq
  ff             99999999999    qq    qqqq
  ff             99999999999     qqqqqqqqqq
  ff             99999999999      qqqqqqqq
0

     0000000      9999999999            //      11       5555555555555           //   99999999999    999
   000000000    99999999999           //       111      5555555555555          //   99999999999  99999999999
  00       00   99        99         //       1111      55                    //   99        99   99        99
  00       00   99        99        //         11      55                   //    99        99   99        99
  00       00   99        99       //          11      55                   //    99        99   99        99
  00       00   99999999999      //            11      5555555555555       //     99999999999  99999999999
  00       00    9999999999    //              11      5555555555555      //       99999999999   9999
  00       00                99  //            11                 55    //                  99            99
  00       00                99 //             11                 55   //                   99            99
   00     00                 99   //           11      55        55  //                    99
    000000000    99999999999    //            11111111  5555555555555 //          99999999999  99999999999
     0000000     99999999999   //             11111111   555555555555 //           99999999999  99999999999
0

     0000000      9999999999                    0000000          11                          11       555555555555
   000000000    99999999999                   000000000         111                        111       55555
  00       00   99        99        :::      00       00       1111              :::       1111       55
  00       00   99        99        :::      00       00         11              :::         11       55
  00       00   99999999999                   00       00         11                          11       5555555555555
  00       00   9999999999                    00       00         11                          11       5555555555555
  00       00                99        :::      00       00         11              :::         11
  00       00                99        :::      00       00         11              :::         11                 55
   00     00                 99        :::       00     00          11              :::         11       55        55
    000000000    99999999999                    000000000       11111111                    11111111   55555
     0000000     99999999999                     0000000        11111111                    11111111    5555555555
1
```

Figure 5.1 Example header-page output

5.2 Program-Verification Information

Following the header page, the program-verification information is provided for quality-assurance purposes. An example of the program-verification information is provided in Figure 5.2. The information includes the code name, the creation date of the load module, library that contains the load module, the computer-code name from the configuration-control table and the revision number. In addition, the job name, date, and time of execution are printed.

```
*******************************************************************************
*******************************************************************************
*******************************************************************************
*****                                                                     *****
*****            program verification information                          *****
*****                                                                     *****
*****           code system:  unknown  version:                           *****
*****                                                                     *****
*******************************************************************************
*******************************************************************************
*****                                                                     *****
*****                                                                     *****
*****          program:  polident                                          *****
*****                                                                     *****
*****      creation date:  15_sep_1                                       *****
*****                                                                     *****
*****          library:  /home/f9q/ampx/updates                           *****
*****                                                                     *****
*****                                                                     *****
*****   this is not a unknown  configuration controlled code              *****
*****                                                                     *****
*****          jobname:  f9q                                              *****
*****                                                                     *****
*****   date of execution:  09/15/99                                      *****
*****                                                                     *****
*****   time of execution:  09:01:15                                      *****
*****                                                                     *****
*****                                                                     *****
*******************************************************************************
*******************************************************************************
*******************************************************************************
```

Figure 5.2 Example program-verification output

5.3 Problem-Verification Information

Execution of POLIDENT requires the development of a user-input file as described in Section 4. Due to the possibility of typing errors, incorrect unit specification, etc., the code provides a problem- or input-verification table in the output. This section should be evaluated to ensure the desired input information is provided. Following the program-verification information is the problem-verification information. An example of the problem-verification information is provided in Figure 5.3. As shown in Figure 5.3, the amount of storage that is allocated for POLIDENT is printed. In addition, the logical units for the ENDF tape and output cross-section library are provided in the output. Furthermore, the convergence or reconstruction tolerance for generating the cross-section energy mesh is provided with maximum number of energy points (IPOINTS) per decade (i.e., 10-eV interval). The problem-verification output also indicates the type of interpolation used for the problem. For the problem in Figure 5.3, one interpolation region is defined (PSN=1) that will have linear-linear interpolation (TYPE=2). The maximum number of points in a record (N2MAX) and the maximum number of interpolation regions (N1MAX) are provided following the interpolation types. Additional problem-verification information includes the material to be processed (MATNO), ENDF library format (MODE), version of ENDF data (NVERS), background cross-section value and processing temperature.

```
1
0   -1$ array      1 entries read
0    0$ array      3 entries read
0    1$ array      1 entries read
0    0t

 ==============================================================================
 logical 31 (output point library)
 dataset name:  unknown
       volume:
 ==============================================================================

 Polident has been allocated 2000000 words of core
 A Tab1 file will be written on logical 31
0    2$ array      4 entries read
0    6$ array      4 entries read
0    0t

 tolerance for combining arrays for this nuclide:   0.10 percent

 ==============================================================================
 logical 11 (input endf/b library)
 dataset name:  unknown
       volume:
 ==============================================================================

 reconstruction tolerance for mesh  1.00000E-02
 maximum number of points per decade       3500

 The following interpolation types will be used
   psn  type
 ----- -----
     1     2

 The maximum number of points in a record is   10000

 The maximum number of interpolation regions is      20
0        processing mat:         9443
   endf data on logical:           11
  mode of the endf data:            2
                   sig0:     1.00E+10
            temperature:         0.00
           endf version:            6
```

Figure 5.3 Example problem-verification-information output

5.4 File 1 Processing

Prior to processing the resonance parameters in File 2, POLIDENT reads the information in File 1 of the ENDF tape. POLIDENT writes the directory information (MT=451) on LOGP1 and processes the $\bar{\nu}$ data if applicable (i.e., MT = 452, 453 and 456). If fission-neutron-multiplicity data are available, POLIDENT processes the data and writes descriptive processing information to the output file. Figure 5.4 provides a $\bar{\nu}$ processing output example. The $\bar{\nu}$ data can be in either a functional or tabular format, and POLIDENT indicates the appropriate data format. As shown in Figure 5.4, the functional output provides the energy points and corresponding values for $\bar{\nu}$. In addition, the interpolation table is provided for interpolating between energy points. The first number in the interpolation table is the number of regions, followed by the number of points and type of interpolation. For the average total number of neutrons per fission (MT = 452) in the example, one linear-linear interpolation region is provided with five data points.

```
starting processing for mt=  452

         nu representation: 2


values of function
   1 1.00000E-05 2.94530E+00
   2 4.72100E+05 2.94530E+00
   3 4.00000E+06 3.50650E+00
   4 7.00000E+06 3.96220E+00
   5 2.00000E+07 5.97120E+00

interpolation table
              1           5           2

starting processing for mt=  455

precursor family decay constants (1/s):
1.35990E-02 2.99660E-02 1.16730E-01 3.06910E-01 8.70100E-01 3.00280E+00

         nu representation: 2

values of function
    1 1.00000E-05 1.62000E-02
    2 4.00000E+06 1.62000E-02
    3 7.00000E+06 8.40000E-03
    4 2.00000E+07 8.40000E-03

interpolation table
              1           4           2

starting processing for mt=  456

         nu representation: 2

values of function
   1 1.00000E-05 2.92910E+00
   2 4.72100E+05 2.92910E+00
   3 4.00000E+06 3.49030E+00
   4 7.00000E+06 3.95380E+00
   5 2.00000E+07 5.96280E+00

interpolation table
              1           5           2
```

Figure 5.4 Example File 1 processing output

5.5 File 2 Processing

Typically, the majority of CPU time is devoted to processing the resonance-parameter information in File 2. Initially, POLIDENT processes the resolved-resonance information followed by the unresolved data. The subsequent sections describe the output from both resonance regions.

5.5.1 Resolved-Resonance Data

A section of an example POLIDENT output file is provided in Figure 5.5 for processing the resolved-resonance parameters. As shown in Figure 5.5, the energy range(s) within the resolved-resonance region are provided with the corresponding resonance formula (e.g., Reich-Moore, single-level Breit-Wigner, etc.). For the energy range of interest, the boundaries of the region are provided with the total number of grid points that are used to construct the fine energy grid. The resulting number of auxiliary-grid points is also provided with the amount of CPU time that is required for generating the energy mesh. Following the mesh information is a table of the resolved-resonance parameters that includes the number of resonances, energy of each resonance and the corresponding resonance parameters. After listing the resonance parameters, POLIDENT summarizes the results for each reaction in the resolved-resonance region and prints the amount of CPU required for processing the resolved data.

```
pr ogram has located the head record for file 2

     energy range 1 extends from 1.00000E-05 to 3.00000E+02 ev

              reich-moore parameters are being processed
 mesh parameters for material      9443
 energy range:  1.00000E-05 to  3.00000E+02
 total number of fine grid points         39689
 total number of auxiliary grid points      5577
 cpu time for processing  mesh    59.00 seconds

 mesh from 1.00000E-05 to 3.00000E+02 has    5577 points

 resonance data--number of resonances  243
 resonance      energy           j          gn          gg         gfa         gfb
 --------- ----------- ----------- ----------- ----------- ----------- -----------
         1-5.95300E+01 2.00000E+00 5.96100E-01 4.45000E-02 4.15300E-01 4.29800E-02
         2-5.58000E+00 3.00000E+00 2.28900E-03 3.64400E-02 1.64800E+00 1.57600E-02
         3-1.40500E+00 3.00000E+00 1.51300E-06 1.92500E-02-1.87100E-02 3.69400E-04
         4-1.22500E-01 3.00000E+00 1.67700E-05 4.37600E-02 2.05400E-02 9.39600E-04
         5 2.64695E-01 3.00000E+00 4.25200E-05 3.33500E-02-5.04200E-02 2.53600E-02
         6 1.73500E+00 2.00000E+00 2.08000E-06 4.00000E-02 6.82200E-02 2.74200E-01
         7 4.28200E+00 3.00000E+00 5.75600E-04 3.27300E-02 2.68000E-02 0.00000E+00
         8 4.58200E+00 2.00000E+00 4.78400E-04 3.92800E-02-1.96900E-02 1.24900E-01
         9 5.81100E+00 2.00000E+00 2.77300E-03 6.35700E-02-1.13900E+00 2.10100E-01
        10 6.93200E+00 3.00000E+00 6.19500E-04 3.56500E-02-1.04400E-01 4.51700E-04
                                                   .
                                                   .
                                                   .
       235 2.93450E+02 3.00000E+00 2.16600E-03 4.00000E-02-2.94900E-01 1.29500E-01
       236 2.94340E+02 3.00000E+00 3.05900E-03 4.00000E-02 5.61600E-02-1.70500E-01
       237 2.96880E+02 2.00000E+00 3.86000E-03 4.16300E-02-3.36300E-02 2.05800E-02
       238 2.97510E+02 3.00000E+00 1.33600E-02 4.00000E-02-1.43600E-01 9.19800E-02
       239 2.97840E+02 2.00000E+00 4.11000E-03 4.28800E-02 2.05400E-02-7.55800E-02
       240 2.99120E+02 2.00000E+00 7.36800E-03 4.00000E-02 1.59800E-01 1.66300E-01
       241 3.01780E+02 2.00000E+00 1.76500E-02 4.00000E-02 2.99100E-01 7.22200E-02
       242 3.20000E+02 3.00000E+00 1.87900E-02 4.00000E-02 2.81300E-01 3.58500E-01
       243 4.00000E+02 2.00000E+00 4.39700E-01 4.00000E-02 6.43100E-01 5.46600E-01

 ========================================================
 MAT:9443 MT:  1 Temperature: 0.00000E+00
 The original function has   5577 points
 The new function has   5577 points
0resolved resonance cross-section vectors with  5577 points have been created from 1.00000E-05   to 3.00000E+02 ev for mt   1
 ========================================================
 MAT:9443 MT:  2 Temperature: 0.00000E+00
 The original function has   5577 points
 The new function has   5528 points
0resolved resonance cross-section vectors with  5528 points have been created from 1.00000E-05 to 3.00000E+02 ev for mt   2
 ========================================================
 MAT:9443 MT: 18 Temperature: 0.00000E+00
 The original function has   5577 points
 The new function has   5577 points
0resolved resonance cross-section vectors with  5577 points have been created from 1.00000E-05 to 3.00000E+02 ev for mt  18
 ========================================================
 MAT:9443 MT:102 Temperature: 0.00000E+00
 The original function has   5577 points
 The new function has   5577 points
0resolved resonance cross-section vectors with  5577 points have been created from 1.00000E-05 to 3.00000E+02 ev for mt 102

 =========================time for this block:  1.03 min
```

Figure 5.5 Example resolved-resonance-processing output

5.5.2 Unresolved-Resonance Data

Once the resolved-resonance processing is complete, POLIDENT processes the unresolved-resonance information in File 2. An example of the unresolved-resonance-processing section of the output is provided in Figure 5.6. As shown in Figure 5.6, the energy boundaries of the unresolved-resonance region are provided with a table of the corresponding unresolved-resonance parameters. In addition, the output includes the amount of CPU time required for processing the unresolved data as well as the total CPU time required for processing the entire resonance region. Note that the total CPU time includes the time for generating the energy mesh.

```
     energy range 2 extends from 3.00000E+02 to 4.02000E+04 ev

          unresolved slbw parameters are being processed
1temperature: 0.00000E+00 sig0: 1.00000E+10
 point      energy     elastic     capture     fission       total competitive   transport
     1 3.00000E+02 1.29684E+01 6.77152E+00 2.37804E+01 4.35204E+01 0.00000E+00 4.36015E+01
     2 3.50000E+02 1.29578E+01 6.29081E+00 2.17868E+01 4.10354E+01 0.00000E+00 4.11228E+01
     3 4.50000E+02 1.28672E+01 5.43292E+00 1.89875E+01 3.72876E+01 0.00000E+00 3.73859E+01
     4 5.50000E+02 1.25235E+01 4.49302E+00 1.57542E+01 3.27708E+01 0.00000E+00 3.28702E+01
     5 6.00000E+02 1.21376E+01 3.71308E+00 1.34975E+01 2.93482E+01 0.00000E+00 2.94397E+01
     6 6.50000E+02 1.18208E+01 3.14167E+00 1.09985E+01 2.59609E+01 0.00000E+00 2.60417E+01
     7 7.00000E+02 1.17531E+01 2.88488E+00 1.04983E+01 2.51363E+01 0.00000E+00 2.52185E+01
     8 7.50000E+02 1.17528E+01 2.80779E+00 1.00986E+01 2.46592E+01 0.00000E+00 2.47443E+01
     9 8.00000E+02 1.17907E+01 2.78877E+00 9.99848E+00 2.45780E+01 0.00000E+00 2.46681E+01
    10 8.50000E+02 1.18352E+01 2.78876E+00 9.89855E+00 2.45226E+01 0.00000E+00 2.46179E+01
    11 9.00000E+02 1.18984E+01 2.79882E+00 9.99855E+00 2.46958E+01 0.00000E+00 2.47980E+01
    12 9.50000E+02 1.19734E+01 2.83707E+00 1.00982E+01 2.49087E+01 0.00000E+00 2.50181E+01
    13 1.00000E+03 1.20599E+01 2.85610E+00 1.02982E+01 2.52142E+01 0.00000E+00 2.53319E+01
    14 1.50000E+03 1.22022E+01 2.56292E+00 8.95223E+00 2.37173E+01 0.00000E+00 2.38745E+01
    15 2.00000E+03 1.20496E+01 2.09424E+00 7.89869E+00 2.20426E+01 0.00000E+00 2.22238E+01
    16 2.50000E+03 1.19283E+01 1.76337E+00 7.09533E+00 2.07870E+01 0.00000E+00 2.09869E+01
    17 3.50000E+03 1.18515E+01 1.49677E+00 6.09629E+00 1.94446E+01 0.00000E+00 1.96842E+01
    18 5.50000E+03 1.16854E+01 1.19134E+00 4.89731E+00 1.77741E+01 0.00000E+00 1.80717E+01
    19 7.50000E+03 1.15029E+01 1.00061E+00 4.09810E+00 1.66016E+01 0.00000E+00 1.69332E+01
    20 9.50000E+03 1.13937E+01 8.95802E-01 3.69835E+00 1.59879E+01 0.00000E+00 1.63575E+01
    21 1.50000E+04 1.11924E+01 7.43907E-01 3.29758E+00 1.52338E+01 0.00000E+00 1.57144E+01
    22 2.50000E+04 1.08639E+01 5.93828E-01 2.80664E+00 1.42644E+01 0.00000E+00 1.48613E+01
    23 3.50000E+04 1.06282E+01 5.14892E-01 2.51645E+00 1.36595E+01 0.00000E+00 1.43311E+01
    24 4.02000E+04 1.05299E+01 4.85545E-01 2.42607E+00 1.34415E+01 0.00000E+00 1.41482E+01
 ========================time for this block:  0.00 min
               total resonance processing time:  1.03 min
```

Figure 5.6 Example unresolved-resonance-processing output

5.6 Post-Resonance Processing

The subroutines that process the cross-section data after the resonance calculations are discussed in Section 3. In particular Subroutine BLOCKR evaluates the data from the different energy ranges of the resonance region. The resolved and unresolved data are divided into separate energy ranges. Consequently, BLOCKR must evaluate the blocks of cross-section data from the different regions and combine the data to form a single cross-section array for each reaction. While processing the different blocks of data, POLIDENT provides diagnostic information that describes the different blocks of cross-section data from the resonance region. Figure 5.7 provides an example output for processing data blocks from the resonance region. Prior to combining the arrays, a directory of the different cross-section data blocks is provided that includes the reaction identifier, energy boundaries, number of interpolation regions, number of grid points per region and the corresponding record location of each block on the scratch device. Once the initial directory is created, POLIDENT sorts the blocks of cross-section data based on the upper- and lower-energy boundaries as well as the reaction identifier. A directory based on each sorting criterion is also provided in the output file as shown in Figure 5.7. After sorting the data, BLOCKR combines the blocks of cross-section data for each reaction and summarizes the results of the combining process in the output file. For each reaction, the energy range and number of points in each block is provided with the corresponding interpolation information. Once the data blocks are combined, the final function is written in a TAB1 format to a scratch device that is noted in the output. The descriptive output identifies the associated MAT, MF and MT for the complete function with the total number of points. Additional information that includes the atomic weight ratio, ZA number, temperature and background cross-section value is also provided in the output as shown in Figure 5.7.

```
original map of stuff on n18
 mt          el          eh   n1   n2 irec
--- ----------- ----------- ---- ---- ----
  1 1.00000E-05 3.00000E+02    1 5577    2
  2 1.00000E-05 3.00000E+02    1 5528    5
 18 1.00000E-05 3.00000E+02    1 5577    8
102 1.00000E-05 3.00000E+02    1 5577   11
  1 3.00000E+02 4.02000E+04    1   24   14
  2 3.00000E+02 4.02000E+04    1   24   17
 18 3.00000E+02 4.02000E+04    1   24   20
102 3.00000E+02 4.02000E+04    1   24   23

map after sorting on eh
 mt          el          eh   n1   n2 irec
--- ----------- ----------- ---- ---- ----
  1 1.00000E-05 3.00000E+02    1 5577    2
  2 1.00000E-05 3.00000E+02    1 5528    5
 18 1.00000E-05 3.00000E+02    1 5577    8
102 1.00000E-05 3.00000E+02    1 5577   11
  1 3.00000E+02 4.02000E+04    1   24   14
  2 3.00000E+02 4.02000E+04    1   24   17
 18 3.00000E+02 4.02000E+04    1   24   20
102 3.00000E+02 4.02000E+04    1   24   23

map after sorting on el
 mt          el          eh   n1   n2 irec
--- ----------- ----------- ---- ---- ----
  1 1.00000E-05 3.00000E+02    1 5577    2
  2 1.00000E-05 3.00000E+02    1 5528    5
 18 1.00000E-05 3.00000E+02    1 5577    8
102 1.00000E-05 3.00000E+02    1 5577   11
  1 3.00000E+02 4.02000E+04    1   24   14
  2 3.00000E+02 4.02000E+04    1   24   17
 18 3.00000E+02 4.02000E+04    1   24   20
102 3.00000E+02 4.02000E+04    1   24   23

map after sorting on mt
 mt          el          eh   n1   n2 irec
--- ----------- ----------- ---- ---- ----
  1 1.00000E-05 3.00000E+02    1 5577    2
  1 3.00000E+02 4.02000E+04    1   24   14
  2 1.00000E-05 3.00000E+02    1 5528    5
  2 3.00000E+02 4.02000E+04    1   24   17
 18 1.00000E-05 3.00000E+02    1 5577    8
 18 3.00000E+02 4.02000E+04    1   24   20
102 1.00000E-05 3.00000E+02    1 5577   11
102 3.00000E+02 4.02000E+04    1   24   23
============================================================
process:    1 has a function which goes from  1.00000E-05 up to 3.00000E+02
and has   5577 points and   1 interpolation regions
to stack with a second function which goes from 3.00000E+02 up to  4.02000E+04
and has     24 points and   1 interpolation regions
--------->a tab1 record has been written on logical 14
          for mat:9443 mf: 3 mt:   1 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5600
============================================================
process:    2 has a function which goes from  1.00000E-05 up to  3.00000E+02
and has    5528 points and   1 interpolation regions
to stack with a second function which goes from 3.00000E+02 up to  4.02000E+04
and has     24 points and   1 interpolation regions
--------->a tab1 record has been written on logical 14
          for mat:9443 mf: 3 mt:   2 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5551
============================================================
process:   18 has a function which goes from  1.00000E-05 up to  3.00000E+02
and has   5577 points and   1 interpolation regions
to stack with a second function which goes from 3.00000E+02 up to  4.02000E+04
and has     24 points and   1 interpolation regions
--------->a tab1 record has been written on logical 14
          for mat:9443 mf: 3 mt:  18 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points: 5600
============================================================
process:  102 has a function which goes from  1.00000E-05 up to  3.00000E+02
and has   5577 points and   1 interpolation regions
to stack with a second function which goes from 3.00000E+02 up to  4.02000E+04
and has     24 points and   1 interpolation regions
--------->a tab1 record has been written on logical 14
          for mat:9443 mf: 3 mt: 102 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5600
```

Figure 5.7 Example output for processing data blocks from resonance region

After processing the blocks of data from the resonance region, Subroutine FILE3 in POLIDENT reads the cross-section data from File 3 and combines the appropriate data with the cross-section functions from the resonance region. A complete description of FILE3 is provided in Section 3. In comparison with BLOCKR, a similar combining procedure is used in FILE3 to complete the construction of the continuous-energy cross-section function for each reaction in File 3. In particular, FILE3 evaluates the cross-section information from File 3 and the corresponding cross-section function from the resonance region. FILE3 merges the two functions and treats any discontinuities that may exist. Figure 5.8 provides an example output for combining the cross-section data from the resonance region with the File 3 data. For each reaction, the output includes the number of points and energy boundaries of each function that is used to form the final cross-section function. During the combining process, any discontinuities are identified with the energy and cross-section values at the point of discontinuity. The resulting energy and cross-section values that are used in the final function is noted. After processing the reaction, the number of points and interpolation information are provided for each reaction. To complete each reaction, a description of the TAB1 record that includes MAT, MF and MT values for the complete function with the total number of points is provided. Likewise, the atomic-weight ratio, ZA number, temperature and background cross-section value are also provided for the final function.

```
1processing material: 9443
==========================================================================
        process:   1
 original function has   108 points and integrates to 1.31704E+08
 expanded function has   389 points and integrates to 1.31704E+08
          relative difference in functions is: 0.00000E+00
                     tabular function with    389 points being numerically doppler broadened to 0.00000E+00k
 --------->add in resonance data
 combining panels with   389 and  5600 points
 with interpolation tables of    1 and    1 points
 first function from 1.00000E-05 to 2.00000E+07
 second function from 1.00000E-05 to 4.02000E+04
 combining range is 1.00000E-05 to 2.00000E+07
                        discontinuity=======ktype 0
                        xa1: 4.02000E+04 ya1: 0.00000E+00
                        xa2: 4.02000E+04 ya2: 1.36000E+01
                        xb1: 4.02000E+04 yb1: 1.34415E+01
                        xb2: 4.02000E+04 yb2: 0.00000E+00
                         v2: 1.36000E+01 el: 4.02000E+04 eh: 4.02000E+04
 a function with  5987 points and  1 interpolation regions has been produced
 error flag is 0
 ==========================================================================
 --------->a tab1 record has been written on logical 31
          for mat:9443 mf: 3 mt:   1 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5987
 ==========================================================================

        process:   2
 original function has   108 points and integrates to 7.19737E+07
 expanded function has   389 points and integrates to 7.19737E+07
 relative difference in functions is: 0.00000E+00
          tabular function with    389 points being numerically doppler broadened to 0.00000E+00k
 --------->add in resonance data
 combining panels with   389 and  5551 points
 with interpolation tables of    1 and    1 points
 first function from 1.00000E-05 to 2.00000E+07
 second function from 1.00000E-05 to 4.02000E+04
 combining range is 1.00000E-05 to 2.00000E+07
                        discontinuity=======ktype 0
                        xa1: 4.02000E+04 ya1: 0.00000E+00
                        xa2: 4.02000E+04 ya2: 1.06870E+01
                        xb1: 4.02000E+04 yb1: 1.05299E+01
                        xb2: 4.02000E+04 yb2: 0.00000E+00
                         v2: 1.06870E+01 el: 4.02000E+04 eh: 4.02000E+04
 a function with  5938 points and  1 interpolation regions has been produced
 error flag is 0
 ==========================================================================
 --------->a tab1 record has been written on logical 31
          for mat:9443 mf: 3 mt:   2 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5938
 ==========================================================================
                              .
                              .
                              .
                              .
                  process: 102
 original function has   108 points and integrates to 4.03014E+05
 expanded function has   389 points and integrates to 4.03014E+05
 relative difference in functions is: 0.00000E+00
 tabular function with    389 points being numerically doppler broadened to 0.00000E+00k
 --------->add in resonance data
 combining panels with   389 and  5600 points
 with interpolation tables of    1 and    1 points
 first function from 1.00000E-05 to 2.00000E+07
 second function from 1.00000E-05 to 4.02000E+04
 combining range is 1.00000E-05 to 2.00000E+07
                        discontinuity=======ktype 0
                        xa1: 4.02000E+04 ya1: 0.00000E+00
                        xa2: 4.02000E+04 ya2: 4.83190E-01
                        xb1: 4.02000E+04 yb1: 4.85545E-01
                        xb2: 4.02000E+04 yb2: 0.00000E+00
                         v2: 4.83190E-01 el: 4.02000E+04 eh: 4.02000E+04
 a function with  5987 points and  1 interpolation regions has been produced
 error flag is 0
 ==========================================================================
 --------->a tab1 record has been written on logical 31
          for mat:9443 mf: 3 mt: 102 awr: 2.390E+02 za: 9.42410E+04
          t: 0.000E+00 sig0: 1.000E+10 interp reg:  1 points:  5987
```

Figure 5.8 Example output for combining resonance-region cross sections with background data

5.7 Termination of Output File

The remaining portion of the output file identifies the completion of the calculation. Once the last reaction is processed, a final message is printed. If the problem has executed successfully, the final message "POLIDENT has terminated normally" is printed in the last line of the output file.

6 MESSAGES

POLIDENT provides warning and error messages that indicate problems with program execution. The warning messages may indicate a possible error; however, the code can continue the calculation despite the message. The user should evaluate the warning message to determine if a problem exists. When an error is encountered, the code prints the appropriate message and stops execution if the error is too severe. The warning and error messages that are presented in this section may provide an underscore _____ to indicate the appropriate data.

6.1 Warning Messages

COMB2:

butted functions don't agree at join
value of "a" function at _____ is _____
value of "b" function at _____ is _____

In COMB2 **a** and **b** are two functions that are to be combined, and the last energy point in function **a** equals the first energy point in **b**; however, the cross-section values at the common energy point do not agree. This message is not necessarily an error, but the code alerts the user to the discontinuity.

butted functions don't agree at join
value of "b" function at _____ is _____
value of "a" function at _____ is _____

In COMB2 **a** and **b** are two functions that are to be combined, and the last energy point in function **b** equals the first energy point in **a**; however, the cross-section values at the common energy point do not agree. This message is not necessarily an error, but the code alerts the user to the discontinuity.

function ____ interpolation panel ____ is bad ==> _____

The interpolation code or type for the function is not between 1 and 5.

in function ____ the break point in panel ____ is less than the previous one

The end of an interpolation panel in a function is less than a preceding panel. In particular, nbt(i) is less than or equal to nbt(i-1).

the final interpolation panel in function ____ should be ____ not ____

The last interpolation panel does not correspond to the last energy point in the function. In particular, nbt(n) is not equal to n.

RESN:

this evaluation uses SLBW parameters.
MLBW processing will be FORCED!

If the MLBW flag is greater than 0 in the 6$ block of the input, single-level Breit-Wigner evaluations will be processed with the multi-level Breit-Wigner formulae. The code alerts the user to the change.

resonance calculation is not performed because the restart option is specified
calculated cross-sections will be read from logical unit ____

A restart unit has been specified in the 1$ block of the input. Therefore, POLIDENT will attempt to read the calculated cross sections from the restart unit. While processing an evaluation, a problem could be encountered following the resonance calculation (e.g., storage limit exceeded, parameter index exceeded, etc.). The cross sections from the resonance calculation are typically stored on scratch device 18. Therefore, the user can rename logical unit 18 to a restart unit (LOGRES). After correcting the input problem, the cross sections from the resonance region can be read from the restart unit as opposed to repeating the resonance calculation.

TMF2:

adjusting nismax up to ____

While reading the ENDF File 2 information for the requested material, the number of isotopes in the ENDF file exceeds the default value of 7. This message indicates that POLIDENT will increase the limit and attempt to process the material.

adjusting nermax up to ____

While reading the ENDF File 2 information for the requested material, the number of energy ranges for the isotope has been exceeded. This message indicates that POLIDENT will increase the limit and attempt to process the material.

adjusting nlsmax up to ____

While reading the ENDF File 2 information for the requested material, the limit on the number of ℓ-states (i.e., angular momentum) for the isotope has been exceeded. This message indicates that POLIDENT will increase the limit and attempt to process the material.

adjusting njsmax up to ____

While reading the ENDF File 2 information for the requested material, the limit on the number of different J values (i.e., spin of the resonance) for the isotope has been exceeded. This message indicates that POLIDENT will increase the limit and attempt to process the material.

6.2 Error Messages

ADDMOR:

number of points per decade exceeded in mesh calculation
ipoints must be greater than: ____

During the fine-energy-mesh construction, the limit on the number of points (IPOINTS) in a 10-eV interval has been exceeded. By default, the limit is 3500 points. Attempt to increase the limit in the input file and resubmit the problem.

BLOCKR:

the amount of core needed in subroutine blockr exceeds ____
or
need at least ____ more words of storage in subroutine blockr

After processing the data in the resonance region, BLOCKR combines the appropriate blocks of cross-section data to form a single function. Either message indicates that the storage limit is not sufficient to complete the combination process. The core allocation should be increased accordingly in the input file.

COMB2:

you have more than 25 discontinuities in your ____ function into comb2

COMB2 checks each function for discontinuities, and the identified function has too many discontinuities.

problems in combining from ____ to ____
first function index: ____ n2: ____ 3 ordinates ____, ____, ____
2nd function index: ____ n2: ____ 3 ordinates ____, ____, ____

COMB2 is attempting to combine two functions; however, either the first or second function has errors.

A function with ____ points and ____ interpolation regions has been produced
error flag is ____

After constructing each function, COMB2 provides a summary of the function. If the error flag is zero, no errors were encountered. If the error flag is not equal to zero, problems were encountered during the combining process and the code terminates the problem.

COMBI2 and COMBI3:

COMB2 requires too many points you have only specified ____

The value of N2MAX in either COMBI2 or COMBI3 has been exceeded, and the code needs more space to construct the function. Attempt to increase N2MAX in the input file and resubmit the problem.

too many interpolation regions

The value of N1MAX in either COMBI2 or COMBI3 has been exceeded, and the code needs more space for additional interpolation regions. Attempt to increase N1MAX in the input file and resubmit the problem.

FILE1P:

mat ____ is not on logical ____

POLIDENT could not locate the desired material identifier (MATNO) on the ENDF tape that is mounted on logical-unit NDFB. Check to make sure that the correct material identifier and tape number are specified for the problem.

FILE3:

default dimensions need to be increased in sparky n1max: ____ you need at least ____
n2max: ____ you need at least ____

The values of N1MAX and/or N2MAX have been exceeded in Subroutine FILE3. Attempt to increase the limits in the input file and resubmit the problem.

library positioning error in locating the end of file 2

This message indicates that POLIDENT has trouble reading the ENDF tape. As a result, the code cannot read the cross sections from File 3 to complete the construction of the cross sections.

MESHPT:

POLIDENT needs ____ more words of storage

Subroutine MESHPT defines the array pointers for the energy-mesh calculation. This message indicates that POLIDENT needs additional storage to complete the calculation. The amount of core should be increased, and the problem should be resubmitted.

POLIDENT:

unable to get ____ words for polident

Subroutine ALOCAT is unable to get the requested storage for execution. On unix machines, the stack size may need to be increased for the problem.

RESN:

mat: ____ is not on logical ____

POLIDENT is unable to locate the requested material (MATNO) on the ENDF tape that is mounted on logical-unit NDFB. Check to make sure that the correct material identifier and tape number are specified for the problem.

TMF2:

this version of tmf2 cannot read energy dependent p(e)

Upon reading the resonance-parameter information in File 2, Subroutine TMF2 has determined that the scattering radius is not energy independent (i.e., NRO $\neq$ 0). If the scattering radius is energy dependent, the current version of POLIDENT cannot process the evaluation.

7 REFERENCES

1. "ENDF-102 Data Formats and Procedures for the Evaluated Nuclear Data File ENDF-6," BNL-NCS-44945, Rev. 10/91 (ENDF/B-VI), Brookhaven National Laboratory, October 1991.

2. N. M. Greene, W. E. Ford III, L. M. Petrie and J. W. Arwood, *AMPX-77: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Cross-section Libraries from ENDF/B-IV and/or ENDF/B-V*, ORNL/CSD/TM-283, Martin Marietta Energy Systems, Inc., Oak Ridge National Laboratory, October 1992.

3. L. C. Leal, *Resonance Analysis and Evaluation of the ^{235}U Neutron Induced Cross Sections*, Ph.D. Dissertation, University of Tennessee, Knoxville, May 1990.

4. N. M. Larson, *Updated Users' Guide for SAMMY: Multilevel R-Matrix Fits to Neutron Data Using Bayes' Equation*, ORNL/TM-9179/R4, Lockheed Martin Energy Research Corporation, Oak Ridge National Laboratory, December 1998.

5. L. C. Leal, G. de Saussure and R. B. Perez, "An R Matrix Analysis of the ^{235}U Neutron-Induced Cross Sections up to 500 eV," *Nucl. Sci. Eng.* **109**, 1–17 (1991).

6. R. N. Hwang, "A Rigorous Representation of Multilevel Cross Sections and Its Practical Applications," *Nucl. Sci. Eng.* **96**, 192–209 (1987).

7. R. E. Schenter, J. L. Baker, and R. B. Kidman, *ETOX, A Code to Calculate Group Constants for Nuclear Reactor Calculations*, BNWL-1002, Battelle Northwest Laboratory, 1962.

8. H. Henryson II, B. J. Toppel and C. G. Stenberg, *MC^2-2: A Code to Calculate Fast Neutron Spectra and Multigroup Cross Sections*, ANL-8144 (ENDF 239), Argonne National Laboratory, June 1976.

9. R. E. MacFarlane and D. W. Muir, "NJOY 97.0: Code System for Producing Pointwise and Multigroup Neutron and Phonton Cross Sections from ENDF/B Data," RSICC Code Package PSR-368, Oak Ridge National Laboratory, May 1998.

APPENDIX A

ALPHABETICAL INDEX OF SUBROUTINES

APPENDIX A

ALPHABETICAL INDEX OF SUBROUTINES

This section provides an alphabetical index of the subroutines used by POLIDENT. For each entry, a list of subroutines that call the given subroutine is provided as well as a list of subroutines called by the given subroutine.

Table A.1 Index of subroutines

Subroutine	Calling Subroutine	Called Subroutine
ADDMOR	MESHED	LRF1234R
ADLER	RESN1	SMESHA
AHALF	TESTER	
AJKU	RESU2	QUIKW
ALOCAT	POLIDENT	SPARKY
BLOCKR	SPARKY	COMB2 FSORT ISORT MOVEBK STACKR WRTAB1
CALGNU	FILE1P	
CLEAR	MESHED SMACK	
COMB2	BLOCKR FILE3	COMBI2 DISCON
COMBI2	COMB2	COMBI3 TERP1
COMBI3	COMBI2	TERP1
COMPER	LRF123 RESR	TERP1
DATE_AND_TIME	DATIM	
DATIM	LISTQA MESAGE	DATE_AND_TIME
DISCON	COMB2	TERP1

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
DTASET	FILLY	GETFILE
ECSI	GRATE	STOP
ERRTRA	STOP	
EXIT	STOP	
FFPACK	FFREAD	
FFREAD	FIDAS	FFPACK Y0TRNS
FHLPR	LISTQA MESAGE	
FIDAS	POLIDENT SPARKY	FFREAD
FILE1P	SPARKY	CALGNU
FILE3	SPARKY	COMB2 MOVE PSNLIB RTAB1 RCONT STOP THICKN WRTAB1 XCONT
FILL	SMESHA	
FILLY	POLIDENT SPARKY	DTASET
FILNAM	OPNFIL	
FINDIT	MESHED TESTER	STOP
FINDQA	LISTQA	GETARG GETFILE GETMTM
FRES	SMESHA	
FSORT	BLOCKR	
GETARG	FINDQA	

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
GETFILE	DTASET FINDQA	
GETMTM	FINDQA	
GETNAM	JOBNUM	
GRATE	THICKN	ECSI STOP
ISORT	BLOCKR	
ITYPE	COMBI2 COMBI3	
JOBNUM	LISTQA MESAGE	GETNAM
JSTIME	MESHED	
LISTQA	MESAGE	DATIM FHLPR FINDQA JOBNUM
LRF123	RESN1	COMPER MOVE NVERT STOP XMULT
LRF1234R	ADDMOR MESHED	NVERT STOP XMULT
MAXMIN	MESHED	
MAXMIN2	MESHED	
MESAGE	POLIDENT	DATIM FHLPR JOBNUM LISTQA

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
MESHED	SMACK	ADDMOR CLEAR FINDIT JSTIME LRF1234R MAXMIN MAXMIN2 ORDER ORDER2 ORDER3 SCRUB SLICED STOP TESTER
MESHPT	SMACK	
MOVE	FILE3 LRF123 RESR	
MOVEBK	BLOCKR	
NVERT	LRF123 LRF1234R	
OPNFIL	POLIDENT RESTARTR SPARKY	FILNAM
ORDER	MESHED	
ORDER2	MESHED	
ORDER3	MESHED	
POLIDENT		ALOCAT FIDAS FILLY MESAGE OPNFIL UWTAB
PSNLIB	FILE3	
QUIKW	AJKU	UWTAB
RCONT	FILE3 TMF2	

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
RDENDF	TMF2	
RESN	SPARKY	RESN1 TMF2
RESN1	RESN	ADLER LRF123 RESR RESU SMACK
RESR	RESN1	COMPER MOVE STOP
RESU	RESN1	RESU1 RESU2
RESU1	RESU	
RESU2	RESU	AJKU TERPO TERPN
RESTARTR	SPARKY	OPNFIL
RHEAD	TMF2	
RLIST	TMF2	
RTAB1	FILE3	
SCRUB	MESHED	
SLICED	MESHED	
SMACK	RESN1	CLEAR MESHED MESHPT
SMESHA	ADLER	FILL FRES

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
SPARKY	ALOCAT	BLOCKR FIDAS FILE1P FILE3 FILLY OPNFIL RESN RESTARTR
STACKR	BLOCKR	
STOP	ECSI FILE3 FINDIT GRATE LRF123 LRF1234R MESHED RESR	ERRTRA EXIT
TERPN	RESU2	
TERPO	RESU2	
TERP1	COMPER COMBI2 COMBI3 DISCON THICKN	STOP
TESTER	MESHED	AHALF FINDIT
THICKN	FILE3	GRATE TERP1
TMF2	RESN	RCONT RDENDF RHEAD RLIST
UW	UWTAB	
UWTAB	POLIDENT QUIKW	UW
WRTAB1	BLOCKR FILE3	

Table A.1 (continued)

Subroutine	Calling Subroutine	Called Subroutine
XCONT	FILE3	
XMULT	LRF123 LRF1234R	
Y0TRNS	FFREAD	

APPENDIX B

SAMPLE PROBLEMS

APPENDIX B

SAMPLE PROBLEMS

This appendix contains sample problems that demonstrate the procedures to execute POLIDENT. Three sample problems are presented that demonstrate the features of POLIDENT. In particular, two sample input files are presented for ^{23}Na and ^{235}U, and the third input file is used to process C and Hf. Because the sample problems were developed in a unix environment, the input for each sample problem is provided using lower-case letters. In addition, the FIDO input method, which is described in Appendix D, is used to specify the input options for each sample problem.

Sample Problem 1:

The first sample case considers the ENDF/B-VI evaluation for ^{23}Na (i.e., MAT = 1125). The objective of the sample problem is to generate continuous-energy cross sections at 0 K from the ENDF data in Files 2 and 3. The following is a sample POLIDENT input file for ^{23}Na:

```
=polident
0$$ 31 e
1$$ 1 t
2$$ 1125 11 2 6
4** a5 0.001 e
6$$ a3 0 e t
end
```

In the above input deck, the cross sections that are calculated by POLIDENT are stored in a TAB1 format on unit 31, as specified in the 0$ array. Moreover, the 1$ array specifies that a single nuclide/isotope is processed, and the 2$ array specifies the ^{23}Na material identifier (1125), the logical-unit number for the ENDF/B library (11), the format of the ENDF tape (BCD) and the ENDF version number (6). In the 4* array, the fifth parameter specifies an energy-mesh-convergence tolerance for the resolved-resonance region of 0.1%. The convergence tolerance in the 4* array is the user-specified or auxiliary-grid-convergence tolerance that is described in Section 2.2.1.

Sample Problem 2:

The second example problem considers the ENDF/B-VI evaluation for ^{235}U (i.e., MAT = 9228), and the following input deck can be used to calculate the continuous-energy cross sections at 0 K:

```
=polident
-1$$ 3000000
0$$ 31 e
1$$ 1 t
2$$ 9228 11 2 6
4** a5 0.001 e
6$$ a3 0 5000 t
end
```

For the ^{235}U evaluation, the –1$ array is used to increase the core allocation from the default value of 1,000,000 words to 3,000,000 words. The fourth parameter in the 6$ array is used to specify the maximum number of points in a 10-eV interval.

Sample Problem 3:

The third example problem considers the ENDF/B-VI evaluations for C and Hf (i.e., MAT = 600 and 7500, respectively):

```
=polident
0$$ 31 e
1$$ 2
t
2$$  600 11 2 6
4** a5 0.001 e
6$$ a3 0 e t
2$$ 7200 12 2 6
4** a5 0.001 e
6$$ a3 0 e t
end
```

As noted in Section 4, the first block of the input deck is used to specify the core allocation, output units and number of nuclides/isotopes to process. In the above input deck, two nuclides are specified in the 1$ array. Subsequently, the second block of data is repeated two times in order to define the input for C and Hf. Additional evaluations can be processed in a single POLIDENT case; however, the cross-section output file could become very large if several different nuclides/isotopes are processed.

APPENDIX C

NUMERICAL EXPRESSIONS FOR REICH-MOORE RESOLVED-RESONANCE EQUATIONS

APPENDIX C

NUMERICAL EXPRESSIONS FOR REICH-MOORE RESOLVED-RESONANCE EQUATIONS

Equations are presented in Section 2.1.1.2 for the calculation of energy-dependent cross sections using the Reich-Moore (R-M) formalism. As noted in Section 2.1.1.2, the equations for the absorption and capture cross sections [i.e., Equations (2.27) and (2.31), respectively] can lead to the subtraction of small complex quantities that are approximately equal. Experience has revealed that numerical instability problems occur if the single- or double-precision FORTRAN complex functions are used to evaluate the absorption and capture cross sections that are defined by Equations (2.27) and (2.31), repsectively. In particular, the evaluation of the following quantity is required to calculate the absorption cross section:

$$Re\rho_{11} - |\rho_{11}|^2 \ , \tag{C.1}$$

where

Re = the real component of the complex quantity.

The evaluation of Equation (2.31) for the capture cross section requires the calculation of the following quantity:

$$Re\rho_{11} - |\rho_{11}|^2 - |\rho_{12}|^2 - |\rho_{13}|^2 \ . \tag{C.2}$$

In the following discussion, expressions are provided to calculate the quantities that are defined by Equations (C.1) and (C.2). Note that the subscript value of 1 in Equations (C.1) and (C.2) denotes the neutron channel in the R-M approximation. Moreover, the subscript values of 2 and 3 denote the first- and second-fission channels, respectively.

As noted in Section 2.1.1.2, the complex matrix ρ_{nc} is defined as follows:

$$\rho_{nc} = \left(I - (I - K)^{-1}\right)_{nc} = -\left((I - K)^{-1}K\right)_{nc} \ , \tag{C.3}$$

where

I = the identity matrix,

and

$$(I - K)_{nc} = \delta_{nc} - \frac{i}{2}\sum_r \frac{\Gamma_{nr}^{1/2}\,\Gamma_{cr}^{1/2}}{(E_r - E) - i\dfrac{\Gamma_{\gamma r}}{2}} \ . \tag{C.4}$$

In Equations (C.3) and (C.4), n and c represent the entrance and exit channels, respectively. In Equation (C.3), the quantity ρ_{nc} is a $n \times c$ matrix. For neutron-induced reactions in the Reich-Moore formalism, ρ_{nc} is a 3×3 matrix. Likewise, the matrix quantities $(I - K)^{-1}$ and K can be defined as follows:

$$A = (I - K)^{-1} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \tag{C.5}$$

and

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}. \tag{C.6}$$

The equation for the matrix ρ_{nc} is obtained by multiplying $(I - K)^{-1}$ and K:

$$-\rho_{nc} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \times \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}. \tag{C.7}$$

In order to evaluate the expressions in Equations (C.1) and (C.2), the quantities ρ_{11}, ρ_{12} and ρ_{13} must be calculated. Based on the matrix multiplication in Equation (C.7) and noting that K is a symmetric matrix, the following expressions are obtained for ρ_{11}, ρ_{12} and ρ_{13}:

$$-\rho_{11} = A_{11}K_{11} + A_{12}K_{12} + A_{13}K_{13}\ , \tag{C.8}$$

$$-\rho_{12} = A_{11}K_{12} + A_{12}K_{22} + A_{13}K_{23}\ , \tag{C.9}$$

$$-\rho_{13} = A_{11}K_{13} + A_{12}K_{23} + A_{13}K_{33}\ . \tag{C.10}$$

The individual elements of the K matrix can be obtained from Equation (C.4); however, the elements of the A matrix require some additional effort.

In order to evaluate the A matrix, the matrix quantity $(I - K)$ that is defined by Equation (C.4) can be expressed as follows:

$$B = (I - K) = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} 1-K_{11} & -K_{12} & -K_{13} \\ -K_{21} & 1-K_{22} & -K_{23} \\ -K_{31} & -K_{32} & 1-K_{33} \end{bmatrix} . \tag{C.11}$$

Since A is the inverse of B, the following identity must be true:

$$AB = I . \tag{C.12}$$

Equation (C.12) leads to a set of nine equations with nine unknowns. Using the algebraic expressions from Equation (C.12), the following equations are obtained for the quantities A_{11}, A_{12} and A_{13}:

$$A_{11} = \frac{(1 - K_{22})(1 - K_{33}) - K_{23}^2}{DEN} , \tag{C.13}$$

$$A_{12} = \frac{K_{13}K_{23} + K_{12}(1 - K_{33})}{DEN} , \tag{C.14}$$

$$A_{13} = \frac{K_{12}K_{23} + K_{13}(1 - K_{22})}{DEN} , \tag{C.15}$$

where

$$\begin{aligned} DEN = {} & (1 - K_{11})(1 - K_{22})(1 - K_{33}) - 2K_{12}K_{13}K_{23} \\ & - K_{23}^2(1 - K_{11}) - K_{12}^2(1 - K_{33}) - K_{13}^2(1 - K_{22}) . \end{aligned} \tag{C.16}$$

The elements of the K matrix are defined in Table C.1.

Table C.1 Definitions of the elements of the K matrix[a]

$K_{11} = -\frac{1}{2}(B' - iA')$	$A' = \sum_{r=1}^{NR} \frac{\Gamma_{1r}\, a}{a^2 + b^2}$	$B' = \sum_{r=1}^{NR} \frac{\Gamma_{1r}\, b}{a^2 + b^2}$
$K_{12} = -\frac{1}{2}(D - iC)$	$C = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{1r}}\sqrt{\Gamma_{2r}}\, a}{a^2 + b^2}$	$D = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{1r}}\sqrt{\Gamma_{2r}}\, b}{a^2 + b^2}$
$K_{13} = -\frac{1}{2}(F - iE)$	$E = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{1r}}\sqrt{\Gamma_{3r}}\, a}{a^2 + b^2}$	$F = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{1r}}\sqrt{\Gamma_{3r}}\, b}{a^2 + b^2}$
$K_{22} = -\frac{1}{2}(N - iM)$	$M = \sum_{r=1}^{NR} \frac{\Gamma_{2r}\, a}{a^2 + b^2}$	$N = \sum_{r=1}^{NR} \frac{\Gamma_{2r}\, b}{a^2 + b^2}$
$K_{23} = -\frac{1}{2}(H - iG)$	$G = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{2r}}\sqrt{\Gamma_{3r}}\, a}{a^2 + b^2}$	$H = \sum_{r=1}^{NR} \frac{\sqrt{\Gamma_{2r}}\sqrt{\Gamma_{3r}}\, b}{a^2 + b^2}$
$K_{33} = -\frac{1}{2}(P - iO)$	$O = \sum_{r=1}^{NR} \frac{\Gamma_{3r}\, a}{a^2 + b^2}$	$P = \sum_{r=1}^{NR} \frac{\Gamma_{3r}\, b}{a^2 + b^2}$

[a]Note that $a = E_r - E$, $b = \Gamma_\gamma/2$, r = resonance index, and NR = number of resonances.

An equation for ρ_{11} is obtained by substituting the expressions for A_{11}, A_{12} and A_{13} into Equation (C.8). Using the expressions for the elements of the K matrix in Table C.1, the equation for ρ_{11} is expressed as follows:

$$\rho_{11} = \frac{\alpha(2\psi + \alpha) + \gamma(2\chi + \gamma)}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} + i\,\frac{\alpha(2\chi + \gamma) - \gamma(2\psi + \alpha)}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2}\ , \tag{C.17}$$

where

$$\alpha = (B'x - A'y) - (B'u - A'v) + 2w - q - s\ , \tag{C.18}$$

$$\gamma = (B'y + A'x) - (B'v - A'u) + 2z - r - t\ , \tag{C.19}$$

$$\psi = x - u\ , \tag{C.20}$$

$$\chi = y - v\ , \tag{C.21}$$

$$x = (2 + N)(2 + P) - OM\ , \tag{C.22}$$

$$y = O(2 + N) + M(2 + P)\ , \tag{C.23}$$

$$w = D(FH - EG) - C(FG + EH)\ , \tag{C.24}$$

$$z = D(FG + EH) + C(FH - EG)\ , \tag{C.25}$$

$$q = (2 + P)(D^2 - C^2) - 2CDO\ , \tag{C.26}$$

$$r = 2CD(2 + P) + O(D^2 - C^2)\ , \tag{C.27}$$

$$s = (2 + N)(F^2 - E^2) - 2EFM\ , \tag{C.28}$$

$$t = 2EF(2 + N) + M(F^2 - E^2)\ , \tag{C.29}$$

$$u = (H^2 - G^2)\ , \tag{C.30}$$

$$v = 2GH\ . \tag{C.31}$$

Using the expression for ρ_{11} in Equation (C.17), an equation can be developed for $|\rho_{11}|^2 = \rho_{11}\,(\rho_{11})^*$:

$$|\rho_{11}|^2 = \frac{[\ \alpha(2\psi + \alpha) + \gamma(2\chi + \gamma)\ ]^2 + [\ \alpha(2\chi + \gamma) - \gamma(2\psi + \alpha)\ ]^2}{[\ (2\psi + \alpha)^2 + (2\chi + \gamma)^2\ ]^2} \ . \tag{C.32}$$

An equation for the quantity $Re\rho_{11} - |\rho_{11}|^2$ is obtained by substituting Equations (C.17) and (C.32) into Equation (C.1). After simplification, the following expression is obtained for $Re\rho_{11} - |\rho_{11}|^2$:

$$Re\rho_{11} - |\rho_{11}|^2 = \frac{2(\psi\alpha + \chi\gamma)}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} \ . \tag{C.33}$$

In order to calculate the quantity that is defined by Equation (C.2), expressions must be developed for $|\rho_{12}|^2$ and $|\rho_{13}|^2$. An expression for ρ_{12} is obtained by substituting Equations (C.13) through (C.15) into Equation (C.9). After the substitution and algebraic simplifications, the following expression is obtained for ρ_{12}:

$$\rho_{12} = \frac{-[\ f(2\psi + \alpha) - g(2\chi + \gamma)\ ]}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} - i\,\frac{[\ f(2\chi + \gamma) + g(2\psi + \alpha)\ ]}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} \ , \tag{C.34}$$

where

$$f = \chi C - \psi D + M\lambda_2 - N\lambda_1 + G\lambda_4 - H\lambda_3 \ , \tag{C.35}$$

$$g = \psi C + \chi D + M\lambda_1 + N\lambda_2 + G\lambda_3 + H\lambda_4 \ , \tag{C.36}$$

$$\lambda_1 = (FH - EG) - [\ D(2 + P) - CO\ ] \ , \tag{C.37}$$

$$\lambda_2 = (FG + EH) - [\ DO + C(2 + P)\ ] \ , \tag{C.38}$$

$$\lambda_3 = (DH - CG) - [\ F(2 + N) - EM\ ] \ , \tag{C.39}$$

$$\lambda_4 = (DG + CH) - [\ FM + E(2 + N)\ ] \ . \tag{C.40}$$

Using the expression for ρ_{12} in Equation (C.34), an equation can be developed for $|\rho_{12}|^2$:

$$|\rho_{12}|^2 = \frac{[\ f(2\psi + \alpha) - g(2\chi + \gamma)\ ]^2 + [\ f(2\chi + \gamma) + g(2\psi + \alpha)\ ]^2}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} \ . \tag{C.41}$$

An expression for ρ_{13} is obtained by substituting Equations (C.13) through (C.15) into Equation (C.10). After the substitution and algebraic simplifications, the following expression is obtained for ρ_{13}:

$$\rho_{13} = \frac{-[\ h(2\psi + \alpha) - k(2\chi + \gamma)\ ]}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} - i\ \frac{[\ h(2\chi + \gamma) + k(2\psi + \alpha)\ ]}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2}\ , \tag{C.42}$$

where

$$h = \chi E - \psi F + G\lambda_2 - H\lambda_1 + O\lambda_4 - P\lambda_3\ , \tag{C.43}$$

$$k = \psi E + \chi F + G\lambda_1 + H\lambda_2 + O\lambda_3 + P\lambda_4\ . \tag{C.44}$$

After multiplying ρ_{13} by the complex conjugate, the following expression is obtained for $|\rho_{13}|^2$:

$$|\rho_{13}|^2 = \frac{[\ h(2\psi + \alpha) - k(2\chi + \gamma)\ ]^2 + [\ h(2\chi + \gamma) + k(2\psi + \alpha)\ ]^2}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2}\ . \tag{C.45}$$

An equation for the quantity $Re\rho_{11} - |\rho_{11}|^2 - |\rho_{12}|^2 - |\rho_{13}|^2$ is obtained by substituting Equations (C.33), (C.41) and (C.45) into Equation (C.2):

$$Re\rho_{11} - |\rho_{11}|^2 - |\rho_{12}|^2 - |\rho_{13}|^2 = \frac{2(\psi\alpha + \chi\gamma) - (f^2 + g^2 + h^2 + k^2)}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2}\ . \tag{C.46}$$

For resonance parameters that are expressed in the Reich-Moore formalism, Equation (C.33) can be substituted into Equation (2.27) to calculate the absorption cross section, and Equation (C.46) can be substituted into Equation (2.31) to calculate the capture cross section as a function of energy.

In Section 2.1.1.2, Equation (2.23) provides an expression for calculating the elastic-scattering cross section as a function of energy. The equation for elastic scattering is repeated in this appendix for further discussion:

$$\sigma_{el}(E) = \frac{4\pi}{k^2} \sum_J g_J \left\{ \sin^2\phi_1(1 - 2\ Re\rho_{11}) + \sin(2\phi_1)\ Im\rho_{11} + |\rho_{11}|^2 \right\}\ . \tag{C.47}$$

As noted in Section 2.1.1.2, the equation for the elastic scattering cross section [i.e., Equation (C.47)] does not account for all of the channel-spin terms in the potential scattering, and Equation (C.47) can be recast into a different form that includes an explicit summation over the channel spin, j:

$$\sigma_{el}(E) = \frac{4\pi}{k^2} \left\{ \sum_J g_J \left\{ -2\sin^2\phi_1 \; Re\rho_{11} + \sin(2\phi_1) \; Im\rho_{11} + |\rho_{11}|^2 \right\} + \sum_{j=\left|SPI-\frac{1}{2}\right|}^{SPI+\frac{1}{2}} \sum_{J=|\ell-j|}^{\ell+j} g_J \sin^2\phi_1 \right\} . \tag{C.48}$$

Note that Equation (C.48) is the same equation that is documented as Equation (2.28) in Section 2.1.1.2.

In POLIDENT, Equation (C.48) is used to calculate the elastic-scattering cross section as a function of energy; however, POLIDENT initially calculates $\sigma_{el}(E)$ using Equation (C.47) and adds the missing potential scattering contributions from the channel-spin terms that have not been included in the cross-section calculation. As a result, the calculation of the elastic-scattering cross section is represented by Equation (C.48). This discussion is provided to highlight an additional numerical instability problem that occurs in the calculation of $\sigma_{el}(E)$ using Equation (C.47). In particular, if the single- or double-precision FORTRAN complex functions are used to evaluate $\sigma_{el}(E)$, a numerical instability can occur in Equation (C.47) under the following condition:

$$\sin^2\phi_1 \approx -2\sin^2\phi_1 \; Re\rho_{11} + \sin(2\phi_1) \; Im\rho_{11} + |\rho_{11}|^2 . \tag{C.49}$$

To circumvent this problem, an expression can be developed for the quantity within braces in Equation (C.47). In particular, the following definition can be used to express the quantity within braces in Equation (C.47):

$$qty = \sin^2\phi_1 - 2\sin^2\phi_1 \; Re\rho_{11} + \sin(2\phi_1) \; Im\rho_{11} + |\rho_{11}|^2 . \tag{C.50}$$

Using the expressions for ρ_{11} and $|\rho_{11}|^2$ from Equations (C.17) and (C.32), respectively, an equation can be developed for *qty*. After substituting Equations (C.17) and (C.32) into Equation (C.50) and simplifying the expression, the following equation is obtained for *qty*:

$$qty = \frac{2[\; 2\sin^2\phi_1(\psi^2 + \chi^2) + \sin 2\phi(\alpha\chi - \gamma\psi) \;] + (\alpha^2 + \gamma^2)(1 - \sin^2\phi_1)}{(2\psi + \alpha)^2 + (2\chi + \gamma)^2} . \tag{C.51}$$

In POLIDENT, the elastic-scattering cross section is calculated by using Equation (C.51) to calculate the quantity within braces in Equation (C.47). Subsequently, POLIDENT determines the missing channel-spin contributions to the potential scattering and adds the missing contributions to $\sigma_{el}(E)$ as defined by Equation (C.48).

APPENDIX D

FIDO INPUT

APPENDIX D

FIDO INPUT

D.1 INTRODUCTION

The FIDO input method is specially devised to allow entering or modifying large data arrays with minimum effort. Advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos and was first applied to the DTF-II code. Since that time, numerous features requested by users have been added, a free-field option has been developed, and the application of FIDO has spread to innumerable codes.

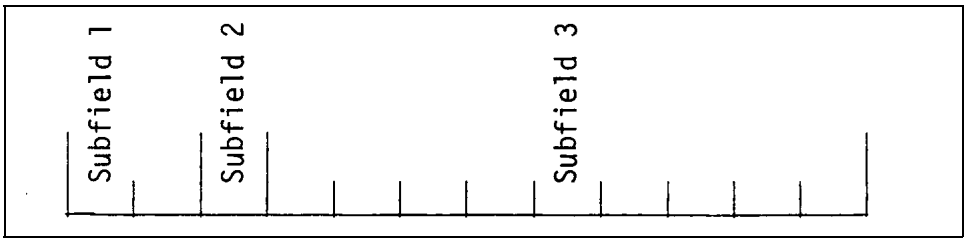
The data are entered in units called "arrays." An array comprises a group of contiguous storage locations that are to be filled with data at the same time. These arrays usually correspond on a one-to-one basis with FORTRAN arrays used in the program. A group of one or more arrays read with a single call to the FIDO package forms a "block," and a special delimiter is required to signify the end of each block. Arrays within a block may be read in any order with respect to each other, but an array belonging to one block must not be shifted to another. The same array can be entered repeatedly within the same block. For example, an array could be filled with "0" using a special option, and then a few scattered locations could be changed by reading in a new set of data for that array. If no entries to the arrays in a block are required, the delimiter alone satisfies the input requirement.

Three major types of input are available: fixed-field input, free-field input, and user-field input.

D.2 FIXED-FIELD INPUT

The fixed-field input option is documented here for completeness. **The use of fixed-field input is NOT recommended. Use the free-field input option documented in Section D.3.**

Each card is divided into six 12-column data fields, each of which is divided into three subfields. The following sketch illustrates a typical data field. The three subfields always comprise 2, 1, and 9 columns, respectively.



To begin the first array of a block, an <u>array originator field</u> is placed in any field on a card:

Subfield 1: An integer <u>array identifier</u> <100 specifying the data array to be read in.

Subfield 2: An <u>array-type indicator</u>:

"$" if the array is integer data
"*" if the array is real data
"#" if the array is double-precision data

Subfield 3: Blank

Data are then placed in successive fields until the required number of entries have been located. A sample data sheet, shown in Table D.1, illustrates this point.

In entering data, it is convenient to think of an "index" or "pointer" as a designator that is under the control of the user and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at array location No. 1 by entering the array originator field. The pointer subsequently moves according to the data operator chosen. Blank fields are a special case in that they do not cause any data modification and do not move the pointer.

A data field has the following form:

Subfield 1: The data numerator, an integer <100. We refer to this entry as N_1 in the following discussion.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to be read in F9.0 format. It will be converted to an integer if the array is a "$" array or if a special array operator, such as Q, is being used. Note that an exponent is permissible but not required. Likewise, a decimal is permissible but not required. If no decimal is supplied, it is assumed to be immediately to the left of the exponent, if any; and, otherwise, to the right of the last column. This entry is referred to as N_3 in the following discussion.

A list of data operators and their effect on the array being input follows:

"Blank" indicates a single entry of data. The data entry in the third subfield is entered in the location indicated by the pointer, and the pointer is advanced by 1. However, an entirely blank field is ignored.

"+" or "–" indicates exponentiation. The data entry in the third field is entered and multiplied by $10^{\pm N_1}$, where N_1 is the data numerator in the first subfield, given the sign indicated by the data operator itself. The pointer advances by 1. In cases where an exponent is needed, this option allows the entering of more significant figures than the blank option.

"&" has the same effect as "+".

"R" indicates that the data entry is to be repeated N_1 times. The pointer advances by N_1.

Table D.1 General example of FIDO input

Name General Example of FIDO Input **Charge** _____ **Date** _____ **Page** _____

			IDENTIFICATION	REMARKS (DO NOT PUNCH)
1	1 $		73 80	Begin the 1$ array, fixed-field, integral
13		1		Enter 1
25	F	2		Fill array with 2
37	2 *			Begin the 2* array fixed-field, real
49		1 • 2 3 4		Enter 1.234
61		1 2 • 3 4 – 1	1 0	″ ″
1	5 –	1 2 3 4 + 0 2	73 80	″ ″
13	3 –	1 2 3 4		″ ″
25		7		″ 7.0
37				A blank field is always ignored
49	T			Terminate this block
61			2 0	No entries may follow T on a card
1	3 *		73 80	Begin 3* array, fixed-field real
13	9 I	0		Enter 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10
25	3 R	1 0		as real numbers
37	3 * *	1 0 S 1 0		Repeat 3* in free-field, skip
49	1 1	1 2		to 11th entry, correct sequence to
61			3 0	- - - 9, 10, 11, 12
1	4 * *	2 I 1 4 • 0	73 80	Begin 4* array, free-field, real
13	2 Q 4			Enter 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4
25		E		End reading this array; remainder of array unchanged.
37		T		Terminate this block
49				
61			4 0	

R - REPEAT I - INTERPOLATE S - SKIP T - TERMINATE

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry in the third subfield is entered, followed by Nj interpolated entries equally spaced between that value and the data entry found in the third subfield of the next nonblank field. The pointer is advanced by $N_1 + 1$. The field following an "I," field is than processed normally, according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"L" indicates logarithmic interpolation. The effect is the same as that of "I," except that the resulting data are evenly separated in log-space. This feature is especially convenient for specifying an energy mesh.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by the third subfield, N_3. The sequence of N_3 entries is to be repeated N_1 times. The pointer advances by N_1*N_3. If either N_1 or N_3 is 0, then a sequence of $N_1 + N_3$ is repeated one time only, and the pointer advances by $N_1 + N_3$. This feature is especially valuable for geometry specification.

The "N" option has the same effect as "Q," except that the order of the sequence is reversed each time it is entered. This feature is valuable for the type of symmetry possessed by S_n quadrature coefficients.

"M" has the same effect as "N," except that the sign of each entry in the sequence is reversed each time the sequence is entered. For example, the entries

1 2 3 2M2

would be equivalent to

1 2 3 -3 -2 2 3.

This option is also useful in entering quadrature coefficients.

"Z" causes $N_1 + N_3$ locations to be set at 0. The pointer is advanced by $N_1 + N_3$.

"C" causes the position of the last array entered to be printed. This is the position of the pointer, less 1. The pointer is not moved.

"O" causes the print trigger to be changed. The trigger is originally off. Successive "O" fields turn it on and off alternately. When the trigger is on, each card image is listed as it is read.

"S" indicates that the pointer is to skip N_1 positions, leaving those array positions unchanged. If the third subfield is blank, the pointer is advanced by N_1. If the third subfield is nonblank, that data entry is entered following the skip, and the pointer is advanced by $N_1 + 1$.

"A" moves the pointer to the position N_3, specified in the third subfield.

"F" fills the remainder of the array with the datum entered in the third subfield.

"E" skips over the remainder of the array. The array-length criterion is always satisfied by an E, no matter how many entries have been specified. No more entries to an array may be given following an "E," except that data entry may be restarted with an "A."

The reading of data to an array is terminated when a new array origin field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given; and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array origin was read. Otherwise, control is returned to the calling program.

A block termination consists of a field having "T" in the second subfield. Entries following "T" on a card are ignored, and control is returned from FIDO to the calling program.

Comment cards can be entered within a block by placing an apostrophe (') in column 1. Then columns 2–80 will be listed, with column 2 being used for printer carriage control. Such cards have no effect on the data array or pointer.

D.3 FREE-FIELD INPUT

With free-field input, data are written without fixed restrictions as to field and subfield size and positioning on the card. The options used with fixed-field input are available, although some are slightly restricted in form. In general, fewer data cards are required for a problem, the interpreting print is easier to read, a card listing is more intelligible, the cards are easier to keypunch, and certain common keypunch errors are tolerated without affecting the problem. Data arrays using fixed- and free-field input can be intermingled at will within a given block,

The concept of three subfields per field is still applicable to free-field input; but if no entry for a field is required, no space for it need be left. Only columns 1–72 may be used, as with fixed-field input. A field may not be split across cards.

The array originator field can begin in any position. The array identifiers and type indicators are used as in fixed-field input. The type indicator is entered twice to designate free-field input (i.e., "$$," "**," or "##"). The blank third subfield required in fixed-field input is not required. For example,

31**

indicates that array 31, a real-data array, will follow in free-field format.

Data fields may follow the array origin field immediately. The data field entries are identical to the fixed-field entries with the following restrictions:

1. Any number of blanks may separate fields, but at least one blank must follow a third subfield entry if one is used.

2. If both first- and second-subfield entries are used, no blanks may separate them (i.e., 24S, but not 24 S).

3. Numbers written with exponents must not have imbedded blanks (i.e., 1.0E+4, 1.0-E4, 1.0+4, or even 1+4, but *not* 1.0 E4). A zero should never be entered with an exponent. For example, 0.00-5 or 0.00E-5 will be interpreted as -5×10^{-2}.

4. In third-subfield data entries only 9 digits, including the decimal but not including the exponent field, can be used (i.e., 123456.89E07, but *not* 123456.789E07).

5. The Z entry must be of the form: 738Z, *not* Z738 or 738 Z.

6. The + or - data operators are not needed and are not available.

7. The Q, N, and M entries are restricted: 3Q4, 1N4, M4, but *not* 4Q, 4N, or 4M.

D.4 USER-FIELD INPUT

If the user follows the array identifier in the array originator field with the character "U" or "V," the input format is to be specified by the user. If "U" is specified, the FORTRAN FORMAT to be used must be supplied in columns 1–72 of the next card. The format must be enclosed by the usual parentheses. Then the data for the entire array must follow on successive cards. The rules of ordinary FORTRAN input as to exponents, blanks, etc., apply. If the array data do not fill the last card, the remainder must be left blank.

"V" has the same effect as "U," except that the format read in the last preceding "U" array is used.

D.5 CHARACTER INPUT

If the user wishes to enter character data into an array, at least three options are available. The user may specify an arbitrary format using a "U" and reading in the format. The user may follow the array identifier by a "/". The next two entries into subfield 3 specify the beginning and ending indices in the array into which data will be read. The character data are then read starting with the next data card in an 18A4 format.

Finally, the user may specify the array as a free-form "*" array and then specify the data entries as "nH" character data, where n specifies how many characters follow H.

INTERNAL DISTRIBUTION

1. S. M. Bowman, 6011, MS-6370
2. B. L. Broadhead, 6011, MS-6370
3–7. W. C. Carter (5), 6011, MS-6370
8. R. L. Childs, 6011, MS-6370
9. F. C. Difilippo, 6025, MS-6363
10–14. M. E. Dunn (5), 6011, MS-6370
15. M. D. DeHart, 6011, MS-6370
16. K. R. Elam, 6011, MS-6370
17. R. J. Ellis, 6025, MS-6363
18. M. B. Emmett, 6011, MS-6370
19. C. Y. Fu, 6011, MS-6370
20. I. C. Gauld, 6011, MS-6370
21. J. C. Gehin, 6025, MS-6363
22. S. Goluoglu, 6025, MS-6370
23–26. N. M. Greene (4), 6011, MS-6370
27. D. F. Hollenbach, 6011, MS-6370
28. C. M. Hopper, 6011, MS-6370
29. B. L. Kirk, 6025, MS-6363
30. M. A. Kuliasha, 6025, MS-6435
31. N. M. Larson, 6011, MS-6370
32. L. C. Leal, 6011, MS-6370
33. B. D. Murphy, 6011, MS-6370
34–37. C. V. Parks (4), 6011, MS-6370
38. L. M. Petrie, 6011, MS-6370
39. R. T. Primm, 6025, MS-6363
40. C. E. Pugh, 9201-3, MS-8063
41. B. T. Rearden, 6011, MS-6370
42. I. Remec, 6025, MS-6363
43. J.-P. Renier, 6025, MS-6363
44. R. W. Roussin, 6025, MS-6363
45. C. H. Shappert, 4500N, MS-6237
46. J. C. Wagner, 6011, MS-6370
47. R. M. Westfall, 6011, MS-6370
48. J. E. White, 6025, MS-6363
49. Central Research Library, Doc. Ref. Section, 4500N, MS-6191
50. ORNL Laboratory Records — RC, 4500N, MS-6285

EXTERNAL DISTRIBUTION

51. M. G. Bailey, NMSS, U.S. Nuclear Regulatory Commission, MS T8-A23, Washington, DC 20555-0001
52. R. N. Blomquist, Argonne National Laboratory, RA/208, 9700 S. Case Ave., Argonne, IL 60439-4842
53. D. E. Carlson, NMSS/SFPO/TRD, U.S. Nuclear Regulatory Commission, MS O13-D13, Washington, DC 20555-0001
54. P. Cousinou, Institute of Protection and Nuclear Safety, B. P. 6 - 92265 Fontenay-Aux-Roses, Cedex, France
55. D. R. Damon, NMSS/FCSS/FLIB, U.S. Nuclear Regulatory Commission, MS T8-A33, Washington, DC 20555-0001
56. H. L. Dodds, University of Tennessee, Nuclear Engineering Dept., 214 Pasqua Engineering Bldg., Knoxville, TN 37922

57–61. D. D. Ebert, RES/DSARE/SMSAB, U.S. Nuclear Regulatory Commission, MS T10-K8, Washington, DC 20555-0001

62. F. Eltawila, RES/DSARE/SMSAB, U.S. Nuclear Regulatory Commission, MS T10-E32, Washington, DC 20555-0001
63. H. D. Felsher, NMSS/FCSS/FLIB, U.S. Nuclear Regulatory Commission, MS T8-A33, Washington, DC 20555-0001
64. P. Finck, Argonne National Laboratory, 9700 South Cass Avenue, RA/208, Argonne, IL 60439-4842
65. E. K. Fujita, Argonne National Laboratory, 9700 South Cass Avenue, RA/208, Argonne, IL 60439-4842
66. A. S. Garcia, U.S. Department of Energy, Idaho Operations Office, 850 Energy Dr., MS 1154, Idaho Falls, ID 83401-1563
67. R. C. Little, Los Alamos National Laboratory, MS F663, P.O. Box 1663, Los Alamos, NM 87545
68. R. McBroom, U.S. Department of Energy, Oak Ridge Operations Office, YSO, P.O. Box 2001, Oak Ridge, TN 37831
69. J. McKamy, U.S. Department of Energy, EH-34, 19901 Germantown Road, Germantown, MD 20874-1290
70. R. D. McKnight, Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439-4842
71. D. E.I. Mennerdal, E M Systems, Starvãgen 12, Täby, SWEDEN S-18357
72. L. Montierth, INEEL, P.O. Box 1625, MS-3458, Idaho Falls, ID 83415-6370
73. D. C. Morey, NMSS/FCSS/FCOB, U.S. Nuclear Regulatory Commission, MS T8-A33, Washington, DC 20555-0001

74–75. Office of Scientific and Technical Information, U.S. Department of Energy, P.O. Box 62, Oak Ridge, TN 37831

76. V. A. Perin, NMSS/DWM/HLWB, U.S. Nuclear Regulatory Commission, NMSS/DWM, MS T7-F3, Washington, DC 20555-0001
77. R. E. Pevey, University of Tennessee, Nuclear Engineering Dept., 214 Pasqua Engineering Bldg., Knoxville, TN 37922
78. C. Tripp, NMSS/FCSS/FSPB, U.S. Nuclear Regulatory Commission, MS T8-A33, Washington, DC 20555-0001
79. J. J. Wagschal, Racah Institute of Physics, The Hebrew University of Jerusalem, 91904, Jerusalem, ISRAEL
80. S. A. Whaley, NMSS/SFPO/TRD, U.S. Nuclear Regulatory Commission, MS O13-D13, Washington, DC 20555-0001
81. B. H. White IV, NMSS/SFPO/TRD, U.S. Nuclear Regulatory Commission, MS O13-D13, Washington, DC 20555-0001
82. Mark Williams, Louisiana State University, Baton Rouge, LA 70803-5820
83. C. J. Withee, NMSS/SFPO/TRD, U.S. Nuclear Regulatory Commission, MS O13-D13, Washington, DC 20555-0001

NRC FORM 335 (2-89)
NRCM 1102
3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)

NUREG/CR-6694
ORNL/TM-2000/035

2. TITLE AND SUBTITLE

POLIDENT: A Module for Generating Continuous Energy Cross Sections from ENDF Resonance Data

3. DATE REPORT PUBLISHED

MONTH	YEAR
December	2000

4. FIN OR GRANT NUMBER
W6479

5. AUTHOR(S)

M. E. Dunn and N. M. Greene

6. TYPE OF REPORT
Technical

7. PERIOD COVERED ***(Inclusive Dates)***

8. PERFORMING ORGANIZATION — NAME AND ADDRESS ***(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)***

Oak Ridge National Laboratory
Managed by UT-Battelle, LLC
Oak Ridge, TN 37831-6370

9. SPONSORING ORGANIZATION — NAME AND ADDRESS ***(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Regulatory Commission, and mailing address.)***

Division of Systems Analysis and Regulatory Effectiveness
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

D. D. Ebert, NRC Project Manager

11. ABSTRACT ***(200 words or less)***

POLIDENT (POint LIbraries of Data from ENDF/B Tapes) is an AMPX module that accesses the resonance parameters from File 2 of an ENDF/B library and constructs the continuous-energy cross sections in the resonance energy region. The cross sections in the resonance range are subsequently combined with the File 3 background data to construct the cross-section representation over the complete energy range. POLIDENT has the capability to process all resonance reactions that are identified in File 2 of the ENDF/B library. In addition, the code has the capability to process the single- and multi-level Breit-Wigner, Reich-Moore and Adler-Adler resonance formalisms that are identified in File 2. POLIDENT uses a robust energy-mesh-generation scheme that determines the minimum, maximum and points of inflection in the cross-section function in the resolved-resonance region. Furthermore, POLIDENT processes all continuous-energy cross-section reactions that are identified in File 3 of the ENDF/B library and outputs all reactions in an ENDF/B TAB1 format that can be accessed by other AMPX modules.

12. KEY WORDS/DESCRIPTORS ***(List words or phrases that will assist researchers in locating the report.)***

continuous-energy cross sections, resonance parameters, resolved-resonance region, unresolved-resonance region, ENDF/B data, AMPX module

13. AVAILABILITY STATEMENT
unlimited

14. SECURITY CLASSIFICATION

(This Page) unclassified

(This Report) unclassified

15. NUMBER OF PAGES

16. PRICE