AMPX: A MODULAR SYSTEM FOR MULTIGROUP CROSS-SECTIOI GENERATION AND MANIPULATION

N. M. Greene, W. E. Ford, III, L. M. Petrie, B. R. Diggs, C. C. Webste
J. L. Lucius, J. E. White, R. Q. Wright, and R. M. Westfall
Computer Sciences Division
Union Carbide Corporation, Nuclear Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, U.S.A.

ABSTRACT

The AMPX system, developed at the Oak Ridge National Laboratory over the past seven years, is a collection of computer programs in a modular arrangement. Starting with ENDF-formatted nuclear data files, the system includes a full range of features needed to produce and use multigroup neutron, gamma-ray production, and gamma-ray interaction cross-section data. The balance between production and analysis is roughly even; thus, the system serves a wide variety of needs. The modularity is particularly attractive, since it allows the user to choose an arbitrary execution sequence from the approximately 40-50 modules available in the system. The modularity also allows selection from different treatments; e.g., the Nordheim method, a full blown integral transport calculation, the Bondarenko method, or other alternatives can be selected for resonance shielding.

INTRODUCTION

In early 1971, an effort was initiated to build a system of computer programs capable of producing multigroup cross sections. This work was funded by the Defense Nuclear Agency which had a particular interest in producing "coupled neutron-gamma cross sections" for use in weapons shielding and effects studies. The system was named AMPX which is an acronym for Automation of MUG[1], POPOP4[2] and XSDRN[3], three codes which were then in use at the Oak Ridge National Laboratory (ORNL) to generate multigroup gamma-ray interaction, gamma-ray production, and neutron cross sections, respectively.

The production of multigroup cross sections has always been (and still is) a particularly distasteful process involving the execution of many programs in transforming point data into multigroup form. To further complicate matters, this field is wrapped in its own mysterious jargon, e.g., ENDF/B, self-shielding, $S_{\alpha,\beta}$, P_ℓ matrices, transport corrections, etc. The selection of the most appropriate treatment from the vast collection of algorithms available for producing multigroup cross sections is a challenge even for the experienced analyst.

The standard procedures for making cross sections involve the execution of several codes in sequence. Many of these executions are large, long running computer jobs. Consider Figure 1 which aptly describes the 1971 ORNL procedure for producing coupled neutron-gamma cross sections.

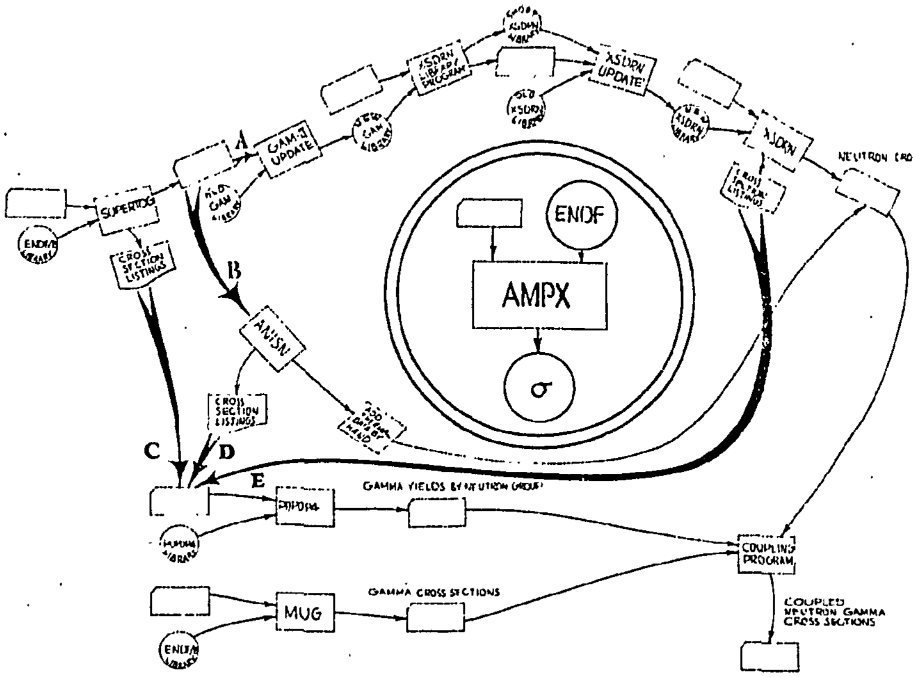

Figure 1.

The procedure was as follows:

Using ENDF/B data for neutrons, the SUPERTOG code was run to produce multigroup data in a reasonably fine group structure (approximately 100 groups). Depending on the option selected, SUPERTOG output was available in two formats: 1) a GAM-II[4] "update" stream and 2) ANISN[5] format. Option 1 was commonly used because it allowed for the eventual self-shielding of resonance nuclides. In cases involving non-resonance nuclides, the second option was used. For option 1, the GAM-II Update Code was run to produce a library in GAM-II format. After this run, a code which merged the GAM-II data with THERMOS[6] thermal data was executed. This produced a short XSDRN library which was coupled with existing neutron data through yet another updating run. At this point, an XSDRN run self-shielded resonance nuclides using the Nordheim Integral Treatment[7] and performed a discrete ordinates spectral calculation for collapsing to a few-group structure. Output was on cards for the ANISN code. For the ANISN option, one made an ANISN collapsing run analagous to the XSDRN run after first adding thermal values by hand into the ANISN cross sections.

Another part of this procedure required the execution of the POPOP4 program to calculate the coefficients used to determine gamma sources from neutron interactions. This code had its own "gamma yield" library and required multigroup cross sections for specific reactions from the neutron averaging runs. Thus, its start depended on all the aforementioned codes having been run.

- A completely independent path used the MUG code to generate gamma-ray interaction cross sections.

- After all other steps were completed, a "COUPLING CODE" was run to combine the cross sections for all three sequences into a coupled neutron-gamma set. At this point the user was able to perform the calculation he wanted to make in the first place, e.g., a Monte Carlo or an S_n calculation, etc.

This was a very time consuming procedure. Many of the codes required input from previous codes which had to be laboriously transferred from computer printout to a card input form. An error in some of the initial codes could invalidate almost all runs in the procedure, requiring practically all of the codes to be rerun. These schemes literally took weeks to produce a few sets of cross sections.

AMPX DEVELOPMENT

A more palatable way of attacking a problem involving the execution of multiple codes in a loosely defined sequence is to build a modular system which encompasses the complete collection of codes.

Modular System

A modular system is a collection of codes which can be run in either a preassigned and/or an arbitrary sequence and which communicate to each other through external interfaces. This external communication is the key to a true modular system and is accomplished through the use of tapes, drums, disks, etc. (Many large programs lay claim to the word "modular" in a description of the programming techniques employed, but a close scrutiny reveals them to be "FORTRAN programs" which communicate through COMMON's and argument lists, just like any other program!)

In a true modular system, the module many times is a program which has a clearly defined function, e.g., to calculate multigroup neutron cross sections or to calculate neutron fluxes. Since communication to other modules is through external interfaces, it is convenient to substitute modules in execution sequences, e.g., to substitute a diffusion theory code for a transport theory code.

The modular approach is fairly widely supported with notable examples being: the ARC[8] system at Argonne National Laboratory, the NOVA[9] system at Knolls Atomic Power Laboratory, the JOSHUA[10] system at Savannah River

Laboratory, and the CCCC[11] effort developed for use in several computer code development areas supported by the Department of Energy (DOE).

The means for accomplishing the linking together and communication between programs varies from a simple to a very complicated level: 1) the simplest approach provides no module for driving computer codes at all. Standard communication interfaces are rigidly defined, and a user still has to make independent submissions for each code he runs or he ties codes together using job control procedures. This type of system is represented by the CCCC modules. 2) The next level of sophistication is to define the rigid interfaces with a "driver" module which allows the user to run several modules back-to-back during a single execution. The AMPX system is characteristic of this type. A FORTRAN driver serves for the CDC versions of the system while a very flexible assembler language driver serves on IBM systems. 3) A variation of the second approach produces yet another increase in convenience. This is the case wherein the user supplies input for a complete module execution sequence to a control program which both creates all input files for the various modules and selects those modules in the proper order. In this scheme individual modules generally don't read card input streams, except for the control program. This type of system is characterized by the ARC, SCALE[12] (an extension of AMPX), and VENTURE[13] systems. (VENTURE is a major "module" in the CCCC array of codes.) 4) Another level is reached when the "driver" module dynamically decides which modules are to be run based on results obtained <u>during</u> an execution sequence. The JOSHUA and the NOVA systems use a central data base management system (DBMS) for communication links between modules.

AMPX Design Considerations

The major design features for AMPX are shown in Figure 2. Because of the obvious advantages for an area that requires a very loose and variable sequence of codes to be run, the modular system approach was chosen for AMPX. Many cross-section applications do not require both neutron <u>and</u> gamma-ray data. The modular approach is a convenient vehicle for supporting both areas (and others) in a manner which allows any improvement or modification to be universally available.

Basic Data Source

Evaluated Nuclear Data Files[14] (ENDF) are the standard means for distributing current basic neutron and gamma-ray cross sections in the United States. Basic processing modules in AMPX were to exclusively rely on ENDF-formatted libraries as their data source.

Variable Dimensioning

A technique widely used in reactor physics codes, e.g., ANISN and XSDRN, is to maximize the use of data array space by having all arrays a code uses stacked into one large container array. The size of this

ORNL-DWG 72-7329

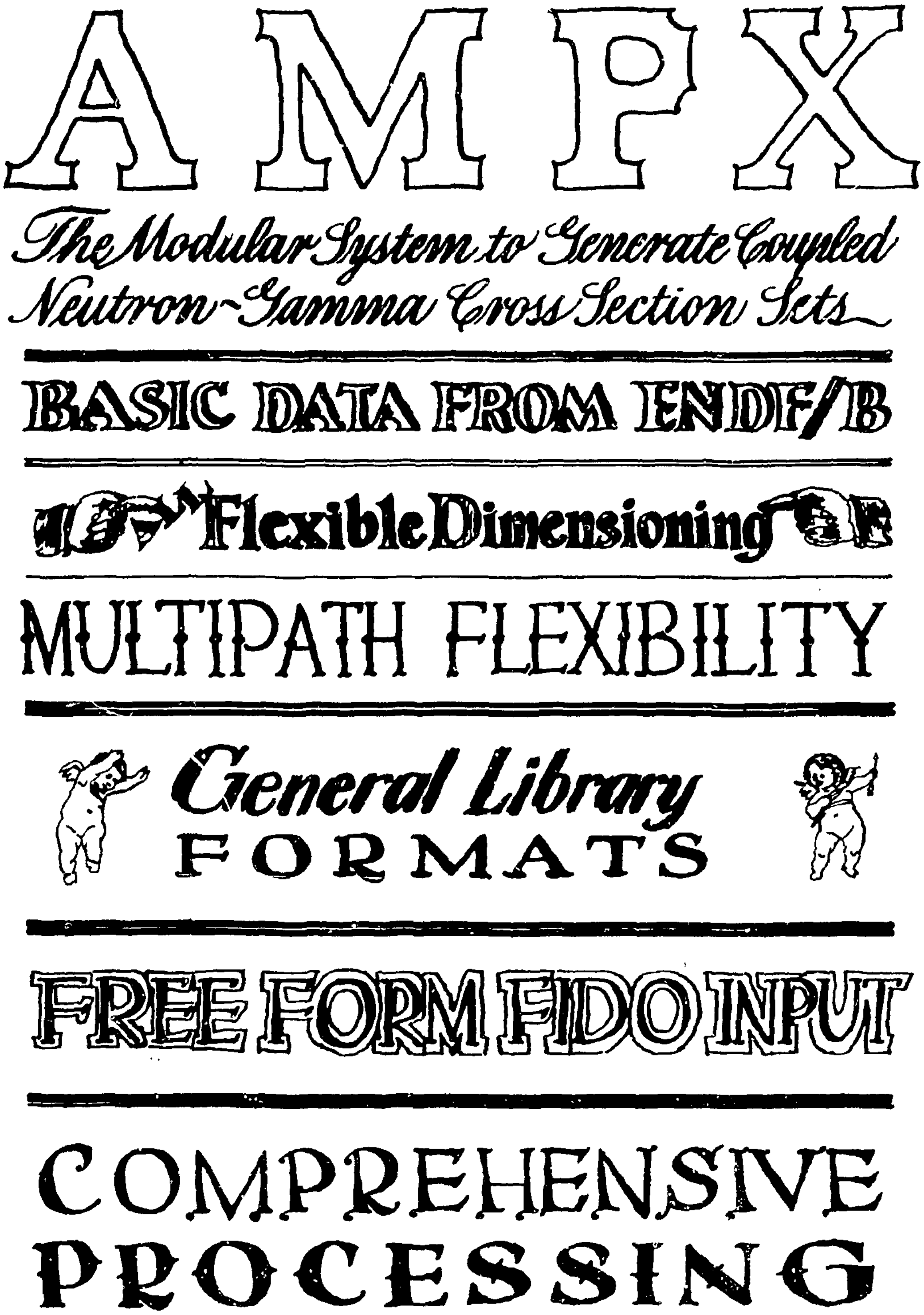

Figure 2

array is that amount left over from the core region assigned a program after one accounts for the program space (load module size) and possible buffer regions required for input/output. The program itself keeps an internal set of pointers which locate the various arrays in the large array. These pointers are set to the actual size needed by the problem being run. This kind of flexibility is highly desirable for a cross-section processing system as there are many arrays whose lengths vary significantly from case to case as a function of the number of neutron groups, number of gamma-ray groups, order of P_n, weighting spectrum, etc. Each increase in computing capacity (time-and space-wise) has always been matched by a corresponding increase in the degree of complexity which goes into a "typical job." The variable dimensioning approach minimizes and ideally eliminates the amount of reprogramming needed to accommodate these demands. In IBM versions of AMPX (and some CDC versions), special routines are available which will look at the core areas available and automatically assign the maximum available amount of data storage to the container array for a module. This is a particularly attractive feature as it allows varying the amount of data storage by merely varying the region of core assigned to a job without having to recompile a small control program.

Multipath Flexibility

The principal advantage of having a system where one can select an arbitrary set of modules and tie them together is that it reduces duplication. A module can be inserted at any point in an execution sequence that the module's function is required. A modular system permits simpler modules since modules have fewer options. It is a positive approach because the user never has to "turn off" options he doesn't want. He selects only those pieces needed. Examples of features which can easily serve at several points in a sequence are: conversion of cross-section library formats, data editing, data checking, data collapsing, etc.

General Library Formats and Communication Interfaces

In systems tailored for producing and using multigroup data, communication between codes is primarily made with cross-section data in multigroup or basic (point, resonance parameter, etc.) forms. An examination of the 1971 ORNL procedure shown in Figure 1 supports this. In each case it was a block of neutron data, gamma-ray interaction data, or gamma-ray production data which formed a communication link. A more careful examination reveals that the different blocks had several different formats (in fact, some of the codes served for little more than format converters).

Since formats are restrictive and the addition of new schemes might require modifications in many places in different ways, the use of different formats for different types of cross-section data is a very undesirable situation. Many of the cross-section blocks are not self-describing and require many descriptors to augment them (for example, ANISN blocks require a user to specify number of energy groups, total cross section location, etc.). To circumvent many of these difficulties, one of the first steps in AMPX development was to design a reasonably general and flexible format for passing multigroup data. It would have many features, including the following:

1. including the neutron and/or gamma-ray energy group structures in the library,
2. the ability to serve for neutron, gamma-ray interaction, gamma-ray production or coupled neutron-gamma data libraries where the different types of data are well identified,
3. a capability for accommodating any number of separate processes represented in angle to a completely arbitrary and variable order (elastic neutron data might be represented to order 8, while discrete inelastic levels were all given to order 3, other neutron processes to order 0, gamma-ray data to order 10, etc.),
4. the ability to carry all data needed in typical multigroup operations, including resonance parameters for self-shielding via the Nordheim treatment and/or Bondarenko[15] factor data for self-shielding,
5. options to specify a temperature dependence on cross sections,
6. provisions for truncating the "zeroes" in cross-section scattering matrices so as to both keep from wasting space and to allow codes which use the data to consider only the "real" data (for example, the calculation of an inscattering source term would only consider those transfers which are nonzero).

This general format, called an AMPX Master Interface, covers most of the interfacing requirements for AMPX. In retrospect, the generality in the format has never been regreted and the modifications considered desirable would be to expand it to allow more generality.

Another standard interface for AMPX is the Working Interface, which is a library with individual processes summed together into "total matrices" and which is ready for use in a nuclear calculation. This also implies that a resonance calculation has been made for resonance nuclides.

Free Form Input Data

In a system of several codes, many applications can require a considerable amount of card input data for the different modules. To reduce the tedium of preparing this input, AMPX modules use an input system called FIDO which has evolved over the years from input routines developed for the predecessors of ANISN. These routines presently allow a formidable array of convenience options, in addition to free form provisions, making the preparation of input less of a chore. As examples, provisions are allowed to repeat entries, to interpolate between entries, to initialize arrays, etc.

Comprehensive Processing

AMPX attempts to span the full range of cross-section processing needed at ORNL. This requires a large number of capabilities, many of which are listed in the next section of this paper.

AMPX CAPABILITIES

The capabilities of AMPX modules can be loosely categorized into the following areas:

1) Basic data processing
2) Resonance self shielding
3) Spectral collapsing
4) Format conversion
5) Service functions
6) Miscellaneous

Basic Data Processing

AMPX has several modules which start with ENDF/B data and use it to obtain multigroup values. XLACS is a module to process neutron data. It is unique among processors of this type in that it produces a set of cross sections containing both fast and epithermal data merged together with thermal data. The thermal data is based on $S_{\alpha,\beta}$ data on special ENDF/B thermal libraries. (This is somewhat contrary to the "modular" philosophy where such distinctly separate areas would be separated into different modules.) LAPHNGAS is a module which processes gamma-ray yield information, i.e., the data which allows one to determine the gamma sources caused from different neutron interactions. SMUG is analogous to XLACS except that it processes gamma-gamma interaction data. Scattering matrices from this code are based on the Klein-Nishina formula. JFC is a newer gamma-gamma processor which treats gamma-ray form factor data given in newer ENDF/B evaluations. This code is capable of generating proper data for photons at x-ray energies. Though not in the system, the MINX[17] code at ORNL produces an AMPX Master Interface allowing its output to be accessed by other AMPX modules.

Resonance Self-Shielding

The NITAWL module contains an upgraded version of the Nordheim Integral Treatment[7] which has been widely used at ORNL, especially in thermal reactor and in criticality safety calculational studies. Among improvements in the AMPX version are: an ability to self-shield elastic scattering, compliance with the Breit-Wigner equations expected by ENDF/B, the ability to treat "Multi-isotope" ENDF/B evaluations, and inclusion of a capability of self-shielding p-wave resonances. The BONAMI module provides for a Bonarenko iterative self-shielding treatment. Indicative of this module's generality is the ability to perform the BONARENKO calculation over energy ranges that vary nuclide by nuclide and process by process. The ROLAIDS module gives a one dimensional integral transport solution to a system with an arbitrary number of spatial zones, each of which contain an arbitrary mixture. The fluxes from this solution are used to produce self-shielded cross sections which include self-shielded elastic transfer matrices.

Spectral Collapsing

Modules that spectrally collapse use a calculated or a previously prepared multigroup spectrum to flux weight cross sections. This generaliy involves a reduction in the number of energy groups, though, in the case of cell weighting, this may not be the case. The simplest AMPX module of this type is COMAND, which will accept an ANISN library on cards or binary tape along with a spectrum and will collapse to another ANISN library on cards or on binary tape. (This module is sometimes used just for its ability to convert ANISN data on cards to a tape, or vice versa.) ARID is an obsolete (and retired) AMPX module that accomplished the same thing to a very specific ANISN library, e.g., one with 37 neutron and 21 gamma groups. The MALOCS module is analogous to COMAND, except that the input and output libraries are in AMPX Master Interface form. This module collapses cross sections, Bonarenko factors, etc., in a manner which retains the full capability for subsequent self-shielding calculations. In the case of both MALOCS and COMAND, all sets of data on the original library are treated without the user selecting specific data. A more prevalent kind of weighting, however, comes in the use of the XSDRNPM module. This module accepts an AMPX working library as input along with a description of a spectral calculation to be performed (one dimensional discrete ordinates, one dimensional diffusion theory, or infinite medium theories are available) and uses the results of the calculation to weight cross sections. A user has a variety of options available including zone and cell weighting. The output cross sections are written in AMPX Working Interface form, and can also be requested in ANISN form (on cards or tape) or in CCCC[11] ISOTXS form. XSDRNPM is also used many times just for its one dimensional calculation capability. A particularly attractive feature of the code is its built-in routines for calculating S_n quadrature sets, thereby relieving the user of having to maintain files of these data. Different quadratures are used for spherical, slab, and cylindrical geometries.

Format Conversion

Several major codes and code systems are used at ORNL and elsewhere for neutron and gamma-ray calculations. Unfortunately, these different systems generally require different and sometimes unique formats for cross-section libraries. AMPX provides several modules for converting to and from these formats. As previously mentioned, the XSDRNPM module can convert from AMPX Working Interface format to ANISN or CCCC[11] ISOTXS format. NITAWL can convert from AMPX Master or Working Interface formats to the ANISN format. The CONVERT module converts a library written for the stand-alone XSDRN[3] code to AMPX Master Interface format. REVERT accomplished the same mapping in the other direction. The OCTAGN module converts AMPX working formats to CITATION[16] libraries. LAVA is a module for converting ANISN libraries to AMPX working formats. The CONTAC module takes an AMPX working format and converts it to either a CCCC ISOTXS file or into an ANISN library.

Service Functions

There are a myraid of operations needed to maintain multigroup libraries. Historically these functions were provided in the module needing the operation (e.g., XLACS, NITAWL, XSDRNPM, have fairly complete data editing options); but, as AMPX has evolved, more and more of these functions are relegated to independent service modules. This is very desirable in that it makes modules more compact and easier to maintain. The CHOX and CHOXM modules serve to create an AMPX Master Interface containing coupled neutron-gamma data starting with data separated on a neutron only, a gamma-ray interaction only, and a gamma-ray production only library. CHOX uses all AMPX Master Interfaces as input, while CHOXM uses CCCC interfaces for the neutron only interface. UNITAB is a more general module of this type. It allows constructing an AMPX Master Interface where individual parts of the cross-section data are selected from any AMPX Master Interface. For example, one can create a set consisting of the Bonarenko data from one library, the averaged neutron data by group and process from another library, the elastic neutron matrices from one library, the (n,2n) matrices from another, etc. The final set can be a coupled library or a neutron only library, etc. UNITAB can split a coupled library back into its neutron, gamma-ray production components. DIAL is a module to edit Master or Working Interfaces. PAL will punch data from a Master or Working Interface. The ICE module accepts a Working Interface, creates mixtures and outputs these mixtures onto another Working Interface or onto ANISN libraries. For those embarrassing cases where one finds a library which is improperly normalized (e.g., the total cross sections are not the sums of all partial values, or the averaged inelastic value does not agree with the sum of all P_0 transfers, etc.), the COMET module is provided. This module accepts a Master Interface and can be made to force normalizations; it can modify resonance data; it can create new cross section vectors according to a user specified set of directions. For those cases where the error on a Master Interface is a bad number, the CLAROL module is provided. This module can replace group averaged values or can replace terms or add to transfer matrices on the library. (This module has a direct coupling to the ROLAIDS module wherein ROLAIDS does a self-shielding calculation and prepares an input stream for CLAROL which then modifies the library.) The module allows a user to easily introduce a change to many processes without having to explicitly make the changes. As an example, the user may change the (n,γ) cross section and have CLAROL automatically include the change in the capture, absorption, and total cross sections. AJAX is a module which allows a very general capability for merging sets of data from different libraries onto a single master library. Or it can be used to select a part of a large library, etc. A reasonably general capability for plotting cross section data is given by the VASELINE module. This code allows a user to plot group averaged data by process taken from a Master or Working Interface. It also provides for plotting point data from ENDF/B libraries or from AMPX Point Interfaces. Point-versus-multigroup plots can be made in a procedure which will allow an arbitrary number of curves to be put on a single graph.

Miscellaneous

This section covers those modules which don't easily fit under one of the headings listed above. Many applications require just the point data for selected processes. The NPTXS module accesses an ENDF/B library and calculates point values of the elastic scattering, fission, capture, and total cross sections for resonance nuclides. These are written on an AMPX Point Interface. Provisions are given for (ψ,χ) Doppler broadening or for a numerical treatment taken from the MINX code. The JERGENS code can take the point strings from NPTXS or elsewhere and perform mathematical operations based on the input strings to form other point strings. The mathematical operations are all performed to a user specified accuracy. For example, many operations could use the product of cross section times flux in a single array, or other applications need $1/E\Sigma_t$ values where Σ_t is a macroscopic point function consisting of a combination of many point functions. JERGENS also allows for the generation of several commonly needed functions on a point mesh, e.g., a Maxwellian or a fission spectrum. AIM is a module which converts an AMPX Master Interface (in binary form) into an equivalent BCD form. It also serves to convert the BCD form back to binary form and, in this roll, is used to pass data between different types of computers. An added advantage of the code is that the BCD form was designed as a normal FIDO card input stream, so that the user can create a Master Interface starting with cards. For an automatic checking of any AMPX Master or Working Interface, RADE is available. This module makes many simple checks, including summing group averaged values to check against "total" averaged values, summing transfer matrices to compare with averaged values for the process, checking higher order Legendre coefficients for reasonableness, looking for suspicious negative numbers, checking record counters in directories, etc. (In retrospect, this is one of the more valuable modules in AMPX in that it has many times caught improperly processed results before they were included in any calculational study.) In addition to the AMPX Master or Working Interface checking, this module has options to check ANISN libraries.

FUTURE DEVELOPMENT

A successful modular system is probably never through its development stage. The features of an existing modular system can be used as a starting point for code development in new areas. The modularity makes it easy to substitute or upgrade specific operations without affecting the total system. The modular system makes it as easy (and efficient time and space-wise) to execute a code as it is to execute the code in a stand-alone mode. Except for maintenance, new development will only involve the new areas. With a little instruction, non-AMPX personnel write modules which can efficiently interact with other modules in the system.

Trends

Over the course of the development of AMPX, modules have tended to become more compact and specific in their functions. This trend has minimized effort that was being expended doing redundant programming. Another trend is the move toward a variation on AMPX wherein the control programs create input streams for other modules and select module execution sequences. Since the user only makes input to the control program, this eliminates redundant input. The need for this type of operation has intensified as the modularity of the system has increased and the typical operation involves more and smaller modules.

Present and Future Modules Under Development

At least 10—15 new modules are in varying stages of development. These include four small modules which perform service function on AMPX point interfaces. Another module can recover data from a Master Interface when a portion of tape becomes unreadable. Another module processes ENDF/B unresolved data and creates point files which are fed to another module which produces Bondarenko factors from these data. A new thermal processor has been written which converts $S_{\alpha,\beta}$ data to ENDF/B File 6 form and processes it. A code has been written to produce "sensitivity" Working Interfaces from Master Interfaces. A new generalized cross section weighting module has been planned, etc. These new modules will be added to the RSIC AMPX collection as they become fully operational.

REFERENCES

1. J. R. Knight and F. R. Mynatt, MUG: A Program for Generating Multigroup Photon Cross Sections, CTC-17 (January 1970).

2. W. E. Ford, III, and D. H. Wallace, POPOP4 - A Code for Converting Gamma-Ray Spectra to Secondary Gamma-Ray Production Cross Sections, CTC-12, (May 1969).

3. N. M. Greene and C. W. Craven, Jr., XSDRN; A Discrete Ordinates Spectral Averaging Code, ORNL-TM-2500 (July 1969).

4. G. D. Joanou and J. S. Dudek, GAM-II A B_3 Code for the Calculation of Fast-Neutron Spectra and Associated Multigroup Constants, GA-4265 (September 1963).

5. W. W. Engle, Jr., A Users Manual for ANISN, K-1693, (March 1967).

6. H. C. Honeck, THERMOS: A Thermalization Transport Theory Code for Reactor Lattice Calculations, BNL-5826 (September 1961).

7. L. W. Nordheim, "The Theory of Resonance Absorption," Proceedings of Symposia in Applied Mathematics, American Mathematical Soc. Vol. XI, p. 58, Garrett Birkhoff and Eugene P. Wigner, Eds., (1961).

8. C. N. Kelber, G. Jensen, L. Just, and B. J. Toppel, The Argonne Reactor Computation System, ARC, Proceedings of the International Conference on the Utilization of Research Reactors and Reactor Mathematics and Computation, Mexico City, pp. 1428–1445. (1967). See also:

   B. J. Toppel, The Argonne Reactor Computation System, ARC, ANL-7332, (1968). See also:

   B. J. Toppel, The Argonne Reactor Computation (ARC) System, Reactor Physics Division Annual Report, July 1, 1966 to June 30, 1967, ANL-7310, pp. 433–436. See also:

   L. Just, and S. G. Sparck, The ARC System, Applied Mathematics Division Internal Technical Memorandum No. 157 (1968) (unpublished).

9. E. D. Reilly and W. H. Turner, "The Automation of Reactor Design Calculations at the Knolls Atomic Power Laboratory." Proc. Conf. Application of Computing Methods to Reactor Problems, Argonne, Ill., May 27–29, 1965. USAEC Report ANL-7050, Argonne National Laboratory, pp. 251–63 (1965).

10. H. C. Honeck, The JOSHUA System, DP-1380, (April 1975).

11. B. M. Carmichael, Standard Interface Files and Procedures for Reactor Physics Codes, Version III, LA-5486-MS (February 1974).

12. R. M. Westfall, Private Communication.

13. D. R. Vondy, T. B. Fowler, and G. W. Cunningham, VENTURE: A Code Block for Solving Multigroup Neutronics Problems Applying the Finite-Difference Diffusion-Theory Approximation to Neutron Transport, Version II, ORNL-5062/R1, (November 1977).

14. ENDF-102-Data Formats and Procedures for the Evaluate Nuclear Data File, ENDF, Edited by D. Garber, C. Dunford and S. Pearlstein, BNL-NCS-50496 (END 102), (October 1975).

15. I. I. Bondarenko (Ed.) Group Constants for Nuclear Reactor Calculations, (Consultants Bureau, New York 1964).

16. T. B. Fowler, and D. R. Vondy, Nuclear Reactor Depletion and Dynamics Code: CITATION, ORNL-TM-2496, (July 1969).

17. C. R. Weisbin et.al, MINX: A Multigroup Interpretation of Nuclear X-Sections from ENDF/B, LA-6486-MS, (September 1976).

18. N. M. Greene and F. R. Mynatt, "The AMPX Modular Code System for Generating Coupled Neutron-Gamma Multigroup Cross-Section Sets," Trans. Am. Nucl. Soc.. 15, 568 (1972); see also N. M. Greene et al., AMPX: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Libraries from ENDF/B, ORNL-TM-3706 (March 1976).