DP-1380

660331

The JOSHUA System

H. C. HONECK

DUPONT
REG. U.S. PAT. OFF.

Savannah River Laboratory

Aiken, South Carolina

NOTICE

Printed in the United States of America

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

Price: Printed Copy $4.00; Microfiche $2.25

663051

DP-1380

Distribution Category UC-32

The JOSHUA System

H. C. HONECK

Approved by

J. W. Wade, Director
Computer Sciences Section

Publication Date: April 1975

E. I. du Pont de Nemours & Co.
Savannah River Laboratory
Aiken, S. C. 29801

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION UNDER CONTRACT AT(07-2)-1

ABSTRACT

A major computational system called JOSHUA has been under development at the Savannah River Laboratory since 1968. The JOSHUA System has two major parts: the Operating System and the Application System. The Operating System has been in production use since 1970 and provides data management, terminal, and job execution facilities. The Application System uses these facilities in solving problems in reactor physics and engineering. Features of the Application System are the two-dimensional lattice physics and three-dimensional transient reactor physics capabilities, which have been in use since 1971 and 1974, respectively. This report summarizes the capabilities of the JOSHUA System along with statistics on size, use, and development effort.

ACKNOWLEDGMENTS

The development of the JOSHUA System involved many people in the Savannah River Laboratory. Their names are listed below, and their contributions to both the development of the system and the writing of this report are gratefully acknowledged.

Dr. John E. Suich, former Director of Computer Sciences Section at Savannah River Laboratory, deserves special recognition for his contributions to the concepts of the system and his development of the first versions of the data manager and terminal display facilities.

P. L. Ames
M. M. Anderson
C. E. Bailey
W. V. Baxter
R. L. Boyce
W. H. Brotherton, Jr.
M. R. Buckner
J. H. Carswell
J. P. Church
B. Crain
E. A. Crider
B. P. Compton
A. H. Dobyns
H. L. Dodds
D. R. Finch
F. C. Fortune
W. E. Graves
M. V. Gregory
T. C. Gorrell
S. D. Harris
W. R. Hartshorn
R. C. Haywood
C. E. Henderson
J. H. Hightower
W. H. Hightower
J. R. Hilley
H. E. Hootman
D. C. Irving
J. C. Jensen
J. L. Kilpatrick
F. D. Knight
D. H. Knoebel
J. H. Maddox
F. J. McCrosson
J. P. Morin
D. R. Muhlbaier
C. E. Parker
R. A. Priest
R. J. Pryor
H. R. Reeve
P. L. Roggenkamp
K. R. Routt
J. M. Sandler
D. A. Sharp
D. G. Shelton
J. M. Sicilian
E. R. Siegmann
J. A. Smith
J. W. Stewart
M. B. Stroud
J. R. Taylor
V. D. Vandervelde
R. G. Willard
R. S. Wood
L. F. Zimmerman

CONTENTS

Introduction to the JOSHUA System

DESCRIPTION OF THE REPORT

Purpose

The purpose of this report is to:

- Describe the major features of the JOSHUA System
- Document the cost of its development
- Report the benefits
- Introduce the capabilities to other laboratories

The JOSHUA System has been extensively documented in the publications listed in Appendix A and in the series of manuals listed in Appendix B. This report provides a more concise description.

Audience

This report is aimed at a wide audience of both technical and administrative people. However, it is difficult to describe an extensive computational system and make the report informative to the nonspecialist and, at the same time, provide enough information for the specialist to evaluate the system. Three techniques have been employed in the report to make it readable without sacrificing too much detail:

- Module names and acronyms have been omitted except for the names of major components.
- Many details have been removed from the text and placed in *Fact Sheets.* These *Fact Sheets* are placed in the report near the relevant text but are not referenced.
- Drawings and pictures are used extensively.

Organization

The report is organized in the following manner. The remainder of this section provides some background material on scientific computing at Savannah River and the concept of a modular data-based system. The following sections discuss the major components of the JOSHUA System:

- JOSHUA Operating System
- Generalized Application System
- Specialized Application System

The final sections discuss:

- Development effort
- Size and use statistics
- Future development
- Availability

Scope

The JOSHUA System consists of both an:

- Operating System featuring powerful data management, job execution, and terminal facilities
- Application Systems developed to use these facilities in solving problems at Savannah River

The JOSHUA System was developed to perform the extensive reactor physics and engineering calculations required to design charges for the Savannah River reactors. Most of the development effort has been for this purpose, and this report will describe only those applications. The facilities provided by the JOSHUA Operating System have also been used for several other applications:

- Plant simulation and forecasting
- Financial information
- Environmental transport

Role of Computing

The role of computing in reactor design and operation at Savannah River was described in some detail by J. E. Suich in a 1970 memorandum. That memorandum begins with the paragraph:

"Computing has become an integral part of Savannah River's reactor technology. This has happened because program objectives at Savannah River have consistently posed increasingly complex reactor problems; and, at the same time, computing techniques and equipment have developed rapidly. Fortunately, solutions to Savannah River problems by computing have proved feasible and economical; alternative solutions through expansion of experimental facilities and staff would have been more expensive and of doubtful feasibility. Since startup, computing power in use at Savannah River has grown exponentially, with a doubling time of about one year, so that present facilities have a power almost a millionfold greater than the rudimentary equipment initially installed."

This growth is illustrated in Figure 1. The role of computing and the needs of the users are still much the same at present as they were in 1970 despite the normal programmatic changes at Savannah River.

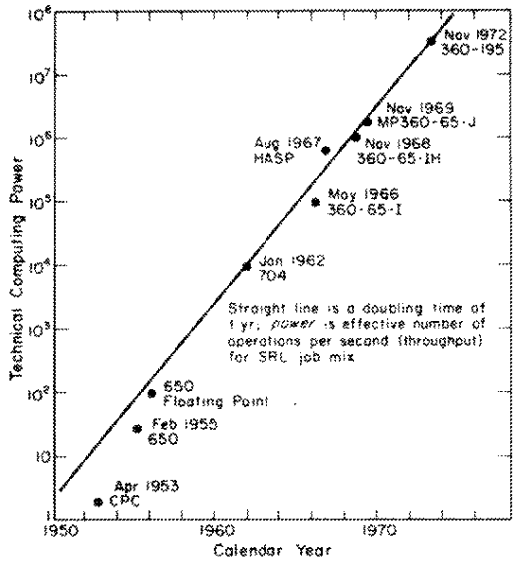


FIG. 1 Technical Computing Power

Charge Analysis

Most of the computing at Savannah River is directly related to the design and operation of reactor charges. The analysis of these charges requires a wide variety of computational procedures to predict the neutronic and engineering behavior of the reactors during both normal and abnormal operating conditions. These computational procedures were implemented in numerous computer codes, which were used on a production basis once they had been developed and tested.

Computer Configuration (December 1974)

Central Processor:	*IBM 360/195 (2KK Bytes)* *IBM 360/30 (32K Bytes)*
Disk Storage:	*IBM 3330 (16 Packs)* *IBM 2314 (16 Packs)* *IBM 2305-1 (1)* *IBM 2305-2 (2)* *IBM 2311 (2)*
Tapes:	*IBM 2402 (15)*
Unit Record:	*IBM 2540 (2)* *IBM 1403 (3)*
Terminals:	*IBM 2260 (28)* *VG 11 (2), PDP 11/45*
Operating System:	*OS/360* *HASP*
Microfilm:	*FR-80*

Methods Development

One aspect of scientific computing at Savannah River is the development of the physics, engineering, numerical, and computational methods for these codes. A strong methods development program has been in progress for many years and has led to a mature technology for reactor design. Improved understanding of the reactors coupled with a demand for more detailed safety analyses has led to a highly complex technology requiring a massive system of strongly coupled computational procedures. It became evident in the late 1960s that calculations of this type required a new computing environment, that of the modular data-based system.

Production Computing

Reactor physics and engineering codes were run on a production basis both to design new charges and to follow the operation of a charge. A reactor design project requires thousands of individual computer runs using a wide variety of codes. Some statistics gathered in 1968 for a typical design project illustrate the magnitude of the problem. The number of computer runs submitted per month for the design project is shown in Figure 2. An average of 153 runs were submitted per month, or 7 runs per day. Considering that there were several concurrent design projects, and that relatively few people run these codes, there was a clear need to simplify the routine use of these codes by organizing them into a modular data-based system.

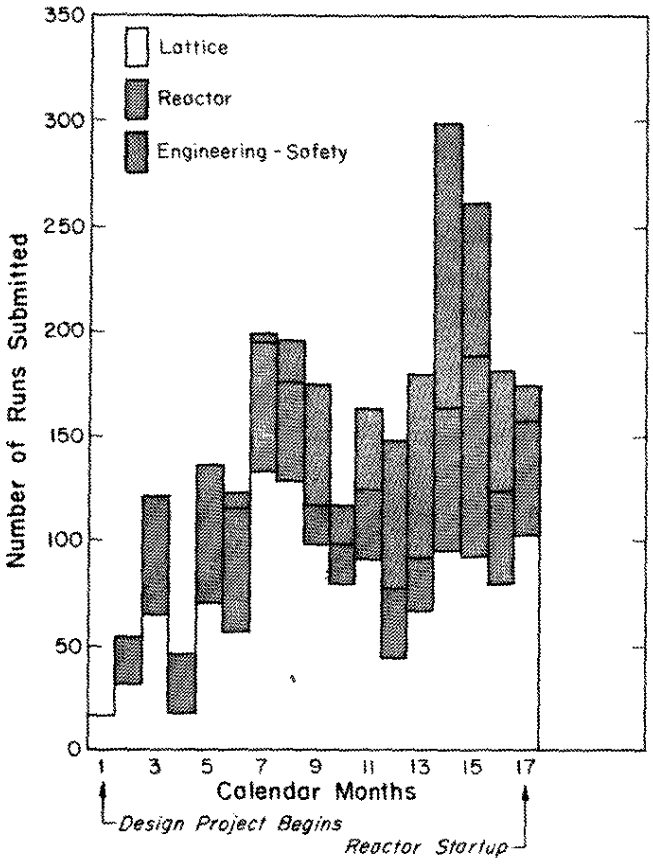


FIG. 2 Computer Runs Submitted by Problem Type

MODULAR DATA-BASED SYSTEM CONCEPT

JOSHUA is a modular data-based computational system for the multistep iterative design and analysis of nuclear reactors. What do these terms mean?

Multistep Iterative Computation

The design of a nuclear reactor requires many computational steps (Figure 3). A step might be the neutronic calculation of power in a fuel tube, or the engineering calculation of temperatures in a fuel tube and coolant. Steps are often repeated using different input assumptions. Groups of steps are repeated until a design criterion is achieved, such as varying fuel element size and content to simultaneously satisfy nuclear criticality, heat transfer, temperature, and power criteria. This design process is characterized by the terms "multistep" and "iterative."

FIG. 3 Multistep Iterative Calculation

Computer Codes

In the early 1960s when a step was implemented as a computer code, the input data were punched on input cards, and the computed results were printed on an output report (Figure 4).

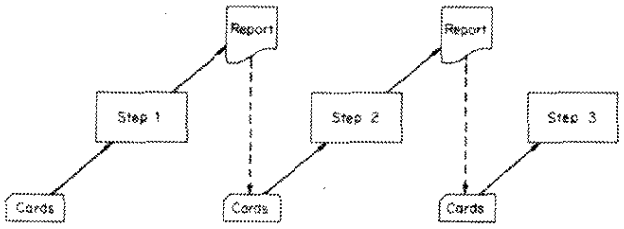


FIG. 4 Manual Transfer of Data from Reports to Cards

If some output results from one step were needed as input to a later step, they were transcribed by hand from the output report to input keypunch forms. Clearly this was a tedious and wasteful effort for reactor designers.

Disk Storage Devices

In the middle 1960s, magnetic disk storage devices became widely available. These devices made it practical to save results from one step and make them available to later steps (Figure 5).

If there is only one later step using these output data, then the selection of the output data and its organization on disk storage are dictated only by the needs of the one later step. In most cases, however, the results of one step are needed by several later steps, each of which has its own requirements for the data and its organization.

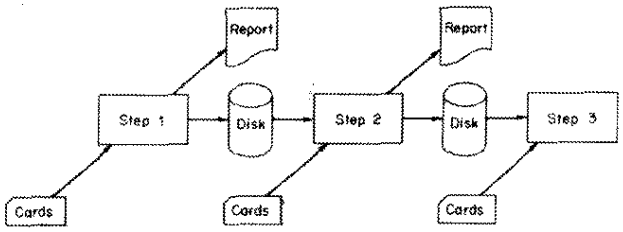


FIG. 5 Transfer of Data Using Disks

Modular Data-Based System

When selected output data from a computational step are placed in a "pool" of data residing on disk storage and made available to all other computational steps, the resulting system is a "data-based" system, and the computational steps are called "modules." Modules can execute other modules to allow complex computational procedures to be developed.

Two other systems of this kind had been developed before JOSHUA. The ARC System[1] developed at Argonne National Laboratory is intended primarily for fast reactor research. The NOVA System[2] developed at Knolls Atomic Power Laboratory is used for the design of naval reactors. Discussions with personnel at these two laboratories contributed greatly to the conceptual design of JOSHUA.

1. C. N. Kelber, G. Jensen, L. Just, and B. J. Toppel. "The Argonne Reactor Computation System, ARC." *Proc. Intern. Conf. Utilization of Research Reactors and Reactor Mathematics and Computation, Mexico City, May 2-4, 1967.* Report CNM-R-2 (Vol. III), pp 1428-45 (1967).
2. 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).

Terminals

A modular data-based system provides the mechanism for building both massive safety analysis systems and automated design systems. The use of computer terminals (Figure 6) provides several additional capabilities:

- Entry of data directly into the data base
- Inspection of results in the data base
- Modification of data in the data base
- Execution of modules

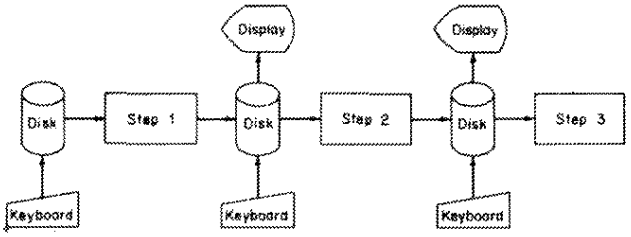


FIG. 6 Terminal Access to the Disks

These capabilities are important in the design process. They allow the reactor designer to dynamically control the data base and computational sequence without using input cards and output reports (although these facilities are still available if needed). The reactor designer can then perform his multistep iterative task using minimum time and effort.

Two types of terminals are used in the JOSHUA System. Most widely used is the IBM 2260 terminal (Figure 7), which has a keyboard and TV-like display for alphanumeric data. The Vectographics 11 terminals (Figure 8) are full interactive graphic terminals used to display graphical data.

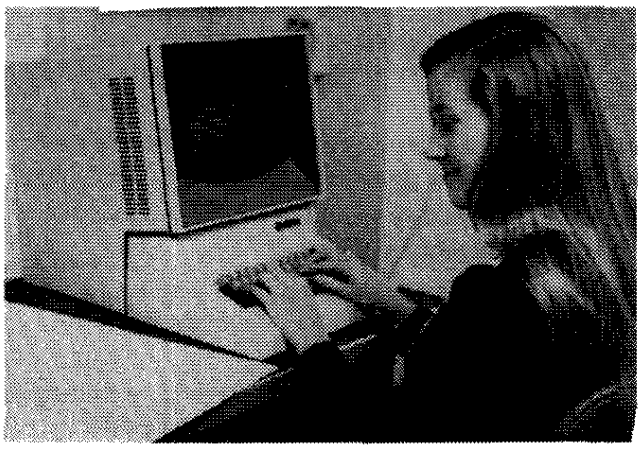


FIG. 7 IBM 2260 Terminal

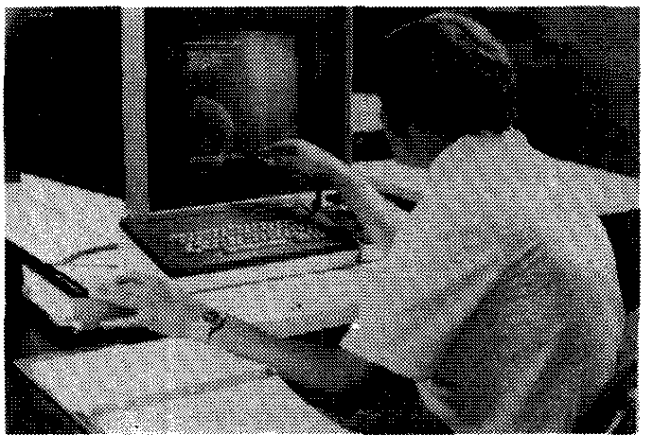


FIG. 8 Vectographics 11 Terminal

JOSHUA Operating System

FACILITIES REQUIRED

Three major facilities are required to support the concepts described in the previous section:

- A random access data management system usable from both modules and terminals
- A terminal monitor system to provide terminal access to the data base and to initiate the execution of modules
- A flexible dynamic linking facility allowing one module to call one or more other modules into execution

These facilities are normally part of the operating system supplied by the computer manufacturer. The facilities supplied in 1968 by IBM OS/360 were not adequate for this purpose; therefore, it was necessary to develop them at Savannah River. The collection of routines developed to provide the three facilities listed above is called the JOSHUA Operating System. It does not replace IBM OS/360, but extends the OS/360 capabilities and makes them available to both the reactor and module designer.

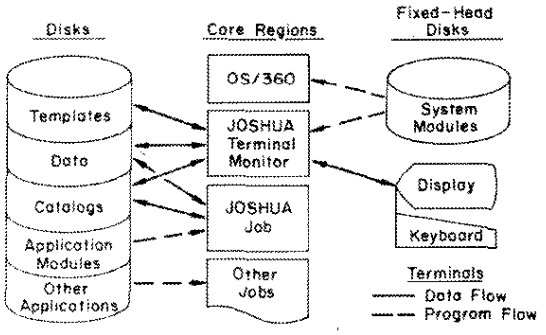


FIG. 9 Relationship of JOSHUA to the Computing System

The JOSHUA Operating System has been in production use since 1970.

Figure 9 illustrates the relationship of some of these routines to OS/360, the data base, the terminals, the application modules, and other non-JOSHUA applications.

DATA MANAGER

The Data Manager is the set of routines that controls the storage and retrieval of data on the disk storage devices.

Named Data Records

The individual collections of data stored on the disks are called records. These records must be accessed (read or written) randomly by both modules and terminal users. To accomplish this each record must have a unique name. While modules might be programmed to remember short record names or numbers, the user must "talk" to the System in terms familiar to him. Fortunately, most scientific data can be described by a hierarchy of qualification. For example, the broad category of multigroup cross section data is subdivided into isotopes, and isotopes into reaction types. The record containing fission data for ^{235}U might then be assigned the name

MULTIGRP.U235.FISSION

This is called a qualified name string. Each qualifier (e.g., U235) may be 1-8 characters in length. Up to 16 qualifiers are allowed, but no more than 10 have been used.

The record names and record data structures are arbitrary and are specified by the designer of an application subsystem. While a good choice of record names by the subsystem designer will help the user remember the names, terminal facilities are available to refresh the user's memory.

Module Data Access Statements

The FORTRAN Language does not provide for the reading and writing of named data records. The IBM FORTRAN

Language does provide for direct-access statements of the type:

READ (record location) Input/Output List

The JOSHUA System extends the FORTRAN Language to include statements of the type:

READ (record name) Input/Output List

A Precompiler is used to convert these new statements to IBM FORTRAN compatible statements. In effect, the Precompiler inserts statements to call the Data Manager, which takes the record name, looks up the record location in catalogs, and supplies the record location to the IBM FORTRAN direct-access statement.

Data Manager Operation

The major functions of the Data Manager are to relate data record names with their physical location on the disks, and dynamically manage the disk space. To accomplish these functions, the Data Manager must maintain catalogs that are also stored on the disk. This is illustrated in Figure 9.

Disk space is allocated by OS/360 to the JOSHUA System. This disk space is dynamically subdivided into logical Data Sets of an arbitrary size allocated according to the needs of the System and its users. Catalogs are maintained to reflect the current status of this disk space. These catalogs can be displayed at a terminal.

The catalogs that relate record name and location are hierarchal or "tree-structured" catalogs, each qualifier in the record name corresponding to one level of the tree. Since the catalogs can be very large, they are subdivided into pages stored as records on the disk. A search through the catalogs to find a record location could require many disk accesses. Several powerful algorithms have been developed and implemented to minimize the number of catalog disk accesses.

Once the location of a record has been determined by the Data Manager, the physical input or output of the record is accomplished using the FORTRAN I/O routines. Some of these routines have been modified to use the chained scheduling and buffer management needed for efficient transfer of large records.

Data Protection

It is essential to protect a user's data both from other users and from simple mistakes he might make himself. The Data Manager provides a very powerful protection method using a three-level hierarchy of Standard, User, and Job Data Sets. The operation of this protection method is illustrated in Figure 10.

The Standard Data Set is shared by all authorized users on a read-only basis. These data can be written or updated only by authorized personnel designated as custodians of the Standard Data.

Upon request, a User Data Set is permanently allocated to each user (or group of users). This Data Set is managed by the user with utilities provided so that he can back up and clean up his Data Set. The user has complete control (read, write, modify, delete) of his Data Set from a terminal. Terminal-created data are usually stored in the User Data Set.

A Job Data Set is dynamically allocated when a job is executed. Modules executed during this job write only to this Job Data Set (and hence cannot alter either the Standard or User Data Sets). When a module reads data, the Data Manager first looks for the data in the Job Data Set. If found, these data are supplied to the module. If the data are not found in the Job Data Set, the Data Manager next looks for the data in the User Data Set, and then in the Standard Data Set. This hierarchal search is automatically done by the Data Manager and is transparent to the module. The method is implemented by adding a high order qualifier to each record name. This qualifier is either the word STD (for Standard), the User Data Set Name, or the Job Name. The module may, however, define its own search hierarchy which may include other Job and User Data Sets.

A Job Data Set is automatically deleted after some period of time (currently 5 days). It is up to the user to inspect the output from a Job execution and copy those results that he wishes to save into his User Data Set. The assumption is that most results are of temporary interest and thus not worth saving so that no action is required to delete results, but a positive (though simple) action is required to save results.

Each user has a Badge Number which identifies him to the System. Unauthorized use of the System from a terminal is prevented by supplying each user with a Password unknown to other users. A terminal user is recognized by the System when he supplies both his Badge Number and Password.

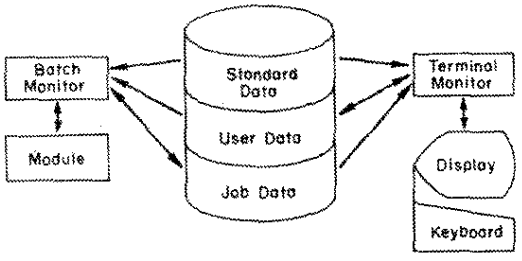


FIG. 10 Standard, User, and Job Data

The Terminal Monitor is a program that interprets and processes commands from the alphanumeric terminals. The Terminal Monitor is a high-priority, nonterminating batch job which is loaded into the computer at the beginning of the day shift. It is usually removed at the end of the day shift to release main storage for other batch jobs. As part of this initial loading, the Data Manager is made resident. The Terminal Monitor is written as a resident control program which, on demand, calls in function programs from drum (fixed-head disk) storage. Figure 11 illustrates the Terminal Monitor with two function programs attached: the Display Program and the Editor.

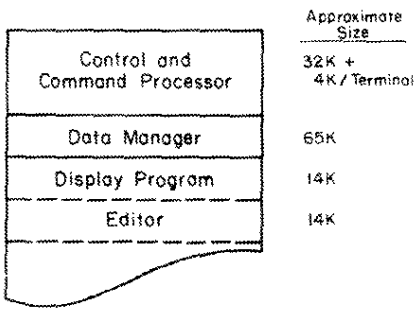


FIG. 11 Terminal Monitor Components

Display Program

Data records in binary form can be retrieved from disk storage using the Data Manager. They must be converted to alphanumeric form before they can be displayed on the screen of the terminal. The Display Program performs this function with the aid of Templates stored like data records on disk storage. The Template contains information about:

- Structure of the record
- Conversion from binary to alphanumeric
- Placement of data on the screen
- Explanatory text, headings, etc.

This is the same type of information usually contained in the FORTRAN I/O List and FORMAT statement used for printing data. Unlike this FORTRAN FORMAT capability, the Templates are created dynamically at a terminal by creating on the terminal screen the display exactly as it is to appear when data are to be displayed. A Template is created for each type (same name and list structure) of data record to be displayed. When a specific record is to be displayed, the Data Manager obtains the data record and the Template for that data record type, and the Display Program creates the display on the screen of the terminal.

Data Manipulation Facilities

The Terminal Monitor provides the user with a complete set of facilities to maintain his data. These include the facilities to create, modify, delete, copy, and rename data records. These facilities are invoked using a simple command language and are performed using the Display Program and Data Manager. Data Set maintenance facilities allow the user to clean up and reorder his Data Set.

Free-Form Data

The formatted display facilities provided by the Display Program and Templates require that the content and structure of the data record be predefined to the System. A free-form input capability is also provided which is used to enter and store unformatted, keyword driven data. The input can be interactively checked for syntax, keyword, and data mode errors if the checking data have been stored on the data base. The free-form input is extremely useful for supplying application modules with option and data modification information.

Editor

The Editor allows users to create, modify, and delete information in the form of card images stored on the disk. In the JOSHUA System, this capability is used primarily to maintain FORTRAN source decks and Job Control Language decks. The Editor is also used to create and modify input data decks for programs not run under the JOSHUA System. The maintenance of card images using the terminals has proved to be so convenient that the resulting heavy load has severely strained the Terminal Monitor. A continuing program of Terminal Monitor improvement has been necessary to retain a fast terminal response.

Job Entry

Users may submit jobs for execution using a terminal. Approximately 97% of the JOSHUA workload is submitted in this manner. The job is automatically supplied with predefined Job Control Language, and the user need

fill out only a simple one-page form displayed on the terminal screen. The Terminal Monitor causes the Job to be placed in the batch queue where it will await execution along with jobs submitted through the card reader. The user receives a Job Number and can use this number to inquire about the status of his job and of the jobs ahead of him in the queue.

Interactive Graphics

Interactive graphics terminals were obtained in late 1973. Programs have been developed to use these terminals for a few applications. Software for the full integration of these terminals with the Terminal Monitor (or a separate Graphics Monitor) is being developed, but will not be completed until late 1975.

JOB EXECUTION

JOSHUA jobs may be submitted using either the card reader or a terminal. The components of a JOSHUA job are illustrated in Figure 12.

A nucleus containing the Batch Monitor, the Data Manager, and the FORTRAN I/O routines is attached to each job (several jobs may be in execution at the same time). These resources are shared by all modules that may be executed during the job.

A control module (Module A in Figure 12) is specified by the user and is loaded and executed by the Batch Monitor. Standard OS/360 facilities are used to accomplish this. If the control module should call another module (for example, Module B in Figure 12), the Batch Monitor loads and executes Module B, and so forth. Each module may also be segmented into overlays. When a module execution has been completed, the module is deleted from main storage and the space recovered.

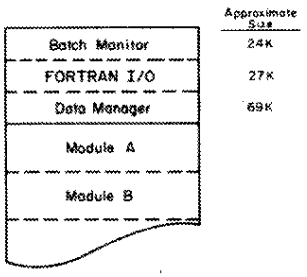


FIG. 12 Batch Job Components

PRECOMPILER

The extended data management and module execution facilities provided by the JOSHUA Operating System could be used by modules without extending the IBM FORTRAN IV Language. However, the required coding procedures would have been awkward to use and difficult to teach to casual programmers. New statements were therefore added as extensions to the FORTRAN IV Language, and a Precompiler was written to convert the new statements to standard FORTRAN IV statements. These new statements are used for:

- Data management
- Module execution
- Module and subroutine timing
- Error messages
- Including cataloged program segments

The Precompiler is executed prior to the FORTRAN compilation as part of a JOSHUA cataloged procedure. Figure 13 illustrates the Precompiler and the Editor used to maintain source decks.

The Precompiler has repaid its development costs many times over. During the development of the Data Manager, at least two major revisions were made to the detailed calling sequence. Without the Precompiler, all application source decks would have required changes. With the Precompiler, the changes were implemented in the Precompiler, and it was only necessary to recompile all source decks.

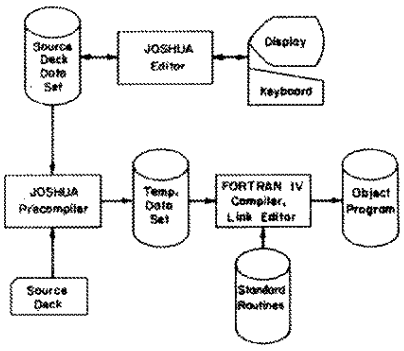


FIG. 13 Module Source Deck Processing

Generalized Application Subsystems

APPLICATION SCOPE

The purpose of the JOSHUA Application System is to provide the computational support for the design and operation of the Savannah River reactors. The largest and most important part of the required computational support is in the areas of reactor physics and engineering, and the development effort has been concentrated in these two areas.

Generalized Versus Specialized Applications

The JOSHUA Application System can best be described in two parts: one dealing with Generalized Application Subsystems, and the other with Specialized Application Subsystems.

When the JOSHUA Operating System became available in 1970, several groups of existing codes were modified to run under this System. The required modifications could be done quickly to provide a special purpose capability at a modest cost. The resulting systems are called Specialized Application Subsystems and are discussed in later sections of this report.

The major effort was the development of new computational capabilities called the Generalized Application Subsystems. The features desired in these Subsystems were:

- Comprehensive Data Base design for all reactor physics and engineering applications
- State-of-the-art reactor physics and engineering models
- State-of-the-art numerical and computing methods
- Efficient use of the available computing resources
- Oriented toward production use but capable of supporting research in methods development
- Expandable framework for future research and development
- Useful at other laboratories

Major Subsystems and Data Sets

The four major Generalized Application Subsystems illustrated in Figure 14 are:

- Basic Data Analysis Subsystem
- General Lattice Analysis Subsystem
- Correlation Reduction Analysis Subsystem
- General Reactor Analysis Subsystem

The Data Base is logically subdivided into Data Sets. Two classes of Data Sets are illustrated in Figure 14. The Data Sets on the left are created by the users of the System and are called the Input Data Sets. The Data Sets on the right are created by a Subsystem and are called the Results Data Sets. Both classes of Data Sets are kept on-line and have lifetimes of weeks to years.

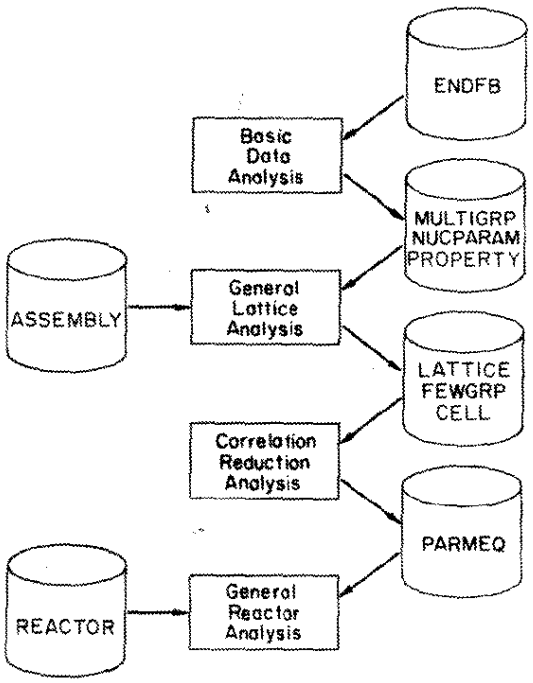


FIG. 14 Generalized Application Subsystems and Data Sets

The top-to-bottom flow of data shown by the arrows in Figure 14 reflects the dominant flow typical of the reactor design process. However, all Data Sets are available to all Subsystems, and there is some upward flow of data.

Before discussing the detailed function of Data Sets and Subsystems, it is necessary to first describe what is meant by a Subsystem and how a Subsystem appears both to the user and to the programmer.

The User's View

When a reactor designer uses a terminal to perform a computational step, he executes a "functional module," that is, a sequence of computations that perform a well-defined function for him. The Subsystems shown in Figure 14 are, from the user's view, functional modules.

A user supplies input data to the Subsystem, not to individual modules within the Subsystem. These data are defined and structured for the maximum convenience of the user and are permanently stored so that new data can be easily created by modifying old data. Additional input data for a Subsystem come from the Results Data Sets created by previous calculations. The output data from a Subsystem are saved on Results Data Sets if they are needed by other Subsystems or by the user. Both user and programmer are responsible for the specification of the Data Sets and functions of the Subsystem.

The Programmer's View

Figure 15 illustrates how the programmer views the Subsystem. The programmer is concerned with a large collection of computational modules and an Intermediate Data Set, which is usually retained only for the duration of a job. Three classes of modules are: Compute Modules, Input/Output Modules, and Control Modules (not illustrated in Figure 15). The Input Modules read data from the Results and Input Data Sets, organize and reformat the data for maximum convenience of the Compute Modules, and write the data to the Intermediate Data Set. The Compute Modules perform the required computations moving data to and from the Intermediate Data Set. The Output Modules then write selected data to the Results Data Sets and reports. Control Modules direct the sequence of module execution based on user supplied parameters and current values in the Intermediate Data Set. The subdivision of the Subsystem into modules, overlays, and subroutines and the structuring of the Intermediate Data Set are done by the programmer to maximize computing efficiency and minimize development and maintenance costs.

Advantages of the Subsystem Structure

The separation of the overall Application System into Subsystems, and the internal structuring of a Subsystem as illustrated above have several important advantages:

- The user is divorced from the internal operation of the Subsystem, a requirement for production use.
- The input data can be tailored to the user without affecting the performance of the Subsystem.
- A continuing program of Subsystem upgrading has minimum impact on the user.
- A high efficiency can be achieved by proper design of the internal structure of the Subsystem.
- Reuse of results on the Intermediate Data Set can substantially speed up multicase calculations.
- The Compute Modules are insensitive to changes in the design of the Input and Results Data Sets.
- The disk storage space is efficiently utilized by separating the data into high volume transitory data and low volume permanent data.

Comparison of the Data Sets

The Data Sets can be characterized by their:

- Permanence
- Structure
- Frequency of use

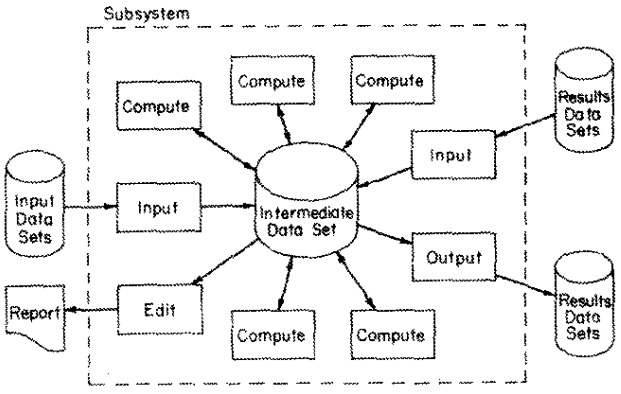


FIG. 15 Internal Structure of a Subsystem

The Input and Results Data Sets contain data of lasting value and are saved for at least the duration of the design project. Records in these Data Sets are small and structured for use by people who will create and inspect them at terminals. Records are usually read or written only once by the Input/Output Modules for each job and hence have a low frequency of use.

The Intermediate Data Set exists only for the duration of the job (or saved for at most, a few days). Records are large and structured for efficient use by the Compute Modules. Since all of the data needed by the Compute Modules can rarely be contained in main storage, the Compute Modules transfer records between main storage and disk as needed. Hence the frequency of use is high.

Because of these differences it appeared that two separate data management systems would be required. The Data Manager described previously was first developed for the user-oriented Data Sets. When used for the Intermediate Data Sets, the initial performance of the Data Manager was poor. Subsequent improvements in the Data Manager have solved these problems, and the current Data Manager is highly efficient for both types of Data Sets.

Main Storage Management

Dynamic allocation of main storage is a technique used in most JOSHUA modules to:

- Efficiently utilize available main storage
- Minimize recompilation resulting from changing problem size
- Minimize data transfer between disk and main storage

Most problem data arrays are packed into a single master data array whose size can be expanded at execution time to fill the available space in the job partition. A Main Storage Management Routine is used to manage the master data array. This routine also uses the JOSHUA Data Manager to transfer problem data arrays between disk and main storage. There is usually not enough main storage to contain all problem data arrays. Some arrays can be resident in main storage, but other arrays must be transferred between disk and main storage as needed. Several modules determine an optimum strategy based on the problem size and available main storage to minimize the transfer between disk and main storage. The Main Storage Management Routine is used to implement the optimum strategy.

BASIC DATA ANALYSIS SUBSYSTEM

Function

The function of the Basic Data Analysis Subsystem (BDASS) is to prepare all basic constants needed for both reactor physics and engineering calculations. These data fall into four classes:

- Properties of isotopes
- Neutron interaction data
- Photon production and interaction data
- Engineering properties

Sources of Data

The isotope and engineering properties were taken from a variety of handbooks and correlations done at Savannah River. The neutron and photon data are obtained from the National Neutron Cross Section Center at Brookhaven National Laboratory in the form of magnetic tapes in the format of the Evaluated Nuclear Data File/Version B (ENDF/B). These data are placed on a JOSHUA Data Set named ENDFB.

Energy and Spatial Resolution

The neutronic analysis of a reactor is usually accomplished in several steps, each of which is characterized by a different combination of neutron energy and spatial resolution. Energy is divided into discrete energy groups so that resolution is measured by the number of energy groups. Similarly, space is divided into discrete mesh points so that resolution is measured by the number of mesh points. The four typical steps are:

- Preparation of a high energy resolution (fine-group) set of neutron interaction data from the infinite resolution ENDF/B data. An assumed weighting spectrum is used.
- Calculation of the spectrum in a homogeneous medium (with leakage) using fine-group data for a few typical compositions. The resulting spectrum is used to produce a medium energy resolution (multigroup) set of data.
- Calculation of the flux in a lattice using the multigroup data and many mesh points. The resulting spectrum is used to produce a coarse energy resolution (few-group) set of data.
- Calculation of the flux in a reactor using few-group data and a large number of mesh points.

The last three steps are illustrated as shaded areas in Figure 16.

The second step is usually not necessary for thermal reactor analysis, and in BDASS it is replaced by an averaging process using an assumed spectrum. However, the Data Sets needed by each of the above steps are part of the Data Base. They are named the FINEGRP, MULTIGRP, and FEWGRP Data Sets. The design range for each of these Data Sets is shown in Figure 16.

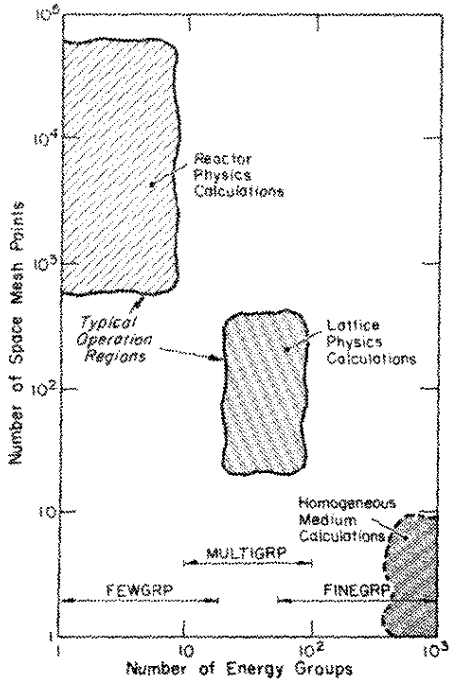


FIG. 16 Regions of Spatial and Energy Resolutions

Operation

The operation of BDASS is illustrated in Figure 17. The basic processing steps are:

- Conversion of ENDFB data to FINEGRP and NUCPARAM (resonance parameters, isotope properties, etc.) data
- Compression of the FINEGRP data to MULTIGRP data using a predefined spectrum
- Further compression of the MULTIGRP data using a predefined spectrum

The spectrum may be predefined by the user or may be obtained from a lattice physics calculation. This latter feature allows highly compressed MULTIGRP data to be prepared for parametric studies on a given lattice.

Because of the large size of the ENDFB and FINEGRP Data Sets and the infrequent need to use the BDASS, these Data Sets are not kept on-line.

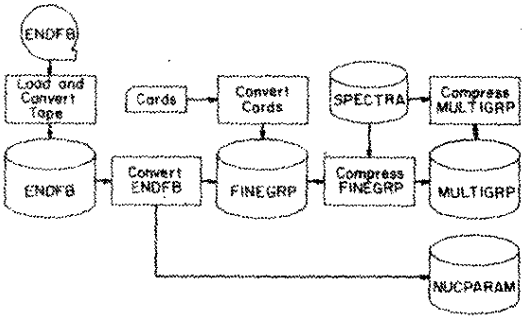


FIG. 17 Basic Data Analysis Subsystem Components

Engineering Properties

Correlated engineering properties for all materials of interest are stored on the PROPERTY Data Set. Because of the limited volume of data, the PROPERTY Data Set is maintained entirely using terminal facilities.

Data Contained on the FINEGRP and MULTIGRP Data Sets

Data cataloged by isotope, temperature, group, Legendre moment, version

Neutron reaction cross sections - total, capture, fission, transport, ν, (n,2n), (n,α), (n,d), (n,γ), (n,n'), (n,p), (n,t)

Neutron reaction matrices - elastic, inelastic, (n,2n), total

Fission production matrices

Delayed neutron production cross section, yield, spectrum by family

Photon reaction cross sections - total, coherent scattering, incoherent scattering and absorption, gamma heating, pair production, photoelectric

Photon reaction matrices - Compton and incoherent scattering

Photon production matrices - fission, (n,γ), inelastic

Data Contained on the NUCPARAM Data Set

Data cataloged by isotope and version

Single-level Breit-Wigner resolved and unresolved resonance parameters

Q-values - fission, (n,2n) effective

Radioactive decay - energy release, parent, daughter, particle emitted

Depletion chain specifications

Fission product yields

Isotope Z, A, state, weight

Data Contained on the PROPERTY Data Set

Data cataloged by material

Property types - density, heat of fusion, heat of vaporization, melting temperature, saturation pressure, saturation temperature, specific heat, surface tension, thermal conductivity, viscosity

Properties may be functions of enrichment, pressure, temperature, weight composition

Both single- and multiple-variable correlations

Correlation in polynomial form within each range of the variables

Assemblies and Lattices

The individual components loaded into the Savannah River reactors are called assemblies. Cross sections of typical fuel, target, and control assemblies are shown in Figures 18-20. Data describing the dimensions and composition of assemblies are prepared by the reactor designers and stored on an ASSEMBLY Data Set.

Assemblies are loaded into the reactor in a variety of arrangements. Each arrangement usually has a group of assemblies, which is repeated to fill a major segment of the reactor. The periodic extension of one arrangement to fill all space is called a lattice. Figure 21 illustrates an arrangement of seven cells frequently used as a lattice pattern.

Function

The function of the Generalized Lattice Analysis Subsystem (GLASS) is to provide detailed physics and engineering analyses of a lattice. These analyses fall into the following categories:

- Benchmark calculations to compare with experimental lattice measurements
- Scoping studies to provide preliminary lattice designs and performance estimates
- Design studies to predict lattice performance during normal operation
- Safety studies to predict lattice performance during abnormal operation

The first category involves a single lattice analyzed using a variety of analytic methods to establish a method that is both accurate and economical. The last three categories involve multiple lattices (usually a base case and systematic variations from the base case) using the established analytic method.

The information needed to perform these analyses comes from the ASSEMBLY, MULTIGRP, NUCPARAM, and PROPERTY Data Sets (Figure 22) along with user-supplied options. The principal results of the analysis along with the input options are stored on the LATTICE, FEWGRP, and CELL Data Sets.

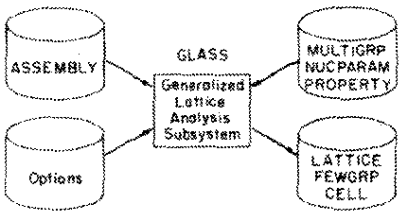


FIG. 22 GLASS Interface Data Sets

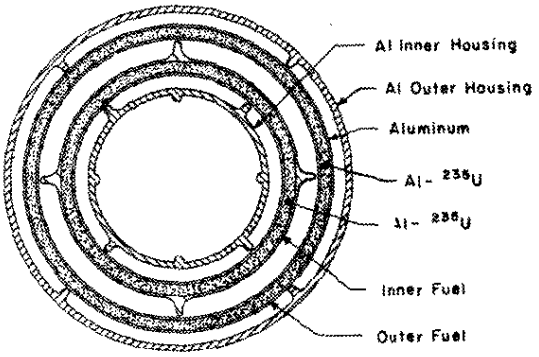


FIG. 18 Fuel Assembly

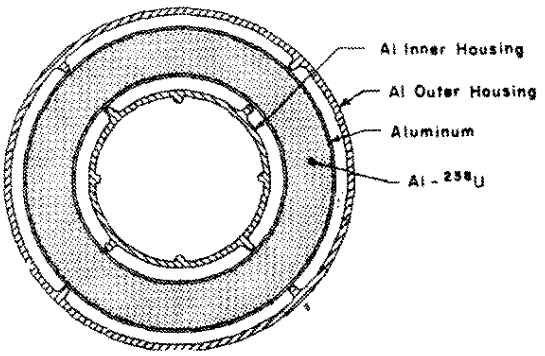


FIG. 19 Target Assembly

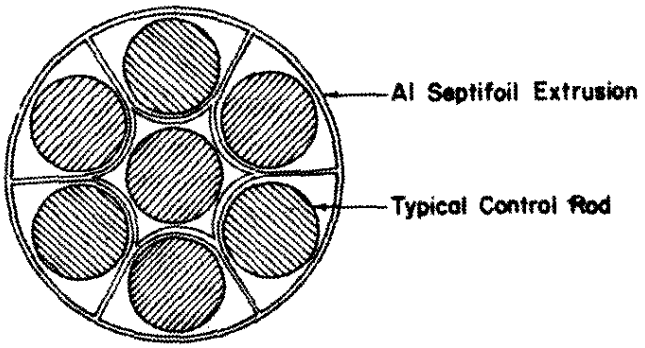


FIG. 20 Control Assembly

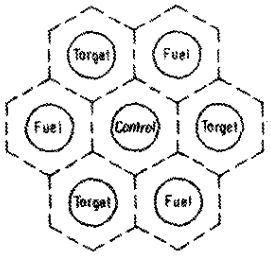


FIG. 21 Mixed Lattice Pattern

Analyses

Lattice analyses are performed in the nine major computational steps shown in Figure 23.

Input Preparation

The data on the ASSEMBLY Data Set are physical descriptions of assemblies in terms of the size and composition of each component of the assembly. These components are called regions. The computation of the neutron flux requires a spatial subdivision of regions into subregions resulting in the computational description of the assembly. When several assemblies are used as cells to form a lattice, the result is a computational description of the lattice. A computational description of the lattice in Figures 18-21 is illustrated in Figure 24. Because of the symmetry of the lattice, only three assemblies are used in the computation.

The GLASS computational modules require a lengthy and highly detailed computational description of the lattice and the computational options in the form best suited for the computational modules. On the other hand, the user can give fairly concise descriptions of what he wants done, especially in those frequent cases where he is modifying a previous calculation. Three features are provided in GLASS to transform the user's specifications into module input:

- Sufficient information is cataloged along with the lattice results to permit reconstruction of the complete input data.
- A keyword driven, free-form input language allows the user to supply the minimum information required to construct a new lattice from previously cataloged lattices.
- Terminal functions and computational modules transform the user specifications into detailed input data.

An example of the free-form language is the following:

```
/MOD=1,CELL=1,REGION=FUEL 1,TEMP=(300-600)
/MOD=2,CELL=2,REGION=COOLANT,DENSITY=(0.5,0.6,0.9)
/PROB=1,MOD=(1,2),STEPS=(4,3)/
```

These three input statements set up 12 calculations on a previously defined lattice. The first statement defines modification 1 as a variation in temperature between 300° and 600° in the region named FUEL1 of cell 1. The second statement defines modification 2 as three discrete densities in the region named COOLANT of cell 2. The third statement defines problem 1 using all permutations of modifications 1 and 2 with four equally spaced temperature steps and the three density steps.

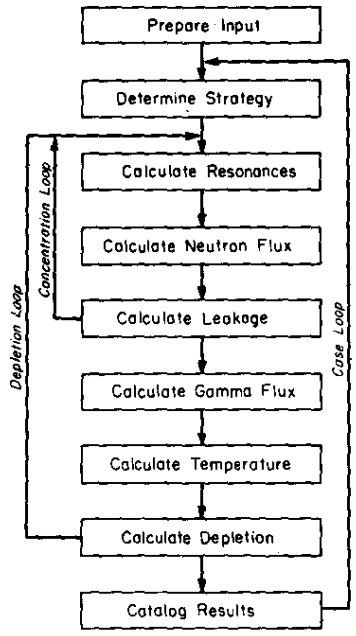


FIG. 23 GLASS Processing Steps

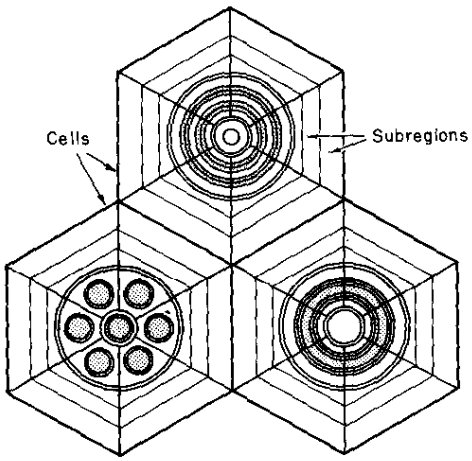


FIG. 24 Computational Representation of a Lattice

Strategy

A user sets up an average of ten cases for a GLASS execution (the case loop in Figure 23). Usually these cases are variations on the same basic lattice. All calculations must be done for the first case, but many calculations need not be repeated for the following cases. A computational strategy is determined by comparing the input data for the current case with the input data for the previous case and identifying those data records on an Intermediate Data Set that must be recomputed and those modules that need to be re-executed. A control module uses this information to minimize the computation. Successive cases are executed, on the average, in one-third of the time required for the first case. It should be noted that it is the dynamic module execution and random access data base features of a modular data-based system that makes these efficiency gains possible.

Resonance Capture

The main features of the methods used to calculate resonance integrals are:

- Nordheim integral treatment
- Multiregion annular geometry description of cell
- Cosine-current calculation of collision probabilities
- Interpolation of tabular collision probabilities
- Cell-to-cell interactions treated by conservative boundary conditions

Neutron Flux

The neutron flux is calculated in each energy group and each subregion of the lattice. This flux is called the fine-structure flux. The basic equation used is the multigroup neutron transport equation. Three numerical methods for solving the transport equation are available to the user:

- Collision Probability Method
- Transmission Probability Method (also called the Interface Current Method)
- Monte Carlo Method

Each of these methods can treat both an arbitrary group structure with full or partial group transfer matrices and an arbitrary two-dimensional geometry subdivided by linear segments and circular arcs. However, the current implementation of the Transmission Probability Method and the Input Preparation Modules supports only those geometries used at Savannah River plus those needed for square lattices.

Each of these methods can be used in two ways. The first (and usual) way is to treat all cells in the lattice simultaneously to obtain the complete fine-structure flux. The second is to treat each cell separately by first computing a neutron current response matrix for each cell and then solving for the cell-to-cell currents in the lattice. This approach is called the response matrix method. Response matrices are stored in the CELL Data Set for later use in reactor calculations. However, implementation for routine use of the response matrices in lattice calculations has not been completed.

Leakage

The effect of leakage on the neutron flux is estimated by performing a multigroup B_1 calculation on the uniform material obtained when the lattice is spatially homogenized using the fine-structure flux as a weighting function. A variety of eigenvalue searches can be performed during the B_1 calculation. Given any three of the following four items:

- Effective multiplication factor
- Time decay constant
- Geometric buckling
- Isotopic concentration

the fourth item will be determined. Since the time decay constant (treated as a negative absorber) and isotopic concentration affect the fine-structure flux, prediction of these quantities requires some iteration. This is illustrated in Figure 23 as a concentration loop.

Gamma Flux

The fine-structure neutron flux is used along with the photon production data on the MULTIGRP Data Set to calculate a gamma source distribution in the lattice. A multigroup collision probability calculation predicts the resulting gamma heat deposited in each subregion of the lattice.

Temperature

The two-dimensional fission and gamma heat distributions in the lattice are multiplied by a known axial shape factor to obtain the heat generation rates in the three-dimensional lattice. A multichannel thermal calculation is used to predict temperatures at each location in the lattice. The required engineering parameters are taken from the PROPERTY Data Set.

Depletion

The neutron flux can be used to calculate the expected isotopic depletion by region during constant power operation. The constant power may apply to one assembly or the entire lattice. The changes in isotope concentrations cause the neutron flux to change so that it is necessary to repeat the neutron flux calculation (depletion loop in Figure 23) periodically. The frequency of repeating the neutron flux calculation is determined by the depletion

modules based on a user-specified change in the concentration of an isotope in a region. Criticality of the lattice during depletion is maintained by varying buckling, eigenvalue, time eigenvalue, or control isotope concentration.

When the lattice is placed in a reactor, the depletion history will not be identical to the idealized depletion history of the lattice alone. The intent of depleting the lattice is to obtain reasonable neutron flux spectra from which few-group microscopic reaction cross sections can be calculated. The microscopic cross sections will then be used to deplete assemblies during a reactor depletion calculation.

Cataloging of Results

Results of lasting value are cataloged and stored on the FEWGRP and LATTICE Data Sets. These results generated at a rate of several hundred lattices per day for the past several years provide a valuable library of lattice physics information. Inquiry modules locate for the user those lattices that satisfy his search criteria. For example, he may want to know what lattices have a k_{eff} between 0.99 and 1.02, a poison concentration in a given range, and a ^{235}U content of a given amount.

Data Contained in the CELL Data Set

Data cataloged by cell and version

Back-reference to ASSEMBLY data set

Changes to cell (homogenization, subdivision)

Group structure and spectrum for surface currents

For each surface current component and fission generation:

- *Outward surface currents by group and surface*
- *Cell average flux by group*
- *Cell average reaction rates by group*
- *Cell average scattering matrix*

GLASS Problem Size Statistics

	Typical	Maximum
Neutron Groups	37	100
Subregions	100	800
Different Cells	3	50
Materials	15	200
Isotopes	15	200
Resonance Isotopes	10	200
Depletion Steps	5	30
Depletion Chains	3	200
Depletion Isotopes	50	200
Photon Groups	6	100

Data Contained on the ASSEMBLY Data Set

Data cataloged by cell name and version

Geometric data for annular, cluster, hexagonal, and square type subassemblies

Material densities and temperatures

Isotope concentrations

Definition of standard materials

Data Contained in the LATTICE Data Set

Data cataloged by lattice name

ASSEMBLY data for each cell

Lattice pattern description

For each version of this lattice:

- *Calculation options (homogenization, subdivision method, etc.)*
- *For each state of this version:*
  - *Changes producing this state*
  - *Integral results (k_{eff}, B^2_m, etc.)*
  - *Lattice average Legendre flux and slowing-down density by group*
  - *Fission rate by region*

Data Contained on the FEWGRP Data Set

Data cataloged by lattice, cell, group, lattice state

Flux

Macroscopic cross sections - diffusion, capture, fission, production

Scattering and production matrices

Legendre scattering matrices

Microscopic cross sections by isotope - smooth and resonance capture, smooth and resonance fission, neutrons/fission, self-shielding factor, (n,α), (n,p), (n,γ^*), $(n,2n)$

Delayed neutron inverse velocity, production cross section, spectrum

GLASS CPU Time for Typical Cases

Calculation	CPU Time (seconds)
Resonance	40
Neutron Flux	50
Leakage	1
Gamma Flux	40
Temperature	10
Depletion	15

Function

The state of each lattice cell changes with time during a reactor kinetic or depletion calculation. The state is characterized by composition, densities, temperatures, and environment. However, it is not known in advance what states will actually be experienced during the reactor calculation.

The strategy employed in JOSHUA is to perform lattice calculations at a carefully selected set of discrete lattice states which span the range of expected states, and correlate the lattice parameters into a set of parametric equations. The parametric equations give the lattice parameters as a continuous function of the lattice state. Lattice parameters for each state needed during the reactor calculations are obtained by evaluation of the parametric equations, which, in effect, interpolate between the discrete states actually calculated.

The function of the Correlation Reduction Analysis Subsystem (CRASS) as illustrated in Figure 25 is to take results of lattice calculations stored on the LATTICE and FEWGRP Data Sets, correlate them based on user-supplied options, and store the resulting parametric equations on the PARMEQ Data Set.

The analysis steps are shown in Figure 26. The iteration shown in Figure 26 reflects the trial-and-error nature of the correlation procedure.

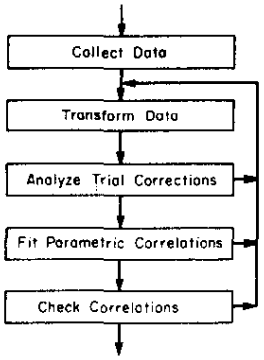


FIG. 26 CRASS Processing Steps

Applications

The two most important applications of correlated lattice parameters are charge design and reactor kinetics. For each of these applications, variables describing the lattice states are called the independent variables, and the lattice parameters are called the dependent variables.

The independent variables used for charge design are:

- Design parameters such as fuel loading, burnable poison concentration, enrichment, control rod strength, etc.

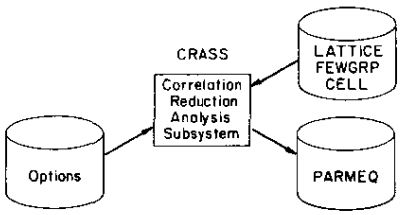


FIG. 25 CRASS Interface Data Sets

- Operating parameters such as exposure and control rod position

The dependent variables used in charge design are:

- Macroscopic cross sections by energy group and cell
- Microscopic cross sections by energy group, cell, isotope, and reaction type

The independent variables used in a reactor kinetics calculation are:

- Metal temperatures by cell
- Coolant density by cell
- Moderator density
- Fraction of metal melted by cell
- Concentration of melted particles in the moderator
- Concentration of dissolved poison in the moderator

The dependent variables used in a reactor kinetics calculation are:

- Macroscopic cross sections by energy group and cell

Data Collection

The typical numbers of dependent and independent variables and lattice states are listed in Table 1. Reorganization of collected data is necessary because the lattice calculation produces all dependent variables for one state, but the correlation requires one dependent variable for all states. The JOSHUA data management facilities are essential for both the storage of the lattice parameters and their reorganization prior to correlation.

TABLE 1

Typical Correlations

	Charge Design	*Reactor Kinetics*
Number of Independent Variables	5	10
Total Number of Lattice States	65	230
Number of Dependent Variables/State	200	20
Total Number of Dependent Variables	13,000	4,600

Parametric Equations

The functional form of the parametric equations was chosen to be a multivariate, low-order polynomial form. This form was chosen because the coefficients could be obtained by linear least-squares fitting methods. Although low-order polynomials are adequate to describe weak variations, transformations of the variables are required for the stronger variations.

Transformations

CRASS allows the user to transform both the dependent and independent variables into new variables that better correlate the data. Physics arguments indicate that certain combinations of variables (such as spectral indices) are better correlating variables than the state variables, and certain combinations of cross sections (such as the diffusion length) are better correlated than the individual cross sections. Mathematical arguments indicate that selective stretching of the dependent and independent variable axes increases the accuracy of the low-order polynomial form. CRASS provides the mechanism to accomplish and evaluate the transformations.

Analyses

CRASS does not automate the correlation process. There are too many human decisions to make. CRASS assists the user in the correlation process by performing the data management and computational tasks and by providing analyses of the correlation. Several types of analyses have been implemented: regression, mathematical transformation, and interactive graphics.

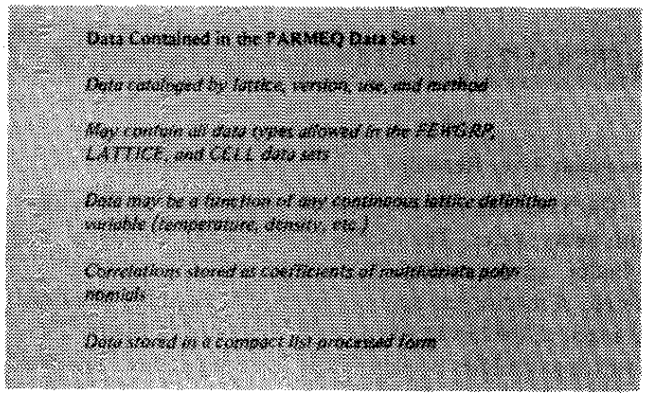
A standard regression analysis is used to indicate the degree of correlation between the pairs of powers of the dependent and independent transformed variables, helping the user to select possible transformations and polynomial forms. A mathematical transformation analysis uses result: of a trial correlation to predict axis stretching parameters.

Interactive graphics, the most poweful analysis tool developed to date, is used to:

- Execute modules to perform transformations and trial correlations
- Select and display results in graphical form (Figure 27)
- Judge the behavior of the curves and quality of the correlation

Fitting

When the user has decided how the correlation is to be accomplished, CRASS fits the data to the selected form and stores the results on the PARMEQ Data Set. The correlation for a new type of lattice requires the elaborate analysis discussed previously. However, subsequent correlations on closely related lattices would not require a full analysis to find the best parametric equations and transformation. In these cases, the fitting procedure will use the results of the previous correlation along with the new data to produce a new set of parametric equations.



Checking

Even when the data are fit to an acceptable precision at the lattice states used in the fitting process, further checking must be done before the parametric equations can be used in a reactor calculation. Redundant data in the parametric equations are used to evaluate the accuracy of combinations of dependent variables at lattice states not used in the fitting process. For example, values of the effective multiplication factor (k_{eff}) and migration area (M^2) are correlated along with the cross sections from which they can be deduced. Values of these quantities and their derivatives with respect to each state variable are obtained by:

- Evaluation of the parametric equations
- Calculation from cross sections obtained by evaluation of the parametric equations

A comparison of these results for a large number of randomly chosen states indicates the accuracy of the correlations when used in a one-group reactor calculation.

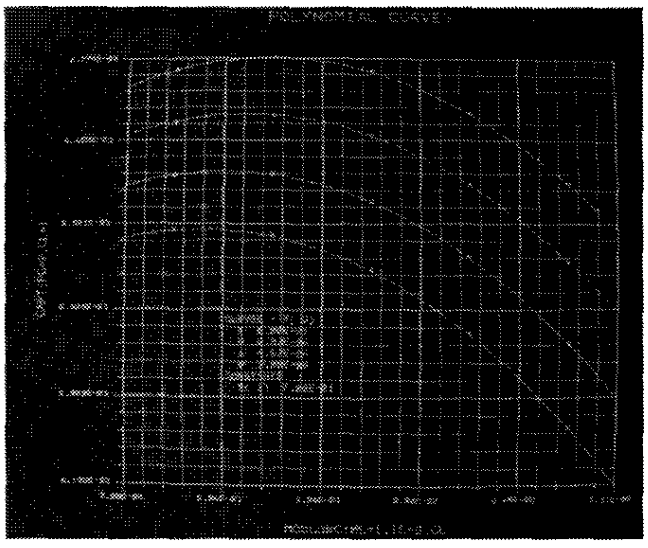


FIG. 27 Graphical Display of Typical Correlated Data

Savannah River Reactors

The Savannah River reactors are cooled and moderated with heavy water. The reactor tank is a right circular cylinder, approximately 18 feet high and 16 feet in diameter. A reactor has approximately 673 assembly positions arranged in the hexagonal pattern illustrated in Figure 28. Control assemblies occupy 61 of these positions. Sparjets and gas ports occupy 12 positions. The remaining 600 positions are occupied by either fuel or target assemblies. Six coolant inlet nozzles are shown surrounding the tank in Figure 28.

Function of GRASS

The function of the Generalized Reactor Analysis Subsystem (GRASS) is to provide detailed physics and engineering design and safety analyses of these reactors. The data needed to perform these analyses are (Figure 29):

- Cross section data for each different type of assembly or lattice of assemblies represented as parametric equations on the PARMEQ Data Set
- A description of the reactor loading pattern, reactor geometry, and its subdivisions created by the user and stored on the REACTOR Data Set
- User-supplied control and calculational options

The analyses performed by GRASS fall into the two major categories of static and kinetic analyses.

Static Analyses

The steps used for static analyses are illustrated in Figure 30.

Input Preparation

GRASS modules require a detailed, three-dimensional description of the reactor. Whereas in GLASS, powerful language and terminal facilities have been developed to allow convenient input data preparation, comparable facilities have yet to be developed for GRASS. However, reactor descriptions previously cataloged on the permanent REACTOR Data Set can be easily modified to create new reactor descriptions.

The description of reactor geometry proceeds in the following steps:

- A partitioning of the reactor into planar zones and axial segments. The three-dimensional combination of a zone and segment is called a reactor region.
- A regular partitioning of a reactor region into mesh points used for the flux calculation.

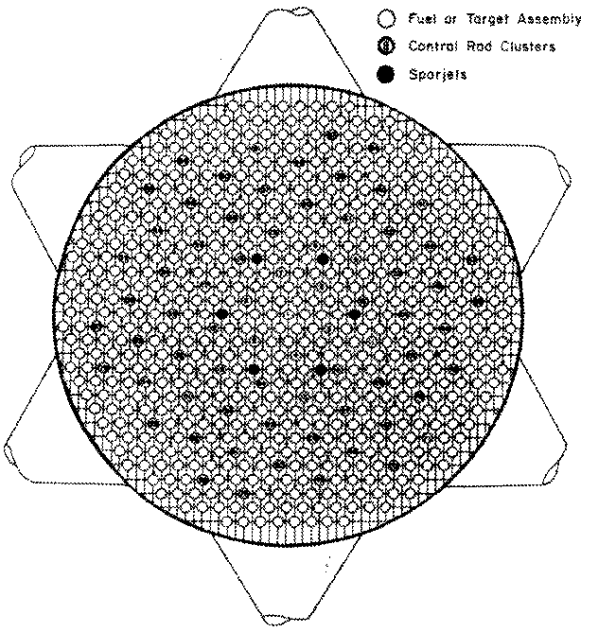


FIG. 28 SRP Reactor Lattice

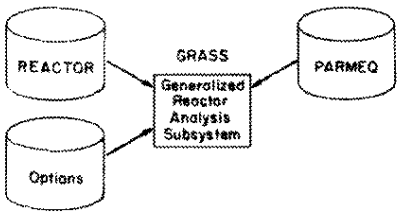


FIG. 29 GRASS Interface Data Sets

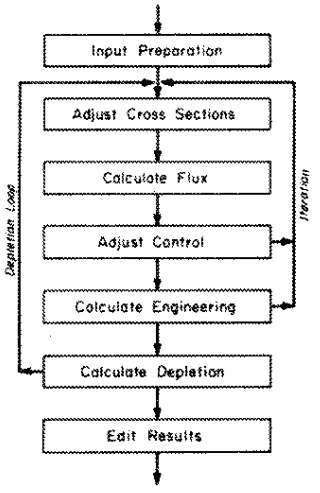


FIG. 30 GRASS Processing Steps of Static Analyses

- An arbitrary collecting of reactor regions into six other aggregations called:

Cross Section Regions	Core Regions
Control Regions	Moderator Regions
Depletion Regions	Edit Regions

The remaining input information describes the properties of regions. For example, cross section sets on the PARMEQ Data Set are assigned to cross section regions, control settings are assigned to control regions, thermal-hydraulic constants are assigned to core regions, and so forth.

Input preparation modules convert these user-supplied descriptions to the detailed tables required by the calculational modules and store them on an Intermediate Data Set.

Cross Section Adjustment

Each type of region has associated with it a set of state variables which are uniform within the region. For example, the state variables for a depletion region are isotopic concentrations, the state variables for a core region are fuel temperature, coolant density, fraction of metal melted, and so forth. The initial values of the state variables may be user-specified, obtained from previous calculations, or obtained from default conditions. Many of the values of the state variables will change during the course of the GRASS calculation as indicated by the iteration and depletion loop in Figure 30.

The parametric equations assigned to a cross section region give cross sections as a function of the state variables in that cross section region. Since the various types of regions are each arbitrary collections of reactor regions, the first step is the calculation of the average value of the state variables which occur in a cross section region. The average state variables are then used in the parametric equations to prepare the cross sections needed for the flux, control, and depletion calculations.

Data Contained in the REACTOR Data Set

Data cataloged by reactor name and version

Data partitioning the reactor into planar zones and axial segments. The zone-segment combination is a reactor region.

Data partitioning zones and segments into mesh points for the flux calculation

Data collecting reactor regions into:

Cross Section	*Core*
Control	*Moderator*
Depletion	*Edit*

Material names by cross section region

Control strength by control region

Direct or adjoint source by reactor region and group

Direct or adjoint flux guess by mesh point and group

Neutron Flux

The neutron flux in the reactor is calculated using finite-difference, few-group diffusion theory, usually with either two or four neutron groups. Most calculations are performed for two-dimensional hexagonal or three-dimensional hex-z geometry with 1, 3, or 6 mesh points per hex. Modules for three-dimensional orthogonal geometries are also available but are not yet fully supported by the input preparation and other processing steps.

A variety of flux calculational options are available to the user:

- Direct or adjoint flux
- Fixed source or eigenvalue search
- Eigenvalue may be k_{eff}, buckling, or poison concentration
- Control adjustment

The following are the main features of the flux calculation:

- Cross sections given by the material in each cross section region
- Arbitrary scattering matrices and fission spectra given by material
- Calculation done one energy group at a time
- Coarse mesh rebalancing, fission source extrapolation, and simultaneous line relaxation convergence acceleration methods
- Dynamic main storage allocation based on actual problem size
- Dynamic data roll-in/roll-out strategy determined at execution time

Control Adjustment

The spatial distribution of control absorption may be adjusted until the reactor power distribution agrees (within a tolerance) with a user-specified distribution. This procedure is also called power flattening. The iteration required to obtain the desired distribution is performed in two phases. The first phase involves varying a control capture cross section distribution until the desired power distribution is obtained. This phase is performed simultaneously with the flux calculation. The second phase involves adjusting all cross sections using the parametric equations and the current estimate of control capture. This is shown as an iteration in Figure 30.

Engineering Model

The computed reactor power distributions can be used to predict the engineering state of the reactor. The models used are steady-state versions of the kinetic models that will be described later. The cross sections vary with engineering state variables; the iteration shown in Figure 30 is required to obtain a reactor solution for which engineering and neutronic states are consistent.

Depletion

The instantaneous state of a reactor is predicted by calculations described in preceding sections. A depletion calculation predicts the change in isotopic composition which occurs during reactor operation at constant power. The depletion calculation continues until the changing composition necessitates the recalculation of the power distribution. This recalculation is shown as the depletion loop in Figure 30. The frequency of recalculation is determined by the fractional change in isotopes and regions specified by the user.

Two methods are used to calculate depletion. In the first method, isotopic reaction rates in each depletion region are calculated from the average few-group flux and the microscopic cross sections from the parametric equations. The depletion equations are then solved for isotope concentrations as a function of time. This calculational method is the same as used in the lattice calculation. In the second method, an exposure rate is calculated from the average few-group flux in each depletion region and the macroscopic fission cross section. The isotope concentrations as a function of time are obtained from the parametric equations using exposures calculated from the constant exposure rate. The first method is the more accurate but can be time consuming when many depletion regions are used. The second method requires little computing time but is valid only if the depletion history of the lattice calculated by GLASS is similar to the actual history in the reactor.

GRASS CPU Times for Typical Static Cases

	CPU Time (seconds) per Calculation
Cross Section	2
Flux	30
Engineering	30
Depletion	180

Edits

At the present time, no JOSHUA Subsystem uses the results of the GRASS calculation. Consequently, no permanent output Data Set has been defined. Selected results are saved on the REACTOR Data Set to facilitate input preparation and calculation of other cases. Most results are saved (temporarily) on the Intermediate Data Set for use in a GRASS kinetic calculation.

Response Matrix Method

Transport theory calculations using the response matrix method are used to evaluate the accuracy of the finite-difference diffusion theory methods. The response matrices for each cell type are calculated in GLASS. The cells are assembled into a reactor using the loading pattern on the REACTOR Data Set, and the cell coupling currents are obtained by iteration. Features of the method are:

- Two-dimensional hexagonal geometry
- Multigroup integral transport theory used within a cell
- Cells coupled with few-group interface currents
- Interface currents on each hexagonal face may have up to four angular/spatial modes

GRASS Problem Size Statistics

	Typical	Maximum[a]
Neutron Groups	2	20
Reactor Regions	3010	60000
Mesh Points	3010	360000
Cross Section Regions	3010	60000
Control Regions	26	60000
Depletion Regions	300	60000
Core Regions	2290	34000
Moderator Regions	410	40000

a. These maxima are computed from fixed dimensions in the modules. Actual maxima are dictated by the size of available main storage.

Kinetic Analyses

A major objective in the development of the JOSHUA System is to provide computational models for the safety analyses of the Savannah River reactors. One facet of safety analysis is the prediction of the kinetic behavior of the reactors prior to extensive core configuration changes. These analyses are here called kinetic analyses.

The approach taken to provide kinetic analysis capability is to first develop the most comprehensive model possible and then use the results of this model both to validate the current production models and to guide the development of more advanced production models. The comprehensive model of the reactor system developed in GRASS is a full three-dimensional model of all physical phenomena thought to be important and for which models could be constructed. The resulting computational system requires several hours of computing time for a single kinetic calculation. This system demonstrates that a comprehensive model can be built and that it in fact models observed reactor behavior. Refinement of the numerical and computational methods will hopefully reduce the computing time to that needed for extensive parametric studies.

Computational Steps

The computational steps involved in a GRASS kinetic analysis are illustrated in Figure 31. A steady-state reactor is first calculated using the methods previously described. A time-dependent perturbation is introduced, and the state of the reactor is calculated at successive time steps (the time loop in Figure 31).

Reactor Perturbations

A wide variety of both real and hypothetical perturbations can be simulated to initiate a reactor transient. These include:

- Cross section changes
- Control rod movements
- Loss of assembly flow
- Pump shaft breaks
- Pump coastdown from loss of pump power
- Pipe breaks
- Injection of soluble poison

Cross Section Adjustment

Feedback from changes in the engineering state of the reactor is treated by adjusting neutron cross sections using the parametric equations obtained by correlating cross sections with engineering state parameters. The methods are the same as previously described for static analyses.

Neutron Flux

The neutron flux at a given time step is calculated by finite-difference, few-group diffusion theory. The methods used are extensions of those previously described for static analyses. The main features of the transient flux calculation are:

- Fully implicit time differencing
- Frequency transformation
- Arbitrary number of delayed groups
- Group velocities and delayed neutron parameters given by material and correlated like cross sections
- Coarse mesh rebalancing, fission source extrapolation, and simultaneous line relaxation convergence acceleration methods at each time step

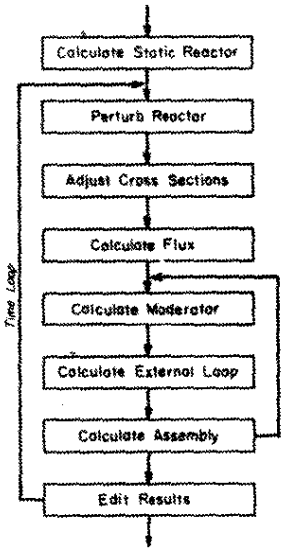


FIG. 31 GRASS Processing Steps of Kinetic Analyses

Engineering Models

The engineering models describe the space- and time-dependent thermal, hydraulic, and mechanical state of the reactor. A schematic of the reactor hydraulic system is given in Figure 32. Within the reactor tank, the engineering models can use a spatial resolution as fine as a reactor region. Three major parts of the engineering calculation are:

- Moderator calculation
- External loop and assembly flow calculation
- Assembly thermal calculation

Iteration as illustrated in Figure 31 between these three parts is necessary because of the strong pressure feedback during postulated reactor transients that lead to steam generation or loss of coolant.

Moderator Model

Heavy water flows into the moderator space from the bottom fittings of assemblies and from sparjets and control assemblies. The three-dimensional flow of the moderator is illustrated by the arrows in Figure 32.

A transient, three-dimensional (hex-z), two-phase, potential flow model is used to describe moderator flow. The main features of this model are:

- Separate diffusion coefficients for axial and radial flow
- Spatial resolution may be either a hex or a patch (group of 7 hexes as illustrated in Figure 21)
- Spatially dependent boiling and condensation rates determined by assuming thermodynamic equilibrium or by using correlations in terms of degree of subcooling and void fraction
- Soluble poison transport
- Transport of particles from melted assemblies (currently being developed)
- Gamma heat sources
- Heat transfer from assemblies
- Detailed description of the reactor vent system yielding time-dependent system pressures

External Loop and Assembly Flow Model

The external hydraulic system (Figure 32) consists of six independent loops each having a pump and two parallel heat exchangers. These six loops feed heavy water coolant into six nozzles spaced around the outside of the coolant plenum. These six nozzles are illustrated surrounding the reactor lattice in Figure 28. The plenum feeds coolant to the top fittings of fuel and target assemblies and sparjets.

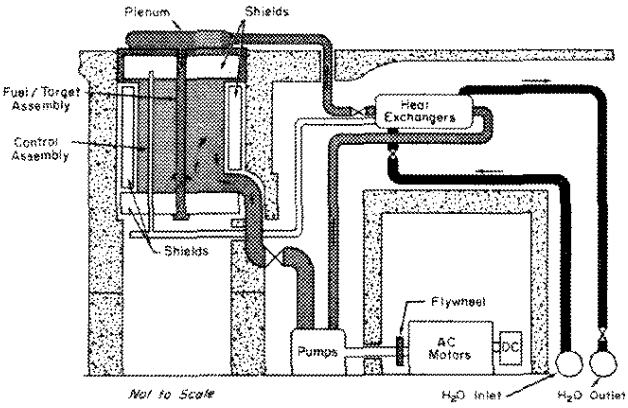


FIG. 32 SRP Reactor Coolant System

The main features of the external loop and assembly flow model are:

- Two-dimensional hexagonal mesh used to describe flow paths in the plenum
- Transient flow - ΔP relationships describe every flow path in external piping and assemblies
- Hardy-Cross iterative technique employed to solve nonlinear flow - ΔP equations
- External leaks and pump conditions such as cavitation and coastdown are treated

Assembly Thermal Model

Coolant flows from the plenum, through annular assembly coolant channels, and into the moderator space (Figure 32). An equivalent single-channel, two-phase transient thermal-hydraulic model is used to determine fuel temperatures, housing temperatures, coolant temperatures, qualities, and pressures. The main features of this model are:

- Each assembly or groups of assemblies can be described
- Stable (liquid phase) assembly dynamics described by conservation of mass, energy, and momentum
- Flow instability described by experimental correlations and conservation of mass and energy
- Fuel and housing melting treated with empirical models

Edits

Data on the Intermediate Data Set describe the state of the reactor at the current computation time. Selected data are saved at user-specified times throughout the transient calculation for post-execution editing and computation restart.

Transient Response Matrix Method

Time-dependent diffusion theory with homogenized cross sections may not be adequate to describe low density (voided) regions and regions with large flux gradients. The static response matrix method (previously described) has been extended to treat time-dependent problems in two-dimensional geometries. The transient response matrix method is being used to investigate the validity and accuracy of:

- Cross section homogenization procedures
- Diffusion theory in voided regions
- Albedos to replace reflectors
- New procedures for obtaining diffusion parameters

Analysis of GRASS Results

Three-dimensional, time-dependent GRASS calculations generate a large volume of data of potential interest to the user. If GRASS were a mature production Subsystem, the users could specify the limited collection of results that he wants, and these results could be printed and/or saved in the data base. GLASS is an example of this type of mature Subsystem. However, GRASS is still in the category of a research tool that generates a large volume of potentially useful results. The user must be able to select those results that are significant. An interactive graphics system combined with the JOSHUA data management system is being developed for this purpose. This system will display on request the following types of graphs:

- Reactor-average results as a function of time (Figure 33)
- Planar-average or single-channel axial distributions at selected times (Figure 34)
- Axial-average or single-level planar distributions at selected times (Figure 35)
- Perspective drawings of selected two-dimensional distributions

GRASS CPU Times for Typical Kinetic Cases

	CPU Time (seconds) per Reactor Time Step
Cross Section	3
Flux	15
Moderator	10
External Loop	10
Assembly	10

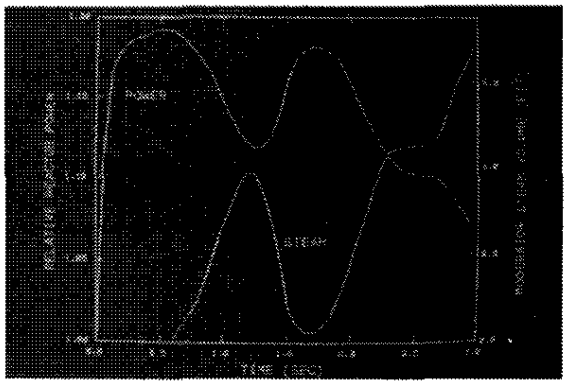


FIG. 33 Reactor Power and Steam Volume as a Function of Time for a Sample Kinetics Calculation (photograph of the graphics terminal screen)

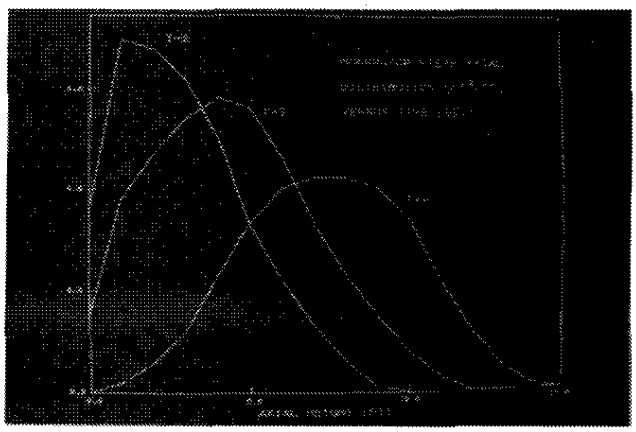


FIG. 34 Axial Distribution of Steam at Selected Times for a Sample Kinetics Calculation (photograph of the graphics terminal screen)

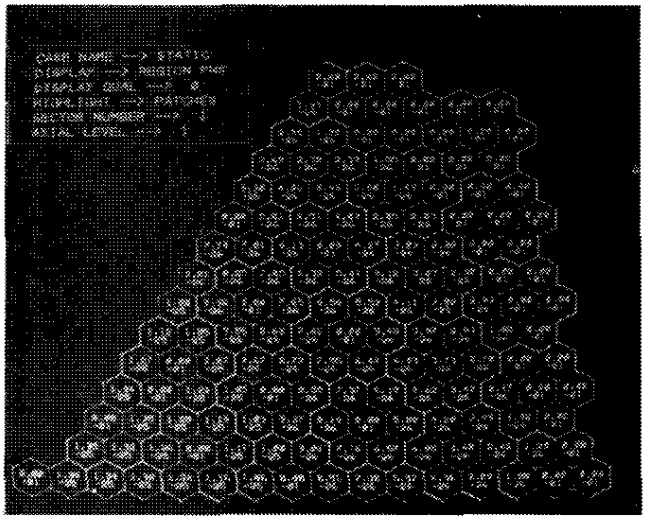


FIG. 35 One-Sector Power Map (photograph of the graphics terminal screen)

Specialized Application Subsystems

SCOPE

Specialized Application Subsystem is the term used here to describe a wide variety of application modules executed using the JOSHUA Operating System. The incentives for developing these Subsystems include:

- Extensive use of existing codes and technology
- Convenience of terminal input/output
- Use of the Data Manager to interface codes
- Use of job execution facilities to automate code execution
- Satisfy near-term needs
- Minimal development cost

Many such Subsystems (or single modules) have been developed; only the five major Subsystems involved with reactor physics and engineering will be described here. These five Subsystems are illustrated in Figure 36.

The Subsystems shown in Figure 36 may be executed individually. Control modules have also been developed to execute the Subsystems in the proper sequence. A frequently used combination is:

- Flow-Zoning Analysis
- Thermal-Hydraulic Limits Analysis
- Film-Boiling Burnout Analysis

This combination is referred to in later sections as the Operating Limits Analysis Subsystem.

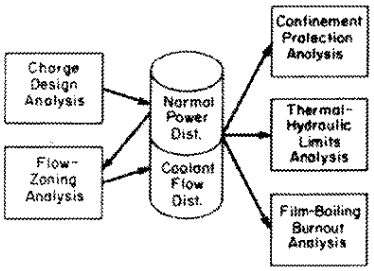


FIG. 36 Specialized Application Subsystems

CHARGE DESIGN ANALYSIS SUBSYSTEM

Charge Design Analysis is used to predict the normal operating characteristics of a reactor charge as a function of depletion. The main steps in the analysis are illustrated in Figure 37.

Radial Power Distribution

The reactor power distribution is factored into a specified axial shape and a calculated radial power distribution. The radial power distribution is calculated using finite-difference, two-group diffusion theory in hexagonal geometry.

Fine-Structure Corrections

In some cases, the diffusion theory prediction of power in the nonfuel assemblies is not adequate. In these cases, the transport theory predictions of the fine-structure flux from GLASS calculations are used to correct the power estimates for the nonfuel assemblies.

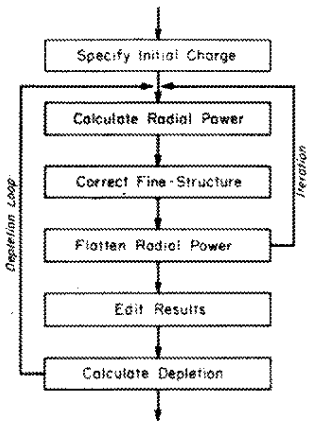


FIG. 37 Charge Design Analysis Processing Steps

Power Flattening

Diffusion parameters at the control rod positions are adjusted until the radial power distribution satisfies user-specified criteria. Options allow the user to specify the assemblies to be used in calculating flatness and the aggregation of control rods used to achieve flatness.

Edits

A variety of results computed and saved for later use include:

- Thermal power distributions as the sum of fission and nonfission power
- Power and exposure distributions which can be directly compared with those reported by the reactor on-line computer
- Buckling distributions

Depletion

Isotope concentrations are precalculated using GLASS and correlated as a function of fission exposure. A three-dimensional power distribution is synthesized from the specified axial distribution and computed radial distribution. The reactor is depleted by prescribing a reactor-average depletion increment, computing local exposures at the end of the increment, and calculating isotope concentrations by interpolation in the correlation tables. The three-dimensional composition distributions are collapsed to two dimensions for use in the next radial power calculation.

The depletion loop (Figure 37) continues until the prescribed endpoint is reached.

FLOW-ZONING ANALYSIS SUBSYSTEM

Flow-zoning analysis is used to determine the number of distinct flow zones required and the flow rates for each zone. The main steps in the analysis are illustrated in Figure 38.

Power Distribution

The normal power distribution calculated in the Charge Design Analysis could serve as the basis for deriving the optimum distribution of assembly coolant flow. If prior experience with similar reactor charges is available, however, measured/calculated assembly power ratios are used to refine the calculated distribution.

Zone Assignment

A preliminary estimate of the "best" flow distribution is derived by assuming that the highest powered assemblies in each zone operate at the same coolant effluent temperature. For any given number of zones, the boundaries between zones are adjusted to yield the highest total reactor power. The number of zones considered is increased until the total reactor power is within some fraction of the theoretical limit resulting when each assembly has a flow proportional to its power. This calculation assigns assemblies to flow zones and provides preliminary estimates of relative zone flows.

Relative Limits

The preliminary estimate from the flow zone calculation is adjusted by using constraints derived from consideration

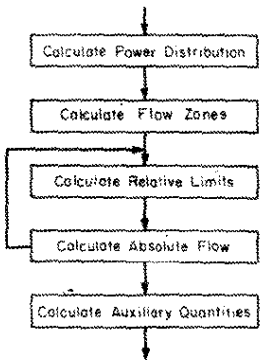


FIG. 38 Flow-Zoning Analysis Processing Steps

of reactor operating procedures. One such constraint is that flow zones containing assemblies in regions of the reactor where the power distribution cannot be readily "trimmed" by adjusting control rods may not approach limits as closely as those within the span of the control rods. Flows to such zones are adjusted by empirically established margins. The other major constraint is that, among neighboring fuel and target assemblies, fuel assemblies must approach limits more closely than target assemblies. If necessary, target flows are adjusted with respect to fuel flows.

Absolute Flow

Absolute flow rates are calculated from the derived relative coolant flows in fuel and target assemblies and the known hydraulic characteristics of the remainder of the D_2O circulating system (including the control rod cooling system, the moderator jet flow system, and the external loop). Values were assumed for absolute flows in the preceding Relative Limits Calculation; therefore, iteration between these two calculations is required as indicated in Figure 38.

Auxiliary Quantities

The intended assembly coolant flow rates are realized by supplying appropriate top orifice plates for each assembly. The Absolute Flow Calculation establishes the required pressure drops across these pieces. The pressure drops are translated to numbers and sizes of orifices, using empirical relationships.

CONFINEMENT PROTECTION ANALYSIS SUBSYSTEM

One type of routine safety analysis at Savannah River predicts the pressure surge in the reactor confinement system for hypothetical accidents in which the safety system fails. Analyses of these accidents are used to design charges and define operating limits which ensure that the reactor confinement system will not be breached.

If the safety system fails to operate following an initiating perturbation, the course of the resulting transient has three main phases:

- Initial power increase caused by the initiator
- Assembly flow instability resulting in steam injection into the moderator
- Assembly melting which terminates the transient

Several assumptions are made to simplify the analysis:

- The three phases occur in the sequence given above.
- The configuration of the reactor at the start of each phase can be precalculated.
- The space-time power distribution can be synthesized from a point-kinetics prediction of power level, a series of calculated static two-dimensional planar distributions, and a specified axial distribution. The planar distributions are obtained using two-group diffusion theory.
- Assemblies of the same type having similar power histories can be treated as a group with events occurring simultaneously in all assemblies in the group.

Using these assumptions the analyses can proceed in the sequence shown in Figure 39.

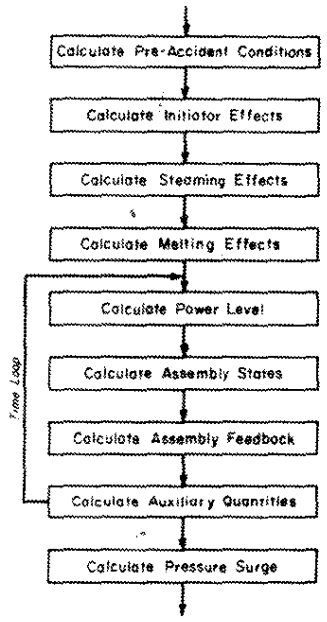


FIG. 39 Confinement Protection Analysis Processing Steps

Pre-Accident Conditions

Two-dimensional power distributions and reactivity are computed for the reactor prior to the start of the accident. The results are used along with the initiator effects to describe the initiator phase of the accident.

Initiator Effects

Two-dimensional power distributions and reactivity are computed for the reactor with a specified fraction of the full initiator in effect. The pre-accident and post-accident distributions along with the known time variation of the initiator determine the sequence in which individual assemblies will steam and melt. These are collected into assembly groups, and all assemblies in a group are assumed to steam and melt at the same time.

Steaming Effects

The condition of the reactor at the onset of steaming is presumed known from the previous calculation of initiator effects. The spatial distribution of steam injected into the moderator space by a single steaming assembly is predefined as a function of time after the onset of steaming. The distribution from several assemblies steaming is calculated from the locations of the steaming assemblies. A two-group, three-dimensional static diffusion theory calculation is done with the maximum amount of steam from the first assembly group. The resulting power distortions (collapsed to two dimensions) and reactivity have the predefined time variation. Similar calculations are done for the subsequent assembly groups assuming each prior assembly group is at maximum steaming.

Melting Effects

The condition of the reactor at the onset of melting is assumed to be the same as calculated for initiator effects. A two-group, two-dimensional static diffusion theory calculation is done with full melting of the first assembly group. The resulting power distortions and reactivity are assumed to follow predefined functions of the melted fraction. Similar calculations are done for subsequent assembly groups assuming each prior assembly group has fully melted.

Power Level

The reactor-average power level as a function of time is computed using a one-group, point-kinetics model. Delayed neutrons, xenon coefficients, and temperature coefficients are treated as reactor-average effects. Additional reactivity effects are derived from assembly feedback.

Reactor-average temperatures used for temperature feedback are calculated from one-dimensional thermal-hydraulic calculations of temperatures in typical assemblies, the moderator, and the external hydraulic loop. Predefined statistical weights are used to obtain the average of these temperatures.

Assembly State

The current reactor power level is used to update the relative power and state of each assembly group. The relative power is determined by the initial power plus variations caused by the initiator, steaming, and melting effects. The progress of each effect is determined by the nature of the initiator, the time elapsed following the onset of steaming, and energy generated following the onset of melting, respectively.

The relative power and current state of the assembly group are used to predict a new state. The states considered are stable flow, flashing, flow instability, adiabatic heating, and melting. Assemblies can move forward or backward (except from melting) through these states.

Assembly Feedback

Assembly groups contribute to reactivity feedback by steaming or melting. The progress of each of these effects determined in the previous step is combined with the precalculated maximum reactivity of each effect to produce the total reactivity used in the point-kinetics calculation.

Auxiliary Quantities

A variety of auxiliary quantities used to assess the nature and extent of damage to the reactor and to the Confinement System are calculated and reported. Among these are:

- The number of assemblies in each of the "states" (defined previously) at any given time
- An index denoting how close a steaming assembly comes to recovering stable flow (as opposed to proceeding to the melting state)
- The mass of steam generated during each increment of time (ignoring quenching effects)

The last quantity is used as input to the pressure calculation described below.

Pressure Surge

Steam generated by assemblies in the state of flow instability is assumed transmitted immediately to the reactor room, without quenching and with no pressure drops due to passage through any realistic flow paths. This source of steam passes through the radioactivity confinement system (a filtered ventilation system), up the stack to the atmosphere. Pressures at key points along the flow path are calculated as functions of time. The maximum values derived must be less than prescribed limits.

Another type of routine safety analysis predicts the transients in coolant effluent temperatures that would result from hypothetical accidents terminated by automatic safety systems. Analyses of these accidents are used to define limits on normal operating temperatures.

With the safety systems working, some initiating mechanism (power surge or flow reduction) could cause an increase in coolant temperatures until the safety systems shut the reactor down. By calculating the course of the transient, a value for initial temperature can be found such that the maximum temperature achieved does not exceed the saturation temperature. The steps in this analysis are given in Figure 40. The analysis is repeated for a standard series of accidents.

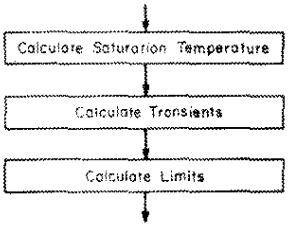


FIG. 40 Thermal-Hydraulic Limits Analysis Processing Steps

Saturation Temperature

The saturation temperature (boiling point) of D_2O is a known function of absolute pressure. The task is to calculate the absolute pressure at all points of interest. These include the minimum pressure points in each coolant channel, in the assembly bottom fittings, and in the reactor effluent pipes. Pressures are calculated using conventional hydraulics equations with parameters derived from the Flow-Zoning Analysis previously described. Where the accident initiating mechanism is a flow reduction, pressures (and corresponding saturation temperatures) are calculated as functions of flow rate.

Transients

The reactor-average power level is computed as a function of time, using a one-group, point-kinetics model. Delayed neutrons and xenon and temperature reactivity effects are treated as reactor-average quantities. Given the reactor power level, assembly coolant temperatures (the maximum for each flow zone) are computed from conventional heat transfer equations. Assembly coolant temperatures are monitored to determine the time at which automatic safety circuits would be tripped. The reactivity transient due to dropping safety rods follows a precalculated course, beginning at a known delay time after the safety circuits are tripped. The principal results derived from this calculation are the maximum/initial assembly ΔT values during the transient.

Limits

A coolant effluent temperature may be expressed as the sum of the inlet temperature and the temperature rise (ΔT) across an assembly (or an individual coolant channel or the entire reactor). This relationship, plus the maximum/initial ΔT ratios from the Transient Calculation, may be used to derive initial effluent temperatures that would just approach saturation temperatures during the transient. This calculation is done as a function of inlet temperature. The operating temperature limits, which are the final objectives of this analysis, are set lower than the derived initial temperatures by empirically established margins.

FILM-BOILING BURNOUT ANALYSIS SUBSYSTEM

Another type of routine safety analysis predicts the extent of fuel damage due to film-boiling burnout under static and transient conditions. These analyses are used to define limits on the heat flux at the fuel-coolant interface. Limits are expressed in terms of a Burnout Safety Factor (BOSF), the ratio of burnout heat flux to actual heat flux. Quantitative analysis of the phenomenon is based on experiments from which the degree of damage (amount of fuel melted) can be correlated with BOSF. The analysis proceeds in the three steps shown in Figure 41.

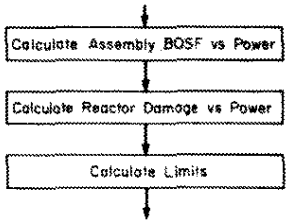


FIG. 41 Film-Boiling Burnout Analysis Processing Steps

Assembly BOSF

A prescribed axial power distribution plus heat and flow splits to each coolant channel are used to calculate BOSF as a function of assembly power, once for each flow zone.

Reactor Damage

For an assumed value of reactor power, the normal power distribution derived from the Charge Design Calculation is used to establish individual assembly powers. The empirical Damage versus BOSF relationship is used with the BOSF versus power relationship from the preceding step to derive the total amount of damage resulting at a given reactor power. The calculated amount of damage is compared with limits, as discussed in the following step and reactor power adjusted until the amount of damage is precisely the limit prescribed.

Limits

BOSF limits are calculated for both static and transient conditions. In either case, reactor power is adjusted until total reactor damage is precisely equal to some prescribed amount. The limit for a given type of assembly becomes the minimum BOSF for that type of assembly in the reactor under the condition,

Calculated Damage = Prescribed Damage

The static limit pertains to normal, constant power operating conditions. Because of the statistical nature of the Damage versus BOSF correlation, zero damage may not be prescribed. Instead, an acceptably low damage value is prescribed.

The transient limit pertains to the extreme conditions reached during the same transients considered in the Thermal-Hydraulic Limits Analyses (terminated by automatic safety systems). In view of the reduced probability of occurrence of such transients, some higher allowable damage value is prescribed. Also, the maximum/initial assembly power ratios noted during the Thermal-Hydraulic Limits Analyses are used in computing damage.

The static and transient BOSF limits are compared. The lower of the two, minus some pad, is used as the operating limit.

JOSHUA System Statistics

DEVELOPMENT EFFORT

The JOSHUA System has been under development since early 1968. The total level of effort in each of these seven years is shown in Figure 42. The largest effort occurred in the 1971-72 period during which nearly 22 people were involved. It is expected that a continuing effort of about 8 people for several more years will be required to refine and extend the Analysis Subsystems.

The development effort by Analysis Subsystem and year is given in Table 2. As expected, the largest effort has gone into Lattice Analysis, Reactor Analysis, and the Operating System. The 30.8 man-years expended on the Operating System can be broken down as follows:

- 65% development
- 20% maintenance
- 15% training and assistance

These percentages are probably reasonable also for the Analysis Subsystems.

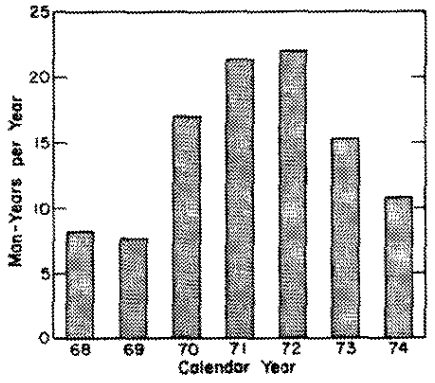


FIG. 42 Level of Effort in JOSHUA Development

TABLE 2

Development Effort (man-years)
Calendar Year

Analysis Subsystem	*1968*	*1969*	*1970*	*1971*	*1972*	*1973*	*1974*	*Total*
Basic Data	1.5	1.5	1.8	0.9	0.3	1.5	-	7.5
Generalized Lattice	3.8	2.7	3.7	2.8	3.7	3.0	2.0	21.7
Correlation Reduction	-	-	-	0.6	1.9	2.0	1.7	6.2
Generalized Reactor	-	0.4	5.0	11.0	7.1	3.5	3.5	30.5
Charge Design	-	-	-	-	1.0	0.3	0.2	1.5
Operating Limits	-	-	-	-	0.7	1.5	1.0	3.2
Confinement Protection	-	-	-	-	1.0	-	-	1.0
Application Total	5.3	4.6	10.5	15.3	15.7	11.8	8.4	71.6
Operating System	2.8	3.1	6.5	6.1	6.3	3.5	2.5	30.8
System Total	8.1	7.7	17.0	21.4	22.0	15.3	10.9	102.4

OVERALL USE OF JOSHUA

The number of people using JOSHUA (excluding those using only the Editor facility) has grown rapidly over the past years (Figure 43). Currently 98 users probably represent 90% of the total number of potential users.

The growth in computing time used by JOSHUA Analysis Subsystems is shown in Figure 44. The JOSHUA workload is measured in percent of total available CPU time and currently runs about 76%. The increased workload in 1971 reflects the first production use of the Lattice Analysis Subsystem.

Terminals were installed in 1970, and their use has grown rapidly since then. The workload submitted from terminals is also shown in Figure 44. Nearly 97% of the JOSHUA workload is submitted from terminals. The user has a choice of submitting jobs from cards or terminals; the 97% workload demonstrates the user preference for terminals.

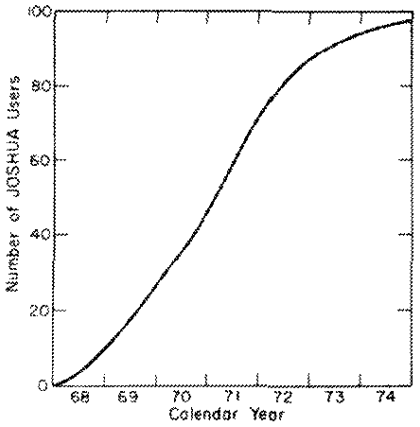


FIG. 43 Number of JOSHUA Users

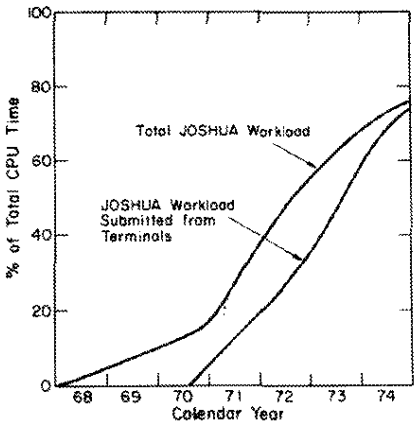


FIG. 44 JOSHUA Workload

SIZE OF THE ANALYSIS SUBSYSTEMS

All Analysis Subsystems were originally written in the IBM FORTRAN IV Language. In some cases Basic Assembly Language equivalents of critical routines were also written. The size of the FORTRAN source decks is given in Table 3. The total number of FORTRAN cards is approximately 3×10^5. The total number of modules is 74, giving an average module size of about 4000 cards.

The current allocated size of the JOSHUA Data Base is given in Table 4. Of the total allocated space of 2×10^8 bytes, 6% is used for directories to the data, and 94% is used for data. Over half of the allocated space is used for temporary Job Data Sets.

TABLE 3

Program Size for Each Analysis Subsystem

Analysis Subsystem	*No. of Modules*	*Total No. of Source Cards (thousands)*
Basic Data	5	40
Generalized Lattice	33	150
Correlation Reduction	6	20
Generalized Reactor	14	50
Charge Design	4	14
Operating Limits	7	13
Confinement Protection	5	18
Total	74	305

TABLE 4

Allocated[a] Data Base
(November 1974)

Category	*Size (megabytes)*	*% of Total*
Total Data Base	199	100
Directory Space	12	6
Data Space	187	94
Standard Data Sets	35	18
User Data Sets	56	28
Job Data Sets	108	54

a. The amount of space in active use varies from 70 to 90% of the allocated space on a day-to-day basis.

USE OF THE ANALYSIS SUBSYSTEMS

The statistics on use of the Analysis Subsystems during November 1974 are given in Table 5. The statistics will vary considerably from month to month, but the Lattice and Reactor Analysis Subsystems always consume the largest number of CPU hours. The high use of the Reactor Analysis Subsystem during November reflects the extensive testing of the kinetics modules during the latter part of 1974.

The number of jobs submitted in November (1861) is fairly typical as is the number of jobs per day (91). Since each job may consist of multiple cases, the number of cases submitted is quite large and typical of a production computing environment. For example, an average of 10 lattices are calculated per job so that a total of about 1920 lattices were calculated during November, or about 90 lattices per day.

Most Analysis Subsystems use dynamic main storage allocation combined with a data roll-in/roll-out strategy determined at execution time. If the user requests the minimum main storage required for a job, a large amount of transfer of data between main storage and disk will occur. Conversely, if the user requests a large amount of main storage, the data transfer will be minimized. In the multiprogramming (MVT) environment at Savannah River with about 1600K of available main storage, the best operation is obtained with at least three concurrent jobs in main storage. Therefore, most jobs are executed in 400-600K as indicated in Table 5.

TABLE 5

Joshua Use Statistics
(November 1974)

Analysis Subsystem	*No. of Jobs*	*CPU Hours*	*% of JOSHUA CPU Hours*	*Avg Core K*	*Avg Jobs/Day*
Basic Data	82	3.2	1.8	350	4
Generalized Lattice	192	20.0	11.5	475	9
Correlation Reduction	92	0.8	0.5	400	5
Generalized Reactor	885	81.8	46.7	550	43
Charge Design	83	5.7	3.3	300	4
Confinement Protection	309	8.3	4.7	275	15
Operating Limits	218	1.4	0.8	300	11
Total	1861	121.2	69.3	440	91

OPERATING SYSTEM STATISTICS

The JOSHUA Operating System has been in production use since 1970. The source decks contain some 40,000 cards, 35% of which are written in FORTRAN, and 65% in Basic Assembly Language.

Extensive statistics on the use and operation of the Operating System have been gathered over the years. These statistics have been vital to the continued improvement of the system's performance. An example of the activity and efficiency of the Operating System is given in Table 6. Average daily statistics for both terminal and batch activities were measured in late 1972. Since all modules are executed in the batch mode, the statistics for the batch are those for the Analysis Subsystem modules. The statistics for the terminals reflect the creation, modification, and inspection of data along with Data Set maintenance.

The ratio of directory I/O requests to data management requests is an indication of the efficiency of the directory search algorithm and the sequence of data management requests. The ratio of 1.32 for terminals reflects the random sequence of requests, whereas the ratio of 0.2 for the batch reflects a combination of random requests followed by sequential requests. The system's overhead is the disk access time required to locate data. The measured overhead of 21 minutes/day is quite satisfactory and provides little incentive for further improvement.

TABLE 6

Average Daily[a] Statistics for Terminals and Batch

	Terminals	*Batch*
Data Management Requests	17,359	148,258
Directory I/O Requests	22,892	29,083
$\frac{\text{Directory I/O Requests}}{\text{Data Management Requests}}$	1.32	0.20
Time for Directory I/O Requests	9 minutes	12 minutes

a. 8-hour terminal day, 24-hour batch day

Future Development

SCOPE

The JOSHUA System currently provides the users at Savannah River with a basic set of computational tools for reactor design and analysis. Future developments fall into two categories:

- Improving the existing capability
- Adding new capabilities

The possible improvements and extensions of the JOSHUA System are discussed in this section. Whether these improvements and extensions are ever implemented will depend on programmatic needs, priorities, manpower limitations, and the availability of outside funding.

NEW APPLICATIONS

Two major new application systems are anticipated:

- Production Planning and Control
- Environmental Transport

The Production Planning and Control System is currently under development. Major components of the System include:

- Accountability inventory management
- Reactor forecasting and production control
- Centralized essential materials control
- $Np\text{-}^{238}Pu$ forecasting
- Reactor cooling basin analysis
- Heavy water inventory control

The Environmental Transport System would incorporate several existing atmospheric, surface water, and ground water transport models of the Savannah River Plant locality. Data obtained from on-line atmospheric monitoring would be used with atmospheric dispersion models to predict the distribution and consequences of potential radioactive releases.

OPERATING SYSTEM

Continued development of the Operating System is required to:

- Incorporate interactive graphics
- Adapt to new hardware
- Provide for future needs

Interactive computer graphics equipment was installed in late 1973. Some software has been developed, but full integration into the existing Terminal Monitor or, alternatively, the development of a separate Graphics Operating System remains to be done. The specific direction of the development depends on the future changes in the Terminal Monitor.

Two major hardware changes are anticipated. The first change is the acquisition of additional terminals. These new terminals will not be of the IBM 2260 type currently in use. The Terminal Monitor will have to be modified to interface with a variety of terminals. A second hardware change is the possible acquisition of a foreground computer, probably one of the IBM 370 series with Virtual Storage capabilities. Conversion of the Operating System to this multiprocessing environment will require extensive efforts to produce an efficient system capable of handling an increasing number of terminals. The Time Sharing Option (TSO) of IBM is being evaluated for possible use in handling these expanded terminal requirements.

Although the facilities provided by the JOSHUA Operating System are adequate for scientific applications, one of the planned new applications (Production Planning and Control) requires data management facilities of both the scientific and business type. Many of the facilities associated with business data management could be profitably used by the existing JOSHUA scientific applications. There is then an incentive to develop a terminal and data management system useful for both business and scientific applications.

BASIC DATA ANALYSIS SUBSYSTEM

No changes are needed except as required by changes in the ENDF/B formats.

GENERALIZED LATTICE ANALYSIS SUBSYSTEM

For the most part, this Subsystem is a mature and stable Subsystem. Several ongoing improvements need to be completed:

- Refinement of the Transmission Probability Method
- Implementation needed to make the various methods for calculating fine-structure flux fully interchangeable
- Optimizing the numerical and coding methods

Several new developments deserve consideration:

- A perturbation theory option for the Transmission Probability Method could reduce the number of lattice calculations required to prepare correlated cross sections for kinetics calculations.
- The present resonance capture treatment is both slow and restrictive. Faster methods with improved accuracy are available.
- The development of transient response matrices is in progress. This development could be applied to the verification of the lattice homogenization procedures for transient calculations.
- Input processing modules could be extended to treat light water reactor lattice geometries.

CORRELATION REDUCTION ANALYSIS SUBSYSTEM

Considerable additional development of this Subsystem is required. Most of the needed effort is in understanding the:

- Selection of lattice states to use in the correlation
- Selection of correlating variables
- Selection of variable transformations
- Required accuracy of the correlation

Interactive graphics has proved to be a powerful tool for gaining this understanding. The integration of interactive graphics in this Subsystem needs to be completed.

GENERALIZED REACTOR ANALYSIS SUBSYSTEM

This subsystem is relatively new and much remains to be done. Further developments in the static analyses include:

- Improve input preparation facilities
- Extend input preparation to include orthogonal geometries
- Reduce depletion calculation time
- Implement spatial synthesis method to reduce static three-dimensional flux calculation time

Needed developments in the kinetic analyses include:

- Improved numerical, coding, and logic methods to reduce computing time of existing modules
- Development of simplified models based on results of the full models
- Add neutronic kinetic capability for orthogonal geometries
- Continued development of kinetic response matrices to verify diffusion theory results
- Development of the data management and interactive graphics capabilities needed to analyze the results of reactor transient calculations

Availability

SCOPE

The JOSHUA System was developed to satisfy the computational requirements of Savannah River but may also be useful to other installations. The JOSHUA Operating System and Generalized Application Subsystems are nonproprietary and unclassified. The possible use of these parts of the JOSHUA System at other installations is discussed in terms of:

- Applicability to other reactor types
- Computer restrictions
- Availability

APPLICABILITY TO OTHER REACTOR TYPES

Some portions of the JOSHUA System are not applicable to other types of reactors. In particular, the Specialized Application Systems and many of the engineering models in the Generalized Application Subsystems are unique to the Savannah River reactors. However, much of the Generalized Application Subsystems is directly applicable or extendable to a wide variety of reactor types.

The Basic Data Analysis Subsystem is directly applicable to all thermal reactors. The Data Sets are applicable to fast reactor analysis, but additional spectrum calculation modules would be needed.

The Generalized Lattice Analysis Subsystem is useful only for thermal reactors. The flux calculation modules can treat most practical two-dimensional geometries using multigroup transport theory. However, the input processing modules treat only those geometries of interest at Savannah River.

The Correlation Reduction Analysis Subsystem is applicable to all types of reactors.

The neutronics part of the Generalized Reactor Analysis Subsystem is quite general, treating all practical reactor geometries. Although the number of energy groups is currently limited to 20, this limit can be easily increased. The basic calculational procedures appear to be applicable to all reactor types. The power flattening algorithm and the input processing modules are at present unique to the Savannah River reactors.

COMPUTER RESTRICTIONS

The JOSHUA System was developed for use on an IBM 360/195. The System has been successfully operated on an IBM 360/65 and 75. It appears that the JOSHUA System can be operated on any large IBM 360 or 370 class computer with at least one megabyte of core and several disk packs. If IBM 2260 equivalent terminals are not available, the System can be operated entirely in the batch mode.

Conversion of the JOSHUA System to operate on non-IBM computers is feasible but may be prohibitively expensive. The cost of conversion to operate on a CDC-7600 has been estimated at 8 to 10 man-years. The reason for the high conversion cost is that many features supported by IBM FORTRAN (but not supported by CDC FORTRAN) have been used extensively in both the application modules and operating system.

AVAILABILITY

Nearly 4,000 pages of detailed documentation of the JOSHUA System have been produced. This documentation is described in Appendix B. Approximately half of the documentation is current and is available from ERDA Technical Information Center, P. O. Box 62, Oak Ridge, Tennessee 37830. The remaining half is being updated and will be available in mid-1975.

Source decks for the JOSHUA Operating System and the Generalized Lattice Analysis Subsystem are in the Argonne Code Center. The remaining portions of the Generalized Application Subsystems which may have use outside of Savannah River will be sent to the Argonne Code Center during 1975.

APPENDIX A - PUBLICATIONS

1969

H. C. Honeck, J. E. Suich, J. C. Jensen, C. E. Bailey, and J. W. Stewart. "JOSHUA - A Reactor Physics Computational System." *Proc. Conf. Effective Use of Computers in the Nuclear Industry, Knoxville, Tenn., April 21-23, 1969.* USAEC Report CONF-690401, pp 324-36 (1969).

J. E. Suich, J. C. Jensen, and H. C. Honeck. "Data Management for Multiaccess Computational Systems." *Proc. Conf. Effective Use of Computers in the Nuclear Industry, Knoxville, Tenn., April 21-23, 1969.* USAEC Report CONF-690401, pp 471-81 (1969).

1970

D. R. Finch. "Cross Sections in the JOSHUA System." *A Review of Multigroup Nuclear Cross Section Preparation Theory, Techniques, and Computer Codes.* D. K. Trubey and Jane Gurney (Compilers). USAEC Report ORNL-RSIC-27, pp 55-60 (1970).

J. C. Jensen, J. E. Suich, F. D. Knight, and H. C. Honeck. "Alphameric Displays in Nuclear Reactor Computations." *Trans. Am. Nucl. Soc. 13(2),* 617 (1970).

M. Becker, D. R. Finch, and J. W. Stewart. "On the Formulation of Time-Dependent Multichannel Synthesis Equations." *Dynamics of Nuclear Systems.* D. L. Hetrick (Editor). University of Arizona Press, Tucson, Ariz. (1970).

1971

D. R. Finch, H. C. Honeck, J. W. Stewart, and W. V. Baxter. "ENDF/B Cross Section Preparation and Testing in the JOSHUA System." *Proc. Conf. New Developments in Reactor Mathematics and Applications, Idaho Falls, Idaho, March 29-31, 1971.* USAEC Report CONF-710302 (Vol. 2), pp 1024-30 (1971).

H. C. Honeck. "Integral Transport Theory Methods in the JOSHUA System." *Trans. Am. Nucl. Soc. 14*(1), 224 (1971).

H. C. Honeck, R. J. Pryor, C. E. Henderson, and W. H. Hightower. "JOSHUA Lattice Physics Subsystem." *Proc. Conf. New Developments in Reactor Mathematics and Applications, Idaho Falls, Idaho, March 29-31, 1971.* USAEC Report CONF-710302(Vol. 2), pp 913-19 (1971).

J. C. Jensen, J. E. Suich, R. L. Boyce, and R. C. Haywood. "The JOSHUA Terminal System." *Proc. Conf. New Developments in Reactor Mathematics and Applications, Idaho Falls, Idaho, March 29-31, 1971.* USAEC Report CONF-710302(Vol. 1), pp 508-15 (1971).

1972

H. L. Dodds, J. W. Stewart, and C. E. Bailey. "Comparison of the Quasistatic and Direct Methods for Two-Dimensional Thermal Reactor Dynamics." *Trans. Am. Nucl. Soc. 15*(2), 786 (1972).

H. C. Honeck and J. W. Stewart. "Simultaneous Line Over-Relaxation (SLOR) in Hexagonal Lattice." *National Topical Meeting on New Developments in Reactor Physics and Shielding, Kiamesha Lake, N. Y., September 12-15, 1972.* USAEC Report CONF-720901, Book 1, pp 253-65 (1972).

P. B. Parks, C. E. Ahlfeld, J. D. Spencer, and J. W. Stewart. "Application of Space-Time Diffusion Theory to Subcritical Pulsed-Neutron Experiments." *Trans. Am. Nucl. Soc. 15*(2), 903 (1972).

P. B. Parks, R. L. Currie, N. P. Baumann, and J. W. Stewart. "Space-Time Kinetics Experiments Emphasizing Delayed-Neutron Holdback." *Trans. Am. Nucl. Soc. 15*(2), 926 (1972).

1973

M. M. Anderson and H. C. Honeck. "An Interface Current Technique for Two-Dimensional Cell Calculations." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich., April 9-11, 1973.* USAEC Report CONF-730414-P1, pp I 53-67 (1973).

M. R. Buckner and H. C. Honeck. "Correlation of Lattice Physics Parameters for Reactor Calculations." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich., April 9-11, 1973.* USAEC Report CONF-730414-P2, pp IX 42-58 (1973).

R. J. Pryor and W. E. Graves. "Response Matrix Method for Treating Reactor Calculations." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich. April 9-11, 1973.* USAEC Report CONF-730414-P2, pp VII 179-186 (1973).

F. D. Knight, J. C. Jensen, and C. E. Parker. "Data Management for Scientific Modular Systems." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich., April 9-11, 1973.* USAEC Report CONF-730414-P1, pp IV 34-45 (1973).

J. W. Stewart. "Recent Developments of Methods for Multidimensional Reactor Dynamics." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich., April 9-11, 1973.* USAEC Report CONF-730414-P2, pp IX 1-14 (1973).

H. C. Honeck. "Current Trends in the Development of Computer Systems for Reactor Analysis." *Proc. Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Mich., April 9-11, 1973.* USAEC Report CONF-730414-P1, pp IV 1-10 (1973).

H. L. Dodds, P. B. Parks, C. E. Bailey, N. P. Baumann, R. L. Currie, and J. W. Stewart. "Space-Dependent Transient Reactor Experiments and Calculations at Savannah River Laboratory." *Trans. Am. Nucl. Soc. 16,* 292 (1973).

J. W. Stewart and M. R. Buckner. "Direct Iterative Calculation of Transient Neutron Flux States for 3-D Hexagonal Lattices." *Trans. Am. Nucl. Soc. 17,* 255 (1973).

R. J. Pryor. "Response Matrix Method and the Cosine-Currents Approximation." *Trans. Am. Nucl. Soc. 17,* 262 (1973).

1974

M. M. Anderson and J. M. Sicilian. "Mean-Chord-Length Approximations in the Interface-Currents Formulation of the Collision-Probabilities Technique." To be published in *Nucl. Sci. Eng.*

R. J. Pryor. "Recent Developments in the Response Matrix Method. To be published in *Proc. ANS Topical Meeting on Reactor Physics Methods, Atlanta, Ga., September 8-11, 1974.*

1975

M. R. Buckner and D. W. Hayes. "Pollutant Transport in Natural Streams." *Proc. ANS Topical Meeting on Computational Methods in Nuclear Engineering, Charleston, S. C., April 15-17, 1975.* USERDA Report CONF-750413, Vol. II, pp IV-123 - IV-142 (1975).

J. M. Sicilian and R. J. Pryor. "TRASCAL, A Two-Dimensional, Multigroup, Response Matrix Kinetics Code." *Proc. ANS Topical Meeting on Computational Methods in Nuclear Engineering, Charleston, S. C., April 15-17, 1975.* USERDA Report CONF-750413, Vol. II, pp VI-103 - VI-122 (1975).

M. M. Anderson, M. R. Buckner, J. H. Carswell, H. L. Dodds, M. V. Gregory, K. R. Routt, J. W. Stewart, and H. C. Honeck. "Three-Dimensional Coupled Neutronics and Engineering Calculations of Savannah River Reactors." *Proc. ANS Topical Meeting on Computational Methods in Nuclear Engineering, Charleston, S. C., April 15-17, 1975.* USERDA Report CONF-750413, Vol. II, pp VI-123 - VI-145 (1975).

M. R. Buckner and J. W. Stewart. "Multidimensional Space-Time Kinetics Studies, Part I - Theoretical." Submitted for publication in *Nucl. Sci. Eng.*

APPENDIX B - DOCUMENTATION

The JOSHUA System has been extensively documented in the series of loose-leaf volumes, DPSTM-500. There are at present 9 volumes in the series, 4 of which are available on microfiche from ERDA Technical Information Center, P. O. Box 62, Oak Ridge, Tennessee 37830. The remaining volumes are being updated and will be released to Technical Information Center in mid-1975. The following table lists the volume numbers, titles, number of single-spaced pages, and status.

Volume	*Title*	*Approximate Pages*	*Released to TIC*
1	General	300	
2	User's Guide	475	x
3	Data Set Specifications	285	x
4	Lattice Physics	640	
5	Basic Data Processing	200	x
6	Static Reactor Physics	330	
7	Transient Reactor Physics	260	
8	Correlation	200	
9	Operating System	1230	x
	Total	3920	