664046

DP-1027

AEC RESEARCH AND DEVELOPMENT REPORT

USER'S MANUAL FOR SRL - HERESY I

HEterogeneous REactor SYstems Code for the IBM 704

D. R. Finch

DUPONT
REG. U. S. PAT. OFF.

Savannah River Laboratory

Aiken, South Carolina

Printed in USA. Price $4.00
Available from the Clearinghouse for Federal Scientific and Technical Information, National Bureau of Standards, U. S. Department of Commerce, Springfield, Virginia

664046

DP-1027

Mathematics and Computers
(TID-4500)

USER'S MANUAL FOR SRL - HERESY I

HEterogeneous REactor SYstems Code for the IBM 704

by

Donald R. Finch

Approved by

J. L. Crandall, Research Manager
Experimental Physics Division

March 1966

E. I. DU PONT DE NEMOURS & COMPANY
SAVANNAH RIVER LABORATORY
AIKEN, SOUTH CAROLINA

CONTRACT AT(07-2)-1 WITH THE
UNITED STATES ATOMIC ENERGY COMMISSION

ABSTRACT

A detailed description of the SRL - HERESY I code (a production version of the HERESY I code developed by C. N. Klahr) is presented in a manner primarily intended for the user of the code. The heterogeneous reactor theory on which the code is based is developed briefly and followed by detailed coding information. Methods are given for solving finite or semi-infinite lattices containing up to 6500 lattice assemblies of up to 50 different types with unique geometrical or flux symmetry properties. Special techniques used with reflected reactor lattices are also described. Several examples of computations are presented as aids in testing and using the code. Detailed descriptions of programming and operating procedures are presented as appendices.

CONTENTS

USER'S MANUAL FOR SRL - HERESY I

HEterogeneous REactor SYstems Code for the IBM 704

INTRODUCTION

In most heterogeneous reactors, essentially all of the neutron births occur in the lattice fuel elements. Most of the neutron deaths take place in the fuel elements, the target elements, or outside of the reactor. Heterogeneous reactor theory treats these facts explicitly by considering the individual lattice elements as neutron sinks and sources of appropriate strength and then summing the interactions between all the individual elements to obtain the over-all reactor equations. This approach is particularly attractive to the reactor operator because the output of the theory is directly in terms of events at the individual lattice elements, as are the monitoring information and most of the power limitations in the actual reactor. Further, the heterogeneous theory is uniquely well equipped to deal with reactors composed of mixtures of lattice elements or with the insertion of a foreign element into a uniform lattice, since each of the lattice assemblies may be treated individually without recourse to homogenization procedures.

Several computer codes have been developed to take advantage of these attractive features of heterogeneous theory in practical reactor calculations. The HERESY I code[1,2], written by C. N. Klahr, was one of the first of the heterogeneous codes. It is a rather direct formulation of the equations developed by Feinberg[3] and Galanin[4]. Although the original code gave good results, it required a very extensive input, necessitated a very careful verification of all input data, and presented only limited output information. A recoding of the original code into the SRL - HERESY I code was undertaken to eliminate these inconveniences and to permit the use of stored data to reduce the code running time in repetitive problems.

SUMMARY

The SRL - HERESY I code performs reactor lattice calculations based on the heterogeneous reactor theory formulated by Feinberg[3]. The code will perform calculations on finite lattices, with or without reflector, or semi-infinite lattices. The lattice is limited to a maximum of 6500 total lattice assemblies and to 50 types of lattice locations possessing unique geometrical or flux symmetry properties. The allowed number of physically differing sorts of lattice assemblies is 48.

The code solves for the thermal neutron absorptions in each lattice assembly normalized to the most highly absorbing assembly in the lattice. Resonance neutron absorptions are computed on the basis of a single resonance energy. The edit of the solved problem produces lattice-averaged values of the resonance escape probability, p; the thermal utilization, f; the average, η; and the leakage-corrected static coefficient of reactivity, k.

Three criticality searches built into the code permit variation of a selected parameter in the calculations until critical requirements are reached. The variations may be made in the thermal absorption or thermal η parameter of a specified group of physically identical lattice assemblies or in axial buckling of the lattice with a cosine axial flux shape.

DISCUSSION

I. SUMMARY OF HERESY I EQUATIONS

This section will present a summary of the equations used in SRL - HERESY I. It is not meant to be a complete treatment of the theory underlying the code, nor of heterogeneous reactor systems in general. Insofar as is possible, reference is made to original papers in which the theory is developed.

A. Equations Solved[2,3,4]

The general approach of heterogeneous reactor theory is to consider the individual reactor lattice elements (referred to as rods in most of the text) as line sources and sinks of neutrons and to sum the interactions between these elements to obtain the over-all reactor equation.

The fundamental equation solved by the SRL - HERESY I Code is

$$\gamma_n i_n = \sum_{m=1}^{N} \left(\frac{\eta_m}{k} F^*_{nm} - f_{nm} \right) i_m \qquad m = 1,2...N \qquad (I.A.1)$$

where

i_n = number of thermal neutron absorptions per unit length per second in n^{th} rod in the lattice

γ_n = ratio of asymptotic neutron flux at the surface of the rod (flux that would be calculated at the rod surface using infinite-medium line-source kernels) to the thermal neutron absorptions per unit length per second in the rod

η_m = number of fission neutrons produced per thermal neutron absorption in the m^{th} rod

F^*_{nm} = fission-to-thermal source kernel (with resonance absorption included). This factor gives the thermal neutron flux at the n^{th} rod due to fissions in m^{th} rod. It is evaluated for an infinitely long line source of fast neutrons in an infinite moderator at a source intensity of 1 fission neutron per second per unit length of the m^{th} rod. This factor includes absorptions in the moderator.

f_{nm} = thermal source kernel. This factor gives the thermal neutron flux at the n^{th} rod due to thermal neutrons emitted at the m^{th} rod. It is evaluated for an infinitely long line source of thermal neutrons in an infinite moderator at a source intensity of 1 thermal neutron per second per unit length of the m^{th} rod.

k = static reactivity coefficient. It corresponds to the effective multiplication constant and is used as a criticality condition.

N = number of rods in the lattice. (Lattice symmetry will enable this number to be reduced later, Section II-C.)

The F^*_{nm} fission-to-thermal source kernel accounts for resonance absorption by subtracting a neutron sink at the resonance absorption energy from the fission-to-thermal source kernel without resonance absorption. If F_{nm} is the fission-to-thermal source kernel with no resonance absorption, then

$$F^*_{nm} = F_{nm} - \sum_{t=1}^{N} F^r_{nt} A_t g_{tm} \qquad \text{(I.A.2)}$$

where

A_t = ratio of resonance absorptions per unit rod length at an equivalent single resonance energy to the asymptotic slowing down density averaged over rod t at this single equivalent energy

g_{tm} = fission to resonance slowing down kernel giving number of neutrons per cubic centimeter per second slowing down to the equivalent resonance energy at rod t due to an infinite line source at rod m having a source strength of 1 fission neutron per second per unit length

F^r_{nt} = thermal neutron flux at rod n due to an infinite line source at rod t with source strength of one neutron per second per unit length emitted at the single equivalent resonance energy

Section II-D gives the details of the age diffusion theory kernels used.

B. Self-Consistent Calculation for γ Values

It is not immediately clear how to compute a value of γ for any rod. C. N. Klahr has made detailed studies of this problem for a variety of lattices[2], and concluded that the best values are provided by the so-called Self-Consistent Method.

The value of γ, in the Self-Consistent Method, is calculated by considering an infinite regular lattice of rods of the same physical kind. Equation I.A.1 then becomes

$$\gamma_n = \frac{\eta_n}{k} \sum_{m=1}^{\infty} F^*_{nm} - \sum_{m=1}^{\infty} f_{nm} \qquad \text{(I.B.1)}$$

where η_n is the eta value for the n^{th} rod kind, and use has been made of the facts that $i_m = i_n$ and $\eta_m = \eta_n$. From a relation shown by Klahr[2] that

$$\sum_{m=1}^{\infty} F^*_{nm} = p \sum_{m=1}^{\infty} F_{nm} \qquad \text{(I.B.2)}$$

where p is the resonance escape probability, and the fact discussed in Section I-C that η_n includes ϵ and fuel rod captures, equation I.B.1 becomes, after setting $k = \eta p f$ and using I.B.2.

$$\gamma_n = \frac{1}{f_n} \sum_{m=1}^{\infty} F_{nm} - \sum_{m=1}^{\infty} f_{nm} \qquad \text{(I.B.3)}$$

where f_n is obtained by a cell calculation or by experiment as

$$f_n = \frac{\text{abs. in all cell components except mod. and coolant}}{\text{abs. in all cell components including mod. and coolant}}$$

Feinberg shows[3] that the first sum is approximately given by

$$\sum_{m=1}^{\infty} F_{nm} \cong \frac{1}{V_n \Sigma_a} \qquad \text{(I.B.4)}$$

where V_n is the cell volume, and Σ_a is the moderator macroscopic absorption cross section. Except for the diagonal elements, the second term of I.B.3 can be summed by use of the Poisson Sum Formula for an infinite regular lattice.

In general the diagonal terms of the f_{nm} sum are very sensitive to the geometrical radius used for evaluation, since in the diffusion theory approximation they are infinite if evaluated at r = 0. To circumvent this difficulty, the physical radius of the rod may be used to yield the following expression (valid in the diffusion theory approximation):

$$f_{nn} = \frac{1}{2\pi D_n} K_o\left(\frac{r_n}{L'}\right) \qquad \text{(I.B.5)}$$

where L' is the moderator diffusion length corrected for axial leakage, and $K_o\left(\frac{r_n}{L'}\right)$ is the modified Bessel function of the second kind.

Using then the results stated in equations I.B.3, I.B.4, and I.B.5 reduces all of equation I.B.3 (N identical equations) to

$$\gamma_n = \frac{1}{V_n \Sigma_a f_n} - \sum_{m=1}^{\infty'} f_{nm} - \frac{1}{2\pi D_n} K_o\left(\frac{r_n}{L'}\right)$$

where the prime on the sum indicates the omission of the n^{th} term. After detailed work on summing over the f_{nm} terms, this takes the following form:

$$\gamma_n = \frac{1}{V_n \Sigma_a} \frac{(1 - f)}{f} - \frac{1}{2\pi D_n} K_o\left(\frac{r_n}{L'}\right) + \frac{1}{4\pi D_n} \left[\ln\left(\frac{L'^2}{V_n}\right) + 2.8741\right] \qquad \text{(I.B.6)}$$

The constant 2.8741 is geometrical in origin and corresponds to the proper value for a hexagonal lattice; however, this constant does not change appreciably for most lattices, so will give reasonably accurate values for all lattice types.

Axial leakage is taken into account in the computation of γ_n by assuming a cosine axial flux shape. The axial buckling then is defined by

$$B_z^2 = \left(\frac{\pi}{H + \lambda}\right)^2 \qquad \text{(I.B.7)}$$

where H is the height of the reactor, and λ is the sum of extrapolation distances at the top and bottom of the reactor. A correction may be applied to the moderator absorption cross section and thermal diffusion area to account for the leakage in the form

$$L'^2 = \frac{L^2}{1 + L^2 B_z^2}$$

$$\Sigma_a' = \Sigma_a(1 + L^2 B_z^2) \qquad \text{(I.B.8)}$$

where Σ_a and L^2 are the uncorrected or axially infinite values of Σ_a' and L'^2.

Fast axial leakage is accounted for by decreasing the reactivity k by a factor $\exp(-\tau B_z^2)$:

$$k_{corr} = k_{heresy}\, e^{-\tau B_z^2} \qquad \text{(I.B.9)}$$

C. Fast Fission Correction

A first order correction may be made to η to take account of fast fission. Basically this amounts to redefining η used in HERESY to be

$$\eta_{heresy} = \frac{1}{f_{heresy}} (\eta f)_{cell} \epsilon$$

where ϵ and $(\eta f)_{cell}$ are obtained separately from cell calculations, and f_{heresy} is the heterogeneous f defined in Section B.

D. Resonance Absorption Parameter

Several different methods have been devised to give reasonable values of A. Klahr[2] suggests one of the following:

1. An approximate method deriving A from resonance escape probability as

$$p = 1 - \frac{A}{V_n} \qquad \text{(I.D.1)}$$

where V_n is the lattice area per rod. This is applicable (in D_2O lattices) only for lattice pitches $\lesssim 8$ inches, for which the resonance flux is approximately flat.

2. A more exact form of equation I.D.1 derived from the self-consistent form is

$$p = 1 - A_n \sum_{m=1}^{\infty} g_{nm} \qquad \text{(I.D.2)}$$

3. If an effective resonance integral is available for the rod in question, A is given by:

$$A = \frac{N_f V_f}{(\xi \Sigma_s)_{cell}} (RI_{eff}) \qquad \text{(I.D.3)}$$

where all terms have their conventional meaning.

E. Method of Solving Equations

Equation I.A.1 is first reduced to a standard eigenvalue problem as follows:

1. The sum on the right side of I.A.1 is broken into two parts, the $\sum_{m=1}^{N} f_{nm} i_m$ being brought to the left side of the equation.

2. The equation is now rewritten in the form:

$$\sum_{m=1}^{N} (f_{nm} + \gamma_m \delta_{nm}) i_m = \frac{1}{k} \sum_{m=1}^{N} \eta_m F^*_{nm} i_m \qquad \text{(I.E.1)}$$

or, in matrix form:

$$AI = \frac{1}{k} BI \qquad \text{(I.E.2)}$$

3. Inverting matrix A and multiplying I.E.2 through by A^{-1} from the left yield

$$I = \frac{1}{k} A^{-1} BI \quad \text{or} \quad CI = kI \qquad \text{(I.E.3)}$$

where

$$C = A^{-1} B$$

4. A standard power iteration is then used to solve for k as the largest eigenvalue of equation I.E.3. The eigenvectors i_n normalized to the largest value are also determined from this same power iteration.

5. Criticality searches enter at this point, if desired. The three types of searches are:

a. Searches on η merely change the B matrix in equation I.E.2, thus reforming matrix C of equation I.E.3, and re-solving the eigenvalue problem is all that is required to re-solve the equation. Convergence is tested externally.

b. Searches on gamma demand that matrix A of equation I.E.2 be re-formed and inverted to form equation I.E.3. Hence matrix inversion and solving the eigenvalue problem are required for each iteration of a criticality search on gamma.

c. Searches over axial buckling alter the kernel matrices of equation I.E.1, hence require regeneration of these matrices from new kernels formed with the current buckling value. Each iteration must therefore, re-solve the entire problem.

II. DESCRIPTION OF CODE

A. General Description - Machine Requirements

SRL - HERESY I was written for an IBM 704 computer having a 32K core storage, an on-line printer, and a minimum of five tape units.

The program itself is in two parts or links, these being connected by means of COMMON STORAGE using the CHAIN subroutine (see Appendix D). Link 1 of the code has 13 primary subprograms, a package of library function subroutines, and a package of FORTRAN II control subroutines. The purpose of this link is to (a) read data; (b) check input data for gross errors; (c) generate appropriate kernels and geometries; (d) compute input parameters by the Self-Consistent Method of Section I-B; and (e) output a summary of all input data and generated quantities.

Link 2 of the code takes the data generated in Link 1 and proceeds to solve the problem by the method discussed in Section I-E. This link contains 15 primary subprograms, a package of library subroutines, and a package of FORTRAN II control subroutines.

Links 1 and 2 have 15,725 words of COMMON STORAGE shared between them. This enables information to be permanently stored in the machine such that both Links 1 and 2 can use it. In general, Link 1 reads and stores, computes, or modifies the information contained in the shared storage portion of COMMON STORAGE, and Link 2 draws from this modified information the necessary data to solve any particular problem. After a particular problem is completed, the numbers that are stored in the shared storage are those that correspond to the solved problem. For example, the initial guess to the eigenvectors for the next problem to be run will be the solution to the problem just completed unless this quantity is re-specified in the input to the next problem. Again, if a criticality search has been called for and completed, the η (or γ or B_z^2) value in the shared storage will be that value corresponding to the converged criticality search just completed. One might utilize this latter feature in a sequence of problems in which the first problem normalizes a particular parameter to correspond to $k_{eff} = 1$, and later problems then use this value of the parameter to determine changes that arise from changing other lattice or rod parameters.

Several operations have been streamlined in the SRL code to enable faster running time than for the original HERESY code. An example occurs in the first stages of computation in Link 2. The matrices corresponding to the symbols $\{F_{nm}\}$, $\{F^r_{nm}\}$, $\{g_{nm}\}$, and $\{f_{nm}\}$

are formed in a subprogram, MATELM, and depend only upon rod coordinates and the kernel used. These matrices are placed in the shared portion of storage and re-used in sequences of problems using the same kernel and geometry. This is of particular significance in problems containing a large number of rod types, since the time required to form these matrices for any problem increases approximately as the square of the dimensionality of the matrices.

B. Geometry Tapes

The re-use of matrix elements in large sets of problems leads to a convenience feature with rod coordinates. A total of 13,000 locations in shared storage are allocated to store the X and Y coordinates of all rods (i.e., 6,500 rods maximum). To enable the matrices mentioned in Section II-A to be placed in the shared storage (a total of 10,000 locations are required), the X and Y coordinates of each rod are written out on binary tape for either temporary or permanent storage as the user specifies. The coordinates are always available for use, when properly specified, if a new kernel requires the matrices to be re-formed. If new coordinates are generated in Link 1, a new binary tape, either temporary or permanent, will be written out as specified by the user.

Temporary geometry tapes are written on, or read from, tape unit 2, while permanent geometry tapes are written on or read from, tape unit 3.

A temporary tape will contain one binary record of 13,000 words of storage, or the entire contents of the X and Y coordinate arrays. If a new geometry is generated by Link 1 the subroutine XYLIB will overwrite the previous record with the new coordinates. No END of FILE mark is placed at the end of this tape at any time. The input variable NFILE must be zero to use this method of geometry storage.

A permanent tape will contain several sets of coordinates, each set occupying two binary records on the tape. The first of these records is called the File Number and is specified by the number NFILE in the input data. The second record is the 13,000 binary words contained in the X and Y coordinate arrays plus a checksum word as an add-and-carry logical check on tape operation. Three indicators are necessary to specify the use of the permanent tape: (1) the number NFILE, which is the file number of the desired geometry to be used in the problem; (2) the indicator NWRITE, which tells if a new geometry is to be written out or not; and (3) KØØR if the initial geometry is to be read from a permanent tape. NFILE may be any number positive or negative (but not zero) having up to five fixed point digits, but

not exceeding 2^{15}. NWRITE is either zero (do not write new coordinates on tape) or 1 (write new coordinates on tape). Any time that tape 3 is read all coordinates taken from the tape will be scaled by the current value of an arbitrary scale factor XCØR.

The following operations may be performed with the permanent tapes:

1. If NFILE is specified, and no new coordinates occupy the X and Y coordinate arrays (NWRITE = 0), the coordinates in the second record corresponding to NFILE in the first record may be read, but writing out the coordinates is redundant, hence is skipped.

2. If NFILE is specified and new coordinates occupy the X and Y coordinate arrays (NWRITE = 1), two cases arise:

a. If NFILE is a file number already on the tape, the new coordinates will be written in place of the old coordinates on the tape.

b. If NFILE is a file number not on the tape, the two record file number and coordinate geometry with checksum will be added at the end of the tape. An END of FILE mark will be placed following the last record on the tape.

Use of the permanent tapes allows often used geometries to be utilized repeatedly for most rapid running of the code. (See Appendix C for operating instructions on generating permanent tapes.)

C. Rod Specifications

The rods that make up any particular lattice may be made of several different materials, and may be placed in the reactor with different degrees of flux symmetry. The following scheme is used in SRL - HERESY I for specifying material properties and symmetry properties.

The term "rod" is used to designate a lattice assembly of any variety, i.e., a rod may be a fuel element, a control rod, a control rod cluster, an instrument thimble, or any other assembly as specified by the user.

The term "rod kind" is used to designate a rod with specific material properties independent of any geometrical or flux symmetry properties in the problem.

The term "rod type" is used to designate rods of the same "rod kind" that have the same geometrical and flux symmetry properties in the problem. Usually several rods in the problem will have equivalent geometrical position and flux symmetry properties and, therefore, be of the same "rod type" by definition. The absorptions in these rods are identical; it is only necessary to solve for the absorption in one rod of each given "type". This property enables the dimensionality of the matrices considered in solving the problem (see Section I-E) to be reduced to the number of different "rod types" that occur in the problem instead of the total number of rods.

In forming any problem, the rods are classified primarily according to their respective "types". Coordinates are produced and classified according to "type", matrices are formed having indices based on the number of "types", and running times are dependent mainly on the number of "rod types". The maximum number of "types" is 50.

Each "rod type" (numbered from 1 to NTYPE) has associated with it several other parameters in the code. The mnemonics NRPRT(I) give the number of rods in the lattice of each "rod type", the mnemonics KIND(I) give the "rod kind" that is associated with each "rod type", and the mnemonics NICPRT(I) supply information to the geometry generators to indicate the number of initial equivalent coordinate sets that are input for a given "rod type".

All rod material parameters are entered by "kind" rather than by "type", and through the mnemonics KIND(I) are internally associated with the correct rod type. The properties for each rod "kind" are specified by three parameters:

η_m = number of fast neutrons per thermal absorption
= fission parameter

γ_m = ϕ asymp./i_m
= thermal absorption parameter

A_m = resonance absorptions per slowing down density
= resonance absorption parameter

Calculations of γ_m by the Self-Consistent Method are performed for a rod "kind" by setting γ for the "kind" to be calculated equal to zero and supplying a value of f (mnemonic THERU(I)) and V_{cell} (mnemonic VCELL(I)) for this rod "kind".

The maximum number of rod kinds is 48 for any problem.

D. Kernel Generators

The SRL - HERESY I code utilizes two kernel generators: one in Link 1 and a shorter version in Link 2. The larger generator in Link 1 examines the input information and calculated information for errors prior to attempting to use the information in solving the problem. The small generator in Link 2 does not contain any error checking routines (data are assumed to be sufficiently checked in Link 1), and is used only for the Buckling Criticality Search.

The kernels formed are age diffusion kernels having slowing down distributions q(r) of the form

$$q(r) = \sum_{i=1}^{3} \frac{B_i}{4\pi\tau_i} e^{-\frac{r^2}{4\tau_i}}$$

where r is the radial distance from the source line; τ_i corresponds to either fission-to-thermal age, or to fission-to-effective resonance energy age; and B_i is a weighting fraction corresponding to the specified τ_i value. This is a triple gaussian sum with a single effective neutron age, τ.

Using the above definition of the slowing down source in the diffusion theory approximation leads to the following kernels[1] (see Chapter I for definitions of symbols).

$$g(r) = \sum_{i=1}^{3} \frac{B_i}{4\pi\tau_i^R} e^{-\frac{r^2}{4\tau_i^R}}$$

$$F(r) = \sum_{i=1}^{3} \frac{B_i}{2\pi D} e^{\frac{\tau_i}{L'^2}} \left[K_o\left(\frac{r}{L'}\right) + \frac{\tau_i}{2L'^2} e^{\frac{r^2}{4\tau_i}} - \frac{1}{2}\left(1 + \frac{r^2}{4L'^2}\right) E_1\left(\frac{r^2}{4\tau_i}\right) \right]$$

$$F^R(r) = \sum_{i=1}^{3} \frac{B_i}{2\pi D} e^{\frac{(\tau_i - \tau_i^R)}{L'^2}} \left[K_o\left(\frac{r}{L'}\right) + \frac{(\tau_i - \tau_i^R)}{2L'^2} e^{\frac{r^2}{4(\tau_i - \tau_i^R)}} - \frac{1}{2}\left(1 + \frac{r^2}{4L'^2}\right) E_1\left(\frac{r^2}{4(\tau_i - \tau_i^R)}\right) \right] \qquad \text{(II.D.1)}$$

In the thermal region, a standard diffusion theory kernel is used of the form

$$f(r) = \frac{1}{2\pi D} K_o\left(\frac{r}{L'}\right) \qquad \text{(II.D.2)}$$

Self-contributions (i.e., diagonal elements of f_{nm}) of any rod i to itself in the thermal region have a special form since

$$r \xrightarrow{\text{Lim}} o \; K_o\left(\frac{r}{L'}\right) \rightarrow \infty$$

This problem is eliminated in the standard heterogeneous fashion of evaluating these kernel elements at the rod radius. This leads to a group of numbers (one for each rod type) designated the rod SMF(I) and defined by

$$SMF(I) = f(R_i) = \frac{1}{2\pi D} K_o\left(\frac{R_i}{L'}\right) \qquad \text{(II.D.3)}$$

where R_i is the radius of the i^{th} rod.

Reference to equation I.B.6 shows that this quantity SMF(I) appears as the second term of the right side of that equation. However, the quantities γ_n and SMF(I) are never used separately, only as the sum of the two. Because of this fact it is more convenient usually to define a new quantity γ_n' as

$$\gamma_n' = \gamma_n + SMF(I)$$

$$\gamma_n' = \frac{1}{V_n \Sigma_a} \frac{(1-f)}{f} + \frac{1}{4\pi D_n}\left(\ln \frac{L'^2}{V_n} + 2.8741\right) \qquad \text{(II.D.4)}$$

This equation now has no dependence on the physical radius of the lattice assembly and is far more convenient to use.

The use of the quantities γ_n' in the code is facilitated in the following manner.

1. If rod radii are provided in the input data, γ_n and SMF(I) are calculated and have their respectively defined meanings.

2. If rod radii are not provided (i.e., are zero) in the input data a signal value of 0.10101 is assigned to each SMF(I) and the γ_n corresponding to this value of SMF(I) is calculated. In this case, γ_n and SMF(I) do not individually have physical meaning, but their sum γ_n' is correct.

Kernels are stored as tabular arrays of maximum size 250 points. The finite height of a reactor is taken into account in the kernel generators through a correction on L^2 and Σ_a. This correction is of the form

$$L^2_{corr} = \frac{L^2}{1 + L^2 B_z^2} \text{ and } \Sigma_a^{corr} = (1 + L^2 B_z^2)\Sigma_a$$

and is the same correction as used in calculating certain terms in the expression for γ in equation I.B.6.

Two restrictions are applied to the kernels to assure that proper Bessel's interpolation may be carried out for points not directly on a kernel point. Such a condition occurs in the subprograms MATELM and USKER where the kernel matrices are formed. The restrictions are:

- The mesh spacing must be less than or equal to one-half of the smallest center-to-center rod spacing in the lattice

$$\text{XMESH} \leqslant \text{smallest value } \frac{|r_i - r_j|}{2}$$

- The kernel shall overlap the edge of the lattice on all sides, by at least two points

$$247 * \text{XMESH} \geqslant \text{largest value } |r_i - r_j|$$

The first restriction arises due to the possibility that in some problems the kernel points used for interpolation might fall as much or more than a full lattice pitch away from the position desired. This does not affect the equations formulated in Chapter I; however, it is physically unsatisfying to know that the interpolation is using points this widely spaced. The value of 1/2 as the smallest center-to-center rod spacing simply assures that the points used for interpolation are at most exactly one lattice pitch away from the rod in question.

The second restriction arises in order to prevent the use of the 251st and 252nd points of a 250-point kernel from being used. Obviously these points would pick up values from other quantities stored at these locations in the memory (see Chapter V, problem 3). Since one point is assigned at position zero, the outer edge of the kernel is at 249 *XMESH cm from the origin point. Now, because Bessel's interpolation uses points m-1, m, m+1, and m+2 for interpolation, m can have a value of 247 at most. Therefore, the smallest kernel mesh that may be used is equal to

$$\text{XMESH} \geqslant \frac{\text{largest center-to-center spacing in lattice}}{247}$$

For infinite lattices (Section G), this requirement is modified to the extent that the largest center-to-center spacing for a rod type is considered to be the distance between the initial coordinates of the "rod type" as input to the infinite lattice generator and the rod that is the greatest distance from it.

In addition to the radially infinite problems for which the above kernels are defined, the code can also treat, in an approximate manner, the problem of a radially finite reactor. This is accomplished by forming kernel matrices that extrapolate to zero at some radius from the center of the lattice.

The technique used employs the method of images. Each rod in the reactor is treated as having an image rod which provides a weighted negative contribution to each kernel element balancing the positive contribution from the real rod. The weighting of the image rod is provided by a factor $\sqrt{\rho_i/\rho_o}$ where ρ_i is the distance of the image rod from the center of the reactor, and ρ_o is the distance of the real rod from the center of the reactor. The image rod is positioned a distance outside the reactor equal to the radial distance; the real rod is situated inside the specified reflector radius (mnemonic RADRFL) which may be loosely defined as the reactor radius.

In practice it has been found that a ratio of the quantity RADRFL to the actual extrapolation radius (R_{extrap}) for a uniform fuel lattice varies over an approximate range

$$0.9 \leqslant \frac{\text{RADRFL}}{R_{extrap}} \leqslant 1.0$$

depending on the lattice. Because of this fact it is recommended that a test problem using a uniform fuel lattice be run to determine the real kernel extrapolation distance before using a particular finite kernel.

Another radius that may be input is the quantity ROEMIN which causes the code to omit calculating image contributions for rods within this radius, since their images are very far away from the lattice.

E. Criticality Searches

The SRL - HERESY I code has three built-in criticality searches that may be selected as options. These enable any specified eigenvalue (corrected for axial buckling) to be searched for as a function of γ, η, or B_z^2.

Searches on γ and η require a "rod kind" to be specified for the search. Searches on B_z^2 require that the SMF(I) all be equal to 0.10101 (Section II-D).

Two edit options are available with all searches. The user may obtain edits for each iteration of the search, or edit only the final iteration following convergence. A short summary of each iteration will be printed out in this latter case.

The γ searches recompute the thermal utilization (f_{heresy}) based on the γ value at convergence. This value is inserted back into the proper storage position for the "rod kind" specified in the search.

F. Finite Lattice Geometry Generators

Three geometry generators incorporated into the code accept a bare minimum of data to produce uniform lattices of rods on hexagonal or square pitches, or patterns of rods on circles about the axis of a cylindricized reactor.

1. Hexagonal Lattices

Hexagonal lattices require for input the X and Y coordinates of each rod with angles in the range $0^\circ \leqslant \theta < 60^\circ$ inclusive. The code then generates the remaining five rotationally symmetric elements around the reactor. Provision is made for more than six rods at a given radius to be of the same rod type. This is accomplished by providing more than one set of initial coordinates for a particular rod type to a limit of three sets of coordinates.

For hexagonal lattices, this latter provision is often used in the outer hexes of the lattice. A second reason for its inclusion is that in studying large lattices it is often convenient to approximate the outer edge of a reactor by making several distinctly different rods equivalent to a single rod type.

2. Square Lattices

Square lattices require for input X and Y coordinates of each rod in quadrant 1 with angles in the range $0^\circ \leqslant \theta \leqslant 45^\circ$ inclusive. The code makes use of the fact that if a rod in this sector has polar coordinates R and θ, then the rods of this type will have polar coordinates of radius R and angles θ, $\pi/2 - \theta$, $\pi/2 + \theta$, $\pi - \theta$, $\pi + \theta$, $3\pi/2 - \theta$, $3\pi/2 + \theta$, and $2\pi - \theta$. Rods at 0° and 45° are not included twice. Provision is made for two rods at different positions (same radius) to be of the same rod type by allowing more than one set of initial coordinates to be input for a particular rod type.

3. Circular Patterns of Rods

This generator allows a pattern of rods all centered on circles about the axis of the reactor to be produced. Three initial coordinates are required in this case, these being the X and Y coordinates of the first rod at a particular radius as one proceeds counterclockwise from the positive X axis, and the angular separation (in degrees) between any two rods at the particular radius. Rods are generated from the initial position (at angle θ) to an angle equal to the initial angle plus 360 degrees (or once around the circle). Single rods may be generated by making the angular separation between rods greater than 360 degrees.

It will be noted that a hexagonal lattice is a special case of this lattice; however, hexagonal lattices are used far more than these circular patterns, hence are included as a special generator.

With the use of these three basic types of lattice patterns, most lattices may be generated quite easily. If, in addition, the permanent library tape option is used the generation of new lattices becomes more and more infrequent.

All three of the generators internally compute the number of rods for each "rod type" and store them. It is therefore unnecessary to enter these data when geometry is being generated.

G. Infinite Lattice Geometry Generator

The infinite lattice geometry generator considers only overlapping patterns of rods on rectangular pitches. However, if a block of rods in a hexagonal or other type of lattice can be found that displays infinite repeatability in rectilinear translation, the infinite lattice generator may be used. The user would then specify each rod in the block as a distinct "rod type" and the X and Y pitch of the rectangular lattice of this type as being the X and Y translational distance to the next rod of that type in the nearest symmetry block.

The information necessary to generate lattices of this type is (a) the X and Y pitches for each rod type (as defined above), (b) the X and Y coordinates of the nearest rod of each type to the origin, and (c) the maximum radius to which rods are to be generated.

As in the finite lattice, the number of rods for each rod type are determined by the generator and stored.

H. Print, Punch, and Plot Options

Along with the geometry generators go these three auxiliary subprograms which will respectively list the rod coordinates for each rod type, punch a deck of cards suitable for direct input back into the code (this is restricted to finite lattices only), and plot to a specified scale (XSCALE and YSCALE values) the lattice geometry. These routines may be used to visually check each coordinate geometry for a particular lattice, and as such are recommended during the first use of any new lattice geometry that may be generated. The punch subroutine allows a seldom used lattice to be stored away on reloadable cards for future use, but is hardly the most convenient method of storage. In addition it requires a large amount of time to punch a deck for a big lattice.

III. INPUT INFORMATION

A. Data Preparation

All data are broken down into three groups and input in the following order:

1. Alphanumeric data (1 card only)
2. Fixed point data
3. Floating point data

For each problem run there must be at least one card of each type of data for input. The three types of data are described separately in the following paragraphs.

1. Alphanumeric Data

These data consist of one card of alphanumeric symbols contained in columns 1-72. If a 1 is placed in column 1, the output from the code will begin on a new page; any other standard carriage control symbol may be placed in column 1 (see IBM 704 FORTRAN Reference Manual, C28-6106, page 45).

2. Fixed Point Data

Fixed point data are input in a fixed card format that enables any number of individual datum up to five to be read from a single card. The card format is schematically drawn below:

Card Columns

1	5	6-12	13-24	25-36	37-48	49-60	61-72
Last card indicator	Number of datum on this card (if blank or zero is assumed to be 5)	Relative location of datum 1 of this card (see following list)	Datum 1	Datum 2	Datum 3	Datum 4	Datum 5

All fixed point input data are stored in a large block of core designed as the LL(225) array. Each datum is assigned a relative location in this array, and a list of these locations follows. The relative location of datum 1 of the card being read is punched in columns 6-12 of the card.

Data 2, 3, 4, and 5 are then stored in sequential relative locations in the LL(225) array. If it is not desirable to input a full card of data (5 datum), it is only necessary to place the number of datum actually appearing on the card to be read in column 5 (a blank or zero will cause 5 datum to be read). For instance, if it is necessary to change one property of one rod type in a list of 50 rod types, the relative location of the datum may be determined from the following list and placed in columns 6-12, a number 1 placed in column 5 of the card, and the new datum placed in the datum 1 field of the card as defined above. All fixed point data are entered to the extreme right of the field defined, blank columns being read as zeros.

The last card of fixed point data must have a 1 punched in column 1 to so indicate it is the last card.

Relative locations for fixed point data are contained in the following list:

Problem Specification Information

1.	LL(1) = INF	Finite - Infinite Indicator
	= 0	Finite lattice.
	= 1	Infinite lattice.
2.	LL(2) = KERNEL	Kernel Generator Indicator
	= 0	Use old kernels.
	= 1	Generate new kernels. (Must be 1 for first problem).
3.	LL(3) = KØØR	Coordinate Indicator
	= 0	Use previous coordinates.
	= 1	Generate new coordinates (Finite or infinite).
	= 2	New coordinates are contained in the floating point input data.
	= 3	Read geometry from permanent tape on unit 3.

Iteration Criteria

4. LL(4) = N1 — Maximum number of iterations for matrix inversion correction (set equal to 20 if left blank).

5. LL(5) = N2 — Maximum number of iterations on eigenvalue iteration (set equal to 200 if left blank).

6. LL(6) = N3 — Maximum number of iterations on criticality searches (set equal to 25 if left blank).

Kernel Information

(Must be included if KERNEL = 1)

7. LL(7) = NOGRID — Number of grid points in kernel ($\leqslant 250$).

Geometry Generator Information

(Must be included if KØØR = 1)

8. LL(9) = LAT — Lattice Type Indicator (Finite lattice only)
   - = 0 Uniform hexagonal lattice.
   - = 1 Uniform square lattice.
   - = 2 Pattern of rods centered on circles about axis of reactor.

9. LL(10)... = NICPRT(J) — Number of sets of initial equivalent coordinates input for each rod type. NTYPE NUMBERS. NICPRT(J) $\leqslant 3$. Include for finite lattices only.

10. LL(60) = KPRNT — Print Option Indicator
    - = 0 Do not list geometry.
    - = 1 List geometry.

11. LL(61) = KPLOT — Plot Option Indicator
    - = 0 Do not plot generated geometry.
    - = 1 Plot generated geometry.

12. LL(62) = KSPIP — Punch Option Indicator (Finite lattices only)
    = 0 — Do not punch geometry deck.
    = 1 — Punch geometry deck.

Rod Information

13. LL(63) = NTYPE — Number of rod types ($\leqslant 50$)

14. LL(64) = NKIND — Number of rod kinds ($\leqslant 48$)

15. LL(65)... = NRPRT(J) — Number of individual rods for each rod type. NTYPE NUMBERS (may be omitted if KØØR = 1; the geometry generators then supply these numbers).

16. LL(115)... = KIND(J) — Rod kind associated with each rod type. NTYPE NUMBERS.

17. LL(165) = LCRIT — Criticality Search Indicator
    = 0 — No criticality search.
    = 1 — Search on ETA.
    = 2 — Search on GAMMA.
    = 3 — Search on axial buckling.

18. LL(166) = KSERCH — Rod KIND to be searched over.

19. LL(167) = NSEPR — Edit Option on Criticality Search
    = 0 — Edit only converged solution.
    = 1 — Edit each iteration of criticality search.

Miscellaneous Information

20. LL(168) = KDAT — Input Data Print Option
    = 0 — Data will be printed.
    = 1 — Omit data print.

21. LL(169) = KTEST — Last Problem Indicator
    = 0 — More problems follow.
    = 1 — Last problem.

22.	LL(224) = NWRITE	New File for Permanent Tape Indicator
	= 0	Do not write new geometry file.
	= 1	Write new geometry file.
23.	LL(225) = NFILE	File number of geometry on designated permanent tape (on unit 3)(if zero uses scratch tape on unit 2).

3. Floating Point Data

Floating point data are also input in a fixed card format that enables any number of individual datum up to five to be read from a single card. The card format is identical to that discussed for fixed point data, the only change made for floating point data is for the data to be read in FORTRAN FIXED DECIMAL FLOATING POINT notation (standard FORTRAN F type conversion). The number 10^{-6} would thus appear as 0.000001 on the input card. A floating point datum may be placed anywhere within the defined fields on the card so long as the decimal point is assigned the correct position in the number. If no decimal point is written, it is automatically assigned a position following the character appearing in the last column of the datum field. Blank columns are read as zeros.

All floating point input data are stored in a large block of core designated as the AA(15,500) array. Each datum is assigned a relative location in this array, and a list of these relative locations appears below. The relative location of datum 1 on the card must be punched into columns 6-12 of the card, and provision for reading less than five datum is identical to that for fixed point data.

The last card of floating point information must have a 1 punched in column 1 to so indicate it is the last.

Relative locations for fixed point data are contained in the following list:

Convergence Criteria

1. AA(1) = CR1 — Convergence criteria on matrix inversion (if left blank is set equal to 0.000001).

2. AA(2) = CR2 — Convergence criteria on eigenvalue iteration (if left blank is set equal to 0.00001).

3. AA(3) = CR3 — Convergence criterion on criticality search (if left blank is set equal to 0.00001).

Scale Factors

4. AA(4) = XCØR — Arbitrary scale factor (cm) applied to all coordinates (finite or infinite lattices).

5. AA(5) = XSCALE — Plotting increment along X axis in geometry plotter (if left blank is set equal to XCOR).

6. AA(6) = YSCALE — Plotting increment along Y axis in geometry plotter (if left blank is set equal to XCOR).

Reflector Information

7. AA(7) = RADRFL — Reflector outer radius (cm) (if zero problem is treated as having an infinite reflector).

8. AA(8) = RØEMIN — Radius (cm) within which the reflector may be treated as infinite.

Criticality Search Information

9. AA(9) = EIGEN — Eigenvalue to be searched for (if left blank will be set equal to 1.0).

Kernel Information

(Must be included if KERNEL = 1)

10. AA(10) = XMESH — Mesh spacing between kernel points (cm) must be such that

$$\frac{\text{largest } |r_i - r_j|}{247} \leqslant \text{XMESH} \leqslant \frac{\text{smallest } |r_i - r_j|}{2}$$

11.	AA(11) = BSQ	Axial buckling ($1/cm^2$)
12.	AA(12) = XLSQ	L^2 of moderator uncorrected for leakage (cm^2)
13.	AA(13) = SIGA	Σ_a moderator (1/cm)
14.	AA(14)... = B(J)	Weighting factors on TAU's and TAUR's for each term of triple Gaussian. (if left blank uses values 0.56, 0.44, 0.0)
15.	AA(17)... = TAU(J)	Fission to thermal age for each term of triple Gaussian (cm^2)(if left blank uses 154.3, 68.3, 0.0).
16.	AA(20)... = TAUR(J)	Fission to resonance energy age for each term of triple Gaussian (cm^2)(if left blank uses 126.5, 40.5, 0.0).

Rod Information

17.	AA(25)... = XLSQ2(J)	L^2 for each rod kind. NKIND VALUES (cm^2), if left blank are set equal to XLSQ.
18.	AA(75)... = SIGC(J)	Σ_a used to compute γ for each kind. NKIND VALUES (if left blank are set equal to Σ_a moderator).
19.	AA(125)... = VCELL(J)	Cell area used to compute γ for each rod kind. NKIND VALUES (cm^2).
20.	AA(175)... = GAMMA(J)	Thermal absorption parameter. NKIND VALUES. (if left blank must have THERU(J) value).
21.	AA(225)... = THERU(J)	Thermal utilization. NKIND VALUES (If GAMMA (J) is zero, computes GAMMA(J) using this value).
22.	AA(275)... = ETA(J)	Number of fast neutrons produced/thermal absorption. NKIND VALUES.
23.	AA(325)... = AR(J)	Resonance absorption parameter. NKIND VALUES.
24.	AA(375)... = RADIUS(J)	Rod radius (cm). NKIND VALUES (If zero assigns SMF(J) = 0.10101).

Finite Geometry Generator Information

(included if KØØR = 1 and INF = 0)

25. AA(425)... = XØ(I,1)
    AA(475)... = XØ(I,2)
    AA(525)... = XØ(I,3)
    Initial X coordinates {1 ⩽ I ⩽ NTYPE} (see Notes 2 and 3 and Section-II-E)

26. AA(575)... = YØ(I,1)
    AA(625)... = YØ(I,2)
    AA(675)... = YØ(I,3)
    Initial Y coordinates {1 ⩽ I ⩽ NTYPE} (see Notes 2 and 3 and Section-II-F)

27. AA(725)... = THETA(I,1)
    AA(775)... = THETA(I,2)
    AA(825)... = THETA(I,3)
    Angular separation in degrees between rods on same radius about axis of reactor. {1 ⩽ I ⩽ NTYPE} (see Note 2)(Not required for square or hexagonal lattices).

Infinite Geometry Generator Information

(Included if KØØR = 1 and INF = 1)

28. AA(875)... = WDX(J)
    X pitch for lattice J. NTYPE VALUES (see Note 3).

29. AA(925)... = WDY(J)
    Y pitch for lattice type J. NTYPE VALUES (see Note 3.)

30. AA(975)... = WX1(J)
    X coordinate of rod nearest origin for lattice J. NTYPE VALUES (see Note 3.)

31. AA(1025)... = WY1(J)
    Y coordinate of rod nearest origin for lattice J. NTYPE VALUES (see Note 3).

32. AA(1075)... = RADII
    Radius beyond which no more rods are considered (cm).

Miscellaneous Information

33. AA(1080)... = VØ(J)
    Initial guess to eigenvectors. NTYPE VALUES (if left blank are set equal to 1.0)

34. AA(2500)... = X(J)
    X coordinates for all rods. (see Notes 1 and 3).

$$\sum_{I=1}^{NTYPE} NRPRT(I) \text{ VALUES}$$

35. AA(9000)... = Y(J) — Y coordinates for all rods. (see Notes 1 and 3).

$$\sum_{I=1}^{NTYPE} NRPRT(I) \text{ VALUES}$$

Note 1

Coordinates are input in sequence starting with rod type 1 and running over all rods of that type. These are followed by rod type 2, 3, 4..., where, for each type, the coordinates of all rods of that type are successively entered.

Note 2

Entering the initial coordinates XØ(I,J), YØ(I,J), and THETA(I,J) for the finite lattice geometry generator must be done in a special order. The matrices for these quantities are 50 by 3 or 50 rows and 3 columns. Data are entered columnwise which implies that for each value of J the index I runs through its full range. Hence using XO as an example the values would be entered as follows:

XØ(1,1)...,XØ(I,1)...,XØ(50,1),XØ(1,2)...,XØ(I,2)...
XØ(50,2),XØ(1,3)...,XØ(I,3),...,XØ(50,3)

Some relative locations associated with these are:

F(425) = XØ(1,1)	F(575) = YØ(1,1)	F(725) = THETA(1,1)
F(475) = XØ(1,2)	F(625) = YØ(1,2)	F(775) = THETA(1,2)
F(525) = XØ(1,3)	F(675) = YØ(1,3)	F(825) = THETA(1,3)

Note 3

All coordinates and pitches contained in items 25, 26, 28, 29, 30, 31, 34, and 35 will be multiplied by the number XCØR contained in 4.

B. Problem Sequencing

The data that are input into the code according to the format described in Section III-A are stored permanently in the memory of the computer and may be re-used at any time. In survey type calculations or where a sequence of different values of one or more parameters are the only changes from one problem to the next, it is only necessary to enter the changes to the immediately previous problem. However, the restriction that each problem has at least one card of each of the three types of data (see Section III-A) is maintained and must be satisfied.

This type of data input, while convenient, has some pitfalls that require care to avoid. The more important of these pitfalls are the following:

1. If geometry is generated in the 1st problem and is re-used in the 2nd problem, the data for the second problem must "turn off" the geometry generator or else the geometry will be regenerated. A similar comment applies to the kernel generator, geometry print, geometry plot, and geometry punch options.

2. If the self-consistent method is to be used to calculate a γ in any problem after the first, the γ value must be set equal to zero in the input data as a signal to perform this calculation.

3. In buckling searches an additional requirement is made that the SMF(I) values be those specified by the code (see Section II-E).

4. At the end of any criticality search the values stored in the memory will be those corresponding to the converged search values. If the original input data are desired, it must be restored in the input for the next problem.

5. When a new kernel is to be generated with different moderator properties from a previous problem, care must be taken to see that Σ_a and L^2 values for each rod kind used in the Self-Consistent calculation are correctly reassigned for the new moderator.

IV. ERROR MESSAGES

Error messages are produced at all levels in the SRL - HERESY I Code. A special subroutine checks the input information for each problem to see that no gross errors have been made that will lead to an error stop at a later time. In each subroutine as it is called, both in Link 1 and in Link 2, certain conditions can arise which will lead to error messages being placed on tape 10 and/or on the on-line printer.

Another class of messages that are printed out give information concerning certain conditions in the program that are not harmful to the successful solution of the problem.

All error messages, information messages, or error conditions displayed in the computer address lights that are built into this code are shown in the following list, subroutine by subroutine, as well as the meaning of the message. The notation in parentheses indicates where the message will be output (i.e. on tape 10 or on-line).

Link 1

A. Main Program 1 (Link 1 executive program)

AN ERROR HAS BEEN DETECTED IN THE INPUT DATA FOR THIS PROBLEM. AN ATTEMPT WILL BE MADE TO CONTINUE WITH SUCCESSIVE PROBLEMS (on-line) - This message appears on the on-line printer following the alphanumeric card for the problem. It means that an error has been detected in the input information and a diagnostic placed on tape 10. Problems that follow will be attempted, but no guarantee is made they will run even if the data are correctly input. Any error detected in main Program 1 will produce this on-line message.

B. Subroutine DATCHK (data checking routine)

1. THE NUMBER OF ROD KINDS IS GREATER THAN THE NUMBER OF ROD TYPES (tape 10) - self-explanatory.

2. THE NUMBER OF ROD TYPES EXCEEDS 50 (tape 10) - self-explanatory.

3. A ROD TYPE HAS ZERO RODS (tape 10) - This can arise from incorrect input data, or errors in inputing data to the geometry generators.

4. A ROD TYPE HAS NO KIND NUMBER ASSIGNED (tape 10) - self-explanatory.

5. THE LATTICE TYPE TO BE GENERATED IS NOT OF A RECOGNIZED TYPE (tape 10) - The mnemonic LAT >2.

6. A ROD KIND IS NOT SPECIFIED FOR THE CRITICALITY SEARCH (tape 10) - Can only occur for criticality searches on γ and η.

7. THE INFORMATION GIVEN FOR THE KERNEL GENERATOR IS INSUFFICIENT (tape 10) - One or more of the three quantities XMESH, XLSQ, or SIGA is $\leqslant 0$.

8. THE INFORMATION GIVEN FOR ROD KIND (I) IS INSUFFICIENT TO COMPUTE A GAMMA (tape 10) - γ_I is zero; hence calculation of a gamma value is called for using the thermal utilization f_I. One of the two quantities THERU(I) or VCELL(I) is $\leqslant 0$; hence calculation of the γ_I value cannot be done.

Following each of the above errors, the following message will be placed on tape 10:

> BECAUSE OF THE ERROR(S) DETECTED ABOVE THIS PROBLEM CANNOT BE CONTINUED. SUCCESSIVE PROBLEMS WILL BE ATTEMPTED, ALTHOUGH THEY MAY BE AFFECTED BY THE ERRORS ABOVE (tape 10).

C. Subroutine KERGEN (kernel generator)

1. THE ARGUMENT OF THE E_1 INTEGRAL IN THE KERNEL GENERATOR EXCEEDS 87.3/THE VALUE OF THE INTEGRAL WILL BE SET EQUAL TO 1.0E-30 AND COMPUTATIONS WILL CONTINUE (tape 10) - An information statement arising due to the fact that the exponential subroutine has a maximum argument of 87.3.

2. THE ARGUMENT OF THE E_1 INTEGRAL IN THE KERNEL GENERATOR IS WITHIN 1.0 E-05 OF ZERO/THE VALUE OF THE INTEGRAL WILL BE SET EQUAL TO 1.0E+30 AND COMPUTATIONS WILL CONTINUE (tape 10) - An information statement concerning a condition that could create an accumulator overflow and lead to a HPR 77(EFM) program stop.

3. A TAU(R) VALUE IS LARGER THAN OR EQUAL TO ITS CORRESPONDING TAU VALUE. A KERNEL CANNOT BE GENERATED UNDER THESE CONDITIONS (tape 10) - An error condition arising from placing the effective resonance energy lower than thermal energy.

4. A MACHINE ERROR HAS OCCURRED IN THE KERNEL GENERATOR - RESTART PROBLEM (tape 10 and on-line) - Arises from a branching error by the machine.

D. Subroutine GEOM (geometry executive routine)

1. THE NUMBER OF RODS EXCEEDS 6500 (tape 10) - Arises due to the fact that 6500 rods is the maximum number available for solving problems.

E. Subroutine INGEOM (infinite geometry generator)

1. THE NUMBER OF RODS EXCEEDS 6500 (tape 10) - Same as above for subroutine GEOM.

2. A MACHINE ERROR HAS OCCURRED IN GENERATING AN INFINITE LATTICE - RESTART PROBLEM (tape 10 and on-line) - Arises from a branching error that occurred.

F. Subroutine HEX and SQUARE (hexagonal and square lattice geometry generators)

1. ERROR IN GEOMETRY GENERATOR - INITIAL COORDINATES ARE IN WRONG SECTOR (tape 10) - Arises when the initial X and Y coordinates for a rod type are not in the sector $0^{\circ} \leqslant \theta \leqslant 45^{\circ}$ for the square lattice generator, or not in the sector $0^{\circ} \leqslant \theta \leqslant 60^{\circ}$ for the hexagonal lattice generator.

Link 2

G. Subroutine USKER (Bessel's interpolation of kernel)

1. ERROR - KERNEL SIZE IS TOO SMALL FOR LATTICE BEING USED (tape 10) - Arises because the required condition that the distance corresponding to the 250th kernel point be at least two mesh intervals larger than the largest center-to-center rod spacing in the lattice is not met.

2. ERROR - KERNEL MESH IS TOO LARGE (tape 10) - Arises because the required condition that the mesh spacing XMESH $\leqslant$ 1/2 times smallest center-to-center rod spacing is not met.

Following either of the above two error conditions, the following will be placed on tape 10 and on the on-line printer respectively.

- THIS ERROR OCCURS IN FORMING THE MATRICES FROM THE KERNEL AND GEOMETRY/AN ATTEMPT WILL BE MADE TO RUN SUCCESSIVE PROBLEMS, BUT IT IS UNLIKELY THEY WILL RUN (tape 10).

- PROBLEM DID NOT RUN TO END (on-line).

H. Subroutine INVERT (matrix inversion)

1. THE SPECIFIED NUMBER OF ITERATIONS ON MATRIX INVERSION HAS BEEN EXCEEDED/NO. ITERATIONS COMPLETED = (N1 + 1) CONVERGENCE CRITERION = (CR1)/MATRIX TO BE INVERTED FOLLOWS (tape 10) - Arises from the number of iterations through the inversion process requiring more than N1 passes. Following this message a listing of each matrix element will appear in the order M(1,1)...M(1,I)...M(1,NTYPE), M(2,1)...M(2,I)... M(2,NTYPE), M(3,1)...M(3,I)...

Double spaces will appear in the listing at each point at which the first index changes. Following this listing N1 is increased by 10 and the iterations will continue. If the number of iterations exceeds N1 + 10 the problem is aborted and conditions are reset to start another problem. If this occurs a line will be placed on the on-line printer which reads PROBLEM DID NOT RUN TO END (on-line).

I. Subroutine EIGEN1 (power iteration to determine maximum eigenvalue)

1. THIS PROBLEM HAS EXCEEDED THE MAXIMUM NUMBER OF ITERATIONS SPECIFIED FOR THE EIGENVALUE ITERATION (N2)/CURRENT ITERATION NUMBER = (N2 + 10M, M = 1,2...), CURRENT EIGENVALUE = (FK2)/

(followed by heading) EIGENVECTOR V2/A2
" "
" "
" " (tape 10)

Arising from the stated conditions, the above heading is shown first at iteration number N2 + 11 and then for every tenth iteration until the iteration number reaches 2*N2. At this time the problem is aborted and conditions are reset for a new problem. If this aborting occurs, a line is printed on the on-line printer which reads PROBLEM DID NOT RUN TO END (on-line). The values listed under the headings of EIGENVECTOR and V2/A2 are respectively the absorptions computed during the current iteration (un-normalized) and the ratio of absorptions to maximum absorption (i.e. normalized eigenvectors).

J. Subroutine CONVER (convergence test for criticality searches)

1. THE MAXIMUM NUMBER OF ITERATIONS HAVE BEEN EXCEEDED IN THE SEARCH WITHOUT CONVERGING. THIS PROBLEM WILL BE SKIPPED OVER AND SUCCESSIVE PROBLEMS RUN (tape 10) - Arises because the iteration counter for

criticality search has exceeded N3. Due to the time involved in some searches no extra allowance is made for this condition. An edit of the current iteration will follow this statement so that the speed of convergence can be estimated and proper alteration of the problem made. If this occurs a line will be put out by the on-line printer which reads PROBLEM DID NOT RUN TO END (on-line); conditions are then reset for more problems to be run.

K. Subroutine KERGEN (kernel generator for Buckling Search)

1. A MACHINE ERROR HAS OCCURRED IN THE SUBROUTINE KERGEN-RESTART PROBLEM FROM THE BEGINNING (on-line) - Arises from a machine branching error that occurred.

L. Errors Leading to Program Stops

The standard BSS or LUCY loader errors are all in force during the running of both links of the CHAIN PROGRAM. In addition, the following program stops may occur in the course of running a problem which may be interrogated by the address lights.

Storage Register Address Lights	Subroutine in Which Stop Occurs	Reason for Stop
HPR 77	LFM - EFM	Accumulator overflow.
HTR 3,4	CHAIN	Program link called for is not on the designated tape unit. (Push start to transfer back to calling program).
HTR 5,4	XYLIB	File number called for is not on the tape mounted on tape 3. (Push start to transfer back to calling program).
HPR 464	XYLIB	Checksum error in reading tape 2. (Push start to ignore this).
HPR 547	XYLIB	Checksum error in reading tape 3. (Push start to ignore this).
HPR 610	XYLIB	Misplaced EOF or misread of tape 3. (Push start to return to calling program).

V. EXAMPLE PROBLEMS

The objective of this chapter is to give several examples of calculations made with the SRL - HERESY I code so that verification of the code's accuracy may be initially or periodically tested, and to give examples of the input and the output to and from the code in a manner designed to clarify questions that may arise.

A. Finite Lattice Problems

Problem 1. Control Rod at Center of Small Fuel Lattice

Because of its rapid running time, this problem is the basic debugging problem for SRL - HERESY I. The lattice is shown in Figure 1. It consists of a control rod situated at the center of a small array of fuel assemblies. The lattice is a square type with a pitch of 10.0 inches (25.4 cm). It has 8-fold symmetry, and hence can be generated by the square lattice generator. The problem was treated without axial leakage ($B_z^2 = 0$), and the values of B(I), TAU(I), and TAUR(I) within the code were used. An infinite moderator was used. GAMMA values were computed by the self-consistent method from thermal utilizations. SMF's were specified internally by setting rod radii to zero.

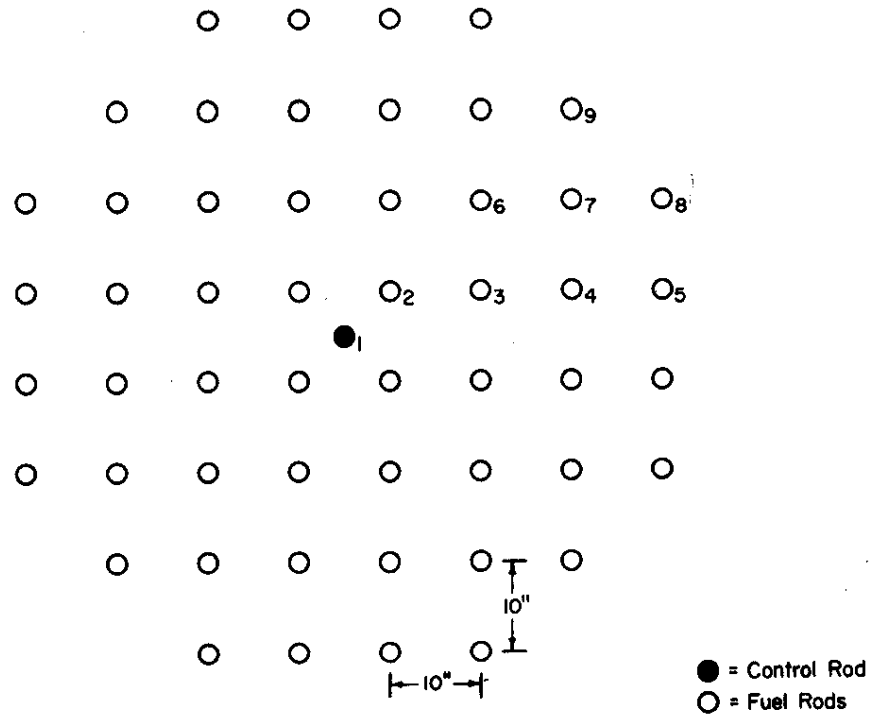


FIG. 1 LATTICE FOR SMALL FINITE TEST PROBLEM

The basic problem was run with no criticality searches, then each of the criticality searches was applied in turn to the basic problem. A listing and plot of geometry were obtained, but are not shown since they were used only to verify the correctness of the geometry as generated. Figure 2 shows a summary, as produced by the code, of the input data for the basic problem. Figure 3 shows the solution to the problem.

FIGURE 2

```
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**BASIC PROBLEM

                    SUMMARY OF HERESY INPUT DATA

                              INPUT OPTIONS
FINITE/INFINITE OPTION =  0     KERNEL OPTION =  1     COORDINATE OPTION =  1     PURE THERMAL KERNEL OPTION =  0
LATTICE TYPE =  1     GEOMETRY PRINT =  C     GEOMETRY PLOT =  0     GEOMETRY PUNCH =  0

                          CONVERGENCE CRITERIA
MATRIX INVERSION ITERATIONS =   20     EIGENVALUE ITERATIONS =   200     CRITICALITY SEARCH ITERATIONS =   25
INVERSION CONVERGENCE = .000001     EIGENVALUE CONVERGENCE = .COCO10     CRITICALITY CONVERGENCE = .000010

                              KERNEL DATA
NUMBER OF GRID POINTS =   250     MESH SIZE =    1.000  CM
B(1)    =   0.58     B(2)    =   0.44     B(3)    =   0.
TAU(1)  = 154.30     TAU(2)  =  68.30     TAU(3)  =   0.
TAUR(1) = 126.50     TAUR(2) =  40.50     TAUR(3) =   0.
B SQUARE = .          L SQUARE = 12626.7000     MODERATOR SIGMA(A) =   0.00006963

REFLECTOR RADIUS =  C.  RHO(MIN) =  0.

                          DATA FOR EACH ROD KIND

ROD KIND    GAMMA      SMF      1./(GAM+SMF)      ETA       RES. COEF.      LSQ      SIGMA(A)      F       CELL VOL.    RADIUS

    1      1.14909   0.10101     0.799934        0.          0.          12626.7    .0000696   0.98544   322.48000    0.
    2      0.74668   0.10101     1.179676      1.3C53C      58.8260      12626.7    .0000696   0.98590   645.16000    0.

                          DATA FOR EACH ROD TYPE

ROD TYPE   ROD KIND   NO. RODS    GAMMA       SMF      1./(GAMMA+SMF)

    1          1          1      1.14909   0.101C1      0.799934
    2          2          4      0.74668   0.101C1      1.179676
    3          2          8      0.74668   0.101C1      1.179676
    4          2          4      0.74668   0.101C1      1.179676
    5          2          8      0.74668   0.101C1      1.179676
    6          2          8      0.74668   0.101C1      1.179676
    7          2          4      0.74668   0.101C1      1.179676
    8          2          8      0.74668   0.10101      1.179676
    9          2          8      0.74668   0.10101      1.179676
```

FIGURE 3

```
NO EIGENVALUE ITERATIONS      EIGENVALUE      PREVIOUS EIGENVALUE
          21                   1.0828493           1.0828522

                              ABSORPTIONS FOR EACH ROD TYPE
ROD TYPE     NO RODS/TYPE     ROD SMF      GAMMA          ETA          A          THERMAL ABSRB     RESONANCE ABSRB
    1              1          0.101C1     1.149093       C.           0.           0.417634          0.
    2              4          0.101C1     0.74668C       1.30530      58.8260      0.943900          0.121672
    3              8          0.101C1     0.74668C       1.30530      58.8260      1.000000          0.125911
    4              4          0.10101     0.74668C       1.30530      58.8260      0.975386          0.122873
    5              8          0.101C1     0.74668C       1.30530      58.8260      0.936836          0.118093
    6              8          0.10101     0.74668C       1.30530      58.8260      0.892689          0.111465
    7              4          0.101C1     0.74668C       1.30530      58.8260      0.827345          0.091342
    8              8          0.10101     0.74668C       1.30530      58.8260      0.815398          0.094986
    9              8          0.101C1     0.74668C       1.30530      58.8260      0.782690          0.085284

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.08284929     AVERAGE ROD ABSORPTION =  0.8834916

AVERAGE ETA =  1.2936580     THERMAL UTILIZATION = 0.98589589     RESONANCE ESCAPE PROBABILITY = 0.89936797

APPROX MODERATOR THERMAL ABSORPTIONS =  0.66987     TOTAL LEAKAGE =  2.8166     NON-LEAKAGE PROBABILITY =  0.9440176

                          AVERAGE ABSORPTIONS FOR EACH ROD KIND
                    ROD KIND   TOTAL RODS   AVG. THERMAL ABS.   AVG. RESONANCE ABS.
                        1           1          0.417634              0.
                        2          52          0.892450              0.108259
```

Using this basic problem as a starting point for each criticality search is next in order. Figure 4 shows the summary of input data produced by the code for a search over the η value of rod kind 2. It was specified that this problem edit each iteration of the search, and these edits are shown in Figures 5, 6, and 7. Figure 8 shows the input data summary for a gamma criticality search and the summary of the search that is provided when each iteration is not edited. The edited final solution is shown in Figure 9. A new thermal utilization is computed in this case; however, if γ is not originally obtained from a self-consistent calculation, the re-computation of the f corresponding to the new γ value is not possible.

FIGURE 4

1 SMALL TEST PROBLEM FOR FINITE LATTICE••SRL-HERESY I••ETA SEARCH

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS

FINITE/INFINITE OPTION = -0 KERNEL OPTION = -0 COORDINATE OPTION = -0 PURE THERMAL KERNEL OPTION = 0

CONVERGENCE CRITERIA

MATRIX INVERSION ITERATIONS = 20 EIGENVALUE ITERATIONS = 200 CRITICALITY SEARCH ITERATIONS = 25
INVERSION CONVERGENCE = .000001 EIGENVALUE CONVERGENCE = .000010 CRITICALITY CONVERGENCE = .000010

KERNEL DATA

NUMBER OF GRID POINTS = 250 MESH SIZE = 1.000 CM
B(1) = 0.56 B(2) = 0.44 B(3) = 0.
TAU(1) = 154.30 TAU(2) = 68.30 TAU(3) = 0.
TAUR(1) = 126.50 TAUR(2) = 40.50 TAUR(3) = 0.
B SQUARE = . L SQUARE = 12626.7000 MODERATOR SIGMA(A) = 0.00006963

REFLECTOR RADIUS = 0. RHO(MIN) = 0.

CRITICALITY SEARCH ON ETA - EIGENVALUE TO BE SEARCHED FOR = 1.0000000
ROD KIND TO BE SEARCHED ON = 2

DATA FOR EACH ROD KIND

ROD KIND	GAMMA	SMF	1./(GAM+SMF)	ETA	RES. COEF.	LSQ	SIGMA(A)	F	CELL VOL.	RADIUS
1	1.14909	0.10101	0.799934	0.	0.	12626.7	.0000696	0.98544	322.48000	0.
2	0.74668	0.10101	1.179676	1.30530	58.8260	12626.7	.0000696	0.98590	645.16000	0.

DATA FOR EACH ROD TYPE

ROD TYPE	ROD KIND	NO. RODS	GAMMA	SMF	1./(GAMMA+SMF)
1	1	1	1.14909	0.10101	0.799934
2	2	4	0.74668	0.10101	1.179676
3	2	8	0.74668	0.10101	1.179676
4	2	4	0.74668	0.10101	1.179676
5	2	8	0.74668	0.10101	1.179676
6	2	8	0.74668	0.10101	1.179676
7	2	4	0.74668	0.10101	1.179676
8	2	8	0.74668	0.10101	1.179676
9	2	8	0.74668	0.10101	1.179676

FIGURE 5

CRITICALITY SEARCH ITERATION NO 0

NO EIGENVALUE ITERATIONS	EIGENVALUE	PREVIOUS EIGENVALUE
15	1.0828390	1.0828367

ABSORPTIONS FOR EACH ROD TYPE

ROD TYPE	NO RODS/TYPE	ROD SHF	GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	1	0.10101	1.149093	0.	0.	0.417642	0.
2	4	0.10101	0.746680	1.30530	58.8260	0.943913	0.121674
3	8	0.10101	0.746680	1.30530	58.8260	1.000000	0.125912
4	4	0.10101	0.746680	1.30530	58.8260	0.975375	0.122872
5	8	0.10101	0.746680	1.30530	58.8260	0.936817	0.118091
6	8	0.10101	0.746680	1.30530	58.8260	0.892664	0.111463
7	4	0.10101	0.746680	1.30530	58.8260	0.827315	0.091339
8	8	0.10101	0.746680	1.30530	58.8260	0.815367	0.094984
9	8	0.10101	0.746680	1.30530	58.8260	0.782659	0.085281

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.08283895 AVERAGE ROD ABSORPTION = 0.8834738

AVERAGE ETA = 1.2936575 THERMAL UTILIZATION = 0.98589589 RESONANCE ESCAPE PROBABILITY = 0.89936788

APPROX MODERATOR THERMAL ABSORPTIONS = 0.66986 TOTAL LEAKAGE = 2.8170 NON-LEAKAGE PROBABILITY = 0.9440090

AVERAGE ABSORPTIONS FOR EACH ROD KIND

ROD KIND	TOTAL RODS	AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	1	0.417642	0.
2	52	0.892432	0.108258

FIGURE 6

CRITICALITY SEARCH ITERATION NO 1

NO EIGENVALUE ITERATIONS	EIGENVALUE	PREVIOUS EIGENVALUE
1	0.9931400	0.9931391

ABSORPTIONS FOR EACH ROD TYPE

ROD TYPE	NO RODS/TYPE	ROD SHF	GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	1	0.10101	1.149093	0.	0.	0.417640	0.
2	4	0.10101	0.746680	1.19717	58.8260	0.943910	0.121673
3	8	0.10101	0.746680	1.19717	58.8260	1.000000	0.125912
4	4	0.10101	0.746680	1.19717	58.8260	0.975378	0.122872
5	8	0.10101	0.746680	1.19717	58.8260	0.936821	0.118092
6	8	0.10101	0.746680	1.19717	58.8260	0.892670	0.111463
7	4	0.10101	0.746680	1.19717	58.8260	0.827322	0.091340
8	8	0.10101	0.746680	1.19717	58.8260	0.815374	0.094985
9	8	0.10101	0.746680	1.19717	58.8260	0.782667	0.085282

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 0.99314001 AVERAGE ROD ABSORPTION = 0.8834781

AVERAGE ETA = 1.1864924 THERMAL UTILIZATION = 0.98589590 RESONANCE ESCAPE PROBABILITY = 0.89936790

APPROX MODERATOR THERMAL ABSORPTIONS = 0.66986 TOTAL LEAKAGE = 2.8169 NON-LEAKAGE PROBABILITY = 0.9440110

AVERAGE ABSORPTIONS FOR EACH ROD KIND

ROD KIND	TOTAL RODS	AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	1	0.417640	0.
2	52	0.892437	0.108258

FIGURE 7

CRITICALITY SEARCH ITERATION NO 2

NO EIGENVALUE ITERATIONS	EIGENVALUE	PREVIOUS EIGENVALUE
0	1.0000008	1.0000000

ABSORPTIONS FOR EACH ROD TYPE

ROD TYPE	NO RODS/TYPE	ROD SHF	GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	1	0.10101	1.149093	0.	0.	0.417640	0.
2	4	0.10101	0.746680	1.20544	58.8260	0.943909	0.121673
3	8	0.10101	0.746680	1.20544	58.8260	1.000000	0.125912
4	4	0.10101	0.746680	1.20544	58.8260	0.975378	0.122872
5	8	0.10101	0.746680	1.20544	58.8260	0.936823	0.118092
6	8	0.10101	0.746680	1.20544	58.8260	0.892671	0.111463
7	4	0.10101	0.746680	1.20544	58.8260	0.827324	0.091340
8	8	0.10101	0.746680	1.20544	58.8260	0.815376	0.094985
9	8	0.10101	0.746680	1.20544	58.8260	0.782669	0.085282

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.00000083 AVERAGE ROD ABSORPTION = 0.8834792

AVERAGE ETA = 1.1946882 THERMAL UTILIZATION = 0.98589590 RESONANCE ESCAPE PROBABILITY = 0.89936791

APPROX MODERATOR THERMAL ABSORPTIONS = 0.66986 TOTAL LEAKAGE = 2.8168 NON-LEAKAGE PROBABILITY = 0.9440117

AVERAGE ABSORPTIONS FOR EACH ROD KIND

ROD KIND	TOTAL RODS	AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	1	0.417640	0.
2	52	0.892438	0.108258

FIGURE 8

1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**GAMMA SEARCH

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS

FINITE/INFINITE OPTION = -0 KERNEL OPTION = -0 COORDINATE OPTION = -0 PURE THERMAL KERNEL OPTION = 0

CONVERGENCE CRITERIA

MATRIX INVERSION ITERATIONS = 20 EIGENVALUE ITERATIONS = 200 CRITICALITY SEARCH ITERATIONS = 25
INVERSION CONVERGENCE = .000001 EIGENVALUE CONVERGENCE = .000010 CRITICALITY CONVERGENCE = .000010

KERNEL DATA

NUMBER OF GRID POINTS = 250 MESH SIZE = 1.000 CM
B(1) = 0.56 B(2) = 0.44 B(3) = 0.
TAU(1) = 154.30 TAU(2) = 68.30 TAU(3) = 0.
TAUR(1) = 126.50 TAUR(2) = 40.50 TAUR(3) = 0.
B SQUARE = . L SQUARE = 12626.7000 MODERATOR SIGMA(A) = 0.00006963

REFLECTOR RADIUS = C. RHO(MIN) = 0.

CRITICALITY SEARCH ON GAMMA - EIGENVALUE TO BE SEARCHED FOR = 1.0000000
ROD KIND TO BE SEARCHED ON = 2

DATA FOR EACH ROD KIND

ROD KIND	GAMMA	SMF	1./(GAM+SMF)	ETA	RES. COEF.	LSQ	SIGMA(A)	F	CELL VOL.	RADIUS
1	1.14909	0.10101	0.799934	0.	0.	12626.7	.0000696	0.98544	322.48000	0.
2	0.74668	0.10101	1.179676	1.30530	58.8260	12626.7	.0000696	0.98590	645.16000	0.

DATA FOR EACH ROD TYPE

ROD TYPE	ROD KIND	NO. RODS	GAMMA	SMF	1./(GAMMA+SMF)
1	1	1	1.14909	0.10101	0.799934
2	2	4	0.74668	0.10101	1.179676
3	2	8	0.74668	0.10101	1.179676
4	2	4	0.74668	0.10101	1.179676
5	2	8	0.74668	0.10101	1.179676
6	2	8	0.74668	0.10101	1.179676
7	2	4	0.74668	0.10101	1.179676
8	2	8	0.74668	0.10101	1.179676
9	2	8	0.74668	0.10101	1.179676

SUMMARY OF CRITICALITY SEARCH
EIGENVALUE BEING SEARCHED FOR = 1.0000000

ITERATION NO	CURRENT EIGENVALUE	PREVIOUS EIGENVALUE	CURRENT GAMMA VALUE	ROD KIND BEING SEARCHED ON
1	1.0947788	1.0828446	0.6818	2
2	0.9079699	1.0947788	1.8940	2
3	1.0335259	0.9079699	1.0307	2
4	1.0096160	1.0335259	1.1785	2
5	0.9985274	1.0096160	1.2495	2
6	1.0000483	0.9985274	1.2396	2
7	1.0000032	1.0000483	1.2399	2

FIGURE 9

NO EIGENVALUE ITERATIONS	EIGENVALUE	PREVIOUS EIGENVALUE
1	1.0000032	1.0000027

ABSORPTIONS FOR EACH ROD TYPE

ROD TYPE	NO RODS/TYPE	ROD SMF	GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	1	0.10101	1.149093	0.	0.	0.936613	0.
2	4	0.10101	1.239935	1.30530	58.8260	0.932519	0.130471
3	8	0.10101	1.239935	1.30530	58.8260	1.000000	0.136275
4	4	0.10101	1.239935	1.30530	58.8260	0.978130	0.133267
5	8	0.10101	1.239935	1.30530	58.8260	0.940163	0.128065
6	8	0.10101	1.239935	1.30530	58.8260	0.893862	0.120481
7	4	0.10101	1.239935	1.30530	58.8260	0.807285	0.096972
8	8	0.10101	1.239935	1.30530	58.8260	0.804716	0.101615
9	8	0.10101	1.239935	1.30530	58.8260	0.761840	0.090280

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.00000319 AVERAGE ROD ABSORPTION = 0.8870377

AVERAGE ETA = 1.2792953 THERMAL UTILIZATION = 0.96522497 RESONANCE ESCAPE PROBABILITY = 0.89929765

APPROX MODERATOR THERMAL ABSORPTIONS = 1.69378 TOTAL LEAKAGE = 5.3800 NON-LEAKAGE PROBABILITY = 0.9005308

AVERAGE ABSORPTIONS FOR EACH ROD KIND

ROD KIND	TOTAL RODS	AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	1	0.936613	0.
2	52	0.886084	0.116473

THE VALUE OF THERMAL UTILIZATION CORRESPONDING TO A GAMMA VALUE OF 1.239935 FOR ROD TYPE 2 IS 0.96482265

The last problem of this set of four problems is that of a buckling search starting from the basic problem. This problem requires a longer running time than the previous three combined. Figure 10 summarizes the input data and search summary, and Figure 11 shows the converged solution.

FIGURE 10

```
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**BUCKLING SEARCH

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS
FINITE/INFINITE OPTION = -0     KERNEL OPTION = -0     COORDINATE OPTION = -0     PURE THERMAL KERNEL OPTION =  0

CONVERGENCE CRITERIA
MATRIX INVERSION ITERATIONS =   20     EIGENVALUE ITERATIONS =   200     CRITICALITY SEARCH ITERATIONS =   25
INVERSION CONVERGENCE = .000001     EIGENVALUE CONVERGENCE = .000010     CRITICALITY CONVERGENCE = .000010

KERNEL DATA
NUMBER OF GRID POINTS =   250     MESH SIZE =    1.000  CM
B(1)   =   0.56     B(2)   =   0.44     B(3)   =   0.
TAU(1) = 154.30     TAU(2) =  68.30     TAU(3) =   0.
TAUR(1) = 126.50    TAUR(2) =  40.50    TAUR(3) =   0.
B SQUARE = .          L SQUARE = 12626.7000     MODERATOR SIGMA(A) =   0.00006963

REFLECTOR RADIUS =   C.  RHO(MIN) =   0.

CRITICALITY SEARCH ON BUCKLING - EIGENVALUE TO BE SEARCHED FOR =  1.0000000

DATA FOR EACH ROD KIND

ROD KIND    GAMMA      SMF      1./(GAM+SMF)      ETA      RES. COEF.      LSQ      SIGMA(A)      F      CELL VOL.    RADIUS
    1      1.14909    C.10101     0.799934       0.          0.         12626.7    .0000696    0.98544   322.48000     0.
    2      0.74668    C.10101     1.179676     1.30530     58.8260      12626.7    .0000696    0.98590   645.16000     0.

DATA FOR EACH ROD TYPE

ROD TYPE   ROD KIND   NO. RODS    GAMMA      SMF      1./(GAMMA+SMF)

    1          1          1      1.14909   0.10101      C.799934
    2          2          4      0.74668   0.10101      1.179676
    3          2          8      0.74668   0.10101      1.179676
    4          2          4      0.74668   0.10101      1.179676
    5          2          8      0.74668   0.10101      1.179676
    6          2          8      0.74668   0.10101      1.179676
    7          2          4      0.74668   0.10101      1.179676
    8          2          8      0.74668   0.10101      1.179676
    9          2          8      0.74668   0.10101      1.179676

SUMMARY OF CRITICALITY SEARCH
EIGENVALUE BEING SEARCHED FOR =  1.0000000
ITERATION NO      CURRENT EIGENVALUE      PREVIOUS EIGENVALUE      CURRENT BUCKLING VALUE
      1               C.9837358               1.C828429               0.00025000
      2               C.9975945               0.9837358               0.00020897
      3               1.0000624               0.9975945               0.00020185
      4               1.0000096               1.0000624               0.00020203
```

FIGURE 11

```
NO EIGENVALUE ITERATIONS      EIGENVALUE      PREVIOUS EIGENVALUE
                  2            1.0238174           1.C238195

ABSORPTIONS FOR EACH ROD TYPE
ROD TYPE     NO RODS/TYPE     ROD SMF      GAMMA          ETA          A          THERMAL ABSRB      RESONANCE ABSRB
    1              1          0.10101     1.034394       C.           0.           0.432026            0.
    2              4          0.10101     0.631981     1.30530      58.8260        0.968079          0.134405
    3              8          0.10101     0.631981     1.30530      58.8260        1.000000          0.135665
    4              4          0.10101     0.631981     1.30530      58.8260        0.950584          0.129016
    5              8          0.10101     0.631981     1.30530      58.8260        0.889412          0.120772
    6              8          0.10101     0.631981     1.30530      58.8260        0.821611          0.110742
    7              4          0.10101     0.631981     1.30530      58.8260        0.706140          0.085673
    8              8          0.10101     0.631981     1.30530      58.8260        0.701128          0.089272
    9              8          0.10101     0.631981     1.30530      58.8260        0.646464          0.077601

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE =  1.00000955     AVERAGE ROD ABSORPTION =  0.8188709

AVERAGE ETA =  1.2923664     THERMAL UTILIZATION = 0.98589543     RESONANCE ESCAPE PROBABILITY = 0.89892645

APPROX MODERATOR THERMAL ABSORPTIONS =  0.62090     TOTAL LEAKAGE =  6.3959     NON-LEAKAGE PROBABILITY =  0.8731395

AVERAGE ABSORPTIONS FOR EACH ROD KIND
ROD KIND   TOTAL RODS   AVG. THERMAL ABS.   AVG. RESONANCE ABS.
    1          1          0.432026              0.
    2         52          0.826310              0.109015
```

A listing of the input cards for these problems is shown in Figure 12.

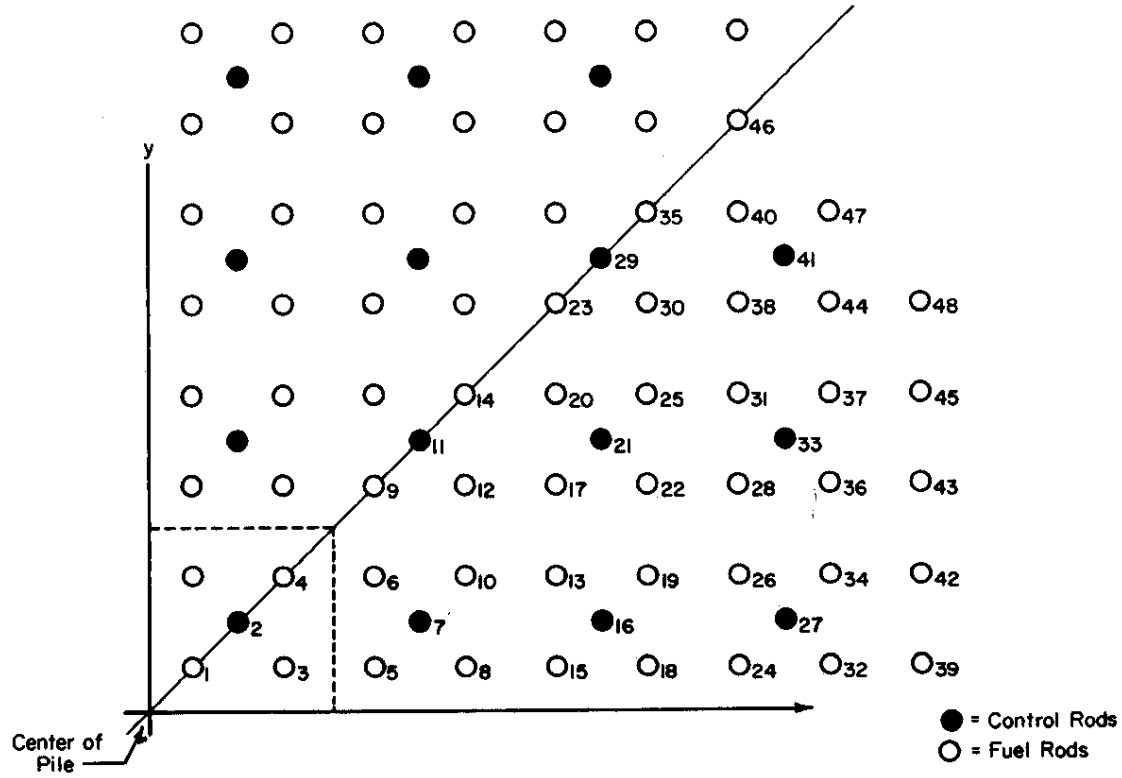
FIGURE 12

```
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**BASIC PROBLEM
     3       1              0              1              1
     5       7            250              0              1              1              1
     5      12              1              1              1              1              1
     2      17              1              1
     2      63              9              2
     5     115              1              2              2              2              2
1    4     120              2              2              2              2
     1       4  12.7
     4      10  1.0           0.0           12626.7       .00006963
     2     125  322.48        645.16
     2     225  .98544        .98590
     2     275  0.0           1.3053
     2     325  0.0           58.826
     5     425  0.0           1.0            3.0           3.0           5.0
     4     430  5.0           5.0            7.0           7.0
     5     575  0.0           1.0            1.0           3.0           1.0
1    4     580  3.0           5.0            1.0           3.0
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL HERESY I** XYLIB TEST
1    2     224              1              1
1    2     175
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**GAMMA SEARCH
     5       1
     2     224                             1
1    2     165              2              2
     1       4  1.0
1    1     276  1.3053
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**ETA SEARCH
1    3     165              1              2              1
     2     175
1    2     225  .98544        .9859
1 SMALL TEST PROBLEM FOR FINITE LATTICE**SRL-HERESY I**BUCKLING SEARCH
1    5     165              3                                                    1
1    1     276  1.3053
C
```

Problem 2. Large Lattice of Fuel and Control Rods

This problem is an extension of the previous problem to that of determining the control rod effects in a large lattice of fuel assemblies. Total rod types in this case is 48, of which 8 are control rods, as shown in Figure 13. Once more 8-fold symmetry is exhibited by the square lattice taken at a 10-inch (25.4-cm) pitch; hence, the internal lattice generator was used. Parameters for input were prepared in exactly the same manner as for Problem 1, and it will be noted that the fuel is the same as for Problem 1.

FIGURE 13 Sector of Large Finite Lattice



EXAMPLE PROBLEM

The input data for Problem 1 is summarized in Figure 14.

FIGURE 14

1 LARGE TEST PROBLEM FOR FINITE LATTICE ** SRL-HERESY I

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS

FINITE/INFINITE OPTION = 0 KERNEL OPTION = 1 COORDINATE OPTION = 1 PURE THERMAL KERNEL OPTION = 0
LATTICE TYPE = 1 GEOMETRY PRINT = 1 GEOMETRY PLOT = 1 GEOMETRY PUNCH = 0

CONVERGENCE CRITERIA

MATRIX INVERSION ITERATIONS = 20 EIGENVALUE ITERATIONS = 200 CRITICALITY SEARCH ITERATIONS = 25
INVERSION CONVERGENCE = .000001 EIGENVALUE CONVERGENCE = .000010 CRITICALITY CONVERGENCE = .000010

KERNEL DATA

NUMBER OF GRID POINTS = 250 MESH SIZE = 2.500 CM
B(1) = 0.56 B(2) = 0.44 B(3) = 0.
TAU(1) = 154.30 TAU(2) = 68.30 TAU(3) = 0.
TAUR(1) = 126.50 TAUR(2) = 40.50 TAUR(3) = 0.
B SQUARE = . L SQUARE = 12626.7000 MODERATOR SIGMA(A) = 0.00006963

REFLECTOR RADIUS = -0. RHO(MIN) = -0.

DATA FOR EACH ROD KIND

ROD KIND	GAMMA	SMF	1./(GAM+SMF)	ETA	RES. COEF.	LSQ	SIGMA(A)	F	CELL VOL.	RADIUS
1	0.74668	0.10101	1.179676	1.30530	58.8260	12626.7	.0000696	0.98590	645.16000	0.
2	4.17239	0.10101	0.234006	0.	0.	12626.7	.0000696	0.92365	322.48000	0.

DATA FOR EACH ROD TYPE

ROD TYPE	ROD KIND	NO. RODS	GAMMA	SMF	1./(GAMMA+SMF)
1	1	4	0.74668	0.10101	1.179676
2	2	4	4.17239	0.10101	0.234006
3	1	8	0.74668	0.10101	1.179676
4	1	4	0.74668	0.10101	1.179676
5	1	8	0.74668	0.10101	1.179676
6	1	8	0.74668	0.10101	1.179676
7	2	8	4.17239	0.10101	0.234006
8	1	8	0.74668	0.10101	1.179676
9	1	4	0.74668	0.10101	1.179676
10	1	8	0.74668	0.10101	1.179676
11	2	8	4.17239	0.10101	0.234006
12	1	8	0.74668	0.10101	1.179676
13	1	8	0.74668	0.10101	1.179676
14	1	8	0.74668	0.10101	1.179676
15	1	8	0.74668	0.10101	1.179676
16	2	8	4.17239	0.10101	0.234006
17	1	8	0.74668	0.10101	1.179676
18	1	8	0.74668	0.10101	1.179676
19	1	8	0.74668	0.10101	1.179676
20	1	8	0.74668	0.10101	1.179676
21	2	8	4.17239	0.10101	0.234006
22	1	8	0.74668	0.10101	1.179676
23	1	4	0.74668	0.10101	1.179676
24	1	8	0.74668	0.10101	1.179676
25	1	8	0.74668	0.10101	1.179676
26	1	8	0.74668	0.10101	1.179676
27	2	8	4.17239	0.10101	0.234006
28	1	8	0.74668	0.10101	1.179676
29	2	4	4.17239	0.10101	0.234006
30	1	8	0.74668	0.10101	1.179676
31	1	8	0.74668	0.10101	1.179676
32	1	8	0.74668	0.10101	1.179676
33	2	8	4.17239	0.10101	0.234006
34	1	8	0.74668	0.10101	1.179676
35	1	4	0.74668	0.10101	1.179676
36	1	8	0.74668	0.10101	1.179676
37	1	8	0.74668	0.10101	1.179676
38	1	8	0.74668	0.10101	1.179676
39	1	8	0.74668	0.10101	1.179676
40	1	8	0.74668	0.10101	1.179676
41	2	8	4.17239	0.10101	0.234006
42	1	8	0.74668	0.10101	1.179676
43	1	8	0.74668	0.10101	1.179676
44	1	8	0.74668	0.10101	1.179676
45	1	8	0.74668	0.10101	1.179676
46	1	4	0.74668	0.10101	1.179676
47	1	8	0.74668	0.10101	1.179676
48	1	8	0.74668	0.10101	1.179676

The solution to Problem 2 is shown in Figure 15. The number of iterations for the eigenvalue power iteration have markedly increased over the 9 rod types of Problem 1. This, however, is expected due simply to the increase of rod types. In fact, a simple ratio of rod types shows:

$$\frac{\text{No. types}_2}{\text{No. types}_1} = \frac{48}{9} = 5.3333 \cong \frac{\text{No. iterations}_2}{\text{No. iterations}_1} = \frac{101}{21} = 4.8095$$

from which the time increase is evidently fairly linear with number of rod types. Not shown by the output is the increase in time required to form the matrix elements. This should be proportional to the number of matrix elements formed times the total number of rods in the lattice:

$$\frac{(\text{No. elements 2})(\text{No. rods 2})}{(\text{No. elements 1})(\text{No. rods 1})} = \frac{(48)^2(352)}{(9)^2(53)} = 188.9$$

FIGURE 15

NO EIGENVALUE ITERATIONS 101 EIGENVALUE 1.1031582 PREVIOUS EIGENVALUE 1.1031590

ROD TYPE	NO RODS/TYPE	ROD SMF	ABSORPTIONS FOR EACH ROD TYPE GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	4	0.10101	0.746680	1.30530	58.8260	1.000000	0.124763
2	4	0.10101	4.172394	0.	0.	0.097868	0.
3	8	0.10101	0.746680	1.30530	58.8260	0.980397	0.122319
4	4	0.10101	0.746680	1.30530	58.8260	0.957996	0.119513
5	8	0.10101	0.746680	1.30530	58.8260	0.946001	0.118022
6	8	0.10101	0.746680	1.30530	58.8260	0.910903	0.116074
7	8	0.10101	4.172394	0.	0.	0.090236	0.
8	8	0.10101	0.746680	1.30530	58.8260	0.894804	0.113923
9	4	0.10101	0.746680	1.30530	58.8260	0.860294	0.110252
10	8	0.10101	0.746680	1.30530	58.8260	0.777796	0.123812
11	8	0.10101	4.172394	0.	0.	0.081147	0.
12	8	0.10101	0.746680	1.30530	58.8260	0.770296	0.122271
13	8	0.10101	0.746680	1.30530	58.8260	0.764106	0.118380
14	8	0.10101	0.746680	1.30530	58.8260	0.824550	0.111798
15	8	0.10101	0.746680	1.30530	58.8260	0.756574	0.122955
16	8	0.10101	4.172394	0.	0.	0.077800	0.
17	8	0.10101	0.746680	1.30530	58.8260	0.867611	0.102111
18	8	0.10101	0.746680	1.30530	58.8260	0.829320	0.105578
19	8	0.10101	0.746680	1.30530	58.8260	0.840363	0.101647
20	8	0.10101	0.746680	1.30530	58.8260	0.841596	0.101997
21	8	0.10101	4.172394	0.	0.	0.082167	0.
22	8	0.10101	0.746680	1.30530	58.8260	0.819485	0.101046
23	4	0.10101	0.746680	1.30530	58.8260	0.803758	0.099230
24	8	0.10101	0.746680	1.30530	58.8260	0.792883	0.098896
25	8	0.10101	0.746680	1.30530	58.8260	0.791338	0.098810
26	8	0.10101	0.746680	1.30530	58.8260	0.784054	0.097838
27	8	0.10101	4.172394	0.	0.	0.075375	0.
28	8	0.10101	0.746680	1.30530	58.8260	0.765694	0.095521
29	4	0.10101	4.172394	0.	0.	0.075121	0.
30	8	0.10101	0.746680	1.30530	58.8260	0.753404	0.094021
31	8	0.10101	0.746680	1.30530	58.8260	0.734786	0.091678
32	8	0.10101	0.746680	1.30530	58.8260	0.725331	0.090617
33	8	0.10101	4.172394	0.	0.	0.071332	0.
34	8	0.10101	0.746680	1.30530	58.8260	0.715163	0.089354
35	4	0.10101	0.746680	1.30530	58.8260	0.700009	0.087429
36	8	0.10101	0.746680	1.30530	58.8260	0.694434	0.086751
37	8	0.10101	0.746680	1.30530	58.8260	0.660562	0.082562
38	8	0.10101	0.746680	1.30530	58.8260	0.691595	0.086294
39	8	0.10101	0.746680	1.30530	58.8260	0.659965	0.075347
40	8	0.10101	0.746680	1.30530	58.8260	0.635764	0.078599
41	8	0.10101	4.172394	0.	0.	0.062264	0.
42	8	0.10101	0.746680	1.30530	58.8260	0.649481	0.074159
43	8	0.10101	0.746680	1.30530	58.8260	0.627870	0.071709
44	8	0.10101	0.746680	1.30530	58.8260	0.615239	0.076105
45	8	0.10101	0.746680	1.30530	58.8260	0.593526	0.067842
46	4	0.10101	0.746680	1.30530	58.8260	0.598279	0.064728
47	8	0.10101	0.746680	1.30530	58.8260	0.578067	0.062829
48	8	0.10101	0.746680	1.30530	58.8260	0.556709	0.059473

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.10315818 AVERAGE ROD ABSORPTION = 0.6349238

AVERAGE ETA = 1.2760129 THERMAL UTILIZATION = 0.98441141 RESONANCE ESCAPE PROBABILITY = 0.89282393

APPROX MODERATOR THERMAL ABSORPTIONS = 3.53911 TOTAL LEAKAGE = 3.7739 NON-LEAKAGE PROBABILITY = 0.9836492

AVERAGE ABSORPTIONS FOR EACH ROD KIND

ROD KIND	TOTAL RODS	AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	288	0.758607	0.096203
2	64	0.078352	0.

This problem is a reasonable example of the use of SRL - HERESY I in practical problems. The tendency to approach nearly 50 rod types in most problems has been observed in practice, and arises from attempting to approximate real lattices to as fine a detail as possible.

Data cards for this problem are shown in Figure 16.

FIGURE 16

```
1 LARGE TEST PROBLEM FOR FINITE LATTICE ** SRL-HERESY I
          1          0          1          1          0          0
          7        250          0          1          1          1
         12          1          1          1          1          1
         17          1          1          1          1          1
         22          1          1          1          1          1
         27          1          1          1          1          1
         32          1          1          1          1          1
         37          1          1          1          1          1
         42          1          1          1          1          1
         47          1          1          1          1          1
         52          1          1          1          1          1
         57          1                     1          1          1
         63         48          2
        115          1          2          1          1          1
        120          1          2          1          1          1
        125          2          1          1          1          1
        130          2          1          1          1          1
        135          2          1          1          1          1
        140          1          2          1          2          1
        145          1          1          2          1          1
        150          1          1          1          1          1
        155          2          1          1          1          1
1       160          1          1          1
          4 12.7
         10 2.5                          12626.7    .00006963
        125 645.16        322.48
        225 .9859         .92365
        275 1.3053        0.0
        325 58.826        0.0
        425 1.0           2.0          3.0        3.0        5.0
        430 5.0           6.0          7.0        5.0        7.0
        435 6.0           7.0            9.0      7.0        9.0
        440 10.0          9.0          11.0       11.0       9.0
        445 10.0          11.0         9.0        13.0       11.0
        450 13.0          14.0         13.0       10.0       11.0
        455 13.0            15.0       14.0       15.0       11.0
        460 15.0          15.0         13.0       17.0       13.0
        465 14.0          17.0         17.0       15.0       17.0
        470 13.0          15.0         17.0
        575 1.0           2.0          1.0        3.0        1.0
        580 3.0           2.0          1.0        5.0        3.0
        585 5.0           4.0          2.0        6.0        1.0
        590 2.0           5.0          1.0        3.0        7.0
        595 6.0           5.0          9.0        1.0        7.0
        600 3.0           2.0          5.0        10.0       9.0
        605 7.0           1.0          6.0        3.0        11.0
        610 5.0           7.0          9.0        1.0        11.0
        615 10.0          3.0          5.0        9.0        7.0
1       620 13.0          11.0         9.0
C
```

B. Infinite Lattice Problems

Problem 3. Fundamental Methods Associates, Inc. Test Problem II

As the first example of the infinite lattice problem, the test problem formulated by Fundamental Methods Associates, Inc. described in Reference 1, p 59, was run on the SRL - HERESY I code. Input data are summarized in Figure 17 by the code, and reference to the original work cited above shows its source. (Note the change of the maximum rod radius to 260.0 cm rather than 280.0 cm as given by Fundamental Methods Associates.)

Results for this problem are shown in Figure 18 and are not the same as those quoted by Fundamental Methods Associates, Inc. Using their original version of HERESY I produces the results shown in Figure 18 for a maximum outer radius of 260.0 cm, but SRL - HERESY I cannot produce Fundamental Methods Associates' HERESY I results for a maximum outer radius of 280.0 cm. This problem arises because the original HERESY I code uses the 61st kernel point (with a maximum kernel size of 60 points) in its Bessel's interpolation of the kernel. Reducing the maximum outer radius of rods considered to 260.0 cm avoids this problem and produces agreement between the two codes.

Input for this problem is shown in Figure 19.

FIGURE 17

1 FUNDAMENTAL METHODS ASSOCIATES, INC. TEST PROBLEM II ** SRL-HERESY I

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS

FINITE/INFINITE OPTION = 1 KERNEL OPTION = 1 COORDINATE OPTION = 1 PURE THERMAL KERNEL OPTION = 0
LATTICE TYPE = -0 GEOMETRY PRINT = 1 GEOMETRY PLOT = 1 GEOMETRY PUNCH = 0

CONVERGENCE CRITERIA

MATRIX INVERSION ITERATIONS = 20 EIGENVALUE ITERATIONS = 200 CRITICALITY SEARCH ITERATIONS = 25
INVERSION CONVERGENCE = .000001 EIGENVALUE CONVERGENCE = .000010 CRITICALITY CONVERGENCE = .000010

KERNEL DATA

NUMBER OF GRID POINTS = 60 MESH SIZE = 5.000 CM
B(1) = 1.00 B(2) = 0.00 B(3) = 0.
TAU(1) = 350.00 TAU(2) = 68.30 TAU(3) = 0.
TAUR(1) = 262.50 TAUR(2) = 40.50 TAUR(3) = 0.
B SQUARE = . L SQUARE = 2800.0000 MODERATOR SIGMA(A) = 0.00040000

REFLECTOR RADIUS = -0. RHO(MIN) = -0.

RADIUS BEYOND WHICH NO MORE RODS ARE CONSIDERED = 260.000

DATA FOR EACH ROD KIND

ROD KIND	GAMMA	SMF	1./(GAM+SMF)	ETA	RES. COEF.	LSQ	SIGMA(A)	F	CELL VOL.	RADIUS
1	0.25500	0.48216	1.356558	1.34000	51.0000	2800.0	.0004000	0.	0.	2.0000
2	0.17800	0.48216	1.514784	1.60500	51.0000	2800.0	.0004000	0.	0.	2.0000

DATA FOR EACH ROD TYPE

ROD TYPE	ROD KIND	NO. RODS	GAMMA	SMF	1./(GAMMA+SMF)
1	1	127	0.25500	0.48216	1.356558
2	1	127	0.25500	0.48216	1.356558
3	1	127	0.25500	0.48216	1.356558
4	1	127	0.25500	0.48216	1.356558
5	2	127	0.17800	0.48216	1.514784

FIGURE 18

NO EIGENVALUE ITERATIONS = 3 EIGENVALUE = 1.1125614 PREVIOUS EIGENVALUE = 1.1125614

ROD TYPE	NO RODS/TYPE	ROD SMF	ABSORPTIONS FOR EACH ROD TYPE GAMMA	ETA	A	THERMAL ABSRB	RESONANCE ABSRB
1	127	0.48216	0.255000	1.34000	51.0000	0.962513	0.193800
2	127	0.48216	0.255000	1.34000	51.0000	0.962513	0.193800
3	127	0.48216	0.255000	1.34000	51.0000	0.962513	0.193800
4	127	0.48216	0.255000	1.34000	51.0000	0.962512	0.193800
5	127	0.48216	0.178000	1.60500	51.0000	1.000000	0.194092

A THERMAL UTILIZATION VALUE FOR A ROD KIND IS ZERO. MODERATOR THERMAL ABSORPTIONS CANNOT BE SOLVED FOR.
THE MODERATOR ABSORPTIONS WILL BE LUMPED INTO THE TOTAL LEAKAGE

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.11256137 AVERAGE ROD ABSORPTION = 0.9700101

AVERAGE ETA = 1.3946386 THERMAL UTILIZATION = 1.00000000 RESONANCE ESCAPE PROBABILITY = 0.84056976

APPROX MODERATOR THERMAL ABSORPTIONS = 0. TOTAL LEAKAGE = 33.0686 NON-LEAKAGE PROBABILITY = 0.9490488

INFINITE LATTICE MATERIAL BUCKLING = 0.00030475

ROD KIND	TOTAL RODS	AVERAGE ABSORPTIONS FOR EACH ROD KIND AVG. THERMAL ABS.	AVG. RESONANCE ABS.
1	508	0.962513	0.193800
2	127	1.000000	0.194092

FIGURE 19

```
1 FUNDAMENTAL METHODS ASSOCIATES, INC. TEST PROBLEM II ** SRL-HERESY I
          1          1          1          1
          7         60
         60          1          1          0          5          2
        115          1          1          1          1          2
1       165                                                      1
          4 20.0          10.0       10.0
         10 5.0           0.0        2800.0     .0004      1.0
         15.00000000001   0.0        350.0      0.0        0.0
         20 262.5         0.0        0.0
        175 .255          .178
        275 1.34          1,605
        325 51.0          51.0
        375 2.0           2.0
        875 2.0           2.0        2.0        2.0        2.0
        925 2.0           2.0        2.0        2.0        2.0
        975 0.0           1.0        0.0        1.0        0.5
       1025 0.0           0.0        1.0        1.0        0.5
1      1075 260.0
C
```

Problem 4. Large Infinite Lattice Test Problem

This problem is the extension of Problem 2 to as large a radius as possible without exceeding a maximum of 6500 rods as specified by the code. The problem contains a total of 6355 rods or 1271 rods of each of five rod types. Figure 13 shows the infinitely repeating pattern of five rods within the dashed lines near the center of the lattice. Figure 20 shows a summary of the input data, and Figure 21 shows the results of the calculations.

Comparing "rod type" 2 of the Finite Lattice Problem 2 and "rod type" 5 of the infinite lattice problem indicates about 0.1% difference between them. This is reasonable agreement for the two problems since the finite lattice considered is large enough to approach being a semi-infinite lattice near its center, and the effects of the infinite moderator are clearly small near the center.

The input cards for this problem are shown in Figure 22.

FIGURE 20

1 LARGE TEST PROBLEM FOR INFINITE LATTICE ** SRL-HERESY I

SUMMARY OF HERESY INPUT DATA

INPUT OPTIONS

FINITE/INFINITE OPTION = 1 KERNEL OPTION = 1 COORDINATE OPTION = 1 PURE THERMAL KERNEL OPTION = 0
LATTICE TYPE = -0 GEOMETRY PRINT = 1 GEOMETRY PLOT = 1 GEOMETRY PUNCH = 0

CONVERGENCE CRITERIA

MATRIX INVERSION ITERATIONS = 20 EIGENVALUE ITERATIONS = 200 CRITICALITY SEARCH ITERATIONS = 25
INVERSION CONVERGENCE = .000001 EIGENVALUE CONVERGENCE = .000010 CRITICALITY CONVERGENCE = .000010

KERNEL DATA

NUMBER OF GRID POINTS = 250 MESH SIZE = 4.500 CM
B(1) = 0.56 B(2) = 0.44 B(3) = 0.
TAU(1) = 154.30 TAU(2) = 68.30 TAU(3) = 0.
TAUR(1) = 126.50 TAUR(2) = 40.50 TAUR(3) = 0.
B SQUARE = . L SQUARE = 12626.7000 MODERATOR SIGMA(A) = 0.00006963

REFLECTOR RADIUS = -0. RHO(MIN) = -0.

RADIUS BEYOND WHICH NO MORE RODS ARE CONSIDERED = 1025.000

DATA FOR EACH ROD KIND

ROD KIND	GAMMA	SMF	1./(GAM+SMF)	ETA	RES. COEF.	LSQ	SIGMA(A)	F	CELL VOL.	RADIUS
1	0.74668	0.10101	1.179676	1.30530	58.8260	12626.7	.0000696	0.98590	645.16000	0.
2	4.17239	0.10101	0.234006	0.	0.	12626.7	.0000696	0.92365	322.48000	0.

DATA FOR EACH ROD TYPE

ROD TYPE	ROD KIND	NO. RODS	GAMMA	SMF	1./(GAMMA+SMF)
1	1	1271	0.74668	0.10101	1.179676
2	1	1271	0.74668	0.10101	1.179676
3	1	1271	0.74668	0.10101	1.179676
4	1	1271	0.74668	0.10101	1.179676
5	2	1271	4.17239	0.10101	0.234006

FIGURE 21

```
NO EIGENVALUE ITERATIONS      EIGENVALUE      PREVIOUS EIGENVALUE
                  1           1.1232000          1.1232004

                                          ABSORPTIONS FOR EACH ROD TYPE
ROD TYPE     NO RODS/TYPE     ROD SHF     GAMMA          ETA          A          THERMAL ABSRB     RESONANCE ABSRB
    1            1271         0.10101     0.746680     1.30530     58.8260        0.999999          0.122910
    2            1271         0.10101     0.746680     1.30530     58.8260        1.000000          0.122910
    3            1271         0.10101     0.746680     1.30530     58.8260        1.000000          0.122910
    4            1271         0.10101     0.746680     1.30530     58.8260        0.999999          0.122910
    5            1271         0.10101     4.172394     0.          0.             0.099292          0.

EIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = 1.12320000     AVERAGE ROD ABSORPTION = 0.8198580

AVERAGE ETA = 1.2736834     THERMAL UTILIZATION = 0.98429321     RESONANCE ESCAPE PROBABILITY = 0.89423694

APPROX MODERATOR THERMAL ABSORPTIONS = 83.14140     TOTAL LEAKAGE = -9.9669     NON-LEAKAGE PROBABILITY = 1.0018865

INFINITE LATTICE MATERIAL BUCKLING = 0.00037757

                                AVERAGE ABSORPTIONS FOR EACH ROD KIND
                    ROD KIND   TOTAL RODS   AVG. THERMAL ABS.   AVG. RESONANCE ABS.
                        1         5084          1.000000            0.122910
                        2         1271          0.099292            0.
```

FIGURE 22

```
1 LARGE TEST PROBLEM FOR INFINITE LATTICE ** SRL-HERESY I
          1          1          1          1
          7        250
         60          1          1          0          5          2
1       115          1          1          1          1          2
          4 12.7
         10 4.5                    12626.7     .00006963
        125 645.16     322.48
        175
        225 .9859      .92365
        275 1.3053     0.0
        325 58.826     0.0
        875 4.0        4.0        4.0        4.0        4.0
        925 4.0        4.0        4.0        4.0        4.0
        975 1.0        3.0        1.0        3.0        2.0
       1025 1.0        1.0        3.0        3.0        2.0
1      1075 1025.0
```

VI. OUTPUT INFORMATION

The SRL HERESY - I code outputs a variety of information both as a data summary and as an edit of the solved problem. In some cases information is printed more than once for convenience. The output format changes according to the particular problem being run so that additional lines appear in a few cases.

A. Equations Used in Edit of Solved Problem

At the end of the sequence of mathematical operations described in Section I-E, the code has solved for the thermal absorption in each rod type (normalized to the largest absorption), and the largest eigenvalue, k, of the system of equations given in equation I.A.1. These quantities are then used, with the input parameters, to obtain a variety of lattice parameters.

The lattice parameters calculated by the edit routine satisfy the relation

$$\overline{\eta} p f \chi = k_{corr} \qquad \text{(VI.A.1)}$$

where k_{corr} is the eigenvalue solved for above corrected for fast axial leakage, χ is the lattice nonleakage probability, f is the lattice thermal utilization, p is the lattice resonance escape probability, and $\overline{\eta}$ is the lattice average η value. The equations used to obtain these parameters, as well as other quantities of interest, are given in the following paragraphs.

1. Eigenvalue Corrected for Fast Axial Leakage

The eigenvalue obtained in solving equation I.A.1 takes into account both fast and slow radial leakage, but only slow axial leakage. Fast axial leakage is accounted for by assuming a cosine axial flux shape, and leads to a corrected k given by

$$k_{corr} = k \exp(-\tau B_z^2) \qquad \text{(VI.A.2)}$$

where k is the eigenvalue solved for in I.A.1, τ is the thermal age, and B_z^2 is the axial geometrical buckling. The age τ is given by

$$\tau = \sum_{j=1}^{3} B_j \tau_j \qquad \text{(VI.A.3)}$$

where the B_j are weighting fractions and the τ_j are thermal ages for each of the three components of the triple Gaussian slowing down distribution given in Section II-D.

2. Resonance Absorptions

Resonance absorptions occur in rods at the single equivalent resonance level, and are normalized to the highest thermal absorption. They are computed from the equation

$$i_n^R = \frac{A_n}{k_{corr}} \sum_{m=1}^{N} g_{nm} i_m \eta_m \qquad \text{(VI.A.4)}$$

where N is the number of rod types in the lattice, η_m is the thermal η parameter for the m^{th} rod type, i_m is the thermal absorption for the m^{th} rod type, g_{nm} is the number of neutrons born in all rods of type m that reach the resonance energy at a rod type n, A_n is the resonance absorption coefficient (see Section I-D) for rod type n, and k_{corr} is the corrected k from equation VI.A.2.

3. Approximate Moderator Thermal Absorption

The calculation of moderator absorption makes use of the definition of the heterogeneous f parameter given in Section I-B. From this definition, the relationship between moderator absorptions and rod absorptions in a cell with reflecting boundaries is

$$i_{mod} = i_{rod} \frac{(1 - f_H)}{f_H} \qquad \text{(VI.A.5)}$$

where i_{rod} is the total rod thermal absorptions, i_{mod} is the total moderator thermal absorptions, and f_H is the HERESY-defined parameter taken from an appropriate cell calculation.

This same expression may be extended in approximate form to the lattice calculation as

$$M = \sum_{j=1}^{K} N_j \bar{i}_j \frac{(1 - f_j)}{f_j} \qquad \text{(VI.A.6)}$$

where K is the number of rod kinds, N_K is the number of rods in the lattice of the k^{th} rod kind, and f_j is the HERESY-defined thermal f parameter for the k^{th} rod kind. M is the moderator thermal absorptions.

4. Total Leakage

The total leakage is defined as the difference between the total fission source and the number of neutrons absorbed (1) at resonance energy, (2) thermally absorbed in rods, and (3) thermally in moderator. This is expressed by

$$L = \frac{1}{k_{corr}} \sum_{m=1}^{N} i_m \eta_m (N_m - \sum_{t=1}^{N} N_t A_t g_{tm}) - \sum_{m=1}^{N} N_m i_m - M \qquad \text{(VI.A.7)}$$

where N_m is the number of rods of the m^{th} rod type in the lattice, and all other symbols are defined as previously.

5. Nonleakage Probability

This quantity is defined to be the total thermal absorptions divided by total thermal neutron source.

$$\chi = \frac{\sum_{m=1}^{N} N_m i_m + M}{\sum_{m=1}^{N} N_m i_m + M + L} \qquad \text{(VI.A.8)}$$

where all symbols have been defined previously.

6. Thermal Utilization

The definition of this quantity for the lattice is much the same as for the individual rods.

$$f_{lattice} = \frac{\text{Total thermal absorptions in rods}}{\text{Total thermal absorptions in rods plus moderator}}$$

or

$$f_{lattice} = \frac{\sum_{m=1}^{N} N_m i_m}{\sum_{m=1}^{N} N_m i_m + M} \qquad \text{(VI.A.9)}$$

where all symbols have been defined previously.

7. Resonance Escape Probability

A standard definition is used for computations as

$$p_{lattice} = \frac{\sum_{m=1}^{N} N_m i_m + M + L}{\sum_{m=1}^{N} N_m i_m + M + L + \sum_{m=1}^{N} N_m i_m^R} \qquad \text{(VI.A.10)}$$

where all symbols have been defined previously.

8. Average Rod Absorption

This is obtained by a simple arithmetic average over the rods in the lattice.

$$\bar{i} = \frac{\sum_{m=1}^{N} N_m i_m}{\sum_{m=1}^{N} N_m} \qquad \text{(VI.A.11)}$$

where all symbols have been previously defined.

9. Average Thermal Eta

A simple arithmetic average weighted by thermal absorptions is used as

$$\bar{\eta} = \frac{\sum_{m=1}^{N} N_m \eta_m i_m}{\sum_{m=1}^{N} N_m i_m} \qquad \text{(VI.A.12)}$$

where all symbols have been defined previously.

10. Average Absorptions for Each Rod Kind

A simple arithmetic average is made as

$$\bar{i}_p = \frac{\text{Absorptions in all rods of kind p}}{\text{Total number of rods of kind p}} \qquad \text{(VI.A.13)}$$

11. Infinite Lattice Material Buckling

If the lattice considered is radially infinite and $B_z^2 = 0$, then the eigenvalue found from solving equation I.A.1 is k_∞. For this case, the material buckling of the lattice may be solved for under the assumption that

$$L^2_{lattice} = (1 - f_{lattice}) L^2_{moderator} \qquad \text{(VI.A.14)}$$

The material buckling is obtained by successive iterations on the age-diffusion critical equation in the form

$$k_\infty = [1 + L^2_{mod}(1 - f)B_m^2]\exp(\tau B_m^2) \qquad \text{(VI.A.15)}$$

where all symbols have been defined previously.

B. Output Formats

Several examples of output formats for both finite and infinite problems were shown in Section V. The quantities are output in a particular order subject to modification by selection of the many options. The order of output for a routine problem is given, then possible modifications that can occur are itemized.

1. Routine Problem Format

A standard problem will produce the following format:

- A line summarizing the number of iterations necessary to converge to the final eigenvalue, and the last two eigenvalues used in the iteration. A comparison of the two eigenvalues assures that the problem did indeed converge.
- A table summarizing the rod parameters, thermal absorptions, and resonance absorptions for each rod type.
- A list of the edited lattice parameters. This includes k_{corr}, $\overline{i}$, $\overline{\eta}$, f, p, M, L, and χ.
- A table summarizing the average thermal and resonance absorptions for each rod kind.

2. Modifications to Routine Output Format

- If an edit of each iteration of a criticality search is requested, a line is printed prior to the line of the routine format (first item) giving the number of the current iteration (Figure 5).
- If the thermal utilization is zero for a rod kind, it is not possible to calculate the moderator absorptions as given in Section VI-A-3. Under these conditions, the value of M is set to zero and a line is printed under second item of the routine format to so indicate this. The effect of setting M = 0 is to force f to equal 1.0 and lump the moderator absorptions into the total leakage, however equation VI.A.1 is still satisfied.

- If the problem concerns an infinite lattice and B_z^2 is zero, the lattice material buckling is computed as described in Section VI-A-11 and inserted following third item of the routine format.

- If a criticality search on gamma of a rod kind has been performed, a line will be inserted following fourth item of the routine format giving the thermal utilization corresponding to this gamma value if it is possible to re-compute f. If insufficient information is available to re-compute f, a line will be inserted to so indicate.

VII. REFERENCES

A. Cited

1. C. N. Klahr, L. B. Mendelsohn, and J. Heitner. Heterogeneous Reactor Calculation Methods. USAEC Report NYO-2678, TRG, Inc., Syosset, N. Y. (1960). (TRG-129-QTR-6)

2. C. N. Klahr, J. Heitner, and N. Stein. Test and Verification of Heterogeneous Reactor Calculation Methods - Annual Report - April 1963 - January 1964. USAEC Report NYO-3194-1, Fundamental Methods Associates, Inc., New York.

3. S. M. Feinberg. "Heterogeneous Methods for Calculating Reactors," Proc. U.N. Intern. Conf. Peaceful Uses Atomic Energy, 1st, Geneva, 5, 484-500 (1955). P/669

4. A. D. Galanin. "Critical Size of Heterogeneous Reactor with Small Number of Rods." Ibid., 462-465 (1955). P/663

5. C. N. Klahr. Heterogeneous Reactor Calculation Methods. TRG, Incorporated, Syosset, N. Y. NYO-2673 (TRG-129-QTR-1) (1959).

B. General

(1) C. N. Klahr and L. B. Mendelsohn. Heterogeneous Reactor Calculation Methods. USAEC Report NYO-2674, TRG, Inc., Syosset, N. Y. (1959). (TRG-129-QTR-2)

(2) Ibid. NYO-2675 (1959). (TRG-129-QTR-3)

(3) Ibid. NYO-2676 (1959). (TRG-129-QTR-4)

(4) Ibid. NYO-2677 (1960). (TRG-129-QTR-5)

(5) Ibid. NYO-2679 (1960). (TRG-129-QTR-7)

(6) C. Sheffield. "The Individual Source-Sink Method for Systems of Finite Height." J. Nuclear Energy, Pts. A&B. Reactor Sci. and Technol., 16: 245-259 (May 1962).

(7) J. Ligou and C. Mignot. "Tridimensional Theory of Heterogeneous Reactors." Nucl. Sci. Eng. 19, 58-73 (May 1964).

(8) W. A. Horning. Small Source Model of a Thermal Pile. USAEC Report HW-24282, General Electric, Hanford Works, Richland, Wash. (1952)(declassified 1957).

APPENDIX A - Mnemonics Used in Code

The list below contains all of the primary symbols used in SRL - HERESY I, their dimensions, and their locations in the three large storage arrays. The code uses three arrays in COMMON STORAGE to store data, designated as the LL(225) or fixed point array, AA(15,500) or Link 1 floating point array, and the AAA(10,250) or Link 2 floating point array. Main Program 1 uses only the LL(225) and AA(15,500) arrays while Main Program 2 uses the LL(225), AA(15,500), and AAA(10,250)arrays. Shared storage is made up of arrays LL(225) and AA(15,500), the AAA(10,250) being the necessary amount of temporary storage to handle the 50 x 50 arrays used in Link 2.

The following list contains the majority of the mnemonics used. Those used internally, hence not appearing in the input, are indicated by an asterisk.

	Mnemonic	Location	Use
*	A(50)	AA(2342)...	Resonance absorption parameters for each rod type
*	A1	AA(1056)	Maximum value of V1 matrix
*	A2	AA(1077)	Maximum value of V2 matrix
	AR(48)	AA(325)...	Resonance absorption parameters for each rod kind
*	ABSRB(50)	AAA(10001)...	Normalized thermal absorptions for each rod type
	B(3)	AA(14)...	Weighting fractions for each term of triple Gaussian slowing down distribution
*	BGF(250)	AA(1142)...	Fission to thermal kernel values
*	BGFR(250)	AA(1392)...	Resonance to thermal kernel values
	BSQ	AA(11)	Axial buckling
	CR1	AA(1)	Convergence criterion on matrix inversion

	Mnemonic	Location	Use
	CR2	AA(2)	Convergence criterion on power iteration
	CR3	AA(3)	Convergence criterion on criticality search
*	D(48)	AA(2192)...	Moderator diffusion coefficient for each rod kind
	EIGEN	AA(9)	Eigenvalue to be searched for
*	ETA(50)	AA(2292)	Eta values for each rod type
	ETAI(48)	AA(275)	Eta values for each rod kind
*	FK1	AA(1078)	Value of uncorrected eigenvalue for the current iteration minus one
*	FK2	AA(1069)	Value of uncorrected eigenvalue for current iteration
*	FKMAX(50)	AA(2442)...	No. rods/type in floating point
	GAM(I)	AA(175)...	Gamma values for each rod kind
*	GAMMA(I)	AA(2242)...	Gamma values for each rod type
*	IMAX1	LL(174)	Number of largest element of V1 array
*	IMAX2	LL(175)	Number of largest element of V2 array
	INF	LL(1)	Finite/infinite lattice indicator
*	IT1	LL(170)	Iteration counter for matrix inversion
*	IT2	LL(171)	Iteration counter for power eigenvalue iteration
*	IT3	LL(172)	Print counter in power eigenvalue iteration
*	IT4	LL(173)	Iteration counter for criticality search

	Mnemonic	Location	Use
	KDAT	LL(168)	Omit data print indicator
	KERNEL	LL(2)	New kernel indicator
	KIND(50)	LL(115)...	Rod kind for each rod type
	KOOR	LL(3)	New coordinate indicator
	KPLOT	LL(61)	Geometry plot indicator
	KPRNT	LL(60)	Geometry list indicator
	KSEPR	LL(167)	Separate edit indicator (criticality searches)
	KSERCH	LL(166)	Rod kind to be searched over
	KSKIP	LL(62)	Geometry punch indicator
	KTEST	LL(169)	Last problem indicator
	LAT	LL(9)	Lattice type indicator
	LCRIT	LL(165)	Criticality search indicator
	N1	LL(94)	Max. No. iterations on matrix inversion
	N2	LL(95)	Max. No. iterations on eigenvalue
	N3	LL(96)	Max. No. iterations in criticality search
*	NERR	AA(24)	Error indicator
*	NF(50)	LL(9176)...	Indicator for f calculation for each rod kind
	NICPRT(50)	LL(10)...	No. initial coordinates for each rod type
	NKIND	LL(64)	Number of rod kinds
	NOGRID	LL(7)	Number of kernel mesh points

	Mnemonic	Location	Use
	NTYPE	LL(63)	Number of rod types
	NWRITE	LL(224)	Write indicator for library tapes
	RADII	AA(1075)	Radius beyond which no more rods are considered in infinite lattices
	RADIUS(48)	AA(375)...	Radius of each rod kind
	RADRFL	AA(7)	Reflector radius
*	RESABS(50)	AAA(10051)...	Resonance absorption for each rod type
	ROEMIN	AA(8)	A radius within which the reflector may be considered infinite
	SIGA	AA(13)	Moderator absorption cross section
	SIGC(48)	AA(75)...	Absorption cross section for gamma calculations for each rod kind
	SM(48)	AA(2142)...	Self-contribution to thermal diffusion kernel for each rod kind
*	SMF(250)	AA(1892)...	Thermal diffusion kernel points
*	SMFO(50)	AA(2392)...	Self-contribution to thermal diffusion kernel for each rod type
*	SMGR(250)	AA(1642)...	Resonance to thermal kernel values
*	SR(50,50)	AA(8500)...	Resonance to thermal slowing down matrix
*	T(50,50)	AA(3500)...	Fission to thermal slowing down matrix
*	T1(50,50)	AAA(1)...	Temporary matrix for computations
*	T2(50,50)	AAA(2501)...	Temporary matrix for computations
*	T3(50,50)	AAA(5001)...	Temporary matrix for computations
*	T4(50,50)	AAA(7501)...	Temporary matrix for computations

	Mnemonic	Location	Use
*	T5(50)	AA(13,500)...	Temporary matrix for computations
*	T6(50)	AA(13,550)...	Temporary matrix for computations
	TAU(3)	AA(17)...	Fission to thermal age
	TAUR(3)	AA(20)...	Fission to resonance age
*	TAUTOL	AA(23)	= EXPF {-BSQ*[B(1)*TAU(1)+ B(2)*TAU(2)+B(3)*TAU(3)]}
	THERU(50)	AA(225)...	Thermal utilization for each rod kind
	THETA(50,3)	AA(725)...	Angular initial coordinates
	TITLE(12)	AA(1130)...	Problem alphanumeric data
*	TR(50,50)	AA(6000)...	Fission to resonance slowing down matrix
*	V1(50)	AAA(10151)...	Previous set of unnormalized eigenvectors
*	V2(50)	AAA(10201)...	Current set of unnormalized eigenvectors
	VCELL(48)	AA(125)...	Cell volume for each rod kind
	V0(50)	AA(1080)...	Initial approximation to eigenvectors
*	VRATIO(50)	AAA(10101)...	Ratio of V2/A2 for each rod type
*	W(8)	AA(2492)...	Miscellaneous transfer locations
	WDX(50)	AA(875)...	X direction pitch in infinite lattice
	WDY(50)	AA(925)...	Y direction pitch in infinite lattice
	WX1(50)	AA(975)...	Initial X position in infinite lattice

Mnemonic	Location	Use
WY1(50)	AA(1025)...	Initial Y position in infinite lattice
X(6500)	AA(2500)	X coordinates of all rods
XCOR	AA(5)	Scale factor on all coordinates
XLSQ	AA(12)	L^2 for moderator
XLSQ2(50)	AA(25)...	L^2 of moderator for each rod kind - used in self-consistent calculation of Y
XMESH	AA(10)	Mesh size for kernel
XO(50,3)	AA(425)...	Initial X coordinates
XSCALE	AA(5)	Scale increment along X axis in plotter
Y(6500)	AA(9000)...	Y coordinates of all rods
YO(50,3)	AA(575)...	Initial Y coordinates
YSCALE	AA(6)	Scale increment along Y axis in plotter
* Z(50,50)	AA(11,000)...	Thermal diffusion kernel matrix

The three Arrays LL, AA, and AAA are stored in order in common storage, i.e. LL(1) = location 774628 in storage. All mnemonics are then assigned to locations in these three arrays by EQUIVALENCE statements. Information concerning the three large arrays follow.

Array	First Position in Storage	Last Position in Storage
$LL(225_{10})$	$32,562_{10}$ or 77462_8	$32,338_{10}$ or 77122_8
$AA(15,500_{10})$	$32,337_{10}$ or 77121_8	$16,838_{10}$ or 40706_8
$AAA(10,250_{10})$	$16,837_{10}$ or 40705_8	$6,588_{10}$ or $14,674_8$

APPENDIX B - FORTRAN II Listings

The source decks for all FORTRAN II and IBM 704 SAP subroutines used in SRL - HERESY I are listed on the following pages. Two sections appear which correspond to the two links in the CHAIN code. The CHAIN subroutine and related codes (i.e. CPTG, EPTC, and SLCI routines) are special versions not available from another source. Appendix D gives information on the CHAIN codes, as well as listings of their source cards.

Link 1

Link 1 consists of 13 FORTRAN II subprograms, a package of library function subroutines, and a package of FORTRAN II control subroutines. COMMON STORAGE begins at location 40705_8 in this link.

FORTRAN II Subprograms

Subprogram Name	Function
MAIN PROG. 1	Main Prog 1 executive routine
DATIN	Data reading subroutine
DATCHK	Data checking subroutine
KERGEN	Kernel generator subroutine
GEOM	Finite geometry subroutine
INGEOM	Infinite geometry subroutine
PLOT	Geometry plotting subroutine
PRINT	Geometry listing subroutine
DATPRT	Data summary output subroutine
HEX	Hexagonal geometry subroutine
SQARE	Square geometry subroutine
CIRCLE	Circular pattern geometry subroutine
PUNCH	Geometry punch subroutine

FORTRAN II Library Function Subroutines

The following functions must appear in the program with the call names as listed:

FORTRAN Name	Function
BESSKK (arg)	K_o (arg)
ATANF (arg)	$\tan^{-1}$ (arg)
LOGF (arg)	ln (arg)
EXPF (arg)	$e^{(arg)}$
EXPIF (arg)	E_1(arg) integral
SINF (arg)	sin (arg)
COSF (arg)	cos (arg)
SQRFT (arg)	$\sqrt{arg}$

FORTRAN II Control Subroutines

FORTRAN Name	Operation
(FIL)	Format interpretive loader
(BDC)	Binary to decimal conversion
(DBC)	Decimal to binary conversion
(IOH)O	Input - Output Hollerith, Output
(IOH)I	Input - Output Hollerith, Input
(SPH)	Storage to printer Hollerith
(STH)	Storage to tape Hollerith
(TSH)	Tape to storage Hollerith
(SCH)	Storage to card Hollerith
(RTN)	Return
(LEV)	Leave
(LFM)	Leave floating point trap
(EFM)	Enter floating point trap
CHAIN	Link search routine
CPTG	Program tape link generator

```
C      DIMENSION DECK FOR SRL HERESY 1 MAIN PROGRAM 1
C
C           DIMENSION DECK FOR MAIN PROGRAM 1      VERSION OF 10/2/64
C
      DIMENSION LL(225),AA(15500)
      DIMENSION NICPRT(50),NRPRT(50),KIND(50),NF(50)
      DIMENSION B(3),TAU(3),TAUR(3),XLSQ2(50),SIGC(50),VCELL(50),GAM(5
     1     ,THERU(50),ETAI(50),AL(50),RADIUS(50),XO(50,3),YO(50,3),
     2   THETA(50,3),WDX(50),WDY(50),WX1(50),WY1(50),VO(50),W(8),
     3     TITLE(12),BGF(250),BGFR(250),SMGR(250),SMF(250),SM(50),D(50
     4     GAMMA(50),ETA(50),A(50),SMFO(50),FKMAX(50),X(6500),Y(6500)
      COMMON LL,AA
C
C           EQUIVALENCES FOR LL ARRAY
C
      EQUIVALENCE (INF,LL(1)),(KERNEL,LL(2)),(KOOR,LL(3)),(N1,LL(4)),
     1     (N2,LL(5)),(N3,LL(6)),(NOGRID,LL(7)),(NTHER,LL(8)),(LAT,
     2     LL(9)),(NICPRT,LL(10)),(KPRNT,LL(60)),(KPLOT,LL(61)),(KSKIP,
     3     LL(62)),(NTYPE,LL(63)),(NKIND,LL(64)),(NRPRT,LL(65)),(KIND,
     4     LL(115)),(LCRIT,LL(165)),(KSERCH,LL(166)),(NSEPR,LL(167)),
     5     (KDAT,LL(168)),(KTEST,LL(169)),(IT1,LL(170)),(IT2,LL(171)),
     6     (IT3,LL(172)),(IT4,LL(173)),(IMAX1,LL(174)),(IMAX2,LL(175)),
     7(NF,LL(176))
C
C           EQUIVALENCES FOR AA ARRAY
C
      EQUIVALENCE (CR1,AA(1)),(CR2,AA(2)),(CR3,AA(3)),(XCOR,AA(4)),
     1     (XSCALE,AA(5)),(YSCALE,AA(6)),(RADRFL,AA(7)),(ROEMIN,AA(8)),
     2     (EIGEN,AA(9)),(XMESH,AA(10)),(BSQ,AA(11)),(XLSQ,AA(12)),
     3     (SIGA,AA(13)),(B,AA(14)),(TAU,AA(17)),(TAUR,AA(20)),(TAUTOL,
     4     AA(23)),(NERR,AA(24)),(XLSQ2,AA(25)),(SIGC,AA(75)),(VCELL,
     5     AA(125)),(GAM,AA(175)),(THERU,AA(225)),(ETAI,AA(275)),(AL,
     6     AA(325)),(RADIUS,AA(375)),(XO,AA(425)),(YO,AA(575)),(THETA,
     7     AA( 725)),(WDX,AA( 875)),(WDY,AA( 925)),(WX1,AA( 975)),(WY1,
     8     AA(1025)),(RADII,AA(1075)),(A1,AA(1076)),(A2,AA(1077)),(FK1,
     9     AA(1078)),(FK2,AA(1079)),(VO,AA(1080)),(W,AA(2492))
      EQUIVALENCE (BGF,AA(1142)),(BGFR,AA(1392) ),(SMGR, AA(1642)),
     1     (SMF,AA(1892)),(SM,AA(2142)),(D,AA(2192)),(GAMMA,AA(2242)),
     2     (ETA,AA(2292)),(A,AA(2342)),(SMFO,AA(2392)),(FKMAX,AA(2442))
     3     (X,AA(2500)),(Y,AA(9000)),(TITLE,AA(1130))
G
```

```
      SRL HERESY I MAIN PROGRAM 1                VERSION OF 10/12/64          HER-P
C                                                                             HER-P
C     SUBROUTINE MAIN                                                          DIME
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1                                HER-P
C                                                                             HER-P
      CALL EFM                                                                HER-P
      IN=8                                                                    HER-P
      NOUT=10                                                                 HER-P
    5 READ INPUT TAPE IN,10,(TITLE(I),I=1,12)                                 HER-P
   10 FORMAT(12A6)                                                            HER-P
      NERR=0                                                                  HER-P
      PRINT 15,(TITLE(I),I=1,12)                                              HER-P
   15 FORMAT(1HO,12A6)                                                        HER-P
      WRITE OUTPUT TAPE NOUT,10,(TITLE(I),I=1,12)                             HER-P
      CALL DATIN                                                              HER-P
      CALL DATCHK                                                             HER-P
      IF(NERR)40,40,20                                                        HER-P
   20 PRINT 30                                                                HER-P
   30 FORMAT(1HO,61HAN ERROR HAS BEEN DETECTED IN THE INPUT DATA FOR THI      HER-P
     1S PROBLEM/1H ,60HAN ATTEMPT WILL BE MADE TO CONTINUE WITH SUCCESSI      HER-P
     2VE PROBLEMS)                                                            HER-P
      GO TO 5                                                                 HER-P
   40 IF(KERNEL)60,60,50                                                      HER-P
   50 CALL KBRGEN                                                             HER-P
      IF(NERR)60,60,20                                                        HER-P
   60 IF(KOOR-1)100,70,100                                                    HER-P
   70 IF(INF)90,80,90                                                         HER-P
   80 CALL GEOM                                                               HER-P
      IF(NERR)100,100,20                                                      HER-P
   90 CALL INGEOM                                                             HER-P
      IF(NERR)100,100,20                                                      HER-P
  100 IF(KPLOT)105,105,101                                                    HER-P
  101 CALL PLOT                                                               HER-P
  105 IF(KPRNT)109,109,106                                                    HER-P
  106 CALL PRINT                                                              HER-P
  109 DO 110 I=1,NTYPE                                                        HER-P
  110 FKMAX(I)=NRPRT(I)                                                       HER-P
      TAUTOL=EXPF(-BSQ*(B(1)*TAU(1)+B(2)*TAU(2)+B(3)*TAU(3)))                 HER-P
      DO 160 I=1,NKIND                                                        HER-P
      D(I)=XLSQ2(I)*SIGC(I)                                                   HER-P
      XYLSQ=XLSQ2(I)/(1.+BSQ*XLSQ2(I))                                        HER-P
      XXLSQ=SQRTF(XYLSQ)                                                      MAT
      IF(RADIUS(I))120,120,130                                                HER-P
  120 SM(I)=0.101010                                                          HER-P
      GO TO 140                                                               HER-P
  130 ARG1=RADIUS(I)/XXLSQ                                                    HER-P
      ARG2=0.0                                                                HER-P
      ARG3=0.0                                                                HER-P
      ARG4=0.0                                                                HER-P
      ARG5=0.0                                                                HER-P
      ARG6=0.0                                                                HER-P
      ARG7=0.0                                                                HER-P
      CALL BESSI(ARG1,ARG2,ARG3,ARG4,ARG5)                                    HER-P
      CALL BESSKK(ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7)                         HER-P
      SM(I)=(1./(6.2831852*D(I)))*ARG4                                        HER-P
  140 IF(GAM(I))160,150,160                                                   HER-P
  150 GAM(I)=XLSQZ(I)*(1.-THERU(I))/(D(I)*VCELL(I)*THERU(I))+                 HER-P
     1(1./(12.56637 *D(I)))*LOGF(    XYLSQ/VCELL(I))+.228714/D(I)-SM(I)
```

```
      NF(I)=1                                                          HER-?
  160 CONTINUE                                                         HER-?
      DO 170 I=1,NTYPE                                                 HER-?
      NX=KIND(I)                                                       HER-?
      ETA(I)=ETAI(NX)                                                  HER-?
      GAMMA(I)=GAM(NX)                                                 HER-?
      A(I)=AL(NX)                                                      HER-?
  170 SMFO(I)=SM(NX)                                                   HER-?
      IF(KOOR-2)210,180,210
  180 NX=0                                                             HER-?
      DO 190 I=1,NTYPE                                                 HER-?
  190 NX=NX+NRPRT(I)                                                   HER-?
      DO 200 I=1,NX                                                    HER-?
      X(I)=X(I)*XCOR                                                   HER-?
  200 Y(I)=Y(I)*XCOR                                                   HER-?
  210 IF(KDAT)220,220,230                                              HER-?
  220 CALL DATPRT                                                      HER-?
  230 CALL CHAIN(2,1)                                                  HER-?
      END(2,1,0,0,0)                                                   HER-?
```

```
C       SRL HERESY 1 SUBROUTINE DATIN          VERSION OF 10/2/64      DA
C                                                                      DA
      SUBROUTINE DATIN                                                 DA
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
      CALL EFM                                                         DA
      IN=8                                                             DA
   10 READ INPUT TAPE IN,20,LAST,NUMD,LOCATE,(KK(I),I=1,5)             DA
   20 FORMAT(I1,3XI1,2XI5,5I12)                                        DA
      IF(NUMD)22,22,24                                                 DA
   22 NUMD=5                                                           DA
   24 DO 30 I=1,NUMD                                                   DA
      M=LOCATE+I-1                                                     DA
   30 LL(M)=KK(I)                                                      DA
      IF(LAST)10,10,40                                                 DA
   40 READ INPUT TAPE IN,50,LAST,NUMD,LOCATE,(AX(I),I=1,5)             DA
   50 FORMAT(I1,3XI1,2XI5,5F12.)                                       DA
      IF(NUMD)52,52,54                                                 DA
   52 NUMD=5                                                           DA
   54 DO 60 I=1,NUMD                                                   DA
      M=LOCATE+I-1                                                     DA
   60 AA(M)=AX(I)                                                      DA
      IF(LAST)40,40,70                                                 DA
   70 RETURN                                                           DA
      END(2,1,0,0,0)                                                   DA
```

```
C       SRL HERESY 1 SUBROUTINE  DATCHK  VERSION OF 10/2/64                    DAT
C                                                                              DAT
      SUBROUTINE DATCHK                                                        DAT
C                                                                             DIME
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C                                                                             DIME
      CALL EFM                                                                 DAT
      NOUT=10                                                                  DAT
   40 IF(NKIND-NTYPE)70,70,50                                                  DAT
   50 WRITE OUTPUT TAPE NOUT,60                                                DAT
   60 FORMAT(1HO,63HTHE NUMBER OF ROD KINDS IS GREATER THAN THE NUMBER O       DAT
     1F ROD TYPES)                                                            DAT
      NERR=1                                                                   DAT
   70 IF(NTYPE-50)100,100,80                                                   DAT
   80 NERR=1                                                                   DAT
      WRITE OUTPUT TAPE NOUT,90                                                DAT
   90 FORMAT(1HO,34HTHE NUMBER OF ROD TYPES EXCEEDS 50)                        DAT
  100 IF(KOOR-1)105,128,105                                                    DAT
  105 DO 125 I=1,NTYPE                                                         DAT
      IF(NRPRT(I))110,110,125                                                  DAT
  110 WRITE OUTPUT TAPE NOUT,120                                               DAT
  120 FORMAT(1HO,26H  A ROD TYPE HAS ZERO RODS)                                DAT
      NERR=1                                                                   DAT
  125 CONTINUE                                                                 DAT
  128 DO 160 I=1,NTYPE                                                         DAT
  130 IF(KIND(I))140,140,160                                                   DAT
  140 WRITE OUTPUT TAPE NOUT,150                                               DAT
  150 FORMAT(1HO,38HA ROD TYPE HAS NO KIND NUMBER ASSIGNED)                    DAT
      NERR=1                                                                   DAT
  160 CONTINUE                                                                 DAT
      IF(N1)170,170,180                                                        DAT
  170 N1=20                                                                    DAT
  180 IF(N2)190,190,200                                                        DAT
  190 N2=200                                                                   DAT
  200 IF(N3)203,203,208                                                        DAT
  203 N3=25                                                                    DAT
  208 IF(KOOR-1)240,210,240                                                    DAT
  210 IF(LAT-2)240,240,220                                                     DAT
  220 WRITE OUTPUT TAPE NOUT,230                                               DAT
      NERR=1                                                                   DAT
  230 FORMAT(1HO,60HTHE LATTICE TYPE TO BE GENERATED IS NOT OF A RECOGNI       DAT
     1ZED TYPE)                                                               DAT
  240 IF(LCRIT)230,280,250                                                     DAT
  250 IF(LCRIT-2)255,255,280                                                   DAT
  255 IF(KSERCH)260,260,280                                                    DAT
  260 NERR=1                                                                   DAT
      WRITE OUTPUT TAPE NOUT,270                                               DAT
  270 FORMAT(1HO,54HA ROD KIND IS NOT SPECIFIED FOR THE CRITICALITY SEAR       DAT
     1CH)                                                                     DAT
  280 IF(CR1)290,290,300                                                       DAT
  290 CR1=.000001                                                              DAT
  300 IF(CR2)310,310,320                                                       DAT
  310 CR2=.00001                                                               DAT
  320 IF(CR3)330,330,340                                                       DAT
  330 CR3=.00001                                                               DAT
  340 IF(B(1))350,350,360                                                      DAT
  350 B(1)=0.56                                                                DAT
  360 IF(B(2))370,370,380                                                      DAT
  370 B(2)=0.44                                                                DAT
```

```
380 IF(B(3))390,390,400                                                    DAT
390 B(3)=0.0                                                               DAT
400 IF(TAU(1))410,410,420                                                  DAT
410 TAU(1)=154.3                                                           DAT
420 IF(TAU(2))430,430,440                                                  DAT
430 TAU(2)=68.3                                                            DAT
440 IF(TAU(3))450,450,460                                                  DAT
450 TAU(3)=0.0                                                             DAT
460 IF(TAUR(1))470,470,480                                                 DAT
470 TAUR(1)=126.5                                                          DAT
480 IF(TAUR(2))490,490,500                                                 DAT
490 TAUR(2)=40.5                                                           DAT
500 IF(TAUR(3))510,510,520                                                 DAT
510 TAUR(3)=0.0                                                            DAT
520 DO 560 I=1,NKIND                                                       DAT
    IF(XLSQ2(I))530,530,540                                                DAT
530 XLSQ2(I)=XLSQ                                                          DAT
540 IF(SIGC(I))550,550,560                                                 DAT
550 SIGC(I)=SIGA                                                           DAT
560 CONTINUE                                                               DAT
    IF(XCOR)558,555,558                                                    DAT
555 XCOR=1.0                                                               DAT
558 IF(KOOR-1)610,570,610                                                  DAT
570 IF(XSCALE)580,580,590                                                  DAT
580 XSCALE=XCOR                                                            DAT
590 IF(YSCALE)600,600,610                                                  DAT
600 YSCALE=XCOR                                                            DAT
610 IF(LCRIT)630,630,620                                                   DAT
620 IF(EIGEN)625,625,630                                                   DAT
625 EIGEN=1.0                                                              DAT
630 DO 650 I=1,NTYPE                                                       DAT
    IF(VO(I))640,640,650                                                   DAT
640 VO(I)=1.0                                                              DAT
650 CONTINUE                                                               DAT
    IF(KERNEL-1)720,675,675                                                DAT
675 IF(XMESH)700,700,680                                                   DAT
680 IF(XLSQ)700,700,690                                                    DAT
690 IF(SIGA)700,700,720                                                    DAT
700 NERR=1                                                                 DAT
    WRITE OUTPUT TAPE NOUT,710                                             DAT
710 FORMAT(1HO,62HTHE INFORMATION GIVEN FOR THE KERNEL GENERATOR IS IN     DAT
   1SUFFICIENT)                                                            DAT
720 DO 780 I=1,NKIND                                                       DAT
    IF(GAM(I))730,730,780                                                  DAT
730 IF(THERU(I))760,760,740                                                DAT
740 IF(VCELL(I))760,760,750                                                DAT
750 IF(SIGC(I))760,760,780                                                 DAT
760 NERR=1                                                                 DAT
    WRITE OUTPUT TAPE NOUT,770,I                                           DAT
770 FORMAT(1HO,35HTHE INFORMATION GIVEN FOR ROD KIND I2,35H IS INSUFFI     DAT
   1CIENT TO COMPUTE A GAMMA)                                              DAT
780 CONTINUE                                                               DAT
    IF(NERR)810,810,790                                                    DAT
790 WRITE OUTPUT TAPE NOUT,800                                             DAT
800 FORMAT(1HO,71HBECAUSE OF THE ERROR(S) DETECTED ABOVE THIS PROBLEM      DAT
   1CANNOT BE CONTINUED/1H ,88HSUCCESSIVE PROBLEMS WILL BE ATTEMPTED,      DAT
   2ALTHOUGH THEY MAY BE AFFECTED BY THE ERRORS ABOVE)                     DAT
810 RETURN                                                                 DAT
    END(2,1,0,0,0)                                                         DAT
```

```
C          SRL HERESY 1  SUBROUTINE KERGEN 1       VERSION OF 10/2/64
C
      SUBROUTINE KERGEN
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EBM
      NOUT=10
      I1=0
      I2=0
      DENL=1.+BSQ*XLSQ
      XXLSQ=XLSQ/DENL
      SIGMA=SIGA*DENL
      CC=1./(6.2831852*XXLSQ*SIGMA)
      RLSQ=SQRTF(XXLSQ)
      IF(ABSF(B(2)))340,20,10
   10 IF(ABSF(B(3)))340,30,40
   20 NN=1
      GO TO 50
   30 NN=2
      GO TO 50
   40 NN=3
   50 DO 380 I=1,NOGRID
      FI=I-1
      DX=FI*XMESH
      ARG1=DX/RLSQ
      IF(ARG1)340,60,70
   60 IFPATH=3
      GO TO 80
   70 IFPATH=1
      ARG2=0.0
      ARG3=0.0
      ARG4=0.0
      ARG5=0.0
      ARG6=0.0
      ARG7=0.0
      CALL BESSI(ARG1,ARG2,ARG3,ARG4,ARG5)
      CALL BESSKK(ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7)
   80 SMGR(I)=0.0
      D2BY4=DX**2/4.
      DO 90 J=1,NN
   90 SMGR(I)=SMGR(I)+(B(J)/(TAUR(J)*12.5663705))*EXPF(-D2BY4/TAUR(J))
      DO 100 J=1,NN
  100 WTAU(J)=TAU(J)
  105 WBG=0.0
      GO TO (210,210,110,110),IFPATH
  110 DO 180 J=1,NN
      IF(ABSF(WTAU(J)/XXLSQ)-87.3)140,140,120
  120 IF(I1)125,125,135
  125 WRITE OUTPUT TAPE NOUT,130
  130 FORMAT(1H0,68HTHE ARGUMENT OF THE E1 INTEGRAL IN THE KERNEL GENERA
     1TOR EXCEEDS 87.3/1H ,85HTHE VALUE OF THE INTEGRAL WILL BE SET EQUA
     2L TO 1.0E-30 AND COMPUTATIONS WILL CONTINUE)
      I1=1
  135 CON(J)=1.0E-30
      GO TO 180
  140 IF(ABSF(WTAU(J)/XXLSQ)-.00001)150,150,170
  150 IF(I2)155,155,165
  155 WRITE OUTPUT TAPE NOUT,160
```

```
160 FORMAT(1HO,81HTHE ARGUMENT OF THE E1 INTEGRAL IN THE KERNEL GENERA
   1TOR IS WITHIN 1.0E-05 OF ZERO/1H ,85HTHE VALUE OF THE INTEGRAL WILL
   2L BE SET EQUAL TO 1.0E+30 AND COMPUTATIONS WILL CONTINUE)
    I2=1
165 CON(J)=1.0E+30
    GO TO 180
170 CON(J)=EXPF(-WTAU(J)/XXLSQ)
180 CONTINUE
    DO 190 J=1,NN
190 WBG=WBG-(B(J)/2.)*CC*EXPF(WTAU(J)/XXLSQ)*CON(J)
    GO TO (340,340,200,360),IFPATH
200 BGF(I)=WBG
    IFPATH=4
    GO TO 290
210 DO 260 J=1,NN
    IF(ABSF(D2BY4/WTAU(J))-87.3)230,230,220
220 IF(I1)225,225,228
225 WRITE OUTPUT TAPE NOUT,130
    I1=1
228 CON(J)=1.0E-30
    GO TO 260
230 IF(ABSF(D2BY4/WTAU(J))-.00001)240,240,250
240 IF(I2)245,245,248
245 WRITE OUTPUT TAPE NOUT,160
    I2=1
248 CON(J)=1.0E+30
    GO TO 260
250 CON(J)=EXPF(-D2BY4/WTAU(J))
260 CONTINUE
    DO 270 J=1,NN
270 WBG=WBG+CC*EXPF(WTAU(J)/XXLSQ)*B(J)*(ARG4+(WTAU(J)/2./XXLSQ)*
   1EXPF(-D2BY4/WTAU(J))+(.5+D2BY4/(2.*XXLSQ))*CON(J))
    GO TO (280,360,340,340),IFPATH
280 BGF(I)=WBG
    IFPATH=2
290 DO 300 J=1,NN
300 WTAU(J)=TAU(J)-TAUR(J)
    DO 330 J=1,NN
    IF(WTAU(J))310,310,330
310 WRITE OUTPUT TAPE NOUT,320
320 FORMAT(1HO,69HA TAU(R) VALUE IS LARGER THAN OR EQUAL TO ITS CORRES
   1PONDING TAU VALUE/1H ,51HA KERNEL CANNOT BE GENERATED UNDER THESE
   2CONDITIONS)
    NERR=1
    RETURN
330 CONTINUE
    GO TO (340,105,340,105),IFPATH
340 WRITE OUTPUT TAPE NOUT,350
350 FORMAT(1HO,70HA MACHINE ERROR HAS OCCURRED IN THE KERNEL GENERATOR
   1 - RESTART PROBLEM)
    PRINT 350
    NERR=1
    RETURN
360 BGFR(I)=WBG
    SMF(I)=CC*ARG4
    IF(NTHER)380,380,370
370 BGF(I)=SMF(I)
    BGFR(I)=SMF(I)
    SMGR(I)=1.
380 CONTINUE
430 RETURN
    END(2,1,0,0,01
```

```
C       SRL HERESY 1 SUBROUTINE  GEOM    VERSION OF 10/14/64
C
      SUBROUTINE GEOM
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL BPM
      NOUT=10
      IF(LAT)10,10,20
   10 CALL HEX
      GO TO 50
   20 IF(LAT-1)30,30,40
   30 CALL SQUARE
      GO TO 50
   40 CALL CIRCLE
   50 IF(KSKLP)90,90,80
   80 CALL PUNCH
   90 NX=0
      DO 120 I=1,NTYPE
  120 NX=NX+NRPRT(I)
      DO 130 I=1,NX
      X(I)=X(I)*XCOR
  130 Y(I)=Y(I)*XCOR
      IF(NX-6500)160,160,140
  140 WRITE OUTPUT TAPE NOUT,150
  150 FORMAT(1H0,37HTHE TOTAL NUMBER OF RODS EXCEEDS 6500)
      NERR=1
  160 RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE HEX      VERSION OF 10/2/64
C
      SUBROUTINE HEX
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EPM
      NOUT=10
      K=0
      DO 160 I=1,NTYPE
      L=1
      J=0
    5 IF(YO(L,L))10,40,10
   10 IF(XO(I,L))60,20,60
   20 NERR=1
      WRITE OUTPUT TAPE NOUT,30
   30 FORMAT(1HO,69HERROR IN GEOMETRY GENERATOR - INITIAL COORDINATES AR
     1E IN WRONG SECTOR)
      RETURN
   40 ANGLE=0.0
      IF(XO(L,L))70,50,70
   50 NN=1
      M=K+NN
      X(M)=0.0
      Y(M)=0.0
      GO TO 150
   60 TAN=YO(I,L)/XO(I,L)
      ANGLE=ATANF(TAN)
   70 RADIX=SQRTF(XO(I,L)*XO(I,L)+YO(I,L)*YO(I,L))
      DO 120 N=1,6
      XN=FLOATF(N)
      NN=J+N
      M=J+N+K
      X(M)=RADIX*COSF(ANGLE+(XN-1.)*1.0471975)
   80 Y(M)=RADIX*SINF(ANGLE+(XN-1.)*1.0471975)
      IF(ABSF(X(M))-.001)90,90,100
   90 X(M)=0.0
  100 IF(ABSF(Y(M))-.001)110,110,120
  110 Y(M)=0.0
  120 CONTINUE
  130 IF(NICPRT(I)-L)150,150,140
  140 L=L+1
      J=J+6
      GO TO 5
  150 NRPRT(I)=NN
      K=K+NN
  160 CONTINUE
      RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE SQUARE        VERSION OF 10/2/64
C
      SUBROUTINE SQUARE
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EFM
      NOUT=10
      K=0
      DO 260 I=1,NTYPE
      L=1
      J=0
      N=1
   10 IF(YO(I,L))20,40,20
   20 IF(XO(I,L))60,30,60
   30 NERR=1
      WRITE OUTPUT TAPE NOUT,35
   35 FORMAT(1HO,69HERROR IN GEOMETRY GENERATOR - INITIAL COORDINATES AR
     1E IN WRONG SECTOR)
      RETURN
   40 ANGLE=0.0
      IF(XO(I,L))70,50,70
   50 M=K+N
      X(M)=0.0
      Y(M)=0.0
      GO TO 250
   60 TAN=YO(I,L)/XO(I,L)
      ANGLE=ATANF(TAN)
   70 RADIX=SQRTF(XO(I,L)*XO(I,L)+YO(I,L)*YO(I,L))
      ANGLE1=ANGLE
   80 M=K+N
      X(M)=RADIX*COSF(ANGLE1)
      Y(M)=RADIX*SINF(ANGLE1)
      IF(ABSF(X(M))-.001)90,90,100
   90 X(M)=0.0
  100 IF(ABSF(Y(M))-.001)110,110,120
  110 Y(M)=0.0
  120 J=J+1
      GO TO (150,160,170,180,190,200,210,230),J
  130 IF(ABSF(ANGLE2-ANGLE1)-.001)120,120,140
  140 N=N+1
      ANGLE1=ANGLE2
      GO TO 80
  150 ANGLE2=1.5707963-ANGLE
      GO TO 130
  160 ANGLE2=1.5707963+ANGLE
      GO TO 130
  170 ANGLE2=3.1415926-ANGLE
      GO TO 130
  180 ANGLE2=3.1415926+ANGLE
      GO TO 130
  190 ANGLE2=4.7123889-ANGLE
      GO TO 130
  200 ANGLE2=4.7123889+ANGLE
      GO TO 130
  210 ANGLE2=6.2831852-ANGLE
      IF(ANGLE2-6.2831852)220,120,120
  220 IF(ABSF(ANGLE2-6.2831852)-.001)120,120,130
  230 IF(NICPRT(I)-L)250,250,240
  240 L=L+1
      N=N+1
      J=0
      GO TO 10
  250 NRPRT(I)=N
      K=K+N
  260 CONTINUE
      RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE CIRCLE          VERSION OF 10/2/64
C
      SUBROUTINE CIRCLE
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EPM
      NOUT=10
      K=0
      DO 250 I=1,NTYPE
      L=1
      J=1
   20 IF(XO(I,L))70,30,70
   30 IF(YO(I,L))40,60,50
   40 ANGLE=4.7123889
      GO TO 140
   50 ANGLE=1.5707963
      GO TO 140
   60 M=K+J
      X(M)=0.0
      Y(M)=0.0
      GO TO 240
   70 IF(XO(I,L))80,80,110
   80 IF(YO(I,L))90,100,100
   90 YO(I,L)=ABSF(YO(I,L))
      XO(I,L)=ABSF(XO(I,L))
      ANGLE=3.1415926+ATANF(YO(I,L)/XO(I,L))
      GO TO 140
  100 YO(I,L)=ABSF(YO(I,L))
      XO(I,L)=ABSF(XO(I,L))
      ANGLE=3.1415926-ATANF(YO(I,L)/XO(I,L))
      GO TO 140
  110 IF(YO(I,L))120,120,130
  120 YO(I,L)=ABSF(YO(I,L))
      XO(I,L)=ABSF(XO(I,L))
      ANGLE=6.2831852-ATANF(YO(I,L)/XO(I,L))
      GO TO 140
  130 YO(I,L)=ABSF(YO(I,L))
      XO(I,L)    =ABSF(XO(I,L))
      ANGLE=ATANF(YO(I,L)/XO(I,L))
  140 RADIX=SQRTF(XO(I,L)*XO(I,L)+YO(I,L)*YO(I,L))
      ANGLE1=ANGLE
  150 M=J+K
      X(M)=RADIX*COSF(ANGLE1)
      Y(M)=RADIX*SINF(ANGLE1)
      IF(ABSF(X(M))-.001)160,160,170
  160 X(M)=0.0
  170 IF(ABSF(Y(M))-.001)180,180,190
  180 Y(M)=0.0
  190 ANGLE2=ANGLE1+THETA(I,L)/57.29578
      IF(ANGLE2-ANGLE-6.2831822)200,200,220
  200 IF(ABSF(ANGLE2-ANGLE-6.2831852)-.001)220,220,210
  210 ANGLE1=ANGLE2
      J=J+1
      GO TO 150
  220 IF(NICPRT(I)-L)240,240,230
  230 L=L+1
      J=J+1
      GO TO 20
  240 NRPRT(I)=J
      K=K+J
  250 CONTINUE
      RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE PRINT          VERSION OF 10/2/64
C
      SUBROUTINE PRINT
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EFM
      NOUT=10
      K=0
      WRITE OUTPUT TAPE NOUT,10,(TITLE(I),I=1,12),XCOR
   10 FORMAT(1H1,47X26HLISTING OF HERESY GEOMETRY/1HO,23X12A6/1HO,7HXCOR
     1 = F10.3)
      DO 40 I=1,NTYPE
      WRITE OUTPUT TAPE NOUT,20,I
   20 FORMAT(1HO,10HROD TYPE  I3/1HO,5(8X1HX,9X1HY,4X))
      L=K+1
      M=K+NRPRT(I)
      WRITE OUTPUT TAPE NOUT,30,((X(N),Y(N)),N=L,M)
   30 FORMAT(5(4XF9.4,1XF9.4))
      K=K+NRPRT(I)
   40 CONTINUE
      RETURN
      END(2,1,0,0,0)
```

```
C           SRL HERESY 1   SUBROUTINE PUNCH    VERSION OF 10/2/64
C
      SUBROUTINE PUNCH
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL BRM
      K=5
      NX=0
      N=0
      DO 10 I=1,NTYPE
   10 NX=NX+NRPRT(I)
      LOCATE=2500
   15 IF(NX-K)40,20,20
   20 PUNCH 30,LOCATE,AA(LOCATE),AA(LOCATE+1),AA(LOCATE+2),AA(LOCATE+3),
     1AA(LOCATE+4)
   30 FORMAT(I12,5F12.6)
      LOCATE=LOCATE+5
      K=K+5
      GO TO 15
   40 K=K-5
      M=NX-K
      DO 50 I=1,5
   50 BB(I)=0.0
      DO 60 I=1,M
      MX=LOCATE-1+I
   60 BB(I)=AA(MX)
      PUNCH 30,LOCATE,(BB(I),I=1,M)
      IF(N)70,70,80
   70 LOCATE=9000
      K=5
      N=1
      GO TO 15
   80 RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE PLOT          VERSION OF 10/2/64
C
      SUBROUTINE PLOT
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EFM
      NOUT=10
      K=0
      KMAX=0
      DO 10 I=1,NTYPE
   10 KMAX=KMAX+NRPRT(I)
      LPAGE=1
      WRITE OUTPUT TAPE NOUT,20,LPAGE
   20 FORMAT(1H145X29HPLOT OF FINITE HERESY LATTICE,38X5HPAGE I2)
      WRITE OUTPUT TAPE NOUT,30,(TITLE(I),I=1,12)
   30 FORMAT(1HO,23X12A6)
      WRITE OUTPUT TAPE NOUT,40,XSCALE,YSCALE,NTYPE
   40 FORMAT(1HO,35HEACH INCREMENT ALONG X AXIS EQUALS F10.5,2X2HCM,5X
     135HEACH INCREMENT ALONG Y AXIS EQUALS F10.5,2X2HCM/1HO,5X20HNUMBER
     2 OF ROD TYPES I2)
      IF(NTYPE-25)80,80,50
   50 WRITE OUTPUT TAPE NOUT,60,(I,I=1,25)
   60 FORMAT(1HO,8X8HROD TYPE,25(2XI2))
      WRITE OUTPUT TAPE NOUT,70,(NRPRT(I),I=1,25)
   70 FORMAT(1H ,16HNO RODS/ROD TYPE,25(I4))
      LX=26
      LH=NTYPE
      GO TO 90
   80 LX=1
      LH=NTYPE
   90 WRITE OUTPUT TAPE NOUT,60,(I,I=LX,LH)
      WRITE OUTPUT TAPE NOUT,70,(NRPRT(I),I=LX,LH)
      L=1
  100 LPAGE=LPAGE+1
      WRITE OUTPUT TAPE NOUT,110,(TITLE(I),I=1,12),LPAGE
  110 FORMAT(1H1,12A6,40X5HPAGE I2)
      GO TO (120,130,140,150),L
  120 DO 125 I=1,26
      KK(I)=-26+I
  125 NN(I)=26-I
      GO TO 160
  130 DO 135 I=1,26
      KK(I)=I-26
  135 NN(I)=1-I
      GO TO 160
  140 DO 145 I=1,26
      KK(I)=I-1
  145 NN(I)=26-I
      GO TO 160
  150 DO 155 I=1,26
      KK(I)=I-1
  155 NN(I)=1-I
  160 WRITE OUTPUT TAPE NOUT,170,(KK(I),I=1,26)
  170 FORMAT(1HO,13HX INTERVALS =,26(1XI3)//12H Y INTERVALS)
      DO 330 N=1,26
      FN=FLOATF(NN(N))
      DO 180 LXY=1,26
  180 MM(LXY)=0
```

```
      DO 310 K=1,KMAX
      IF(Y(K)-(FN-.5)*YSCALE)310,190,190
  190 IF(Y(K)-(FN+.5)*YSCALE)200,200,310
  200 DO 300 I=1,26
      FI=FLOATF(KK(I))
      IF(X(K)-(FI+.5)*XSCALE)210,210,300
  210 IF(X(K)-(FI-.5)*XSCALE)300,220,220
  220 KRAX=NRPRT(1)
      DO 290 KX=1,NTYPE
      IF(K-KRAX)250,250,230
  230 KRAX=KRAX+NRPRT(KX+1)
      GO TO 290
  250 MM(I)=KX
      GO TO 310
  290 CONTINUE
  300 CONTINUE
  310 CONTINUE
      WRITE OUTPUT TAPE NOUT,320,NN(N),(MM(K),K=1,26)
  320 FORMAT(1H0,10XI3,26(1XI3))
  330 CONTINUE
      L=L+1
      IF(L-4)100,100,340
  340 RETURN
      END(2,1,0,0,0)
```

```
C          SRL HERESY 1 SUBROUTINE INGEOM          VERSION OF 10/8/64
C
      SUBROUTINE INGEOM
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EFM
      NOUT=10
      IX=1
      RSQ=RADII**2
      DO 80 I=1,NTYPE
      XDX=WDX(I)*XCOR
      YDY=WDY(I)*XCOR
      XX1=WX1(I)*XCOR
      YY1=WY1(I)*XCOR
      NRPRT(I)=0
      MAXX=RADII/XDX
      X(IX)=0.0
      DO 20 J=1,MAXX
      MAXY=(SQRTF(RSQ-X(IX)**2))/YDY
      Y(IX)=0.0
      NRPRT(I)=NRPRT(I)+1
      DO 10 K=1,MAXY
      IK=IX+K
      Y(IK)=Y(IK-1)+YDY
      X(IK)=X(IX)
   10 NRPRT(I)=NRPRT(I)+1
      KX=IX+MAXY+1
      X(KX)=X(IX)+XDX
   20 IX=KX
      IX=IX-1
      K4=IX-NRPRT(I)+1
      K2=0
      DO 40 J=K4,IX
      IF(X(J))90,40,30
   30 K2=K2+1
      K3=IX+K2
      Y(K3)=Y(J)
      X(K3)=-X(J)
   40 CONTINUE
      IX=IX+K2
      NRPRT(I)=NRPRT(I)+K2
      K2=0
      K4=IX-NRPRT(I)+1
      DO 60 J=K4,IX
      IF(Y(J))90,60,50
   50 K2=K2+1
      K3=IX+K2
      Y(K3)=-Y(J)
      X(K3)=X(J)
   60 CONTINUE
      NRPRT(I)=NRPRT(I)+K2
      IX=IX+K2
      K4=IX-NRPRT(I)+1
      DO 70 J=K4,IX
      X(J)=X(J)+XX1
   70 Y(J)=Y(J)+YY1
      IX=IX+1
   80 CONTINUE
```

```
      GO TO 105
   90 NERR=1
      WRITE OUTPUT TAPE NOUT,100
  100 FORMAT(1HO,81H A MACHINE ERROR HAS OCCURRED IN GENERATING AN INFIN
     1ITE LATTICE - RESTART PROBLEM)
      GO TO 140
  105 IX=0
      DO 110 I=1,NTYPE
  110 IX=IX+NRPRT(I)
      IF(IX-6500)140,140,120
  120 WRITE OUTPUT TAPE NOUT,130
  130 FORMAT(1HO,31HTHE NUMBER OF RODS EXCEEDS 6500)
      NERR=1
  140 RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE DATPRT        VERSION OF 10/15/64          DATP
C                                                                          DATP
      SUBROUTINE DATPRT                                                    DATP

C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 1
C
      CALL EFM                                                             DATP
      NOUT=10                                                              DATP
      WRITE OUTPUT TAPE NOUT,5,(TITLE(I),I=1,12)                           DATP
    5 FORMAT(1H1,23X12A6)                                                  DATP
      WRITE OUTPUT TAPE NOUT,10                                            DATP
   10 FORMAT(1HO,45X28HSUMMARY OF HERESY INPUT DATA/1HO,53X13HINPUT OPTI   DATP
     1ONS)                                                                 DATP
      WRITE OUTPUT TAPE NOUT,20,INF,KERNEL,KOOR,NTHERM                     DATP
   20 FORMAT(1H ,25HFINITE/INFINITE OPTION = I2,5X16HKERNEL OPTION = I2,   DATP
     15X20HCOORDINATE OPTION = I2,5X29HPURE THERMAL KERNEL OPTION = I2)    DATP
      IF(KOOR-1)62,50,62                                                   DATP
   50 WRITE OUTPUT TAPE NOUT,60,LAT,KPRNT,KPLOT,KSKIP                      DATP
   60 FORMAT(1H ,15HLATTICE TYPE = I2,5X17HGEOMETRY PRINT = I2,5X16HGEOM   DATP
     1ETRY PLOT = I2,5X17HGEOMETRY PUNCH = I2)                             DATP
   62 IF(KOOR-3)70,65,70                                                 DATPRT
   65 WRITE OUTPUT TAPE NOUT,68,NF(50)                                     DATP
   68 FORMAT(1HO,25HGEOMETRY TAKEN FROM FILE I4,32H OF PERMANENT TAPE ON   DATP
     1TAPE UNIT 3)                                                         DATP
   70 WRITE OUTPUT TAPE NOUT,80,N1,N2,N3,CR1,CR2,CR3                       DATP
   80 FORMAT(1HO,49X20HCONVERGENCE CRITERIA/1H ,30HMATRIX INVERSION ITER   DATP
     1ATIONS = I4,5X24HEIGENVALUE ITERATIONS = I5,5X32HCRITICALITY SEARC   DATP
     2H ITERATIONS = I4/1H ,24HINVERSION CONVERGENCE = F7.6,5X25HEIGENVA   DATP
     3LUE CONVERGENCE = F7.6,5X26HCRITICALITY CONVERGENCE = F7.6)          DATP
      WRITE OUTPUT TAPE NOUT,90,NOGRID,XMESH,B(1),B(2),B(3),TAU(1),TAU(2   DATP
     1),TAU(3),TAUR(1),TAUR(2),TAU(3),BSQ,XLSQ,SIGA                        DATP
   90 FORMAT(1HO,58X11HKERNEL DATA/1H ,24HNUMBER OF GRID POINTS = I5,5X    DATP
     112HMESH SIZE = F8.3,2X2HCM/1H ,10HB(1)     = F6.2,5X10HB(2)     = F6  DATP
     2.2,5X10HB(3)     = F6.2/1H ,10HTAU(1)  = F6.2,5X10HTAU(2)  = F6.2,   DATP
     35X10HTAU(3)  = F6.2/1H ,10HTAUR(1) = F6.2,5X10HTAUR(2) = F6.2,5X10   DATP
     4HTAUR(3) = F6.2/1H ,11HB SQUARE = F7.6,5X11HL SQUARE = F10.4,5X21H   DATP
     5MODERATOR SIGMA(A) = F12.8)                                          DATP
      WRITE OUTPUT TAPE NOUT,100,RADRFL,ROEMIN                             DATP
  100 FORMAT(1HO,19HREFLECTOR RADIUS = F6.2,11HRHO(MIN) = F6.2)            DATP
      IF(INF)108,108,102                                                   DATP
  102 WRITE OUTPUT TAPE NOUT,104,RADII                                     DATP
  104 FORMAT(1HO,52HRADIUS BEYOND WHICH NO MORE RODS ARE CONSIDERED =      DATP
     1F8.3)                                                                DATP
  108 IF(LCRIT)220,220,110                                                 DATP
  110 IF(LCRIT-1)120,120,140                                               DATP
  120 WRITE OUTPUT TAPE NOUT,130,EIGEN                                     DATP
  130 FORMAT(1HO,60HCRITICALITY SEARCH ON ETA - EIGENVALUE TO BE SEARCHE   DATP
     1D FOR = F10.7)                                                       DATP
      GO TO 190                                                            DATP
  140 IF(LCRIT-2)150,150,170                                               DATP
  150 WRITE OUTPUT TAPE NOUT,160,EIGEN                                     DATP
  160 FORMAT(1HO,62HCRITICALITY SEARCH ON GAMMA - EIGENVALUE TO BE SEARC   DATP
     1HED FOR = F10.7)                                                     DATP
      GO TO 190                                                            DATP
  170 WRITE OUTPUT TAPE NOUT,180,EIGEN                                     DATP
  180 FORMAT(1HO,65HCRITICALITY SEARCH ON BUCKLING - EIGENVALUE TO BE SE   DATP
     1ARCHED FOR = F10.7)                                                  DATP
```

```
190 IF(LCRIT-2)200,200,220                                              DATP
200 WRITE OUTPUT TAPE NOUT,210,KSERCH                                   DATP
210 FORMAT(1H ,29HROD KIND TO BE SEARCHED ON = I3)                      DATP
220 WRITE OUTPUT TAPE NOUT,230                                          DATP
230 FORMAT(1H0,48X22HDATA FOR EACH ROD KIND)                            DATP
    WRITE OUTRUT TAPE NOUT,240                                          DATP
240 FORMAT(1H0, 8HROD KIND,4X5HGAMMA,6X3HSMF,5X12H1./(GAM+SMF),5X3HETA  DATP
   1,5X10HRES. COEF.,5X3HLSQ,5X8HSIGMA(A),6X1HF,6X9HCELL VOL.,3X6HRADI  DATP
   2US//)                                                               DATP
    DO 250 I=1,NKIND                                                    DATP
    XXLL=1./(GAM(I)+SM(I))                                              DATP
250 WRITE OUTPUT TAPE NOUT,260,I,GAM(I),SM(I),XXLL,ETAI(I),AL(I),XLSQ2  DATP
   1(I),SIGC(I),THERU(I),VCELL(I),RADIUS(I)                             DATP
260 FORMAT(1H ,3X12,6XF7.5,3XF7.5,5XF8.6,5XF7.5,4XF8.4,4XF7.1,3XF8.7,   DATP
   13XF7.5,3XF9.5,3XF6.4)                                               DATP
    WRITE OUTPUT TAPE NOUT,270                                          DATP
270 FORMAT(1H0,48X22HDATA FOR EACH ROD TYPE)                            DATP
    WRITE OUTPUT TAPE NOUT,280                                          DATP
280 FORMAT(1H0,8HROD TYPE,3X8HROD KIND,3X8HNO. RODS,4X5HGAMMA,6X3HSMF,  DATP
   15X14H1./(GAMMA+SMF)//)                                              DATP
    DO 290 I=1,NTYPE                                                    DATP
    XXLL=1./(GAMMA(I)+SMFO(I))                                          DATP
290 WRITE OUTPUT TAPE NOUT,300,I,KIND(I),NRPRT(I),GAMMA(I),SMFO(I),     DATP
   1XXLL                                                                DATP
300 FORMAT(1H ,3X12,9XI2,7XI4,6XF7.5,3XF7.5,6XF8.6)                     DATP
    RETURN                                                              DATP
    END(2,1,0,0,0)                                                      DATP
```

Link 2

Link 2 of the program tape consists of 13 FORTRAN II subprograms, 2 SAP subprograms, a package of library function subroutines, and a package of FORTRAN II control subroutines. COMMON STORAGE begins at location 14673_8 in this link.

FORTRAN II and SAP Subprograms

Subprogram Name	Function
MAIN PROG 2	Link 2 - executive routine
XYLIB (SAP)	Library tape routine
MATELM	Slowing down matrices
INVERT	Matrix inversion
EIGEN1	Eigenvalue power iteration
EDIT	Edit solution and output result
HEAD	Write heading for search summary
COUT	Output search summary
CONVER	Test for criticality convergence
ETASER	Eta search routine
GAMSER	Gamma search routine
BUCK	Buckling search routine
KERGEN	Kernel generator
LEQ(SAP)	Linear equation program
USKER	Bessel's interpolation routine

The following are 10 small matrix routines used in code:

ABMXVC	Finds max. value of a column vector
DGMPMX	Diagonal mx. multiplied by a square mx.
MXADDG	Square mx. added to a diagonal mx.
MXMPCV	Square mx. multiplied by a column vector
MXMPDG	Square mx. multiplied by diagonal mx.
MXMPMX	Square mx. multiplied by square mx.
MXSBMX	Square mx. subtracted from square mx.
RVMPCV	Row vector multipled by column vector
RVMPDG	Row vector multipled by diagonal mx.
RVMPMX	Row vector multipled by square mx.

FORTRAN II Library Function Subroutines

FORTRAN Name	Function
EXPF (arg)	$e^{(arg)}$
LOG (arg)	$\ln(arg)$
EXPIF (arg)	$E_i(arg)$ integral
SQRTF (arg)	$\sqrt{arg}$

FORTRAN II Control Subroutines

Subroutine Name	Operation
XLOCF (arg)	Locate variable in storage
LFM	Leave floating point trap
EFM	Enter floating point trap
(FIL)	Format interpretive loader
(DBC)	Decimal to binary conversion
(BDC)	Binary to decimal conversion
(IOH)O	Input - Output Hollerith, Output
(IOH)I	Input - Output Hollerith, Input
(RTN)	Return
(LEV)	Leave
(STH)	Storage to tape Hollerith
(SPH)	Storage to printer Hollerith
CHAIN	Chain search routine
CPTG	CHAIN program tape generator

```
C      DIMENSION DECK FOR SRL HERESY 1 MAIN PROGRAM 2
C
C           DIMENSION DECK FOR MAIN PROGRAM 2      VERSION OF 10/2/64
C
       DIMENSION LL(225),AA(15500),AAA(10250)
       DIMENSION NICPRT(50),NRPRT(50),KIND(50),NF(50)
       DIMENSION B(3),TAU(3),TAUR(3),XLSQ2(50),SIGC(50),VCELL(50),GAM(5
      1     ,THERU(50),ETAI(50),AL(50),RADIUS(50),XO(50,3),YO(50,3),
      2   THETA(50,3),WDX(50),WDY(50),WX1(50),WY1(50),VO(50),W(8),
      3     TITLE(12),BGF(250),BGFR(250),SMGR(250),SMF(250),SM(50),D(50)
      4     GAMMA(50),ETA(50),A(50),SMFO(50),FKMAX(50),X(6500),Y(6500)
       DIMENSION T1(50,50),T2(50,50),T3(50,50),T4(50,50),T5(50),T6(50),
      1     T(50,50),TR(50,50),SR(50,50),Z(50,50),ABSRB(50),RESABS(50),
      2     VRATIO(50),V1(50),V2(50)
C
       COMMON LL,AA,AAA
C
C           EQUIVALENCES FOR LL ARRAY
C
       EQUIVALENCE (INF,LL(1)),(KERNEL,LL(2)),(KOOR,LL(3)),(N1,LL(4)),
      1     (N2,LL(5)),(N3,LL(6)),(NOGRID,LL(7)),(NTHER,LL(8)),(LAT,
      2     LL(9)),(NICPRT,LL(10)),(KPRNT,LL(60)),(KPLOT,LL(61)),(KSKIP,
      3     LL(62)),(NTYPE,LL(63)),(NKIND,LL(64)),(NRPRT,LL(65)),(KIND,
      4     LL(115)),(LCRIT,LL(165)),(KSERCH,LL(166)),(NSEPR,LL(167)),
      5     (KOAT,LL(168)),(KTEST,LL(169)),(IT1,LL(170)),(IT2,LL(171)),
      6     (IT3,LL(172)),(IT4,LL(173)),(IMAX1,LL(174)),(IMAX2,LL(175)),
      7(NF,LL(176))
C
C           EQUIVALENCES FOR AA ARRAY
C
       EQUIVALENCE (CR1,AA(1)),(CR2,AA(2)),(CR3,AA(3)),(XCOR,AA(4)),
      1     (XSCALE,AA(5)),(YSCALE,AA(6)),(RADREL,AA(7)),(ROEMIN,AA(8)),
      2     (EIGEN,AA(9)),(XMESH,AA(10)),(BSQ,AA(11)),(XLSQ,AA(12)),
      3     (SIGA,AA(13)),(B,AA(14)),(TAU,AA(17)),(TAUR,AA(20)),(TAUTOL,
      4     AA(23)),(NERR,AA(24)),(XLSQ2,AA(25)),(SIGC,AA(75)),(VCELL,
      5     AA(125)),(GAM,AA(175)),(THERU,AA(225)),(ETAI,AA(275)),(AL,
      6     AA(325)),(RADIUS,AA(375)),(XO,AA(425)),(YO,AA(575)),(THETA,
      7     AA( 725)),(WDX,AA( 875)),(WDY,AA( 925)),(WX1,AA( 975)),(WY1,
      8     AA(1025)),(RADII,AA(1075)),(A1,AA(1076)),(A2,AA(1077)),(FK1,
      9     AA(1078)),(FK2,AA(1079)),(VO,AA(1080)),(W,AA(2492))
       EQUIVALENCE (BGF,AA(1142)),(BGFR,AA(1392) ),(SMGR, AA(1642)),
      1     (SMF,AA(1892)),(SM,AA(2142)),(D,AA(2192)),(GAMMA,AA(2242)),
      2     (ETA,AA(2292)),(A,AA(2342)),(SMFO,AA(2392)),(FKMAX,AA(2442))
      3     (X,AA(2500)),(Y,AA(9000)),(TITLE,AA(1130)),(T,AA(2500)),
      4     (TR,AA(5000)),(SR,AA(7500)),(Z,AA(10000)),(T5,AA(12500)),
      5     (T6,AA(12550))
C
C           EQUIVALENCES FOR AAA ARRAY
C
       EQUIVALENCE (T1,AAA(1)),(T2,AAA(2501)),(T3,AAA(5001)),(T4,
      1     AAA(7501)),(ABSRB,AAA(10001)),(RESABS,AAA(10051)),(VRATIO,
      2     AAA(10101)),(V1,AAA(10151)),(V2,AAA(10201))
C
```

```
C          SRL-HERESY 1   MAIN PROGRAM 2      VERSION OF 10/19/64          MAI
C                                                                           MAI
C                                                                           MAI
      SUBROUTINE MAIN                                                      DIME
C
      INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C                                                                           MAI
C                                                                           MAI
      CALL EFM                                                              MAI
      NOUT=10                                                               MAI
      NTEMP=2                                                               MAI
      REWIND NTEMP                                                          MAI
      IT4=0
      IF(KOOR)4,4,2
    2 GO TO (8,8,5),KOOR                                                    MAI
    4 IF(KERNEL)45,45,5
    5 CALL XYLIB(NF(49),NF(50),0,XCOR)                                      M.
    8 CALL MATELM                                                           MAI
      CALL XYLIB(NF(49),NF(50),1,XCOR)                                      M.
      DO 10 I=1,NTYPE                                                       MAI
      DO 10 J=1,NTYPE                                                       MAI
      T(I,J)=T1(I,J)                                                        MAI
      TR(I,J)=T2(I,J)                                                       MAI
      SR(I,J)=T3(I,J)                                                       MAI
   10 Z(I,J)=T4(I,J)                                                        MAI
   45 CALL MXADDG(NTYPE,Z,GAMMA,T1)                                         MAI
      CALL INVERT                                                           MAI
  220 CALL DGMPMX(NTYPE,A,SR,T2)                                            MAI
      CALL MXMPMX(NTYPE,TR,T2,T4)                                           MAI
      CALL MXSBMX(NTYPE,T,T4,T1)                                            MAI
      CALL MXMPDG(NTYPE,T1,ETA,T4)                                          MAI
      CALL MXMPMX(NTYPE,T3,T4,T1)                                           MAI
      CALL EIGEN1                                                           MAI
  360 DO 370 I=1,NTYPE                                                      MAI
  370 VO(I)=V2(I)                                                           MAI
      IF(LCRIT)490,490,380                                                  MAI
  380 IF(NSEPR)400,400,390                                                  MAI
  390 CALL EDIT                                                             MAI
      GO TO 430                                                             MAI
  400 IF(IT4)420,410,420                                                    MAI
  410 CALL HEAD                                                             MAI
      GO TO 430                                                             MAI
  420 CALL COUT                                                             MAI
  430 CALL CONVER                                                           MAI
      NXNXNX=IT1                                                            M.
      GO TO (435,480,490),NXNXNX
  435 IF(LCRIT-1)480,440,450                                                MAI
  440 CALL ETASER                                                           MAI
      GO TO 220                                                             MAI
  450 IF(LCRIT-2)430,460,470                                                MAI
  460 CALL GAMSER                                                           MAI
      GO TO 45                                                              MAI
  470 CALL BUCK                                                             MAI
      GO TO 5                                                               MAI
  480 IF(NSEPR)490,490,500                                                  MAI
  490 CALL EDIT                                                             MAI
  500 IF(KTEST)510,510,520                                                  MAI
  510 CALL CHAIN(1,1)                                                       MAI
  520 CALL CHAIN(1,50)                                                      MAI
      END(2,1,0,0,0)
```

```
      REM FORTRAN II SUBROUTINE XYLIB          VERSION OF 10/21/64    XYLB
      REM                                                             XYLB
      REM TAPE CONTROL SUBROUTINE FOR SRL-HERESY I GEOMETRY TAPES     XYLB
      REM                                                             XYLB
      FUL                      P  C  G                                XYLB
      ORG 0                    R  A  E                                XYLB
      MZE *,,ENTRY-2           O  R  N                                XYLB
      PZE                      G  D  E                                XYLB
      PZE BREAK,,0             R     R                                XYLB
      PZE 32562                A     A                                XYLB
      BCO 1XYLIB               M     T                                XYLB
      PZE EP                         O                                XYLB
ENTRY EQU *                          R                                XYLB
      ORG 0                                                           XYLB
      REL                                                             XYLB
STORE PZE                                                             XYLB
      PZE                                                             XYLB
      PZE                                                             XYLB
      PZE 0
 FILE PZE 0
      PZE 0
SCALE PZE 0
   EP SXD STORE,1              STORE INDEX REGS.                      XYLB
      SXD STORE+1,2                                                   XYLB
      SXD STORE+2,4                                                   XYLB
      SXD EOF,2                                                       XYLB
      TXI **+1,2,1                                                    XYLB
      CLA 1,4
      STA **+1
      CLA **
      STO STORE+3              STORE WRITE INDICATOR
      CLA 2,4
      STA **+1
      CLA **
      STO STORE+4              STORE FILE NUMBER
      CLA 3,4
      STA **+1
      CLA **
      STO STORE+5              STORE READ-WRITE INDICATOR
      CLA 4,4
      STA **+1
      CLA **
      STO STORE+6              STORE SCALE FACTOR
      CLA STORE+4             GET FILE NUMBER
      TNZ RDID                 BRANCH ON NON ZERO TO LIBRARY TAPE     XYLB
      CLA STORE+5             GET READ-WRITE INDICATOR
      TNZ WRITE                BRANCH ON NON ZERO TO WRITE ROUTINE    XYLB
      CLA ZERO                                                        XYLB
      LXA ONE,1                                                       XYLB
      RTB 2                    READ BINARY TAPE 2                     XYLB
      CAD 29839,1                                                     XYLB
      TRA **+3                                                        XYLB
      TRA ERROR                TRANSFER TO HPR 550                    XYLB
      TRA ERROR                TRANSFER TO HPR 550                    XYLB
      TIX *-4,1,1                                                     XYLB
      SLW CHECK
      CLA CHECK
      CPY CHECK                                                       XYLB
      SUB CHECK                                                       XYLB
```

```
       TZE  *+2                                                    XYLB
       HPR  *          CHECKSUM ERROR - HPR 424                   XYLB
       REW  2          REWIND TAPE 2                              XYLB
       TRA  RETN       TRANSFER TO EXIT                           XYLB
WRITE  LXA  ONE,1                                                 XYLB
       CLA  ZERO                                                  XYLB
       WTB  2           WRITE BINARY TAPE 2                       XYLB
       CAD  29839,1                                               XYLB
       TIX  *-1,1,1                                               XYLB
       SLW  CHECK
       CPY  CHECK                                                 XYLB
       REW  2          REWIND TAPE 2                              XYLB
       TRA  RETN       TRANSFER TO EXIT                           XYLB
 RDID  RTB  3          BEGIN FILE SEARCH ON TAPE 3                XYLB
       CPY  FILE1                                                 XYLB
       TRA  *+3                                                   XYLB
       TRA  TEST-1     TRANSFER TO MAKE NEW TAPE
       TRA  TEST-1
       CLA  FILE                                                  XYLB
       SUB  FILE1                                                 XYLB
       TZE  RIGHT      FILE IS CORRECT
       CLA  FOE        FILE IS NOT CORRECT
       SUB  FILE1
       TNZ  RDID       TRANSFER TO READ NEW RECORD
  EOF  TIX  REWTP,2,0  REWIND TAPE ONCE ON FOE MARK
       TRA  TEST       FILE NOT ON TAPE,TRANSFER TO TEST
RIGHT  CLA  STORE+5    FILE IS CORRECT,GET READ-WRITE INDICATOR
       TZE  RDTP       TRANSFER TO TAPE 3 READ                    XYLB
       CLA  STORE+3    GET NEW COORDINATE INDICATOR
       TNZ  *+2        TRANSFER TO NEW COORDINATE WRITE           XYLB
       TRA  RETN       TRANSFER TO EXIT                           XYLB
       CLA  ZERO                                                  XYLB
       LXA  ONE,1                                                 XYLB
       WTB  3          WRITE NEW COORDINATES IN OLD FILE
       CAD  29839,1                                               XYLB
       TIX  *-1,1,1                                               XYLB
       SLW  CHECK
       CPY  CHECK                                                 XYLB
       TRA  RETN                                                  XYLB
 RDTP  LXA  ONE,1      READ FILE FROM TAPE 3                      XYLB
       CLA  ZERO                                                  XYLB
       RTB  3          READ COORDINATES FROM OLD FILE             XYLB
       CAD  29839,1                                               XYLB
       TRA  *+3                                                   XYLB
       TRA  ERROR      TRANSFER TO HPR 550
       TRA  ERROR      TRANSFER TO HPR 550
       TIX  *-4,1,1                                               XYLB
       SLW  CHECK
       CLA  CHECK
       CPY  CHECK                                                 XYLB
       SUB  CHECK                                                 XYLB
       TZE  *+2                                                   XYLB
       HPR  *          CHECKSUM ERROR - HPR 507                   XYLB
       LXD  ONE,1                                                 XYLB
       EFM                                                        XYLB
       LDQ  SCALE      PLACE SCALE FACTOR IN MQ                   XYLB
       FMP  29839,1    FLOATING MULTIPLY EACH COORDINATE VALUE    XYLB
       STO  29839,1    STORE BACK IN COORDINATE ARRAY             XYLB
       TIX  *-3,1,1                                               XYLB
```

```
       LFM                                                          XYLB
       TRA RETN                                                     XYLB
REWTP  REW 3                  REWIND TAPE 3                         XYLB
       TRA RDID                                                     XYLB
       REW 3                  TAPE 3 REWIND FOR NEW TAPE
 TEST  CLA STORE+3
       TNZ *+5                TRANSFER TO WRITE NEW COORDINATE FILE XYLB
       LXD STORE,1                                                  XYLB
       LXD STORE+1,2                                                XYLB
       LXD STORE+2,4                                                XYLB
       HTR 5,4                ERROR - FILE IS NOT ON TAPE  HTR 5,4  XYLB
       BST 3                  BACKSPACE TAPE 3 ONE RECORD
       WTB 3                                                        XYLB
       CPY FILE               WRITE NEW FILE NUMBER                 XYLB
       CLA ZERO                                                     XYLB
       LXA ONE,1                                                    XYLB
       WTB 3                  WRITE NEW COORDINATES                 XYLB1
       CAD 29839,1                                                  XYLB1
       TIX *-1,1,1                                                  XYLB1
       SLW CHECK
       CPY CHECK              WRITE CHECKSUM ON TAPE 3
       WTB 3
       CPY FOE                WRITE FOE MARK ON TAPE 3
       WEF 3                  END FILE TAPE 3                       XYLB1
       REW 3
       TRA RETN                                                     XYLB1
ERROR  HPR *                  READ ERROR ON TAPE 2 OR 3   HPR 550
 RETN  LXD STORE,1                                                  XYLB1
       LXD STORE+1,2                                                XYLB1
       LXD STORE+2,4                                                XYLB1
       TRA 5,4                EXIT TO MAIN PROGRAM                  XYLB1
  ONE  PZE 13000                                                    XYLB1
FILE1  PZE 0                                                        XYLB1
 ZERO  PZE 0                                                        XYLB1
CHECK  PZE 0                                                        XYLB1
  FOE  PZE 64625
BREAK  EQU *                                                        XYLB1
       END **                                                       XYLB1
```

```
C       SRL HERESY 1 SUBROUTINE MATELM          VERSION OF 10/2/64
C
      SUBROUTINE MATELM
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      PI=3.1415926
      KMN=1
      DO 110 I=1,NTYPE
      N=NRPRT(I)
      KR=1
      KMNN=KMN+N-1
      DO 100 J=1,NTYPE
      T1(J,I)=0.0
      T2(J,I)=0.0
      T3(J,I)=0.0
      T4(J,I)=0.0
      IF(RADRFL)10,10,20
   10 IRL=0
      GO TO 40
   20 ROERL=SQRTF(X(KMN)**2+Y(KMN)**2)
      IF(ROERL-ROEMIN)10,10,30
   30 ROEIM=2.*RADRFL-ROERL
      THET=ROEIM/ROERL
      IRL=1
   40 DO 90 K=KMN,KMNN
      IF(KR-K)60,50,60
   50 T1(J,I)=T1(J,I)+BGF(1)
      T2(J,I)=T2(J,I)+BGFR(1)
      T3(J,I)=T3(J,I)+SMGR(1)
      T4(J,I)=T4(J,I)+SMFO(I)
      GO TO 70
   60 DX=SQRTF((X(KR)-X(K))**2+(Y(KR)-Y(K))**2)
      FF1=1.0
      CALL USKER(DX,FF1,I,J)
   70 IF(IRL-1)90,80,90
   80 XIMAG=THET*X(K)
      YIMAG=THET*Y(K)
      DX=SQRTF((X(KR)-XIMAG)**2+(Y(KR)-YIMAG)**2)
      FF1=-SQRTF(ROEIM/ROERL)
      CALL USKER(DX,FF1,I,J)
   90 CONTINUE
  100 KR=KR+NRPRT(J)
  110 KMN=KMN+N
      RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE USKER          VERSION OF 10/2/64
C
      SUBROUTINE USKER(DX,FIMREL,I,J)
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EPM
      NTEMP=2
      NOUT=10
      FNOGR=NOGRID
      IF(DX-XMESH*(FNOGR-3.))30,30,10
   10 WRITE OUTPUT TAPE NOUT,20
   20 FORMAT(1H0,55HERROR - KERNEL SIZE IS TOO SMALL FOR LATTICE BEING U.
     1SED)
      GO TO 70
   30 M=DX/XMESH+1.0
      IF(M-2)40,60,60
   40 WRITE OUTPUT TAPE NOUT,50
   50 FORMAT(1H0,32HERROR - KERNEL MESH IS TOO LARGE)
      GO TO 70
   60 U=(MODF(DX,XMESH))/XMESH
      P=U*(U-1.)/4.
      Q=1.-U-P
      R=U-P
      F1F=FIMREL*(P*(BGF(M-1)+BGF(M+2))+Q*BGF(M)+R*BGF(M+1))
      T1(J,I)=T1(J,I)+F1F
      F2F=FIMREL*(P*(BGFR(M-1)+BGFR(M+2))+Q*BGFR(M)+R*BGFR(M+1))
      T2(J,I)=T2(J,I)+F2F
      F3F=FIMREL*(P*(SMGR(M-1)+SMGR(M+2))+Q*SMGR(M)+R*SMGR(M+1))
      T3(J,I)=T3(J,I)+F3F
      F4F=FIMREL*(P*(SMF(M-1)+SMF(M+2))+Q*SMF(M)+R*SMF(M+1))
      T4(J,I)=T4(J,I)+F4F
      RETURN
   70 WRITE TAPE NTEMP,(AA(I),I=2500,15500)
      REWIND NTEMP
      WRITE OUTPUT TAPE NOUT,80
   80 FORMAT(1H ,70HTHIS ERROR OCCURS IN FORMING THE MATRICES FROM THE K
     1ERNEL AND GEOMETRY/1H ,84HAN ATTEMPT WILL BE MADE TO RUN SUCCESSIV
     2E PROBLEMS, BUT IT IS UNLIKELY THEY WILL RUN)
      PRINT 90
   90 FORMAT(1H ,26HPROBLEM DID NOT RUN TO END)
      CALL CHAIN (1,1)
      END(2,1,0,0,0)
```

```
C          SRL HERESY 1  SUBROUTINE INVERT   VERSION OF 10/2/64
C
      SUBROUTINE INVERT
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      NOUT = 10
      IT1=0
      NXN=0
      DO 70 I=1,NTYPE
      DO 70 J=1,NTYPE
      T2(I,J)=T1(I,J)
      IF(I-J)60,50,60
   50 T3(I,J)=1.0
      GO TO 70
   60 T3(I,J)=0.0
   70 CONTINUE
      KIAX=XLOCF(T2(1,1))-XLOCF(T2(1,2))
      CALL LEQ(T2,T3,NTYPE,NTYPE,KIAX,KIAX,DET)
   75 CALL MXMPMX(NTYPE,T1,T3,T4)
      DO 100 I=1,NTYPE
      DO 100 J=1,NTYPE
      IF(I-J)90,80,90
   80 IF(ABSF(T4(I,J)-1.0)-CR1)100,100,110
   90 IF(ABSF(T4(I,J))-CR1)100,100,110
  100 CONTINUE
      RETURN
  110 IT1=IT1+1
      IF(IT1-N1)180,180,120
  120 IF(NXN)125,125,170
  125 WRITE OUTPUT TAPE NOUT,130,IT1,CR1
  130 FORMAT(1H1,72HTHE SPECIFIED NUMBER OF ITERATIONS ON MATRIX INVERSI
     1ON HAS BEEN EXCEEDED/1H ,26HNO ITERATIONS COMPLETED = I4,5X24HCONV
     2ERGENCE CRITERION = F10.7/1H0,29HMATRIX TO BE INVERTED FOLLOWS)
      DO 140 I=1,NTYPE
  140 WRITE OUTPUT TAPE NOUT,150,(T1(I,J),J=1,NTYPE)
  150 FORMAT(1H0,8E14.7/(1H ,8E14.7))
      N1=N1+10
  160 NXN=1
      GO TO 180
  170 PRINT 175
  175 FORMAT(1H ,26HPROBLEM DID NOT RUN TO END)
      CALL CHAIN(1,1)
  180 DO 210 I=1,NTYPE
      DO 210 J=1,NTYPE
      T2(I,J)=T3(I,J)
      IF(I-J)200,190,200
  190 T4(I,J)=2.-T4(I,J)
      GO TO 210
  200 T4(I,J)=-T4(I,J)
  210 CONTINUE
      CALL MXMPMX(NTYPE,T2,T4,T3)
      GO TO 75
      END(2,1,0,0,0)
```

```
C          SRL HERESY 1  SUBROUTINE EIGEN      VERSION OF 10/2/64
C
      SUBROUTINE EIGEN1
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EPM
      NOUT=10
      NN=2
      DO 230 I=1,NTYPE
  230 V1(I)=V0(I)
      IT3=0
      IT2=0
      CALL ABMXVC(NTYPE,V1,A1,IMAX1)
      FK2=1.0
  240 CALL MXMPCV(NTYPE,T1,V1,V2)
      CALL ABMXVC(NTYPE,V2,A2,IMAX2)
      IF(V1(IMAX2))250,320,250
  250 FK2=V2(IMAX2)/V1(IMAX2)
      GO TO (260,300),NN
  260 IT3=IT3+1
      IF(IT3-10)300,300,270
  270 IT3=0
      DO 280 I=1,NTYPE
  280 VRATIO(I)=V2(I)/A2
      WRITE OUTPUT TAPE NOUT,290,IT2,FK2,((V2(I),VRATIO(I)),I=1,NTYPE)
  290 FORMAT(1H1,102HTHIS PROBLEM HAS EXCEEDED THE MAXIMUM NUMBER OF ITE
     1RATIONS SPECIFIED FOR THE EIGENVALUE ITERATION (N2)/1HO,27HCURRENT
     2 ITERATION NUMBER = I4,5X21HCURRENT EIGENVALUE = F10.7/1HO,10X11HE
     3IGENVECTOR,9X5HV2/A2/(1H ,10XE12.5,5XE12.4))
      IF(IT2-2*N2)300,300,292
  292 PRINT 295
  295 FORMAT(1H ,26HPROBLEM DID NOT RUN TO END)
      CALL CHAIN(1,1)
  300 DO 310 I=1,NTYPE
      IF(ABSF(V1(I)-V2(I)/FK2)-CR2*A1)310,310,320
  310 CONTINUE
      RETURN
  320 IT2=IT2+1
      IF(IT2-N2)340,340,330
  330 NN=1
  340 DO 350 I=1,NTYPE
  350 V1(I)=V2(I)
      A1=A2
      IMAX1=IMAX2
      FK1=FK2
      GO TO 240
      END(2,1,0,0,0)
```

```
C       SRL-HERESY 1 SUBROUTINE HEAD      VERSION OF 10/2/64
C
      SUBROUTINE HEAD
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      NOUT=10
      WRITE OUTPUT TAPE NOUT,10,EIGEN
   10 FORMAT(1HO,29HSUMMARY OF CRITICALITY SEARCH/1H ,32HEIGENVALUE BEIN
     1G SEARCHED FOR = F10.7)
      IF(LCRIT-1)20,20,40
   20 WRITE OUTPUT TAPE NOUT,30
   30 FORMAT(1H ,12HITERATION NO,5X18HCURRENT EIGENVALUE,5X19HPREVIOUS E
     1IGENVALUE,5X17HCURRENT ETA VALUE,5X26HROD KIND BEING SEARCHED ON)
      GO TO 90
   40 IF(LCRIT-2)50,50,70
   50 WRITE OUTPUT TAPE NOUT,60
   60 FORMAT(1H ,12HITERATION NO,5X18HCURRENT EIGENVALUE,5X19HPREVIOUS E
     1IGENVALUE,5X19HCURRENT GAMMA VALUE,5X26HROD KIND BEING SEARCHED ON
     2)
      GO TO 90
   70 WRITE OUTPUT TAPE NOUT,80
   80 FORMAT(1H ,12HITERATION NO,5X18HCURRENT EIGENVALUE,5X19HPREVIOUS E
     1IGENVALUE,5X22HCURRENT BUCKLING VALUE)
   90 RETURN
      END(2,1,0,0,0)
```

```
C       SRL-HERESY 1 SUBROUTINE COUT      VERSION OF 10/8/64
C
C      SUBROUTINE COUT
C
C      INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
C
       CALL EFM
       NOUT=10
       KSERCH=KSERCH
       W(1)=W(1)
       FFK1=W(1)
       FFK2=FK2*TAUTOL
       IF(LCRIT-1)10,10,30
   10  WRITE OUTPUT TAPE NOUT,20,IT4,FFK2,FFK1,ETAI(KSERCH),KSERCH
   20  FORMAT(1H ,5XI2,14XF10.7,13XF11.7,13XF9.6,21XI2)
       GO TO 80
   30  IF(LCRIT-2)40,40,60
   40  WRITE OUTPUT TAPE NOUT,50,IT4,FFK2,FFK1,GAM(KSERCH),KSERCH
   50  FORMAT(1H ,5XI2,14XF10.7,13XF11.7,14XF9.4,22XI2)
       GO TO 80
   60  WRITE OUTPUT TAPE NOUT,70,IT4,FFK2,FFK1,BSQ
   70  FORMAT(1H ,5XI2,14XF10.7,13XF11.7,15XF10.8)
   80  RETURN
       END(2,1,0,0,0)
```

```
C          SRL-HERESY  1  SUBROUTINE CONVER      VERSION OF 10/2/64
C
      SUBROUTINE CONVER
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EPM
      NOUT=10
      IF(ABSF(EIGEN-FK2*TAUTOL)-CR3)10,10,20
   10 IT1=2
      RETURN
   20 IF(IT4-N3)50,50,30
   30 WRITE OUTPUT TAPE NOUT,40
   40 FORMAT(1H ,84HTHE MAXIMUM NUMBER OF ITERATIONS HAVE BEEN EXCEEDED
     1IN THE SEARCH WITHOUT CONVERGING/1H ,61HTHIS PROBLEM WILL BE SKIPP
     2ED OVER AND SUCCESSIVE PROBLEMS RUN)
      IT1=3
      RETURN
   50 IT1=1
      RETURN
      END(2,1,0,0,0)
```

```
C          SRL-HERESY 1     SUBROUTINE GAMSER II    VERSION OF 10/2/64
C
      SUBROUTINE GAMSER
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
      CALL EFM
      KSERCH=KSERCH
      IF(IT4)10,10,30
   10 IF(NF(KSERCH))11,11,12
   11 XLIMIT=0.0
      GO TO 13
   12 XLIMIT=(1./(12.56637*D(KSERCH)))*LOGF(XLSQ2(KSERCH)/((1.+BSQ*
     1XLSQ2(KSERCH))*VCELL(KSERCH)))+.228714/D(KSERCH)
   13 GAMX=1./(GAM(KSERCH)+SM(KSERCH))
      W(2)=FK2
      FK=(EIGEN-FK2*TAUTOL)/EIGEN
      GAM1=GAMX*(1.-FX)
      XXXX=1./GAM1
      IF(XXXX-XLIMIT)50,50,15
   15 GAM(KSERCH)=1./GAM1-SM(KSERCH)
      DO 20 I=1,NTYPE
      N=KIND(I)
   20 GAMMA(I)=GAM(N)
      IT4=IT4+1
      W(1)=FK2*TAUTOL
      RETURN
   30 GAM2=1./(GAM(KSERCH)+SM(KSERCH))
      GAM1=GAM2+(GAM2-GAMX)*(EIGEN-FK2*TAUTOL)/((FK2-W(2))*TAUTOL)
      XXXX=1./GAM1
      IF(XXXX-XLIMIT)50,50,35
   35 GAM(KSERCH)=1./GAM1-SM(KSERCH)
      W(2)=FK2
      GAMX=GAM2
      DO 40 I=1,NTYPE
      N=KIND(I)
   40 GAMMA(I)=GAM(N)
      IT4=IT4+1
      W(1)=FK2*TAUTOL
      RETURN
   50 GAM1=XLIMIT+.0001
      IF(IT4)15,15,35
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE ETASER          VERSION OF 10/2/64
C
      SUBROUTINE ETASER
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      KSERCH=KSERCH
      IF(IT4)10,10,30
   10 ETAIX=ETAI(KSERCH)
      W(3)=FK2
      W(1)=FK2*TAUTOL
      FX=(EIGEN-FK2*TAUTOL)/EIGEN
      ETAI(KSERCH)=ETAI(KSERCH)*(1.+FX)
      DO 20 I=1,NTYPE
      N=KIND(I)
   20 ETA(I)=ETAI(N)
      IT4=IT4+1
      RETURN
   30 ETMX=ETAI(KSERCH)
      ETAI(KSERCH)=ETMX+(EIGEN-FK2*TAUTOL)*(ETMX-ETAIX)/((FK2-W(3))*
     1TAUTOL)
      ETAIX=ETMX
      W(3)=FK2
      DO 40 I=1,NTYPE
      N=KIND(I)
   40 ETA(I)=ETAI(N)
      IT4=IT4+1
      W(1)=FK2*TAUTOL
      RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY I SUBROUTINE BUCK          VERSIO OF 10/8/64
C
      SUBROUTINE BUCK
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      IF(IT4)10,10,40
   10 BSQ2=BSQ
      IF(FK2*TAUTOL-EIGEN)20,90,30
   20 BSQ=BSQ-2.5E-04
      NUM=-1
      GO TO 80
   30 BSQ=BSQ+2.5E-04
      NUM=+1
      GO TO 80
   40 BSQ2=BSQ
      IF(FK2*TAUTOL-EIGEN)50,90,60
   50 IF(NUM)20,70,70
   60 IF(NUM)70,70,30
   70 BSQ=BSQ2+(BSQ2-BSQ1)*(FK2*TAUTOL-EIGEN)/(FFK1-FK2*TAUTOL)
      NUM=0
   80 FFK1=FK2*TAUTOL
      TAUTOL=EXPF(-BSQ*(B(1)*TAU(1)+B(2)*TAU(2)+B(3)*TAU(3)))
      CALL KERGEN
      IT4=IT4+1
      BSQ1=BSQ2
      W(1)=FFK1
      DO 100 I=1,NKIND
      D(I)=XLSQ2(I)*SIGC(I)
      XYLSQ=XLSQ2(I)/(1.+BSQ*XLSQ2(I))
      IF(NF(I))100,100,95
   95 GAM(I)=XLSQ2(I)*(1.-THERU(I))/(D(I)*VCELL(I)*THERU(I))+
     1(1./(12.56637*D(I)))*LOGF(XYLSQ/VCELL(I))+.228714/D(I)-SM(I)
  100 CONTINUE
      DO 110 I=1,NTYPE
      NX=KIND(I)
      GAMMA(I)=GAM(NX)
  110 CONTINUE
   90 RETURN
      END(2,1,0,0,0)
```

```
C       SRL HERESY 1 SUBROUTINE KERGEN   VERSION OF 10/2/64
C
      SUBROUTINE KERGEN
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C
      CALL EFM
      FIZCON(1)=.0045813
      FIZCON(2)=.0360768
      FIZCON(3)=.2659732
      FIZCON(4)=1.2067492
      FIZCON(5)=3.0899424
      FIZCON(6)=3.5156229
      FIZCON(7)=1.0
      FKZCN1(1)=.0000074
      FKZCN1(2)=.0001075
      FKZCN1(3)=.00262698
      FKZCN1(4)=.0348859
      FKZCN1(5)=.23069756
      FKZCN1(6)=.42278420
      FKZCN1(7)=-.57721566
      FKZCN2(1)=.00053208
      FKZCN2(2)=-.0025154
      FKZCN2(3)=.00587872
      FKZCN2(4)=-.01062446
      FKZCN2(5)=.02189568
      FKZCN2(6)=-.07832358
      FKZCN2(7)=1.25331414
      DENL=1.+BSQ*XLSQ
      XXLSQ=XLSQ/DENL
      SIGMA=SIGA*DENL
      ZLSQ=SQRTF(XXLSQ)
      CC=1./(6.2831852*XXLSQ*SIGMA)
      IF(ABSF(B(2)))260,20,10
   10 IF(ABSF(B(3)))260,30,40
   20 NN=1
      GO TO 50
   30 NN=2
      GO TO 50
   40 NN=3
   50 DO 310 I=1,NOGRID
      FJM1=I-1
      DX=FJM1*XMESH
      ARG=DX/ZLSQ
      IF(ARG)260,60,70
   60 IFPATH=3
      GO TO 140
   70 IFPATH=1
      FKZER=0.0
      IF(ARG-2.)80,80,120
   80 FIZARG=(ARG/3.75)**2
      FIZER=0.0
      DO 90 J=1,7
   90 FIZER=FIZER*FIZARG+FIZCON(J)
  100 FKZARG=(ARG/2.)**2
      DO 110 J=1,7
  110 FKZER=FKZER*FKZARG+FKZCN1(J)
      FKZER=FKZER-LOGF(.5*ARG)*FIZER
      GO TO 140
```

```
120 FKZARG=2./ARG
    DO 130 J=1,7
130 FKZER=FKZER*FKZARG+FKZCN2(J)
    FKZER=FKZER/(SQRTF(ARG)*EXPF(ARG))
140 SMGR(I)=0.0
    D2BY4=DX**2/4.
    DO 150 J=1,NN
150 SMGR(I)=SMGR(I)+B(J)*EXPF(-D2BY4/TAUR(J))/(TAUR(J)*12.5663706)
    DO 160 J=1,NN
160 WTAU(J)=TAU(J)
170 WBG=0.0
    GO TO (210,210,180,180),IFPATH
180 DO 190 J=1,NN
190 WBG=WBG+(B(J)/2.)*CC*EXPF(WTAU(J)/XXLSQ)*(-EXPIF(-WTAU(J)/XXLSQ))
    GO TO (260,260,200,280),IFPATH
200 BGF(I)=WBG
    IFPATH=4
    GO TO 240
210 DO 220 J=1,NN
220 WBG=WBG+CC*EXPF(WTAU(J)/XXLSQ)*B(J)*(FKZER+(WTAU(J)/2./XXLSQ)*
   1EXPF(-D2BY4/WTAU(J))+(.5+D2BY4/(2.*XXLSQ))*EXPIF(-D2BY4/WTAU(J)))
    GO TO (230,280,260,260),IFPATH
230 BGF(I)=WBG
    IFPATH=2
240 DO 250 J=1,NN
250 WTAU(J)=TAU(J)-TAUR(J)
    GO TO (260,170,260,170),IFPATH
260 PRINT 270
270 FORMAT(1H0,54H A MACHINE ERROR HAS OCCURRED IN THE SUBROUTINE KERG
   1EN/1H ,35H RESTART PROBLEM FROM THE BEGINNING)
    NERR=1
    GO TO 330
280 BGFR(I)=WBG
    SMF(I)=CC*FKZER
    IF(NTHER)300,300,290
290 BGF(I)=SMF(I)
    BGFR(I)=SMF(I)
    SMGR(I)=1.
300 CONTINUE
310 CONTINUE
320 CONTINUE
330 RETURN
    END(2,1,0,0,0)
```

```
C      SRL-HERESY I    SUBROUTINE EDIT          VERSION OF 8/10/65
C
      SUBROUTINE EDIT
C
C     INSERT DIMENSION DECK FOR MAIN PROGRAM 2
C                                                                      DI
      CALL EFM                                                         DI
      NOUT=10
      KSERCH=KSERCH
      DO 10 I=1,NTYPE
   10 ABSRB(I)=V2(I)/A2
      CALL RVMPDG(NTYPE,ABSRB,ETA,T5)
      CALL MXMPCV(NTYPE,SR,T5,T6)
      CALL RVMPDG(NTYPE,T6,A,T5)
      FINK2=FK2*TAUTOL
      DO 20 I=1,NTYPE
   20 RESABS(I)=T5(I)/FINK2
      IF(LCRIT)28,28,22
   22 IF(NSEPR)28,28,24
   24 WRITE OUTPUT TAPE NOUT,26,IT4
   26 FORMAT(1H1,32HCRITICALITY SEARCH ITERATION NO I4)
      GO TO 32
   28 WRITE OUTPUT TAPE NOUT,29
   29 FORMAT(1H1)
   32 WRITE OUTPUT TAPE NOUT,30,IT2,FK2,FK1
   30 FORMAT(1HO,24HNO EIGENVALUE ITERATIONS,5X10HEIGENVALUE,5X19HPREVIO
     1US EIGENVALUE/1H ,16X13,10XF10.7,9XF11.7)
      WRITE OUTPUT TAPE NOUT,35
   35 FORMAT(1HO,45X29HABSORPTIONS FOR EACH ROD TYPE)
      WRITE OUTPUT TAPE NOUT,40
   40 FORMAT(1H , 8HROD TYPE,5X12HNO RODS/TYPE, 5X7HROD SMF, 8X5HGAMMA,1
     10X3HETA,10X1HA, 8X13HTHERMAL ABSRB, 5X15HRESONANCE ABSRB)
      DO 50 I=1,NTYPE
   50 WRITE OUTPUT TAPE NOUT,60,I,NRPRT(I),SMFO(I),GAMMA(I),ETA(I),A(I),
     1ABSRB(I),RESABS(I)
   60 FORMAT(1H ,3X12,11X14,10XF7.5,5XF11.6,5XF7.5,5XF7.4,7XF9.6,10XF9.6
     1)
      IF(LCRIT-2)550,500,550
  500 IF(NF(KSERCH))520,520,510
  510 I=KSERCH
      THERU(I)=1./(1.+VCELL(I)*SIGC(I)*(GAM(I)+SM(I)-.228714/D(I)-
     1(1./(12.56637*D(I)))*LOGF(XLSQ2(I)/((1.+BSQ*XLSQ)*VCELL(I)))))
      GO TO 550
  520 THERU(KSERCH)=0.0
  550 CALL RVMPDG(NTYPE,FKMAX,ETA,T5)
      CALL RVMPCV(NTYPE,FKMAX,ABSRB,HI)
      CALL RVMPCV(NTYPE,T5,ABSRB,HETAI)
      AVETA=HETAI/HI
      CALL RVMPDG(NTYPE,FKMAX,A,T5)
      CALL RVMPMX(NTYPE,T5,SR,T6)
      DO 70 I=1,NTYPE
   70 T6(I)=FKMAX(I)-T6(I)
      SYSYSY=0.0
      UYUYUY=0.0
      VYVYVY=0.0
      CALL RVMPDG(NTYPE,ETA,T6,T5)
      CALL RVMPCV(NTYPE,ABSRB,T5,SYSYSY)
      CALL RVMPCV(NTYPE,ABSRB,FKMAX,UYUYUY)
      CALL RVMPCV(NTYPE,RESABS,FKMAX,VYVYVY)
      D2OABS=0.0
      DO 75 I=1,NKIND
```

```
      RPX=0.0
      DO 73 J=1,NTYPE
      IF(KIND(J)-I)73,72,73
   72 RPX=RPX+ABSRB(J)*FKMAX(J)
   73 CONTINUE
      IF(THERU(I))430,430,74
   74 D2OABS=D2OABS+RPX*(1.-THERU(I))/THERU(I)
   75 CONTINUE
   78 XMOD2=SYSYSY/FINK2-UYUYUY
      THUT=UYUYUY/(UYUYUY+D2OABS)
      REESPR=(UYUYUY+XMOD2)/(UYUYUY+VYVYVY+XMOD2)
      XLEAK=XMOD2-D2OABS
      XNLEAK=(UYUYUY+D2OABS)/(UYUYUY+XMOD2)
      FKMASB=0.0
      SMFKMX=0.0
      DO 80 I=1,NTYPE
      SMFKMX=SMFKMX+FKMAX(I)
   80 FKMASB=FKMAX(I)*ABSRB(I)+FKMASB
      FKMASB=FKMASB/SMFKMX
      WRITE OUTPUT TAPE NOUT,90,FINK2,FKMASB,AVETA,THUT,REESPR,D2OABS,
     1XLEAK,XNLEAK
   90 FORMAT(1HO,46HEIGENVALUE CORRECTED FOR FAST AXIAL LEAKAGE = F10.8,
     15X25HAVERAGE ROD ABSORPTION = F10.7/1HO,14HAVERAGE ETA = F10.7,5X2
     22HTHERMAL UTILIZATION = F10.8,5X31HRESONANCE ESCAPE PROBABILITY =
     3F10.8/1HO,39HAPPROX MODERATOR THERMAL ABSORPTIONS = F9.5,5X16HTOTA
     4L LEAKAGE = F9.4,5X26HNON-LEAKAGE PROBABILITY = F10.7)
      IF(INF)190,190,120
  120 IF(BSQ)190,130,190
  130 IF(LCRIT-3)140,190,140
  140 BSQ1=.001
      XKO1=(1.+XLSQ*(1.-THUT)*BSQ1)*EXPF(BSQ1*(B(1)*TAU(1)+B(2)*TAU(2)+
     1 B(3)*TAU(3)))
      BSQ2=.0008
      XKO2=(1.+XLSQ*(1.-THUT)*BSQ2)*EXPF(BSQ2*(B(1)*TAU(1)+B(2)*TAU(2)+
     1 B(3)*TAU(3)))
  150  BSQ=(FK2-XKO1)*(BSQ2-BSQ1)/(XKO2-XKO1)+BSQ1
      XKO =(1.+XLSQ*(1.-THUT)*BSQ )*EXPF(BSQ *(B(1)*TAU(1)+B(2)*TAU(2)+
     1 B(3)*TAU(3)))
      IF(ABSF(FK2-XKO)-.00001)170,160,160
  160 XKO1=XKO2
      BSQ1=BSQ2
      XKO2=XKO
      BSQ2=BSQ
      GO TO 150
  170 WRITE OUTPUT TAPE NOUT,180,BSQ
  180 FORMAT(1HO,37HINFINITE LATTICE MATERIAL BUCKLING = F10.8)
      BSQ=0.0
  190 WRITE OUTPUT TAPE NOUT,200
  200 FORMAT(1HO,41X37HAVERAGE ABSORPTIONS FOR EACH ROD KIND/1H ,27X8HRO
     1D KIND,3X10HTOTAL RODS,4X17HAVG. THERMAL ABS.,3X19HAVG. RESONANCE
     2ABS.)
      DO 230 I=1,NKIND
      N=0
      SUM1=0.0
      SUM2=0.0
      DO 220 J=1,NTYPE
      IF(KIND(J)-I)220,210,220
  210 N=N+NRPRT(J)
      FN=NRPRT(J)
      SUM1=SUM1+ABSRB(J)*FN
      SUM2=SUM2+RESABS(J)*FN
```

```
220 CONTINUE
    FN=N
    AVG1=SUM1/FN
    AVG2=SUM2/FN
    WRITE OUTPUT TAPE NOUT,225,I,N,AVG1,AVG2
225 FORMAT(1H ,30XI2,9XI4,11XF9.6,12XF9.6)
230 CONTINUE
    IF(LCRIT-2)239,231,239
231 I=KSERCH
    IF(NF(I))233,233,235
233 WRITE OUTPUT TAPE NOUT,234,I
234 FORMAT(1H0,45HA NEW THERMAL UTILIZATION VALUE FOR ROD KIND I2,44H
   1CANNOT BE COMPUTED DUE TO INSUFFICIENT DATA/1H ,109HTHE MODERATOR
   2ABSORPTIONS ABOVE ARE IN ERROR BECAUSE OF THIS,HOWEVER THE PRODUCT
   3 OF LATTICE F AND NON-LEAKAGE/1H ,23HPROBABILITY IS CORRECT.)
    GO TO 239
235 WRITE OUTPUT TAPE NOUT,236,GAM(I),I,THERU(I)
236 FORMAT(1H0,67HTHE VALUE OF THERMAL UTILIZATION CORRESPONDING TO A
   1GAMMA VALUE OF F10.6,14H FOR ROD TYPE I2,4H IS F10.8)
239 IF(LCRIT)250,250,240
240 IF(IT2-2)290,250,270
250 PRINT 260
260 FORMAT(1H ,26HPROBLEM RAN THROUGH TO END)
    GO TO 290
270 PRINT 280
280 FORMAT(1H ,26HPROBLEM DID NOT RUN TO END)
290 RETURN
430 WRITE OUTPUT TAPE NOUT,440
440 FORMAT(1H0,103HA THERMAL UTILIZATION VALUE FOR A ROD KIND IS ZERO.
   1 MODERATOR THERMAL ABSORPTIONS CANNOT BE SOLVED FOR./1H ,63HTHE MO
   2DERATOR ABSORPTIONS WILL BE LUMPED INTO THE TOTAL LEAKAGE)
    D2OABS=0.0
    GO TO 78
    END(2,1,0,0,0)
```

```
C      MATRIX SUBROUTINES FOR HERESY 1
C
       SUBROUTINE ABMXVC(JMAX,V,A,IMAX)
       DIMENSION V(50)
       IMAX=1
       A=ABSF(V(1))
       DO 54 I=2,JMAX
       IF(A-ABSF(V(I)))53,54,54
 53    A=ABSF (V(I))
       IMAX=I
 54    CONTINUE
       RETURN
       END(2,1,0,0,1)
C
       SUBROUTINE DGMPMX(JMAX,XS,YS,ZS)
       DIMENSION XS(50),YS(50,50),ZS(50,50)
       DO 1 I=1,JMAX
       DO 1 J=1,JMAX
 1     ZS(I,J)=XS(I)*YS(I,J)
       RETURN
       END(2,1,0,0,1)
C
       SUBROUTINE MXMPCV(JMAX,XS,YS,ZS)
       DIMENSION XS(50,50),YS(50),ZS(50)
       DO 1 I=1,JMAX
       ZS(I)=0.0
       DO 1 J=1,JMAX
 1     ZS(I)=ZS(I)+XS(I,J)*YS(J)
       RETURN
       END(2,1,0,0,1)
C
       SUBROUTINE MXADDG(JMAX,XS,YS,ZS)
       DIMENSION XS(50,50),YS(50),ZS(50,50)
       DO 10 J=1,JMAX
       DO 10 I=1,JMAX
       IF(I-J)11,12,11
 11    ZS(I,J)=XS(I,J)
       GO TO 10
 12    ZS(I,J)=XS(I,J)+YS(I)
 10    CONTINUE
       RETURN
       END(2,1,0,0,1)
C
       SUBROUTINE MXMPDG(JMAX,XS,YS,ZS)
       DIMENSION XS(50,50),YS(50),ZS(50,50)
       DO 1 I=1,JMAX
       DO 1 J=1,JMAX
 1     ZS(I,J)=XS(I,J)*YS(J)
       RETURN
       END(2,1,0,0,1)
C
       SUBROUTINE MXSBMX(JMAX,XS,YS,ZS)
       DIMENSION XS(50,50),YS(50,50),ZS(50,50)
       DO 1 J=1,JMAX
       DO 1 I=1,JMAX
 1     ZS(I,J)=XS(I,J)-YS(I,J)
       RETURN
       END(2,1,0,0,1)
C
```

```
C
      SUBROUTINE MXMPMX(JMAX,XS,YS,ZS)
      DIMENSION XS(50,50),YS(50,50),ZS(50,50)
      DO 1 I=1,JMAX
      DO 1 J=1,JMAX
      ZS(I,J)=0.0
      DO 1 K=1,JMAX
 1    ZS(I,J)=ZS(I,J)+XS(I,K)*YS(K,J)
      RETURN
      END(2,1,0,0,1)
C
      SUBROUTINE RVMPCV(JMAX,XS,YS,ZS)
      DIMENSION XS(50),YS(50)
      ZS=0.0
      DO 1 I=1,JMAX
 1    ZS=ZS+XS(I)*YS(I)
      RETURN
      END(2,1,0,0,1)
C
      SUBROUTINE RVMPDG(JMAX,XS,YS,ZS)
      DIMENSION XS(50),YS(50),ZS(50)
      DO 1 I=1,JMAX
 1    ZS(I)=XS(I)*YS(I)
      RETURN
      END(2,1,0,0,1)
C
      SUBROUTINE RVMPMX(JMAX,XS,YS,ZS)
      DIMENSION XS(50),YS(50,50),ZS(50)
      DO 1 J=1,JMAX
      ZS(J)=0.0
      DO 1 I=1,JMAX
 1    ZS(J)=ZS(J)+XS(I)*YS(I,J)
      RETURN
      END(2,1,0,0,1)
```

APPENDIX C - IBM 704 Operating Instructions

1. ROUTINE PROBLEM OPERATION

(a) Machine Requirements

To run a routine problem on the 704 requires the following minimum equipment:

- IBM 704 with 32 k memory
- Five tape units
- Card reader
- On-Line printer

(b) Tape Units and Functions

The five tape units are dialed to numbers 1, 2, 3, 8, and 10, and are assigned the following functions:

Tape 1: Program tape (file protect on)

Tape 2: Geometry temporary storage (must always be mounted)

Tape 3: Geometry permanent storage (mounted only if used)

Tape 8: Input data in BCD card image form

Tape 10: Output of all information

(c) Card Reader Function

The card reader is used only to place data on tape 8 and to initiate the CHAIN program. This initiation is done by first preparing all the tapes in (b) that are used, then placing in the card reader the following two cards:

- Self-Loading Chain Initiator Program card (SLCI card)
- Control Card calling for Link 1 of tape 1 [C.C(1,1) CARD]

Clear and load cards. The program from this point will operate automatically unless an error stop occurs. If stop does occur refer to Section 2 of this appendix for the proper procedure.

(d) On-Line Printer Functions

The alphameric card read as the first card of each problem is printed out, and will be followed by one of four messages which give a summary of what happened to the problem. These messages are listed below with their meaning.

- AN ERROR HAS BEEN DETECTED IN THE INPUT DATA FOR THIS PROBLEM. AN ATTEMPT WILL BE MADE TO CONTINUE WITH SUCCESSIVE PROBLEMS - The error checking routines in the first link of the code have detected an error that cannot allow the problem to run to the end - no action is required of the operator, for restart has been programmed for the next problem.

- PROBLEM RAN THROUGH TO END - Any problem run to the end will have this on the on-line printer - no action is called for by operator.

- PROBLEM DID NOT RUN TO END - This indicates an error condition arose at some point of the calculations in the second link of the program. Information explaining the stop is on tape 10 - no action is called for from operator.

- MACHINE ERROR - RESTART PROBLEM - A branching error has occurred in the machine - Operator should stop machine; rewind tapes 1, 2, 3, and 8; and restart the problem from the beginning using the procedure of paragraph c above.

(e) End of Computations

The signal that the end of the problem has been reached may be one of two transfers that depend on the input data. In some cases a HPR 0,1 will occur, while in others the Card Reader will be selected. More often than otherwise the stop will select the Card Reader; however, the HPR 0,1 is equally valid under the assumption that a part of the input data was left out.

2. RESTART PROCEDURE FOR PROGRAM STOPS

If any standard BSS program stop occurs, other than the HPR 0,1 mentioned in (1e) above, the stop is considered an error stop. Certain steps may and should be taken to interrogate the status of the machine. These are:

(a) Check on-line printer

- If problem in which stop occurred is the first problem, the entire run should be cancelled after the address lights are recorded.
- If at least one problem has been completed, continue with paragraph (b).

(b) The restart procedure following case ii of (a) above is as follows:

- Rewind tape 1
- Place SLCI and C.C.(1,1) cards in card reader
- Load cards WITHOUT CLEARING MACHINE

(c) If further stops occur, the procedure of (b) above should be followed until a HPR 0,1 appears in the STORAGE REGISTER lights.

This procedure assures that problems following the one in which the stop originally occurred will not be affected by errors in previous data if at all possible.

3. PROGRAM TAPE PREPARATION

The method of preparing program tapes is contained in a Special Version of the IBM CHAIN MONITOR program written at SRL. This system is described in Appendix D, and reference to this appendix will explain the operations involved.

4. PERMANENT GEOMETRY TAPE PREPARATION

No special preparations are necessary unless the problem is the first one on the tape. If a new tape is used, the only preparation (other than mounting the tape on unit 3) is to place an END OF FILE mark on the tape in its rewind position. This may be done by placing a WEF 3 instruction into the machine manually from the console keys. The remainder of the tape processing will be controlled by the mnemonics NFILE, NWRITE, and XCOR specified by the user.

5. ADDITIONAL PROGRAM STOPS

In addition to the standard BSS LOADER program stops, the following stops have been programmed.

Storage Register Address Lights	Error Condition
HPR 464	CHECKSUM ERROR in reading geometry scratch tape
HPR 547	CHECKSUM ERROR in reading geometry permanent tape
HPR 610	Misplaced EOF on tape 3 or misread of tape 3

Pressing start allows any of the above three stops to be ignored.

HTR 5, 4	File number called for is not on tape 3

Pressing start returns control to calling program.

APPENDIX D - Operation and Preparation of CHAIN Programs

The version of the Chain Monitor system used in the SRL - HERESY I code was written by J. E. Suich of SRL. This system consists of a package of four routines which permit operation on the IBM 704 of programs written within the specifications of the IBM-MONITOR routine, CHAIN. These four routines are described below.

(1) SUBROUTINE CHAIN (LINK, IOUNIT)

This subroutine performs the function of the IBM-FORTRAN MONITOR subroutine of the same name (see, e.g. Reference Manual, 709/7090 FORTRAN (IBM-C28-6054-2)). LINK is the fixed point integer identification number of the next chain link desired, and IOUNIT is the logical tape designation (fixed point integer 1-10) of the tape on which this link is stored. Links may be called in any order from any number of tapes. Execution of the calling sequence causes the desired link to be brought into memory, and control to be passed to it, without destroying COMMON storage.

Termination of any program or the initiation of another CHAIN job is accomplished by calling a tape designation that does not exist (i.e. >10). Execution of this command will initiate a card load sequence which will in turn may cause the SELF-LOADING CHAIN INITIATOR described below to be executed. By this process a sequence of CHAIN programs may be continuously run.

Only one error stop has been programmed: HTR 3,4 indicates that the desired LINK was not found on the designated IOUNIT. Pressing start causes control to return to the calling program.

(2) SELF-LOADING CHAIN INITIATOR (SLCI)

To provide a convenient method of beginning the chain job from a previously prepared chain program tape, a self-loader has been written which reads a single Control Card designating the desired initial values of LINK and IOUNIT. The Control Card format is row binary: LINK in the address field of the 9L row, and IOUNIT in the address field of the 9R row. For example, the Control Card to enter the link with identification number 5 on the chain program tape mounted on tape 9 has 9's punched in columns 34, 36, 69, and 72. The control cards may be punched with any additional characters (except I, R, Z, and 9) in any columns (for identification purposes).

(3) CHAIN PROGRAM TAPE GENERATOR (CPTG)

This routine fulfills the IBM-MONITOR function of preparing chain program tapes, although in substandard fashion. To use this program tape generator, the main program of each link must be converted to a subroutine named MAIN, either by compiling with the header card SUBROUTINE MAIN, or by punching 7's in columns 1, 4, 8, 12, 14, 15, 18, 19, 22, 24, 25, 26, 31, and 32 of the program card for the main program of each link and correcting the check sums. The various links are then stacked in the card hopper, tapes readied, and CLEAR, LOAD CARDS pressed. Figure D-1 shows schematically the deck makeup.

(4) END PROGRAM TAPE CARD (EPTC)

The program tape generated by the above procedure requires an END of FILE mark following the last link. This self-loader reads a Control Card (see discussion under (2) above), ignores the LINK value, and ends the tape designated by IOUNIT. The card should be placed behind the transfer card of the last link when making up a program tape (Figure D-1). There should be an END TAPE loader and Control Card pair for each IOUNIT designated.

FIGURE D-1

EXAMPLE OF THE DECK MAKEUP TO WRITE LINKS
1, 2, AND 5 ON TAPE 1 AND 1, 4, AND 3 ON TAPE 9

```
BSS LOADER
MAIN
SUBROUTINES (INCLUDING CHAIN)
CPTG
TRANSFER CARD
CONTROL CARD (1,1)

BSS
MAIN
SUBR
CPTG
TRA
C.C.(1,9)

 .
 :
C.C.(2,1)

 :
C.C.(5,1)

 :
C.C.(4,9)

 :
C.C.(3,9)

EPTC
C.C.(0,1)
EPTC
C.C.(0.9)
```

Listings of the Four Subroutines

```
       REM SUBROUTINE
       FUL
       ORG 0
       MZE *,,ENTRY-2
       PZE
       PZE BREAK,,0
       PZE 32562
       BCD 1CHAIN
       PZE EP
ENTRY  EQU *
       ORG 0
       REL
   EP  SXD EOF,2
       TXI *+1,2,1
   ID  CLA 1,4
       STA *+1
       CLA **
       ARS 18
       STO ID
   IO  CLA 2,4
       STA *+1
       CLA **
       ARS 18
       ADM BASE
       STA RDID
       STA REWTP
       STA TEMP
       CLA TEMP
       SUB OUT
       TMI RDID
       TRA CREAD
 RDID  RTB 1
       CPY 16
       TRA *+3
  EOF  TIX REWTP,2,0
       HTR 3,4
       CLA ID
       SUB 16
       TNZ RDID
 BASE  PXD 144,0
       CAD 16
       CAD 17
       LXA 16,4
       TIX 16,4,18
REWTP  REW 1
       TRA RDID
CREAD  RCD
       CPY 0
       CPY 1
       TRA 0
OUT    PZE 155
TEMP   PZE 0
BREAK  EQU *
       END **
```

```
CHAIN (ID,IO)

P         G
R    C    E
O    A    N
G    R    E
R    D    R
A         A
M         T
          O
          R
```

DESIRED LINK IDENT

LOGICAL TAPE DESIGNATION

TAPE UNIT ADRESS

TAPE LINK IDENT

EOF, REWIND ONCE
LINK NOT ON TAPE, RETURN

NOT THIS LINK
LINK IS FOUND
LOAD
TAPE
SEQUENCE

```
      REM ROUTINE TO MAKE A CHAIN PROGRAM TAPE
      FUL                                          G
      ORG 0                                P       E
      MZE *,,ENTRY-2                       R   C   N
      PZE                                  O   A   E
      PZE BREAK,,1                         G   R   R
      PZE 32562                            R   D   A
      BCD 1000000                          A       T
      PZE BP+1                             M       O
ENTRY EQU *                                        R
      ORG 0
      REL
   EP BCD 1MAIN                        MAIN PGM ENTRY POINT
      CLA *                            PROGRAM BREAK +1
      SUB ONE
      STA LOAD                         ESTABLISH
      STA WTPGM+5                      ABSOLUTE
      CLA EP                           BINARY
      STA LOAD+6                       PROGRAM
      STA LOAD+7                       LOADER
      RCD                              READ CONTROL CARD
      CPY EP                           LINK IDENT
      CPY EP+1                         TAPE DESIGNATION
      CLA EP+1                         FORM
      ADM BASE                         TAPE ADDRESS
      STA WTPGM
      LXA EIGHT,4                      SHIFT TAPE LOADER
      CLA LOAD+8,4
      STO 24,4
      TIX *-2,4,1
WTPGM WTB 1                            WRITE BINARY TAPE
 BASE PXD 144,0
      LXA *+3,4
      CPY EP                           LINK IDENT
      TIX *+1,4,16
      CAD **,4                         PROGRAM
      TIX *-1,4,1
      STO EP
      CPY EP                           CHECKSUM
      RCD                              LOAD NEXT LINK
      CPY 0
      CPY 1
      TRA 0
 LOAD CAD **,4
      TIX 16,4,1
      STO 1
      CPY 0
      CLA 0
      SUB 1
      TZE **
      HTR **
  ONE PZE 1
EIGHT PZE 8
BREAK EQU *
      END **
```

```
      REM SELF LOADING CHAIN INITIATOR
      FUL
      ORG 0                              SELF
      LXD **,4                           LOAD
      CPY 22274,4                        SEQUENCE
      TIX *-1,4,1                        EOF
      HTR **                             CONTROL CARD
      RCD                                LINK
      CPY 0                              TAPE UNIT
      CPY 1
      CLA 1
      ADM BASE
      STA RDTP
RDTP  RTB 1                              TAPE ID
      CPY 1
      TRA *+3                            TAPE NOT REWOUND
      HTR RDTP                           OR WRONG TAPE
      HTR RDTP                           COMPARE
      CLA 0                              IDENTS
      SUB 1                                WRONG LINK
      TNZ RDTP                           RIGHT LINK
BASE  PXD 144,0                          LOAD
      CAD 16                             TAPE
      CAD 17                             SEQUENCE
      LXA 16,4
      TIX 16,4,18
      END **
```

```
      REM SELF LOADER TO END CHAIN TAPE
      FUL
      ORG 0                              SELF
      LXD **,4                           LOAD
      CPY 22274,4                        SEQ
      TIX *-1,4,1
      HTR **                             CONTROL CARD
      RCD                                IGNORE
      CPY 0                              TAPE UNIT - LOGIGAL
      CPY 1
      CLA 1                                        - ADDRESS
      ADM BASE
      STA WEFTP
      STA REWTP                          END FILE
WEFTP WEF 1                              REWIND
REWTP REW 1                              LOAD
      RCD                                NEXT
      CPY 0                              END
      CPY 1                              FILE
      TRA 0
BASE  PXD 144,0
      END **
```