JUL 7 1966

MASTED

WANL-TNR-133-Revised
December 1965

Westinghouse Astronuclear Laboratory

VARI-QUIR:

**A TWO-DIMENSIONAL TIME-DEPENDENT
MULTI-GROUP DIFFUSION CODE**

RELEASED FOR ANNOUNCEMENT
IN NUCLEAR SCIENCE ABSTRACTS

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty, or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

WANL-TNR-133-Revised
December 1965

SUBMITTED BY:

Westinghouse Electric Corporation
Astronuclear Laboratory
Pittsburgh, Pennsylvania 15236

INFORMATION CATEGORY

Unclassified

D W Drawbaugh 12/18/65
Authorized Classifier Date

PREPARED BY:

J. W. Riese
J. W. Riese

George Collier
G. Collier

C. E. Rieck
C. E. Rieck

VARI-QUIR:

A TWO-DIMENSIONAL TIME-DEPENDENT
MULTI-GROUP DIFFUSION CODE

APPROVED BY:

D W Drawbaugh
D. W. Drawbaugh, Manager
Reactor Physics & Mathematics

TABLE OF CONTENTS

List of Illustrations	1
Abstract	ii
Introduction	iii
Chapter I - Problem, Method of Solution, Application	
A. The Problem	1
B. The Variational Principle	2
C. The "Almost-Separable" Form	5
D. The "Quadratic-in-Region" Form (VARI-QUIR)	8
E. Application: A Simple Steady-State Problem	13
F. A Simple Kinetics Problem	14
G. Computer Coding - The Continuity Problem	15
H. Typical Steady-State Problem	21
J. Kinetics - The "Wide Range of Periods" Difficulty	22
K. A More Detailed Kinetics Problem	27
L. Feedback	28
Chapter II - Description of Code	
A. Present Limitations	30
B. Input Preparation for Time-Dependent VARI-QUIR	32
Illustrations	38
Appendix A The Steady-State Variational Principle	51
Appendix B FORTRAN Listing and Test Problem	53
Appendix C Discussion of the New Sample Problem Printout	188
Appendix D Improvements in the Neutronics Code	194

LIST OF ILLUSTRATIONS

Fig. 1.	First Trial of VARI-QUIR - Steady-State Problem	38
2.	Two-Slab Kinetics - Almost-Separable Method	39
3.	Two-Slab Kinetics - VARI-QUIR	40
4.	Five-Region Four-Group Steady-State Problem: Group 1	41
5.	Group 2	42
6.	Group 3	43
7.	Group 4	44
8.	Five-Region Four-Group Kinetics: Total Power vs. Time	45
9.	Power Density vs. Position	46
10.	Group 4 Flux	47
11.	Group 1 Precursors	48
12.	Simple Feedback Test Problem	49
13.	Illustrative Two-Period Time Problem	50
14.	Relative Power as a Function of Time	192

ABSTRACT

A variational approximation is developed to solve the time-dependent neutron diffusion equations. The method is incorporated into a computer code, which allows up to 4 energy groups and 6 precursors. Some sample results are presented.

This report represents an updating and expansion of WANL-TNR-133. The principle changes include modifications of the input-preparation instructions, the inclusion of a listing of the revised code and a detailed sample problem printout in Appendix B, a discussion of this printout in Appendix C, and a discussion of improvements in the neutronics code in Appendix D.

INTRODUCTION

This report describes our progress to date on the non-separable kinetics problem. It is divided into three chapters, to cover:

- I. the problem, our approximation theory, some practical difficulties in applying this theory (presented roughly in the chronological order in which they arose), how these difficulties were handled, and some sample results;
- II. some specific details of the present code, its usage and limitations; and
- III. our proposed further developments in this code, some very easy to make, others involving lengthier work.

A FORTRAN listing of the code is included as Appendix B.

I. PROBLEM, METHOD OF SOLUTION, APPLICATION

A. The Problem

We wish to solve the equations arising from multigroup, space- and time-dependent (and non-separable*), neutron diffusion theory. For some number G of flux energy groups, and J of delayed neutron precursor groups, these simultaneous partial differential equations may be written:

$$g = 1 \text{ to } G: \quad \frac{\partial}{\partial t} \frac{\phi_g}{V_g} = \nabla \cdot D_g \nabla \phi_g - \sigma_g^R \phi_g + \sum_{g' \neq g}^G \sigma_{g' \rightarrow g}^{TR} \phi_{g'} \\ + (1 - \beta_T) \chi_g \sum_{j=1}^J \nu_j \sigma_j^F \phi_j + \sum_{j=1}^J S_{j \rightarrow g} \lambda_j C_j$$

$$j = 1 \text{ to } J: \quad \frac{\partial}{\partial t} C_j = -\lambda_j C_j + \beta_j \sum_{g=1}^G \nu_g \sigma_g^F \phi_g$$

where the notation will be obvious to the reader.

The coefficients σ , D , etc., may themselves be functions of both space and time; in fact, if feedback is involved, they may even depend on ϕ itself, through some further feedback simultaneous equations. The problem is completely defined when the above differential equations are accompanied by initial conditions on both ϕ_g and C_j , and spatial boundary conditions on ϕ_g . The precursors C_j need no boundary conditions, since the precursor equations involve no spatial derivatives.

* "Non-separable" means simply that the solutions are too complicated to be separated into a function of space, multiplied by a function of time. In other words, the spatial shape of the neutron distribution is itself a function of time.

It will sometimes be convenient in what follows to condense our notation by letting ϕ_g refer to either flux or precursor, with index g running from 1 to $G + J$, and the coefficients σ , λ , γ , S , λ , β combined into a matrix. Then the foregoing equations reduce to simply

$$g = 1 \text{ to } G + J: \quad \frac{d}{dx} \frac{\phi_g}{V_g} = \nabla \cdot D_g \nabla \phi_g + \sum_{g'=1}^{G+J} A_{gg'} \phi_{g'} \quad (1.)$$

where, of course, for $g = G + 1$ to $G + J$, $V_g = 1$ and $D_g = 0$.

While (1.) is sufficient (with initial and boundary conditions) to define the exact problem, our approximate solutions will sometimes make reference to the "adjoint" or "importance" function ϕ_g^* . This function is defined as satisfying an equation like (1.), but with all operators replaced by their adjoints. Specifically

$$g = 1 \text{ to } G + J: \quad -\frac{1}{V_g} \frac{d}{dx} \phi_g^* = \nabla \cdot D_g \nabla \phi_g^* + \sum_{g'=1}^{G+J} A_{g'g} \phi_{g'}^* \quad (2.)$$

B. The Variational Principle

Equations (1.) and (2.) may be re-expressed in a mathematical form quite different in appearance, but completely equivalent in information. (Our reasons for wanting to so re-write the problem will become evident toward the end of Section B. Essentially, it is a more convenient point of departure from the exact problem to approximate solutions).

This equivalent form, the Variational Principle, may be derived as follows. Suppose we multiply the flux equation, (1.), by some small variation $\delta \phi^*$ of the adjoint solution from its exact value; this variation $\delta \phi^*$ may be quite different

in shape from the adjoint ϕ^* , but it must be small (approaching zero) in magnitude. If we similarly multiply (2.) by $\delta\phi$, add the two equations, integrate the result over space and time, and recognize sums like $(\delta\phi^*) \nabla \phi + \phi^* \nabla \delta\phi$ to be equal to $\nabla(\phi^* \nabla \phi)$ (since the variations were assumed very small), we finally obtain

$$\delta I = 0 \quad (3.)$$

where
$$I = \sum_{g=1}^{G+J} \int_V d^3x \int_0^T dt \left\{ -\phi_g^* \frac{\partial}{\partial t} \frac{\phi_g}{\nabla_g} - D_g (\nabla \phi_g^*) \cdot (\nabla \phi_g) + \sum_{g'=1}^{G+J} \phi_g^* A_{gg'} \phi_{g'} \right\}$$

+ surface terms (3-a.)

The surface terms arise in the integration by parts necessary to split the ∇^2 operator between ϕ and ϕ^* . They consist of terms like $[(\delta\phi^*) \nabla_n \phi + (\delta\phi) \nabla_n \phi^*]_S$ and are therefore equal to zero if either the flux is prescribed, or its normal gradient is set equal to zero, everywhere on the boundary, which we shall assume to be the case at present.* The time integration by parts may also yield "surface" terms at $t = 0$ and T , but these will have no significance in our final usage.

Conversely, we may readily re-derive (1.) and (2.) as Euler equations starting from (3.) and the definition (3-a.). Taking the variation of I , we remove the integrations by realizing that ϕ and ϕ^* are to be varied freely at each point in space and time. Assuming further that ϕ and ϕ^* are to be varied independently of each other, the coefficients of $\delta\phi$ and $\delta\phi^*$ must therefore be separately equal to zero, which splits (3.) back up into the two sets (1.) and (2.)

* For the important case of linear extrapolation distance, in which a combination of the flux and its gradient are set equal to zero at the boundaries, a correction will need to be made for non-zero surface terms. Of course, if the extrapolation distance is roughly the same in all groups, one can merely re-define a new boundary at which the flux goes to zero, and again no correction will be necessary.

Equation (3.) says that, considering the class of all possible functions which satisfy our initial and boundary conditions, those particular functions which render I stationary will be the ϕ and ϕ^* we seek. By this we mean that small first-order changes in ϕ and ϕ^* will lead to only second-order changes in I . As opposed to self-adjoint problems, where $\phi^* = \phi$, and the stationary point in function space is a true minimum or maximum, we have in the present case a saddle-point. As a simple analog, the function xy , where x and y vary independently, has a saddle-point at the origin, whereas if $x = y$, the resultant function x^2 has a true minimum at 0.

The variational principle (3.) is exact, and completely equivalent to the combination (1.) and (2.). It is, in addition, a convenient starting point for approximate solutions, as follows. Instead of considering the class of all possible functions with the correct boundary conditions, suppose we consider some more limited sub-class of this class. It may then be possible to examine this sub-class in detail, to find the particular member which optimizes I . At the same time, if the sub-class is chosen wisely, its "best" member should be very close to the "best" member of the entire class - i.e., the exact solution.

The foregoing approximation procedure is known as the variational method. It has at least three strong points:

- (1.) one is completely free in the choice of the mathematical form of the solution, hence this form may be chosen on the basis of ease of manipulation in the ensuing calculations;

- (2.) one can make use of any extraneous information or intuition concerning the solution, by wise choice of the physical content, or behavior range, of the subclass to be investigated; and
- (3.) one has great flexibility in trading between accuracy and computational ease, by making the sub-class larger or smaller.

Because of their great complexity, no serious consideration was ever given to obtaining the exact solution of equations (1.) and (2.). Needing an approximation technique, the variational method seemed the most fertile ground to search. The next question was: what solution-form should be used?

C. The "Almost-Separable" Form

Perhaps the most obvious choice for a solution form would be arrived at as follows. If we must solve a non-separable problem, let's make it almost separable. I.e., if $a(t)f(\vec{x})$ won't do, let's try

$$\phi_g(\vec{x}, t) \approx a_g(t) f_g(\vec{x}) + b_g(t) h_g(\vec{x}) \quad (4.)$$

If a reactor starts at one steady-state and ends at another, we might make $f_g(\vec{x})$ be the initial spatial shape, and $h_g(\vec{x})$ the final, for group g . These two known (from steady-state calculations) functions can certainly fit the flux shape at $t = 0$ and $t = \infty$, so the assumption is that, at all intermediate times, the flux can be represented as some linear combination of them. Making a similar assumption for ϕ_g^* , substituting into (3.), performing the space-integrations explicitly, and then taking the variation, we obtain as Euler equations ordinary differential equations in time for $a(t)$ and $b(t)$. Their solution, even with non-constant

coefficients, is relatively simple. One finds that the equations for the unknown flux time functions a and b are completely decoupled from those for the adjoint functions a^* and b^* (though the known spatial shapes f , h mix with f^* , h^* in both sets of equations). This decoupling is a consequence of the fact that equation (3-a.) contains only products like $a^* b$ in the unknowns, never a^2 or $a \cdot b$.

This solution-form has received wide theoretical attention in the literature, to the exclusion of almost all others. On the other hand, very few numerical results of its use have been published. It was therefore the first approach which we investigated.

Several simple examples are presented and discussed in TMI-567. The principal advantages of this solution-form are:

- (1) it is fairly simple and straightforward to apply (hence utilizing strong point #1 of the variational method, discussed toward the end of Section B.);
- (2) it takes full advantage of the well-developed steady-state codes in existence (hence utilizing strong point #2); and
- (3) it can give good accuracy in problems which are truly almost separable, i.e., which involve only small shape changes.

Some disadvantages are:

- (1) partial prescription of the solution at $t = 0$ and at $t = \infty$ is a very unnatural way to handle a first-order time equation like the diffusion equation. Ideally, one should prescribe the solution only at $t = 0$, then follow wherever the differential equation leads, in both magnitude and shape. Partly as a result of this somewhat contrived approach,

- (2) in problems involving more drastic shape changes, there is no guarantee* that the shape at intermediate times can be adequately represented as a combination of initial and final shapes; e.e., the transition may take place much faster in some spatial regions than in others, and may even be propagated from one spatial region to another[†];
- (3) one must go to great lengths to handle the very detailed, highly accurate steady-state data, only to lose much of this accuracy in the relatively sloppy time-dependent solution; and
- (4) there is very little flexibility in the direction of increasing accuracy (hence strong point #3 of the Variational Method is largely wasted). (One could, of course, increase equation (4.) to three or more terms, but what space shapes are to be used in these additional terms?[‡]

In an attempt to meet some of these problems, the following rather radical departure in solution-form was developed. It is, to the best of our knowledge, original.

* Quite the contrary! For a simple counter-example, see graph labelled "Case III" in TMI-567.

[†] See, e.g., Fig. 10 of this report.

[‡] This question is not impossible to answer, merely messy. One might, for example, calculate steady-state fluxes using some of the initial precursors as sources, under intermediate or final reactor conditions.

D. The "Quadratic-in-Region" Form (VARI-QUIR)

In studying plots of steady-state flux solutions, one observes that they are usually very smooth and simple curves within each material region. Sharp corners occur only in going from one material to another, and sharp dips or peaks only as a result of such transitions. In the more involved time-dependent case, the fluxes may vary rapidly in both magnitude and shape as functions of time, but one would still expect, at each particular instant of time, the same smooth spatial shape within a region. It would therefore seem possible to express the flux accurately within each different region as a (different) low-order polynomial in the spatial variables; the polynomial coefficients would be functions of time, their differential equations to be determined from the variational principle. This is the essence of our method.

Before going into any further detail, let us consider this proposed solution-form in relation to our previous discussion. I.e., how well does it utilize the strong points of the variational method (Section B), and to what extent does it overcome the disadvantages of the "almost-separable solution-form" (Section C)?

With reference to strong point #1, freedom to choose easily manipulated functions, our choice of polynomials would be hard to beat; the operations of equation (3-a.) on polynomial space-functions are almost trivial. Our use of point #2 (choice of a solution class with the right behavior) was already treated in the first paragraph of this section, and was in fact the earliest motivation for our present approach. In contrast to Fourier-type expansions over the entire reactor space, which would require

a great many terms to match the abrupt behavior of the flux near an interface, our solution-form anticipates these discontinuities and their positions. It furthermore automatically inserts more degrees of freedom (i.e., more polynomials) in areas where they are needed to match more complicated flux shapes (namely, in areas where there are more material regions).

Finally, to utilize strong point #3, we can make our solution as accurate as we like by merely adjusting the degree of the polynomial. As an alternative, we may subdivide any real material region into several mathematical regions, to increase the degrees of freedom present. While this latter procedure is theoretically somewhat less efficient than the former, because it provides for possible abrupt changes where none are needed, we have preferred it for the practical convenience of keeping the programming from becoming too awkward. Hence we have worked only with the simplest polynomial that allows smooth curves in a region, with possibly continuous slopes between regions, namely the quadratic.

To proceed in the other direction (greater simplicity, less accuracy), the procedure would be simply to homogenize several adjacent regions. While perhaps some prescription particularly appropriate to the present solution-form might be found, standard homogenizing methods would be adequate as a starter.

Our method also answers all of the objections (to the "almost-separable" technique) raised in Section C. Insertion of the solution-form into equations (3.) and (3-a.) will yield ordinary differential equations in time for the quadratic coefficients.

Since these equations are first-order, and flux and adjoint are now completely decoupled, we need only the starting values of the flux coefficients (hence, of the flux) to solve for the fluxes at all times. Any shape changes will occur naturally as a result of the differential equations, so there is no danger in overlooking any intermediate shapes. The data-handling problem is eliminated because the complicated flux shapes have been summarized in values of a relatively few critical coefficients. Finally, the complete flexibility in trading accuracy vs. computational ease has been shown in the preceding two paragraphs.

Having concluded that the proposed solution-form looks promising from several angles, let us now examine some details. There is only one more decision that need be made to completely specify our method before applying it, namely, the question of continuity. While the precursor densities may suffer discontinuities from region to region, we know that the exact solutions for the fluxes will be continuous in value across an interface, because of the ∇^2 operator in equation (1-a.). Hence we could require our approximate solutions to also be continuous at interfaces. As a matter of fact, this is not merely desirable, but necessary, for otherwise non-integrable infinities will arise in the evaluation of (3.), due to the product $\nabla\phi \cdot \nabla\phi^*$. If limits are properly taken, it turns out that the variational principle will, before yielding any other equations, use up as many degrees of freedom as necessary to require this continuity, first and foremost. Since computers cannot easily handle such limit-taking, we must require continuity of value ahead of time.

This does not apply to slopes, however. While the exact solution would require the proper discontinuity in flux slope to keep $D_g \nabla \phi_g$ (the current) continuous at an interface, errors in slope discontinuity give no trouble in equation (3-a.). Rather than requiring current continuity, we are thus further ahead to let it remain arbitrary, and to let the variational principle pick the best over-all solution, by juggling a mis-fit at an interface against errors inside a region. If one objects that this may give rise to line sources at an interface, we must answer as follows. Since our entire solution is only approximate, it is exactly the solution only for some fictitious distribution of source and sink density. There is therefore no particular reason to abhor the lumping of some of this erroneous source or sink density at an interface.

In final clarification, application of our method to a one-dimensional slab, N-region, problem would result in an apparent $3N(G + J)$ unknown functions of time - the 3 coefficients in $a_i + b_i x + c_i x^2$, for each region $i = 1$ to N , for each of G energy groups and J precursor groups. However, the requirement on the fluxes of continuity at each of $N-1$ interfaces, plus the boundary conditions at the 2 edges, reduces this to $3NJ + (2N-1)G$ independent functions.

When applied to more than one dimension, e.g., X-Y geometry, the obvious generalization of our method is to make the fluxes and precursors quadratic in both x and y within a region; i.e., all 9 possible terms like $x^i y^j$, $0 \leq i \leq 2$, $0 \leq j \leq 2$, must be included. The fluxes must now be continuous along an entire

line interface. For example, along the line interface $x = x_1$, the quadratic in y must be the same when approached from either side of x_1 . This led us to restrict our code in the following convenient (through not strictly necessary) manner.

If two short regions A and B, on the right, were to butt up against one taller region C on the left (see sketch 1-a. below), the two sectional quadratics in y which are approached by A and B from the right would have to behave in a very special way to match any single quadratic in y approached by C from the left. To avoid prescribing the necessary relationships (and, what is much worse, taking these into account when varying only the independent quantities in equation 3.), we disallow such geometries. This does not mean disallowing the problem, but merely modifying it as in sketch 1-b. below. Two (artificial) regions C_1 and C_2 (both having, of course, the same material) replace the original region C. In short, before any problem can be solved on the present code,

C	A
	B

(a.) Not allowed

C_1	A
C_2	B

(b.) Allowed Replacement

SKETCH 1.

any region-dividing lines which end in the interior of the reactor must be continued to both external boundaries. The net effect is merely to increase the accuracy- and labor-somewhat from the original problem by introducing extra artificial regions.

Because our approximation uses the variational method, with quadratic-in-region solution form, it will henceforth be referred to as VARI-QUIR.

E. Application: A Simple Steady-State Problem

The ultimate criterion for judging VARI-QUIR is how well it solves problems. Our early investigation of this question was confined to slab, few-region, few-group, steady-state problems, simple enough so that exact solutions for comparison would be easily available. Despite the absence of the time-dependence, the question of whether our spatial solution form has sufficient flexibility to match typical flux shapes can still be answered here - at least in the negative. If the method does not work on steady-state problems, one can hardly hope to use it for kinetics.

Because the steady-state problem is an eigen-value problem, its variational principle[†] is quite different from the kinetics principle, equations (3.) and (3-a.)

Nevertheless, upon insertion of our VARI-QUIR solution form, the resultant Euler equations for the quadratic coefficients are almost identical with the corresponding kinetics equations, except that the time derivatives are missing. They are therefore merely algebraic equations for the (now constant) coefficients, forming a matrix

[†]For a derivation and statement of the steady-state variational principle, see Appendix A.

eigenvalue problem. It happens that the eigenvalue is itself the functional varied. Now this functional varies from its exact value only in second order, under first-order variation of the flux and adjoint, by definition of the variational principle. Therefore, we may expect an extremely accurate eigenvalue, even when the fluxes are only fair.

A typical result of this phase of the investigation is shown in Fig. 1. The problem was a two-group, three-region (reflector-core-reflector) symmetric slab-reactor. We used the coarsest possible solution form short of homogenization, namely, three regions - not even taking advantage of the symmetry, so that we effectively had only $1\frac{1}{2}$ regions. In other words, of our $2N-1 = 5$ coefficients in each group, there were 2 symmetric sets, or only 3 independent coefficients. Figure 1 also shows, for comparison, the AIM-5 (200-point) solution, which we may regard as exact. Considering the crudity of our approximation, the fit was quite encouraging. For reasons discussed in the preceding paragraph, we were not too surprised at the very accurate eigenvalue obtained - 1.7570 compared to AIM's 1.7578.

F. A Simple Kinetics Problem

With the method checking out well so far, we moved next to simple kinetics problems. A useful example here is one borrowed from TMI-567 (useful for comparison because both the exact solution and the old "almost-separable" approximate solution were already worked out there), and denoted in that report as Case I. It involves two homogeneous just-critical slabs, both running at the same power level, which are joined together at $t = 0$ into a combination below prompt critical. The old results are reprinted here as Fig. 2 for review. When we used VARI-QUIR with just

one region in each slab, results were not much better than the old method. But we now had available unlimited (theoretically) flexibility toward increasing accuracy. When each slab was split into three regions, the result was the remarkably perfect fit shown in Fig. 3. The VARI-QUIR solution is only marked at certain points, because its curve would be indistinguishable from the exact.

G. Computer Coding - The Continuity Problem

Fortified by successful results like those shown in E. and F., we were now ready to code the method for larger problems. At this point, the following difficulty arose. As described in Section D., the fluxes and their adjoints must be continuous at material interfaces. Thus our quadratic coefficients are subject to restraining equations, so that not all of them are independent. The dependent ones must be eliminated before taking the variation (3.), so that only independent quantities are varied (otherwise our variational equations, plus the constraint equations, would exceed in number the unknown coefficients). In coding for larger problems, the question then arises as to which coefficients to eliminate, and how to perform this elimination systematically, so that the remaining unknowns satisfy equations with some degree of regularity. In the two-dimensional case, where the fluxes must be continuous along an entire line interface, the handling of these constraints gets even messier.

After some deliberation, the entire problem was eliminated by the following trick, which we demonstrate first for the one-dimensional case. Instead of specifying the quadratic in a given region by its 3 coefficients, one can specify it uniquely by its values at 3 points. If two of these points are edge points of the region (the third naturally being the center point), and if the right edge point of one region

is called by the same name as the left edge point of the next region (see sketch 2. below), the two curves are automatically continuous. The only variables to appear, namely the values of the fluxes at each point shown, are all

SKETCH 2.

independent, for the dependent variables have been eliminated merely by giving edge points (e.g., ϕ_2) the same name as seen from either region. One can readily verify by counting that there are just $2N+1$ such points in an N -region problem. With the two outside points determined by boundary conditions, this leaves just $2N-1$ points free to vary, which agrees with the discussion of Section D. For the precursors, which may be discontinuous, we preserve at each edge two independent points, one infinitesimally to the left of the edge, the other just to the right.

With the foregoing "handle" on our quadratics, the variational principle[†] eventually reduces to a set of difference equations for the point values ϕ_n , with the desired high regularity. Since these equations may be of somewhat more general interest, we list here the way the pertinent quantities (see equation 1., and Sketch 2) transform. Since the precursors have two values at an edge, we will add here a second subscript (again increasing to the right) to distinguish them; thus,

let C_{21} be the value of a precursor density at the right edge of Region 1, and C_{22} the value at the left edge of Region 2. Then the following transformation scheme obtains in the flux equations:

At a center point (e.g., 1):

$$\left. \begin{aligned} A\phi &\longrightarrow \frac{A_1}{10} [\phi_0 + 8\phi_1 + \phi_2] \\ \nabla \cdot D \nabla \phi &\longrightarrow \frac{4D_1}{L_1^2} [\phi_0 - 2\phi_1 + \phi_2] \\ AC &\longrightarrow \frac{A_1}{10} [C_{02} + 8C_1 + C_{21}] \end{aligned} \right\} \quad (5.)$$

At an edge point (e.g., 2):

$$\left. \begin{aligned} A\phi &\longrightarrow \frac{1}{5(L_1+L_2)} [A_1L_1(-\phi_0 + 2\phi_1 + 4\phi_2) + A_2L_2(4\phi_2 + 2\phi_3 - \phi_4)] \\ \nabla \cdot D \nabla \phi &\longrightarrow \frac{2}{(L_1+L_2)} \left[\frac{D_1}{L_1} (-\phi_0 + 8\phi_1 - 14\phi_2) + \frac{D_2}{L_2} (-14\phi_2 + 8\phi_3 - \phi_4) \right] \\ AC &\longrightarrow \frac{1}{5(L_1+L_2)} [A_1L_1(C_{02} + 2C_1 + 4C_{21}) + A_2L_2(4C_{22} + 2C_3 - C_{41})] \end{aligned} \right\} \quad (6.)$$

For the precursor equations, which involve no spatial derivatives, we find:

At a center point (e.g., 1):

$$\left. \begin{aligned} A\phi &\longrightarrow \frac{A_1}{10} [\phi_0 + 8\phi_1 + \phi_2] \\ AC &\longrightarrow \frac{A_1}{10} [C_{02} + 8C_1 + C_{21}] \end{aligned} \right\} \quad (7.)$$

At an edge point (e.g., 21):

$$\left. \begin{aligned} A\phi &\longrightarrow \frac{A_1}{5} [-\phi_0 + 2\phi_1 + 4\phi_2] \\ AC &\longrightarrow \frac{A_1}{5} [-C_{02} + 2C_1 + 4C_{21}] \end{aligned} \right\} \quad (8.)$$

[†]Either (3.) for the time-dependent case or Appendix A for the static case, it makes no difference. Results are the same except for the time derivative, which behaves just like another multiplier as far as our present discussion of spatial difference equations is concerned.

Several comments are in order. Note first that A could have been either a multiplier or a time derivative operator, the application of the spatial difference equations proceeding the same in either case. Secondly, since the transformations lead to a homogeneous difference equation at each point, the overall normalization of each set (like 5.) is arbitrary; we have merely tried to choose a normalization meaningful to the reader. On the other hand, in the actual computation, it will be convenient to omit such overall multipliers as $\frac{1}{L_1 + L_2}$ in equation (6.). Finally, since the precursor equations for the 3 points in a particular region involve only the material constants A of that region, and no derivatives, the resultant 3 equations may be recombined in such a way as to solve for each point independently. This means, in effect, that transformations (7.) and (8.) could just as well have been written:

For the precursor equations:

At a center point (e.g., 1):

$$\left. \begin{array}{l} A \phi \longrightarrow A_1 \phi_1 \\ A C \longrightarrow A_1 C_1 \end{array} \right\} \quad (7.')$$

At an edge point (e.g., 21):

$$\left. \begin{array}{l} A \phi \longrightarrow A_1 \phi_2 \\ A C \longrightarrow A_1 C_2 \end{array} \right\} \quad (8.')$$

This simpler, but effectively equivalent, transformation was used in the code, to obviate any iterations on the precursors.

On the other hand, the flux equations (5.) and (6.), even in simplest form, are distinct from the usual difference equations in several respects:

- (1) quite different equations apply to middle points vs. edge points, one involving 2 neighbors, one 4;
- (2) in each difference equation, not only the spatial derivatives, but even simple multipliers, are replaced by expressions involving the values of the flux at several neighboring points;
- (3) these difference equations are to be interpreted, not as giving the best value of the solution at the points themselves, but as giving the best overall solution when quadratics are passed through these points in the specified manner; and as a result,
- (4) the same accuracy as for ordinary difference equations can be achieved with fewer points, as we have seen in E. and F., and will observe further in the next Sections.

From one viewpoint, we have obtained finite difference equations with a pre-specified interpolation, that interpolation being properly weighted into the difference equations themselves. This has enabled us to reduce the usual large, sparse matrix to a smaller but denser one.

Another helpful point of view is to consider each point-value ϕ_n as being the coefficient of a particular space-function. The associated functions are quite easy to identify, by merely setting $\phi_n = \delta_{in}$, where n is the point of interest. Thus each center point controls a function which is a parabola inside the region,

goes to zero at each edge, and is identically zero everywhere outside the region. Each edge point like ϕ_2 (Sketch 2.) controls a function which is zero outside regions 1 and 2, and a separate parabola inside these regions, going through unity at their interface, and zero at their middles (points 1 and 3) and outer edges (0 and 4.).

This same approach is readily applied to the "two-dimensional quadratic", which has 9 independent coefficients, each multiplying a term like $x^i y^j$, where $0 \leq i \leq 2$, $0 \leq j \leq 2$. The corresponding 9 points at which to specify the value of the function are the center point, the 4 corners, and the 4 edge-bisectors (Sketch 3, below). The resultant difference equations are straightforwardly, if somewhat

SKETCH 3

tediously, derived from the variational principle. There are now three different kinds of equations, corresponding to the three different kinds of points in Sketch 3.

The same comments apply to R- $\mathbb{Z}$ geometry. In this case, of course, the difference equations involve not merely the dimensions of a region, but also its distance from the z-axis.

H. Typical Steady-State Problem

Utilizing the foregoing considerations, the problem was now programmed for both X-Y and R-Z options. The code was written to handle both steady-state and kinetics problems. The steady-state section was useful for at least two reasons: at first, it could be used to check the accuracy of our method on large problems (against existing codes like CURE); later, as a permanent part of the program, it could be used to initialize kinetics problems. In this capacity, it is preferable to other steady-state codes, because it is completely consistent in approximation with our kinetics method. Thus a steady-state VARI-QUIR solution, with the multiplication factor adjusted to critical, can be depended upon to give no power rise in the VARI-QUIR kinetics code until some change occurs in the nuclear constants. On the other hand, a "foreign" steady-state solution might lead to small spurious power rises, due merely to differences in approximation employed.

The steady-state difference equations are solved in the code by a combination of "inner" and "outer" iterations. The inner iterations, keeping the fission "source" constant, use Seidel's method, and converge quite rapidly because of the diagonal dominance of the space matrix (see equations 5. & 6.). At each outer iteration, the eigenvalue is re-computed from its stationary form (Appendix A.)

The program was checked out and debugged on several small problems.

For our first large problem, we chose a cylindrical reflected reactor, homogeneous in the z-direction, and with 5 radial regions: core, reflector, reflector, absorbing ring, and more reflector. Four energy groups were used. Again we began with a steady-state problem.

Because the core was rather large compared to the other regions, we broke it into two regions for VARI-QUIR. Two problems were run, the ring present and absent. Results with the ring in (the more complicated flux shape, hence the more difficult for our approximation to match) are shown in Figs. 4-7. The CURE output is plotted as a full line, the VARI-QUIR solution at crucial points marked on with crosses. The fit is everywhere quite good, except for being low in the center of the core in Group 4. Suspecting this might be due to core size, we added still another core region, and got the excellent results marked as circles. Each of our sets of results were obtained in slightly over 2 minutes, as compared to about 16 to 20 minutes for CURE. To be sure, the CURE running time might be cut down by someone more familiar with the code. But at least our steady-state code was in the same general range as CURE.

J. Kinetics - The "Wide Range of Periods" Difficulty

As anticipated in TMI-567, the ICE subroutine was used to integrate the time-differential equations. It operates roughly as follows. To start an integration step, it calculates the derivatives of all the ϕ_n 's at a point

in time, step-integrates to calculate the values of ϕ at the end of the time step, calculates the new derivative there, and then re-integrates the same step using the average derivative. Next, the same step is broken into 2 steps of half the size, the above process used to integrate successively both steps, and the results compared to the single-step case. If the error is too large (as prescribed by the user), these smaller time steps are again halved and the process repeated; if the error is small enough, it is further reduced by combining the 2-step and 1-step results in an extrapolation to zero step size; the code then moves on to the next time step. If the error is below a certain minimum (again prescribed by the user), a larger time step is used on the next step.

The time-dependent code was checked out and debugged on several small problems. As our first large problem, we decided to start from a steady-state problem like that of Section H, but with the absorbing ring absent, then drop in this absorber suddenly at $t = 0$, and watch the fluxes change shape and decay. Results were rather disappointing. In several minutes of computing time, our code had "raced" along to about 10^{-11} sec. of real time!

The trouble was found to be due to the very high neutron velocities in the higher energy flux groups. It showed up only in this "more realistic" problem because our earlier simple problems had used more moderate velocities. To understand the difficulty, we must first look at the qualitative behavior of the fluxes during time changes.

Consider a one-group, one-precursor, space-independent problem, of the type treated by Glasstone and Edlund*. Following a step input of reactivity of less

*Elements of Nuclear Reactor Theory, 3rd ed., Van Nostrand, N.Y., 1955, p.305.

than \$1.00, the flux rise is shown in Fig. 13, Curve A. There occurs first a rapid, short-time rise, during which the flux is adjusting to an equilibrium with the (as yet practically unchanged) precursors. This is followed by a much slower rise along a second period, governed mainly by the slow rise of precursors - and the fast continuous adjustment of the flux to this precursor "source". In terms of the simplified governing equations, the process

$$\frac{d}{dt} \frac{\phi}{V} = [k(1-\beta) - 1] \sigma^{abs.} \phi + \lambda C \quad (9.)$$

$$\frac{dC}{dt} = -\lambda C + \beta k \sigma^{abs.} \phi \quad (10.)$$

is mathematically as follows. Because of the large value of V , and its cross-multiplication into the right-hand side of (9.), the flux time-derivative is quite large unless the right-hand side of (9.) is zero. Therefore the flux jumps rapidly until it reaches the point where the right-hand side of (9.) is practically zero. After this the flux merely continues to rise slowly to maintain this zero, following the slow precursor rise given by equation (10.).

Now, in our more complex problem, there are several energy groups and several precursors, so that there are more than just two periods in the curve corresponding to Fig. 13. Furthermore, all of the flux groups have such high velocities V_g that all the "prompt jumps" occur almost instantaneously ($\sim 10^{-5}$ sec. for thermal to 10^{-9} sec. for the highest group), resulting in the replacement

of the "shoulder" in Curve A, Fig. 13, by a fairly sharp corner. Since ICE automatically slows down when rapid changes are occurring, to preserve accuracy (as explained earlier), this slows down our problem in two ways.

First, it must match the real jump, just described, which occurs near $t = 0$. Secondly, at any later stage in the problem, if, through some error, the right-hand side of (9.) or its equivalent deviates from zero, this will act like another step-change, and ICE will slow the problem down to follow the fast-flux adjustment during this pseudo-step-change. In other words, small errors in the right-hand side of (9.), due to cross-multiplication by the V_g factor, lead to large spurious values for the flux time derivatives. But, because ICE is a step-integrator, these small errors occur all the time - therefore so do the pseudo-steps. The ultimate result is that ICE will slow down enough to handle these steps correctly, but the time required is exorbitant.

The crux of the problem, then, is the tremendous range ($\sim 10^9$) of time constants or periods involved. Notice that this is not peculiar to our method, or even the variational method in general, but is a property of the problem itself under any solution-scheme.

Once isolated, the problem almost suggests its own solution. If it is difficult to solve problems involving all these disparate periods, it would be just as difficult to use the results. One can hardly be simultaneously* interested in time lags of 10^{-9} sec. and of 1 sec. When looking at long times, therefore, we may just as well regard the very fast transients as instantaneous, i.e., the

*True, the time range could be varied, e.g., by plotting flux vs. logarithm of time, to display all these effects on one graph. But, in the large-time region, the small flux lags would still be invisible. Hence the different time constants are not simultaneously of interest.

very high velocities as infinite. This means merely that, instead of allowing the right-hand side of equation (9.) to approach zero very rapidly, we make it identically zero at all times. Thus the left-hand side is replaced by zero, and the new (9.) is now solved as an algebraic equation (therefore no longer in ICE) for ϕ , which adjusts instantaneously to its changing precursor source.

Curves B-D. of Fig. 13 illustrate the method (but not its accuracy!) for the simple problem already considered. Curve B. leaves the flux in ICE for awhile (as one would do if interested in fast transients), then removes the flux from ICE (resulting in a jump to instantaneously follow its precursor source) as one would do to proceed to larger times. Curve D. leaves the flux out of ICE entirely, as one would do if not interested in fast transients. The results at longer times are not independent of the short-time approximation, as they must be if the method is to work; rather, they display a time lag of the order of the fast period. On the other hand, in the real problem we are trying to solve, this fast period is of the order of 10^{-5} to 10^{-9} sec. (depending on energy group considered), and the time lag is then truly negligible at times of ~ 1 sec., so that the approximation should be highly accurate here.

To re-phrase, the sample problem considered here runs perfectly well with the flux left in ICE, and does not require the flux-jump approximation - nor does the approximation work well here. But the method is highly accurate in those very problems where it is needed, namely, where the periods differ by many orders of magnitude.

This approximation was introduced into our code as an option whereby any or all of the faster flux groups can be taken out of ICE. For example, if interested in transients down to 10^{-5} sec. (but not 10^{-9} sec.), one might remove the higher flux groups, but leave the thermal group in. The number of groups in and out can also be changed with time, permitting successively the observation of fast transients and slower behavior. Because this approximation is equivalent to assuming infinitely fast response, we refer to it as taking the fluxes out "above ICE".

There are, on the other hand, precursor groups which change so slowly with time (i.e., have a small λ_i and not too large β_i) that to use the involved ICE error checks on them would be a waste of time. Therefore provision was also made in the program to take out an arbitrary number of precursor groups "below ICE", in the sense that they are treated as constants during an ICE step, and merely updated by a straightforward time integration at the end of each ICE step.

K. A More Detailed Kinetics Problem

With the above modifications included in the code, the problem mentioned at the beginning of Section J. ran smoothly and successfully. Some typical results, using 4 energy groups and 6 precursors (and a 6-region VARI-QUIR mesh) are shown in Figs. 8-11. For times less than 2.7×10^{-5} sec., only the lowest flux group was kept in ICE (all precursors were "below" ICE.) For larger times, all four flux groups were taken out "above" ICE, and only the three fastest-decaying precursor groups were kept in ICE.

Fig. 10 is particularly interesting in that it shows the flux dropping first in the vicinity of the absorber (at very short times); only later is the effect spread over the entire reactor. In other words, one can actually observe the diffusion of thermal neutrons. Figs. 9 and 11. show the overall shape similarity between power distribution and precursor density, for the faster precursors. The total power (Fig. 8) as a function of time displays very clearly the initial prompt jump (when the fluxes are adjusting to equilibrium between the sudden absorption step and the still unchanged precursors), followed by a gradual power decay governed mainly by the precursors.

L. Feedback

Feedback was introduced into the code through three subroutines - to initialize the feedback, to carry it through, and to print out any desired feedback variables at various times. These subroutines are written by the user, so that any desired feedback may be introduced.

The mechanism of this part of the code was checked out on an extremely simple problem - one-group, no precursors, separable, step input in reactivity. The cross-sections were assumed to vary as $(\text{temperature})^{-\frac{1}{2}}$. The temperature itself rose due to heat accumulation, but heat also leaked off in time.

The numbers used, fairly devoid of any physical significance, were chosen merely to yield an "underdamped" case in which, following the step reactivity insertion, the power and temperature made several excursions before settling down to their new values. This seemed a severe test, being probably more extreme than

one might encounter ordinarily. The code worked fine, giving the results shown in Fig. 12.

This feedback section of the program, still in its infancy, will be the subject of much of our efforts during the coming months. The biggest limitation at present is the computer space left in which to write the feedback equations.

II. DESCRIPTION OF CODE

A. Present Limitations

Geometry - XY or RZ.

Boundary conditions - $\phi = 0$ or $\nabla_n \phi = 0$ on each of 4 boundaries.[†]

Energy groups - 4 or less.

Precursor groups - 6 or less.

Flux energy groups "above" ICE - any or all; may change with time.

Precursor groups "below" ICE - same (but at least one group, either flux or precursor, must be in ICE.)

Regions - 36 or less.

Regions in either direction - 7 or less.

Different material regions - 20 or less

Starting options - (1.) starting from scratch, compute steady-state starting fluxes and precursors, then begin kinetics.

(2.) using input flux guess, proceed as in (1.).

(3.) using input flux and precursor specification, begin kinetics immediately.

[†]The physical assumption is that $\phi = 0$ on all 4 boundaries, but advantage is taken of single or double symmetry. For single symmetry, the symmetry axis must be taken as $y = 0$. For R-Z problems, where x becomes z and y becomes r , the symmetry about $r = 0$ is automatic.

Programmed time dependence of nuclear coefficients - any sequence of
of ramps[†] (including steps as special cases) in any of all coefficients.

Feedback - arbitrary, written by user in three subroutines, to initialize
it, carry it through, and print it out. Space very limited at
present.

Accuracy limits - optional (suggest 6×10^{-5} min. error, 7×10^{-3} max.)

Successive problems - any number may be stacked.

All of the above may be changed, with varying degrees of effort. The easiest
will probably be to include linear extrapolation distance specification as a
boundary condition.

[†]Ramps in time are extremely time-saving because the nuclear constants need
be reduced to the form Agg' (see equation 1.) only at the ends of each ramp,
and at intermediate times the quantity Agg' can be calculated linearly from
its end values. The only exception is that the group velocities, while allowed
to be time varying ramps, may not be step-functions. This is obviously not a
practical limitation.

INPUT PREPARATION FOR TIME DEPENDENT VARI-QUIR

Card No. 1

This card may contain any alphanumerical information that will adequately describe the problem. It is printed at the beginning of the output, following the problem number.

Card No. 2

This card provides information which defines the problem in terms of overall structure.

Punch in columns --

1-4	NPROB	Problem number
8	NG	Number of groups
12	NJ	Number of precursors
15-16	NM	Number of materials
20	NX	Number of X (or Z)* regions
24	NY	Number of Y (or R) regions
27-28	NGEOM	Geometry specification
		NGEOM = -2: XY double symmetry
		= -1: XY single symmetry
		= 0: XY no symmetry
		= 1: RZ single symmetry
		= 2: RZ double symmetry
31-32	NGES	Initial flux option
		NGES = -1: Initial fluxes and precursors are computed from scratch.
		= 0: Initial fluxes and precursors are computed from input flux guess.
		= 1: Precise initial fluxes and precursors are read into the program.
36	INTPX	This is the number of interpolated points given in the printout in addition to the three points per region in the X or Z direction, e.g. if INTPX is 1, the program will print out five points per region in the X or Z direction. One interpolated point between each regular point. Number of points per region = 2*INTPX+3.
40	INTPY	The same as above only in the Y or R direction.

*This is dependent upon geometry specification.

The following are only required when the initial fluxes are computed in the program (NGES = -1, or 0).

44	LAMB	$\neq 0$: Eigenvalue found in steady-state solution is multiplied through all λ 's to make kinetics starting conditions just critical.* $= 0$: λ 's are left as is (non-critical starting conditions).
45-48	ITOUT	Number of outer iterations on the initial flux computation.
49-52	ITINN	Number of inner iterations per outer iteration on the initial flux computation.
55-56	ITPRNT	Number of outer iterations between printouts.
57-60	ISSW	$= > 0$: This permits one to terminate the steady-state flux iterations before ITOUT is exceeded, by putting down sense switch No. 1.** The code will then go into the time dependent part.

Card Group No. 3

The X (or Z) - lengths of each region, starting from the left (where X = 0), is listed, using successive 10-column fields.

Card Group No. 4

The Y (or R) - lengths of each region, starting from the bottom (Y = 0) is listed as No. 3.

Card Group No. 5

These cards list the material number for each region, using one Y-row per card and as many cards as there are Y regions. On each card, the material numbers for the X regions, starting from the left, are listed in successive 4-column fields.

* So that all subsequent power changes are effects only of changes deliberately made in the nuclear constants.

**One determines this by the on-line eigenvalue print out.

The next four card groups are the nuclear constants for each material.

Card Group No. 6

This card group is the fission spectrum for each precursor group. The format is (5E15.8). Start a new card for each energy group.

Example-- DO 3 IG = 1, NG
SI(IG, IJ), IJ = 1, NJ

Card Group No. 7

This card group is the fraction β and the decay constant λ for each precursor group. Format is (5E15.8).

(BETAL(IJ), XLAML(IJ), IJ = 1,NJ)

Card Group No. 8

SABSI(IG) is $\sum_a$ for group IG of material IM.
SFIL(IG,IM) is $\sum_f$ for group IG of material IM.
XNUL(IG) is γ for group IG of material IM.
CHIL(IG,IM) is χ for group IG of material IM.
DI(IG,IM) is D for group IG of material IM.
STRI(IG,IGP) is $\sum^{TR}$ into group IG of material IM.

These nuclear constants may be generated with a computer code, called GAM-TNS, which punches them on cards with the proper format for VARI-QUIR. If one chooses to have these nuclear constants punched by hand, they should refer to the program.

Card Group No. 9

This card group is the velocities of each energy group. Format is (5E15.8).

$$(V(IG), IG = 1, NG)$$

Now, card groups 6, 7, 8 and 9 should be repeated for each different material.

Card Group No. 10 (Omitted if NGES = -1)

These cards contain the fluxes to either exactly specify starting conditions (if NGES = +1) or to merely furnish a good guess for the steady-state computation (if NGES = 0). They are listed in 15-column fields, one energy group (several cards) at a time. Within each group, the flux points are counted from the left, a row at a time. There are $2NY + 1$ rows and $2NX + 1$ values per row, where NX and NY are the numbers of X- and Y-regions.

Card Group No. 11 (Omitted if NGES = 0 or -1)

These cards list the starting precursors when that option (NGES = 1) is chosen. Format resembles Card Group No. 10 except that there are now $3NY$ rows and $3NX$ values per row, for each group.

Card Group No. 12

Specifies the starting time (e.g., $t = 0.0$) in the first 10-column field.

Card Group No. 13 (repeated for each time-ramp in the nuclear parameters)

For each time-ramp, these cards specify the length of the ramp, which nuclear parameters change during this time, and what values they reach (starting from present values) at the end of the ramp. Correspondingly, this group is made up of one 13.1 card, and as many pairs of 13.2 and 13.3 cards (a pair at a time) as are necessary to specify all the changes, 8 to a card.

Card 13.1

Punch columns --		
1-10	DT	Time-duration of this ramp.
11-20	PP	Time interval between solution printouts, within this ramp.
21-30	E1	Lower bound error for controlling ICE stepsize, (6.0 E-5 or greater).
31-40	E2	Upper bound error for controlling ICE stepsize (7.0 E-3 or greater; $E2 = 2^7 E1$)
41-50	E3	Error control for fluxes that are out of ICE.
51-60	E4	Error control for fluxes that are in ICE ($E3 = E4 = E1$).
64	NSTOP	= 2 if this is the final ramp of a problem. ≠ 2 otherwise.
		Controls transfer to next problem.
65-68	NCHGE	Number of nuclear parameters to be changed at the beginning of this ramp.
72	NORM	= 1 if it is desired to normalize fluxes at the end of this time ramp.
76	NGOUT	Number of flux groups taken out "above" ICE (see Part I, Section J of WANL-TNR-133).
80	NJOUT	Number of precursor groups taken out "below" ICE.

Card 13.2

List in 10-column fields (fixed point) the W-indices of the nuclear parameters to be "ramped". The W variable is a one-dimensional array that is in EQUIVALENCE with the nuclear parameter (ν , Σ_a , etc.).

Card 13.3

List in 10-column fields (floating point) the values to which these parameters have changed by the end of this ramp.

The above data cards are sufficient to use this code for "programmed reactivity" problems, e.g., problems where the behavior of the nuclear parameters is a pre-specified function of time. For feedback problems, the user must put in the desired feedback equations in three FORTRAN subroutines, as explained in Chapter I, Section L. and Chapter II Section A of WANL-TNR-133. The use of these feedback subroutines and some additions to the code are explained in Appendix D.

FIGURE 1 FIRST TRIAL OF VARI-QUIR METHOD
(STEADY-STATE SLAB PROBLEM)

FLUX IN TWO CRITICAL SLABS, EACH OF THICKNESS L ,
 BROUGHT TOGETHER AT $t = 0$, AS A FUNCTION OF DISTANCE
 X FROM THEIR INTERFACE (Figures 2-3)

FIGURE 2 ALMOST-SEPARABLE APPROXIMATION

FLUX IN TWO CRITICAL SLABS, EACH OF THICKNESS L ,
BROUGHT TOGETHER AT $t = 0$, AS A FUNCTION OF DISTANCE
 X FROM THEIR INTERFACE (Figures 2-3)

FIGURE 3 VARI-QUIR APPROXIMATION

FIVE-RADIAL-REGION FOUR-GROUP STEADY-STATE PROBLEM
(Figures 4-7)

FIGURE 4 GROUP 1 FLUX

CURVE NO. 595432

RADIAL POSITION

FIVE-RADIAL-REGION FOUR-GROUP STEADY-STATE PROBLEM
(Figures 4-7)

FIGURE 5 GROUP 2 FLUX

FIVE-RADIAL-REGION FOUR-GROUP STEADY-STATE PROBLEM
(Figures 4-7)

FIGURE 6 GROUP 3 FLUX

FIVE-RADIAL-REGION FOUR-GROUP STEADY-STATE PROBLEM

FIGURE 7 GROUP 4 FLUX

FIVE-REGION FOUR-GROUP KINETICS
(Figures 8-11)

FIGURE 8 TOTAL POWER VS. TIME

FIVE-REGION FOUR-GROUP KINETICS
(Figures 8-11)

FIGURE 9 POWER DENSITY VS. POSITION

RADIAL POSITION

FIVE-REGION FOUR-GROUP KINETICS
(Figures 8-11)

5958388

FIGURE 10 GROUP 4 FLUX

RADIAL POSITION

CURVE NO 595434

FIVE-REGION FOUR-GROUP KINETICS
(Figures 8-11)

FIGURE 11 GROUP 1 (FASTEST DECAY) PRECURSORS

FIGURE 12 SIMPLE FEEDBACK TEST PROBLEM

595837A

FIGURE 13 ILLUSTRATIVE TWO-PERIOD TIME PROBLEM
(ONE ENERGY GROUP, ONE PRECURSOR, REACTIVITY
STEP INSERTED AT $t = 0$)

Appendix A. - The Steady-State Variational Principle

In this problem the left-hand side of equation (1.) is zero, the A-matrix breaks up into removal A' and fission B parts, and the problem is to find the overall factor k by which the fission matrix B must be multiplied if the reactor is to be just critical. Mathematically, we wish to find the lowest eigenvalue k of

$$g = 1 \text{ to } G: \quad \nabla \cdot D_g \nabla \phi_g + \sum_{g'=1}^G [A'_{gg'} \phi_{g'} + k B_{gg'} \phi_{g'}] = 0 \quad (11.)$$

Removal of the time-dependence for the precursors has reduced their equations to explicit solutions in terms of fluxes, allowing the precursors to be eliminated; hence g only goes from 1 to G in equation (1.).

If one knew the exact solution ϕ_g of equation (11.), k could be found by solving for it in any one of these G equations, at any point in space. If one has only an approximate solution for the ϕ_g 's, the best estimate of k would be obviously attained by somehow averaging equation (11.) over all space and all G equations, and then solving for k. Calling the weighting function for this averaging process ϕ_g^* , one obtains:

$$k = \frac{P}{Q} \quad (12.-a.)$$

$$\text{where } P = \sum_{g=1}^G \int d^3r \left\{ D_g (\nabla \phi_g^*) \cdot (\nabla \phi_g) - \sum_{g'=1}^G \phi_g^* A'_{gg'} \phi_{g'} \right\} \quad (12.-b.)$$

$$\text{and } Q = \sum_{g=1}^G \sum_{g'=1}^G \int d^3r \phi_g^* B_{gg'} \phi_{g'} \quad (12.-c.)$$

A good value of the eigenvalue k will thus be calculable from equations (12.) as soon as we can get some estimate of the solution ϕ_f and the proper weighting function ϕ^* .

The correct weighting function will, by definition, be the one which gives very small errors δk in k when our estimate of ϕ errs by $\delta\phi$; i.e., $\frac{\delta k}{\delta\phi} = 0$ gives us a criterion for ϕ^* . On the other hand, we already know by the way we set up (11.) that any weighting function would work if we knew ϕ exactly, i.e., k is independent of ϕ^* when ϕ is correct, or $\frac{\delta k}{\delta\phi^*} = 0$ can be used to define ϕ^* . Combining these two results, we see that simultaneous first-order variations in ϕ and ϕ^* will lead to only second-order variations in k , through the product $\delta\phi \cdot \delta\phi^*$. Therefore k is stationary for small variation in ϕ and ϕ^* , i.e.,

$$\delta k = 0 \quad (12.-d.)$$

Equation (12.-d.), along with the definitions (12.-a.) to (12.-c.), is the required variational principle. It can easily be shown to lead to two Euler equations, one being just (11.), the other a similar equation with all operators replaced by their adjoints and ϕ by ϕ^* .

Appendix B -

FORTTRAN LISTING AND TEST PROBLEM

*CLASS 502

* XEQ

* CHAIN (1,4)

C TWO-DIM. VARI-QUIR KINETICS J.W. RIESE AND G. CCLLIER

C PROGRAMMED NUCLEAR CONSTANTS

C

```

      DIMENSION P(3,3),Q(3,3),PR(3,3),QR(3,3),M(7,7),DX(7),DY(7),PRR(3,3)
      1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20),
      2SFI1(4,20),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20),
      3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4,
      4,4,20),SABS2(4,20),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,
      520),BETA2(6,20),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20),
      6A3(4,4,20),AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20)
      7),VD3(4,20),W1(2800),W2(2800),Y(2620),YD(2620),YDS( 676),X5N( 6780
      8),BTOT1(20),B(4,4,20),DEN(676),YS(1690)

```

C

```

      EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)
      1,(X5N(2121),BTOT1),(X5N(3981),W1,S1),(X5N(4401),STR1),(X5N(4781),
      2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510
      31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550
      41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1),
      5(W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2),
      6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128
      71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)
      8,ADF2)

```

C

```

      COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YD,YDS,M,DX,DY,P,Q,PR
      1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,
      2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW

```

C

```

      COMMON DEN,NGECM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,
      1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUMD,SUMS,KIT,KKIT,KRETC
      2,E1,E2,E3,E4,ITINN,ITOUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW

```

C

```

91 TIME = CLOCKF(CARD,CHGE)
   DO 1 I = 1,3
   DO 1 J=1,3
   S=32-28*((I-2)**2+(J-2)**2)+5*(I-2)*(J-2)+27*(I-2)**2*(J-2)**2
   P(I,J)=S/2.0
   Q(I,J)=(16-24*((I-2)**2+(J-2)**2)+(I-2)*(J-2)*(3+36*(I-2)*(J-2)))*
110
   SR=(4-(I-2)*(J-2))*(I+J-4)
   PR(I,J)=SR/4.0
   1 QR(I,J)=((3*(I-2)*(J-2)-2)*(I+J-4))*10
899 READ INPLT TAPE 5,193,(W2(I),I=1,14)
193 FORMAT(13A6,A2)
   READ OUT TAPE 5,100,NPROB,NG,NJ,NM,NX,NY,NGECM,NGES,INTPX,INTPY

```

1, LAMB, ITCUT, ITINN, ITPRNT, ISSW	CH1	45
100 FORMAT(2CI4)	CH1	46
NX2 = 2*NX	CH1	47
NPX = NX2 +1	CH1	48
NY2 = 2*NY	CH1	49
NPY = NY2 +1	CH1	50
NGP1= NG +1	CH1	51
NPXJ = 3*NX	CH1	52
NPYJ = 3*NY	CH1	53
NPJ = NPXJ*NPYJ	CH1	54
NP = NPX*NPY	CH1	55
N = NP*NG +NPJ*NJ	CH1	56
NNN=NG*NP+1	CH1	57
NCARD = CARD	CH1	58
NREM = CHGE	CH1	59
WRITE OUTPUT TAPE 6,101,NPROB,TIME,NCARD,NREM	CH1	60
101 FORMAT(12H1PRCBLEM NO.14,45X16HCOMPUTER TIME = -3PF6.3,5XI2,1H/I2,	CH1	61
13H/64)	CH1	62
WRITE OUTPUT TAPE 6,192,(W2(I),I=1,14)	CH1	63
192 FORMAT(1H 13A6,A2///)	CH1	64
WRITE OUTPUT TAPE 6,102,NG,NJ,NX,NY,NM	CH1	65
102 FORMAT(6X7HGRCUPS=I2,6X11HPRECURSORS=I2,6X1CHX REGIONS=I2,6X16HY(O	CH1	66
1R R) REGIONS=I2,6X10H MATERIALS=I2//)	CH1	67
87 IF(NGEOM)3,3,2	CH1	68
2 WRITE OUTPUT TAPE 6,103	CH1	69
103 FORMAT(6X6CHTHIS PROBLEM HAS CYLINDRICAL COORDINATES X,R GEOMETRY	CH1	70
1. /6X4CHTHE SYMBOL Y REPRESENTS R THROUGH CUT.)	CH1	71
6 IF(NGES)199,4,5	CH1	72
199 WRITE OUTPUT TAPE 6,117	CH1	73
117 FORMAT(6X42HINITIAL FLUXES ARE COMPUTED FROM SCRATCH.)	CH1	74
GO TO 7	CH1	75
4 WRITE OUTPUT TAPE 6,104	CH1	76
104 FORMAT(6X51HINITIAL FLUXES ARE COMPUTED FROM INPUT FLUX GUESS.)	CH1	77
GO TO 7	CH1	78
3 WRITE OUTPUT TAPE 6,105	CH1	79
105 FORMAT(6X6CHTHIS PROBLEM HAS RECTANGULAR COORDINATES X,Y GEOMETRY	CH1	80
1.)	CH1	81
GO TO 6	CH1	82
5 WRITE OUTPUT TAPE 6,106	CH1	83
106 FORMAT(6X6CHINITIAL FLUXES ARE READ IN TO PROGRAM IN THIS PROBLEM.	CH1	84
1)	CH1	85
7 READ INPUT TAPE 5,107,(DX(IX),IX=1,NX)	CH1	86
107 FORMAT(8E10.4)	CH1	87
WRITE OUTPUT TAPE 6,123	CH1	88
123 FORMAT(1H)	CH1	89
WRITE OUTPUT TAPE 6,108,((IX,DX(IX)),IX=1,NX)	CH1	90
108 FORMAT(6X11HDELTA X NO.13,2H=1PE14.7)	CH1	91
READ INPUT TAPE 5,107,(DY(IY),IY=1,NY)	CH1	92

IF(NGEOM)8,8,9	CH1 93
8 WRITE OUTPUT TAPE 6,123	CH1 94
WRITE OUTPUT TAPE 6,109,((IY,DY(IY)),IY=1,NY)	CH1 95
109 FORMAT(6X11HDELTA Y NO.13,2H =1PE14.7 ,	CH1 96
DO 88 IY=1,NY	CH1 97
DO 88 I1=1,3	CH1 98
DO 88 I2=1,3	CH1 99
PRR(I1,I2,IY)=P(I1,I2)*DY(IY)	CH1 100
88 QRR(I1,I2,IY)=Q(I1,I2)/DY(IY)	CH1 101
GO TC 9001	CH1 102
9 WRITE OUTPUT TAPE 6,123	CH1 103
WRITE OUTPUT TAPE 6,110,((IY,DY(IY)),IY=1,NY)	CH1 104
110 FORMAT(6X11HDELTA R NO.13,2H =1PE14.7)	CH1 105
RAV= DY(1)/2.C	CH1 106
DO 11 I1=1,3	CH1 107
DO 11 I2=1,3	CH1 108
PRR(I1,I2,1)=RAV*DY(1)*P(I1,I2)+PR(I1,I2)*DY(1)**2	CH1 109
11 QRR(I1,I2,1)=RAV*Q(I1,I2)/DY(1)+QR(I1,I2)	CH1 110
IF(NY-1)9001,9001,12	CH1 111
12 DO 98 IY=2,NY	CH1 112
RAV= RAV + 0.5*(DY(IY)+DY(IY-1))	CH1 113
DO 98 I1=1,3	CH1 114
DO 98 I2=1,3	CH1 115
PRR(I1,I2,IY)=RAV*DY(IY)*P(I1,I2)+PR(I1,I2)*DY(IY)**2	CH1 116
98 QRR(I1,I2,IY)=RAV*Q(I1,I2)/DY(IY)+QR(I1,I2)	CH1 117
9001 DO 9002 I = 1,3	CH1 118
PR(I) = C.C	CH1 119
9002 QR(I) = C.C	CH1 120
IF(NGECM)9003,10,9003	CH1 121
9003 DO 9004 I = 1,3	CH1 122
S = 44 - 50*I + 12*I**2	CH1 123
PR(I) = S*PRR(1,I,1)/3.0	CH1 124
9004 QR(I) = S*QRR(1,I,1)/3.0	CH1 125
PRR(2,2,1) = PRR(2,2,1) + 8.0*PRR(1,2,1)/3.0	CH1 126
PRR(2,3,1) = PRR(2,3,1) + (4.0*PRR(1,3,1) -PRR(1,2,1))/3.0	CH1 127
PRR(3,2,1) = PRR(2,3,1)	CH1 128
PRR(3,3,1) = PRR(3,3,1) - 2.0*PRR(1,3,1)/3.0	CH1 129
PRR(1,3,1) =-PRR(1,1,1)/3.0	CH1 130
PRR(1,2,1) = 4.0*PRR(1,1,1)/3.0	CH1 131
PRR(2,1,1) = PRR(1,2,1)	CH1 132
PRR(3,1,1) = PRR(1,3,1)	CH1 133
PRR(1,1,1) =-PRR(1,1,1)	CH1 134
QRR(2,2,1) = QRR(2,2,1) + 8.0*QRR(1,2,1)/3.0	CH1 135
QRR(2,3,1) = QRR(2,3,1) + (4.0*QRR(1,3,1) -QRR(1,2,1))/3.0	CH1 136
QRR(3,2,1) = QRR(2,3,1)	CH1 137
QRR(3,3,1) = QRR(3,3,1) - 2.0*QRR(1,3,1)/3.0	CH1 138
QRR(1,3,1) =-QRR(1,1,1)/3.0	CH1 139
QRR(1,2,1) = 4.0*QRR(1,1,1)/3.0	CH1 140

```

GRR(2,1) = GRR(1,2,1)
GRR(3,1,1) = GRR(1,3,1)
GRR(1,1,1) = -GRR(1,1,1)
10 WRITE OUTPUT TAPE 6, 123
DO 13 IY = 1, NY
  READ INPUT TAPE 5, 100, (M(IX,IY), IX=1, NX)
  DO 13 IX=1, NX
    13 WRITE OUTPUT TAPE 6, 112, IX, IY, M(IX, IY)
112 FORMAT(6X15HMATERIAL NO. M(I2,1H,I2,3H) =I3 )
    WRITE OUTPUT TAPE 6, 123
    WRITE OUTPUT TAPE 6, 128
128 FORMAT(3X9HLAYOUT--- //6X9HT = TIME // 9X13HFLUX GROUP 1 )
    W1(1) = C.C
    NROW = 2*INTPY +3
    NCOL = 2*INTPX +3
    IL = NCOL*NX
    IG = NROW*NY
    W2(1) = C.C
    DO 196 I = 1, NX
      S = NCOL -1
      S = DX(I)/S
      INDX = (I-1)*NCOL
      DO 5012 J = 1, NCOL
        INDX = INDX +1
5012 W1(INDX+1) = W1(INDX) +S
        INDX = I*NCOL
      196 W1(INDX+1) = W1(INDX)
      DO 197 J = 1, NY
        S = NROW -1
        S = DY(J)/S
        INDX = (J-1)*NROW
        DO 5013 I = 1, NROW
          INDX = INDX +1
5013 W2(INDX+1) = W2(INDX) +S
          INDX = J*NROW
        197 W2(INDX+1) = W2(INDX)
        WRITE OUTPUT TAPE 6, 136, W2(1), (W1(I), I = 1, IL )
136 FORMAT(13X4HY = 1PE14.7, 3X4HX = 1P5E16.7/(38X1P5E16.7))
        WRITE OUTPUT TAPE 6, 137, W2(2), W1(1), (W2(I), I=3, IG )
137 FORMAT(13X4HY = 1PE14.7, 3X4HX = 1PE16.7, 4X4HETC./ (13X4HY = 1PE14.7
1))
      WRITE OUTPUT TAPE 6, 131
131 FORMAT(9X13HFLUX GROUP 2 / 13X4HETC. / 9X4HETC. ///)
      DO 14 IM=1, NM
        DO 900 IG=1, NG
900 READ INPUT TAPE 5, 801, ( S1(IG, IJ, IM), IJ=1, NJ )
      READ INPUT TAPE 5, 801, ( BETA1(IJ, IM), XLAM1(IJ, IM), IJ=1, NJ )
      DO 901 IG=1, NG

```

CH1 141
CH1 142
CH1 143
CH1 144
CH1 145
CH1 146
CH1 147
CH1 148
CH1 149
CH1 150
CH1 151
CH1 152
CH1 153
CH1 154
CH1 155
CH1 156
CH1 157
CH1 158
CH1 159
CH1 160
CH1 161
CH1 162
CH1 163
CH1 164
CH1 165
CH1 166
CH1 167
CH1 168
CH1 169
CH1 170
CH1 171
CH1 172
CH1 173
CH1 174
CH1 175
CH1 176
CH1 177
CH1 178
CH1 179
CH1 180
CH1 181
CH1 182
CH1 183
CH1 184
CH1 185
CH1 186
CH1 187
CH1 188

```

901 READ INPUT TAPE 5,801,SABS1(IG,IM),SF11(IG,IM),XNU1(IG,IM),CHI1(IG,IM) 189
    1,IM),C1(IG,IM), ( STR1(IG,IGP,IM), IGP=1,NG ) CH1 190
    READ INPUT TAPE 5,801,(V1(IG,IM),IG=1,NG) CH1 191
    BTOT1(IM)=C.0 CH1 192
    DO 14 IJ=1,NJ CH1 193
14  BTOT1(IM)= BTOT1(IM)+ BETA1(IJ,IM) CH1 194
    CALL PRINT CH1 195
    DO 97 IM=1,NM CH1 196
    DO 96 IG=1,NG CH1 197
    DO 64 IGP=1,NG CH1 198
64  A1(IG,IGP,IM)= -STR1(IG,IGP,IM) CH1 199
    A1(IG,IG,IM)= A1(IG,IG,IM) + SABS1(IG,IM) CH1 200
    SUM = C.0 CH1 201
    DO 194 IJ = 1,NJ CH1 202
194 SUM = SUM + S1(IG,IJ,IM)*BETA1(IJ,IM) CH1 203
    DO 63 IGP=1,NG CH1 204
    A1(IG,IG,IM)= A1(IG,IG,IM) +STR1(IGP,IG,IM) CH1 205
63  B(IG,IGP,IM)=((1.C-BTOT1(IM))*CHI1(IG,IM)+SUM)*XNU1(IGP,IM)*SF11(IG,IM) CH1 206
    IGP,IM) CH1 207
96  CONTINUE CH1 208
97  CONTINUE CH1 209
    IF(NGES)17,15,15 CH1 210
15  DO 16 IG=1,NG CH1 211
    K1= (IG-1)*NP +1 CH1 212
    K2= IG*NP CH1 213
16  READ INPUT TAPE 5,801,(Y(I),I=K1,K2) CH1 214
    IF(NGES)17,20,86 CH1 215
86  DO 85 IJ=1,NJ CH1 216
    K1=NG*NP+(IJ-1)*NPJ+1 CH1 217
    K2=K1+NPJ-1 CH1 218
85  READ INPUT TAPE 5,801,(Y(I),I=K1,K2) CH1 219
    GO TO 20 CH1 220
17  DO 18 IP=1,NP CH1 221
    DO 18 IG=1,NG CH1 222
    I = (IG-1)*NP +IP CH1 223
18  Y(I) = 1.C CH1 224
20  DO 21 IG=1,NG CH1 225
    DO 22 IY=1,NPY CH1 226
    J = (IG-1)*NP + (IY-1)*NPX +1 CH1 227
    K = J + NPX-1 CH1 228
    Y(J) = 0.C CH1 229
22  Y(K) = 0.C CH1 230
    DO 21 IX=1,NPX CH1 231
    J = (IG-1)*NP +IX CH1 232
    K = IG*NP+1-IX CH1 233
    Y(J) = 0.C CH1 234
21  Y(K) = 0.C CH1 235
70  J = N*OM**2 CH1 236

```

```

      IF(J-1)25,72,24
24  DO 71 IG=1,NG
      DO 71 IY=1,NPY
      K = (IG-1)*NP + (IY-1)*NPX + 1
71  Y(K)= (4.*Y(K+1)-Y(K+2))/3.0
72  DO 73 IG=1,NG
      DO 73 IX=1,NPX
      K=(IG-1)*NP + IX
      K1=K+NPX
      K2=K1+NPX
73  Y(K)= (4.*Y(K1)-Y(K2))/3.0
25  IF(NGES)200,200,26
200 TIME = CLOCKF(CARD,CHGE)
      WRITE OUTPUT TAPE 6,8C02,ITOUT,ITINN,ITPRNT,ISSW
8002 FORMAT( /// 7HCITCUT=I3, 5X6HITINN=I3, 5X7HITPRNT=I2,
1  5X5HISSW=I2 //)
      WRITE OUTPUT TAPE 6,8C4,TIME
804 FORMAT(77HOENTRY TO THE STEADY STATE.
1  COMPUTER TIME = -3PF6.3 )
C TWO-DIM. VARI-QUIR STEADY STATE
      NN= NG*NP
113 FORMAT(1P5E20.7)
119 FORMAT(5X14HFLUX GROUP NO.I3 )
120 FORMAT(1F )
      DO 201 J = 1,NN
201  YD(J) = Y(J)
      KTOUT = C
      KTPRNT = C
      GO TO 350

C
C      EIGENVALUE CALCULATION
300 XUM1 = 0.0
      XUM2 = 0.0
      DEN1 = 0.0
      DO 312 I = 1,NX
      DO 311 J = 1,NY
      IM = M(I,J)
      DO 310 IG = 1,NG
      INDX = (IG-1)*NP + (2*J-1)*NPX + 2*1
      SUM = 0.0
      DO 304 I1 = 1,3
      DO 303 I2 = 1,3
      DO 302 J1 = 1,3
      DO 301 J2 = 1,3
      INDX1 = INDX + (J1-2)*NPX + I1-2
      INDX2 = INDX + (J2-2)*NPX + I2-2
301 SUM = SUM + YD(INDX1)*Y(INDX2)*{PRR(J1,J2,J)*Q(I1,I2)/DX(I)**2 +
1  P(I1,I2)*QRR(J1,J2,J)}

```

CH1 237
CH1 238
CH1 239
CH1 240
CH1 241
CH1 242
CH1 243
CH1 244
CH1 245
CH1 246
CH1 247
CH1 248
CH1 249
CH1 250
CH1 251
CH1 252
CH1 253
CH1 254
CH1 255
CH1 256
CH1 257
CH1 258
CH1 259
CH1 260
CH1 261
CH1 262
CH1 263
CH1 264
CH1 265
CH1 266
CH1 267
CH1 268
CH1 269
CH1 270
CH1 271
CH1 272
CH1 273
CH1 274
CH1 275
CH1 276
CH1 277
CH1 278
CH1 279
CH1 280
CH1 281
CH1 282
CH1 283
CH1 284

302	CONTINUE	CH1	285
303	CONTINUE	CH1	286
304	CONTINUE	CH1	287
	XUM2 = XUM2 + SUM*DX(I)*D1(IG,IM)	CH1	288
	DO 309 IGP = 1,NG	CH1	289
	INDX3 = INDX + (IGP-IG)*NP	CH1	290
	SUM = 0.0	CH1	291
	DO 308 I1 = 1,3	CH1	292
	DO 307 I2 = 1,3	CH1	293
	DO 306 J1 = 1,3	CH1	294
	DO 305 J2 = 1,3	CH1	295
	INDX1 = INDX + (J1-2)*NPX + I1-2	CH1	296
	INDX2 = INDX3 + (J2-2)*NPX + I2-2	CH1	297
305	SUM = SUM + YC(INDX1)*Y(INDX2)*P(I1,I2)*PRR(J1,J2,J)	CH1	298
306	CONTINUE	CH1	299
307	CONTINUE	CH1	300
308	CONTINUE	CH1	301
	XUM1 = XUM1 + SUM*DX(I)*A1(IG,IGP,IM)	CH1	302
309	DEN1 = DEN1 + SUM*DX(I)*B(IG,IGP,IM)	CH1	303
310	CONTINUE	CH1	304
311	CONTINUE	CH1	305
312	CONTINUE	CH1	306
	AMB = (XUM1 + XUM2)/DEN1	CH1	307
	KTPRNT = KTPRNT + 1	CH1	308
	IF(ITPRNT-KTPRNT)313,313,650	CH1	309
313	KTPRNT = 0	CH1	310
	GO TO 600	CH1	311
C		CH1	312
C	DENOMINATOR CALCULATION	CH1	313
350	DO 356 IG = 1,NG	CH1	314
	DO 356 I = 1,NX	CH1	315
	DO 356 J = 1,NY	CH1	316
	INDX = (IG-1)*NP + (2*J-1)*NPX + 2*I	CH1	317
	IM = M(I,J)	CH1	318
	DEN(INDX) = 16.0*PRR(2,2,J)*A1(IG,IG,IM) + D1(IG,IM)*(16.0*QRR(2,	CH1	319
	12,J) + PRR(2,2,J)*Q(2,2)/DX(I)**2)	CH1	320
	IF(NX-I)352,352,351	CH1	321
351	IM1 = M(I+1,J)	CH1	322
	DEN(INDX+1) = PRR(2,2,J)*4.0*(A1(IG,IG,IM)*DX(I) + DX(I+1)*A1(IG,	CH1	323
	1IG,IM1)) + QRR(2,2,J)*4.0*(D1(IG,IM)*DX(I)+DX(I+1)*D1(IG,IM1))+	CH1	324
	2PRR(2,2,J)*(Q(3,3)*D1(IG,IM)/DX(I) + Q(1,1)*D1(IG,IM1)/DX(I+1))	CH1	325
352	IF(NY-J)355,355,353	CH1	326
353	IM2 = M(I,J+1)	CH1	327
	INDX1 = INDX + NPX	CH1	328
	DEN(INDX1)=16.0*(PRR(3,3,J)*A1(IG,IG,IM)+PRR(1,1,J+1)*A1(IG,IG,IM2	CH1	329
	1))+16.0*(QRR(3,3,J)*D1(IG,IM)+QRR(1,1,J+1)*D1(IG,IM2))+Q(2,2)*(PRR	CH1	330
	2(3,3,J)*D1(IG,IM)+PRR(1,1,J+1)*D1(IG,IM2))/DX(I)**2	CH1	331
	IF(NY-I)355,355,354	CH1	332


```

354 IM3 = M(I+1,J+1)                                CH1 333
      DEN(INDX1+1)=PRR(3,3,J)*(A1(IG,IG,IM)*P(3,3)*DX(I)+A1(IG,IG,IM1)* CP1 334
      1P(1,1)*DX(I+1))+4.0*PRR(1,1,J+1)*(A1(IG,IG,IM2)*DX(I) +A1(IG,IG,IMCH1 335
      23)*DX(I+1))+4.0*QRR(3,3,J)*(D1(IG,IM)*DX(I)+D1(IG,IM1)*DX(I+1)) CH1 336
      3+4.0*QRR(1,1,J+1)*(D1(IG,IM2)*DX(I)+D1(IG,IM3)*DX(I+1))+PRR(3,3,J)CH1 337
      4*(Q(3,3)*D1(IG,IM)/DX(I)+Q(1,1)*D1(IG,IM1)/DX(I+1))+PRR(1,1,J+1)* CH1 338
      5(Q(3,3)*D1(IG,IM2)/DX(I)+Q(1,1)*D1(IG,IM3)/DX(I+1)) CH1 339
355 CONTINUE                                         CH1 340
356 CONTINUE                                         CH1 341
      GO TO 202                                       CH1 342
C                                                     CH1 343
C SOURCE CALCULATION                                CH1 344
400 SIZE = C.C                                       CH1 345
      SIZED = C.C                                     CH1 346
      DO 415 IG = 1,NG                               CH1 347
      DO 415 I = 1,NX                                 CH1 348
      DO 415 J = 1,NY                                 CH1 349
      INDX=(IG-1)*NP+(2*J-1)*NPX+2*I                 CH1 350
      IM=M(I,J)                                       CH1 351
      SUM = C.C                                       CH1 352
      SUMD = 0.C                                       CH1 353
      DO 402 IGP = 1,NG                               CH1 354
      INDX1 = INDX + (IGP-IG)*NP                     CH1 355
      SUM1 = 0.C                                       CH1 356
      SUM2 = 0.C                                       CH1 357
      DO 401 I1 = 1,3                                 CH1 358
      DO 401 J1 = 1,3                                 CH1 359
      INDX2 = INDX1 + (J1-2)*NPX + I1-2              CH1 360
      SUM1 = SUM1 + P(I1,2)*PRR(J1,2,J)*YD(INDX2) CH1 361
401 SUM2 = SUM2 + P(I1,2)*PRR(J1,2,J)*Y(INDX2) CH1 362
      SUMD = SUMD + SUM1*B(IGP,IG,IM)                CH1 363
402 SUM = SUM + SUM2*B(IG,IGP,IM)                   CH1 364
      YDS(INDX) = SUMD*AMB                           CH1 365
      YS(INDX) = SUM*AMB                             CH1 366
      SIZE = SIZE + SUM*DX(I)*AMB                     CH1 367
      SIZED = SIZED + SUMD*DX(I)*AMB                 CH1 368
      IF(NX-I)406,406,403                             CH1 369
403 INDX = INDX + 1                                   CH1 370
      SUM = C.C                                       CH1 371
      SUMD = 0.C                                       CH1 372
      DO 405 IS = 1,2                                 CH1 373
      I2=5-2*IS                                       CH1 374
      I1=I+IS-1                                       CH1 375
      IM=M(I1,J)                                       CH1 376
      DO 405 IGP = 1,NG                               CH1 377
      INDX1 = INDX + (IGP-IG)*NP + 2*IS-3           CH1 378
      SUM1 = 0.C                                       CH1 379
      SUM2 = 0.C                                       CH1 380

```

```

DO 404 I1 = 1,3
DO 404 J1 = 1,3
INDX2 = INDX1 + (J1-2)*NPX + I1-2
SUM1 = SUM1 + YD(INDX2)*PRR(J1,2,J)*P(I1,I2)
404 SUM2 = SUM2 + Y(INDX2)*PRR(J1,2,J)*P(I1,I2)
SUMD = SUMD + SUM1*B(IG,IG,IM)*DX(II)
405 SUM = SUM + SUM2*B(IG,IG,IM)*DX(II)
YDS(INDX) = SUMD*AMB
YS(INDX) = SUM*AMB
SIZE = SIZE + SUM*AMB
SIZED = SIZED + SUMD*AMB
406 IF(NY-J)414,414,407
407 INDX = (IG-1)*NP + 2*J*NPX + 2*I
SUM = 0.0
SUMD = 0.0
DO 409 JS = 1,2
J2=5-2*JS
JJ=J+JS-1
IM=M(I,JJ)
DO 409 IGP = 1,NG
INDX1 = INDX + (IG-IG)*NP + (2*JS-3)*NPX
SUM1 = 0.0
SUM2 = 0.0
DO 408 I1 = 1,3
DO 408 J1 = 1,3
INDX2 = INDX1 + (J1-2)*NPX + I1-2
SUM1 = SUM1 + YD(INDX2)*PRR(J1,J2,JJ)*P(I1,2)
408 SUM2 = SUM2 + Y(INDX2)*PRR(J1,J2,JJ)*P(I1,2)
SUMD = SUMD + SUM1*B(IG,IG,IM)
409 SUM = SUM + SUM2*B(IG,IG,IM)
YDS(INDX) = SUMD*AMB
YS(INDX) = SUM*AMB
SIZE = SIZE + SUM*AMB*DX(I)
SIZED = SIZED + SUMD*AMB*DX(I)
IF(NX-I)414,414,410
410 INDX = INDX + 1
SUM = 0.0
SUMD = 0.0
DO 413 IS = 1,2
I2=5-2*IS
II=I+IS-1
DO 413 JS = 1,2
J2=5-2*JS
JJ= J+JS-1
IM = M(II,JJ)
DO 412 IGP = 1,NG
INDX1 = INDX + (IG-IG)*NP + (2*JS-3)*NPX + 2*IS-3
SUM1 = 0.0

```

```

CH1 381
CH1 382
CH1 383
CH1 384
CH1 385
CH1 386
CH1 387
CH1 388
CH1 389
CH1 390
CH1 391
CH1 392
CH1 393
CH1 394
CH1 395
CH1 396
CH1 397
CH1 398
CH1 399
CH1 400
CH1 401
CH1 402
CH1 403
CH1 404
CH1 405
CH1 406
CH1 407
CH1 408
CH1 409
CH1 410
CH1 411
CH1 412
CH1 413
CH1 414
CH1 415
CH1 416
CH1 417
CH1 418
CH1 419
CH1 420
CH1 421
CH1 422
CH1 423
CH1 424
CH1 425
CH1 426
CH1 427
CH1 428

```

SUM2 = 0.0	CH1 429
DO 411 I1 = 1,3	CH1 430
DO 411 J1 = 1,3	CH1 431
INDX2 = INDX1 + (J1-2)*NPX + I1-2	CH1 432
SUM1 = SUM1 + YD(INDX2)*PRR(J1,J2,JJ)*P(I1,I2)	CH1 433
411 SUM2 = SUM2 + Y(INDX2)*PRR(J1,J2,JJ)*P(I1,I2)	CH1 434
SUMD = SLMD + SUM1*B(IG,IG,IM)*DX(I1)	CH1 435
412 SUM = SUM + SUM2*B(IG,IG,IM)*DX(I1)	CH1 436
413 CONTINUE	CH1 437
YDS(INDX) = SLMD*AMB	CH1 438
YS(INDX) = SUM*AMB	CH1 439
SIZE = SIZE + SUM*AMB	CH1 440
SIZED = SIZED + SLMD*AMB	CH1 441
414 CONTINUE	CH1 442
415 CONTINUE	CH1 443
C INNER ITERATION (THROUGH 522)	CH1 444
K TIAN = C	CH1 445
500 SUM = 0.0	CH1 446
SUMS = 0.0	CH1 447
DO 1419 I = 1,NX	CH1 448
S = DX(I)	CH1 449
IF(I-1)1401,1401,1402	CH1 450
1401 I21 = 2	CH1 451
GO TO 1403	CH1 452
1402 I21 = 1	CH1 453
1403 IF(I-NX)1405,1404,1404	CH1 454
1404 I22 = 2	CH1 455
GO TO 1406	CH1 456
1405 I22 = 3	CH1 457
1406 DO 1419 J=1,NY	CH1 458
IM = M(I,J)	CH1 459
IF(J-1)1407,1407,1408	CH1 460
1407 J21 = 2	CH1 461
GO TO 1409	CH1 462
1408 J21 = 1	CH1 463
1409 IF(J-NY)1411,1410,1410	CH1 464
1410 J22 = 2	CH1 465
GO TO 1412	CH1 466
1411 J22 = 3	CH1 467
1412 DO 1415 I1=1,3	CH1 468
SUM1 = 0.0	CH1 469
SUM2 = 0.0	CH1 470
DO 1413 I2 = I21,I22	CH1 471
SUM1=SUM1+P(I1,I2)	CH1 472
1413 SUM2=SUM2+Q(I1,I2)	CH1 473
W2(I1) = SUM1*S	CH1 474
W2(I1+3) = SUM2/S	CH1 475
SUM1 = 0.0	CH1 476

SUM2 = 0.0	CH1 477
DO 1414 J2= J21,J22	CH1 478
SUM1 = SUM1+ PRR(I1,J2,J)	CH1 479
1414 SUM2 = SUM2+ QRR(I1,J2,J)	CH1 480
W2(I1+6) = SUM1	CH1 481
1415 W2(I1+9) = SUM2	CH1 482
DO 1418 IG =1,NG	CH1 483
INDX = (IG-1)*NP + (2*J-1)*NPX + 2*I	CH1 484
SUMD = 0.0	CH1 485
XUMD = 0.0	CH1 486
DO 1416 IGP = 1,NG	CH1 487
XUMD = XUMD + A1(IG,IGP,IM)	CH1 488
1416 SUMD = SUMD + A1(IGP,IG,IM)	CH1 489
SUM1=0.0	CH1 490
SUM2 = 0.0	CH1 491
SUM3 = 0.0	CH1 492
SUM4 = 0.0	CH1 493
DO 1417 I1=1,3	CH1 494
DO 1417 J1=1,3	CH1 495
INDX1 = INDX + I1-2 +(J1-2)*NPX	CH1 496
SUM3 = SUM3 + YD(INDX1)*W2(I1)*W2(J1+6)	CH1 497
SUM4 = SUM4 + YD(INDX1)*(W2(I1)*W2(J1+9)+W2(I1+3)*W2(J1+6))	CH1 498
SUM1 = SUM1 + Y(INDX1)*W2(I1)*W2(J1+6)	CH1 499
1417 SUM2 = SUM2 + Y(INDX1)*(W2(I1)*W2(J1+9) + W2(I1+3)*W2(J1+6))	CH1 500
SUMS = SUMS + SUM3*XUMD + D1(IG,IM)*SUM4	CH1 501
1418 SUM = SUM + SUM1*SUMD + D1(IG,IM)*SUM2	CH1 502
1419 CONTINUE	CH1 503
SUM1 = SIZE/SUM	CH1 504
SUM2 = SIZED/SUMS	CH1 505
J = NP*NG	CH1 506
DO 1420 I =1,J	CH1 507
Y(I) = Y(I)*SUM1	CH1 508
1420 YD(I) = YD(I)*SUM2	CH1 509
DO 522 IG=1,NG	CH1 510
DO 519 I=1,NX	CH1 511
DO 517 J=1,NY	CH1 512
INDX = (IG-1)*NP+(2*J-1)*NPX+2*I	CH1 513
IM=M(I,J)	CH1 514
SUMD=0.0	CH1 515
SUM =0.0	CH1 516
DO 502 I1=1,3	CH1 517
DO 502 J1=1,3	CH1 518
INDX1=INDX+(J1-2)*NPX+I1-2	CH1 519
SUM1=D1(IG,IM)*(Q(I1,2)*PRR(J1,2,J)/DX(I)**2+P(I1,2)*QRR(J1,2,J))	CH1 520
SUMD=SUMD+SUM1*YD(INDX1)	CH1 521
SUM=SUM+SUM1*Y(INDX1)	CH1 522
SUM1=0	CH1 523
?	CH1 524

```

DO 501 I=1,NG
INDX2=INDX1+(IGP-IG)*NP
SUM1=SUM1+A1(IG,IG,IM)*YD(INDX2)
501 SUM2=SUM2+A1(IG,IG,IM)*Y(INDX2)
SUMD=SUMD+SUM1*P(I1,2)*PRR(J1,2,J)
502 SUM=SUM+SUM2*P(I1,2)*PRR(J1,2,J)
YD(INDX)=YD(INDX)+(YDS(INDX)-SUMD)/DEN(INDX)
Y(INDX)=Y(INDX)+(Y S(INDX)-SUM)/DEN(INDX)
IF(NX-I)507,507,503
503 INDX=INDX+1
SUMD=C.0
SUM=C.0
DO 506 IS=1,2
II=I+IS-1
IM=M(II,J)
I2=5-2*IS
DO 505 I1=1,3
DO 505 J1=1,3
INDX1=INDX+(J1-2)*NPX+I1-I2
SUM1=D1(IG,IM)*(Q(I1,J2)*PRR(J1,2,J)/DX(II)+P(I1,I2)*QRR(J1,2,J)*
1 DX(II))
SUMD=SUMD+SUM1*YD(INDX1)
SUM=SUM+SUM1*Y(INDX1)
SUM1=C.0
SUM2=C.0
DO 504 IGP=1,NG
INDX2=INDX1+(IGP-IG)*NP
SUM1=SUM1+A1(IG,IG,IM)*YD(INDX2)
504 SUM2=SUM2+A1(IG,IG,IM)*Y(INDX2)
SUMD=SUMD+SUM1*P(I1,I2)*PRR(J1,2,J)*DX(II)
505 SUM=SUM+SUM2*P(I1,I2)*PRR(J1,2,J)*DX(II)
506 CONTINUE
YD(INDX)=YD(INDX)+(YDS(INDX)-SUMD)/DEN(INDX)
Y(INDX)=Y(INDX)+(Y S(INDX)-SUM)/DEN(INDX)
507 IF(NY-J) 516,516,508
508 INDX=(IG-1)*NP+2*J*NPX+2*I
SUMD=C.0
SUM=C.0
DO 511 JS=1,2
JJ=J+JS-1
IM=M(I,JJ)
J2=5-2*JS
DO 510 I1=1,3
DO 510 J1=1,3
INDX1=INDX+(J1-J2)*NPX+I1-2
SUM1=D1(IG,IM)*(Q(I1,2)*PRR(J1,J2,JJ)/DX(I)**2+P(I1,2)*QRR(J1,J2,
1 JJ))
SUMD=SUMD+SUM1*YD(INDX1)

```

CH1 525
CH1 526
CH1 527
CH1 528
CH1 529
CH1 530
CH1 531
CH1 532
CH1 533
CH1 534
CH1 535
CH1 536
CH1 537
CH1 538
CH1 539
CH1 540
CH1 541
CH1 542
CH1 543
CH1 544
CH1 545
CH1 546
CH1 547
CH1 548
CH1 549
CH1 550
CH1 551
CH1 552
CH1 553
CH1 554
CH1 555
CH1 556
CH1 557
CH1 558
CH1 559
CH1 560
CH1 561
CH1 562
CH1 563
CH1 564
CH1 565
CH1 566
CH1 567
CH1 568
CH1 569
CH1 570
CH1 571
CH1 572

SUM =SUM +SUM1*Y (INDX1)	CH1 573
SUM1=C.0	CH1 574
SUM2=C.0	CH1 575
DO 509 IGP=1,NG	CH1 576
INDX2=INDX1+(IGP-IG)*NP	CH1 577
SUM1=SUM1+A1(IGP,IG,IM)*YD(INDX2)	CH1 578
509 SUM2=SUM2+A1(IG,IGP,IM)*Y (INDX2)	CH1 579
SUMD=SUMD+SUM1*P(I1,2)*PRR(J1,J2,JJ)	CH1 580
510 SUM =SUM +SUM2*P(I1,2)*PRR(J1,J2,JJ)	CH1 581
511 CONTINUE	CH1 582
YD(INDX)=YD(INDX)+(YDS(INDX)-SUMD)/DEN(INDX)	CH1 583
Y (INDX)=Y (INDX)+(Y S(INDX)-SUM)/DEN(INDX)	CH1 584
IF(NX-1) 516,516,512	CH1 585
512 INDX=INDX+1	CH1 586
SUMD=C.0	CH1 587
SUM =C.0	CH1 588
DO 515 IS=1,2	CH1 589
II=I+IS-1	CH1 590
I2=5-2*IS	CH1 591
DO 515 JS=1,2	CH1 592
JJ=J+JS-1	CH1 593
J2=5-2*JS	CH1 594
IM=M(II,JJ)	CH1 595
DO 514 I1=1,3	CH1 596
DO 514 J1=1,3	CH1 597
INDX1=INDX+(J1-J2)*NPX+(I1-I2)	CH1 598
SUM1=D1(IG,IM)*(Q(I1,I2)*PRR(J1,J2,JJ)/DX(II)+P(I1,I2)*QRR(J1,J2,	CH1 599
1 JJ)*DX(II))	CH1 600
SUMD=SUMD+SUM1*YD(INDX1)	CH1 601
SUM =SUM +SUM1*Y (INDX1)	CH1 602
SUM1=C.0	CH1 603
SUM2=C.0	CH1 604
DO 513 IGP=1,NG	CH1 605
INDX2=INDX1+(IGP-IG)*NP	CH1 606
SUM1=SUM1+A1(IGP,IG,IM)*YD(INDX2)	CH1 607
513 SUM2=SUM2+A1(IG,IGP,IM)*Y (INDX2)	CH1 608
SUMD=SUMD+SUM1*P(I1,I2)*PRR(J1,J2,JJ)*DX(II)	CH1 609
514 SUM =SUM +SUM2*P(I1,I2)*PRR(J1,J2,JJ)*DX(II)	CH1 610
515 CONTINUE	CH1 611
YD(INDX)=YD(INDX)+(YDS(INDX)-SUMD)/DEN(INDX)	CH1 612
Y (INDX)=Y (INDX)+(Y S(INDX)-SUM)/DEN(INDX)	CH1 613
516 CONTINUE	CH1 614
517 CONTINUE	CH1 615
IF(NGEOM**2)519,519,518	CH1 616
518 INDX=(IG-1)*NP+2*I	CH1 617
INDX1=INDX+NPX	CH1 618
INDX2=INDX1+NPX	CH1 619
YD(INDX)=(4.0*YD(INDX1)-YD(INDX2))/3.0	CH1 620

```
YD(INDX-1)=(4.C*YD(INDX1-1)-YD(INDX2-1))/3.0
Y(INDX)=(4.0*Y(INDX1)-Y(INDX2))/3.0
Y(INDX-1)=(4.C*Y(INDX1-1)-Y(INDX2-1))/3.C
519 CONTINUE
IF(NGEOM**2-1)522,522,520
520 INDX=(IG-1)*NP+1-NPX
DO 521 J=1,NY2
INDX=INDX+NPX
YD(INDX)=(4.C*YD(INDX+1)-YD(INDX+2))/3.0
521 Y(INDX)=(4.0*Y(INDX+1)-Y(INDX+2))/3.0
522 CONTINUE
KTINN=KTINN+1
IF(KTINN-ITINN)500,202,202
C
C NORMALIZATION
202 YMAX= Y(1)
SUM = YMAX**2
DO 204 I = 2,NN
CHGE = Y(I)**2
IF(CHGE-SUM)204,204,203
203 YMAX = Y(I)
SUM = YMAX**2
204 CONTINUE
YDMAX = YD(1)
SUM = YDMAX**2
DO 206 I = 2,NN
CHGE = YD(I)**2
IF(CHGE-SUM)206,206,205
205 YDMAX = YD(I)
SUM = YDMAX**2
206 CONTINUE
YMAX = YMAX*1.CE-6
YDMAX = YDMAX*1.CE-6
DO 207 I = 1,NN
Y(I) = Y(I)/YMAX
207 YD(I) = YD(I)/YDMAX
GO TO 300
600 WRITE OUTPUT TAPE 6,601,AMB
601 FORMAT(1HC // 13HCEIGENVALUE = 1PE14.7 )
DO 646 IG = 1,NG
WRITE OUTPUT TAPE 6,120
WRITE OUTPUT TAPE 6,119,IG
DO 646 IPY = 1,NPY
I1=(IG-1)*NP+(IPY-1)*NPX+1
I2=I1+NPX-1
646 WRITE OUTPUT TAPE 6,113,(Y(I),I=I1,I2)
DO 647 IG = 1,NG
WRITE OUTPUT TAPE 6,120
```

```
CH1 621
CH1 622
CH1 623
CH1 624
CH1 625
CH1 626
CH1 627
CH1 628
CH1 629
CH1 630
CH1 631
CH1 632
CH1 633
CH1 634
CH1 635
CH1 636
CH1 637
CH1 638
CH1 639
CH1 640
CH1 641
CH1 642
CH1 643
CH1 644
CH1 645
CH1 646
CH1 647
CH1 648
CH1 649
CH1 650
CH1 651
CH1 652
CH1 653
CH1 654
CH1 655
CH1 656
CH1 657
CH1 658
CH1 659
CH1 660
CH1 661
CH1 662
CH1 663
CH1 664
CH1 665
CH1 666
CH1 667
CH1 668
```

```

WRITE OUTPUT TAPE 6,602,IG
602 FORMAT(5X17HADJOINT GROUP NO.13 )
DO 647 IPY = 1,NPY
  I1=(IG-1)*NP+(IPY-1)*NPX+1
  I2=I1+NPX-1
647 WRITE OUTPUT TAPE 6,113,(YD(I),I=I1,I2)
650 KTOUT = KTOUT +1
  IF(ISSW)811,811,812
812 PRINT 809,AMB
809 FORMAT(13H0EIGENVALUE = 1PE14.7 )
  IF(SENSE SWITCH 1)81C,811
810 KTOUT = 10000
811 IF(KTOUT-ITOUT)400,700,700
700 DO 7C1 IM=1,NM
  DO 7C1 IG=1,NG
701 XNU1(IG,IM)=AMB*XNU1(IG,IM)
  DO 7C3 I=1,NX
  DO 7C3 J=1,NY
    IM=M(I,J)
    INDX=2*I+(2*J-1)*NPX-NP
    INDX1=9*NX*(J-1)+NPXJ+3*I-2-NPJ+NNN
    DO 7C3 I1=1,3
    DO 7C3 J1=1,3
      INDX2=INDX+I1-2+(J1-2)*NPX
      INDX3=INDX1+I1-2+(J1-2)*NPXJ
      SUM=C.0
      DO 7C2 IG=1,NG
        INDX4=INDX2+IG*NP
702 SUM=SUM+XNU1(IG,IM)*SFI1(IG,IM)*Y(INDX4)
        DO 7C3 IJ=1,NJ
          INDX4=INDX3+IJ*NPJ
703 Y(INDX4)=SUM*BETA1(IJ,IM)/XLAM1(IJ,IM)
          IF(LAMB)706,704,706
704 DO 7C5 IM=1,NM
      DO 7C5 IG=1,NG
705 XNU1(IG,IM)=XNU1(IG,IM)/AMB
706 CONTINUE
707 DO 800 IG=1,NG
  K1=(IG-1)*NP +1
  K2= IG*NP
800 WRITE OUTPUT TAPE 7,801,(Y(I),I=K1,K2)
801 FORMAT(5E15.8)
  DO 802 IJ=1,NJ
    K1=NG*NP +(IJ-1)*NPJ+1
    K2=K1+NPJ-1
802 WRITE OUTPUT TAPE 7,801,(Y(I),I=K1,K2)
TIME=CLOCKF(CARD,CHGE)
WRITE OUTPUT TAPE 6,803,TIME

```

```

CH1 669
CH1 670
CH1 671
CH1 672
CH1 673
CH1 674
CH1 675
CH1 676
CH1 677
CH1 678
CH1 679
CH1 680
CH1 681
CH1 682
CH1 683
CH1 684
CH1 685
CH1 686
CH1 687
CH1 688
CH1 689
CH1 690
CH1 691
CH1 692
CH1 693
CH1 694
CH1 695
CH1 696
CH1 697
CH1 698
CH1 699
CH1 700
CH1 701
CH1 702
CH1 703
CH1 704
CH1 705
CH1 706
CH1 707
CH1 708
CH1 709
CH1 710
CH1 711
CH1 712
CH1 713
CH1 714
CH1 715
CH1 716

```


803 FORM // 61X16+COMPUTER TIME = -3PF6.3)

26 DO 65 IG=1,NG
DO 65 IM=1,NM
DO 66 IGP=1,NG
66 A1(IG,IGP,IM)=-A1(IG,IGP,IM)+(1.0-BTOT1(IM))*CHI1(IG,IM)*XNU1(IGP,
1IM)*SFI1(IGP,IM)
DO 65 IJ=1,NJ
AFD1(IG,IJ,IM)= S1(IG,IJ,IM)*XLAM1(IJ,IM)
65 ADF1(IJ,IG,IM)= BETA1(IJ,IM)*XNU1(IG,IM)*SFI1(IG,IM)
DO 69 I=1,N
69 YD(I) = C.0
DO 67 I=1,2800
67 W2(I) = W1(I)
CALL CHAIN(2,4)
END
SUBROUTINE PRINT

CH1 717
CH1 718
CH1 719
CH1 720
CH1 721
CH1 722
CH1 723
CH1 724
CH1 725
CH1 726
CH1 727
CH1 728
CH1 729
CH1 730
CH1 731

-69-

C

DIMENSION P(3,3),Q(3,3),PR(3,3),QR(3,3),M(7,7),DX(7),DY(7),PRR(3,3)
1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20),
2SFI1(4,20),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20),
3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4
4,4,20),SABS2(4,20),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,
520),BETA2(6,20),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20),
6A3(4,4,20),AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20)
7),VD3(4,20),W1(2800),W2(2800),Y(2620),YD(2620),YDS(676),X5N(6780
8),BTCT1(20),B(4,4,20),DEN(676),YS(1690)

PNT 1
PNT 2
PNT 3
PNT 4
PNT 5
PNT 6
PNT 7
PNT 8
PNT 9
PNT 10
PNT 11
PNT 12

C

EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)
1,(X5N(2121),BTCT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781),
2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510
31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550
41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1),
5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2),
6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128
71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)
8,ADF2)

PNT 13
PNT 14
PNT 15
PNT 16
PNT 17
PNT 18
PNT 19
PNT 20
PNT 21
PNT 22

C

COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YD,YDS,M,DX,DY,P,Q,PR
1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,
2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW

PNT 23
PNT 24
PNT 25
PNT 26

C

COMMON DEN,NGECM,NGIN,NGIN1,NGCUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,
1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUMC,SUMS,KIT,KKIT,KRET
2,E1,E2,E3,E4,ITINN,ITCUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW

PNT 27
PNT 28
PNT 29
PNT 30

C

DO 40 IM=1,NM
WRITE OUTPUT TAPE 6,9C2,IM
902 FORMAT(16HC MATERIAL NO. I2 ///)

PNT 31
PNT 32
PNT 33

```

WRITE OUTPUT TAPE 6,903,NJ,(BETA1(IJ,IM),IJ=1,NJ)          PNT 34
903 FORMAT( 14X 53HBETA THE FRACTION FOR EACH PRECURSOR GRUOP  IJ = 1PNT 35
1 , I2  // ( 21X1PE14.7,1P3E20.7 ) )                      PNT 36
WRITE OUTPUT TAPE 6,904,NJ,(XLAM1(IJ,IM),IJ=1,NJ)          PNT 37
904 FORMAT( /// 14X 61HLAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR PNT 38
1GROUP  IJ = 1 , I2  // ( 21X1PE14.7,1P3E20.7 ) )          PNT 39
DO 40  IG =1,NG                                              PNT 40
WRITE OUTPUT TAPE 6,906,IG,SABS1(IG,IM),SFI1(IG,IM),XNU1(IG,IM), PNT 41
1D1(IG,IM),CHI1(IG,IM),V1(IG,IM),IG,NG,(STR1(IG,IGP,IM),IGP=1,NG) PNT 42
906 FORMAT( 16HO GROUP NO. I2 // 9X 7HSIGA = 1PE14.7,          PNT 43
1 10X 7HSIGF = 1PE14.7 , 10X 6HNU = 1PE14.7 //              PNT 44
2                                     9X 7HDIFC = 1PE14.7,      PNT 45
3 10X 7HCHI = 1PE14.7 , 10X 6HVEL = 1PE14.7 //              PNT 46
4 14X15HTRANSFERS INTO I2, 2X17HFROM 1, 2, ... , I2,        PNT 47
5 4H -- //(21X1PE14.7,1P3E20.7) )                          PNT 48
WRITE OUTPUT TAPE 6,907,IG,NJ,( S1(IG,IJ,IM),IJ=1,NJ )      PNT 49
907 FORMAT( // 14X25HFISSION SPECTRUM IN GROUP I2, 36H FOR EACH PRECURPNT 50
1SOR GROUP  IJ = 1 , I2 //(21X1PE14.7,1P3E20.7) )          PNT 51
40 CONTINUE                                                  PNT 52
RETURN                                                        PNT 53
END                                                            PNT 54
*
C CHAIN (2,4)
C TIME DEPENDENT CHAIN ( ICE SUBROUTINE )
C
C DIMENSION P(3,3),C(3,3),PR(3,3),QR(3,3),M(7,7),DX(7),DY(7),PRR(3,3CH2 1
1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20), CH2 2
2SFI1(4,20),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20), CH2 3
3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4CH2 4
4,4,20),SABS2(4,20),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,CH2 5
520),BETA2(6,20),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20), CH2 6
6A3(4,4,20), AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20CH2 7
7),VD3(4,20),W1(2800),W2(2800),Y(2620),YD(2620),YDS( 676),X5N( 6780CH2 8
8),BTOT1(20),B(4,4,20),DEN(676),YS(1690) CH2 9
C
C EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)CH2 10
1,(X5N(2121),BTOT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781), CH2 11
2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510CH2 12
31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550CH2 13
41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1), CH2 14
5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2), CH2 15
6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128CH2 16
71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)CH2 17
8,ADF2) CH2 18
C
C COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YD,YDS,M,CX,DY,P,Q,PRCH2 19
1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,CH2 20
2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW CH2 21
COM DEN,NGEOM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,CH2 22

```

111, JJ, IM, INDX, INDX1, INDX2, L1, L2, S, SUM, SUM1, SUMD, SUMS, KIT, KKIT, KRETCH2	27
2, E1, E2, E3, E4, ITINN, ITOUT, ITPRNT, LAMB, AMB, NNN, LINK, IL, NCOL, NROW	CH2 28
	CH2 29
CALL FDSTRT	CH2 30
READ INPLT TAPE 5, 107, T	CH2 31
107 FORMAT(8E10.4)	CH2 32
27 READ INPLT TAPE 5, 114, DT, PP, E1, E2, E3, E4, NSTCP, NCHGE, NORM, NGOUT,	CH2 33
1NJOUT	CH2 34
NGIN = NG-NGOUT	CH2 35
NJIN = NJ-NJOUT	CH2 36
NGIN1 = NGOUT + 1	CH2 37
114 FORMAT(6E10.4, 5I4)	CH2 38
62 DO 29 I=1, 2800	CH2 39
29 W1(I) = W2(I)	CH2 40
WRITE OUTPUT TAPE 6, 115, T	CH2 41
115 FORMAT(4H1T = 1PE14.7, 1X20H, NUCLEAR CONSTANTS)	CH2 42
CALL PRINT	CH2 43
WRITE OUTPUT TAPE 6, 134, E1, E2, E3, E4	CH2 44
134 FORMAT(3CHOERRCR LIMITS FROM NCW ON ARE 1P4E17.7)	CH2 45
IF(NCHGE) 36, 36, 30	CH2 46
30 CHGE = NCHGE	CH2 47
CARD = CHGE/8.0	CH2 48
NCARD = CARD + 0.01	CH2 49
CARD1 = NCARD	CH2 50
NREM = 8.0*(CARD-CARD1)+0.01	CH2 51
IF(NCARD) 33, 33, 31	CH2 52
31 DO 32 I=1, NCARD	CH2 53
READ INPLT TAPE 5, 111, (NIN(K), K=1, 8)	CH2 54
READ INPLT TAPE 5, 107, (WIN(K), K=1, 8)	CH2 55
DO 32 K=1, 8	CH2 56
K1= NIN(K)	CH2 57
32 W2(K1)= WIN(K)	CH2 58
33 IF (NREM) 195, 195, 34	CH2 59
34 READ INPLT TAPE 5, 111, (NIN(K), K=1, NREM)	CH2 60
READ INPLT TAPE 5, 107, (WIN(K), K=1, NREM)	CH2 61
DO 35 K=1, NREM	CH2 62
K1= NIN(K)	CH2 63
35 W2(K1) = WIN(K)	CH2 64
111 FORMAT(8I10)	CH2 65
195 DO 84 IM = 1, NM	CH2 66
BTOT2 = 0.0	CH2 67
DO 75 IJ=1, NJ	CH2 68
75 BTOT2 = BTOT2 + BETA2(IJ, IM)	CH2 69
DO 83 IG=1, NG	CH2 70
DO 76 IGP=1, NG	CH2 71
76 A2(IG, IGP, IM)= STR2(IG, IGP, IM)+(1.0-BTOT2)*CH12(IG, IM)*XNU2(IGP, IM	CH2 72
1)*SF12(IGP, IM)	CH2 73
A2(IG, IG, IM)= A2(IG, IG, IM)-SABS2(IG, IM)	CH2 74

DO 77 IGP=1,NG	CH2 75
77 A2(IG,IG,IM)= A2(IG,IG,IM)-STR2(IGP,IG,IM)	CH2 76
DO 74 IJ=1,NJ	CH2 77
AFD2(IG,IJ,IM)= S2(IG,IJ,IM)*XLAM2(IJ,IM)	CH2 78
74 ADF2(IJ,IG,IM)= BETA2(IJ,IM)*XNU2(IG,IM)*SFI2(IG,IM)	CH2 79
83 CONTINUE	CH2 80
84 CONTINUE	CH2 81
36 IF(DT)198,198,37	CH2 82
37 TF=T+DT	CH2 83
TI = -10.0	CH2 84
WRITE OUTPUT TAPE 6,806	CH2 85
806 FORMAT(52HOENTRY TO SUBROUTINE FEEDBK (INITIALLY IN THIS DT).)	CH2 86
CALL FEEDBK	CH2 87
TI = T	CH2 88
TP = T	CH2 89
DO 68 IG=1,NG	CH2 90
DO 68 IM=1,NM	CH2 91
68 VD3(IG,IM)= (V2(IG,IM)-V1(IG,IM))/DT	CH2 92
NN = NGIN*NP +NJIN*NPJ	CH2 93
NN1 = NGOUT*NP+1	CH2 94
C	CH2 95
C ICE SUBROUTINE	CH2 96
82 CALL ICE (PP,T,TP,E1,E2,NN,Y(NN1),YD(NN1),X5N,KAT)	CH2 97
38 GO TO (1000,2000,3000,4000),KAT	CH2 98
39 KAT = XICEF (0)	CH2 99
GO TO 38	CH2 100
C	CH2 101
C ROUTINE TO NORMALIZE FLUXES AND PRECURSORS	CH2 102
198 IF(NCRM-1)58,41,58	CH2 103
41 YMAX=Y(1)	CH2 104
DO 42 IP=1,NP	CH2 105
DO 42 IG=1,NG	CH2 106
I=(IG-1)*NP+IP	CH2 107
IF(Y(I)-YMAX)42,42,43	CH2 108
43 YMAX=Y(I)	CH2 109
42 CONTINUE	CH2 110
YMAX = YMAX*1.0E-6	CH2 111
DO 44 I =1,N	CH2 112
Y(I)=Y(I)/YMAX	CH2 113
44 YD(I)=YD(I)/YMAX	CH2 114
58 JA=1	CH2 115
GO TO 45	CH2 116
49 IF(NSTOP-2)27,99,99	CH2 117
C	CH2 118
C BOX B (SUPPLEMENTARY COMPUTATIONS AND END OF CASE CHECK)	CH2 119
2000 TI=T-TI	CH2 120
IF(NJOUT)2003,2003,2004	CH2 121
2004 L=2	CH2 122

L1=NJ +1	CH2 123
L2=NJ	CH2 124
GO TC 1100	CH2 125
2001 L=NN +NN1	CH2 126
DO 2002 I= L,N	CH2 127
2002 Y(I)=Y(I)+TI*YD(I)	CH2 128
2003 WRITE OUTPUT TAPE 6,805	CH2 129
805 FORMAT(4CHOENTRY TO SUBROUTINE FEEDBK FROM BCX B.)	CH2 130
CALL FEEDBK	CH2 131
IF(SENSE SWITCH 2)810,811	CH2 132
810 NSTOP= 2	CH2 133
GO TC 198	CH2 134
811 TI = T	CH2 135
IF(T-TF+5.0E-C6*PP)39,198,198	CH2 136
C	CH2 137
C BOX C (PRINT ROUTINE)	CH2 138
3000 IF(NGOUT)60,6C,3001	CH2 139
3001 WRITE OUTPUT TAPE 6,807	CH2 140
807 FORMAT(4CHOENTRY TO SUBROUTINE FLXCUT FROM BCX C.)	CH2 141
CALL FLXCUT	CH2 142
60 JA=2	CH2 143
45 WRITE OUTPUT TAPE 6,118,T	CH2 144
118 FORMAT(1F /// 4HCT =1PE14.7 ///)	CH2 145
TC1 = 1.0	CH2 146
IF(DT)4996,4996,4995	CH2 147
4995 TC1=(TF-T)/DT	CH2 148
4996 TC2=1.0-TC1	CH2 149
SUM=C.0	CH2 150
SUMD=C.0	CH2 151
DO 122 I=1,NX	CH2 152
DO 122 J=1,NY	CH2 153
IM=M(I,J)	CH2 154
DO 122 I1=1,3	CH2 155
DO 122 J1=1,3	CH2 156
SUM1=C.0	CH2 157
INDX=(3*J+J1-4)*NPXJ+3*I+I1-3	CH2 158
DO 129 IG=1,NG	CH2 159
INDX1=(IG-1)*NP+(2*J+J1-3)*NPX+2*I+I1-2	CH2 160
129 SUM1=SUM1+(TC1*SF11(IG,IM)+TC2*SF12(IG,IM))*Y(INDX1)	CH2 161
YDS(INDX)=SUM1	CH2 162
IF(NGEOM)4997,4997,4998	CH2 163
4997 S = C.0	CH2 164
GO TC 4999	CH2 165
4998 S = J1-2	CH2 166
4999 IF(J-1) 8101,8101,8100	CH2 167
8100 G = C.0	CH2 168
GO TC 8102	CH2 169
8101 G = 1.0	CH2 170

```

8102 SR = P(I1,I1)*DX(I)*(PRR(J1,J1,J) + 0.5*S*DY(J)**2 + G*PR(J1) ) CH2 171
      SUM=SUM+SUM1*SR CH2 172
122 SUMD=SUMD+SR CH2 173
      S=SUM/SUMD CH2 174
      WRITE OUTPUT TAPE 6,124,S CH2 175
124 FORMAT(5X70HPOWER DENSITY,FLUX TIMES SIGMA FISSION (AVERAGE) CH2 176
1 POWER DENSITY = 1PE14.7,1H) ) CH2 177
      IG = 0 CH2 178
      KRET = 1 CH2 179
      SUM = INTPY + 1 CH2 180
      SUM1 = INTPX + 1 CH2 181
5000 DO 5003 IY = 1,NY CH2 182
      INDX = 3*NPXJ*(IY-1) CH2 183
      DO 5003 I1 = 1,NROW CH2 184
      SUM2 = I1-INTPY-2 CH2 185
      SUM2 = SUM2/SUM CH2 186
      DO 5002 IX = 1,NX CH2 187
      INDX4 = NCOL*(IX-1) CH2 188
      I = INDX4 - INTPX CH2 189
      DO 5001 I2 = 1,3 CH2 190
      INDX1 = INDX + 3*(IX-1) + I2 CH2 191
      INDX2 = INDX1 + NPXJ CH2 192
      INDX3 = INDX2 + NPXJ CH2 193
      J = I + I2*(INTPX+1) CH2 194
5001 A3(J) = SUM2*(SUM2-1.)*YDS(INDX1)/2. + (1.-SUM2**2)*YDS(INDX2) CH2 195
      1 + SUM2*(SUM2+1.)*YDS(INDX3)/2. CH2 196
      INDX1 = INDX4 + 1 CH2 197
      INDX2 = INDX1 + INTPX + 1 CH2 198
      INDX3 = INDX2 + INTPX + 1 CH2 199
      DO 5002 I2 = 2,NCOL CH2 200
      J = INDX4 + I2 CH2 201
      SUM3 = I2 -INTPX -2 CH2 202
      SUM3 = SUM3/SUM1 CH2 203
5002 A3(J) = SUM3*(SUM3-1.)*A3(INDX1)/2. + (1.0-SUM3**2)*A3(INDX2) CH2 204
      1 + SUM3*(SUM3+1.)*A3(INDX3)/2. CH2 205
5003 WRITE OUTPUT TAPE 6,113,(A3(I2),I2=1,IL) CH2 206
      GO TO (5004,5008),KRET CH2 207
5004 IG = IG + 1 CH2 208
      IF(IG-NG)5005,5005,5007 CH2 209
5005 WRITE OUTPUT TAPE 6,120 CH2 210
      120 FORMAT(1H ) CH2 211
      WRITE OUTPUT TAPE 6,119,IG CH2 212
      119 FORMAT(5X14HFLUX GROUP NO.I3 ) CH2 213
      113 FORMAT(1P5E20.7) CH2 214
      DO 5006 IX = 1,NX CH2 215
      DO 5006 IY = 1,NY CH2 216
      INDX = (IG-1)*NP + (2*IY-3)*NPX + 2*IX-2 CH2 217
      INDX = (3*IY-4)*NPXJ + 3*IX-3 CH2 218

```

DO 5001 I1 = 1,3	CH2 219
DO 5006 I2 = 1,3	CH2 220
INDX2 = INDX + I2*NPX + I1	CH2 221
INDX3 = INDX1 + I2*NPXJ + I1	CH2 222
5006 YDS(INDX3) = Y(INDX2)	CH2 223
GO TC 5000	CH2 224
5007 IG = 0	CH2 225
KRET = 2	CH2 226
5008 IG = IG + 1	CH2 227
IF(IG-NJ)5009,5009,5011	CH2 228
5009 WRITE OUTPUT TAPE 6,120	CH2 229
WRITE OUTPUT TAPE 6,130,IG	CH2 230
130 FORMAT(5X19HPRECURSOR GROUP NO.13)	CH2 231
DO 5010 I = 1,NPJ	CH2 232
INDX = NG*NP + (IG-1)*NPJ + I	CH2 233
5010 YDS(I) = Y(INDX)	CH2 234
GO TC 5000	CH2 235
5011 CONTINUE	CH2 236
CALL FDPRNT	CH2 237
GO TC (49,61,50),JA	CH2 238
61 TP=TP+PP	CH2 239
GO TC 39	CH2 240
C	CH2 241
C BOX C (STATEMENT OF NON-CONVERGENCE)	CH2 242
4000 JA=3	CH2 243
GO TC 45	CH2 244
50 WRITE OUTPUT TAPE 6,51,NPROB	CH2 245
51 FORMAT(27HNONCONVERGENCE PROBLEM NO. 13)	CH2 246
52 IF(NSTOP-2)53,99,99	CH2 247
53 READ INPLT TAPE 5,114,DT,PP,E1,E2,E3,E4,NSTOP,NCHGE,NORM,NGOUT	CH2 248
CHGE=NCHGE	CH2 249
NSET2=CHGE/8.0+0.5	CH2 250
NCARD=2*NSET2	CH2 251
IF(NCARD)52,52,55	CH2 252
55 DO 56 I=1,NCARD	CH2 253
56 READ INPLT TAPE 5,57,K	CH2 254
57 FORMAT(I1)	CH2 255
GO TC 52	CH2 256
C	CH2 257
C BOX A (DERIVATIVE COMPUTATIONS)	CH2 258
1000 TC1=(TF-T)/DT	CH2 259
TC2 = 1.0-TC1	CH2 260
DO 1001 IM=1,NM	CH2 261
DO 1002 IG=1,NG	CH2 262
D3(IG,IM) = 1.0/(TC1/D1(IG,IM) + TC2/D2(IG,IM))	CH2 263
V3(IG,IM)=TC1*V1(IG,IM)+TC2*V2(IG,IM)	CH2 264
VDV=C.0	CH2 265
IF(IG-NGOUT)1049,1049,1048	CH2 266

1048	VDV=VD3(IG,IM)/V3(IG,IM)**2	CH2	267
1049	DO 1CC3 IGP=1,NG	CH2	268
1003	A3(IG,IGP,IM)=TC1*A1(IG,IGP,IM)+TC2*A2(IG,IGP,IM)	CH2	269
	A3(IG,IG,IM)=A3(IG,IG,IM)+VDV	CH2	270
	DO 1C02 IJ=1,NJ	CH2	271
	AFD3(IG,IJ,IM)=TC1*AFD1(IG,IJ,IM)+TC2*AFD2(IG,IJ,IM)	CH2	272
1002	ADF3(IJ,IG,IM)=TC1*ADF1(IJ,IG,IM)+TC2*ADF2(IJ,IG,IM)	CH2	273
	DO 1CC1 IJ=1,NJ	CH2	274
1001	XLAM3(IJ,IM)=TC1*XLAM1(IJ,IM)+TC2*XLAM2(IJ,IM)	CH2	275
	WRITE OUTPUT TAPE 6,1401,T	CH2	276
1401	FORMAT(64X3HT =1PE14.7)	CH2	277
C		CH2	278
C	START FLUXES	CH2	279
	IF(NGOUT)1004,10C4,1054	CH2	280
1054	WRITE OUTPUT TAPE 6,808	CH2	281
808	FORMAT(4CHOENTRY TO SUBROUTINE FLXCUT FROM BOX A.)	CH2	282
	CALL FLXCUT	CH2	283
1343	IF(NGIN)1043,1043,1004	CH2	284
C		CH2	285
C	DERIVATIVE COMPUTATION USING CYLINDRICAL COORDINATES	CH2	286
1004	DO 1C24 IG=NGIN1,NG	CH2	287
	DO 1C24 I=1,NX	CH2	288
	DO 1024 J=1,NY	CH2	289
	L1=NG*NP-NPJ+(3*J-4)*NPXJ+3*I-3	CH2	290
	INDX=(IG-1)*NP+(2*J-1)*NPX+2*I	CH2	291
	IM=M(I,J)	CH2	292
	SUM=C.0	CH2	293
	DO 1C07 I1=1,3	CH2	294
	DO 1C07 J1=1,3	CH2	295
	INDX1= INDX+(J1-2)*NPX+I1-2	CH2	296
	SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,2)*PRR(J1,2,J)/DX(I)**2+P(I1,2)*	CH2	297
	1QRR(J1,2,J))	CH2	298
	SUM1=0.0	CH2	299
	DO 1C05 IGP=1,NG	CH2	300
	INDX2= INDX1+(IGP-IG)*NP	CH2	301
1005	SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	CH2	302
	DO 1C06 IJ=1,NJ	CH2	303
	INDX2 = L1 + IJ*NPJ + J1*NPXJ + I1	CH2	304
1006	SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)	CH2	305
1007	SUM=SUM+SUM1*P(I1,2)*PRR(J1,2,J)	CH2	306
	YDS(INDX)=SUM*V3(IG,IM)/(16.0*PRR(2,2,J))	CH2	307
	IF(NX-I)1013,1013,1008	CH2	308
1008	INDX=INDX+1	CH2	309
	SUMS=C.0	CH2	310
	SUMD=C.0	CH2	311
	DO 1C12 IS=1,2	CH2	312
	L2 = 1 + 3* IS-3	CH2	313
	IF(L2=1)	CH2	314


```

IM=M(I,J)
SUM=C.0
SUMD=SUMD+DX(II)/V3(IG,IM)
I2=5-2*IS
DO 1011 I1=1,3
DO 1011 J1=1,3
INDX1=INDX+(J1-2)*NPX+I1-I2
SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,I2)*PRR(J1,2,J)/DX(II)**2+P(I1,I2CH2 322
1)*QRR(J1,2,J))
SUM1=C.0
DO 1009 IGP=1,NG
INDX2=INDX1+(IGP-IG)*NP
1009 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)
DO 1010 IJ=1,NJ
INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1
1010 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)
1011 SUM=SUM+SUM1*P(I1,I2)*PRR(J1,2,J)
1012 SUMS=SUMS+SUM*DX(II)
DEN(INDX)=4.0*SUMD*PRR(2,2,J)
YDS(INDX)=SUMS/DEN(INDX)
1013 IF(NY-J)1024,1024,1014
1014 INDX=(IG-1)*NP+2*J*NPX+2*I
SUMS=C.0
SUMD=C.0
DO 1018 JS=1,2
L2 = L1 + (3*JS-3)*NPXJ
JJ=J+JS-1
IM=M(I,JJ)
SUM=C.0
J2=5-2*JS
SUMD=SUMD+PRR(J2,J2,JJ)/V3(IG,IM)
DO 1017 I1=1,3
DO 1017 J1=1,3
INDX1= INDX+(J1-J2)*NPX+I1-2
SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,2)*PRR(J1,J2,JJ)/DX(I)**2+P(I1,2)CH2 349
1)*QRR(J1,J2,JJ))
SUM1=C.0
DO 1015 IGP=1,NG
INDX2=INDX1+(IGP-IG)*NP
1015 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)
DO 1016 IJ=1,NJ
INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1
1016 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)
1017 SUM=SUM+SUM1*P(I1,2)*PRR(J1,J2,JJ)
1018 SUMS=SUMS+SUM
DEN(INDX)=16.0*SUMD
YDS(INDX)=SUMS/DEN(INDX)
IF(NX-I)1024,1024,1019

```

CH2 315
CH2 316
CH2 317
CH2 318
CH2 319
CH2 320
CH2 321
CH2 322
CH2 323
CH2 324
CH2 325
CH2 326
CH2 327
CH2 328
CH2 329
CH2 330
CH2 331
CH2 332
CH2 333
CH2 334
CH2 335
CH2 336
CH2 337
CH2 338
CH2 339
CH2 340
CH2 341
CH2 342
CH2 343
CH2 344
CH2 345
CH2 346
CH2 347
CH2 348
CH2 349
CH2 350
CH2 351
CH2 352
CH2 353
CH2 354
CH2 355
CH2 356
CH2 357
CH2 358
CH2 359
CH2 360
CH2 361
CH2 362

1019	INDX=INDX+1	CH2 363
	SUMS=C.0	CH2 364
	SUMD=C.0	CH2 365
	DO 1023 IS=1,2	CH2 366
	II=I+IS-1	CH2 367
	I2=5-2*IS	CH2 368
	DO 1023 JS=1,2	CH2 369
	L2 = L1 + (3*JS-3)*NPXJ + 3*IS-3	CH2 370
	JJ=J+JS-1	CH2 371
	IM=M(II,JJ)	CH2 372
	J2=5-2*JS	CH2 373
	SUM=C.0	CH2 374
	SUMD=SUMD+DX(II)*PRR(J2,J2,JJ)/V3(IG,IM)	CH2 375
	DO 1022 I1=1,3	CH2 376
	DO 1022 J1=1,3	CH2 377
	INDX1=INDX+(J1-J2)*NPX+I1-I2	CH2 378
	SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,I2)*PRR(J1,J2,JJ)/DX(II)**2+P(I1,	CH2 379
	I12)*QRR(J1,J2,JJ))	CH2 380
	SUM1=0.0	CH2 381
	DO 1020 IGP=1,NG	CH2 382
	INDX2=INDX1+(IGP-IG)*NP	CH2 383
1020	SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	CH2 384
	DO 1021 IJ=1,NJ	CH2 385
	INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1	CH2 386
1021	SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)	CH2 387
1022	SUM=SUM+SUM1*P(I1,I2)*PRR(J1,J2,JJ)	CH2 388
1023	SUMS=SUMS+SUM*DX(II)	CH2 389
	DEN(INDX)=4.0*SUMD	CH2 390
	YDS(INDX)=SUMS/DEN(INDX)	CH2 391
1024	CONTINUE	CH2 392
	KIT = 0	CH2 393
1025	KKIT = 1	CH2 394
	DO 1042 IG=NGIN1,NG	CH2 395
	DO 1039 I=1,NX	CH2 396
	DO 1037 J=1,NY	CH2 397
	INDX= (IG-1)*NP+(2*J-1)*NPX+2*I	CH2 398
	SUM=C.0	CH2 399
	DO 1026 I1=1,3	CH2 400
	DO 1026 J1=1,3	CH2 401
	INDX1=INDX +I1-2+(J1-2)*NPX	CH2 402
1026	SUM=SUM+P(I1,2)*PRR(J1,2,J)*YD(INDX1)	CH2 403
	S = YDS(INDX) - SLM /(16.*PRR(2,2,J))	CH2 404
	SUM1 = YD(INDX)	CH2 405
	GO TO (1381,1391),KKIT	CH2 406
1381	KRET = 1	CH2 407
	GO TO 1370	CH2 408
1391	YD(INDX) = SUM1 +S	CH2 409
	IF(NX) 1030,1030,1027	CH2 410

```

1027 INDX=INDX+1
      SUMS=C.0
      DO 1029 IS=1,2
        II=I+IS-1
        IM=M(II,J)
        SUM=C.0
        I2=5-2*IS
        DO 1028 I1=1,3
          DO 1028 J1=1,3
            INDX1=INDX+I1-I2+(J1-2)*NPX
1028    SUM=SUM+YD(INDX1)*PRR(J1,2,J)*P(I1,I2)
1029    SUMS=SUMS+SUM*DX(II)/V3(IG,IM)
        S = YDS(INDX) - SUMS/DEN(INDX)
        SUM1 = YC(INDX)
        GO TO (1382,1392),KKIT
1382 KRET = 2
      GO TO 137C
1392 YD(INDX) = SUM1 +S
1030 IF(NY-J)1037,1037,1031
1031 INDX=(IG-1)*NP+2*J*NPX+2*I
      SUMS=0.0
      DO 1033 JS=1,2
        JJ=J+JS-1
        IM=M(I,JJ)
        SUM=C.0
        J2=5-2*JS
        DO 1032 I1=1,3
          DO 1032 J1=1,3
            INDX1=INDX+(J1-J2)*NPX+I1-2
1032    SUM=SUM+YD(INDX1)*PRR(J1,J2,JJ)*P(I1,2)
1033    SUMS=SUMS+SUM/V3(IG,IM)
        S = YDS(INDX) - SUMS/DEN(INDX)
        SUM1 = YC(INDX)
        GO TO (1383,1393),KKIT
1383 KRET = 3
      GO TO 137C
1393 YD(INDX) = SUM1 +S
      IF(NX-I)1037,1037,1034
1034 INDX=INDX+1
      SUMS=C.0
      DO 1036 IS=1,2
        II=I+IS-1
        I2=5-2*IS
        DO 1036 JS=1,2
          JJ=J+JS-1
          IM=M(II,JJ)
          J2=5-2*JS
          SUM=C.0

```

```

CH2 411
CH2 412
CH2 413
CH2 414
CH2 415
CH2 416
CH2 417
CH2 418
CH2 419
CH2 420
CH2 421
CH2 422
CH2 423
CH2 424
CH2 425
CH2 426
CH2 427
CH2 428
CH2 429
CH2 430
CH2 431
CH2 432
CH2 433
CH2 434
CH2 435
CH2 436
CH2 437
CH2 438
CH2 439
CH2 440
CH2 441
CH2 442
CH2 443
CH2 444
CH2 445
CH2 446
CH2 447
CH2 448
CH2 449
CH2 450
CH2 451
CH2 452
CH2 453
CH2 454
CH2 455
CH2 456
CH2 457
CH2 458

```

DO 1035 I1=1,3	CH2 459
DO 1035 J1=1,3	CH2 460
INDX1=INDX+(J1-J2)*NPX+I1-I2	CH2 461
1035 SUM=SUM+P(I1,I2)*PRR(J1,J2,JJ)*YD(INDX1)	CH2 462
1036 SUMS=SUMS+SUM*DX(I1)/V3(IG,IM)	CH2 463
S = YDS(INDX) - SUMS/DEN(INDX)	CH2 464
SUM1 = YC(INDX)	CH2 465
GO TO (1384,1394),KKIT	CH2 466
1384 KRET = 4	CH2 467
GO TO 1370	CH2 468
1394 YD(INDX) = SUM1 +S	CH2 469
1037 CONTINUE	CH2 470
IF(NGEOM**2)1039,1039,1038	CH2 471
1038 INDX=(IG-1)*NP+2*I	CH2 472
INDX1=INDX+NPX	CH2 473
INDX2=INDX1+NPX	CH2 474
YD(INDX)=(4.0*YD(INDX1)-YD(INDX2))/3.0	CH2 475
YD(INDX-1)=(4.0*YD(INDX1-1)-YD(INDX2-1))/3.0	CH2 476
1039 CONTINUE	CH2 477
IF(NGEOM**2-1)1042,1042,1040	CH2 478
1040 DO 1041 J=1,NY	CH2 479
INDX=(IG-1)*NP+(2*J-1)*NPX+1	CH2 480
INDX1=INDX-NPX	CH2 481
YD(INDX)=(4.0*YD(INDX+1)-YD(INDX+2))/3.0	CH2 482
1041 YD(INDX1)=(4.0*YD(INDX1+1)-YD(INDX1+2))/3.0	CH2 483
1042 CONTINUE	CH2 484
KIT = KIT +1	CH2 485
GO TO (1044,1025),KKIT	CH2 486
1044 WRITE OUTPUT TAPE 6,1045,KIT	CH2 487
1045 FORMAT(6CX15,18H IN-ICE ITERATIONS //)	CH2 488
1043 IF(NJIN)39,39,1099	CH2 489
C	CH2 490
C FINISH FLUXES, START PRECURSORS	CH2 491
1099 L1=1	CH2 492
L2=NJIN	CH2 493
L=1	CH2 494
1100 DO 1104 I=1,NX	CH2 495
DO 1104 J=1,NY	CH2 496
IM=M(I,J)	CH2 497
INDX1=(2*J-3)*NPX+2*I-NP-2	CH2 498
INDX2=NG*NP+NPXJ*(3*J-4)+3*I-3-NPJ	CH2 499
DO 1103 I1=1,3	CH2 500
DO 1103 J1=1,3	CH2 501
INDX3=INDX1+J1*NPX+I1	CH2 502
INDX4=INDX2+J1*NPXJ+I1	CH2 503
DO 1102 IJ=L1,L2	CH2 504
INDX6=INDX4+IJ*NPJ	CH2 505
SUM=-AM3(IJ,IM)*Y(INDX6)	CH2 506

```

DO 1101 IG=1,NG
  INDX5=INDX3+IG*NP
1101 SUM=SUM+ADF3(IJ,IG,IM)*Y(INDX5)
1102 YD(INDX6)=SUM
1103 CONTINUE
1104 CONTINUE
  GO TO (39,2001),L
1370 SUM = ABSF(SUM1)
  IF(SUM-1.0E-6)1374,1374,1371
1374 SUM = 1.0
1371 SUM = ABSF(S/SUM)
  IF(SUM-E4)1373,1373,1372
1372 KKIT = 2
1373 GO TO (1391,1392,1393,1394),KRET

```

C
C

```

99 DO 800 IG=1,NG
  K1=(IG-1)*NP +1
  K2= IG*NP
800 WRITE OUTPUT TAPE 7,801,(Y(I),I=K1,K2)
801 FORMAT(5E15.8)
  DO 802 IJ=1,NJ
  K1=NG*NP +(IJ-1)*NPJ+1
  K2=K1+NPJ-1
802 WRITE OUTPUT TAPE 7,801,(Y(I),I=K1,K2)
  TIME = CLOCK+(CARD CHGE)
  WRITE OUTPUT TAPE 6,803,TIME
803 FORMAT( /// 61X16F"COMPUTER TIME = "3PF6.3 )
  CALL CHAIN(1,4)
  END
  SUBROUTINE FLXCUT

```

CH2 507
CH2 508
CH2 509
CH2 510
CH2 511
CH2 512
CH2 513
CH2 514
CH2 515
CH2 516
CH2 517
CH2 518
CH2 519
CH2 520
CH2 521
CH2 522
CH2 523
CH2 524
CH2 525
CH2 526
CH2 527
CH2 528
CH2 529
CH2 530
CH2 531
CH2 532
CH2 533
CH2 534
CH2 535
CH2 536

FLX 1
FLX 2
FLX 3
FLX 4
FLX 5
FLX 6
FLX 7
FLX 8
FLX 9
FLX 10
FLX 11

C

```

  EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)FLX 12
  1,(X5N(2121),BTCT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781), FLX 13
  2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510FLX 14
  31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550FLX 15
  41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1), FLX 16
  5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2), FLX 17
  6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128FLX 18

```

C

```

71),BETA2),(W2(14C1),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)FLX 19
8,ADF2) FLX 20
FLX 21
COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VC3,Y,YC,YDS,M,DX,DY,P,Q,PRFLX 22
1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,FLX 23
2N,PP,T,TF,DT,II,NGES,INTPX,INTPY,ISSW FLX 24
COMMON DEN,NGECM,NGIN,NGIN1,NGCUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,FLX 25
1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUMC,SUMS,KIT,KKIT,KRETFX 26
2,E1,E2,E3,E4,ITINN,ITOUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW FLX 27
SIZES = C.0 FLX 28
DO 1074 IG = 1,NGCUT FLX 29
DO 1074 I=1,NX FLX 30
DO 1074 J=1,NY FLX 31
INDX=(IG-1)*NP+(2*J-1)*NPX+2*I FLX 32
L1=NG*NP-NPJ+(3*J-4)*NPXJ+3*I-3 FLX 33
IM=M(I,J) FLX 34
SUM=C.0 FLX 35
DO 1057 I1=1,3 FLX 36
DO 1057 J1=1,3 FLX 37
INDX1= INDX+(J1-2)*NPX+I1-2 FLX 38
SUM1=C.0 FLX 39
IF(NGIN)1201,1201,1200 FLX 40
1200 DO 1055 IGP=NGIN1,NG FLX 41
INDX2= INDX1+(IGP-IG)*NP FLX 42
1055 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2) FLX 43
1201 DO 1056 IJ=1,NJ FLX 44
INDX2 = L1 + IJ*NPJ + J1*NPXJ + I1 FLX 45
1056 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2) FLX 46
1057 SUM=SUM+SUM1*P(I1,2)*PRR(J1,2,J) FLX 47
DEN(INDX) = 16.C*PRR(2,2,J)*A3(IG,IG,IM)-D3(IG,IM)*(C(2,2)*PRR(2, FLX 48
12,J)/DX(I)**2 +16.C*QRR(2,2,J)) FLX 49
YDS(INDX) = SUM/DEN(INDX) FLX 50
SIZES = SIZES + DX(I)*SUM FLX 51
IF(NX-I)1063,1063,1058 FLX 52
1058 INDX=INDX+1 FLX 53
SUMS=C.0 FLX 54
SUMD=C.0 FLX 55
DO 1062 IS=1,2 FLX 56
L2 = L1 + 3* IS-3 FLX 57
II=I+IS-1 FLX 58
IM=M(II,J) FLX 59
SUM=C.C FLX 60
SUMD = SUMD + DX(II)*PRR(2,2,J)*A3(IG,IG,IM)-C3(IG,IM)*(17.5*PRR( FLX 61
12,2,J)/DX(II) + QRR(2,2,J)*DX(II)) FLX 62
I2=5-2*IS FLX 63
DO 1061 I1=1,3 FLX 64
DO 1061 J1=1,3 FLX 65
INDX1= INDX+(J1-2)*NPX+I1-I2 FLX 66

```

SUM1=	FLX	67
IF(NGIN)1203,1203,1202	FLX	68
1202 DO 1059 IGP = NGIN1,NG	FLX	69
INDX2=INDX1+(IGP-IG)*NP	FLX	70
1059 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX	71
1203 DO 1060 IJ=1,NJ	FLX	72
INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1	FLX	73
1060 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)	FLX	74
1061 SUM=SUM+SUM1*P(I1,I2)*PRR(J1,2,J)	FLX	75
1062 SUMS=SUMS+SUM*DX(I1)	FLX	76
DEN(INDX)=4.C*SUMC	FLX	77
YDS(INDX) = SUMS/DEN(INDX)	FLX	78
SIZES = SIZES + SUMS	FLX	79
1063 IF(NY-J)1074,1074,1064	FLX	80
1064 INDX=(IG-1)*NP+2*J*NPX+2*I	FLX	81
SUMS=C.0	FLX	82
SUMD=C.0	FLX	83
DO 1068 JS=1,2	FLX	84
L2 = L1 + (3*JS-3)*NPXJ	FLX	85
JJ=J+JS-1	FLX	86
IM=M(I,JJ)	FLX	87
SUM=C.0	FLX	88
J2=5-2*JS	FLX	89
SUMD = SUMD + PRR(J2,J2,JJ)*A3(IG,IG,IM)-D3(IG,IM)*((10.0*PRR(J2,J2	FLX	90
1,JJ)/DX(I)**2 + QRR(J2,J2,JJ))	FLX	91
DO 1067 I1=1,3	FLX	92
DO 1067 J1=1,3	FLX	93
INDX1= INDX+(J1-J2)*NPX+I1-2	FLX	94
SUM1=C.0	FLX	95
IF(NGIN)1205,1205,1204	FLX	96
1204 DO 1065 IGP = NGIN1,NG	FLX	97
INDX2=INDX1+(IGP-IG)*NP	FLX	98
1065 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX	99
1205 DO 1066 IJ=1,NJ	FLX	100
INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1	FLX	101
1066 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)	FLX	102
1067 SUM=SUM+SUM1*P(I1,2)*PRR(J1,J2,JJ)	FLX	103
1068 SUMS=SUMS+SUM	FLX	104
DEN(INDX)=16.C*SUMD	FLX	105
YDS(INDX) = SUMS/DEN(INDX)	FLX	106
SIZES = SIZES + SUMS*DX(I)	FLX	107
IF(NX-I)1074,1074,1069	FLX	108
1069 INDX=INDX+1	FLX	109
SUMS=C.0	FLX	110
SUMD=C.0	FLX	111
DO 1073 IS=1,2	FLX	112
I1=I+IS-1	FLX	113
I2=5-2*IS	FLX	114

DO 1073 JS=1,2	FLX 115
L2 = L1 + (3*JS-3)*NPXJ + 3*IS-3	FLX 116
JJ=J+JS-1	FLX 117
IM=M(II,JJ)	FLX 118
J2=5-2*JS	FLX 119
SUM=C.0	FLX 120
SUMC = SUMC + A3(IG,IG,IM)*PRR(J2,J2,JJ)*DX(II)-D3(IG,IM)*(QRR(J2,	FLX 121
1J2,JJ)*DX(II)+17.5*PRR(J2,J2,JJ)/DX(II))	FLX 122
DO 1072 I1=1,3	FLX 123
DO 1072 J1=1,3	FLX 124
INDX1=INDX+(J1-J2)*NPX+I1-I2	FLX 125
SUM1=C.0	FLX 126
IF(NGIN)1207,1207,1206	FLX 127
1206 DO 1070 IGP = NGIN1,NG	FLX 128
INDX2=INDX1+(IGP-IG)*NP	FLX 129
1070 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX 130
1207 DO 1071 IJ=1,NJ	FLX 131
INDX2 = L2 + IJ*NPJ + J1*NPXJ + I1	FLX 132
1071 SUM1=SUM1+AFD3(IG,IJ,IM)*Y(INDX2)	FLX 133
1072 SUM=SUM+SUM1*P(I1,I2)*PRR(J1,J2,JJ)	FLX 134
1073 SUMS=SUMS+SUM*DX(II)	FLX 135
DEN(INDX)=4.0*SUMC	FLX 136
YDS(INDX)=SUMS/DEN(INDX)	FLX 137
SIZES = SIZES + SUMS	FLX 138
1074 CONTINUE	FLX 139
KIT = 0	FLX 140
1400 SUM = 0.0	FLX 141
DO 1419 I =1,NX	FLX 142
S = CX(I)	FLX 143
IF(I-1)1401,1401,1402	FLX 144
1401 I21 = 2	FLX 145
GO TO 1403	FLX 146
1402 I21 = 1	FLX 147
1403 IF(I-NX)1405,1404,1404	FLX 148
1404 I22 = 2	FLX 149
GO TO 1406	FLX 150
1405 I22 = 3	FLX 151
1406 DO 1419 J=1,NY	FLX 152
IM = M(I,J)	FLX 153
IF(J-1)1407,1407,1408	FLX 154
1407 J21 = 2	FLX 155
GO TO 1409	FLX 156
1408 J21 = 1	FLX 157
1409 IF(J-NY)1411,1410,1410	FLX 158
1410 J22 = 2	FLX 159
GO TO 1412	FLX 160
1411 J22 =	FLX 161
2 I 41 =1	FLX 162

SUM1 = 0.C	FLX 163
SUM2 = 0.C	FLX 164
DO 1413 I2 = I21,I22	FLX 165
SUM1=SUM1+P(I1,I2)	FLX 166
1413 SUM2=SUM2+Q(I1,I2)	FLX 167
YD(I1) = SUM1*S	FLX 168
YD(I1+3) = SUM2/S	FLX 169
SUM1 = 0.C	FLX 170
SUM2 = 0.C	FLX 171
DO 1414 J2= J21,J22	FLX 172
SUM1 = SUM1+ PRR(I1,J2,J)	FLX 173
1414 SUM2 = SUM2+ QRR(I1,J2,J)	FLX 174
YD(I1+6) = SUM1	FLX 175
1415 YD(I1+9) = SUM2	FLX 176
DO 1418 IG =1,NGOUT	FLX 177
INDX = (IG-1)*NP + (2*J-1)*NPX + 2*I	FLX 178
SUMD = 0.C	FLX 179
DO 1416 IGP = 1,NGOUT	FLX 180
1416 SUMD = SUMD + A3(IGP,IG,IM)	FLX 181
SUM1=C.0	FLX 182
SUM2=C.0	FLX 183
DO 1417 I1=1,3	FLX 184
DO 1417 J1=1,3	FLX 185
INDX1 = INDX + I1-2+ (J1-2)*NPX	FLX 186
SUM1 = SUM1 + Y(INDX1)*YD(I1)*YD(J1+6)	FLX 187
1417 SUM2 = SUM2 + Y(INDX1)*(YD(I1)*YD(J1+9) + YD(I1+3)*YD(J1+6))	FLX 188
1418 SUM = SUM +SUM1*SUMD -D3(IG,IM)*SUM2	FLX 189
1419 CONTINUE	FLX 190
S = -SIZES/SUM	FLX 191
J = NP*NGOUT	FLX 192
DO 1420 I =1,J	FLX 193
1420 Y(I) = Y(I)*S	FLX 194
1300 KKIT = 1	FLX 195
DO 1342 IG = 1,NGOUT	FLX 196
DO 1339 I=1,NX	FLX 197
DO 1324 J=1,NY	FLX 198
INDX=(IG-1)*NP+(2*J-1)*NPX+2*I	FLX 199
IM=M(I,J)	FLX 200
SUM=C.0	FLX 201
DO 1307 I1=1,3	FLX 202
DO 1307 J1=1,3	FLX 203
INDX1= INDX+(J1-2)*NPX+I1-2	FLX 204
SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,2)*PRR(J1,2,J)/DX(I)**2+P(I1,2)*	FLX 205
1QRR(J1,2,J))	FLX 206
SUM1=C.0	FLX 207
DO 1305 IGP = 1,NGOUT	FLX 208
INDX2= INDX1+(IGP-IG)*NP	FLX 209
1305 SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX 210

1307	SUM=SUM+SUM1*P(I1,2)*PRR(J1,2,J)	FLX 211
	S = SUM/DEN(INDX) + YDS(INDX)	FLX 212
	SUM1 = Y(INDX)	FLX 213
	GO TO (1351,1361),KKIT	FLX 214
1351	KRET = 1	FLX 215
	GO TO 137C	FLX 216
1361	Y(INDX) = SUM1 -S	FLX 217
	IF(NX-I)1313,1313,13C8	FLX 218
1308	INDX=INDX+1	FLX 219
	SUMS=C.0	FLX 220
	DO 1312 IS=1,2	FLX 221
	II=I+IS-1	FLX 222
	IM=M(II,J)	FLX 223
	SUM=C.0	FLX 224
	I2=5-2*IS	FLX 225
	DO 1311 I1=1,3	FLX 226
	DO 1311 J1=1,3	FLX 227
	INDX1=INDX+(J1-2)*NPX+I1-I2	FLX 228
	SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,I2)*PRR(J1,2,J)/DX(II)**2+P(I1,I2	FLX 229
	1)*QRR(J1,2,J))	FLX 230
	SUM1=C.0	FLX 231
	DO 1309 IGP=1,NGCLT	FLX 232
	INDX2=INDX1+(IGP-IG)*NP	FLX 233
1309	SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX 234
1311	SUM=SUM+SUM1*P(I1,I2)*PRR(J1,2,J)	FLX 235
1312	SUMS = SUMS +SUM*CX(II)	FLX 236
	S =SUMS/DEN(INDX) + YDS(INDX)	FLX 237
	SUM1 = Y(INDX)	FLX 238
	GO TO (1352,1362),KKIT	FLX 239
1352	KRET = 2	FLX 240
	GO TO 137C	FLX 241
1362	Y(INDX) = SUM1 -S	FLX 242
1313	IF(NY-J)1324,1324,1314	FLX 243
1314	INDX=(IG-1)*NP+2*J*NPX+2*I	FLX 244
	SUMS=0.0	FLX 245
	DO 1318 JS=1,2	FLX 246
	JJ=J+JS-1	FLX 247
	IM=M(I,JJ)	FLX 248
	SUM=C.0	FLX 249
	J2=5-2*JS	FLX 250
	DO 1317 I1=1,3	FLX 251
	DO 1317 J1=1,3	FLX 252
	INDX1= INDX+(J1-J2)*NPX+I1-2	FLX 253
	SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,2)*PRR(J1,J2,JJ)/DX(I)**2+P(I1,2)	FLX 254
	1)*QRR(J1,J2,JJ))	FLX 255
	SUM1=0.0	FLX 256
	DO 1315 IGP=1,NGOLT	FLX 257
	INDX2=INDX1+(IGP-IG)*NP	FLX 258

1315	SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX 259
1317	SUM=SUM+SUM1*P(I1,2)*PRR(J1,J2,JJ)	FLX 260
1318	SUMS=SUMS+SUM	FLX 261
	S =SUMS/DEN(INDX) + YDS(INDX)	FLX 262
	SUM1 = Y(INDX)	FLX 263
	GO TO (1353,1363),KKIT	FLX 264
1353	KRET = 3	FLX 265
	GO TO 1370	FLX 266
1363	Y(INDX) = SUM1 -S	FLX 267
	IF(NX-I)1324,1324,1319	FLX 268
1319	INDX=INDX+1	FLX 269
	SUMS=C.0	FLX 270
	DO 1323 IS=1,2	FLX 271
	II=I+IS-1	FLX 272
	I2=5-2*IS	FLX 273
	DO 1323 JS=1,2	FLX 274
	JJ=J+JS-1	FLX 275
	IM=M(II,JJ)	FLX 276
	J2=5-2*JS	FLX 277
	SUM=C.0	FLX 278
	DO 1322 I1=1,3	FLX 279
	DO 1322 J1=1,3	FLX 280
	INDX1=INDX+(J1-J2)*NPX+I1-I2	FLX 281
	SUM=SUM-D3(IG,IM)*Y(INDX1)*(Q(I1,I2)*PRR(J1,J2,JJ)/DX(II)**2+P(I1,	FLX 282
	I12)*QRR(J1,J2,JJ))	FLX 283
	SUM1=C.0	FLX 284
	DO 1320 IGP=1,NGOUT	FLX 285
	INDX2=INDX1+(IGP-IG)*NP	FLX 286
1320	SUM1=SUM1+A3(IG,IGP,IM)*Y(INDX2)	FLX 287
1322	SUM=SUM+SUM1*P(I1,I2)*PRR(J1,J2,JJ)	FLX 288
1323	SUMS=SUMS+SUM*DX(II)	FLX 289
	S =SUMS/DEN(INDX) + YDS(INDX)	FLX 290
	SUM1 = Y(INDX)	FLX 291
	GO TO (1354,1364),KKIT	FLX 292
1354	KRET = 4	FLX 293
	GO TO 1370	FLX 294
1364	Y(INDX) = SUM1 -S	FLX 295
1324	CONTINUE	FLX 296
	IF(NGECM**2)1339,1339,1338	FLX 297
1338	INDX=(IG-1)*NP+2*I	FLX 298
	INDX1=INDX+NPX	FLX 299
	INDX2=INDX1+NPX	FLX 300
	Y (INDX)=(4.C*Y (INDX1)-Y (INDX2))/3.0	FLX 301
	Y (INDX-1)=(4.C*Y (INDX1-1)-Y (INDX2-1))/3.0	FLX 302
1339	CONTINUE	FLX 303
	IF(NGEOM**2-1)1342,1342,1340	FLX 304
1340	DO 1341 J=1,NY	FLX 305
	INDX=(IG-1)*NP+(2*J-1)*NPX+1	FLX 306

```

      INDX1=INDX-NPX
      Y (INDX)=(4.0*Y (INDX+1)-Y (INDX+2))/3.0
1341 Y (INDX1)=(4.0*Y (INDX1+1)-Y (INDX1+2))/3.0
1342 CONTINUE
      KIT = KIT +1
      GO TO (1349,1400),KKIT
1349 WRITE OUTPUT TAPE 6,1348,KIT
1348 FORMAT(6CX15,22H CUT-OF-ICE ITERATIONS )
      RETURN
1370 SUM = ABSF(SUM1)
      IF(SUM-1.0E-6)1373,1373,1371
1371 SUM = ABSF(S/SUM)
      IF(SUM-E3)1373,1373,1372
1372 KKIT = 2
1373 GO TO (1361,1362,1363,1364),KRET
      END
      SUBROUTINE FDSRT
      DIMENSION P(3,3),C(3,3),PR(3,3),QR(3,3),M(7,7),DX(7),DY(7),PRR(3,3)
      1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20),
      2SF11(4,20),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20),
      3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4
      4,4,20),SABS2(4,20),SF12(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,
      520),BETA2(6,20),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20),
      6A3(4,4,20),AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20
      7),VD3(4,20),W1(2800),W2(2800),Y(2620),YD(2620),YDS( 676),X5N( 6780
      8),BTOT1(20),B(4,4,20),DEN(676),YS(1690)
      EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)
      1,(X5N(2121),BTOT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781),
      2SABS1),(X5N(4861),SF11),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510
      31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550
      41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1),
      5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SF12),
      6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128
      71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)
      8,ADF2)
      COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YC,YDS,M,CX,CY,P,C,PR,FDS
      1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,FDS
      2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW
      COMMON DEN,NGECM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,FDS
      1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUMD,SUMS,KIT,KKIT,KRETFDS
      2,E1,E2,E3,E4,ITINN,ITOUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW
      RETURN
      END
      SUBROUTINE FEEDBK
      DIMENSION P(3,3),C(3,3),PR(3,3),QR(3,3),M(7,7),DX(7),DY(7),PRR(3,3)
      1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20),

```

2SFI1(4,2C),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20), FBK 4
 3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4,FBK 5
 4,4,20),SABS2(4,2C),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,FBK 6
 520),BETA2(6,2C),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20), FBK 7
 6A3(4,4,20), AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,2C),D3(4,20),V3(4,20FBK 8
 7),VD3(4,20),W1(28C0),W2(2800),Y(2620),YD(2620),YDS(676),X5N(6780FBK 9
 8),BTCT1(20),B(4,4,20),DEN(676),YS(1690) FBK 10

EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)FBK 11
 1,(X5N(2121),BTOT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781), FBK 12
 2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510FBK 13
 31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550FBK 14
 41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1), FBK 15
 5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2), FBK 16
 6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128FBK 17
 71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)FBK 18
 8,ADF2) FBK 19
 FBK 20

COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YC,YDS,M,CX,DY,P,Q,PRFBK 21
 1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,AP,NPJ,FBK 22
 2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW FBK 23
 COMMON DEN,NGECM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,FBK 24
 1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUMD,SUMS,KIT,KKIT,KRETFBK 25
 2,E1,E2,E3,E4,ITINN,ITCUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW FBK 26
 RETURN FBK 27
 END FBK 28
 SUBROUTINE FDPRT FBK 29

DIMENSION P(3,3),Q(3,3),PR(3,3),QR(3,3),M(7,7),CX(7),DY(7),PRR(3,3FDP 1
 1,7),QRR(3,3,7),NIN(8),WIN(8),S1(4,6,20),STR1(4,4,20),SABS1(4,20), FDP 2
 2SFI1(4,2C),CHI1(4,20),XNU1(4,20),D1(4,20),V1(4,20),BETA1(6,20), FDP 3
 3XLAM1(6,2C),A1(4,4,20),AFD1(4,6,20),ADF1(6,4,20),S2(4,6,20),STR2(4FDP 4
 4,4,20),SABS2(4,2C),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4,FDP 5
 520),BETA2(6,2C),XLAM2(6,2C),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20), FDP 6
 6A3(4,4,2C), AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20FDP 7
 7),VD3(4,20),W1(28C0),W2(28C0),Y(2620),YD(2620),YDS(676),X5N(6780FDP 8
 8),BTCT1(2C),B(4,4,20),DEN(676),YS(1690) FDP 9
 FDP 10

EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN)FDP 11
 1,(X5N(2121),BTOT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781), FDP 12
 2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510FDP 13
 31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550FDP 14
 41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1), FDP 15
 5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2), FDP 16
 6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128FDP 17
 71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),AFD2),(W2(2321)FDP 18
 8,ADF2) FDP 19
 FDP 20

COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YC,YDS,M,CX,DY,P,Q,PRFDP 21
 FDP 22

```
1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,FDP 23
2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW FCP 24
COMMON DEN,NGEOM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,FDP 25
1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUM2,SUMS,KIT,KKIT,KRETFDP 26
2,E1,E2,E3,E4,ITINN,ITCUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW FCP 27
RETURN FCP 28
END FCP 29
SUBROUTINE PRINT PNT 1
```

C

```
PNT 2
DIMENSION P(3,3),C(3,3),PRR(3,3),M(7,7),DX(7),DY(7),PNT 3
1,7},QRR(3,3,7),NIN(8),WIN(8),S1(4,4),STR1(4,4),SABS1(4,20), PNT 4
2SFI1(4,20),CHI1(4,20),XNU1(4,20),V1(4,20),BETA1(6,20), PNT 5
3XLAM1(6,20),A1(4,4,20),AFD1(4,6,20),ADF1(4,4,20),S2(4,6,20),STR2(4, PNT 6
4,4,20),SABS2(4,20),SFI2(4,20),CHI2(4,20),XNU2(4,20),D2(4,20),V2(4, PNT 7
520),BETA2(6,20),XLAM2(6,20),A2(4,4,20),AFD2(4,6,20),ADF2(6,4,20), PNT 8
6A3(4,4,20),AFD3(4,6,20),ADF3(6,4,20),XLAM3(6,20),D3(4,20),V3(4,20) PNT 9
7),VD3(4,20),W1(2800),W2(2800),Y(2620),YD(2620),YDS(676),X5N(6780) PNT 10
8),BTCT1(20),B(4,4,20),DEN(676),YS(1690) PNT 11
```

C

```
PNT 12
EQUIVALENCE (X5N,YS),(X5N(1691),B),(X5N(2101),NIN),(X5N(2111),WIN) PNT 13
1,(X5N(2121),BTCT1),(X5N(3981),W1,S1),(X5N(4461),STR1),(X5N(4781), PNT 14
2SABS1),(X5N(4861),SFI1),(X5N(4941),CHI1),(X5N(5021),XNU1),(X5N(510 PNT 15
31),D1),(X5N(5181),V1),(X5N(5261),BETA1),(X5N(5381),XLAM1),(X5N(550 PNT 16
41),A1),(X5N(5821),AFD1),(X5N(6301),ADF1), PNT 17
5 (W2,S2),(W2(481),STR2),(W2(801),SABS2),(W2(881),SFI2), PNT 18
6(W2(961),CHI2),(W2(1041),XNU2),(W2(1121),D2),(W2(1201),V2),(W2(128 PNT 19
71),BETA2),(W2(1401),XLAM2),(W2(1521),A2),(W2(1841),ADF2),(W2(2321) PNT 20
8,ADF2) PNT 21
```

C

```
PNT 22
COMMON X5N,W2,A3,AFD3,ADF3,XLAM3,D3,V3,VD3,Y,YD,YDS,M,DX,DY,P,C,PRPNT 23
```

C

```
1,QR,PRR,QRR,NPROB,NG,NJ,NM,NX,NY,NX2,NY2,NPX,NPY,NPXJ,NPYJ,NP,NPJ,PNT 24
2N,PP,T,TF,DT,TI,NGES,INTPX,INTPY,ISSW PNT 25
```

C

```
PNT 26
COMMON DEN,NGEOM,NGIN,NGIN1,NGOUT,IG,IGP,IJ,I,J,I1,J1,I2,J2,IS,JS,PNT 27
1II,JJ,IM,INDX,INDX1,INDX2,L1,L2,S,SUM,SUM1,SUM2,SUMS,KIT,KKIT,KRETPNT 28
2,E1,E2,E3,E4,ITINN,ITCUT,ITPRNT,LAMB,AMB,NNN,LINK,IL,NCOL,NROW PNT 29
```

C

```
PNT 30
DO 4C IM=1,NM PNT 31
WRITE OUTPUT TAPE 6,902,IM PNT 32
902 FORMAT(16H0 MATERIAL NO. I2 ///) PNT 33
WRITE OUTPUT TAPE 6,903,NJ,(BETA1(IJ,IM),IJ=1,NJ) PNT 34
903 FORMAT( 14X 53HBETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 PNT 35
1 , I2 // ( 21X1PE14.7,1P3E20.7 ) ) PNT 36
WRITE OUTPUT TAPE 6,904,NJ,(XLAM1(IJ,IM),IJ=1,NJ) PNT 37
904 FORMAT( /// 14X 61HLAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR PNT 38
1GROU IJ = 1 , I2 // ( 21X1PE14.7,1P3E20.7 ) ) PNT 39
DO 4C IG=1,NG PNT 40
WRITE OUTPUT TAPE 6,906-IG,SABS1(IG,IM),SFI1(IG,IM),XNL(IG,IM), PNT 41
```

```

1D1(IG,IM),CHI1(IG,IM),V1(IG,IM),IG,NG,(STR1(IG,IGP,IM),IGP=1,NG) PNT 42
906 FORMAT( 16H0      GROUP NO. I2 // 9X 7HSIGA = 1PE14.7, PNT 43
1 10X 7HSIGF = 1PE14.7 , 10X 6HNU = 1PE14.7 // PNT 44
2 9X 7HDIFC = 1PE14.7, PNT 45
3 10X 7HCHI = 1PE14.7 , 10X 6HVEL = 1PE14.7 // PNT 46
4 14X15HTRANSFERS INTO I2, 2X17HFROM 1, 2, ..., I2, PNT 47
5 4H -- //((21X1PE14.7,1P3E20.7) ) PNT 48
WRITE OUTPUT TAPE 6,907,IG,NJ,( S1(IG,IJ,IM),IJ=1,NJ ) PNT 49
907 FORMAT( // 14X25HFISSION SPECTRUM IN GROUP I2, 36H FOR EACH PRECURPNT 50
1SOR GROUP IJ = 1 , I2 //((21X1PE14.7,1P3E20.7) ) PNT 51
40 CONTINUE PNT 52
RETURN PNT 53
END PNT 54

```

```

* FAP
*ICE SIMULTANEOUS DIFFERENTIAL EQUATION SUBROUTINE, MODIFIED AA ICE3 ICEC0000
COUNT 200 ICEC0010
LBL ICE,X ICEC0020
TTL WESTINGHOUSE LIBRARY FORTRAN VERSION 2 ICEC0030
ENTRY ICE ICEC0040
ENTRY XICE ICEC0050
BSS -2 ICEC0060
ICE SXA SAVE1,1 SAVINGS CF ICEC0070
CLA 1,4 CYCLES WORDS OVER 704 ICEC0080
STA P2 ICEC0090
CLA* P2 1 1 ICEC00100
STO AYCH ICEC00110
CLA 2,4 ICEC00120
STA T3 6 3 ICEC00130
CLA 3,4 ICEC00140
STA TP1 ICEC00150
CLA 4,4 ICEC00160
STA EPSI1 ICEC00170
CLA 5,4 ICEC00180
STA E2TST ICEC00190
CLA* 6,4 ICEC00200
PDX 0,1 ICEC00210
SXA EN,1 5 2 ICEC00220
CLA 7,4 ICEC00230
ADM T3+1 ICEC00240
STA Y1 ICEC00250
STA Y3 8 4 ICEC00260
CLA 8,4 ICEC00270
ADM T3+1 ICEC00280
STA DY3 4 2 ICEC00290
CLA 9,4 ICEC00300
ADM T3+1 ICEC00310
BETA3 STA IY2 2 1 ICEC00320
SBM EN ICEC00330

```

	STA	YBAR1			ICE00340
	SBM	EN			ICE00350
YHAL	STA	YFAL2	2	1	ICE00360
	SBM	EN			ICE00370
	STA	YY1	2	1	ICE00380
	SBM	EN			ICE00390
	STA	DYY1	2	1	ICE00400
	CLA	1C,4			ICE00410
	STA	J1			ICE00420
	CAL	TXL1			ICE00430
	STP	TXH1			ICE00440
	STP	TXH2			ICE00450
	TXI	SIX,4,-10			ICE00460
XICE	SXA	SAVE1,1			ICE00470
THRM1	LXA	EN,1			ICE00480
THREE	TRA	**	1		ICE00490
ALPH4	PXD	*,0	1		ICE00500
PSW	AXT	3,1	1		ICE00510
	ADM	AYCH			ICE00520
	TXH	PAW,1,1			ICE00530
	LDQ	AYCH,1			ICE00540
	TLQ	*+2			ICE00550
	CLA	AYCH,1			ICE00560
	TXI	*+1,1,1			ICE00570
TXL1	TXL	*-4,1,2			ICE00580
	SXA	PSW,1			ICE00590
PAW	STO	HBAR			ICE00600
P2	LDQ	**			ICE00610
	STQ	AYCH			ICE00620
	SBM	AYCH			ICE00630
	TPL	*+4	1		ICE00640
	CLA	HBAR			ICE00650
	LLS	0			ICE00660
	STO	AYCH			ICE00670
TP1	CLA	**			ICE00680
	STO	COMMON			ICE00690
	FSB*	T3			ICE00700
	TNZ	ZNZ			ICE00710
ZPRNT	CLA	COMMON			ICE00720
	XEC	T3	-1		ICE00730
SUB2	PXD	0,1	1		ICE00740
J1	STO	**			ICE00750
SAVE1	AXT	**,1	1		ICE00760
	TRA	1,4	1		ICE00770
ZNZ	STO	Z			ICE00780
	XCL		1		ICE00790
	STQ	HBBAR			ICE00800
	EMP	020000000000			ICE00810

	SBM	COMMON	2	1	ICE00820
	TMI	ZPRNT	1		ICE00830
	AXT	INTEG,1	1		ICE00840
	SXA	THREE,1			ICE00850
	CLA	HBBAR			ICE00860
	SBM	AYCH			ICE00870
	TPL	*+5	1		ICE00880
	AXT	1,1	1		ICE00890
	SXA	PSW,1			ICE00900
	CLA	Z			ICE00910
	STO	AYCH			ICE00920
BETA1	AXT	IY1,1	1		ICE00930
	LDQ	AYCH			ICE00940
	TRA	PASS+2	1		ICE00950
INTEG	CLA	Y3			ICE00960
	LDQ	YBAR1			ICE00970
	STA	LCOP+3			ICE00980
	STA	YBAR1			ICE00990
	STQ	Y3			ICE01000
	CLS	TXH1			ICE01010
	STO	TXH1			ICE01020
	CLS	TXH2			ICE01030
	STO	TXH2			ICE01040
LOOP	LDQ	DTHAL			ICE01050
DY3	FMP	** ,1	6		ICE01060
	STO	COMMON			ICE01070
	FAD	** ,1	1		ICE01080
Y3	STO	** ,1			ICE01090
TXH1	TXH	LOOP+8 ,0			ICE01100
	FAD	COMMON	1		ICE01110
YBAR1	STO	** ,1			ICE01120
	TIX	LCOP,1,1			ICE01130
TXH2	TXL	FIVE ,C			ICE01140
INEUL	CLA	DTIME			ICE01150
	TSX	SLB1,1			ICE01160
	TXI	SLB2,1,*			ICE01170
IY2	FSB	** ,1	1		ICE01180
	XCA		1		ICE01190
	FMP	=.3333333333			ICE01200
	STQ	COMMON+1			ICE01210
	FAD	COMMON	1		ICE01220
	STO*	Y1	-1		ICE01230
	SSM				ICE01240
	ADM	=.00001			ICE01250
	TPL	HhH	1		ICE01260
	CLA	COMMON+1			ICE01270
	FDP	COMMON	5		ICE01280
	CLA	N1			ICE01290

	LRS	0	
	TLQ	*+2	
	STQ	N1	
HHH	TIX	Y1,1,1	
	CLA	N1	
EPSI1	LDQ	**	
	TLQ	E2TST	
	CLA	AYCH	
	FAD	AYCH	1
	STO	AYCH	
SIX	CLA	ALPH1	
	AXT	1,1	1
STATH	STA	THREE	
	TRA	SUB2	1
E2TST	SBM	**	
	TMI	SIX	1
	AXT	YHAL2,1	1
	SXA	BETA,1	
	CLA	DTIME	
	STO	AYCH	
	TSX	SUB11,1	
FIVE	LXA	EN,1	
	STZ	N1	
Y1	CLA	**,1	
	STO	COMMON	
BETA	TRA	**	1
YHAL2	CLA	**,1	
IY1	STO*	IY2	-1
YY1	CLA	**,1	
	STO*	Y1	-1
DYY1	CLA	**,1	
	STO*	DY3	-1
	TIX	Y1,1,1	
TSX	TSX	SUB11,1	
	XCL		1
	FMP	=015540000000	
	SBM	DTHAL	2
	TMI	PASS	1
FAIL	CLA	BETA1	
	TXI	STOFI,1,TSX+4	
PASS	LDQ	DTHAL	
	AXT	YHAL1,1	1
	SXA	BETA,1	
	STQ	DTIME	
	FMP	=.5	
	STO	DTHAL	
	TRA	THRM1	1
YHAL1	STQ	HAL2	-1

ICE01300
ICE01310
ICE01320
ICE01330
ICE01340
ICE01350
ICE01360
ICE01370
ICE01380
ICE01390
ICE01400
ICE01410
ICE01420
ICE01430
ICE01440
ICE01450
ICE01460
ICE01470
ICE01480
ICE01490
ICE01500
ICE01510
ICE01520
ICE01530
ICE01540
ICE01550
ICE01560
ICE01570
ICE01580
ICE01590
ICE01600
ICE01610
ICE01620
ICE01630
ICE01640
ICE01650
ICE01660
ICE01670
ICE01680
ICE01690
ICE01700
ICE01710
ICE01720
ICE01730
ICE01740
ICE01750
ICE01760
ICE01770

```

TIX Y1,1,1
CLA BETA3
STOFI STA BETA
TRA SUB2 1
Y6 CLA* Y1 -1
STO* YY1 -1
CLA* DY3 -1
STG* DYY1 -1
ALPH1 TIX Y6,1,1
CLA ALPH4
TXI STATH,1,1
SUB11 CLS DTIME
SUB1 FAC* T3
T3 STO **
TRA 1,1 1
EN DEC 0
HBBAR DEC 0
HBAR DEC 0
AYCH DEC 0
DTIME DEC 0
DTHAL DEC 0
COMMON -206
N1 COMMON 1
Z COMMON 1
COMMON COMMON 1
END
* DATA

```

```

ICE01780
ICE01790
ICE01800
ICE01810
ICE01820
ICE01830
ICE01840
ICE01850
ICE01860
ICE01870
ICE01880
ICE01890
ICE01900
ICE01910
ICE01920
ICE01930
ICE01940
ICE01950
ICE01960
ICE01970
ICE01980
ICE01990
ICE02000
ICE02010
ICE02020
ICE02030

```

TEST PROB. STEADY STATE AND TIME SOLUTIONS. ABSORBER RING INSERTED AFTER 2 SEC.

1	4	6	3	1	7	1	-1	0	0	1	11	10	5	0
130.0														
5.0		15.0			10.0			10.0		6.0		1.0		7.0
1														
1														
1														
1														
2														
3														
2														
0.0														
0.0														
1.			1.			1.			1.				1.	
1.														
0.0														
0.0														
0.0														
0.0														
.000203		3.87			.001			1.4		.003175				
.311		.001466			.115			.00166		.0317				

.000296	.0127					
0.97447925E-03	0.79013389E-03	0.26468368E 01	0.75356218E 00	0.26827937E 01	100	
0.	0.	0.	C.		101	
0.15963384E-02	0.11891829E-02	0.23311510E 01	C.24643214E-00	0.12395053E 01	102	
0.18970602E-01	0.	0.	C.		103	
0.14424144E-01	0.85840425E-02	0.23173642E 01	C.	0.10389322E 01	104	
0.70583004E-06	0.75373829E-02	0.	C.		105	
0.46856542E-01	0.38559482E-01	0.23134940E 01	C.	0.97209518E 00	106	
0.46313683E-10	0.12080043E-07	0.97972934E-03	0.30161881E-00		107	
3.22	E 09 8.9	E 08 7.28	E 07 1.33	E 06		

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

	3.87		1.4		
.311		.115		.0317	
	.0127				
-0.68736250E-02	0.	0.	0.75356218E 00	0.15577680E 01	700
0.	0.	0.	C.		701
0.30113333E-05	0.	0.	0.24643214E-00	0.61318059E 00	702
0.49542869E-01	0.	0.	0.		703
0.32825833E-04	0.	0.	0.	0.54000440E 00	704
0.	0.27924262E-01	0.	0.		705
0.44762779E-03	0.	-0.	C.	0.52912294E 00	706
0.	0.	0.13289794E-01	0.64573503E 00		707
3.22	E 09 8.9	E 08 7.28	E 07 1.33	E 06	

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

	3.87		1.4		
.311		.115		.0317	
	.0127				
-0.68736250E-02	0.	0.	0.75356218E 00	0.15577680E 01	700
0.	0.	0.	C.		701
0.30113333E-05	0.	0.	0.24643214E-00	0.61318059E 00	702
0.49542869E-01	0.	0.	C.		703
0.32825833E-04	0.	0.	0.	0.54000440E 00	704
0.	0.27924262E-01	0.	C.		705
0.44762779E-03	0.	-0.	C.	0.52912294E 00	706

- 97 -

707

- 97 -

* XEQ
* CHAIN (1,4)
* CHAIN (2,4)
* DATA

BEGIN LOAD TIME 15.823

ENTRY POINTS TO SUBROUTINES REQUESTED FROM LIBRARY,				CHAIN 1 , 4			
(FPT)	CLOCK	(TSHM)	(RTN)	(STHM)	(FIL)	(SPH)	CHAIN
SUBROUTINE ENTRY POINTS AND STARTING LOCATIONS				UPPER PROGRAM 22421		LOWER COMMON 34167	
MAIN-P	00155 00144	PRINT	14257 14251	ANYER	15530 15057	SBRER	15526
OVFLW	15521	UNFLW	15524	BRCDE	15506	EPMD	15477
(SVN)	15064	(SIX)	15063	(FPT)	15065	ERROR	15214
ERTRP	15540	ERSIN	15532	(TSHM)	15570 15560	(TSH)	15570
CHAIN	15636 15631	(BLOK)	16537 16120	(STHD)	16134	(STHM)	16131
(STH)	16131	(SPH)	16615 16611	(WTC)	17120 17047	(WER)	17061
(RDC)	17177 17146	(RER)	17155	(RTN)	20044 17213	(FIL)	20033
(IOH)	20174	CLOCK	21123 21120	(TCO)	21373 21264	(TEF)	21372
(RCH)	21371	(ETT)	21370	(REW)	21367	(WEF)	21366
(BSR)	21365	(WRS)	21364	(RDS)	21363	(IOS)	21267
STCH)	21405	(TCN)	21404	(LCH)	21403	(CHR)	21402
(RUN)	21401	(BSF)	21400	(SDL)	21377	(SDH)	21376
(BTT)	21375	(TRC)	21374	(IOU)	21431 21426	(FT)	21455 21447
(EXEM)	21456	(EXE)	21456	EXIT	22406 22404	(TES)	22420 22420

ENTRY POINTS TO SUBROUTINES REQUESTED FROM LIBRARY,				CHAIN 2 , 4			
(FPT)	(TSPM)	(RTN)	(STHM)	(FIL)	ICE	XICE	CLOCK CHAIN
SUBROUTINE ENTRY POINTS AND STARTING LOCATIONS				UPPER PROGRAM 23757		LOWER COMMON 34167	
MAIN-P	00162 00144	FLXOUT	10340 10332	FDSTRT	15453 15447	FEEDBK	15476 15472
FOPRNT	15521 15515	PRINT	15546 15540	XICE	16416 16346	ICE	16346
ANYER	17324 16653	SBRER	17322	CVFLW	17315	UNFLW	17320
BRCDE	17302	EPMD	17273	(SVN)	16660	(SIX)	16657
(FPT)	16661	ERROR	17010	ERTRP	17334	ERSIN	17326
(TSHM)	17364 17354	(TSH)	17364	CHAIN	17432 17425	(BLOK)	20333 17714
(STHD)	17730	(STHM)	17725	(STH)	17725	(WTC)	20456 20405
(WER)	20417	(RDC)	20535 20504	(RER)	20513	(RTN)	21402 20551
(FIL)	21371	(IOH)	21532	CLOCK	22461 22456	(TCO)	22731 22622
(TEF)	22730	(RCH)	22727	(ETT)	22726	(REW)	22725
(WEF)	22724	(BSR)	22723	(WRS)	22722	(RDS)	22721
(IOS)	22625	STCH)	22743	(TCN)	22742	(LCH)	22741
(---	2240	IN)	273	BSI	22	(S	2 5

(SDH) 22734
(FT) 23013 23005
(TES) 23756 23756

(BTT) 22733
(EXEM) 23014

(TRC) 22732
(EXE) 23014

(IOU) 22767 22764
EXIT 23744 23742

-99-

EXECUTION TIME 15.836

PROBLEM NO. 1

COMPUTER TIME = 15.836

11/30/64

-100-

TEST PROB. STEADY STATE AND TIME SOLUTIONS. ABSORBER RING INSERTED AFTER 2 SEC.

GROUPS= 4

PRECURSORS= 6

X REGIONS= 1

Y(OR R) REGIONS= 7

MATERIALS= 3

THIS PROBLEM HAS CYLINDRICAL COORDINATES X,R GEOMETRY.
THE SYMBOL Y REPRESENTS R THROUGHOUT.
INITIAL FLUXES ARE COMPUTED FROM SCRATCH.

DELTA X NO. 1 = 1.3000000E 02

DELTA R NO. 1 = 5.0000000E 00

DELTA R NO. 2 = 1.5000000E 01

DELTA R NO. 3 = 1.0000000E 01

DELTA R NO. 4 = 1.0000000E 01

DELTA R NO. 5 = 6.0000000E 00

DELTA R NO. 6 = 1.0000000E 00

DELTA R NO. 7 = 7.0000000E 00

MATERIAL NO. M(1, 1) = 1

MATERIAL NO. M(1, 2) = 1

MATERIAL NO. M(1, 3) = 1

MATERIAL NO. M(1, 4) = 1

MATERIAL NO. M(1, 5) = 2

MATERIAL NO. M(1, 6) = 3

MATERIAL NO. M(1, 7) = 2

LAYOUT---

T = TIME

FLUX GROUP 1

Y = 0.

X = 0.

6.5000000E 01 1.3000000E 02

Y = 2.5000000E 00

X = 0.

ETC.

Y = 5.0000000E 00

Y = 5.0000000E 00

Y = 1.2500000E 01

Y = 2.0000000E 01

Y = 2.0000000E 01

Y = 2.5000000E 01

Y = 3.0000000E 01

Y = 3.0000000E 01

Y = 3.5000000E 01

Y = 4.0000000E 01

Y = 4.0000000E 01

Y = 4.0000000E 01

Y = 4.0000000E 01

Y = 4.600000E 01
Y = 4.650000E 01
Y = 4.700000E 01
Y = 4.700000E 01
Y = 5.050000E 01
Y = 5.400000E 01
FLUX GROUP 2
ETC.
ETC.

MATERIAL NO. 1

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

2.0300000E-04	1.0000000E-03	3.1750000E-03	1.4660000E-03
1.6600000E-03	2.9600000E-04		

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00	1.4000000E 00	3.1100000E-01	1.1500000E-01
3.1700000E-02	1.2700000E-02		

GROUP NO. 1

SIGA = 9.7447925E-04	SIGF = 7.9013388E-04	NU = 2.6468368E 00
DIFC = 2.6827937E 00	CHI = 7.5356217E-01	VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ... , 4 --

0.	0.	0.	0.
----	----	----	----

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.	-0.	-0.	-0.
-0.	0.		

GROUP NO. 2

SIGA = 1.5963384E-03	SIGF = 1.1891829E-03	NU = 2.3311510E 00
----------------------	----------------------	--------------------

DIFC = 1.2395053E 00

CHI = 2.4643214E-01

VEL = 8.9000000E 08

-102-

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

1.8970602E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

1.0000000E 00

1.0000000E 00

1.0000000E 00

1.0000000E 00

1.0000000E 00

1.0000000E 00

GROUP NO. 3

SIGA = 1.4424144E-02

SIGF = 8.5840423E-03

NU = 2.3173642E 00

DIFC = 1.0389322E 00

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

7.0583003E-07

7.5373828E-03

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.6856542E-02

SIGF = 3.8559482E-02

NU = 2.3134940E 00

DIFC = 9.7209517E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

4.6313683E-11

1.2080043E-08

9.7972933E-04

3.0161881E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 2

A 1

FR

R 1

I P

JR

GR

=

6

-0.
-0.

-0.
-0.

-0.

-0.

-103-

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00
3.1700000E-02

1.4000000E 00
1.2700000E-02

3.1100000E-01

1.1500000E-01

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ... , 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ... , 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

-104-

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 3

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1, 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1, 6

3.8700000E 00

1.4000000E 00

3.1100000E-01

1.1500000E-01

3.1700000E-02

1.2700000E-02

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

-105-

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.33CC000E 06

-106-

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

ITOUT= 11

ITINN= 10

ITPRNT= 5

ISSW= 0

ENTRY TO THE STEADY STATE.

COMPUTER TIME = 15.837

EIGENVALUE = 1.0168806E 00

FLUX GROUP NO. 1

0.	4.1612725E 05	0.
0.	4.1564614E 05	0.
0.	4.1420275E 05	0.
0.	4.0318189E 05	0.
0.	3.8084065E 05	0.
0.	3.5976879E 05	0.
0.	3.3091359E 05	0.
0.	2.8894584E 05	0.
0.	2.1107895E 05	0.
0.	1.2388358E 05	0.
0.	7.2952800E 04	0.
0.	6.6261433E 04	0.
0.	5.9721179E 04	0.
0.	2.4512140E 04	0.
0.	0.	0.

FLUX GROUP NO. 2

0.	1.0000000E 06	0.
0.	9.9858255E 05	0.
0.	9.9433017E 05	0.
0.	9.6167580E 05	0.
0.	8.7775877E 05	0.

0.	8.3116869E 05	0.
0.	7.4859525E 05	0.
0.	6.4216704E 05	0.
0.	4.9763218E 05	0.
0.	3.2245186E 05	0.
0.	2.0037758E 05	0.
0.	1.8343952E 05	0.
0.	1.6698844E 05	0.
0.	7.2315716E 04	0.
0.	0.	0.

FLUX GROUP NO. 3

0.	4.4061086E 05	0.
0.	4.4006844E 05	0.
0.	4.3844115E 05	0.
0.	4.2565080E 05	0.
0.	4.0235509E 05	0.
0.	3.8399953E 05	0.
0.	3.6765992E 05	0.
0.	3.6014875E 05	0.
0.	3.7705556E 05	0.
0.	3.6154765E 05	0.
0.	2.7883259E 05	0.
0.	2.6252130E 05	0.
0.	2.4600930E 05	0.
0.	1.2319526E 05	0.
0.	0.	0.

FLUX GROUP NO. 4

0.	9.1119990E 03	0.
0.	9.1181964E 03	0.
0.	9.1367875E 03	0.
0.	9.0881478E 03	0.
0.	1.0233417E 04	0.
0.	1.3014696E 04	0.
0.	2.1924030E 04	0.
0.	4.5086110E 04	0.
0.	1.1505700E 05	0.
0.	1.9474850E 05	0.
0.	1.9471686E 05	0.
0.	1.8918166E 05	0.
0.	1.8274915E 05	0.
0.	1.0599577E 05	0.
0.	0.	0.

ADJOINT GROUP NO. 1

0.	5.1927963E 05	0.
0.	5.1876903E 05	0.

0.	5.1723723E 05	0.
0.	5.0382821E 05	0.
0.	4.7537981E 05	0.
0.	4.4962879E 05	0.
0.	4.1860402E 05	0.
0.	3.8301281E 05	0.
0.	3.4342946E 05	0.
0.	2.9054116E 05	0.
0.	2.2234511E 05	0.
0.	2.0947092E 05	0.
0.	1.9603067E 05	0.
0.	9.8123059E 04	0.
0.	0.	0.

ADJOINT GROUP NO. 2

0.	6.0733708E 05	0.
0.	6.0669075E 05	0.
0.	6.0475175E 05	0.
0.	5.8875421E 05	0.
0.	5.5549572E 05	0.
0.	5.2486840E 05	0.
0.	4.8740439E 05	0.
0.	4.4361917E 05	0.
0.	3.9390207E 05	0.
0.	3.2263835E 05	0.
0.	2.3998345E 05	0.
0.	2.2533069E 05	0.
0.	2.1041839E 05	0.
0.	1.0433130E 05	0.
0.	0.	0.

ADJOINT GROUP NO. 3

0.	6.9459019E 05	0.
0.	6.9389413E 05	0.
0.	6.9180596E 05	0.
0.	6.7500818E 05	0.
0.	6.3961661E 05	0.
0.	6.0624503E 05	0.
0.	5.6449890E 05	0.
0.	5.1410621E 05	0.
0.	4.5402264E 05	0.
0.	3.6784545E 05	0.
0.	2.6916680E 05	0.
0.	2.5217106E 05	0.
0.	2.3507182E 05	0.
0.	1.1553174E 05	0.
0.	0.	0.

ADJOINT GROUP NO. 4

-109-

0.	1.0000000E 06	C.
0.	9.9919090E 05	C.
0.	9.9676356E 05	C.
0.	9.7552915E 05	C.
0.	9.2719506E 05	C.
0.	8.8057740E 05	C.
0.	8.1951712E 05	C.
0.	7.3937289E 05	C.
0.	6.1928064E 05	C.
0.	4.6108251E 05	C.
0.	3.1940174E 05	C.
0.	2.9723758E 05	C.
0.	2.7548393E 05	C.
0.	1.3181435E 05	C.
0.	0.	C.

EIGENVALUE = 1.0165686E 00

FLUX GROUP NO. 1

0.	4.1595930E 05	C.
0.	4.1532205E 05	C.
0.	4.1341029E 05	C.
0.	4.0025537E 05	C.
0.	3.7613449E 05	C.
0.	3.5374714E 05	C.
0.	3.2387488E 05	C.
0.	2.8137102E 05	C.
0.	2.0401939E 05	C.
0.	1.1738261E 05	C.
0.	6.5764234E 04	C.
0.	5.9354652E 04	C.
0.	5.3421522E 04	C.
0.	2.1926490E 04	C.
0.	0.	C.

FLUX GROUP NO. 2

0.	1.0000000E 06	C.
0.	9.9821802E 05	C.
0.	9.9287208E 05	C.
0.	9.5555764E 05	C.
0.	8.8527924E 05	C.
0.	8.1906385E 05	C.
0.	7.3444917E 05	C.
0.	6.2670666E 05	C.

0.	4.8239462E 05	0.
0.	3.0805683E 05	0.
0.	1.8606613E 05	0.
0.	1.6969377E 05	0.
0.	1.5423458E 05	0.
0.	6.6496849E 04	0.
0.	0.	0.

FLUX GROUP NO. 3

0.	4.4100676E 05	0.
0.	4.4030161E 05	0.
0.	4.3818614E 05	0.
0.	4.2338246E 05	0.
0.	3.9872429E 05	0.
0.	3.7910743E 05	0.
0.	3.6145771E 05	0.
0.	3.5229846E 05	0.
0.	3.6712098E 05	0.
0.	3.5068541E 05	0.
0.	2.7087774E 05	0.
0.	2.5490090E 05	0.
0.	2.3848857E 05	0.
0.	1.1841490E 05	0.
0.	0.	0.

FLUX GROUP NO. 4

0.	9.1294978E 03	0.
0.	9.1329092E 03	0.
0.	9.1431432E 03	0.
0.	9.0576979E 03	0.
0.	1.0216140E 04	0.
0.	1.3031890E 04	0.
0.	2.2071416E 04	0.
0.	4.5493890E 04	0.
0.	1.1617098E 05	0.
0.	1.9769374E 05	0.
0.	2.0108523E 05	0.
0.	1.9548768E 05	0.
0.	1.8847606E 05	0.
0.	1.0804486E 05	0.
0.	0.	0.

ADJOINT GROUP NO. 1

0.	5.1296420E 05	0.
0.	5.1213248E 05	0.
0.	5.0963733E 05	0.
0.	4.9219348E 05	0.
0	.6(77 ;	(

C.	4.3264875E 05	C.
0.	3.9964021E 05	C.
0.	3.6245908E 05	C.
0.	3.2207761E 05	C.
0.	2.6958335E 05	C.
0.	2.0300754E 05	C.
0.	1.9097728E 05	C.
0.	1.7874805E 05	C.
0.	8.9714285E 04	C.
0.	0.	C.

ADJOINT GRUP NC. 2

0.	5.9873326E 05	C.
0.	5.9775068E 05	C.
0.	5.9480293E 05	C.
0.	5.7420390E 05	C.
0.	5.3692120E 05	C.
0.	5.0347271E 05	C.
0.	4.6391613E 05	C.
0.	4.1899109E 05	C.
0.	3.6967661E 05	C.
0.	3.0092387E 05	C.
0.	2.2232494E 05	C.
0.	2.0861842E 05	C.
0.	1.9479884E 05	C.
0.	9.6624705E 04	C.
0.	0.	C.

ADJOINT GRUP NO. 3

0.	6.8859904E 05	C.
0.	6.8747419E 05	C.
0.	6.8409962E 05	C.
0.	6.6053931E 05	C.
0.	6.1782148E 05	C.
0.	5.7938058E 05	C.
0.	5.3366274E 05	C.
0.	4.8121870E 05	C.
0.	4.2260851E 05	C.
0.	3.4170622E 05	C.
0.	2.4960916E 05	C.
0.	2.3381401E 05	C.
0.	2.1796009E 05	C.
0.	1.0714173E 05	C.
0.	0.	C.

ADJOINT GRUP NC. 4

0.	1.0000000E 06	C.
0.	9.9837230E 05	C.

0.	9.9348919E 05	0.
0.	9.5945295E 05	0.
0.	8.9736432E 05	0.
0.	8.4097968E 05	0.
0.	7.7176832E 05	0.
0.	6.8704079E 05	0.
0.	5.7187877E 05	0.
0.	4.2654060E 05	0.
0.	2.9631812E 05	0.
0.	2.7584078E 05	0.
0.	2.5567076E 05	0.
0.	1.2233409E 05	0.
0.	0.	0.

COMPUTER TIME = 15.851

T = 0.

,NUCLEAR CONSTANTS

113-

MATERIAL NO. 1

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

2.0300000E-04	1.0000000E-03	3.1750000E-03	1.4660000E-03
1.6600000E-03	2.9600000E-04		

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00	1.4000000E 00	3.1100000E-01	1.1500000E-01
3.1700000E-02	1.2700000E-02		

GROUP NO. 1

SIGA = 9.7447925E-04	SIGF = 7.9013388E-04	NU = 2.6906913E 00
DIFC = 2.6827937E 00	CHI = 7.5356217E-01	VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.	0.	0.	0.
----	----	----	----

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.	-0.	-0.	-0.
-0.	0.		

GROUP NO. 2

SIGA = 1.5963384E-03	SIGF = 1.1891829E-03	NU = 2.3697750E 00
DIFC = 1.2395053E 00	CHI = 2.4643214E-01	VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

1.8970602E-02	0.	0.	0.
---------------	----	----	----

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

1.0000000E 00	1.0000000E 00	1.0000000E 00	1.0000000E 00
---------------	---------------	---------------	---------------

1.0000000E 00

1.0000000E 00

-114-

GROUP NO. 3

SIGA = 1.4424144E-02

SIGF = 8.5840423E-03

NU = 2.3557598E 00

DIFC = 1.0389322E 00

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

7.0583003E-07

7.5373828E-03

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.6856542E-02

SIGF = 3.8559482E-02

NU = 2.3518255E 00

DIFC = 9.7209517E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

4.6313683E-11

1.2080043E-08

9.7972933E-04

3.0161881E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 2

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1, 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1, 6

3.0700000E 00

1.4000000E 00

3.1100000E-01

1.15 00E-01

3.1700000E-02

1.2700000E-02

-115-

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

C.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-C.

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

C.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-C.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

C.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ... , 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 3

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00

1.4000000E 00

3.1100000E-01

1.1500000E+01

3.1700000E-02

1.2700000E-02

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ... , 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0

-0.

-0.

0.

-117-

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

-118-

ERROR LIMITS FROM NOW ON ARE 5.9999999E-05 6.9999999E-03 5.9999999E-05 5.9999999E-05

ENTRY TO SUBROUTINE FEEDBK (INITIALLY IN THIS DT).

T = 0.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

14 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX C.

1 OUT-OF-ICE ITERATIONS

T = 0.

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE POWER DENSITY = 2.0485175E 03)

0.	5.6391543E 03	0.
-0.	5.6306093E 03	-0.
0.	5.6049740E 03	0.
0.	5.6049740E 03	0.
-0.	5.4200849E 03	-0.
0.	5.1512244E 03	0.
0.	5.1512244E 03	0.
-0.	4.9954900E 03	-0.
0.	5.0685338E 03	0.
0.	5.0685338E 03	0.
-0.	5.7305572E 03	-0.
0.	8.3463835E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	4.1470272E 05	0.
-0.	4.1406549E 05	-0.
-	4.1215377E 05	-

0.	4.1215377E 05	0.
-0.	3.9901850E 05	-0.
0.	3.7497299E 05	0.
0.	3.7497299E 05	0.
-0.	3.5266154E 05	-0.
0.	3.2289709E 05	0.
0.	3.2289709E 05	0.
-0.	2.8054393E 05	-0.
0.	2.0342112E 05	0.
0.	2.0342112E 05	0.
-0.	1.1700627E 05	-0.
0.	6.5506675E 04	0.
0.	6.5506675E 04	0.
-0.	5.9116676E 04	-0.
0.	5.3206237E 04	0.
0.	5.3206237E 04	0.
-0.	2.1838128E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	9.9708668E 05	0.
-0.	9.9530537E 05	-0.
0.	9.8996146E 05	0.
0.	9.8996146E 05	0.
-0.	9.5271007E 05	-0.
0.	8.8263627E 05	0.
0.	8.8263627E 05	0.
-0.	8.1661527E 05	-0.
0.	7.3225556E 05	0.
0.	7.3225556E 05	0.
-0.	6.2483206E 05	-0.
0.	4.8091877E 05	0.
0.	4.8091877E 05	0.
-0.	3.0702060E 05	-0.
0.	1.8530439E 05	0.
0.	1.8530439E 05	0.
-0.	1.6898387E 05	-0.
0.	1.5358688E 05	0.
0.	1.5358688E 05	0.
-0.	6.6219137E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	4.3973737E 05	0.
-0.	4.3903208E 05	-0.
0.	4.3691618E 05	0.
0.	4.3691618E 05	0.
-0.	4.2213032E 05	-0.

0.	3.9752630E 05	0.
0.	3.9752630E 05	0.
-0.	3.7794492E 05	-0.
0.	3.6031372E 05	0.
0.	3.6031372E 05	0.
-0.	3.5112938E 05	-0.
0.	3.6582507E 05	0.
0.	3.6582507E 05	0.
-0.	3.4932493E 05	-0.
0.	2.6965203E 05	0.
0.	2.6965203E 05	0.
-0.	2.5373088E 05	-0.
0.	2.3739302E 05	0.
0.	2.3739302E 05	0.
-0.	1.1787879E 05	-0.
0.	-0.	0.

FLUX GROUP NO. 4

0.	9.1038846E 03	0.
-0.	9.1072825E 03	-0.
0.	9.1174761E 03	0.
0.	9.1174761E 03	0.
-0.	9.0321709E 03	-0.
0.	1.0190673E 04	0.
0.	1.0190673E 04	0.
-0.	1.3004328E 04	-0.
0.	2.2035254E 04	0.
0.	2.2035254E 04	0.
-0.	4.5429577E 04	-0.
0.	1.1601544E 05	0.
0.	1.1601544E 05	0.
-0.	1.9749540E 05	-0.
0.	2.0109579E 05	0.
0.	2.0109579E 05	0.
-0.	1.9550353E 05	-0.
0.	1.8846645E 05	0.
0.	1.8846645E 05	0.
-0.	1.0795917E 05	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	7.0543817E-01	0.
-0.	7.0436976E-01	-0.
0.	7.0116450E-01	0.
0.	7.0116450E-01	0.
-0.	6.7804661E-01	-0.
0.	6.4431587E-01	0.
0.	6.4431587E-01	0.

-0.	6.2461649E-01	-0.
0.	6.3308068E-01	0.
0.	6.3308068E-01	0.
-0.	7.1418969E-01	-0.
0.	1.0364056E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	9.6060722E 00	0.
-0.	9.5915235E 00	-0.
0.	9.5478771E 00	0.
0.	9.5478771E 00	0.
-0.	9.2330768E 00	-0.
0.	8.7737594E 00	0.
0.	8.7737594E 00	0.
-0.	8.5055097E 00	-0.
0.	8.6207680E 00	0.
0.	8.6207680E 00	0.
-0.	9.7252432E 00	-0.
0.	1.4112912E 01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.3729579E 02	0.
-0.	1.3708785E 02	-0.
0.	1.3646403E 02	0.
0.	1.3646403E 02	0.
-0.	1.3196471E 02	-0.
0.	1.2539987E 02	0.
0.	1.2539987E 02	0.
-0.	1.2156589E 02	-0.
0.	1.2321323E 02	0.

0.	1.2321323E 02	0.
-0.	1.3899906E 02	-0.
0.	2.0171027E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.7143915E 02	0.
-0.	1.7117950E 02	-0.
0.	1.7040055E 02	0.
0.	1.7040055E 02	0.
-0.	1.6478232E 02	-0.
0.	1.5658490E 02	0.
0.	1.5658490E 02	0.
-0.	1.5179746E 02	-0.
0.	1.5385447E 02	0.
0.	1.5385447E 02	0.
-0.	1.7356599E 02	-0.
0.	2.5187253E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 5

0.	7.0424327E 02	0.
-0.	7.0317667E 02	-0.
0.	6.9997685E 02	0.
0.	6.9997685E 02	0.
-0.	6.7689812E 02	-0.
0.	6.4322450E 02	0.
0.	6.4322450E 02	0.
-0.	6.2355850E 02	-0.
0.	6.3200835E 02	0.
0.	6.3200835E 02	0.
-0.	7.1297998E 02	-0.

0.	1.0346501E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 6

0.	3.1344538E 02	0.
-0.	3.1297066E 02	-0.
0.	3.1154647E 02	0.
0.	3.1154647E 02	0.
-0.	3.0127456E 02	-0.
0.	2.8628708E 02	0.
0.	2.8628708E 02	0.
-0.	2.7753411E 02	-0.
0.	2.8129498E 02	0.
0.	2.8129498E 02	0.
-0.	3.1733392E 02	-0.
0.	4.6050320E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

T = 1.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 5.0000000E-01

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 5.0000000E-01

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 1.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 1.0000000E 00

-124-

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX C.

1 OUT-OF-ICE ITERATIONS

T = 1.0000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE POWER DENSITY = 2.0495409E 03)

0.	5.6418536E 03	0.
-0.	5.6333047E 03	-0.
0.	5.6076577E 03	0.
0.	5.6076577E 03	0.
-0.	5.4226829E 03	-0.
0.	5.1537032E 03	0.
0.	5.1537032E 03	0.
-0.	4.9979089E 03	-0.
0.	5.0710337E 03	0.
0.	5.0710337E 03	0.
-0.	5.7334876E 03	-0.
0.	8.3509231E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	4.1490544E 05	0.
-0.	4.1426791E 05	-0.
0.	4.1235529E 05	0.
0.	4.1235529E 05	0.
-0.	3.9921409E 05	-0.
0.	3.7515835E 05	0.
0.	3.7515835E 05	0.

-0.	3.5283718E 05	-0.
0.	3.2305949E 05	0.
0.	3.2305949E 05	0.
-0.	2.8068649E 05	-0.
0.	2.0352475E 05	0.
0.	2.0352475E 05	0.
-0.	1.1706446E 05	-0.
0.	6.5537173E 04	0.
0.	6.5537173E 04	0.
-0.	5.9143952E 04	-0.
0.	5.3230736E 04	0.
0.	5.3230736E 04	0.
-0.	2.1848184E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	9.9756310E 05	0.
-0.	9.9578103E 05	-0.
0.	9.9043479E 05	0.
0.	9.9043479E 05	0.
-0.	9.5316694E 05	-0.
0.	8.8306154E 05	0.
0.	8.8306154E 05	0.
-0.	8.1701026E 05	-0.
0.	7.3261114E 05	0.
0.	7.3261114E 05	0.
-0.	6.2513635E 05	-0.
0.	4.8115227E 05	0.
0.	4.8115227E 05	0.
-0.	3.0716581E 05	-0.
0.	1.8538512E 05	0.
0.	1.8538512E 05	0.
-0.	1.6905674E 05	-0.
0.	1.5365305E 05	0.
0.	1.5365305E 05	0.
-0.	6.6247935E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	4.3994740E 05	0.
-0.	4.3924177E 05	-0.
0.	4.3712485E 05	0.
0.	4.3712485E 05	0.
-0.	4.2233186E 05	-0.
0.	3.9771532E 05	0.
0.	3.9771532E 05	0.
-0.	3.7812342E 05	-0.
0.	3.6048157E 05	0.

0.	3.6048157E 05	C.
-0.	3.5128925E 05	-C.
0.	3.6598588E 05	0.
0.	3.6598588E 05	C.
-0.	3.4946848E 05	-C.
0.	2.6974637E 05	C.
0.	2.6974637E 05	C.
-0.	2.5381826E 05	-C.
0.	2.3747506E 05	C.
0.	2.3747506E 05	0.
-0.	1.1792120E 05	-C.
0.	-0.	C.

FLUX GROUP NO. 4

0.	9.1082847E 03	C.
-0.	9.1116877E 03	-C.
0.	9.1218964E 03	C.
0.	9.1218964E 03	C.
-0.	9.0365846E 03	-0.
0.	1.0195966E 04	0.
0.	1.0195966E 04	C.
-0.	1.3011540E 04	-0.
0.	2.2048425E 04	C.
0.	2.2048425E 04	0.
-0.	4.5457677E 04	-C.
0.	1.1608805E 05	0.
0.	1.1608805E 05	C.
-0.	1.9762469E 05	-C.
0.	2.0124544E 05	C.
0.	2.0124544E 05	0.
-0.	1.9564948E 05	-0.
0.	1.8860504E 05	0.
0.	1.8860504E 05	0.
-0.	1.0803183E 05	-C.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	7.0280512E-01	0.
-0.	7.0173931E-01	-0.
0.	6.9854180E-01	0.
0.	6.9854180E-01	0.
-0.	6.7549622E-01	-C.
0.	6.4189909E-01	0.
0.	6.4189909E-01	0.
-0.	6.2228519E-01	-0.
0.	6.3075811E-01	0.
0.	6.3075811E-01	0.
-0.	7 1145064E	-0

0.	1.0329670E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	9.5875238E 00	0.
-0.	9.5729920E 00	-0.
0.	9.5293964E 00	0.
0.	9.5293964E 00	0.
-0.	9.2150939E 00	-0.
0.	8.7567276E 00	0.
0.	8.7567276E 00	0.
-0.	8.4890947E 00	-0.
0.	8.6044637E 00	0.
0.	8.6044637E 00	0.
-0.	9.7075945E 00	-0.
0.	1.4089227E 01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.3720003E 02	0.
-0.	1.3699218E 02	-0.
0.	1.3636860E 02	0.
0.	1.3636860E 02	0.
-0.	1.3187182E 02	-0.
0.	1.2531194E 02	0.
0.	1.2531194E 02	0.
-0.	1.2148121E 02	-0.
0.	1.2312938E 02	0.
0.	1.2312938E 02	0.
-0.	1.3890887E 02	-0.
0.	2.0159080E 02	0.
0.	-0.	0.

-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.7139494E 02	0.
-0.	1.7113533E 02	-0.
0.	1.7035649E 02	0.
0.	1.7035649E 02	0.
-0.	1.6473942E 02	-0.
0.	1.5654430E 02	0.
0.	1.5654430E 02	0.
-0.	1.5175837E 02	-0.
0.	1.5381575E 02	0.
0.	1.5381575E 02	0.
-0.	1.7352435E 02	-0.
0.	2.5181736E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 5

0.	7.0419320E 02	0.
-0.	7.0312665E 02	-0.
0.	6.9992695E 02	0.
0.	6.9992695E 02	0.
-0.	6.7684954E 02	-0.
0.	6.4317852E 02	0.
0.	6.4317852E 02	0.
-0.	6.2351422E 02	-0.
0.	6.3196450E 02	0.
0.	6.3196450E 02	0.
-0.	7.1293282E 02	-0.
0.	1.0345876E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO.	6	
0.	3.1343645E 02	C.
-0.	3.1296173E 02	-C.
0.	3.1153757E 02	C.
0.	3.1153757E 02	C.
-0.	3.0126590E 02	-C.
0.	2.8627888E 02	C.
0.	2.8627888E 02	C.
-0.	2.7752621E 02	-C.
0.	2.8128716E 02	C.
0.	2.8128716E 02	C.
-0.	3.1732551E 02	-C.
0.	4.6049207E 02	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	0.

T = 2.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 1.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 1.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 2.0000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE PCWER DENSITY = 2.0509024E 03)

0.	5.6455309E 03	0.
-0.	5.6369765E 03	-0.
0.	5.6113135E 03	0.
0.	5.6113135E 03	0.
-0.	5.4262207E 03	-0.
0.	5.1570712E 03	0.
0.	5.1570712E 03	0.
-0.	5.0011845E 03	-0.
0.	5.0743842E 03	0.
0.	5.0743842E 03	0.
-0.	5.7373366E 03	-0.
0.	8.3566859E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	4.1517814E 05	0.
-0.	4.1454019E 05	-0.
0.	4.1262634E 05	0.
0.	4.1262634E 05	0.
-0.	3.9947682E 05	-0.
0.	3.7540619E 05	0.
0.	3.7540619E 05	0.
-0.	3.5307110E 05	-0.
0.	3.2327468E 05	0.
0.	3.2327468E 05	0.
-0.	2.8087441E 05	-0.
0.	2.0366129E 05	0.
0.	2.0366129E 05	0.
-0.	1.1714238E 05	-0.

0.	6.5579891E 04	C.
0.	6.5579891E 04	C.
-0.	5.9182396E 04	-0.
0.	5.3265316E 04	C.
0.	5.3265316E 04	0.
-0.	2.1862377E 04	-C.
0.	-0.	C.

FLUX GROUP NO. 2

0.	9.9821321E 05	C.
-0.	9.9643002E 05	-C.
0.	9.9108043E 05	C.
0.	9.9108043E 05	0.
-0.	9.5378919E 05	-0.
0.	8.8363928E 05	C.
0.	8.8363928E 05	C.
-0.	8.1754587E 05	-C.
0.	7.3309243E 05	0.
0.	7.3309243E 05	C.
-0.	6.2554778E 05	-C.
0.	4.8146876E 05	C.
0.	4.8146876E 05	0.
-0.	3.0736596E 05	-0.
0.	1.8550237E 05	C.
0.	1.8550237E 05	0.
-0.	1.6916328E 05	-0.
0.	1.5374988E 05	0.
0.	1.5374988E 05	C.
-0.	6.6289860E 04	-C.
0.	-0.	C.

FLUX GROUP NO. 3

0.	4.4023384E 05	C.
-0.	4.3952776E 05	-C.
0.	4.3740948E 05	C.
0.	4.3740948E 05	C.
-0.	4.2260690E 05	-C.
0.	3.9797389E 05	C.
0.	3.9797389E 05	C.
-0.	3.7836860E 05	-C.
0.	3.6071398E 05	C.
0.	3.6071398E 05	C.
-0.	3.5151362E 05	-C.
0.	3.6621626E 05	C.
0.	3.6621626E 05	C.
-0.	3.4968247E 05	-C.
0.	2.6990147E 05	C.
0.	2.6990147E 05	C.

-0.	2.5396335E 05	-C.
0.	2.3761103E 05	C.
0.	2.3761103E 05	C.
-0.	1.1798985E 05	-C.
0.	-0.	C.

FLUX GROUP NO. 4

0.	9.1142434E 03	C.
-0.	9.1176507E 03	-C.
0.	9.1278725E 03	C.
0.	9.1278725E 03	C.
-0.	9.0425276E 03	-C.
0.	1.0202853E 04	C.
0.	1.0202853E 04	C.
-0.	1.3020595E 04	-C.
0.	2.2064323E 04	C.
0.	2.2064323E 04	C.
-0.	4.5491008E 04	-C.
0.	1.1617365E 05	C.
0.	1.1617365E 05	C.
-0.	1.9777374E 05	-C.
0.	2.0140770E 05	C.
0.	2.0140770E 05	C.
-0.	1.9580750E 05	-C.
0.	1.8875614E 05	C.
0.	1.8875614E 05	C.
-0.	1.0811445E 05	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 1

0.	7.0450748E-01	C.
-0.	7.0343961E-01	-C.
0.	7.0023599E-01	C.
0.	7.0023599E-01	C.
-0.	6.7714025E-01	-C.
0.	6.4346040E-01	C.
0.	6.4346040E-01	C.
-0.	6.2379684E-01	-C.
0.	6.3228238E-01	C.
0.	6.3228238E-01	C.
-0.	7.1336186E-01	-C.
0.	1.0353925E 00	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.

0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	9.5872847E 00	C.
-0.	9.5727514E 00	-C.
0.	9.5291508E 00	0.
0.	9.5291508E 00	0.
-0.	9.2148377E 00	-0.
0.	8.7565068E 00	C.
0.	8.7565068E 00	C.
-0.	8.4889176E 00	-C.
0.	8.6044030E 00	C.
0.	8.6044030E 00	0.
-0.	9.7077930E 00	-0.
0.	1.4090207E 01	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.3715406E 02	0.
-0.	1.3694623E 02	-C.
0.	1.3632275E 02	C.
0.	1.3632275E 02	0.
-0.	1.3182707E 02	-C.
0.	1.2526971E 02	C.
0.	1.2526971E 02	C.
-0.	1.2144075E 02	-C.
0.	1.2308998E 02	C.
0.	1.2308998E 02	0.
-0.	1.3886802E 02	-0.
0.	2.0154085E 02	0.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	C.
-0.	0.	-C.

0.	-0.	C.
PRECURSOR GROUP NO. 4		
0.	1.7136504E 02	C.
-0.	1.7110545E 02	-C.
0.	1.7032667E 02	C.
0.	1.7032667E 02	C.
-0.	1.6471035E 02	-C.
0.	1.5651684E 02	C.
0.	1.5651684E 02	C.
-0.	1.5173202E 02	-C.
0.	1.5378997E 02	C.
0.	1.5378997E 02	C.
-0.	1.7349733E 02	-C.
0.	2.5178348E 02	C.
0.	-C.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 5		
0.	7.0415517E 02	C.
-0.	7.0308865E 02	-C.
0.	6.9988904E 02	C.
0.	6.9988904E 02	C.
-0.	6.7681257E 02	-C.
0.	6.4314359E 02	C.
0.	6.4314359E 02	0.
-0.	6.2348070E 02	-C.
0.	6.3193166E 02	C.
0.	6.3193166E 02	C.
-0.	7.1289829E 02	-C.
0.	1.0345441E 03	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO. 6

-135-

0.	3.1342950E 02	0.
-0.	3.1295479E 02	-0.
0.	3.1153064E 02	0.
0.	3.1153064E 02	0.
-0.	3.0125915E 02	-0.
0.	2.8627249E 02	0.
0.	2.8627249E 02	0.
-0.	2.7752008E 02	-0.
0.	2.8128115E 02	0.
0.	2.8128115E 02	0.
-0.	3.1731919E 02	-0.
0.	4.6048409E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

MATERIAL NO. 1

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

2.0300000E-04	1.0000000E-03	3.1750000E-03	1.4660000E-03
1.6600000E-03	2.9600000E-04		

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00	1.4000000E 00	3.1100000E-01	1.1500000E-01
3.1700000E-02	1.2700000E-02		

GROUP NO. 1

SIGA = 9.7447925E-04 SIGF = 7.9013388E-04 NU = 2.6906913E 00

DIFC = 2.6827937E 00 CHI = 7.5356217E-01 VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ... , 4 --

0.	0.	0.	0.
----	----	----	----

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.	-0.	-0.	-0.
-0.	0.		

GROUP NO. 2

SIGA = 1.5963384E-03 SIGF = 1.1891829E-03 NU = 2.3697750E 00

DIFC = 1.2395053E 00 CHI = 2.4643214E-01 VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ... , 4 --

1.8970602E-02	0.	0.	0.
---------------	----	----	----

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

1.0000000E 00	1.0000000E 00	1.0000000E 00	1.0000000E 00
---------------	---------------	---------------	---------------

1.0000000E 00

1.0000000E 00

-137-

GROUP NO. 3

SIGA = 1.4424144E-02

SIGF = 8.5840423E-03

NU = 2.3557598E 00

DIFC = 1.0389322E 00

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

7.0583003E-07

7.5373828E-03

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-0.

GROUP NO. 4

SIGA = 4.6856542E-02

SIGF = 3.8559482E-02

NU = 2.3518255E 00

DIFC = 9.7209517E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

4.6313683E-11

1.2080043E-08

9.7972933E-04

3.0161881E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-0.

MATERIAL NO. 2

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1, 6

-0.
-0.-0.
-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1, 6

3.8700000E 00

1.4000000E 00

3.1100000E-01

1.1500000E-01

3.1700000E-02

1.2700000E-02

- 138 -

GROUP NO. 1

SIGA = -6.873625E-03

SIGF = 0.

NU = 0.

DIFC = 1.557768E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-0.

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.-0.
0.

-0.

-0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 3

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1, 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1, 6

3.8700000E 00

1.4000000E 00

3.1100000E-01

1.1500000E-01

3.1700000E-02

1.2700000E-02

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

-140-

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.
-0.

-0.
0.

-0.

-0.

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

0.

-0.

-0.

0.

-14/-

ERROR LIMITS FROM NOW ON ARE 5.9999999E-05 6.9999999E-03 5.9999999E-05 5.9999999E-05

ENTRY TO SUBROUTINE FEEDBK (INITIALLY IN THIS DT).

T = 2.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX C.

1 OUT-OF-ICE ITERATIONS

T = 2.0000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE PCWER DENSITY = 2.0509145E 03)

0.	5.6455413E 03	0.
-0.	5.6369870E 03	-0.
0.	5.6113243E 03	0.
0.	5.6113243E 03	0.
-0.	5.4262320E 03	-0.
0.	5.1570842E 03	0.
0.	5.1570842E 03	0.
-0.	5.0012000E 03	-0.
0.	5.0744085E 03	0.
0.	5.0744085E 03	0.
-0.	5.7373833E 03	-0.
0.	8.3568035E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	4.1517956E 05	0.
-0.	4.1454162E 05	-0.
0.	4.1262779E 05	0.

0.	4.1262779E 05	0.
-0.	3.9947835E 05	-0.
0.	3.7540793E 05	0.
0.	3.7540793E 05	0.
-0.	3.5307302E 05	-0.
0.	3.2327676E 05	0.
0.	3.2327676E 05	0.
-0.	2.8087654E 05	-0.
0.	2.0366295E 05	0.
0.	2.0366295E 05	0.
-0.	1.1714318E 05	-0.
0.	6.5580110E 04	0.
0.	6.5580110E 04	0.
-0.	5.9182567E 04	-0.
0.	5.3265464E 04	0.
0.	5.3265464E 04	0.
-0.	2.1862438E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	9.9821506E 05	0.
-0.	9.9643188E 05	-0.
0.	9.9108233E 05	0.
0.	9.9108233E 05	0.
-0.	9.5379132E 05	-0.
0.	8.8364172E 05	0.
0.	8.8364172E 05	0.
-0.	8.1754848E 05	-0.
0.	7.3309515E 05	0.
0.	7.3309515E 05	0.
-0.	6.2555040E 05	-0.
0.	4.8147077E 05	0.
0.	4.8147077E 05	0.
-0.	3.0736671E 05	-0.
0.	1.8550180E 05	0.
0.	1.8550180E 05	0.
-0.	1.6916265E 05	-0.
0.	1.5374930E 05	0.
0.	1.5374930E 05	0.
-0.	6.6289670E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	4.4023457E 05	0.
-0.	4.3952848E 05	-0.
0.	4.3741022E 05	0.
0.	4.3741022E 05	0.
-0.	4.2260764E 05	-0.

0.	3.9797446E 05	0.
-0.	9.9643188E 05	-0.
0.	9.9108233E 05	0.
0.	9.9108233E 05	0.
-0.	9.5379132E 05	-0.
0.	8.8364172E 05	0.
0.	8.8364172E 05	0.
-0.	8.1754848E 05	-0.
0.	7.3309515E 05	0.
0.	7.3309515E 05	0.
-0.	6.2555040E 05	-0.
0.	4.8147077E 05	0.
0.	4.8147077E 05	0.
-0.	3.0736671E 05	-0.
0.	1.8550180E 05	0.
0.	1.8550180E 05	0.
-0.	1.6916265E 05	-0.
0.	1.5374930E 05	0.
0.	1.5374930E 05	0.
-0.	6.6289670E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	4.4023457E 05	0.
-0.	4.3952848E 05	-0.
0.	4.3741022E 05	0.
0.	4.3741022E 05	0.
-0.	4.2260764E 05	-0.
0.	3.9797446E 05	0.
0.	3.9797446E 05	0.
-0.	3.7836894E 05	-0.
0.	3.6071388E 05	0.
0.	3.6071388E 05	0.
-0.	3.5151286E 05	-0.
0.	3.6621440E 05	0.
0.	3.6621440E 05	0.
-0.	3.4967877E 05	-0.
0.	2.6989532E 05	0.
0.	2.6989532E 05	0.
-0.	2.5395730E 05	-0.
0.	2.3760545E 05	0.
0.	2.3760545E 05	0.
-0.	1.1798747E 05	-0.
0.	-0.	0.

FLUX GROUP NO. 4

0.	9.1142675E 03	0.
-0.	9.1176754E 03	-0.

0.	9.1278994E 03	0.
0.	9.1278994E 03	0.
-0.	9.0425613E 03	-0.
0.	1.0202949E 04	0.
0.	1.0202949E 04	0.
-0.	1.3020802E 04	-0.
0.	2.2064848E 04	0.
0.	2.2064848E 04	0.
-0.	4.5492265E 04	-0.
0.	1.1617702E 05	0.
0.	1.1617702E 05	0.
-0.	1.9778052E 05	-0.
0.	2.0141792E 05	0.
0.	2.0141792E 05	0.
-0.	1.9581752E 05	-0.
0.	1.8876541E 05	0.
0.	1.8876541E 05	0.
-0.	1.0811852E 05	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	7.0450748E-01	0.
-0.	7.0343961E-01	-0.
0.	7.0023599E-01	0.
0.	7.0023599E-01	0.
-0.	6.7714025E-01	-0.
0.	6.4346040E-01	0.
0.	6.4346040E-01	0.
-0.	6.2379684E-01	-0.
0.	6.3228238E-01	0.
0.	6.3228238E-01	0.
-0.	7.1336186E-01	-0.
0.	1.0353925E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	9.5872847E 00	0.
-0.	9.5727514E 00	-0.
0.	9.5291508E 00	0.
0.	9.5291508E 00	0.

-0.	9.2148377E 00	-0.
0.	8.7565068E 00	0.
0.	8.7565068E 00	0.
-0.	8.4889176E 00	-0.
0.	8.6044030E 00	0.
0.	8.6044030E 00	0.
-0.	9.7077930E 00	-0.
0.	1.4090207E 01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.3715406E 02	0.
-0.	1.3694623E 02	-0.
0.	1.3632275E 02	0.
0.	1.3632275E 02	0.
-0.	1.3182707E 02	-0.
0.	1.2526971E 02	0.
0.	1.2526971E 02	0.
-0.	1.2144075E 02	-0.
0.	1.2308998E 02	0.
0.	1.2308998E 02	0.
-0.	1.3886802E 02	-0.
0.	2.0154085E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.7136504E 02	0.
-0.	1.7110545E 02	-0.
0.	1.7032667E 02	0.
0.	1.7032667E 02	0.
-0.	1.6471035E 02	-0.
0.	1.5651684E 02	0.

0.	1.5651684E 02	C.
-0.	1.5173202E 02	-0.
0.	1.5378997E 02	C.
0.	1.5378997E 02	C.
-0.	1.7349733E 02	-C.
0.	2.5178348E 02	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO. 5

0.	7.0415517E 02	0.
-0.	7.0308865E 02	-0.
0.	6.9988904E 02	0.
0.	6.9988904E 02	0.
-0.	6.7681257E 02	-0.
0.	6.4314359E 02	C.
0.	6.4314359E 02	C.
-0.	6.2348070E 02	-C.
0.	6.3193166E 02	0.
0.	6.3193166E 02	C.
-0.	7.1289829E 02	-0.
0.	1.0345441E 03	0.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 6

0.	3.1342950E 02	C.
-0.	3.1295479E 02	-0.
0.	3.1153064E 02	0.
0.	3.1153064E 02	0.
-0.	3.0125915E 02	-0.
0.	2.8627249E 02	0.
0.	2.8627249E 02	0.
-0.	2.7752208E 02	-C.

0.	2.8128115E 02	0.
0.	2.8128115E 02	0.
-0.	3.1731919E 02	-0.
0.	4.6048409E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

T = 2.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

71 OUT-OF-ICE ITERATIONS
T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

72 OUT-OF-ICE ITERATIONS
T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

63 OUT-OF-ICE ITERATIONS
T = 2.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

85 OUT-OF-ICE ITERATIONS
T = 2.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

68 OUT-OF-ICE ITERATIONS
T = 2.0625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

83 OUT-OF-ICE ITERATIONS
T = 2.0625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

66 OUT-OF-ICE ITERATIONS
T = 2.1250000E 00

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

1 OUT-OF-ICE ITERATIONS

T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

68 OUT-OF-ICE ITERATIONS
T = 2.1875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

61 OUT-OF-ICE ITERATIONS
T = 2.1875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

59 OUT-OF-ICE ITERATIONS
T = 2.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.3750000E 00

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

61 OUT-OF-ICE ITERATIONS
T = 2.3125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

55 OUT-OF-ICE ITERATIONS
T = 2.3125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.3750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

53 OUT-OF-ICE ITERATIONS
T = 2.3750000E 00

1 OUT-OF-ICE ITERATIONS

T = 2.5000000E 00

56 OUT-OF-ICE ITERATIONS
T = 2.4375000E 00

50 OUT-OF-ICE ITERATIONS
T = 2.4375000E 00

1 OUT-OF-ICE ITERATIONS
T = 2.5000000E 00

49 OUT-OF-ICE ITERATIONS
T = 2.5000000E 00

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 2.5000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE POWER DENSITY = 3.5396045E 02)

0.	1.0437303E 03	C.
-0.	1.0419016E 03	-C.
0.	1.0364155E 03	C.
0.	1.0364155E 03	C.
-0.	9.9740779E 02	-C.
0.	9.3698938E 02	C.
0.	9.3698938E 02	C.
-0.	8.9560602E 02	-C.
0.	8.8355066E 02	C.
0.	8.8355066E 02	C.
-0.	9.5153753E 02	-C.

0.	1.2862325E 03	C.
0.	-0.	C.
-0.	C.	-C.
0.	-C.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-C.	0.
0.	-0.	0.
-0.	0.	-C.
0.	-C.	0.

FLUX GROUP NO. 1

0.	7.5296918E 04	C.
-0.	7.5160875E 04	-C.
0.	7.4752744E 04	C.
0.	7.4752744E 04	0.
-0.	7.1937690E 04	-C.
0.	6.6795398E 04	C.
0.	6.6795398E 04	0.
-0.	6.2056172E 04	-C.
0.	5.5925136E 04	C.
0.	5.5925136E 04	C.
-0.	4.7656158E 04	-C.
0.	3.4098172E 04	0.
0.	3.4098172E 04	C.
-0.	1.9612395E 04	-C.
0.	1.0979220E 04	0.
0.	1.0979220E 04	C.
-0.	9.9081198E 03	-C.
0.	8.9174923E 03	C.
0.	8.9174923E 03	C.
-0.	3.6601224E 03	-C.
0.	-C.	C.

FLUX GROUP NO. 2

0.	1.8597672E 05	C.
-0.	1.8560571E 05	-C.
0.	1.8449267E 05	C.
0.	1.8449267E 05	C.
-0.	1.7674187E 05	-C.
0.	1.6237155E 05	C.
0.	1.6237155E 05	C.
-0.	1.4904342E 05	-C.
0.	1.3237458E 05	0.
0.	1.3237458E 05	0.
-0.	1.1172028E 05	-C.
0.	8.5016845E 04	C.
0.	1.5111345E 04	C.

-0.	5.3678002E 04	-0.
0.	3.2156771E 04	0.
0.	3.2156771E 04	0.
-0.	2.9295419E 04	-0.
0.	2.6601928E 04	0.
0.	2.6601928E 04	0.
-0.	1.1418748E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	8.1390578E 04	0.
-0.	8.1243994E 04	-0.
0.	8.0804243E 04	0.
0.	8.0804243E 04	0.
-0.	7.7738280E 04	-0.
0.	7.2626195E 04	0.
0.	7.2626195E 04	0.
-0.	6.8529119E 04	-0.
0.	6.4723660E 04	0.
0.	6.4723660E 04	0.
-0.	6.2394527E 04	-0.
0.	6.4276574E 04	0.
0.	6.4276574E 04	0.
-0.	6.0530214E 04	-0.
0.	4.5647400E 04	0.
0.	4.5647400E 04	0.
-0.	4.2779695E 04	-0.
0.	3.9992350E 04	0.
0.	3.9992350E 04	0.
-0.	1.9936848E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 4

0.	1.6705405E 03	0.
-0.	1.6699769E 03	-0.
0.	1.6682861E 03	0.
0.	1.6682861E 03	0.
-0.	1.6359217E 03	-0.
0.	1.7556321E 03	0.
0.	1.7556321E 03	0.
-0.	2.1026370E 03	-0.
0.	3.2768616E 03	0.
0.	3.2768616E 03	0.
-0.	6.3649632E 03	-0.
0.	1.5727310E 04	0.
0.	1.5727310E 04	0.
-0.	2.4826897E 04	-0.
0.	2.0656246E 04	0.

0.	2.0656246E 04	C.
-0.	1.9387900E 04	-C.
0.	1.8782038E 04	C.
0.	1.8782038E 04	C.
-0.	1.1991973E 04	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 1

0.	2.6601852E-01	C.
-0.	2.6559082E-01	-C.
0.	2.6430772E-01	C.
0.	2.6430772E-01	C.
-0.	2.5511291E-01	-C.
0.	2.4134383E-01	C.
0.	2.4134383E-01	C.
-0.	2.3268203E-01	-C.
0.	2.3337226E-01	C.
0.	2.3337226E-01	C.
-0.	2.5863193E-01	-C.
0.	3.6555900E-01	C.
0.	-0.	C.
-0.	C.	-C.
0.	-C.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 2

0.	6.2854710E 00	C.
-0.	6.2757567E 00	-C.
0.	6.2466124E 00	C.
0.	6.2466124E 00	C.
-0.	6.0369373E 00	-C.
0.	5.7284392E 00	C.
0.	5.7284392E 00	0.
-0.	5.5435796E 00	-C.
0.	5.6001263E 00	0.
0.	5.6001263E 00	0.
-0.	6.2826705E 00	-C.
0.	9.0438869E 00	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-C.

0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.2243761E 02	0.
-0.	1.2225125E 02	-0.
0.	1.2169218E 02	0.
0.	1.2169218E 02	0.
-0.	1.1766283E 02	-0.
0.	1.1177346E 02	0.
0.	1.1177346E 02	0.
-0.	1.0831348E 02	-0.
0.	1.0970068E 02	0.
0.	1.0970068E 02	0.
-0.	1.2360459E 02	-0.
0.	1.7905656E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.6433059E 02	0.
-0.	1.6408126E 02	-0.
0.	1.6333326E 02	0.
0.	1.6333326E 02	0.
-0.	1.5793980E 02	-0.
0.	1.5006561E 02	0.
0.	1.5006561E 02	0.
-0.	1.4545719E 02	-0.
0.	1.4739004E 02	0.
0.	1.4739004E 02	0.
-0.	1.6620189E 02	-0.
0.	2.4103751E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.

-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 5

0.	6.9607139E 02	C.
-0.	6.9501665E 02	-C.
0.	6.9185241E 02	C.
0.	6.9185241E 02	C.
-0.	6.6903201E 02	-0.
0.	6.3573001E 02	C.
0.	6.3573001E 02	C.
-0.	6.1626986E 02	-0.
0.	6.2457714E 02	C.
0.	6.2457714E 02	C.
-0.	7.0451488E 02	-C.
0.	1.0221960E 03	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	C.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO. 6

0.	3.1198318E 02	C.
-0.	3.1151058E 02	-C.
0.	3.1009276E 02	C.
0.	3.1009276E 02	C.
-0.	2.9986708E 02	-0.
0.	2.8494608E 02	C.
0.	2.8494608E 02	C.
-0.	2.7622995E 02	-C.
0.	2.7996531E 02	C.
0.	2.7996531E 02	C.
-0.	3.1581928E 02	-0.
0.	4.5827487E 02	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	0.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	0.

T = 2.500000E 00 ,NUCLEAR CONSTANTS

-155-

MATERIAL NO. 1

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

2.0300000E-04	1.0000000E-03	3.1750000E-03	1.4660000E-03
1.6600000E-03	2.9600000E-04		

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00	1.4000000E 00	3.1100000E-01	1.1500000E-01
3.1700000E-02	1.2700000E-02		

GROUP NO. 1

SIGA = 9.7447925E-04	SIGF = 7.9013388E-04	NU = 2.6906913E 00
DIFC = 2.6827937E 00	CHI = 7.5356217E-01	VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.	0.	0.	0.
----	----	----	----

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.	-0.	-0.	-0.
-0.	0.		

GROUP NO. 2

SIGA = 1.5963384E-03	SIGF = 1.1891829E-03	NU = 2.3697750E 00
DIFC = 1.2395053E 00	CHI = 2.4643214E-01	VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

1.8970602E-02	0.	0.	0.
---------------	----	----	----

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

1.0000000E 00	1.0000000E 00	1.0000000E 00	1.0000000E 00
---------------	---------------	---------------	---------------

1.0000000E 00

1.0000000E 00

- 156 -

GROUP NO. 3

SIGA = 1.4424144E-02

SIGF = 8.5840423E-03

NU = 2.3557598E 00

DIFC = 1.0389322E 00

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

7.0583003E-07

7.5373828E-03

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 4.6856542E-02

SIGF = 3.8559482E-02

NU = 2.3518255E 00

DIFC = 9.7209517E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

4.6313683E-11

1.2080043E-08

9.7972933E-04

3.0161881E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 2

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1, 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1, 6

2.87 00E 00

1.4000000E 00

3.1100000E-01

1.1500000E-01

3.1700000E-02

1.2700000E-02

-157-

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ..., 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 3

SIGA = 3.2825832E-05

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ..., 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1, 6

0.

-0.

-0.

-0.

-0.

0.

-158-

GROUP NO. 4

SIGA = 4.4762779E-04

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ..., 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

MATERIAL NO. 3

BETA THE FRACTION FOR EACH PRECURSOR GROUP IJ = 1 , 6

-0.

-0.

-0.

-0.

-0.

-0.

LAMBDA THE DECAY CONSTANT FOR EACH PRECURSOR GROUP IJ = 1 , 6

3.8700000E 00

1.4000000E 00

3.1100000E-01

1.1500000E-01

3.1700000E-02

1.2700000E-02

GROUP NO. 1

SIGA = -6.8736250E-03

SIGF = 0.

NU = 0.

DIFC = 1.5577680E 00

CHI = 7.5356217E-01

VEL = 3.2200000E 09

TRANSFERS INTO 1 FROM 1, 2, ..., 4 --

0.

0.

0.

0.

FISSION SPECTRUM IN GROUP 1 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

0.

-0.

-0.

0.

-159-

GROUP NO. 2

SIGA = 3.0113333E-06

SIGF = 0.

NU = 0.

DIFC = 6.1318058E-01

CHI = 2.4643214E-01

VEL = 8.9000000E 08

TRANSFERS INTO 2 FROM 1, 2, ... , 4 --

4.9542869E-02

0.

0.

0.

FISSION SPECTRUM IN GROUP 2 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 3

SIGA = 7.9999999E-03

SIGF = 0.

NU = 0.

DIFC = 5.4000439E-01

CHI = 0.

VEL = 7.2800000E 07

TRANSFERS INTO 3 FROM 1, 2, ... , 4 --

0.

2.7924262E-02

0.

0.

FISSION SPECTRUM IN GROUP 3 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

GROUP NO. 4

SIGA = 9.9999999E-02

SIGF = 0.

NU = -0.

DIFC = 5.2912293E-01

CHI = 0.

VEL = 1.3300000E 06

TRANSFERS INTO 4 FROM 1, 2, ... , 4 --

0.

0.

1.3289794E-02

6.4573503E-01

FISSION SPECTRUM IN GROUP 4 FOR EACH PRECURSOR GROUP IJ = 1 , 6

0.

-0.

-0.

-0.

-0.

0.

-160-

ERROR LIMITS FROM NCW CN ARE 5.9999999E-05 6.9999999E-03 5.9999999E-05 5.9999999E-05

ENTRY TO SUBROUTINE FEEDBK (INITIALLY IN THIS CT).

T = 2.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX C.

1 OUT-OF-ICE ITERATIONS

T = 2.5000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE PCWER DENSITY = 3.4816274E 02)

0.	1.0266410E 03	0.
-0.	1.0248423E 03	-0.
0.	1.0194461E 03	0.
0.	1.0194461E 03	0.
-0.	9.8107671E 02	-0.
0.	9.2164602E 02	0.
0.	9.2164602E 02	0.
-0.	8.8093915E 02	-0.
0.	8.6907892E 02	0.
0.	8.6907892E 02	0.
-0.	9.3594818E 02	-0.
0.	1.2651508E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	7.4064096E 04	0.
-0.	7.3930268E 04	-0.
0.	7.3528784E 04	0.

0.	7.3528784E 04	0.
-0.	7.0759692E 04	-0.
0.	6.5701481E 04	0.
0.	6.5701481E 04	0.
-0.	6.1039776E 04	-0.
0.	5.5009066E 04	0.
0.	5.5009066E 04	0.
-0.	4.6875470E 04	-0.
0.	3.3539580E 04	0.
0.	3.3539580E 04	0.
-0.	1.9291174E 04	-0.
0.	1.0799494E 04	0.
0.	1.0799494E 04	0.
-0.	9.7459384E 03	-0.
0.	8.7715282E 03	0.
0.	8.7715282E 03	0.
-0.	3.6002125E 03	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	1.8293181E 05	0.
-0.	1.8256689E 05	-0.
0.	1.8147215E 05	0.
0.	1.8147215E 05	0.
-0.	1.7384821E 05	-0.
0.	1.5971279E 05	0.
0.	1.5971279E 05	0.
-0.	1.4660267E 05	-0.
0.	1.3020651E 05	0.
0.	1.3020651E 05	0.
-0.	1.0989013E 05	-0.
0.	8.3624070E 04	0.
0.	8.3624070E 04	0.
-0.	5.2798957E 04	-0.
0.	3.1630624E 04	0.
0.	3.1630624E 04	0.
-0.	2.8816139E 04	-0.
0.	2.6166719E 04	0.
0.	2.6166719E 04	0.
-0.	1.1231924E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 3

0.	8.0057939E 04	0.
-0.	7.9913757E 04	-0.
0.	7.9481212E 04	0.
0.	7.9481212E 04	0.
-0.	7.6465433E 04	-0.

0.	7.1437010E 04	C.
0.	7.1437010E 04	C.
-0.	6.7407033E 04	-C.
0.	6.3663955E 04	C.
0.	6.3663955E 04	C.
-0.	6.1373099E 04	-C.
0.	6.3224605E 04	C.
0.	6.3224605E 04	C.
-0.	5.9540078E 04	-C.
0.	4.4901577E 04	C.
0.	4.4901577E 04	C.
-0.	4.2080805E 04	-C.
0.	3.9338987E 04	C.
0.	3.9338987E 04	C.
-0.	1.9611058E 04	-C.
0.	-0.	0.

FLUX GROUP NO. 4

0.	1.6431855E 03	C.
-0.	1.6426310E 03	-C.
0.	1.6409673E 03	C.
0.	1.6409673E 03	C.
-0.	1.6091307E 03	-C.
0.	1.7268654E 03	0.
0.	1.7268654E 03	C.
-0.	2.0681644E 03	-C.
0.	3.2230974E 03	C.
0.	3.2230974E 03	C.
-0.	6.2604979E 03	-C.
0.	1.5469166E 04	C.
0.	1.5469166E 04	C.
-0.	2.4419182E 04	-C.
0.	2.0316244E 04	C.
0.	2.0316244E 04	C.
-0.	1.9068750E 04	-C.
0.	1.8472971E 04	0.
0.	1.8472971E 04	C.
-0.	1.1794986E 04	-C.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	2.6601852E-01	C.
-0.	2.6559082E-01	-0.
0.	2.6430772E-01	C.
0.	2.6430772E-01	0.
-0.	2.5511291E-01	-C.
0.	2.4134383E-01	0.
0.	2.4134383E-01	0.

-0.	2.3268203E-01	-0.
0.	2.3337226E-01	0.
0.	2.3337226E-01	0.
-0.	2.5863193E-01	-0.
0.	3.6555900E-01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	6.2854710E 00	0.
-0.	6.2757567E 00	-0.
0.	6.2466124E 00	0.
0.	6.2466124E 00	0.
-0.	6.0369373E 00	-0.
0.	5.7284392E 00	0.
0.	5.7284392E 00	0.
-0.	5.5435796E 00	-0.
0.	5.6001263E 00	0.
0.	5.6001263E 00	0.
-0.	6.2826705E 00	-0.
0.	9.0438869E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	1.2243761E 02	0.
-0.	1.2225125E 02	-0.
0.	1.2169218E 02	0.
0.	1.2169218E 02	0.
-0.	1.1766283E 02	-0.
0.	1.1177346E 02	0.
0.	1.1177346E 02	0.
-0.	1.0831348E 02	-0.
0.	1.0970068E 02	0.

0.	1.0970068E 02	C.
-0.	1.2360459E 02	-C.
0.	1.7905656E 02	C.
0.	-C.	C.
-0.	0.	-C.
0.	-C.	C.
0.	-C.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	C.	-C.
0.	-C.	C.

PRECURSOR GROUP NO. 4

0.	1.6433059E 02	C.
-0.	1.6408126E 02	-C.
0.	1.6333326E 02	C.
0.	1.6333326E 02	C.
-0.	1.5793980E 02	-C.
0.	1.5006561E 02	C.
0.	1.5006561E 02	C.
-0.	1.4545719E 02	-C.
0.	1.4739004E 02	C.
0.	1.4739004E 02	C.
-0.	1.6620189E 02	-C.
0.	2.4103751E 02	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-C.	C.
0.	-0.	C.
-0.	C.	-C.
0.	-C.	C.

PRECURSOR GROUP NO. 5

0.	6.9607139E 02	0.
-0.	6.9501665E 02	-0.
0.	6.9185241E 02	C.
0.	6.9185241E 02	0.
-0.	6.6903201E 02	-0.
0.	6.3573001E 02	0.
0.	6.3573001E 02	0.
-0.	6.1626986E 02	-C.
0.	6.2457714E 02	0.
0.	6.2457714E 02	0.
-0.	7.0451488E 02	-C.

0.	1.0221960E 03	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 6

0.	3.1198318E 02	0.
-0.	3.1151058E 02	-0.
0.	3.1009276E 02	0.
0.	3.1009276E 02	0.
-0.	2.9986708E 02	-0.
0.	2.8494608E 02	0.
0.	2.8494608E 02	0.
-0.	2.7622995E 02	-0.
0.	2.7996531E 02	0.
0.	2.7996531E 02	0.
-0.	3.1581928E 02	-0.
0.	4.5827487E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

T = 3.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.6250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-CF-ICE ITERATIONS
T = 2.6250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 2.5625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 2.5625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 2.6250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.6250000E 00

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

1 CUT-OF-ICE ITERATIONS

T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.6875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.6875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 2.7500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 2.8750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.8125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.8125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.8750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.8750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 3.0000000E 00

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.9375000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 2.9375000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 3.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

T = 3.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 3.0000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE PCWER DENSITY = 3.1223277E 02)

0.	9.2073185E 02	C.
-0.	9.1911843E 02	-C.
0.	9.1427817E 02	C.
0.	9.1427817E 02	C.
-0.	8.7986248E 02	-C.
0.	8.2655194E 02	C.
0.	8.2655194E 02	C.
-0.	7.9003599E 02	-C.
0.	7.7938759E 02	C.
0.	7.7938759E 02	C.
-0.	8.3934126E 02	-C.
0.	1.1345389E 03	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-C.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-C.	0.
-0.	0.	-C.
0.	-C.	C.

FLUX GROUP NO. 1

0.	6.6421789E 04	0.
-0.	6.6301761E 04	-C.
0.	6.5941675E 04	C.
0.	6.5941675E 04	C.
-0.	6.3458038E 04	-C.
0.	5.8921181E 04	C.
0.	5.8921181E 04	C.
-0.	5.4740141E 04	-0.
0.	4.9331437E 04	0.
0.	4.9331437E 04	0.
-0.	4.2037081E 04	-C.
0.	3.0077642E 04	0.
0.	3.0077642E 04	C.
-0.	1.7300109E 04	-0.
0.	9.6051017E 03	0.

0.	9.6851017E 03	C.
-0.	8.7402904E 03	-0.
0.	7.8664317E 03	C.
0.	7.8664317E 03	0.
-0.	3.2287219E 03	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	1.6406271E 05	C.
-0.	1.6373536E 05	-C.
0.	1.6275333E 05	C.
0.	1.6275333E 05	C.
-0.	1.5591468E 05	-C.
0.	1.4323476E 05	C.
0.	1.4323476E 05	0.
-0.	1.3147504E 05	-C.
0.	1.1676839E 05	C.
0.	1.1676839E 05	0.
-0.	9.8546579E 04	-0.
0.	7.4991192E 04	0.
0.	7.4991192E 04	C.
-0.	4.7349505E 04	-C.
0.	2.8367588E 04	C.
0.	2.8367588E 04	C.
-0.	2.5843620E 04	-C.
0.	2.3467517E 04	C.
0.	2.3467517E 04	C.
-0.	1.0073258E 04	-C.
0.	-0.	C.

FLUX GROUP NO. 3

0.	7.1799018E 04	C.
-0.	7.1669697E 04	-C.
0.	7.1281732E 04	0.
0.	7.1281732E 04	0.
-0.	6.8576769E 04	-C.
0.	6.4066541E 04	C.
0.	6.4066541E 04	C.
-0.	6.0452053E 04	-0.
0.	5.7095156E 04	C.
0.	5.7095156E 04	C.
-0.	5.5041136E 04	-C.
0.	5.6702853E 04	C.
0.	5.6702853E 04	C.
-0.	5.3401055E 04	-C.
0.	4.0276474E 04	C.
0.	4.0276474E 04	C.
-0.	3.7746644E 04	-C.

0.	3.5287125E 04	C.
0.	3.5287125E 04	C.
-0.	1.7590635E 04	-C.
0.	-0.	0.

FLUX GROUP NO. 4

0.	1.4736554E 03	C.
-0.	1.4731573E 03	-C.
0.	1.4716629E 03	C.
0.	1.4716629E 03	C.
-0.	1.4430998E 03	-C.
0.	1.5486235E 03	C.
0.	1.5486235E 03	C.
-0.	1.8546184E 03	-C.
0.	2.8901571E 03	C.
0.	2.8901571E 03	C.
-0.	5.6136910E 03	-C.
0.	1.3870924E 04	C.
0.	1.3870924E 04	C.
-0.	2.1895555E 04	-C.
0.	1.8213865E 04	C.
0.	1.8213865E 04	C.
-0.	1.7095396E 04	-C.
0.	1.6561696E 04	C.
0.	1.6561696E 04	C.
-0.	1.0575948E 04	-C.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	1.4127763E-01	C.
-0.	1.4103550E-01	-C.
0.	1.4030911E-01	C.
0.	1.4030911E-01	C.
-0.	1.3513513E-01	-C.
0.	1.2717754E-01	C.
0.	1.2717754E-01	C.
-0.	1.2181990E-01	-C.
0.	1.2065166E-01	C.
0.	1.2065166E-01	C.
-0.	1.3080463E-01	-C.
0.	1.7867707E-01	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	0.	-

-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO. 2

0.	3.9542416E 00	0.
-0.	3.9479571E 00	-C.
0.	3.9291033E 00	C.
0.	3.9291033E 00	0.
-0.	3.7938401E 00	-C.
0.	3.5923161E 00	C.
0.	3.5923161E 00	C.
-0.	3.4672723E 00	-C.
0.	3.4851077E 00	0.
0.	3.4851077E 00	C.
-0.	3.8767754E 00	-0.
0.	5.5107087E 00	0.
0.	-0.	C.
-0.	0.	-0.
0.	-C.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-C.	0.

PRECURSOR GROUP NO. 3

0.	1.0794071E 02	C.
-0.	1.0777562E 02	-C.
0.	1.0728034E 02	C.
0.	1.0728034E 02	C.
-0.	1.0371259E 02	-C.
0.	9.8486217E 01	C.
0.	9.8486217E 01	C.
-0.	9.5395592E 01	-C.
0.	9.6536807E 01	C.
0.	9.6536807E 01	C.
-0.	1.0862059E 02	-C.
0.	1.5703138E 02	C.
0.	-0.	0.
-0.	0.	-0.
0.	-C.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	C.
0.	-C.	C.
-0.	0.	-C.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.5674261E 02	C.
-0.	1.5650440E 02	-C.
0.	1.5578976E 02	C.
0.	1.5578976E 02	C.
-0.	1.5063777E 02	-C.
0.	1.4311041E 02	C.
0.	1.4311041E 02	C.
-0.	1.3869511E 02	-C.
0.	1.4049880E 02	C.
0.	1.4049880E 02	C.
-0.	1.5835711E 02	-C.
0.	2.2950504E 02	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-C.	C.
-0.	0.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 5

0.	6.8701313E 02	C.
-0.	6.8597166E 02	-C.
0.	6.8284723E 02	C.
0.	6.8284723E 02	C.
-0.	6.6031501E 02	-C.
0.	6.2742698E 02	C.
0.	6.2742698E 02	C.
-0.	6.0819725E 02	-C.
0.	6.1635017E 02	C.
0.	6.1635017E 02	C.
-0.	6.9514928E 02	-C.
0.	1.0084273E 03	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 6

0.	3.1034845E 02	0.
-0.	3.0987824E 02	-0.
0.	3.0846761E 02	0.
0.	3.0846761E 02	0.
-0.	2.9829394E 02	-0.
0.	2.8344764E 02	0.
0.	2.8344764E 02	0.
-0.	2.7477309E 02	-0.
0.	2.7848059E 02	0.
0.	2.7848059E 02	0.
-0.	3.1412905E 02	-0.
0.	4.5578998E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

T = 3.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.0625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.0625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.1250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 3.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.1875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 3.1875000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 3.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 3.2500000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 3.3750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.3125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.3125000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.3750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.3750000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 3.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.4375000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.4375000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.5000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX C.

1 OUT-OF-ICE ITERATIONS

T = 3.5000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE PCWER DENSITY = 2.8597168E 02)

0.	8.4332227E 02	C.
-0.	8.4184426E 02	-C.
0.	8.3741025E 02	C.
0.	8.3741025E 02	C.
-0.	8.0588282E 02	-C.
0.	7.5704518E 02	C.
0.	7.5704518E 02	C.
-0.	7.2359180E 02	-C.
0.	7.1383000E 02	C.
0.	7.1383000E 02	C.
-0.	7.6873254E 02	-C.
0.	1.0390928E 03	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	C.

FLUX GROUP NO. 1

0.	6.0836359E 04	C.
-0.	6.0726416E 04	-C.
0.	6.0396583E 04	C.

0.	6.0396583E 04	C.
-0.	5.8121578E 04	-C.
0.	5.3965846E 04	C.
0.	5.3965846E 04	C.
-0.	5.0136141E 04	-C.
0.	4.5182071E 04	C.
0.	4.5182071E 04	C.
-0.	3.8501064E 04	-C.
0.	2.7547528E 04	C.
0.	2.7547528E 04	C.
-0.	1.5844804E 04	-C.
0.	8.8703352E 03	C.
0.	8.8703352E 03	C.
-0.	8.0050017E 03	-C.
0.	7.2046565E 03	C.
0.	7.2046565E 03	C.
-0.	2.9571009E 03	-C.
0.	-0.	C.

FLUX GROUP NO. 2

0.	1.5027101E 05	C.
-0.	1.4997114E 05	-C.
0.	1.4907152E 05	C.
0.	1.4907152E 05	C.
-0.	1.4280669E 05	-C.
0.	1.3119085E 05	C.
0.	1.3119085E 05	C.
-0.	1.2041823E 05	-C.
0.	1.0694648E 05	C.
0.	1.0694648E 05	C.
-0.	9.0255505E 04	-C.
0.	6.8681095E 04	C.
0.	6.8681095E 04	C.
-0.	4.3365501E 04	-C.
0.	2.5980777E 04	C.
0.	2.5980777E 04	C.
-0.	2.3669177E 04	-C.
0.	2.1493002E 04	C.
0.	2.1493002E 04	C.
-0.	9.2257282E 03	-C.
0.	-0.	C.

FLUX GROUP NO. 3

0.	6.5762492E 04	C.
-0.	6.5644026E 04	-C.
0.	6.5288626E 04	C.
0.	6.5288626E 04	C.
-0.	6.21 88E 04	-C.

0.	5.8678991E 04	0.
0.	5.8678991E 04	0.
-0.	5.5367886E 04	-0.
0.	5.2292742E 04	0.
0.	5.2292742E 04	0.
-0.	5.0411031E 04	-0.
0.	5.1932737E 04	0.
0.	5.1932737E 04	0.
-0.	4.8908840E 04	-0.
0.	3.6888800E 04	0.
0.	3.6888800E 04	0.
-0.	3.4571792E 04	-0.
0.	3.2319121E 04	0.
0.	3.2319121E 04	0.
-0.	1.6111007E 04	-0.
0.	-0.	0.

FLUX GROUP NO. 4

0.	1.3497505E 03	0.
-0.	1.3492938E 03	-0.
0.	1.3479238E 03	0.
0.	1.3479238E 03	0.
-0.	1.3217537E 03	-0.
0.	1.4183826E 03	0.
0.	1.4183826E 03	0.
-0.	1.6986217E 03	-0.
0.	2.6470297E 03	0.
0.	2.6470297E 03	0.
-0.	5.1414287E 03	-0.
0.	1.2703993E 04	0.
0.	1.2703993E 04	0.
-0.	2.0053520E 04	-0.
0.	1.6681505E 04	0.
0.	1.6681505E 04	0.
-0.	1.5657138E 04	-0.
0.	1.5168355E 04	0.
0.	1.5168355E 04	0.
-0.	9.6862342E 03	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 1

0.	1.1410527E-01	0.
-0.	1.1390601E-01	-0.
0.	1.1330822E-01	0.
0.	1.1330822E-01	0.
-0.	1.0905737E-01	-0.
0.	1.0247061E-01	0.
0.	1.0247061E-01	0.

-0.	9.7956260E-02	-0.
0.	9.6634221E-02	0.
0.	9.6634221E-02	0.
-0.	1.0403085E-01	-0.
0.	1.4050081E-01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 2

0.	2.7205587E 00	0.
-0.	2.7161109E 00	-0.
0.	2.7027667E 00	0.
0.	2.7027667E 00	0.
-0.	2.6072947E 00	-0.
0.	2.4632969E 00	0.
0.	2.4632969E 00	0.
-0.	2.3709860E 00	-0.
0.	2.3705278E 00	0.
0.	2.3705278E 00	0.
-0.	2.6129339E 00	-0.
0.	3.6630895E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	9.5257013E 01	0.
-0.	9.5110615E 01	-0.
0.	9.4671422E 01	0.
0.	9.4671422E 01	0.
-0.	9.1509254E 01	-0.
0.	8.6866771E 01	0.
0.	8.6866771E 01	0.
-0.	8.4103796E 01	-0.
0.	8.5038912E 01	0.

0.	8.5038912E 01	0.
-0.	9.5549663E 01	-0.
0.	1.3785317E 02	0.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 4

0.	1.4943132E 02	C.
-0.	1.4920387E 02	-C.
0.	1.4852151E 02	C.
0.	1.4852151E 02	0.
-0.	1.4360301E 02	-0.
0.	1.3641168E 02	C.
0.	1.3641168E 02	C.
-0.	1.3218460E 02	-C.
0.	1.3386822E 02	C.
0.	1.3386822E 02	C.
-0.	1.5081714E 02	-0.
0.	2.1843743E 02	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	0.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 5

0.	6.7792457E 02	C.
-0.	6.7689645E 02	-C.
0.	6.7381212E 02	C.
0.	6.7381212E 02	C.
-0.	6.5157001E 02	-C.
0.	6.1909941E 02	C.
0.	6.1909941E 02	C.
-0.	6.0010338E 02	-0.
0.	6.0810648E 02	0.
0.	6.0810648E 02	C.
-0.	6.8577393E 02	-C.

0.	9.9466355E 02	0.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	0.
-0.	C.	-0.
C.	-0.	0.

PRECURSOR GROUP NO. 6

0.	3.0869310E 02	0.
-0.	3.0822533E 02	-C.
0.	3.0682200E 02	C.
0.	3.0682200E 02	C.
-0.	2.9670115E 02	-0.
0.	2.8193088E 02	C.
0.	2.8193088E 02	C.
-0.	2.7329888E 02	-C.
0.	2.7697907E 02	0.
0.	2.7697907E 02	0.
-0.	3.1242137E 02	-C.
0.	4.5328290E 02	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	0.
0.	-0.	C.
-0.	0.	-0.
0.	-C.	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	0.

T = 3.6250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.5625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.5625000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS
T = 3.6250000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 OUT-OF-ICE ITERATIONS
T = 3.6250000E 00

1 OUT-OF-ICE ITERATIONS

T = 3.7500000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.6875000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.6875000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.7500000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.7500000E 00

1 OUT-OF-ICE ITERATIONS

T = 3.8750000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.8125000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.8125000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.8750000E 00

1 OUT-OF-ICE ITERATIONS
T = 3.8750000E 00

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

1 CUT-OF-ICE ITERATIONS

T = 4.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

T = 3.9375000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

T = 3.9375000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

T = 4.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

T = 4.0000000E 00

ENTRY TO SUBROUTINE FLXCUT FROM BOX A.

1 CUT-OF-ICE ITERATIONS

ENTRY TO SUBROUTINE FEEDBK FROM BOX B.

T = 4.0000000E 00

POWER DENSITY, FLUX TIMES SIGMA FISSION

(AVERAGE POWER DENSITY = 2.6915900E 02)

0.	7.9376827E 02	0.
-0.	7.9237689E 02	-0.
0.	7.8820271E 02	0.
0.	7.8820271E 02	0.
-0.	7.5852287E 02	-0.
0.	7.1254789E 02	0.
0.	7.1254789E 02	0.
-0.	6.8105419E 02	-0.
0.	6.7185879E 02	0.
0.	6.7185879E 02	0.
-0.	7.2352612E 02	-0.
0.	9.7798302E 02	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

FLUX GROUP NO. 1

0.	5.7260876E 04	0.
-0.	5.7157383E 04	-0.
0.	5.6846904E 04	0.
0.	5.6846904E 04	0.
-0.	5.4705392E 04	-0.
0.	5.0793591E 04	0.
0.	5.0793591E 04	0.
-0.	4.7188753E 04	-0.
0.	4.2525688E 04	0.
0.	4.2525688E 04	0.
-0.	3.6237299E 04	-0.
0.	2.5927737E 04	0.
0.	2.5927737E 04	0.
-0.	1.4913121E 04	-0.
0.	8.3487408E 03	0.
0.	8.3487408E 03	0.
-0.	7.5342889E 03	-0.
0.	6.7810056E 03	0.
0.	6.7810056E 03	0.
-0.	2.7832164E 03	-0.
0.	-0.	0.

FLUX GROUP NO. 2

0.	1.4144223E 05	0.
-0.	1.4115992E 05	-0.
0.	1.4031301E 05	0.
0.	1.4031301E 05	0.
-0.	1.3441518E 05	-0.
0.	1.2348048E 05	0.
0.	1.2348048E 05	0.
-0.	1.1333968E 05	-0.
0.	1.0065837E 05	0.
0.	1.0065837E 05	0.
-0.	8.4947433E 04	-0.
0.	6.4641297E 04	0.
0.	6.4641297E 04	0.
-0.	4.0814883E 04	-0.
0.	2.4452675E 04	0.
0.	2.4452675E 04	0.
-0.	2.2277038E 04	-0.

0.	2.0228864E 04	C.
0.	2.0228864E 04	C.
-0.	8.6831226E 03	-C.
0.	-0.	C.

FLUX GROUP NO. 3

0.	6.1898186E 04	C.
-0.	6.1786664E 04	-C.
0.	6.1452094E 04	C.
0.	6.1452094E 04	C.
-0.	5.9119388E 04	-C.
0.	5.5229951E 04	C.
0.	5.5229951E 04	C.
-0.	5.2112993E 04	-C.
0.	4.9218134E 04	C.
0.	4.9218134E 04	C.
-0.	4.7446629E 04	-C.
0.	4.8878571E 04	C.
0.	4.8878571E 04	C.
-0.	4.6032430E 04	-C.
0.	3.4719334E 04	C.
0.	3.4719334E 04	C.
-0.	3.2538593E 04	-C.
C.	3.0418401E 04	C.
0.	3.0418401E 04	C.
-0.	1.5163492E 04	-C.
0.	-0.	C.

FLUX GROUP NO. 4

0.	1.2704320E 03	C.
-0.	1.2700018E 03	-C.
0.	1.2687110E 03	C.
0.	1.2687110E 03	C.
-0.	1.2440708E 03	-C.
0.	1.3350035E 03	C.
0.	1.3350035E 03	C.
-0.	1.5987511E 03	-C.
0.	2.4913728E 03	C.
0.	2.4913728E 03	C.
-0.	4.8390666E 03	-C.
0.	1.1956867E 04	C.
0.	1.1956867E 04	C.
-0.	1.8874162E 04	-C.
0.	1.5700463E 04	C.
0.	1.5700463E 04	C.
-0.	1.4736340E 04	-C.
0.	1.4276302E 04	C.
0.	1.4276302E 04	C.

-0.	9.1165820E 03	-0.
0.	-0.	0.

PRECURSOR GROUP NO.	1	
0.	1.0297728E-01	0.
-0.	1.0279681E-01	-0.
0.	1.0225542E-01	0.
0.	1.0225542E-01	0.
-0.	9.8406782E-02	-0.
0.	9.2435318E-02	0.
0.	9.2435318E-02	0.
-0.	8.8329804E-02	-0.
0.	8.7073755E-02	0.
0.	8.7073755E-02	0.
-0.	9.3616148E-02	-0.
0.	1.2616739E-01	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO.	2	
0.	2.0489222E 00	0.
-0.	2.0454895E 00	-0.
0.	2.0351913E 00	0.
0.	2.0351913E 00	0.
-0.	1.9616812E 00	-0.
0.	1.8496675E 00	0.
0.	1.8496675E 00	0.
-0.	1.7759615E 00	-0.
0.	1.7671396E 00	0.
0.	1.7671396E 00	0.
-0.	1.9316836E 00	-0.
0.	2.6733219E 00	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.
-0.	0.	-0.
0.	-0.	0.

PRECURSOR GROUP NO. 3

0.	8.4190184E 01	C.
-0.	8.4060166E 01	-C.
0.	8.3670110E 01	C.
0.	8.3670110E 01	C.
-0.	8.0863119E 01	-C.
0.	7.6732984E 01	C.
0.	7.6732984E 01	C.
-0.	7.4259294E 01	-C.
0.	7.5021505E 01	C.
0.	7.5021505E 01	C.
-0.	8.4174490E 01	-C.
0.	1.2118960E 02	C.
0.	-0.	C.
-0.	C.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
0.	-C.	C.
-0.	C.	-C.
0.	-0.	C.

PRECURSOR GROUP NO. 4

0.	1.4241465E 02	C.
-0.	1.4219756E 02	-C.
0.	1.4154627E 02	C.
0.	1.4154627E 02	C.
-0.	1.3685249E 02	-C.
0.	1.2998503E 02	0.
0.	1.2998503E 02	0.
-0.	1.2594023E 02	-0.
0.	1.2751198E 02	C.
0.	1.2751198E 02	0.
-0.	1.4359525E 02	-0.
0.	2.0784954E 02	0.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	0.
0.	-0.	0.
-0.	0.	-C.
0.	-0.	C.
0.	-0.	C.
-0.	0.	-0.
0.	-0.	C.

PRECURSOR GROUP NO. 5

C.	6.6884475E 02	C.
-C.	6.6783002E 02	-C.
0.	6.6478587E 02	C.
C.	6.6478587E 02	C.
-C.	6.4283430E 02	-C.
0.	6.1078237E 02	C.
0.	6.1078237E 02	C.
-C.	5.9202169E 02	-C.
C.	5.9987898E 02	C.
0.	5.9987898E 02	C.
-0.	6.7642406E 02	-C.
0.	9.8095200E 02	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.
C.	-C.	C.
-0.	0.	-C.
C.	-0.	C.
C.	-C.	0.
-C.	0.	-C.
C.	-C.	C.

PRECURSOR GROUP NO. 6

C.	3.0702418E 02	C.
-C.	3.0655887E 02	-C.
0.	3.0516292E 02	C.
0.	3.0516292E 02	C.
-0.	2.9509547E 02	-C.
0.	2.8040213E 02	C.
C.	2.8040213E 02	C.
-C.	2.7181337E 02	-C.
C.	2.7546672E 02	C.
C.	2.7546672E 02	C.
-C.	3.1070263E 02	-C.
C.	4.5076224E 02	C.
0.	-C.	0.
-0.	0.	-C.
0.	-0.	C.
C.	-0.	C.
-0.	0.	-C.
C.	-0.	C.
0.	-0.	C.
-0.	0.	-C.
0.	-0.	C.

Appendix C - Discussion of the New Sample Problem Printout

Appendix B contains a FORTRAN II listing of the VARI-QUIR kinetics code together with a portion of the printout obtained from running a sample problem. A listing of the data cards for this problem is also included. It should be emphasized that this problem is not any of those discussed in the main body of the test, but rather one which was newly devised for inclusion in this revision to illustrate some of the features of the code.

The problem represents the shutdown of a simplified reactor. Initially the reactor is in a steady state, or nearly so, maintaining a nearly constant power level. It consists, at this stage, of a central cylindrical core containing U-235 and U-238 fuel, together with various other materials, and an annular reflector region. The reflector region contains beryllium, and the $\text{Be}(n,2n)$ reaction accounts for the negative absorption cross section in the highest energy group.

Two seconds after the beginning of the problem, the absorption cross sections in a portion of the reflector are changed for groups three and four. The resulting cross sections resemble those which would be found if about 8×10^{19} atoms/cc of B-10 were added to the initial material. In that case the effect on the other cross sections would be slight, and in our problem it is ignored altogether. The change of cross sections is "ramped" over one-half second.

At the end of this one-half-second interval the power has dropped to about 17 percent of its original value. The reactor is now subcritical, and the power subsequently continues to fall off, although much more slowly than

during the insertion of poison. The problem was continued to ten seconds after the starting time, when the power was down to about seven percent of its starting value. The portion of the printout included in Appendix B runs only to four seconds after starting time because the remaining printout does not illustrate anything new about the code itself.

The following comments may be helpful to anyone not previously familiar with the code who wishes to follow the sample problem printout:

1. The first part of the printout states the overall structure of the problem as to groups, geometry and material distribution.
2. Immediately following this is a sample layout showing how the fluxes are to be printed when called for.
2. After the sample layout the nuclear parameters for each of the three materials are printed. These are the parameters initially input to the code, and will be used during the steady-state portion of the problem. All of the parameters for materials two and three are identical at this stage.
3. Following the nuclear parameter printout is a statement as to the number of outer iterations, number of inner iterations between outer iterations, number of outer iterations between printouts and sense switch control number. After this is the notation "Entry to the Steady State". No further printout occurs until five outer iterations have been accomplished (because ITPRNT = 5).

4. After five outer iterations, a tentative eigenvalue (reciprocal k_{eff}) of 1.0168806 is indicated, followed by the fluxes and adjoints which have been computed so far. The next printout, which is made after ten outer iterations, indicates a tentative eigenvalue of 1.0165686 and gives the fluxes and adjoints as now revised.
5. The next printout skips to a new page and gives the revised nuclear parameters. These are obtained from the previous ones simply by multiplying all of the λ 's by a factor which will make the reactor just critical, or, more exactly, by the best guess available of this factor after eleven outer iterations. The factor is the same as the quantity referred to as "eigenvalue" in the printouts, but has been revised by one more outer iteration since last printed. Of course, this modification in the parameters will not be made unless this option is selected.
6. The time-dependent chain has now been entered. The remaining printout is rather lengthy but, being heavily annotated, is hopefully easy to understand. For the first two seconds the fluxes and precursors are found to be nearly constant in time, indicating that the eigenvalue calculated in the steady state chain was pretty good. At $T = 2.0$, a set of nuclear parameters is printed. This is the same set as that last printed because no changes have been made during the preceding ramp. After the parameter printout a new set of fluxes and precursors is printed. The fluxes, but not the precursors, differ slightly from those printed before the change.

The reason for this difference is simply that Subroutine FLXOUT has made a few more iterations. This difference is considered negligible. If the difference were not negligible, this would simply mean that the error control, E3, needed to be reduced until the error reached a point which could be considered negligible.

7. Between $T = 2.0$ and $T = 2.5$ the code encounters a slowing down because of the very drastic change being made in the nuclear parameters. During this interval the parameters are "ramped" from their previous values to those printed after the flux and precursor printout at $T = 2.5$. As previously mentioned the change affects only the group three and group four absorption cross sections of material three. From now on, therefore, material three is no longer identical with material two.
8. The effect of the change in parameters on the fluxes and precursors is evident from the first $T = 2.5$ printout. The subsequent printout can be easily followed if one has followed the prior printout up to this point.

For reference, the power in units of the starting power is shown as a function of time in Figure 14.

In the sample problem fluxes and precursors are punched out at the end of the steady state calculation and also at the end of the problem. It is therefore possible to restart the problem at either $T = 0$ or $T = 10$. For example, to restart the problem at $T = 10$, have $LAMB = 0$ and $NGES = 1$ on Card No. 2, and include the punched fluxes and precursors where called for in the input instructions.

FIGURE 14 RELATIVE POWER AS A FUNCTION OF TIME

it may be noted that the fluxes which are punched at the end of a problem are not iterated upon by subroutine FLXOUT and that consequently the last printout, which gives these fluxes, is unreliable. When the problem is restarted, however, subroutine FLXOUT is called and a new printout is given corresponding to the same time as the last printout of the previous problem. This second printout is reliable, i.e. converged to the limits specified by the error criteria. If a converged solution at the final time of a given problem is desired this can be obtained by adding a time ramp of very small, but positive, duration at the end of the problem.

In any event, restarting a problem requires that the nuclear parameters at the start of the time-dependent chain be the same as those at the end of the problem to be restarted. This generally means repunching a few of the nuclear parameter data cards. In our case the ν 's of material one and the absorption cross sections for groups three and four in material three are different at the end of the problem from their input values, so that six data cards would have to be changed.

To restart at $T = 0$ the same procedure can be followed if desired. However, to obviate the need for punching new cards with the ν 's modified, one can have LAMB = 1, NGES = 0 and ITOUT = 0 on Card No. 2 and include fluxes from the first set of punched fluxes and precursors. This procedure will cause the eigenvalue to be calculated and the ν 's to be multiplied by it before entering the time-dependent chain.

Appendix D - Improvements in the Neutronics Code

Before getting into the non-neutronics (feedback) problem, several improvements and refinements were made in the neutronics code itself. Three of these, to be described below, are fulfillments of the "Further Plans and Suggestions", items #2, 3, and 4, anticipated in WANL-TNR-133.

A. Iteration to Convergence

In the kinetics portion of the code (i.e., beyond the initializing steady-state section), there are two distinct places where iterations are involved. First, the equation for the time derivative of the flux (but not the precursor) density at any point involves derivatives at neighboring points, because of the closely-coupled nature of the variational solution. To use subroutine ICE (Integration with Controlled Error) to integrate these time derivatives, one must first be able to evaluate the derivative at each point separately, hence an iterative procedure is necessary to solve these simultaneous, coupled equations. True, not very many iterations will usually be required, because a good starting guess is available, namely the values of the derivatives at the preceding time point. Nevertheless, sudden changes may occur from time to time, hence the required number of iterations may change. Thus it has been found necessary to require iteration to some pre-specified accuracy, rather than a given number of iterations.

Secondly, for those fast flux groups where the time lag is neglected and the response treated as instantaneous (described as "out-of-ICE" in WANL-TNR-133), a small steady-state problem is solved at each time-step, the "source" being the lower energy flux groups and the precursors. In this

case, the equation for the flux at each point involves not only the fluxes at neighboring space points, but the fluxes in all other energy groups at the same, and neighboring, space points. Precisely the same situation holds as in the preceding paragraph regarding the availability, usually, of a good starting guess, but the uncertainty as to the number of iterations.

In both of these cases, therefore, the code has been changed to allow input of an error criterion, rather than a specified number of iterations. Iteration continues until even the worst point meets the accuracy requirement (E3 and E4).

B. Interpolation

Because of the pre-set "quadratic-in-region" spatial behavior of the solutions, sufficient information to completely specify the solution at any time is given by a knowledge of the value of three points per region (nine points in two-dimensional cases) for each energy and precursor group. However, it is not convenient for the user, e.g., when plotting the solution in space over a large region, to be forced to calculate by hand some intermediate values based on the known quadratic behavior. Provision has therefore been made to input to the code a requirement for any desired number of interpolated points to be printed out in the solution at each time.

This interpolation is done only at each printout time. Thus the code preserves the advantage of working with far fewer points during the bulk of its labor, i.e., the many time-steps between each printout.

C. Increased Computer Space for Feedback

As reported in WANL-TNR-133, one of the major obstacles to the introduction of any very detailed feedback into this code was the amount of core storage taken up by just the neutronics--specifically, all but about 1,000 locations. This large storage requirement resulted from the detailed spatial analysis in this code, compared to most (space-independent) kinetics codes.

To write the feedback equations on a separate computer chain appeared highly undesirable from a standpoint of computing time; with the many time-steps involved in this program, reading a complete program tape in and out at each time-step seemed prohibitive. As an alternative, it was anticipated that the allowable problem size (number of spatial regions, or energy groups, or precursors) might have to be cut down.

However, this alternative has been avoided (or at least postponed depending upon how detailed our feedback eventually becomes) by the following expedient. The entire steady-state part of the program has been deleted from the kinetics program and written as a separate chain. This frees several thousand extra core locations in the kinetics program for feedback equations and storage. At the same time, since the steady-state section is used only once in each problem, for initialization, the increase in computing time is quite negligible.