MIT-4105-6

MITNE-128

# OPTIMIZATION OF MATERIAL DISTRIBUTIONS IN FAST BREEDER REACTORS

by

C. P. Tzanos, E. P. Gyftopoulos, M. J. Driscoll

August, 1971

Department of Nuclear Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Contract AT (30-1) -4105
U.S. Atomic Energy Commission

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF NUCLEAR ENGINEERING
Cambridge, Massachusetts

# OPTIMIZATION OF MATERIAL DISTRIBUTIONS IN FAST BREEDER REACTORS

by

C. P. Tzanos, E. P. Gyftopoulos, M. J. Driscoll

August 1971

MIT-4105-6

MITNE-128

AEC Research and Development Report

UC-34 Physics

Contract AT(30-1)-4105

U. S. Atomic Energy Commission

DISTRIBUTION

MIT-4105-6 MITNE-128

AEC Research and Development Report

UC-34 Physics

U. S. Atomic Energy Commission, Headquarters
Division of Reactor Development + Technology
Reactor Physics Branch (3 copies)

Argonne National Laboratory
Liquid Metal Fast Breeder Reactor Program Office
9700 South Cass Avenue
Argonne, Illinois 60439 (1 copy)

U. S. Atomic Energy Commission
Cambridge Office (2 copies)

Dr. Paul Greebler
General Electric
Atomic Products Division
175 Curtner Ave.
San Jose, California 95125 (1 copy)

Dr. Harry Morewitz
Atomics International
P. O. Box 309
Canoga Park, California 91305 (1 copy)

Mr. M. W. Dyos
Advanced Reactors Division
Westinghouse Electric Corporation
Waltz Mill Site
P. O. Box 158
Madison, Pennsylvania 15663 (1 copy)

Dr. Robert Avery
Argonne National Laboratory
Applied Physics Division
9700 South Cass Avenue
Argonne, Illinois 60439 (1 copy)

Dr. Charles A. Preskitt, Jr.
Mgr. Atomic Nuclear Department
Gulf Radiation Technology
P. O. Box 608
San Diego, California 92112 (1 copy)

ABSTRACT

An iterative optimization method based on linearization and on Linear Programming is developed. The method can be used for the determination of the material distributions in a fast reactor of fixed power output, constrained power density and constrained material volume fractions that maximize or minimize integral reactor parameters which are linear functions of the neutron flux and the material volume fractions.

The method has been applied:

(1) To the problems of optimization of the fuel distribution in the reactor core so as to obtain: (a) a maximum initial breeding gain; (b) a minimum critical mass; and (c) a minimum sodium void reactivity. Numerical results show that the same fuel distribution yields maximum breeding gain, minimum critical mass, minimum sodium void reactivity and uniform power density.
(2) To the problem of optimization of a moderator distribution in the blanket so as to maximize the initial breeding gain. Results indicate that breeding gain is a weak function of the moderator distribution. These results are confirmed by studying the effects on the breeding gain of the insertion of a moderator, homogeneously distributed, in the blanket.

Finally, the effects on the breeding gain of surrounding the blanket by a reflector are investigated. The results show that: (a) savings in blanket thickness may be achieved with choice of a proper reflector without substantial loss in breeding gain; and (b) the transport and absorption properties of a medium, rather than its moderating properties, determine the figure of merit of a fast reactor blanket reflector.

## ACKNOWLEDGEMENTS

This report is based on a thesis submitted by Constantine P. Tzanos to the Department of Nuclear Engineering at Massachusetts Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Science.

Financial support from the U. S. Atomic Energy Commission under contract AT(30-1)-4105 is gratefully acknowledged.

Thanks are due to Barbara Barnes for typing this manuscript.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# INTRODUCTION

## 1.1 THE PROBLEM

The objective of this study is the development and application of a method to optimize the material distributions in a fast reactor of fixed power output, constrained power density and material volume fractions so as to maximize or minimize a given objective function.* An iterative method has been developed based on linearization of the relations describing the system and on Linear Programming. The method can be used to optimize integral reactor quantities which are linear functions of the neutron flux and the material volume fractions.

In what follows, primary emphasis has been placed on the problem of optimization of the fuel distribution in the reactor core and moderator distribution in the reactor blanket so as to obtain a maximum initial breeding gain. In addition, the optimization method has been applied to the problems of optimization of critical mass and sodium void reactivity.

Numerical results show that: (a) the core of maximum initial breeding gain is also the core of minimum critical mass and minimum

---

*The term objective function in this study is used to denote a criterion of optimality.

sodium void reactivity; and (b) the initial breeding gain is a very weak function of the moderator concentration in the blanket.

Fast reactors are of interest primarily because of the economic advantage resulting from their ability to breed more fissile fuel than they consume. It follows that fast reactors should be designed with a breeding potential as high as possible within the framework established by engineering constraints.

A typical fast reactor consists of a core of plutonium-enriched fuel surrounded by a blanket of depleted uranium, which in turn is surrounded by a reflector-shield region. Breeding can be achieved both in the core (internal) and in the blanket (external). In the core, the breeding potential increases monotonically as the spectrum is hardened. Therefore addition of a moderating material in the core is detrimental to internal breeding. In the blanket, however, introduction of a moderating material softens the spectrum and favors captures by the fertile material in the sub-kev energy range. Thus the central question is how should the fuel in the core, and the fertile and moderating materials in the blanket be distributed so that the initial breeding gain is maximized.

In typical demonstration plant and 1000-MWe fast breeder reactor studies, the blanket designs are quite similar. The apparent design strategy is primarily to accommodate as much depleted $UO_2$ as practicable subject to the following constraints. The axial blanket is an extension of the core fuel, and therefore has the same fuel volume fraction; further its thickness is often established by shielding requirements for the protection of core structure, and for this reason is thicker than justified solely by breeding economics. The radial blanket consists of several rows (typically three) of subassemblies having larger diameter rods and a lower coolant volume fraction than the core. The reflector-shield external to the blanket is usually a high-volume-fraction steel region. Thus most of the current work is proceeding within a very narrow envelope of design choices.

Hasnain and Okrent (1) made a preliminary study of the effects of inserting graphite in a fast reactor blanket. They studied four blanket configurations, three of them with graphite, and a reference blanket without graphite. They found a small drop in breeding ratio due to insertion of the graphite, and concluded that inclusion of moderating material in a fast reactor blanket is not promising for a high-power density reactor using optimum fuel cycling.

Perks and Lord (2) studied several blanket configurations containing moderating materials such as graphite, sodium and a graphite-stainless steel mixture. They also found a small drop in breeding ratio for the moderated configurations compared to a reference design without moderating material.

An early blanket design of the British PFR, since dropped, consisted of one row of subassemblies containing a mixture of graphite and steel, one row of subassemblies containing $UO_2$, and two rows of subassemblies containing graphite. In reference (3) it is reported that this arrangement was selected because it leads to a reduction in critical mass and to an improvement in the core radial power form factor. Moreover, it is reported that removal of the moderator improves the breeding gain.

In all the analyses just cited, however, it is not possible to ascertain whether the configuration which gives the maximum breeding is included among the options selected for study.

A primary purpose of the present work is to avoid this deficiency through use of systematic optimization techniques.

## 1.2 THE BREEDING RATIO AND BREEDING GAIN

The breeding ratio and the breeding gain have been defined in a variety of ways. In this section the various definitions of the breeding ratio and breeding gain which have been used in fast reactor studies, and the definition of the breeding gain used in this study are discussed.

The initial (i.e. beginning of life) breeding ratio, b, is usually defined as the ratio of the fissile production rate to the fissile consumption rate. The breeding gain is then defined as production less consumption per unit consumption, or b-1.

In the U.K., the preferred definition of breeding performance of a fast reactor is the breeding gain defined as (3)

Breeding gain = $Pu^{239}$ produced per fission above that required to maintain criticality

Since the plutonium inventory of a fast reactor can arise from sources of plutonium of differing isotopic composition, an "equivalent $Pu^{239}$" quantity is defined as the quantity of $Pu^{239}$ which has the same reactivity worth in fast reactors. For example, for a large ceramic fueled fast reactor the "equivalent $Pu^{239}$" is defined as

$$"Pu^{239}" = Pu^{239} + 1.5PU^{241} + 0.15(Pu^{240} + Pu^{242})$$

In a similar vein, Ott (4) defines the breeding ratio as

$$b_o = \frac{R_c^{238} + \gamma_0 R_c^{239} + \gamma_1 R_c^{240} + \gamma_2 R_c^{241}}{R_a^{239} + \gamma_0 R_a^{240} + \gamma_1 R_a^{241} + \gamma_2 R_a^{242}}$$

i.e., the (spatially integrated) production rate ($R_c$) of the weighted plutonium isotopes over their consumption rate ($R_a$). The weights ($\gamma_i$'s) are defined as

$$\gamma_i = \frac{\bar{N}_i}{\bar{N}_{P_u^{239}}}, \quad i = P_u^{240}, P_u^{241}, P_u^{242}$$

This definition has the advantage that $b_0$ is fairly insensitive to variations in fuel composition.

In this study, the breeding performance of a fast reactor is measured by a breeding gain, defined as the ratio of the net fissile production rate (production rate minus consumption rate) to the thermal power produced. This measure has been selected because: (a) for a power reactor of constant power output, it gives an objective function (breeding gain) for the breeding optimization problem, which is easily linearized about an operating point; and (b) it can be readily used in economic studies, in which power production and plutonium production enter directly as key variables. Because it directly relates the net production of fissile fuel to the power production, which is desirable from the point of view of economic studies, the breeding gain used in the present study could be called the "economist's" breeding gain, as opposed to the "physicist's" or "chemist's" values defined by other authors (5). Compatible with this definition of the total breeding gain, the internal breeding gain is, in turn, defined as the net fissile production in the core per unit total thermal power produced. Similarly the external breeding gain is defined as the net fissile production in the blanket per unit total thermal power produced. These latter definitions of the total, internal and external breeding gain will be used consistently throughout the remainder of this study.

## 1.3 OPTIMIZATION TECHNIQUES

One recurring problem that arises in reactor design, is the selection of the optimum value of a reactor parameter according to a criterion of optimality. Optimization techniques can provide answers to such a problem, since they seek the optimum solution in a systematic way without reliance on intuition or random selection.

In the present work advanced optimization techniques, such as Variational Methods, Dynamic Programming and Linear Programming have been considered. These techniques have previously been used to solve several problems which are more or less related to the present work.

Goertzel (6) solved the problem of optimum fuel distribution in a homogeneous moderator region so as to obtain a thermal reactor of minimum critical mass by using the methods of the classical calculus of variations.

Kochurov (7) solved the same problem with the constraint that the fissile concentration be less than an upper limit, by means of the Maximum Principle of Pontryagin.

Goldschmidt and Quenon (8) used the Maximum Principle of Pontryagin to find the fuel distribution which minimizes the critical mass of a slab geometry fast reactor, described by one-group diffusion theory and subject to the constraints that: (a) the total thermal power be constant; (b) the power density be less than or equal to an upper limit; and (c) the fuel enrichment be bounded.

The Maximum Principle of Pontryagin has also been used by other authors. Zaritskaya and Rudik (9) used it to find the fuel distribution which leads to the minimum critical size of a reactor of given total power and limited power density, and the fuel distribution which gives the maximum total power output of a reactor of known dimensions and bounded maximum flux. Rosztoczy and Weaver (10) used it to determine an optimum reactor shutdown program that minimizes the excess reactivity required to override the xenon poisoning. Finally, Roberts and Smith (11) used it to determine an optimum reactor shutdown program that minimizes the time necessary for shutdown, subject to the constraint that the xenon concentration never exceed the available reactivity override.

Ash (12) used Dynamic Programming to determine an optimal reactor-shutdown program that either minimizes the post-shutdown xenon concentration maximum, or minimizes the xenon concentration itself at a given post-shutdown time.

Wall and Fenech (13) also used Dynamic Programming to optimize the refueling policies of a single-enrichment, three zone PWR core for a minimum unit power cost subject to the constraints that the fuel burnup and power density be bounded.

Gandini, Salvatores and Sena (14) developed a method based on generalized perturbation theory and on Linear Programming to optimize reactor integral parameters, linear or bilinear in the real and adjoint neutron fluxes.

Purica, Pavelescou and Anton (15) developed an algorithm based on game theory, to optimize the dimensions and enrichment of a spherical fast reactor having homogeneous core and blanket and given $U^{238}$ inventory so as to obtain a maximum initial breeding ratio.

A brief review of other optimization studies directly and indirectly related to Nuclear Engineering is given in Appendix A.

For the purposes of this work the Maximum Principle of Pontryagin and Dynamic Programming have been considered for the solution of the breeding optimization problem, but they have not been used. Application of the Maximum Principle of Pontryagin leads to a two-point boundary value problem which is difficult to solve either analytically or numerically. Dynamic Programming, in spite of its conceptual and programming simplicity, imposes exceptionally large fast-access digital computer memory requirements. Instead an iterative method based on linearization of the equations describing the system and on Linear Programming has been developed and successfully applied.

Linear Programming is concerned with the solution of optimization problems for which all relations among the variables are linear both in the constraints and the function to be maximized or minimized (16). Since the problem with which this study is concerned is nonlinear, linearization is used to reduce it to a form suitable for the use of Linear Programming. The linearization procedure and Linear Programming are discussed in Appendix B.

## 1.4 REPORT OUTLINE

This report is organized as follows. In Chapter 2 the theoretical basis of the optimization method used in the study is discussed. In Chapter 3 the method is applied to the optimization of the reactor core. In Chapter 4 the optimization of the reactor blanket is discussed. In Chapter 5 general conclusions and recommendations are discussed. Appendix A contains a brief literature review of publications on theory and applications of optimization methods. In Appendix B Linear Programming and the linearization procedure are discussed. In Appendix C the method of Piecewise Polynomials is briefly discussed and some integral quantities of the piecewise polynomials are evaluated. The computer program written to carry out the computations is discussed and listed in Appendix D.

# Chapter 2

# THE OPTIMIZATION METHOD

As already stated in Section 1.1, the purpose of this study is the development and application of a method for the optimization of the material distributions in a fast reactor of fixed power output, constrained power density and material volume fractions so as to maximize or minimize a given objective function. Without any loss of generality, the method will be developed in this Chapter in connection with the breeding optimization problem. The mathematical statement of this problem is given in Section 2.1, the linearized form of the problem is presented in Section 2.2, the solution of the linearized multigroup diffusion equations is discussed in Section 2.3, the Linear Programming iterative scheme is discussed in Section 2.4, some remarks on the limitations and capabilities of the method are discussed in Section 2.5, and a brief summary of the method is given in Section 2.6.

## 2.1 MATHEMATICAL STATEMENT OF THE PROBLEM

A typical fast reactor consists of a core of plutonium-enriched fuel surrounded by a blanket of depleted uranium, which, in turn, is surrounded by a reflector-shield region as shown schematically in Fig. 2.1. It is a common practice to describe the neutron behavior in a fast reactor by the multigroup diffusion

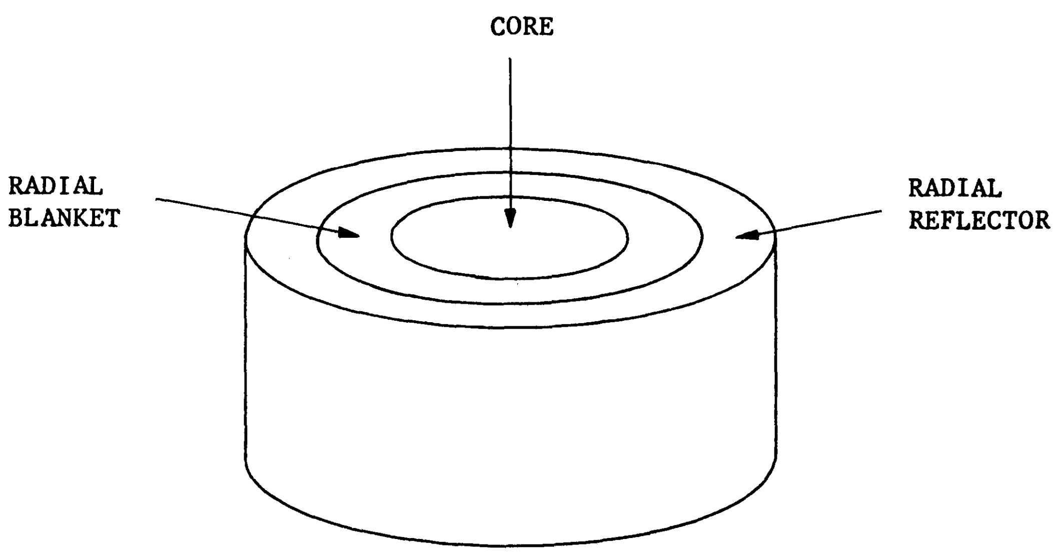



FIG. 2.1 SCHEMATIC REPRESENTATION OF LMFBR CYLINDRICAL GEOMETRY

equations. For an infinite cylindrical geometry the diffusion equation for the i-th group at a point r is written as (17)

$$\nabla D_i(r)\nabla\phi_i(r) - \Sigma_{a,i}(r)\phi_i(r) - \sum_{h=i+1}^{N} \Sigma_{(i-h)}(r)\phi_i(r) +$$

$$\sum_{h=1}^{i-1} \Sigma_{(h-i)}(r)\phi_h(r) + \chi_i \sum_{h=1}^{N} \nu_h \Sigma_{f,h}(r)\phi_h(r) = 0 \qquad (2.1)$$

where

$\phi_i$ = neutron flux in group i

$D_i$ = diffusion coefficient for group i

$\Sigma_{a,i}$ = macroscopic absorption cross section for group i

$\Sigma_{(i-h)}$ = macroscopic down-scattering cross section for transfer from group i to group h by elastic and inelastic scattering

$\chi_i$ = fraction of fission neutrons born into group i

$\nu_h$ = number of neutrons released per fission occuring in group h

$\Sigma_{f,h}$ = macroscopic fission cross section for group h

N = number of neutron groups

The power density P(r) at a point r is given by the relation

$$P(r) = \sum_{i=1}^{N} \{u_f(r)\Sigma_{f,i}^{fs} + [N_0 - u_f(r) - u_m(r)]\Sigma_{f,i}^{fr}\}\phi_i(r) \qquad (2.2)$$

where

$u_f(r)$ = volume fraction of the fissile material

$u_m(r)$ = volume fraction of the moderating material

$\Sigma_{f,i}^{fs}$ = macroscopic fission cross section of pure fissile material for group i

$\Sigma_{f,i}^{fr}$ = macroscopic fission cross section of pure fertile material for group i

$N_0$ = fissile volume fraction + fertile volume fraction + moderator volume fraction

The total thermal power W delivered by the reactor is

$$W = 2\pi \int_0^{t_f} \sum_{i=1}^{N} \{u_f(r)\Sigma_{f,i}^{fs} + [N_0 - u_f(r) - u_m(r)]\Sigma_{f,i}^{fr}\} \phi_i(r) r dr \tag{2.3}$$

where

$t_f$ = outer reactor radius

The breeding gain as defined in Section 1.2 is written as

$$BG = \frac{2\pi \int_0^{t_f} \sum_{i=1}^{N} \{[N_0 - u_f(r) - u_m(r)]\Sigma_{\gamma,i}^{fr} - \Sigma_{a,i}^{fs} u_f(r)\}\phi_i(r) r dr}{W} \tag{2.4}$$

where

$\Sigma_{\gamma,i}^{fr}$ = macroscopic capture cross section of pure fertile material for group i

$\Sigma_{a,i}^{fs}$ = macroscopic absorption cross section of pure fissile material for group i

In terms of the mathematical relations just cited the breeding optimization problem is stated as follows: Find the optimum fissile and moderator distributions, $u_f(r)$ and $u_m(r)$ respectively, which maximize the breeding gain BG (Eq. 2.4) while the following equations and inequalities are satisfied:

1. Multigroup diffusion equations (Eq. 2.1)
2. The power density

$$P(r) \leq p = \text{const.} \tag{2.5}$$

3. The total thermal power

$$W = \text{const.} \tag{2.6}$$

4. The sum of fissile and moderator volume fractions

$$u_m + u_f \leq N_0 = \text{const.} \tag{2.7}$$

## 2.2 THE LINEARIZED FORM OF THE BREEDING OPTIMIZATION PROBLEM

It is seen from Eqs. (2.1), (2.2), (2.3) and (2.4) that the optimization problem of interest is nonlinear. As already mentioned in Section 1.3 it is very difficult to solve such a problem explicitly or numerically through use of nonlinear optimization methods. For this reason computer aided solutions have been sought through use of appropriate mathematical programming techniques. One of these techniques is Linear Programming which has the advantages of simplicity and availability of standard computer subroutines.

Linear Programming is a method for maximizing (minimizing) a linear objective function for a system with linear algebraic constraints. For a nonlinear problem, linearization can be used to reduce the problem into a form suitable for use of Linear Programming.

Application of the linearization procedure discussed in Appendix B to Eqs. (2.1), (2.2), (2.3) and (2.4) results in the following linearized form of these relations.

1. Linearized breeding gain

$$BG = \frac{2\pi}{W} \{ - \int_0^{t_f} u_f(r) \sum_{i=1}^{N} (\Sigma_{\gamma,i}^{fr} + \Sigma_{a,i}^{fs}) \phi_i^0(r) r dr -$$

$$\int_0^{t_f} u_m(r) \sum_{i=1}^{N} \Sigma_{\gamma,i}^{fr} \phi_i^0(r) r dr +$$

$$\int_0^{t_f} \sum_{i=1}^{N} [(N_0 - u_f^0(r) - u_m^0(r)) \Sigma_{\gamma,i}^{fr} - u_f^0(r) \Sigma_{a,i}^{fs}] \phi_i^*(r) r dr +$$

$$\int_0^{t_f} N_0 \sum_{i=1}^{N} \Sigma_{\gamma,i}^{fr} \phi_i^0(r) r dr \} \qquad (2.8)$$

where the superscript 0 is used to denote quantities evaluated at the operating point about which the relations describing the system are linearized, and

$$\phi_i^*(r) = \phi_i(r) - \phi_i^0(r) \qquad (2.9)$$

2. Linearized multigroup diffusion equations

$$\frac{1}{r}\frac{d}{dr}[rD_i^0(r)\frac{d}{dr}\phi_i^*(r)] - \Sigma_{a,i}^0(r)\phi_i^*(r) - \sum_{h=i+1}^{N}\Sigma_{(i-h)}^0(r)\phi_i^*(r) +$$

$$\sum_{h=1}^{i-1}\Sigma_{(h-i)}^0(r)\phi_h^*(r) + \chi_i\sum_{h=1}^{N}\nu_h\Sigma_{f,h}^0(r)\phi_h^* +$$

$$[u_f(r)-u_f^0(r)]\{-[\Sigma_{a,i}^{fs}-\Sigma_{a,i}^{fr}]\phi_i^0(r) - \sum_{h=i+1}^{N}[\Sigma_{(i-h)}^{fs}-\Sigma_{(i-h)}^{fr}]\phi_i^0(r) +$$

$$\sum_{h=1}^{i-1}[\Sigma_{(h-i)}^{fs}-\Sigma_{(h-i)}^{fr}]\phi_h^0(r) + \chi_i\sum_{h=1}^{N}[\nu_h^{fs}\Sigma_{f,h}^{fs}-\nu_h^{fr}\Sigma_{f,h}^{fr}]\phi_h^0(r) -$$

$$\frac{\Sigma_{tr,i}^{fs}-\Sigma_{tr,i}^{fr}}{3[\Sigma_{tr,i}^0(r)]^2}\frac{1}{r}\frac{d}{dr}[r\frac{d\phi_i^0(r)}{dr}]\} + [u_m(r)-u_m^0(r)]$$

$$\{-[\Sigma_{a,i}^{m}-\Sigma_{a,i}^{fr}]\phi_i^0(r) - \sum_{h=i+1}^{N}[\Sigma_{(i-h)}^{m}-\Sigma_{(i-h)}^{fr}]\phi_i^0(r) +$$

$$\sum_{h=1}^{i-1}[\Sigma_{(h-i)}^{m}-\Sigma_{(h-i)}^{fr}]\phi_h^0(r) + \chi_i\sum_{h=1}^{N}[-\nu_h^{fr}\Sigma_{f,h}^{fr}]\phi_h^0(r) -$$

$$\frac{\Sigma_{tr,i}^{m}-\Sigma_{tr,i}^{fr}}{3[\Sigma_{tr,i}^0(r)]^2}\frac{1}{r}\frac{d}{dr}[r\frac{d\phi_i^0(r)}{dr}]\} = 0 \qquad (2.10)$$

where

$\Sigma_{tr,i}$ = macroscopic transport cross section for group i

The superscript m is used to denote properties of the moderating material.

3. Linearized total thermal power

$$W = \int_0^{t_f} u_f(r) \sum_{i=1}^{N} [\Sigma_{f,i}^{fs}-\Sigma_{f,i}^{fr}]\phi_i^0(r)rdr - \int_0^{t_f} u_m(r) \sum_{i=1}^{N} \Sigma_{f,i}^{fr}\phi_i^0(r)rdr +$$

$$\int_0^{t_f} \sum_{i=1}^{N} \Sigma_{f,i}^0(r)\phi_i^*(r)rdr + \int_0^{t_f} \sum_{i=1}^{N} N_0\Sigma_{f,i}^{fr}\phi_i^0(r)dr \qquad (2.11)$$

4. Linearized power density

$$P(r) = u_f(r) \sum_{i=1}^{N} [\Sigma_{f,i}^{fs}-\Sigma_{f,i}^{fr}]\phi_i^0(r) - u_m(r) \sum_{i=1}^{N} \Sigma_{f,i}^{fr}\phi_i^0(r) +$$

$$\sum_{i=1}^{N} \Sigma_{f,i}^0\phi_i^*(r) + \sum_{i=1}^{N} N_0\Sigma_{f,i}^{fr}\phi_i^0(r) \qquad (2.12)$$

When the multigroup diffusion equations are solved to obtain the neutron flux in a reactor, the criticality condition is imposed by the requirement that the eigenvalue of the multigroup diffusion equations be equal to 1. In this study, as explained later in this chapter, the linearized multigroup diffusion equations are used to express $\phi_i^*$ as a function of $u_f$ and $u_m$. For the reactor to remain critical $u_f$ and $u_m$ can not change in an arbitrary way. Perturbation theory can be used to express the criticality condition in the form (18)

$$\int_0^{t_f} -[u_f(r)-u_f^0(r)] \sum_{i=1}^{N} \frac{\Sigma_{tr,i}^{fs}-\Sigma_{tr,i}^{fr}}{3[\Sigma_{tr,i}^0(r)]^2} \nabla\phi_i^0(r)\nabla\psi_i^0(r)rdr +$$

$$\int_0^{t_f} [u_f(r)-u_f^0(r)] \sum_{i=1}^{N} [\Sigma_{a,i}^{fs}-\Sigma_{a,i}^{fr}] \phi_i^0(r)\psi_i^0(r)rdr +$$

$$\int_0^{t_f} [u_f(r)-u_f^0(r)] \sum_{h=1}^{N} \sum_{h=i+1}^{N} [\Sigma_{(i-h)}^{fs}-\Sigma_{(i-h)}^{fr}]\phi_i^0(r)[\psi_i^0(r)-\psi_h^0(r)]rdr -$$

$$\frac{1}{k}\int_0^{t_f} [u_f(r)-u_f^0(r)] \sum_{i=1}^{N} \sum_{h=1}^{N} [\nu^{fs}\Sigma_{f,h}^{fs}-\nu^{fr}\Sigma_{f,h}^{fr}]\chi_i\phi_h^0(r)\psi_i^0(r)rdr -$$

$$\int_0^{t_f} [u_m(r)-u_m^0(r)] \sum_{i=1}^{N} \frac{\Sigma_{tr,i}^{m}-\Sigma_{tr,i}^{fr}}{3[\Sigma_{tr,i}^0(r)]^2} \nabla\phi_i^0(r)\nabla\psi_i^0(r)rdr +$$

$$\int_0^{t_f} [u_m(r)-u_m^0(r)] \sum_{i=1}^{N} [\Sigma_{a,i}^{m}-\Sigma_{a,i}^{fr}]\phi_i^0(r)\psi_i^0(r)rdr +$$

$$\int_0^{t_f} [u_m(r)-u_m^0(r)] \sum_{i=1}^{N} \sum_{h=i+1}^{N} [\Sigma_{(i-h)}^{m}-\Sigma_{(i-h)}^{fr}]\phi_i^0(r)[\psi_i^0(r)-\psi_h^0(r)]rdr -$$

$$\frac{1}{k}\int_0^{t_f} [u_m(r)-u_m^0(r)] \sum_{i=1}^{N} \sum_{h=1}^{N} [-\nu^{fr}\Sigma_{f,h}^{fr}]\chi_i\phi_h^0(r)\psi_i^0(r)rdr = 0 \quad (2.13)$$

where

$\psi_i$ = adjoint flux for group i

k = k-effective

In terms of the linearized relations just cited the breeding optimization problem is stated as follows: Determine the optimum fissile and moderator distributions $u_f(r)$ and $u_m(r)$ respectively, which maximize the breeding gain BG (Eq. 2.8) while the following relations are satisfied:

1. Linearized multigroup diffusion equations (Eqs. 2.10)
2. The total thermal power

$$W = \text{const.} \tag{2.14}$$

3. The power density

$$P(r) \leq p = \text{const.} \tag{2.15}$$

4. Criticality condition as expressed by Eq. (2.13)
5. $$0 \leq u_f,\ 0 \leq u_m,\ u_m + u_f \leq N_0 = \text{const.} \tag{2.16}$$

Even after the linearization the optimization problem does not yet have the proper form for application of Linear Programming. Such a form, however, can be obtained as follows: (a) the reactor is divided into a number, R, of regions, each with spatially uniform material concentrations; and (b) the linearized multigroup diffusion equations are solved to express each $\phi_i^*$ (i=1,N) as a function of $u_{f,j}$, $u_{m,j}$ (j=1,R). Thus, the functional to be maximized and the constraints of the problem become linear algebraic functions of $u_{f,j}$ and $u_{m,j}$ and therefore suitable for application of Linear Programming.

## 2.3 SOLUTION OF THE LINEARIZED MULTIGROUP DIFFUSION EQUATIONS

The linearized multigroup diffusion equations are of the form

$$\underline{L}\ \underline{\phi}^{*} = \underline{f}(u_f^{*},\ u_m^{*}) \tag{2.17}$$

where $\underline{L}$ is the multigroup diffusion matrix operator and

$$u_f^{*} = u_f - u_f^{0},\ u_m^{*} = u_m - u_m^{0} \tag{2.18}$$

We want to express $\underline{\phi}^{*}$ as a function of $u_f^{*}$ and $u_m^{*}$. Application of the finite difference technique gives a set of algebraic equations of the form

$$\underline{M}\ \underline{\phi}^{*} = \underline{f}(\underline{u}_f^{*}, \underline{u}_m^{*}) \tag{2.19}$$

Equations (2.19) can be solved by inversion of the matrix $\underline{M}$. On the other hand even for 5 neutron groups and 100 mesh points $\underline{M}$ is a large (500 x 500) matrix and its inversion requires excessive computer time and gives rise to prohibitive round-off errors.

This difficulty can be avoided by use of the method of Piecewise Polynomials, discussed by Kang (19). A brief description of this method is given in Appendix C. The method of Piecewise Polynomials can be applied to solve the linearized multigroup diffusion equations as follows. The reactor is divided into a number n of mesh points and the flux difference $\phi_i^{*}$ (Eq. 2.9) is approximated by

$$\phi_i^* \simeq \Phi_i^* = \sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_{k,i} \qquad (2.20)$$

where $w_k$ and $v_{k,i}$ are cubic piecewise polynomials (Appendix C). The coefficients $a_{k,i}$ and $\beta_{k,i}$ are determined by requiring

$$\int_V (L_i \Phi_i^*) w_k dV = \int_V f_i(u_f^*, u_m^*) \; w_k dV \qquad (2.21)$$

$$\int_V (L_i \Phi_i^*) v_{k,i} dV = \int_V f_i(u_f^*, u_m^*) \; v_{k,i} dV \qquad (2.22)$$

where

V = reactor volume

The integrations on the right hand side of Eqs. (2.21) and (2.22) can not be carried out since the space dependence of $u_f^*$ and $u_m^*$ is unknown. On the other hand if the reactor is divided into a number, R, of regions with spatially uniform material concentrations in each region, then the right hand side of Eqs. (2.21) and (2.22) can be integrated and a system of algebraic equations results. These equations are of the form

$$\underline{A} \; \underline{a} = \underline{g}(\underline{u}_f^*, \underline{u}_m^*, a_{11}), \qquad (2.23)$$

where $a_{11}$ is the coefficient of the polynomial $w_1$ in Eq. (2.20) for i=1, and the components of the vectors $\underline{u}_f^*$, $\underline{u}_m^*$ are given by

$$u^*_{f,j} = u_{f,j} - u^0_{f,j},\ u^*_{m,j} = u_{m,j} - u^0_{m,j}\quad j=1, R \tag{2.24}$$

The solution of the system of Eqs. (2.23) is

$$\underline{a} = \underline{A}^{-1}\,\underline{g} \tag{2.25}$$

For n mesh intervals and N neutron groups the order of the matrix $\underline{A}$ is equal to 2nN-1. The method of piecewise polynomials, compared to the finite difference technique, gives a very good approximation to $\phi^*_i$ with only a few mesh intervals, n. Since the order of matrix $\underline{A}$ is a function of the number of mesh intervals, n, the method of piecewise polynomials gives a smaller matrix $\underline{A}$ than the finite difference technique for the same accuracy in $\phi^*_i$. Thus for N = 5 and n = 10 the order of $\underline{A}$ is 2 x 10 x 5 - 1 = 99. For the same accuracy in $\phi^*_i$ the finite difference technique gives a 500 x 500 matrix. The inversion of a 99 x 99 matrix is much more advantageous than the inversion of a 500 x 500 matrix from the standpoint of computation time and round-off errors.

## 2.4 THE ITERATIVE SCHEME

The solution of the linearized multigroup diffusion equations results in all constraints and the objective function of the problem being linear algebraic relations of $u_{f,j}$ and $u_{m,j}$ (j = 1,R). This means that the original nonlinear optimization problem has been reduced to a Linear Programming optimization problem.

The linearized form of the breeding optimization problem is a good approximation of the original nonlinear problem only if $u_{f,j}$ and $u_{m,j}$ are sufficiently close to $u^0_{f,j}$ and $u^0_{m,j}$ about which linearization took place. Therefore Linear Programming can be applied to obtain the optimum values of $u_{f,j}$ and $u_{m,j}$ which maximize the objective function while $u_{f,j}$ and $u_{m,j}$ must satisfy the additional constraints

$$u^0_{f,j} - \varepsilon_f \leq u_{f,j} \leq u^0_{f,j} + \varepsilon_f, \quad u^0_{m,j} - \varepsilon_m \leq u_{m,j} \leq u^0_{m,j} + \varepsilon_m,$$

$$(j = 1,R) \qquad (2.26)$$

The parameters $\varepsilon_f$, $\varepsilon_m$ are constants such that $u_{f,j}$ and $u_{m,j}$ remain close enough to $u^0_{f,j}$ and $u^0_{m,j}$ respectively.

This procedure results in a suboptimum solution since $u_{f,j}$ and $u_{m,j}$ are restricted by Eqs. (2.26) to only small variations around $u^0_{f,j}$ and $u^0_{m,j}$. To advance the solution the following iterative scheme is devised. If $u^{(1)}_{f,j}$ and $u^{(1)}_{m,j}$ is the solution given by Linear Programming, the problem is re-linearized about $u^{(1)}_{f,j}$, $u^{(1)}_{m,j}$ and Linear Programming is again applied, while the relations

$$u^{(1)}_{f,j} - \varepsilon_f \leq u_{f,j} \leq u^{(1)}_{f,j} + \varepsilon_f, \quad u^{(1)}_{m,j} - \varepsilon_m \leq u_{m,j} \leq u^{(1)}_{m,j} + \varepsilon_m,$$

$$(j = 1,R) \qquad (2.27)$$

must be satisfied, to obtain another solution $u_{f,j}^{(2)}$, $u_{m,j}^{(2)}$.

This procedure of linearization about the previous solution of Linear Programming and re-application of Linear Programming is repeated until no further improvement of the objective function is achieved. The last Linear Programming solution gives the optimum fissile and moderator distributions which result in the maximum value of the objective function. It must be pointed out that there is no assurance that the determined optimum is a local or a global one. Therefore one should repeat the iterative procedure starting with different initial fissile and moderator distributions and compare the determined optima.

## 2.5 REMARKS

The discussion in this chapter was based on infinite cylindrical geometry. In principle, the optimization method developed can be extended to any reactor geometry. For geometries, however, involving more than one dimension the method becomes very complicated in terms of its numerical implementation.

From among the possible one-dimensional geometries infinite cylindrical geometry has been selected because: (a) cylindrical geometry is, almost without exception, characteristic of practical reactors; and (b) the optimization of the fuel and/or a moderator distribution is likewise of practical importance primarily in the radial direction. Nevertheless, the method can be applied equally well to any one-dimensional geometry.

In addition, it should be noted that many two-dimensional calculations in cylindrical geometry are approximated by one-dimensional calculations by adding to the macroscopic absorption cross section a $DB^2$ term to account for axial leakage (20). This approximation can be incorporated in the optimization method discussed in this chapter by simply adding an appropriate $DB^2$ term to the macroscopic absorption cross section.

## 2.6 SUMMARY

In this chapter the theoretical development of an iterative optimization method has been discussed. Each iteration consists of three steps: (a) the relations describing the system are linearized about the previous Linear Programming solution; (b) the linearized multigroup diffusion equations are solved to express $\phi_i^*$ as a function of $\underline{u}_f$ and $\underline{u}_m$; and (c) Linear Programming is applied. The iterations continue until no further improvement of the objective function is achieved.

Results obtained from the numerical application of the method to the problems of Breeding Optimization, Critical Mass Optimization and Sodium Void Reactivity Optimization are presented in Chapters 3 and 4. The computer program written to carry out the operations described in this chapter is discussed and listed in Appendix D.

# Chapter 3

# CORE OPTIMIZATION

## 3.1 INTRODUCTION

The optimization method discussed in Chapter 2 has been applied to the core of a 1500 MW(th) fast breeder to obtain the fuel distribution that: (a) maximizes the initial breeding gain; (b) minimizes the critical mass; and (c) minimizes the sodium void reactivity. The results are presented in this chapter.

For these studies, an infinite cylindrical geometry reactor is considered. The core is divided into four regions of equal volume. As explained later the optimization procedure involves two reactors of different dimensions. They are designated reactor No. 1 and reactor No. 2. The dimensions of reactor No. 1 are given in Table 3.1. The dimensions of reactor No. 2 are given later. The composition of reactors No. 1 and No. 2 is given in Table 3.2. This composition is representative of LMFBR design studies presented over the last several years (21,22).

The sum of the $PuO_2$ and $UO_2$ volume fractions is constrained to remain constant during optimization and equal to 0.35.

Although for the neutronic calculations an infinite reactor height has been considered, the power of 1500 MW(th) is attributed to a fictitious core length equal to 100 cm.

A value of 550 $w/cm^3$ is used as an upper limit for the power

TABLE 3.1

Dimensions of Reactor No. 1

| Region | | Inner Radius | Outer Radius |
|---|---|---|---|
| Core | 1 | 0.00 cm | 62.64 cm |
| | 2 | 62.64 cm | 90.48 cm |
| | 3 | 90.48 cm | 111.36 cm |
| | 4 | 111.36 cm | 128.76 cm |
| Radial Blanket | 5 | 128.76 cm | 174.00 cm* |

*Extrapolated outer boundary

TABLE 3.2

Reactor Composition

| Material | Core | Blanket | Atomic or Molecular density (for pure materials) $cm^{-3} \times 10^{-24}$ |
|---|---|---|---|
| Na | 50 v/o | 50 v/o | 0.025410 |
| Fe | 15 v/o | 15 v/o | 0.084870 |
| $PuO_2$ | 35 v/o (PuO$_2$ + UO$_2$) | -- | 0.025189 |
| $UO_2$ | | 35 v/o | 0.024444 |

density. This is representative of typical LMFBR design studies (21,22).

For computational convenience the total thermal power has been normalized to 100 and the power density limit to a corresponding value:

$$p = \frac{P \times 2\pi H \times W_n \times 100}{W} \quad \frac{w}{cm^3} \times cm \times \frac{w}{w} = 2.30267 \frac{w}{cm^2} \qquad (3.1)$$

where

P = power density upper limit = 550 $w/cm^3$

H = reactor height = 100 cm

$W_n$ = normalized total power = 100 w

W = total thermal power = $1500 \times 10^6$ w

For the neutronic calculations five neutron groups were used. In principle any number of neutron groups and reactor regions can be employed. The choice is governed by the size of the matrix $\underline{A}$ (Chapter 2).

The ANISN multigroup transport theory code was used to obtain a five-group cross section set by collapsing a sixteen-group modified Hansen-Roach cross section set (23). The five-group structure is shown in Table 3.3.

The three problems of Breeding Optimization, Critical Mass Optimization and Sodium Void Reactivity Optimization are described by the same equations except for the objective function.

## 3.2 BREEDING OPTIMIZATION

The purpose of this section is to present the results obtained for the Breeding Optimization Problem. In Table 3.4, the results

TABLE 3.3

Five-Group Cross Section Set Structure

| Group | Neutron Energy in Mev |
|---|---|
| 1 | 1.400 - ∞ |
| 2 | 0.400-1.400 |
| 3 | 0.100-0.400 |
| 4 | 0.017-0.100 |
| 5 | 0.000-0.017 |

obtained in the successive iterations of the iterative optimization method, from the starting configuration* to the optimum one, are presented. As discussed in Section 2.6 each iteration consists of three steps: (a) the relations describing the system are linearized about the previous Linear Programming solution; (b) the linearized multigroup diffusion equations are solved to express $\phi_i^*$ as a function of $\underline{u}_f$ and $\underline{u}_m$; and (c) Linear Programming is applied. The computation begins with a four region homogeneous core as given by the first row of Table 3.4. The optimum configuration is given by the last row of the same table. The breeding gain listed in the last column of the table is calculated by the relation

$$BG = \frac{2\pi \int_o^{t_f} \sum_{i=1}^{N} [(N_0-u_f)\, \Sigma_{\gamma,i}^{fr} - \Sigma_{a,i}^{fs}\, u_f]\, \phi_i \, rdr}{2\pi \int_0^{t_f} \sum_{i=1}^{N} \Sigma_{f,i}\, \phi_i \, rdr} \qquad (3.2)$$

The peaks of the power density in each core region (which occur at the inner radius of each region) for the initial and optimum configurations are shown in Table 3.5

---

*The term configuration in this study is used to denote a reactor's material composition: in all cases the geometry and size of all regions is fixed.

TABLE 3.4

Fissile Composition and Breeding Gain as a Function of Linear Programming Iteration Number for Reactor No. 1

| Iteration Number | Region 1 ($PuO_2$ v/o) | Region 2 | Region 3 | Region 4 | Breeding Gain* |
|---|---|---|---|---|---|
| 1 | 3.41200 | 3.41200 | 3.41200 | 3.41200 | 0.576527 |
| 2 | 3.40670 | 3.53833 | 3.21200 | 3.21200 | 0.578265 |
| 3 | 3.38110 | 3.69036 | 3.01200 | 3.01200 | 0.579931 |
| 4 | 3.35800 | 3.82934 | 2.81200 | 2.81200 | 0.581669 |
| 5 | 3.33607 | 3.95874 | 2.61200 | 2.61200 | 0.583506 |
| 6 | 3.31556 | 4.07905 | 2.41200 | 2.41200 | 0.585427 |
| 7 | 3.29832 | 4.17795 | 2.24362 | 2.21200 | 0.587314 |
| 8 | 3.29680 | 4.16995 | 2.32654 | 2.01200 | 0.588124 |
| 9 | 3.29543 | 4.16177 | 2.40826 | 1.81200 | 0.588952 |
| 10 | 3.29407 | 4.15375 | 2.48842 | 1.61200 | 0.589804 |
| 11 | 3.29277 | 4.14585 | 2.56699 | 1.41200 | 0.590672 |
| 12 | 3.29146 | 4.13812 | 2.64417 | 1.21200 | 0.591559 |
| 13 | 3.29017 | 4.13053 | 2.71992 | 1.01200 | 0.592458 |
| 14 | 3.28885 | 4.12313 | 2.79443 | 0.81200 | 0.593391 |
| 15 | 3.28765 | 4.11576 | 2.86731 | 0.61200 | 0.594337 |
| 16 | 3.28642 | 4.10857 | 2.93906 | 0.41200 | 0.595300 |
| 17 | 3.28521 | 4.10151 | 3.00954 | 0.21200 | 0.596284 |
| 18 | 3.28402 | 4.09457 | 3.07881 | 0.01200 | 0.597285 |
| 19 | 3.27854 | 4.09062 | 3.03854 | 0.11200 | 0.600014 |
| 20 | 3.27801 | 4.08658 | 3.07689 | 0.00000 | 0.600585 |
| 21 | 3.27801 | 4.08662 | 3.07676 | 0.00000 | 0.600585 |

*Net production of $Pu^{239}$ atoms per fission

TABLE 3.5

Peak Power Densities for Reactor No. 1

| Region | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Initial Configuration | 2.23971 | 1.68232 | 1.15895 | 0.72096 |
| Optimum Configuration | 2.30265 | 2.30264 | 1.14762 | 0.07654 |

Since, as mentioned in Section 2.4, there is no assurance that the determined optimum is a local or a global one, one should repeat the computations with different starting configurations. Table 3.6 shows the results obtained using a different starting configuration. The optimum configuration shown in Table 3.6 is the same as that presented in Table 3.4.

From the results given in Tables 3.4 and 3.5 it is concluded that for the five region reactor with dimensions as given by Table 3.1 (reactor No. 1) the optimum configuration is one for which there is no $PuO_2$ in the fourth region, and the peaks of the power density in regions 1 and 2 are equal to the upper power density limit. The breeding gain of the optimum configuration is 4.08% larger than the breeding gain of the initial homogeneous configuration.

The optimization started with a reactor of four core regions and a 45.24 cm blanket. The optimum configuration consists of three core regions and a 62.64 cm blanket ($PuO_2$ was removed from the 4th core region of the initial configuration). If it were possible to

TABLE 3.6

Fissile Composition and Breeding Gain as a Function of Linear Programming Iteration Number for Reactor No. 1 and a different Starting Configuration

| Iter-ation Number | Region $PuO_2$ v/o | | | | Breeding Gain* |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1 | 3.41200 | 2.95400 | 4.32986 | 3.41200 | 0.571885 |
| 2 | 3.51200 | 2.87773 | 4.22986 | 3.31200 | 0.571959 |
| 3 | 3.49645 | 2.97773 | 4.13002 | 3.21200 | 0.572709 |
| 4 | 3.48061 | 3.07483 | 4.03002 | 3.11200 | 0.573490 |
| 5 | 3.46548 | 3.16738 | 3.93002 | 3.01200 | 0.574320 |
| 6 | 3.45102 | 3.25574 | 3.83002 | 2.91200 | 0.575160 |
| 7 | 3.43694 | 3.34062 | 3.73002 | 2.81200 | 0.576032 |
| 8 | 3.42342 | 3.42190 | 3.63002 | 2.71200 | 0.576907 |
| 9 | 3.41022 | 3.50079 | 3.53002 | 2.61200 | 0.577816 |
| 10 | 3.39757 | 3.57527 | 3.43002 | 2.51200 | 0.578767 |
| 11 | 3.38544 | 3.64733 | 3.33002 | 2.41200 | 0.579714 |
| 12 | 3.37364 | 3.71684 | 3.23002 | 2.31200 | 0.580675 |
| 13 | 3.36216 | 3.78394 | 3.13002 | 2.21200 | 0.581652 |
| 14 | 3.35105 | 3.84866 | 3.03002 | 2.11200 | 0.582644 |
| 15 | 3.34030 | 3.91116 | 2.93002 | 2.01200 | 0.583655 |
| 16 | 3.32991 | 3.97149 | 2.83002 | 1.91200 | 0.584673 |
| 17 | 3.31979 | 4.02992 | 2.73002 | 1.81200 | 0.585703 |
| 18 | 3.29200 | 4.08646 | 2.63002 | 1.71200 | 0.591873 |
| 19 | 3.28789 | 4.14161 | 2.52161 | 1.51200 | 0.593464 |
| 20 | 3.28602 | 4.13460 | 2.60000 | 1.31200 | 0.594340 |
| 21 | 3.28474 | 4.13692 | 2.67641 | 1.11200 | 0.595240 |
| 22 | 3.28346 | 4.11940 | 2.75148 | 0.91200 | 0.596150 |
| 23 | 3.28215 | 4.11205 | 2.82532 | 0.71200 | 0.597095 |
| 24 | 3.28097 | 4.10475 | 2.89756 | 0.51200 | 0.598052 |
| 25 | 3.27974 | 4.09763 | 2.96867 | 0.31200 | 0.599028 |
| 26 | 3.27854 | 4.09062 | 3.03854 | 0.11200 | 0.600023 |
| 27 | 3.27798 | 4.08669 | 3.07674 | 0.00000 | 0.600594 |
| 28 | 3.27808 | 4.08657 | 3.07660 | 0.00000 | 0.600594 |

*Net production of $Pu^{239}$ atoms per fission apply the optimization method to a reactor with a core divided into an arbitrarily large number of regions, the optimum configuration would apparently approach the optimum configuration obtained by an analytical solution of the problem asymptotically as the number of core regions increased. This suggests that a configuration having a further improvement in breeding gain can be obtained by redivision of the core into four regions and reapplication of the optimization procedure. Thus the core of the optimum reactor No. 1 was redivided into four regions of equal volume. Since a typical fast reactor blanket is about 45 cm thick (21,22), the extra blanket was also removed. The dimensions of the new reactor, which will be called reactor No. 2 in the remainder or this study, are shown in Table 3.7. The composition and the peak power densities of the optimum configuration of reactor No. 2 are shown in Table 3.8. The breeding gain of the optimum configuration is equal to 0.582528. As shown in Table 3.8, the peak power densities in the first three core regions of the optimum configuration are all equal to the upper power density limit.

The breeding gain of the optimum configuration of reactor No. 2 is slightly smaller than the breeding gain of the optimum configuration of reactor No. 1. This is due to the fact that reactor No. 2 is smaller than reactor No. 1 and consequently loses more neutrons by leakage. Reduction of the leakage can be achieved by surrounding the blanket by a reflector. The breeding gains of the initial homogeneous version of reactor No. 2, the optimum configuration of reactor No. 1, and the optimum configuration of reactor No. 2,

TABLE 3.7

Dimensions of Reactor No. 2

| Region | | Inner Radius | Outer Radius |
|---|---|---|---|
| Core | 1 | 0.00 cm | 55.68 cm |
| | 2 | 55.68 cm | 80.04 cm |
| | 3 | 80.04 cm | 97.44 cm |
| | 4 | 97.44 cm | 111.36 cm |
| Radial Blanket | 5 | 111.36 cm | 156.60 cm* |

TABLE 3.8

Optimum Configuration of Reactor No. 2

| Region | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $PuO_2$ v/o | 3.23751 | 3.72338 | 5.01528 | 0.50175 |
| Peak Power Density | 2.30267 | 2.30267 | 2.30267 | 0.29742 |

*Extrapolated outer boundary

before and after the addition of a 45.24 cm BeO reflector at the outer periphery of the blanket, are shown in Table 3.9. The optimum reactor No. 2 now has a higher total breeding gain than the homogeneous reactor No. 1 and the optimum reactor No. 1, although it has a core about 25% smaller than the homogeneous reactor No. 1.

Table 3.9 also shows that the addition of the reflector considerably improves the external breeding gain while its effect on the internal breeding gain is very small. An extensive discussion of the effect of the reflector on breeding is given in Chapter 4.

## 3.3 CRITICAL MASS OPTIMIZATION

In this section the results obtained from the Critical Mass Optimization Problem are discussed.

The results obtained by the successive iterations of the iterative optimization method from the starting configuration to the optimum one, are shown in Table 3.10. The computation starts with the homogeneous reactor No. 1. The optimum configuration is given by the last row of the same table. The critical mass listed in the last column of the table is calculated by the relation

$$M_c = \frac{A \times M^{Pu}}{N_A} \int_0^{t_f} 2\pi r \, u_f(r) dr \qquad (3.3)$$

where

$A$ = atom density of Pu in $PuO_2$

$M^{Pu}$ = atomic weight of Pu

$N_A$ = Avogadro's number

TABLE 3.9

Effect of Blanket Reflector on Breeding Gain

| Reactor | Breeding Gain of Unreflected Reactor | | | Breeding Gain after addition of BeO Reflector* | | |
|---|---|---|---|---|---|---|
| | Internal | External | Total | Internal | External | Total |
| Homogeneous No. 1 | 0.405686 | 0.170841 | 0.576527 | 0.405832 | 0.202875 | 0.608707 |
| Optimum No. 1 | 0.345045 | 0.255540 | 0.600585 | 0.345059 | 0.270237 | 0.615296 |
| Optimum No. 2 | 0.377648 | 0.204880 | 0.582528 | 0.378024 | 0.239341 | 0.616365 |

* 45.24 cm BeO Reflector

TABLE 3.10

Fissile Composition and Critical Mass as a Function of Linear Programming Iteration Number for Reactor No. 1

| Iteration Number | Region 1 | Region 2 | Region 3 | Region 4 | Critical Mass in kg x $10^{-1}$ per cm core height |
|---|---|---|---|---|---|
| | $PuO_2$ v/o | | | | |
| 1 | 3.41200 | 3.41200 | 3.41200 | 3.41200 | 1.7756 |
| 2 | 3.40556 | 3.54010 | 3.21200 | 3.21200 | 1.7392 |
| 3 | 3.38058 | 3.69092 | 3.01200 | 3.01200 | 1.7036 |
| 4 | 3.35716 | 3.83033 | 2.81200 | 2.81200 | 1.6667 |
| 5 | 3.33521 | 3.95964 | 2.61200 | 2.61200 | 1.6286 |
| 6 | 3.31461 | 4.08004 | 2.41200 | 2.41200 | 1.5895 |
| 7 | 3.29748 | 4.17807 | 2.24549 | 2.21200 | 1.5523 |
| 8 | 3.29674 | 4.16940 | 2.32623 | 2.01200 | 1.5355 |
| 9 | 3.29536 | 4.16123 | 2.40797 | 1.81200 | 1.5188 |
| 10 | 3.29400 | 4.15322 | 2.48816 | 1.61200 | 1.5019 |
| 11 | 3.29266 | 4.14535 | 2.56682 | 1.41200 | 1.4849 |
| 12 | 3.29134 | 4.13762 | 2.64401 | 1.21200 | 1.4677 |
| 13 | 3.29005 | 4.13004 | 2.71979 | 1.01200 | 1.4503 |
| 14 | 3.28877 | 4.12259 | 2.79418 | 0.81200 | 1.4328 |
| 15 | 3.28751 | 4.11528 | 2.86724 | 0.61200 | 1.4151 |
| 16 | 3.28628 | 4.10809 | 2.93901 | 0.41200 | 1.3972 |
| 17 | 3.28506 | 4.10103 | 3.00951 | 0.21200 | 1.3792 |
| 18 | 3.28386 | 4.09409 | 3.07880 | 0.01200 | 1.3611 |
| 19 | 3.27152 | 4.09379 | 3.08245 | 0.00000 | 1.3584 |
| 20 | 3.27747 | 4.08592 | 3.07626 | 0.00000 | 1.3573 |
| 21 | 3.27746 | 4.08594 | 3.07623 | 0.00000 | 1.3573 |

Note that Eq. (3.3) is also the objective function of the critical mass optimization problem.

Table 3.10 shows that optimization of the fuel distribution in the core results in a reduction of the critical mass by 23.56%. In addition, comparison of Tables 3.10 and 3.4 shows that the configuration of maximum breeding gain of reactor No. 1 is also the configuration of minimum critical mass.

For the reasons explained in Section 3.1 a configuration having a further reduction in critical mass can be obtained by reapplication of the optimization procedure to reactor No. 2. The numerical results show that the critical mass of the optimum configuration of reactor No. 2 is equal to 12.333 kgs/cm, i.e. 30.54% smaller than the critical mass of the homogeneous reactor No. 1. In addition, the results show that the configuration of maximum breeding gain of reactor No. 2 is also the configuration of minimum critical mass.

As has been mentioned in Section 1.3 Goldschmidt and Quenon (8) used the Maximum Principle of Pontryagin to optimize the fissile fuel distribution of a fast reactor so as to obtain minimum critical mass, subject to the constraints that the power output be fixed and the power density and fuel enrichment be bounded. The reactor is of slab geometry and is described by one-group diffusion theory. They found that the optimum reactor consists of three distinct regions: a central region of constant power density, a region of maximum fuel enrichment and an outer region of minimum enrichment corresponding to the blanket. The zone of maximum enrichment disappears for sufficiently high values of maximum enrichment. From the numerical results they give, it is seen that when such a zone exists its thickness decreases as the reactor power output increases.

The same problem has been solved in the present study for a fast reactor of infinite cylindrical geometry described by five-group diffusion theory. The results obtained are similar. Specifically, for a five region reactor the optimum configuration consists of four core regions and a blanket. The three central core regions have a maximum power density equal to the upper limit of the power density. Since in this study we approximate continuous material distributions by region-wise constant distributions, the three central core regions correspond to the region of constant power density of reference (8) which allowed a continuously variable material distribution.

In summary, solutions of the minimum critical mass problem have widely appeared in the literature (6, 7, 8, 24, 25, 26, 27). These solutions, however, either do not consider realistic constraints which are required for practical reactor designs or they use at most two neutron groups for thermal reactors and one neutron group for fast reactors. In this study an improved solution to the minimum critical mass problem has been given by considering fast reactors of fixed power output, limited power density, limited fuel concentration and described by multigroup diffusion theory.

### 3.4 SODIUM VOID REACTIVITY OPTIMIZATION

One of the most important factors involved in the safety of large sodium-cooled fast reactors is the sodium void reactivity, which is defined as the change in reactivity resulting from the loss of sodium coolant from all, or some specified part, of the reactor. If positive, this reactivity can adversely affect the stability and safety of the reactor (28, 29). It follows that consideration should be given to the material distributions in a fast reactor so as to minimize the sodium void reactivity.

The optimization method developed in this study has been applied to a fast reactor of fixed power output, bounded power density and fuel volume fraction, to determine the fuel distribution which leads to a minimum sodium void reactivity. Note that the method can also be applied to determine the optimum distribution of any other material, for example a moderator, so that the sodium void reactivity is minimized.

For the mathematical formulation of the problem the fuel optimization process is viewed as follows: The critical reactor, or part of it, is voided and consequently the reactor becomes subcritical or supercritical. Then the question is raised as to how the fuel should be redistributed in the voided reactors so that: (a) the k-effective of the voided reactor is minimized; and (b) if the sodium is brought back into the reactor, the reactor becomes critical, delivers the same power as before voiding, and the power density is everywhere less than or equal to a given upper limit.

If the fissile fuel distribution of the voided reactor is changed from $u_f^0(r)$ to $u_f(r)$ and if $u_f(r)$ is sufficiently close to $u_f^0(r)$, then perturbation theory gives the following expression for the change in k-effective of the voided reactor

$$\frac{1}{k_v} - \frac{1}{k_v^p} = \int_0^{t_f} -u_f^* \sum_{i=1}^{N} \frac{(\Sigma_{tr,i}^{fs} - \Sigma_{tr,i}^{fr})}{3(\Sigma_{tr,i}^{0})^2} \nabla\phi_i \nabla\psi_i \, rdr +$$

$$\int_0^{t_f} u_f^* \sum_{i=1}^{N} (\Sigma_{a,i}^{fs} - \Sigma_{a,i}^{fr}) \phi_i \psi_i \, rdr +$$

$$\int_0^{t_f} u_f^* \sum_{i=1}^{N} \sum_{h=i+1}^{N} \{[\Sigma_{(i-h)}^{fs} - \Sigma_{(i-h)}^{fr}] \phi_i(\psi_i - \psi_h)\} rdr$$

$$- \frac{1}{k_v} \int_0^{t_f} u_f^* \sum_{i=1}^{N} \sum_{h=1}^{N} [\nu^{fs}\Sigma_{f,h}^{fs} - \nu^{fr}\Sigma_{f,h}^{fr}] \chi_i \phi_h \psi_i \, rdr \tag{3.4}$$

where

$k_v$ = k-effective of voided reactor

$k_p$ = k-effective of voided reactor after the fissile fuel perturbation

and

$$u_f^* = u_f - u_f^0 \tag{3.5}$$

The minimization of the sodium void reactivity is equivalent to the minimization of the quantity $(1/k_v) - (1/k_p)$ given by Eq. (3.4).

From the discussion up to this point it follows that the problem is mathematically described by the same equations as the breeding optimization problem, with the only difference that the objective function here is given by Eq. (3.4). The computational iterative scheme is the same as for the two previous problems.

The numerical results obtained for 100% voiding of the reactor core (but not the blanket) of reactor No. 1 are shown in Table 3.11. Comparison of Tables 3.4, 3.10 and 3.11 shows that for reactor No. 1 the configuration of maximum breeding gain and minimum critical mass is also the configuration of minimum sodium void reactivity.

For the reasons explained in Section 3.1 a configuration having a further reduction in sodium void reactivity can be obtained by reapplication of the optimization procedure to reactor No. 2. The numerical results show that the k-effective of the voided optimum configuration of reactor No. 2 is equal to 1.05507, i.e. the sodium void reactivity of the optimum configuration is 2.9 $ smaller than the same quantity of the homogeneous reactor No. 1 (for a delayed neutron fraction $\beta = 0.0035$). In addition the results show that the configuration of maximum breeding gain and minimum critical mass of reactor No. 2 is also the configuration of minimum sodium void reactivity.

The effect of the fuel distribution on sodium void reactivity was also studied by Allis-Chalmers (30). More specifically, changes

TABLE 3.11

Fissile Distribution and k-effective of Sodium Voided Reactor as a Function of Linear Programming Iteration Number for Reactor No. 1

| Iteration Number | Region 1 | Region 2 | Region 3 | Region 4 | k-effective of Sodium Voided Reactor |
|---|---|---|---|---|---|
| | $PuO_2$ v/o | | | | |
| 1 | 3.41200 | 3.41200 | 3.41200 | 3.41200 | 1.06523 |
| 2 | 3.40556 | 3.54010 | 3.21200 | 3.21200 | 1.06465 |
| 3 | 3.38058 | 3.69090 | 3.01200 | 3.01200 | 1.06401 |
| 4 | 3.35716 | 3.83033 | 2.81200 | 2.81200 | 1.06325 |
| 5 | 3.33521 | 3.95964 | 2.61200 | 2.61200 | 1.06241 |
| 6 | 3.31461 | 4.08004 | 2.41200 | 2.41200 | 1.06151 |
| 7 | 3.29748 | 4.17807 | 2.24549 | 2.21200 | 1.06064 |
| 8 | 3.29674 | 4.16940 | 2.32623 | 2.01200 | 1.06045 |
| 9 | 3.29536 | 4.16123 | 2.40797 | 1.81200 | 1.06027 |
| 10 | 3.29400 | 4.15322 | 2.48816 | 1.61200 | 1.06009 |
| 11 | 3.29266 | 4.14535 | 2.56682 | 1.41200 | 1.05990 |
| 12 | 3.29134 | 4.13762 | 2.64401 | 1.21200 | 1.05971 |
| 13 | 3.29005 | 4.13004 | 2.71979 | 1.01200 | 1.05952 |
| 14 | 3.28877 | 4.12259 | 2.79418 | 0.81200 | 1.05932 |
| 15 | 3.28751 | 4.11528 | 2.86724 | 0.61200 | 1.05913 |
| 16 | 3.28628 | 4.10809 | 2.93901 | 0.41200 | 1.05893 |
| 17 | 3.28506 | 4.10103 | 3.00951 | 0.21200 | 1.05873 |
| 18 | 3.27500 | 4.09409 | 3.07880 | 0.01200 | 1.05765 |
| 19 | 3.27923 | 4.08810 | 3.07797 | 0.00000 | 1.05764 |

in the sodium void reactivity resulting from radially varying the fuel enrichment to achieve radial power flattening in a cylindrical reactor were investigated. It was found that the flat power reactor had a sodium void reactivity 50% less than a homogeneous reactor producing the same total power. This is in agreement with the results of the present study.

## 3.5 SUMMARY

The numerical results discussed in this chapter show that for a fast breeder the fuel distribution which leads to a maximum initial breeding gain, leads also to a minimum critical mass, a minimum sodium void reactivity and a uniform power density (within the practical limits achievable through use of a small number of reactor zones). The significance of these results is obvious. A flat power density core is highly desirable from the aspect of thermal-hydraulic engineering design. This study shows that this highly desirable configuration is also the configuration of maximum breeding gain and minimum critical mass, which are of considerable importance from the point of view of reactor economics, and minimum sodium void reactivity which is of vital significance in reactor safety. Thus for future studies one may confidently choose a reference core without concern that practical designs will deviate far from it. Any further improvement in breeding performance, if it is feasible, will have to come through blanket modifications.

The problem of breeding optimization through blanket modifications is discussed in Chapter 4.

# Chapter 4

# BLANKET OPTIMIZATION

In this chapter the effects on the breeding gain of the insertion of a moderating material into the blanket and of surrounding the blanket by a reflector, are discussed.

Introduction of a moderating material into the blanket softens the spectrum and favors captures by the fertile material in the sub-kev energy range. In addition, if the blanket is surrounded by a good reflector the neutron leakage out of the blanket is reduced, and the capture rate of the fertile material is further improved.

## 4.1 THE EFFECT OF BLANKET MODERATION

The optimization of the distribution of BeO or Na in the blanket was investigated by means of the method described in Chapter 2. It was found that the breeding gain from iteration to iteration changed by an amount of the order of the expected numerical errors and that it changed erratically instead of improving. These results indicate that the breeding gain depends weakly on the moderator distribution. Accordingly, accumulated numerical errors are sufficiently large compared to changes in the optimization variables to preclude the study of optimization of the blanket breeding performance by the method of Chapter 2.

To support these results, the change of the breeding gain as a function of the moderator concentration, homogeneously distributed, was investigated.

The dimensions of an infinite cylindrical geometry reactor considered for the computations are shown in Table 4.1. The reactor compositions for BeO and Na moderated blankets are shown in Tables 4.2 and 4.3 respectively. For the neutronic calculations five neutron groups were used. The structure and cross sections of these groups are described in Section 3.1. The computations were carried out using the appropriate parts of the computer program discussed in Appendix D.

The breeding gain as a function of the moderator volume fraction in the blanket is shown in Table 4.4. From this table it is seen that: (a) for a BeO moderated blanket the breeding gain attains a maximum value for a moderator volume fraction somewhere between 5% and 10%; (b) this maximum value is only 0.096% larger than the breeding gain of a typical fast reactor blanket without any moderator; (c) for a Na moderated blanket, the breeding gain increases monotonically as the Na volume fraction decreases; (d) a change in the Na volume fraction from 10% to 50% decreases the breeding gain by only 3.604%; and (e) as the moderator volume fraction increases the blanket becomes a better core reflector and, consequently, the internal breeding gain increases slightly.

TABLE 4.1

Dimensions of Reactor used in Blanket Studies

| | Region | Inner Radius | Outer Radius |
|---|---|---|---|
| Core | 1 | 0.00 cm | 62.64 cm |
| | 2 | 62.64 cm | 90.48 cm |
| | 3 | 90.48 cm | 111.36 cm |
| Radial Blanket | 4 | 111.36 cm | 160.08 cm |
| Reflector | 5 | 160.08 cm | 206.48* cm |

*Extrapolated outer boundary

TABLE 4.2

Reactor Composition for BeO Moderated Blanket

| Material | Core Regions 1 | 2 | 3 | Blanket | Reflector | Atomic or Molecular Density for Pure Materials $cm^{-3} x 10^{-24}$ |
|---|---|---|---|---|---|---|
| $PuO_2$ | 3.2775 v/o | 4.0859 v/o | 3.0763 v/o | - | - | 0.025189 |
| $UO_2$ | 31.7225 v/o | 30.9141 v/o | 31.9237 v/o | 55 v/o ($UO_2$ + BeO) | - | 0.024444 |
| BeO | - | - | - | | - | 0.071270 |
| Na | 50 v/o | 50 v/o | 50 v/o | 30 v/o | - | 0.025410 |
| Fe | 15 v/o | 15 v/o | 15 v/o | 15 v/o | 100 v/o | 0.084870 |

TABLE 4.3

Reactor Composition for Na Moderated Blanket

| Material | Core Regions 1 | Core Regions 2 | Core Regions 3 | Blanket | Reflector | Atomic or Molecular Density for Pure Materials $cm^{-3}x10^{-24}$ |
|---|---|---|---|---|---|---|
| $PuO_2$ | 3.2775 v/o | 4.0859 v/o | 3.0763 v/o | - | - | 0.025189 |
| $UO_2$ | 31.7225 v/o | 30.9141 v/o | 31.9237 v/o | 85 v/o ($UO_2$ + Na) | - | 0.024444 |
| Na | 50 v/o | 50 v/o | 50 v/o | | - | 0.025410 |
| Fe | 15 v/o | 15 v/o | 15 v/o | 15 v/o | 100 v/o | 0.084870 |

TABLE 4.4

The Breeding Gain as a Function of Moderator Concentration in the Blanket

| Case | Moderator v/o | $U^{238}$ v/o | Breeding Gain | | |
|---|---|---|---|---|---|
| | | | Internal | External | Total |
| **Na Moderator** | | | | | |
| 1 | 10 | 75 | 0.340401 | 0.286165 | 0.626566 |
| 2 | 20 | 65 | 0.341137 | 0.282633 | 0.623770 |
| 3 | 30* | 55 | 0.342077 | 0.277693 | 0.619770 |
| 4 | 40 | 45 | 0.343326 | 0.270523 | 0.613849 |
| 5 | 50 | 35 | 0.345091 | 0.259680 | 0.604771 |
| **BeO Moderator** | | | | | |
| 6 | 0 | 55 | 0.342077 | 0.277693 | 0.619770 |
| 7 | 5 | 50 | 0.344532 | 0.275832 | 0.620364 |
| 8 | 10 | 45 | 0.347181 | 0.272908 | 0.620089 |
| 9 | 20 | 35 | 0.353354 | 0.263742 | 0.617096 |
| 10 | 30 | 25 | 0.361465 | 0.248656 | 0.610121 |
| 11 | 5** | 50 | 0.344557 | 0.275206 | 0.619763 |
| 12 | 5*** | 50 | 0.343183 | 0.271740 | 0.614923 |

* The volume fractions of Na and $UO_2$ of this row are representative of typical fast reactor blanket designs

** $\sigma^{BeO}_{(n,2n)} = 0.0$

*** $\sigma^{BeO}_{\text{down-scattering}} = 0.0$

The 11th row of Table 4.4 shows the breeding gain for a blanket moderated by a fictitious BeO with the cross section for the (n,2n) reaction set equal to zero. The 12th row of the same table shows the breeding gain for a blanket diluted by a fictitious BeO with down-scattering cross sections set equal to zero. Comparison of the 6th, 7th, 11th and 12th rows of Table 4.4 shows that the improvement in breeding due to BeO moderation just offsets the loss in breeding due to reduction of the $U^{238}$ concentration; the net 0.096% improvement of the breeding gain is due to the production of neutrons by BeO through the (n,2n) reaction.

The results just cited support the conclusion of the optimization studies to the effect that the initial breeding gain depends weakly on the moderator volume fraction in the blanket. This weak dependence could be of considerable importance to reactor economics. It suggests that the addition of an appropriate moderator or diluent in the blanket (and consequently the reduction of $U^{238}$ concentration) might reduce the reprocessing and fabrication costs without significant penalties in breeding gain.

Finally, it is noteworthy that the method of Chapter 2 would be applicable to the problem of blanket optimization if the criterion of optimality were a stronger function of the moderator concentration in the blanket. For example, such a criterion might be the contribution of the blanket to the cost of reactor power.

## 4.2 THE EFFECT OF THE REFLECTOR COMPOSITION

The breeding gains for three different reflectors, BeO, graphite and Fe, and for three different blanket thicknesses, a one-row blanket (16.24 cm), a two-row blanket (32.48 cm) and a three-row blanket (48.72 cm) are shown in Table 4.5. It is seen from this table that: (a) surrounding the blanket with a reflector improves the breeding gain, compared to an unreflected blanket; the improvement is more significant as the blanket thickness decreases; (b) BeO is better than graphite, and graphite is better than Fe; (c) the breeding gain becomes a stronger function of the reflector properties as the blanket thickness decreases; (d) the internal breeding gain is practically insensitive to the nature of the reflector (as long as there is at least one row of blanket assemblies between core and reflector); and (e) for a 46.4 cm BeO reflector, the breeding gain of a three-row blanket is larger than that of a one-row blanket by only 3.31%. The results of Table 4.5 suggest that from the standpoint of economics a one- or two-row blanket surrounded by a BeO reflector could be better than a three-row blanket. Reduction of the blanket thickness might reduce the reprocessing and fabrication costs without significant penalties in breeding gain.

On the basis of breeding alone, there are two benefits to be obtained from the addition of reflectors: (a) neutron leakage is reduced from the blanket; and (b) neutron moderation softens the spectrum and favors captures by the fertile material in the sub-kev

TABLE 4.5

The Breeding Gain as a Function of the
Reflector Material and Blanket Thickness

| Blanket Thickness cm | Breeding Gain | | |
|---|---|---|---|
| | Internal | External | Total |
| | **BeO Reflector** | | |
| 16.24 | 0.344334 | 0.256966 | 0.601300 |
| 32.48 | 0.342144 | 0.276049 | 0.618193 |
| 48.72 | 0.342076 | 0.279802 | 0.621878 |
| | **Graphite Reflector** | | |
| 16.24 | 0.343837 | 0.240930 | 0.584767 |
| 32.48 | 0.342133 | 0.271428 | 0.613561 |
| 48.72 | 0.342076 | 0.279611 | 0.621687 |
| | **Iron Reflector** | | |
| 16.24 | 0.343804 | 0.213572 | 0.557376 |
| 32.48 | 0.342196 | 0.263786 | 0.605982 |
| 48.72 | 0.342077 | 0.277693 | 0.619770 |
| | **No Reflector** | | |
| 32.48 | 0.341873 | 0.227775 | 0.569648 |
| 48.72 | 0.342071 | 0.267543 | 0.609614 |

energy range. In this regard BeO is better than graphite and Fe. In addition, BeO has the property of producing neutrons through a (n,2n) reaction for incident neutron energies higher than 1.8 Mev. To evaluate the relative significance of the reflective and moderating properties and of the (n,2n) reaction with respect to the breeding gain, the breeding gain has been computed for a two-row blanket and: (a) a fictitious "infinite mass" BeO reflector with down-scattering cross sections set equal to zero; (b) a fictitious BeO reflector with the cross section for the (n,2n) reaction set equal to zero. The results are shown in Table 4.6. It is seen from this table that: (a) the reduction of neutron leakage is much more significant than moderation; and (b) the effect of the (n,2n) reaction is negligible. These results suggest that a simple figure of merit of a fast reactor blanket reflector could be determined as a function of only the transport and absorption cross sections of the reflector. A mean albedo (calculated using properly weighted cross sections) could be such a figure of merit. If this is so, then all materials could be ranked according to this figure of merit and the best fast reactor blanket reflector material readily selected.

It must be pointed out that all computations up to this point have been done without taking into account any resonance self-shielding corrections. The breeding gains of a two row blanket surrounded by a BeO reflector with shielded and unshielded cross sections for $U^{238}$ are shown in Table 4.7. It is seen from this table that the shielded cross sections give a slightly smaller breeding gain. It is worth

TABLE 4.6

The Breeding Gain as a Function of BeO Reflector Properties

| Reflector | Breeding Gain | | |
|---|---|---|---|
| | Internal | External | Total |
| No Reflector | 0.341873 | 0.227775 | 0.569648 |
| BeO with $\sigma_{down-scat}=0.0$ | 0.342354 | 0.273840 | 0.616194 |
| BeO with $\sigma_{n,2n}=0.0$ | 0.342146 | 0.275884 | 0.618030 |
| BeO | 0.342144 | 0.276049 | 0.618193 |

TABLE 4.7

The Effect of Resonance Self-Shielding on Breeding Gain

| $U^{238}$ cross sections | Breeding Gain | | |
|---|---|---|---|
| | Internal | External | Total |
| Unshielded | 0.342144 | 0.276049 | 0.618193 |
| Shielded | 0.346069 | 0.265469 | 0.611538 |

noting that the effect of self-shielding would be more significant if appreciable amounts of a strong absorber such as plutonium were present in the blanket, as will occur near the end of the blanket fuel sub-assembly irradiation life.

In summary, the results of this chapter show that further investigation should be undertaken to determine if a moderated or diluted blanket, or a thin blanket surrounded by a good reflector are economically attractive. A more thorough examination of alternate high-albedo reflector materials is also indicated.

# Chapter 5

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 CONCLUSIONS

The purpose of this study has been the development and application of a method to optimize the material distributions in a fast reactor of fixed power output constrained power density and constrained material volume fractions, so as to maximize or minimize a given objective function.

An iterative method has been developed based on linearization of the relations describing the system and on Linear Programming. The method can be used to optimize integral reactor quantities which are linear functions of the neutron flux and linear functions of the material volume fractions (i.e. quantities which are integrals containing the material volume fractions and the neutron flux, or their products, to the first power only).

The method has been applied successfully to the problems of optimization of the fuel distribution in the reactor core so as to obtain a maximum initial breeding gain, a minimum critical mass and a minimum sodium void reactivity.

For a four region core numerical results show that the core of maximum breeding gain is also the core of minimum critical mass, minimum sodium void reactivity and uniform power density. It is expected, however, that these results are more general, and would be true regardless of the number of regions.

In addition, numerical results show (Table 3.9) that if the blanket is surrounded by a good reflector such as BeO the optimization of the fuel in the core leads to a small improvement in the breeding gain, while the improvement is considerably larger for a bare blanket. Since in power reactors there is always a reflector surrounding the blanket, the results of Table 3.9 show that a small improvement in breeding gain results from optimization of the fuel distribution in the core. Thus, from an economic standpoint one might argue that the much larger improvement in fissile inventory is more important. Since it has been shown that both optimizations lead to the same result, however, this distinction need not be the source of conflict.

The method has also been applied to the problem of optimization of the distribution of a moderator in a fast reactor blanket so as to obtain a maximum initial breeding gain. Numerical results indicate, however, that initial breeding gain is a weak function of the moderator concentration in the blanket and, therefore, numerical errors are sufficiently large compared to changes in the optimization variables to obviate blanket optimization by this approach.

On the other hand, the dependence of the breeding gain on the moderator concentration homogeneously distributed in the blanket has been studied in Chapter 4. The results show that for even marginally significant changes in the breeding gain large changes in the moderator volume fraction in the blanket are required.

In addition, the results of Chapter 4 show that: (a) when Na replaces $U^{238}$ in the blanket the neutron moderation by Na is not enough to offset the loss in breeding due to reduction of the $U^{238}$ concentration and consequently the breeding gain decreases as the Na concentration increases; (b) when BeO replaces $U^{238}$ in the blanket, for a BeO volume fraction somewhere between 5% and 10% the improvement in breeding due to moderation by BeO just offsets the loss in breeding due to reduction of the $U^{238}$ concentration; for any other BeO concentration the neutron moderation is not enough to offset breeding losses due to reduction of the $U^{238}$ concentration; (c) the breeding gain is a weak function of the blanket thickness if the blanket is surrounded by a good reflector; and (d) the transport and absorption properties of a medium, rather than its moderating properties, determine the figure of merit of a fast reactor blanket reflector.

## 5.2 RECOMMENDATIONS FOR FUTURE WORK

The method developed in Chapter 2 can be used to solve many other important reactor optimization problems. Some of these problems are as follows:

1) Optimization of the fuel distribution or moderator distribution in a fast reactor core so as to maximize the magnitude of the negative Doppler coefficient. In this problem the objective function would be the Doppler coefficient as given by perturbation theory.

2) Optimization of the moderator distribution in a fast reactor core so as to minimize the sodium void reactivity. In this problem the objective function would be an expression for the sodium void reactivity

analogous to Eq. (3.4).

3) Optimization of either the fuel distribution or the moderator distribution or both in a fast reactor core so as to minimize the sodium temperature coefficient. This problem is equivalent to problem No. 2 since reduction of the sodium density due to a temperature increase can be treated as equivalent to small voids in sodium.

4) Optimization of the shape of the reactor core in the axial direction so as to minimize the sodium void reactivity. If the axial leakage from the core is represented by an appropriate $DB_z^2(r)$ term then the problem can be formulated as follows: A fictitious material having an absorption cross section equal to D (the homogenized diffusion coefficient of the core materials), all other cross sections equal to zero, and a concentration equal to $B_z^2(r)$ (axial buckling) is introduced into the core. Then, the optimum radial distribution of this material is sought so as to minimize the sodium void reactivity. If $B_{0,z}^2(r)$ is the optimum buckling distribution, then the optimum core height distribution, $H_0(r)$, is determined by the relation

$$H_0(r) = \frac{\pi}{B_{0,z}(r)} \tag{5.1}$$

In this problem the objective function would also be an expression for the sodium void reactivity analogous to Eq. (3.4).

5) Optimization of the distribution of a control poison so as to minimize the amount of poison required. In this problem the objective function would be of the form

$$I = \int_V u_p dV \tag{5.2}$$

where

$u_p$ = volume fraction of control poison.

As discussed in Chapter 2 the solution of the linearized multigroup diffusion equations involves the inversion of a matrix. This limits the number of reactor regions and neutron groups which can be employed since the inversion of a large matrix requires excessive computer time and gives rise to prohibitive round-off errors. Future work could improve the accuracy of the method and remove the limitations on the number of reactor regions and neutron groups which can be employed, by investigating methods of solution of the linearized multigroup diffusion equations which avoid the matrix inversion.

This study has not considered any time-dependent problems. Many important reactor problems, however, are time-dependent. For example a more detailed study of the breeding optimization problem should take into account the fact that breeding gain is a time-dependent parameter. This suggests the need for the extension of the developed optimization method to time-dependent problems.

Another interesting area for future work is the application of the method to economic optimization problems. This should be a simple matter since many such problems can be cast into forms essentially linear in inventory and breeding gain.

From the results of Chapter 4 it has been concluded that: (a) the breeding gain is a weak function of the moderator distribution in the blanket; (b) the breeding gain is also a weak function of the blanket thickness if the blanket is surrounded by a good reflector; and (c) the effectiveness of a fast reactor blanket reflector is mainly a function of the reflective (as opposed to moderating) properties. These conclusions suggest additional areas for future work. Specifically conclusions (a) and (b) suggest that further investigation should be undertaken to determine if a moderated or diluted blanket, or a thin blanket surrounded by a good reflector are economically attractive. The replacement of uranium in the blanket by an appropriate moderator or diluent or the reduction of the blanket thickness might reduce the reprocessing and fabrication costs without significant penalties in breeding. In addition, conclusion (c) suggests further investigation to determine a specific, simple figure of merit for a fast reactor blanket reflector such as a mean albedo (calculated by using properly weighted cross sections), and its use to survey and rank all materials according to this figure of merit.

# Appendix A

## BIBLIOGRAPHY

This Appendix contains a selection of references on theory and applications of optimization methods. A brief comment is included on each.

1. Athans, M. and P. L. Falb, "Optimal Control", McGraw-Hill, New York, 1965. Theory and applications of optimal control.
2. Balakrishnan, A. V. and L. W. Neustadt, eds., "Computing Methods in Optimization Problems", Academic Press, New York, 1964. Papers presented on the conference on Computing Methods on Optimization Problems at the University of California at Los Angeles in January, 1964.
3. Bellman, R. E. and S. E. Dreyfus, "Applied Dynamic Programming", Princeton University Press, Princeton, New Jersey, 1962. Theory and Applications of Dynamic Programming.
4. Blaine, R. A. and J. L. Watts, "Radial Flux Flattening in the organic moderated Reactor Critical Assembly by variable Fuel-To-Moderator Ratio", NAA-SR-5858 (1961). A calculation is described concerning a flux flattening experiment in the Organic Moderated Reactor Critical Assembly.
5. Bryson, A. E. and Y. Ho, "Applied Optimal Control", Blaisdel Publishing Company, Waltham, Massachusetts, 1969. Theory and applications of optimal control.

6. Bouchey, G. D., C. S. Beightler and B. V. Koen, "Optimization of Nuclear Systems by Geometric Programming", Nucl. Sci. Eng., 44, 267 (1971). The application of Geometric Programming in Nuclear Engineering problems is discussed and the method is illustrated by a few simplified examples.

7. Denn, M. M., "Optimization by Variational Methods", McGraw-Hill, New York, 1969. Optimization theory at an elementary mathematical level, with applications to simple but typical process design and control problems.

8. Heusener, G., "Core-Optimization of Sodium Cooled Fast Breeder Reactors with Methods of Nonlinear Programming", Nucl. Eng. and Design, 14, 3 (1970). Methods of solution of nonlinear optimization problems are described, and one of them has been applied to the core optimization of a 1000 MWe fast breeder.

9. Kotaro, I., "Fast-Reactor-Core Design Optimization by Linear Programming", Nucl. Sci. Eng., 39, 394 (1970). Linear Programming is applied to optimize the design of a fast reactor core.

10. Leondes, C. T., ed., "Advances in Control Systems: Theory and Applications", Academic Press, New York, 1964. Theory and applications of control systems by leading contributors in the field.

11. Mansfield, W. K., ed. "Introduction to Nuclear Engineering", Vol. 2, Simmons-Boardman, New York, 1959. A series of monographs

one of which deals with application of simple optimization techniques to Nuclear Engineering Problems.

12. Mohler, R. R. and C. N. Shen, "Optimal Control of Nuclear Reactors", Academic Press, New York, 1970. Applications of optimization techniques in Nuclear Engineering Problems with brief summaries of reactor dynamics, classical reactor control, and optimal control fundamentals.

13. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidge and E. F. Mishchenko, "The Mathematical Theory of Optimal Processes", Willey (Interscience), New York, 1962. An excellent reference on the Maximum Principle of Pontryagin.

14. Sauer, A., "Optimum Control and Flat Flux", Nucl. Sci. Eng., 5, 71 (1959). The problem of optimizing the distribution of a control poison in a thermal reactor to minimize the amount of the control poison required is discussed.

15. Suzuki, A. and R. Kiyose, "Maximizing the Average Fuel Burnup Over Entire Core: a Poison Management Optimization Problem for Multizone Light-water Reactor Cores", Nucl. Sci. Eng., 44, 121 (1971). A method based on a topological mapping theory and used to find the governing principles in optimal control programming to maximize the average burnup for light-water reactors is described.

16. Sherman, D. C., "A Formal Procedure for Rapid Optimization of Design Performance", KAPL-2114, 1961. A systematic procedure of design performance optimization is discussed.

17. Terney, W. B., and H. Fenech, "Shipboard Reactor Shield Optimization Using the Optimum Gradient Method", Nuclear Applications, 3, 47 (1967). The optimum gradient technique is used to optimize a shipboard reactor shield system consisting of a water-lead primary and a concrete-lead-polyethylene secondary shield.

18. Terney, W. B., "Analytic Solution to the Flat Flux Problem", Nucl. Sci. Eng., 45, 266 (1971). The Maximum Principle of Pontryagin is used to find the optimum $k_\infty$ distribution so as to minimize the integral of the squared deviation of the flux from its average value, while $k_\infty$ is restricted by lower and upper bounds.

19. Wade, D. C. and W. B. Terney, "Optimal Control of Nuclear Reactor Depletion", Nucl. Sci. Eng., 45, 199 (1971). The design and operation of a nuclear reactor are posed as optimal control problems, and the Maximum Principle of Pontryagin is used to derive the necessary conditions of optimality.

20. Wilkins, J. E. Jr., "Minimum Total Mass", Nucl. Sci. Eng., 6, 229 (1959). The problem of designing a reactor with minimum total mass is discussed.

21. Wilkins, J. E. Jr., "Nuclear Reactors with Maximum Prompt Neutron Lifetime", Nucl. Sci. Eng., 10, 331 (1961). The problem of the optimum fuel distribution in a thermal reactor to obtain a maximum prompt neutron lifetime is discussed.

# Appendix B

# LINEAR PROGRAMMING AND LINEARIZATION

In this Appendix the general Linear Programming problem and the linearization procedure are briefly discussed. The reader who may be more deeply interested in Linear Programming is referred to the book by Gass (30) for acquisition of basic material, while Dantzig (31) and Hadley (16) provide a more detailed and sophisticated treatment.

## B.1 LINEAR PROGRAMMING

Linear Programming is concerned with the solution of optimization problems for which all relations among the variables are linear both in the constraints and the function to be maximized or minimized. The general Linear Programming problem can be stated as follows: Given a set of m linear equations, or inequalities, or both, in r variables, find non-negative values of these variables which satisfy the constraints and maximize or minimize some linear function of the variables.

In terms of symbols, this statement is equivalent to the seeking of a vector $\underline{x}$ with non-negative components which satisfies the relations

$$\underline{A}\ \underline{x} \gtreqless \underline{b}, \qquad (B.1)$$

and maximizes or minimizes the function

$$I = \underline{c}\,\underline{x}, \tag{B.2}$$

where the matrix $\underline{A}$, and the vectors $\underline{b}$ and $\underline{c}$ are all independent of $\underline{x}$.

## B.2 LINEARIZATION

Since Linear Programming is a method for maximizing or minimizing a linear objective function for a system of linear algebraic constraining relationships, linearization can be used as a first step to reduce a nonlinear problem into a suitable form for use of Linear Programming. For the sake of generality the linearization procedure is discussed here for a general nonlinear optimization problem.

Such a problem can be stated as follows (33): Determine the optimal control $\underline{u}(t)$ which maximizes (minimizes) the functional

$$I = \int_{t_i}^{t_f} L(\underline{x}, \underline{u}, t)dt + S[\underline{x}(t_f), t_f], \tag{B.3}$$

in a class of functions $\underline{x}(t)$, $\underline{u}(t)$, satisfying the differential equations

$$\frac{d\underline{x}}{dt} = \underline{f}(\underline{x}, \underline{u}, t) \tag{B.4}$$

The terminal point $t_f$ may be fixed or free, the terminal state $\underline{x}(t_f)$ may be fixed, completely free, or specified by a set of equations of the form

$$\underline{h}[\underline{x}(t_f), t_f] = 0 \tag{B.5}$$

The control vector $\underline{u}(t)$ is a member of a set U called the control region, which may be either open or closed. The state vector $\underline{x}(t)$ and the control vector $\underline{u}(t)$ satisfy constraints of the form

$$\underline{\Phi}(t, \underline{x}, \underline{u}) \leq 0 \tag{B.6}$$

The linearization proceeds as follows: Let $\underline{x}^0$, $\underline{u}^0$ be a solution of Eqs. (B.4) and

$$\frac{dx_i}{dt} = f_i(x_i, x_2, .., x_j, u_1, u_2, .., u_k, t) \tag{B.7}$$

a member of the system of Eqs. (B.7). Equation (B.7) can be linearized by means of a Taylor series expansion of $f_i$ about $\underline{x}^0$, $\underline{u}^0$. This series expansion is given by the relation

$$f_i(x_1, .., x_j, u_1, .., u_k, t) = f_i(x_1^0, .., x_j^0, u_1^0, .., u_k^0, t) +$$

$$\frac{\partial f_i}{\partial x_1}(x_1-x_1^0) + \ldots + \frac{\partial f_i}{\partial x_j}(x_j-x_j^0) + \frac{\partial f_i}{\partial u_1}(u_1-u_1^0) + \ldots +$$

$$\frac{\partial f_i}{\partial u_k}(u_k-u_k^0) + \text{higher-order terms,} \tag{B.8}$$

where the derivatives are evaluated at

$$x_1^0, \ldots x_j^0, u_1^0, \ldots u_k^0$$

If changes in $\underline{x}$ and $\underline{u}$ from the solution $\underline{x}^0$, $\underline{u}^0$ are designated as $\underline{x}^*$ and $\underline{u}^*$, defined by the relations

$$\underline{x}^* = \underline{x} - \underline{x}^0, \quad \underline{u}^* = \underline{u} - \underline{u}^0, \tag{B.9}$$

then Eq. (B.8) can be written in terms of $\underline{x}^*$ and $\underline{u}^*$ as

$$f_i(x_1, .., x_j, u_1, .., u_k, t) = f_i(x_1^0, .., x_j^0, u_1^0, .., u_k^0, t) +$$

$$\frac{\partial f_i}{\partial x_1} x_1^* + \ldots + \frac{\partial f_i}{\partial x_j} x_j^* + \frac{\partial f_i}{\partial u_1} u_1^* + \ldots + \frac{\partial f_i}{\partial u_k} u_k^* +$$

higher-order terms. (B.10)

Since

$$\frac{dx_i}{dt} = f_i(x_1, .., x_j, u_1, .., u_k, t),$$

and

$$\frac{dx_i^0}{dt} = f_i(x_1^0, .., x_j^0, u_1^0, .., u_k^0, t),$$

with $\underline{x}^*$ and $\underline{u}^*$ sufficiently close to $\underline{x}^0$ and $\underline{u}^0$, a first-order approximation to Eq. (B.7) is given by the relation

$$\frac{dx_i^*}{dt} = \frac{\partial f_i}{\partial x_1} x_1^* + .. + \frac{\partial f_i}{\partial x_j} x_j^* + \frac{\partial f_i}{\partial u_1} u_1^* + .. + \frac{\partial f_i}{\partial u_k} u_k^* \qquad \text{(B.11)}$$

Equation (B.11) represents the linearized form of the i-th of Eqs. (2.9).

The functional to be maximized (minimized) and the constraints of the problem are linearized in a similar way.

The second step in reducing the problem into a suitable form for use of Linear Programming is to transform the linearized relations describing the problem into linear algebraic relations. For the optimization problem with this study is concerned this is achieved as follows: (a) the reactor is divided into a number, R, of regions, each with spatially uniform material concentrations; and (b) the linearized multigroup diffusion equations are solved to express each $\phi_i^*$(i=1, N) as a function of $u_{f,j}$, $u_{m,j}$ (j = 1,R). As explained in Section 2.3 for the solution of the linearized multigroup diffusion equations the method of Piecewise Polynomials is used. A brief description of this method is given in Appendix C.

# Appendix C

## THE METHOD OF PIECEWISE POLYNOMIALS AND INTEGRALS OF PIECEWISE POLYNOMIALS

### C.1 THE METHOD OF PIECEWISE POLYNOMIALS

The method of Piecewise Polynomials developed by Kang (19) to solve the multigroup diffusion equations has the following characteristics. The reactor is divided into a number, n, of mesh points and the flux, $\phi_i$, for the i-th group is approximated by a sum of properly defined piecewise polynomials. For example, if cubic piecewise polynomials are employed, the flux $\phi_i$ in a cylindrical reactor is approximated by the relation

$$\phi_i \simeq \Phi_i = \sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_{k,i} \qquad \text{(C.1)}$$

where $a_{k,i}$ and $\beta_{k,i}$ are constants and $w_k$ and $v_{k,i}$ are cubic piecewise polynomials defined as

$$w_k = \begin{cases} 3\left(\frac{r-r_{k-1}}{h_-}\right)^2 - 2\left(\frac{r-r_{k-1}}{h_-}\right)^3, & r\varepsilon[r_{k-1},r_k] \\ 3\left(\frac{r_{k+1}-r}{h_+}\right)^2 - \left(\frac{r_{k+1}-r}{h_+}\right)^3, & r\varepsilon[r_k,r_{k+1}] \\ 0 \text{ otherwise} \end{cases} \qquad \text{(C.2)}$$

$$v_{k.i} = \begin{cases} -\dfrac{(r-r_{k-1})^2}{D_{i-}h_-} + \dfrac{(r-r_{k-1})^3}{D_{i-}h_-^2}, & r\varepsilon[r_{k-1},r_k] \\ \dfrac{(r_{k+1}-r)^2}{D_{i+}h_+} - \dfrac{(r_{k+1}-r)^3}{D_{i+}h_+^2}, & r\varepsilon[r_k,r_{k+1}] \\ 0 \text{ otherwise} \end{cases} \quad (C.3)$$

where

$h_-$ = mesh interval to the left of mesh point k

$h_+$ = mesh interval to the right of mesh point k

$D_{i-}$ = diffusion coefficient, for the group i, to the left of mesh point k

$D_{i+}$ = diffusion coefficient, for the group i, to the right of mesh point k

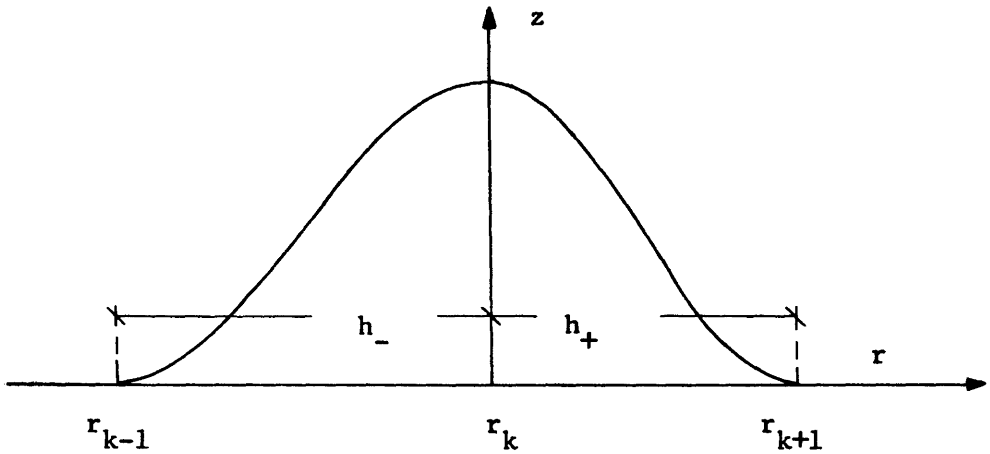
$r_k$ = radial position of mesh point k

The cubic piecewise polynomials $w_k$ and $v_{k,i}$ corresponding to the mesh point k are shown in Fig. C.1. Since

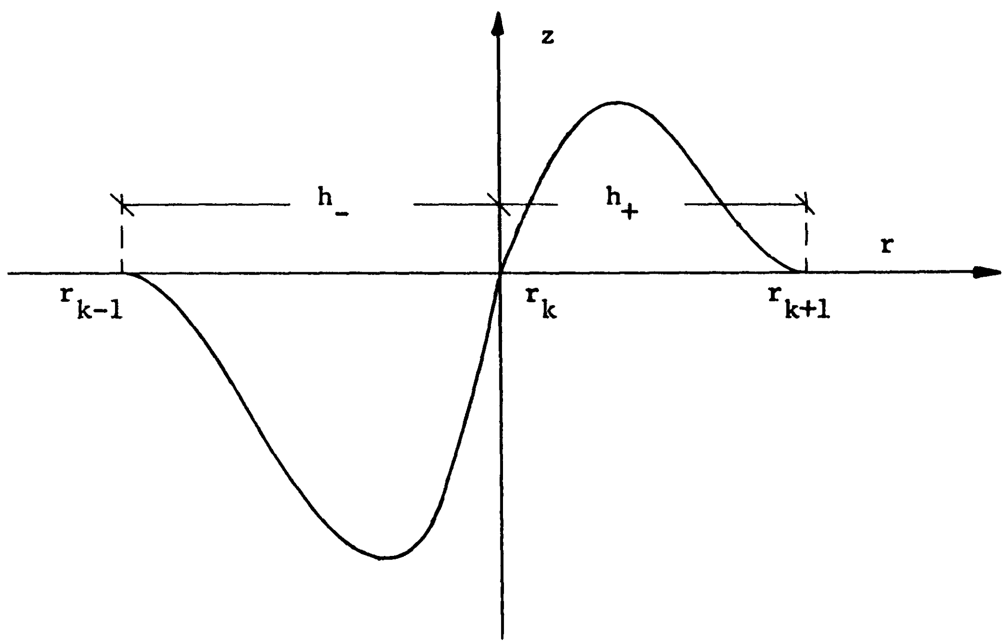
$$\frac{dw_k}{dr} = 0 \text{ at } k-1,\ k,\ k+1$$

$$\frac{dv_{k,i}}{dr} = 0 \text{ at } k-1,\ k+1 \quad (C.4)$$

$$D_{i-}\frac{dv_{k,i}}{dr} = D_{i+}\frac{dv_{k,i}}{dr} \text{ at } k$$




PIECEWISE POLYNOMIAL $w_k$




PIECEWISE POLYNOMIAL $v_{k,i}$

FIG. C.1 THE CUBIC PIECEWISE POLYNOMIALS $w_k$ AND $v_{k,i}$

$w_k = 0, \; v_{k,i} = 0$ at k-1, k+1

$w_k = 1, \; v_{k,i} = 0$ at k

the conditions of continuity of flux and current at interfaces are automatically satisfied by selecting the interface as a mesh point. To satisfy the boundary conditions

$$\frac{d\phi_i}{dr} /_0 = 0, \quad \phi_i(t_f) = 0, \tag{C.5}$$

we define

$$\beta_{1,i} = 0, \quad a_{n,i} = 0 \tag{C.6}$$

The multigroup diffusion equations can be written in the form

$$L_i\phi_i = 0 \tag{C.7}$$

where $L_i$ is the multigroup diffusion operator for the i-th group. Then, the coefficients $a_{k,i}$ and $\beta_{k,i}$ of the piecewise polynomials $w_k$ and $v_{k,i}$ in Eq. (C.1) are determined by requiring that

$$\int_V (L_i\Phi_i)w_k dV = 0, \tag{C.8}$$

$$\int_V (L_i\Phi_i)v_{k,i} dV = 0 \tag{C.9}$$

where k = 1,n

After the integrations are carried out in Eqs. (C.8) and (C.9) a number of linear algebraic equations results equal to the number of the coefficients $a_{k,i}$, $\beta_{k,i}$ from which these coefficients can be determined.

The error involved in approximating $\phi_i$ by $\Phi_i$ is given by (19)

$$|\phi_i - \Phi_i| \leq k(h)^4 , \tag{C.10}$$

where k is a constant and h is the largest mesh interval. Kang (19) has shown that for one-dimensional calculations a reduction by a factor of about 10 in the number of mesh points is possible by the use of cubic piecewise polynomials compared to conventional finite difference calculations of the same accuracy.

C.2 INTEGRALS OF PIECEWISE POLYNOMIALS

For the numerical application of the method of Piecewise Polynomials to solve the linearized multigroup diffusion equations (Section 2.3), the evaluation of some integral quantities involving piecewise polynomials is needed. In this section analytic expressions are given for those which can be evaluated in closed form.

As discussed in Section 2.3, the constants $a_{k,i}$ and $\beta_{k,i}$ of Eq. (2.20) are determined by requiring

$$\int_V (L_i \Phi_i)\, w_k dV = \int_V f_i\, (u_f^*, u_m^*)\, w_k dV , \tag{C.11}$$

and

$$\int_V (L_i \Phi_i)\, v_{k,i} dV = \int_V f_i\, (u_f^*, u_m^*)\, v_{k,i} dV \tag{C.12}$$

or

$$\int_V [L_i\, (\sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_{k,i})]\, w_k dV =$$

$$\int_V f_i\, (u_f^*, u_m^*)\, w_k dV \tag{C.13}$$

and

$$\int_V [L_i\, (\sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_{k,i})]\, v_{k,i} dV =$$

$$\int_V f_i\, (u_f^*, u_m^*)\, v_{k,i} dV \tag{C.14}$$

The left hand side of Eqs. (C.13) and (C.14) is the sum of integrals of products of the piecewise polynomials and of products of their derivatives. Since the piecewise polynomials $w_k$ and $v_{k,i}$ are zero everywhere outside the interval $[r_{k-1}, r_{k+1}]$ the non-zero integrals of these products are (for cubic piecewise polynomials):

$$\int_V w_k w_k \; rdr = \frac{2}{7}(h_-^2 - h_+^2) + \frac{13}{35}(h_- r_{k-1} + h_+ r_{k+1})$$

$$\int_V w_k w_{k+1} \; rdr = \frac{9}{140} \; h_+^2 + \frac{9}{70} h_+ r_i$$

$$\int_V w_k w_{k-1} \; rdr = - \frac{9}{140} h_-^2 + \frac{9}{70} h_- r_i$$

$$\int_V v_{k,i} v_{k,j} \; rdr = \frac{1}{105} [r_{k+1} \frac{h_+^3}{D_{i+} D_{j+}} + r_{k-1} \frac{h_-^3}{D_{i-} D_{j-}} ] + \frac{1}{168} [\frac{h_-^4}{D_{i-} D_{j-}} - \frac{h_+^4}{D_{i+} D_{j+}}]$$

$$\int_V v_{k,i} v_{k+1,j} \; rdr = - \frac{1}{140} r_k \frac{h_+^3}{D_{i+} D_{j+}} - \frac{1}{280} \frac{h_+^4}{D_{i+} D_{j+}}$$

$$\int_V v_{k,i} v_{k-1,j} \; rdr = - \frac{1}{140} r_k \frac{h_-^3}{D_{i-} D_{j-}} + \frac{1}{280} \frac{h_-^4}{D_{i-} D_{j-}}$$

$$\int_V w_k v_{k,i} \; rdr = - \frac{11}{210} (r_{k-1} \frac{h_-^2}{D_{i-}} - r_{k+1} \frac{h_+^2}{D_{i+}}) - \frac{1}{28} (\frac{h_-^3}{D_{i-}} + \frac{h_+^3}{D_{i+}})$$

$$\int_V w_k v_{k+1,i} \; rdr = -\frac{13}{420} \frac{r_k h_+^2}{D_{i+}} - \frac{h_+^3}{70 \; D_{i+}}$$

$$\int_V w_k v_{k-1,i} \; rdr = \frac{13}{420} r_k \frac{h_-^2}{D_{i-}} - \frac{1}{70} \frac{h_-^3}{D_{i-}}$$

$$\int_V \frac{dw_k}{dr} \times \frac{dw_k}{dr} \; rdr = \frac{6 \; r_{k-1}}{5 \; h_-} + 0.6 + \frac{6 \; r_{k+1}}{5 \; h_+} - 0.6^*$$

$$\int_V \frac{dw_k}{dr} \times \frac{dw_{k+1}}{dr} \; rdr = -\frac{6r_k}{5h_+} - 0.6$$

$$\int_V \frac{dw_k}{dr} \times \frac{dw_{k-1}}{dr} \; rdr = -\frac{6r_k}{5h_-} + 0.6$$

$$\int_V \frac{dv_{k,i}}{dr} \times \frac{dv_{k,i}}{dr} \; rdr = \frac{2}{15} \left(\frac{r_{k-1} h_-}{D_{i-}^2} + \frac{r_{k+1} h_+}{D_{i+}^2}\right) + \frac{1}{10} \left(\frac{h_-^2}{D_{i-}^2} - \frac{h_+^2}{D_{i+}^2}\right)$$

---

*The first two terms come from integration to the left of point k and the last two from integration to the right of point k.

$$\int_V \frac{dv_{k,i}}{dr} \times \frac{dv_{k+1,i}}{dr} \, rdr = - \frac{r_k h_+}{30D_{i+}^2} - \frac{h_+^2}{60D_{i+}^2}$$

$$\int_V \frac{dv_{k,i}}{dr} \times \frac{dv_{k-1,i}}{dr} \, rdr = - \frac{h_- r_k}{30D_{i-}^2} + \frac{h_-^2}{60D_{i-}^2}$$

$$\int_V \frac{dw_k}{dr} \times \frac{dv_{k,i}}{dr} \, rdr = \frac{r_k}{10} \left(\frac{1}{D_{i+}} - \frac{1}{D_{i-}}\right) + \frac{1}{10} \left(\frac{h_+}{D_{i+}} + \frac{h_-}{D_{i-}}\right)$$

$$\int_V \frac{dw_k}{dr} \times \frac{dv_{k+1,i}}{dr} \, rdr = \frac{r_k}{10D_{i+}}$$

$$\int \frac{dw_k}{dr} \times \frac{dv_{k-1,i}}{dr} \, rdr = - \frac{r_k}{10D_{i-}}$$

The solution of the linearized multigroup diffusion equations (Section 2.3) gives the coefficients $a_{k,i}$ and $\beta_{k,i}$ of the piecewise polynomials in Eq. (2.20) as a function of $u_f^*$ and $u_m^*$. Thus when integral quantities involving $\phi_i^*$, such as the breeding gain (Eq. 2.8) and the total power (Eq. 2.11), are calculated, the evaluation of integrals $w_k$ and $v_{k,i}$ is required. These integrals are as follows:

$$\int_{r_{k-1}}^{r_k} w_k r dr = \frac{7h_-^2}{20} + 0.5\, r_{k-1} h_-$$

$$\int_{r_k}^{r_{k+1}} w_k \, r dr = -\frac{7h_+^2}{20} + 0.5\, r_{k+1} h_+$$

$$\int_{r_{k-1}}^{r_k} v_{k,i} \, r dr = -\frac{h_-^2 r_{k-1}}{12D_{i-}} - \frac{h_-^3}{20D_i}$$

$$\int_{r_k}^{r_{k+1}} v_{k,i} r dr = \frac{r_{k+1} h_+^2}{12\, D_{i+}} - \frac{h_+^3}{20\, D_{i+}}$$

All the other required integrations were carried out numerically by using Simpson's rule. The integration step size was chosen such as to keep the error of numerical integration less than about $1 \times 10^{-5}$.

# Appendix D

# THE COMPUTER PROGRAM GREKO

## D.1 INTRODUCTION

In this Appendix the computer program written to carry out the computations is discussed and listed. This program is not intended for use as a production program, and hence has not been groomed to minimize storage requirements or running time. It is written in Fortran IV language for the M.I.T. IBM 360/65 computer.

The program consists of four main parts. In the first part the multigroup diffusion equations and the adjoint multigroup diffusion equations are solved to compute the reactor eigenvalue, the neutron fluxes and their adjoints. This part is based on the multigroup diffusion program DIFFUSE written by W. H. Reed at M.I.T. In the second part the coefficients of $(u_f - u_f^0)$ and $(u_m - u_m^0)$ in Eq. (2.13) are computed by using multigroup diffusion perturbation theory. In the third part the linearized multigroup diffusion equations (Eqs. 2.10) are solved to express $\phi_i^*$ as a function of $u_{f,j}$, $u_{m,j}$, $(j = 1,R)$. The subroutine DMINV of this part is based on the subroutine MINV of IBM. In the fourth part the Linear Programming algorithm is used to determine the optimum material distribution which leads to a maximum or minimum value of the objective function. The subroutine SIMPLE of this part is based on the subroutine SIMPLE of RAND Corporation. The first two parts can be used independently of the rest of the program.

For example the case studies of Chapter 4 were done by using only these two parts. In such cases one should put a CALL EXIT card after the card CALL AEDIT of the MAIN (see listing).

The program is dimensioned for the following maximum problem sizes: 200 mesh intervals, 10 compositions, 5 regions, and 5 neutron groups. If only the first two parts of the program are used then the maximum number of regions can be raised to 10. The number of mesh intervals in each region must be of the form 2 x $\ell$ where $\ell$ is an even number. In subroutine BIGMAT the dimensions of the arrays G, LW, MW and the first dimension of the array F must have the value

4*NRG*NGP-1 where:

NRG = number of regions

NGP = number of neutron groups

The same value must also be assigned to the first dimension of the array WK in the COMMON/COWE/ which is contained in the subroutines BASE, BIGMAT, WENDO, BASINT and LINPRO.

The running time is proportional to the number of iterations required to go from the starting configuration to the optimum configuration. The number of iterations depends on how close the initial configuration is to the optimum configuration and on the value of the parameter $\varepsilon$ (Eqs. 2.26, 2.27). The value of the parameter $\varepsilon$ is chosen such that the $u_{f,j}$, (j = 1,R) remain close enough to $u^0_{f,j}$ (Section 2.4). Optimization of the value of this parameter minimizes the number of iterations required for a given initial configuration. In this study the parameter $\varepsilon$ was not optimized. Typical running times for the results presented in Chapter 3 are of the order of 30 minutes.

## D.2 INPUT

Using the nomenclature of the program listing a card-by-card description of the required input is as follows:

Card #1 FORMAT (20A4)

| | |
|---|---|
| TITLE (I), I = 1,20 | Problem title |

Card #2 FORMAT (16I5)

| | |
|---|---|
| NGP | Number of neutron groups |
| NRG | Number of regions |
| NMAT | Number of isotopes or materials |

Card #3 FORMAT (7G10.0)

| | |
|---|---|
| TH(J), J=1, NRG | Thickness of regions (cm) |

Card #4 FORMAT (I5, 5X, 4F15.0/4F15.0/2F15.0)

Repeat card #5 NMAT times

| | |
|---|---|
| IDMAT(I) | ID number of i-th nuclide |
| CONC(I,J), J=1, NRG | Concentration of i-th nuclide (atoms x $cm^{-3}$ x $10^{-24}$) in each region |

Card #5 FORMAT (16I5)

| | |
|---|---|
| NPT(J), J=1, NRG | Number of mesh points assigned to each region |

Card #6 FORMAT (3F15.0, 2I5)

| | |
|---|---|
| EPS1 | Convergence criterion on eigenvalue in inner iteration (recommended value 1.0 x $10^{-4}$) |

| | |
|---|---|
| EPS2 | Convergence criterion on eigenvalue in outer iteration (recommended value $1.0 \times 10^{-5}$) |
| EPS3 | Convergence criterion for flux (recommended value $1.0 \times 10^{-8}$) |
| ITMAXO | Maximum number of outer iterations (typical value 10) |
| ITMAXI | Maximum number of inner iterations (typical value 20) |

Repeat cards #7 through #12 as a unit NMAT times

Card #7 FORMAT (16I5)

| | |
|---|---|
| MMM | Material ID number |
| M1 | = 0, non-fissionable material |
| | = 1, fissionable material |

Card #8 FORMAT (7G10.0)

| | |
|---|---|
| SIGC(JJ,J), J=1, NGP | Total microscopic absorption cross section of material JJ in group J (capture + fission), barns |

Card #9 FORMAT (7G10.0)

| | |
|---|---|
| SIGTR(JJ,J), J=1, NGP | Microscopic transport cross section of material JJ in group J, barns |

Card #10 FORMAT (7G10.0)

Skip if M1 = 0 for this material

| | |
|---|---|
| XNU(JJ,J), J=1, NGP | Fission neutron yield, $\nu$, of material JJ in group J |

Card #11 FORMAT (7G10.0)

Skip if M1=0 for this material

| | |
|---|---|
| SIGF(JJ,J), J=1, NGP | Microscopic fission cross section of material JJ in group J, barns |

Card #12 FORMAT (7G10.0)

Repeat this card NGP times for each material

| | |
|---|---|
| SIGGG(JJ,K,J), J=1, NGP<br>K=1, NGP | Microscopic scattering cross section from group K to J (barns). Give for all groups J from K=1, then for all groups J from K=2, etc. |

Card #13 FORMAT (7G10.0)

| | |
|---|---|
| SPECT(J), J=1, NGP | Fission spectrum (i.e. group value of $\chi$) |

Card #14 FORMAT (F10.0), 2I5)

| | |
|---|---|
| VNO | Volume fraction of fissile material + volume fraction of fertile material |
| NPR | Problem type:<br>= 1, Breeding Optimization<br>= 2, Sodium Void Reactivity Optimization<br>= 3, Critical Mass Optimization |
| NCR | Number of core regions |

Card #15 FORMAT (I5)

Skip if NPR not equal to 2

| | |
|---|---|
| IDNA | ID number of sodium |

Card #16 FORMAT (2I5)

| | |
|---|---|
| IP | ID number of fissile material |
| IU | ID number of fertile material |

Card #17 FORMAT (2F15.0)

| | |
|---|---|
| CONCP(IP) | Concentration of pure fissile material (atoms x $cm^{-3}$ x $10^{-24}$) |
| CONCP(IU) | Concentration of pure fertile material (atoms x $cm^{-3}$ x $10^{-24}$) |

Card #18 FORMAT (7F10.0)

| | |
|---|---|
| UO(L), L=1, NCR | Volume fraction of fissile material in region L |

Card #19 FORMAT (2F10.0)

| | |
|---|---|
| PDL | Power density upper limit (Eq. 3.1) |
| THUO | Value of parameter ε (Eqs. 2.26, 2.27) (Typical value 0.002) |

## D.3 OUTPUT

The output from the program has all entries clearly identified by an appropriate heading using the terminology and nomenclature of this study. The following information is given:

1. Number of energy groups (Input)
2. Number of regions (Input)
3. Number of materials (Input)
4. Problem geometry (Cylinder)
5. Region thickness (Input)
6. Material concentrations (Input)
7. Number of mesh points (Input)
8. Fission spectrum (Input)

9. Cross sections (Input)
10. Concentrations of pure fissile and fertile materials (Input)
11. k-effective
12. k-effective of sodium voided reactor (if NPR=2)
13. Total breeding gain
14. Internal breeding gain
15. External breeding gain
16. Peak power density in each region
17. Total power
18. Neutron flux for each energy group and for each space point (only for the first iteration)
19. Adjoint flux for each energy group and for each space point (only for the first iteration)
20. Critical mass (if NPR=3)
21. Feasibility. If the value of this parameter is equal to zero the problem is feasible, if it is equal to 1 the problem is infeasible.
22. Fissile volume fractions given by the Linear Programming solution
23. Number of iterations

## D.4 LISTING

```
C
C          *******   *****
C          PROGRAM   GREKO
C          *******   *****
C
C      MAIN PROGRAM
       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
      1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
      2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
      3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
      4 SGG(10,5,5),DI(10,5)
       COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
      1  C(201,5) ,W(201,5,5)  ,S(201,5)
       COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
      1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
      2IPVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
      3JDUM,IHOLD(90)
       COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
      1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
       COMMON /ERR/IERR
       COMMON/CONV/ CRMA(30),NPR,KNA,NCR,IDNA
       COMMON/ITER/ NIT
       DIMENSION CONCN(8)
       ABS(ZZ)=DABS(ZZ)
       KNA=0
       NIT=
  1500 IERR=0
       RK1=0.0
       TK=0.0
       BIG=1000.0
       IAJ=
       EFFK=1.0
       KEEP=0
       CC=1.
       ITO=1
```

```
      ITI=0                                                             MAIN0037
      IF(NIT.NE.0) GO TO 1510                                           MAIN0038
      IF(KNA.EQ.1) GO TO  1510                                          MAIN0039
      CALL INDATA                                                       MAIN0040
 1510 IF(IERR.NE.0)GO TO 9999                                           MAIN0041
      CALL MACROX                                                       MAIN0042
      CALL FLUXIN                                                       MAIN0043
      IF(IERR.NE.0)GO TO 9999                                           MAIN0044
 1    CALL XSECT                                                        MAIN0045
 7    CALL TRIDIA                                                       MAIN0046
 5    CALL WEIGHT                                                       MAIN0047
 2    CALL SOLVE                                                        MAIN0048
      CALL RESCAL                                                       MAIN0049
      ITI=ITI+1                                                         MAIN0050
      IF(ITI.GT.ITMAXI) CALL ERROR(1)                                   MAIN0051
      IF(IERR.NE.0)GO TO 9999                                           MAIN0052
      IF(ABS(RK2-TK).LT.EPS1) GO TO 3                                   MAIN0053
      TK=RK2                                                            MAIN0054
      GO TO 2                                                           MAIN0055
 3    IF (ABS(RK2-1.0).LT.EPS2) GO TO 6                                 MAIN0056
      CALL ADJUST                                                       MAIN0057
      IHOLD(ITO)=ITI                                                    MAIN0058
      AHOLD(ITO)=RK2                                                    MAIN0059
      RK1=RK2                                                           MAIN0060
      ITO=ITO+1                                                         MAIN0061
      IF(ITO.GT.ITMAXO) CALL ERROR(6)                                   MAIN0062
      IF(IERR.NE.0)GO TO 9999                                           MAIN0063
      ITI=0                                                             MAIN0064
      GO TO (5,7,1),IOP                                                 MAIN0065
 6    IF (BIG.LT.EPS3) GO TO 100                                        MAIN0066
      BIG=0.0                                                           MAIN0067
      KEEP=1                                                            MAIN0068
      GO TO 2                                                           MAIN0069
 100  IF(JAD.EQ.0)GO TO 9999                                            MAIN0070
      DO 10 J=1,NP                                                      MAIN0071
      DO 10 I=1,NGP                                                     MAIN0072
```

```
      PHL(J,I)=PHI(J,I)
10    CONTINUE
      CALL AJOINT
      CALL  ISCHIS
      IF(NPR.NE.2) GO TO 25
      IF(KNA.EQ.1) GO TO 1531
      IF(NIT.NE.0) GO TO 1530
      DO 20 K=1,NCR
20    CONCN(K)=CONC(IDNA,K)
25    IF(NIT.NE.0) GO TO 1530
      CALL EDIT
      CALL AEDIT
1530  CALL BASE
      CALL  BIGMAT
1531  CALL LINPRO
      IF(NPR.NE.2) GO TO 1540
      IF(KNA.EQ.0) GO TO 1545
      DO 15 K=1,NCR
15    CONC(IDNA,K)=0.0
      GO TO 1540
1545  DO 21 K=1,NCR
21    CONC(IDNA,K)=CONCN(K)
1540  GO TO 1500
9999  CALL EXIT
      END
```

```
      SUBROUTINE INDATA
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),
     1  SIGGG(10,5,5)
      COMMON /ERR/IERR
      REAL TITLE
      DIMENSION TITLE(20)
      NGEOM=2
      JBCL=1
      JBCR=0
      IOP=1
      READ (5,9901) (TITLE(J),J=1,20)
      WRITE(6,991) (TITLE(J),J=1,20)
      READ (5,992) NGP,NRG,NMAT
      WRITE(6,993) NGP,NRG,NMAT
      IF (NGEOM.EQ.1) WRITE (6,994)
      IF (NGEOM.EQ.2) WRITE (6,995)
      IF (NGEOM.EQ.3) WRITE (6,996)
      READ (5,997) (TH(J),J=1,NRG)
      WRITE (6,9913) (J,TH(J),J=1,NRG)
      DO 1 I=1,NMAT
      READ (5,999) IDMAT(I),(CONC(I,J),J=1,NRG)
1     CONTINUE
      WRITE (6,9900)
      DO 2 I=1,NMAT
      WRITE (6,9902) IDMAT(I),(J,CONC(I,J),J=1,NRG)
2     CONTINUE
      READ (5,992) (NPT(J),J=1,NRG)
```

```
      WRITE (6,9903) (J,NPT(J),J=1,NRG)
      READ (5,9912) EPS1,EPS2,EPS3,ITMAXO,ITMAXI
      DO 3 I=1,NMAT
      READ (5,992) MMM,M1,M2
      DO 4 J=1,NMAT
      JJ=J
      IF (MMM.EQ.IDMAT(J)) GO TO 5
4     CONTINUE
      CALL ERROR (2)
      IF(IERR.NE.0) RETURN
 5    READ (5,997) (SIGC(JJ,J),J=1,NGP)
      READ (5,997) (SIGTR(JJ,J),J=1,NGP)
      IF (M1.EQ.1) GO TO 7
      DO 8 J=1,NGP
      XNU(JJ,J)=0.0
      SIGF(JJ,J)=0.0
8     CONTINUE
      GO TO 9
7     READ (5,997)(XNU(JJ,J),J=1,NGP)
      READ (5,997) (SIGF(JJ,J),J=1,NGP)
9     DO 6 K=1,NGP
      READ (5,997)(SIGGG(JJ,K,J),J=1,NGP)
6     CONTINUE
3     CONTINUE
      READ (5,997) (SPECT(J),J=1,NGP)
      WRITE (6,9904) (SPECT(J),J=1,NGP)
      DO 10 I = 1,NMAT
      WRITE (6,9905) IDMAT(I)
      WRITE (6,9906) (SIGC( I,J),J=1,NGP)
      WRITE (6,9907) (SIGF( I,J),J=1,NGP)
      WRITE (6,9911) (XNU(I,J),J=1,NGP)
      WRITE (6,9908) (SIGTR( I,J),J=1,NGP)
      DO 10 K=1,NGP
      WRITE (6,9909) K,(SIGGG(I,K,J),J=1,NGP)
10    CONTINUE
 991  FORMAT (20X,20A4)
```

```
992  FORMAT (16I5)                                                            INDA0073
993  FORMAT (///' NUMBER OF ENERGY GROUPS =',I10,//' NUMBER OF REGIONS        INDA0074
    1=',I10,//' NUMBER OF MATERIALS =',I10,//)                                INDA0075
994  FORMAT (////' PROBLEM GEOMETRY = SLAB')                                  INDA0076
995  FORMAT (////' PROBLEM GEOMETRY = CYLINDER')                              INDA0077
996  FORMAT (////' PROBLEM GEOMETRY = SPHERE')                                INDA0078
997  FORMAT (7G10.0)                                                          INDA0079
998  FORMAT (////10X,'GROUP',10X,'LOWER ENERGY BOUND',/10X,'-----',10X,       INDA0080
    1'------------------',//(10X,I3,10X,D15.5))                               INDA0081
999  FORMAT(I5,5X,4F15.0,/4F15.0,/2F15.0)                                     INDA0082
9900 FORMAT (////10X,'MATERIAL',40X,'REGION / CONCENTRATION',//)              INDA0083
9901 FORMAT (20A4)                                                            INDA0084
9902 FORMAT(/1X,I2,8(I2,' / ',F13.10),/10X,8(I4,' / ',F15.10))                INDA0085
9903 FORMAT (///' REGION / NUMBER OF MESH POINTS',//10(I5,' / ',I3))          INDA0086
9904 FORMAT (///' FISSION SPECTRUM'/(8F15.10))                                INDA0087
9905 FORMAT ('1 CROSS SECTIONS FOR MATERIAL',I10,///)                         INDA0088
9906 FORMAT (' CAPTURE CROSS SECTION',/(8F15.10))                             INDA0089
9907 FORMAT (' FISSION CROSS SECTION',/(8F15.10))                             INDA0090
9908 FORMAT (' TRANSPORT CROSS SECTION',/(8F15.10))                           INDA0091
9909 FORMAT (' TRANSFER CROSS SECTION FROM GROUP',I5,/(8F15.10))              INDA0092
9911 FORMAT (' NU',/(8F15.6))                                                 INDA0093
9912 FORMAT (3F15.0,2I5)                                                      INDA0094
9913 FORMAT(///' REGION / REGION THICKNESS IN CM',/6(5X,I2,' / ',F8.4))       INDA0095
     RETURN                                                                   INDA0096
     END                                                                      INDA0097
```

```
      SUBROUTINE MACROX
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSOIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),
     1  SIGGG(10,5,5)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      DO 4 I=1,NRG
      DO 4 J=1,NGP
      SR(I,J)=0.0
      XA(I,J)=0.0
      SIGFM(I,J)=0.0
      XNUF(I,J)=0.0
      XTR(I,J)=0.0
      DO 4 K=1,NGP
      XGG(I,J,K)=0.0
    4 CONTINUE
      DO 3 J=1,NMAT
      DO 3 I=1,NGP
      DO 3 K=1,NRG
      XA(K,I)=XA(K,I)+CONC(J,K)*SIGC(J,I)
      SIGFM(K,I)=SIGFM(K,I)+CONC(J,K)*SIGF(J,I)
      XNUF(K,I)=XNUF(K,I)+CONC(J,K)*SIGF(J,I)*XNU(J,I)
      XTR(K,I)=XTR(K,I)+CONC(J,K)*SIGTR(J,I)
      DO 3 L=1,NGP
      XGG(K,I,L)=XGG(K,I,L)+CONC(J,K)*SIGGG(J,I,L)
```

```
3     CONTINUE                                         MACRO0037
      DO 2 I=1,NRG                                     MACRO0038
      DO 2 J=1,NGP                                     MACRO0039
2     XGG(I,J,J)=0.0                                   MACRO0040
      DO 5 K=1,NRG                                     MACRO0041
      DO 5 I=1,NGP                                     MACRO0042
      SA(K,I)=XA(K,I)                                  MACRO0043
      SNUF(K,I)=XNUF(K,I)                              MACRO0044
      STR(K,I)=XTR(K,I)                                MACRO0045
      DI(K,I)=1.0/(3.0*STR(K,I))                       MACRO0046
      DO 5 L=1,NGP                                     MACRO0047
      SGG(K,I,L)=XGG(K,I,L)                            MACRO0048
      SR(K,I)=SR(K,I)+XGG(K,I,L)                       MACRO0049
5     CONTINUE                                         MACRO0050
      RETURN                                           MACRO0051
      END                                              MACRO0052
```

```
      SUBROUTINE FLUXIN
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      COMMON /ERR/IERR
      SQRT(ZZ)=DSQRT(ZZ)
      NP=1
      DO 1 J=1,NRG
1     NP=NP+NPT(J)
      DO 2 L=1,NGP
      DO 2 J=1,NP
2     PHI(J,L)=1.0
20    ANORM=SQRT(1.0D0*NP*NGP)
      GO TO (6,6,5),IOP
5     DO 3 J=1,NMAT
      JJ=J
      IF (IDMAT(J).EQ.MVARY) GO TO 4
3     CONTINUE
      CALL ERROR(4)
      IF(IERR.NE.0) RETURN
4     MCODE=JJ
6     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XSECT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),
     1  SIGGG(10,5,5)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)   ,S(201,5)
      DIMENSION F(10)
      GO TO (6,6,5),IOP
    5 Z=(CC-1.0)/CC
      DO 1 K=1,NRG
    1 F(K)=CONC(MCODE,K)*Z
      DO 2 K=1,NRG
      DO 2 I=1,NGP
      XA(K,I)=XA(K,I)+F(K)*SIGC(MCODE,I)
      XNUF(K,I)=XNUF(K,I)+F(K)*SIGF(MCODE,I)*XNU(MCODE,I)
      XTR(K,I)=XTR(K,I)+F(K)*SIGTR(MCODE,I)
      DO 2 L=1,NGP
      XGG(K,I,L)=XGG(K,I,L)+F(K)*SIGGG(MCODE,I,L)
    2 CONTINUE
    6 CONTINUE
      DO 3 I=1,NRG
      DO 3 J=1,NGP
    3 XR(I,J)=XA(I,J)
      DO 4 I=1,NRG
      DO 4 J=1,NGP
      D(I,J)=1.0/(3.0*XTR(I,J))
      DO 4 K=1,NGP
      XR(I,J)=XR(I,J)+XGG(I,J,K)
    4 CONTINUE
      RETURN
```

END XSEC0037

```
      SUBROUTINE TRIDIA
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      DO 3 L=1,NGP
      DO 3 J=1,NP
    3 B(J,L)=0.0
      RWR=0.0
      JJ=0
      DO 1 K=1,NRG
      H=TH(K)/NPT(K)
      HI=1.0/H
      R=RWR-H*0.5
      JMAX = NPT(K)
      DO 2 J=1,JMAX
      JJ=JJ+1
      R=R+H
      IF (NGEOM.EQ.1) RP=1.0
      IF (NGEOM.EQ.2) RP=R
      IF (NGEOM.EQ.3) RP=R*R
      DO 2 L=1,NGP
      A(JJ,L)=RP*D(K,L)*HI
      C(JJ,L)=A(JJ,L)
      Z=A(JJ,L)+((R-H/2.0)**(NGEOM-1))*XR(K,L)*H*0.5
      Z1=A(JJ,L)+((R+H/2.0)**(NGEOM-1))*XR(K,L)*H*0.5
      B(JJ,L)=B(JJ,L)-Z
      B(JJ+1,L)=B(JJ+1,L)-Z1
    2 CONTINUE
      RWR=RWR+TH(K)
    1 CONTINUE
```

```
      IF (JBCL.EQ.0) GO TO 4                                            TRID0037
      DO 5 L=1,NGP                                                      TRID0038
      A(1,L)=2.0*A(1,L)                                                 TRID0039
    5 B(1,L)=-A(1,L)                                                    TRID0040
      GO TO 6                                                           TRID0041
    4 DO 7 L=1,NGP                                                      TRID0042
      B(1,L)=1.0                                                        TRID0043
    7 A(1,L)=0.0                                                        TRID0044
    6 IF (JBCR.EQ.0) GO TO 8                                            TRID0045
      RP = RWR ** (NGEOM-1)                                             TRID0046
      DO 9 L=1,NGP                                                      TRID0047
    9 C(NP-1,L)=2.0*C(NP-1,L)                                           TRID0048
      B(NP,L) = 2.0*B(NP,L)-RP*XR(NRG,L)*H                              TRID0049
      GO TO 10                                                          TRID0050
    8 DO 11 L=1,NGP                                                     TRID0051
      B(NP,L)=1.0                                                       TRID0052
   11 C(NP-1,L)=0.0                                                     TRID0053
   10 CONTINUE                                                          TRID0054
      RETURN                                                            TRID0055
      END                                                               TRID0056
```

```
      SUBROUTINE WEIGHT                                                 WEIG0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         WEIG0002
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),  WEIG0003
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,       WEIG0004
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEFP,MCODE,LBIG,JBIG,IAJ,  WEIG0005
     3JDUM,IHOLD(90)                                                    WEIG0006
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5), WEIG0007
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)                      WEIG0008
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,        WEIG0009
     1  C(201,5) ,W(201,5,5)   ,S(201,5)                                WEIG0010
      JJ=1                                                              WEIG0011
      RWR=0.0                                                           WEIG0012
      DO 1 I=1,NRG                                                      WEIG0013
      H=TH(I)/NPT(I)                                                    WEIG0014
      JMAX=NPT(I)-1                                                     WEIG0015
      R=RWR                                                             WEIG0016
      DO 2 J=1,JMAX                                                     WEIG0017
      JJ=JJ+1                                                           WEIG0018
      R=R+H                                                             WEIG0019
      IF (NGEOM.EQ.1) RP=1.0                                            WEIG0020
      IF (NGEOM.EQ.2) RP=R                                              WEIG0021
      IF (NGEOM.EQ.3) RP=R*R                                            WEIG0022
      F=H*RP                                                            WEIG0023
      DO 3 L=1,NGP                                                      WEIG0024
      DO 3 K=1,NGP                                                      WEIG0025
    3 W(JJ,K,L)=(XGG(I,K,L)+EFFK*SPECT(L)*XNUF(I,K))*F                  WEIG0026
    2 CONTINUE                                                          WEIG0027
      IF (I.EQ.NRG) GO TO 1                                             WEIG0028
      JJ=JJ+1                                                           WEIG0029
      R=R+H                                                             WEIG0030
      RP=R**(NGEOM-1)                                                   WEIG0031
      H2=TH(I+1)/NPT(I+1)                                               WEIG0032
      F=0.5*RP                                                          WEIG0033
      DO 4 L=1,NGP                                                      WEIG0034
      DO 4 K=1,NGP                                                      WEIG0035
    4 W(JJ,K,L)=((XGG(I,K,L)+EFFK*SPECT(L)*XNUF(I,K))*H+(XGG(I+1,K,L)+  WEIG0036
```

```
     1  EFFK*SPECT(L)*XNUF(I+1,K))*H2)*F
      RWR=RWR+TH(I)
 1    CONTINUE
      H1=TH(1)/NPT(1)
      H2=TH(NRG)/NPT(NRG)
      RWR=RWR+TH(NRG)
      RP=RWR**(NGEOM-1)
      F1=H1*(1/IOP)
      F2=H2*RP
      DO 5 L=1,NGP
      DO 5 K=1,NGP
      IF (JBCL) 6,6,7
 6    W(1,K,L)=0.0
      GO TO 8
 7    W(1,K,L)=0.0
 8    IF (JBCR) 9,9,10
 9    W(NP,K,L)=0.0
      GO TO 5
 10   W(NP,K,L)=(XGG(NRG,K,L)+EFFK*SPECT(L)*XNUF(NRG,K))*F2
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOURCE(L)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      DO 1 J=1,NP
    1 S(J,L)=0.0
      DO 2 J=1,NP
      DO 2 K=1,NGP
      IF (IAJ.EQ.0) GO TO 3
      IF (IAJ.EQ.1) GO TO 4
    3 S(J,L)=S(J,L)-W(J,K,L)*PHI(J,K)
      GO TO 2
    4 S(J,L)=S(J,L)-W(J,L,K)*PHI(J,K)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MATINV(X,DL,DD,DU,Y,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1),DL(1),DD(1),DU(1),Y(1),WA(201),GA(201)
      WA(1)=DU(1)/DD(1)
      GA(1)=Y(1)/DD(1)
      DO 1 K=2,N
      T1=1.0/(DD(K)-DL(K-1)*WA(K-1))
      WA(K)=DU(K)*T1
      GA(K)=(Y(K)-DL(K-1)*GA(K-1))*T1
1     CONTINUE
      X(N)=GA(N)
      KMAX=N-1
      DO 2 K=1,KMAX
      J=N-K
      X(J)=GA(J)-WA(J)*X(J+1)
2     CONTINUE
20    RETURN
      END
```

```
      SUBROUTINE SOLVE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      DIMENSION U(201),XL(201),D(201),X(201),Y(201)
      FMAX=0.0
      DO 1 L=1,NGP
      CALL SOURCE(L)
      DO 2 J=1,NP
      U(J)=A(J,L)
      XL(J)=C(J,L)
      D(J)=B(J,L)
      Y(J)=S(J,L)
2     CONTINUE
      CALL MATINV(X,XL,D,U,Y,NP)
      IF (KEEP) 5,5,3
    3 JJJ=NP-1
      DO 4 J=2,JJJ
      CC=X(J)/PHI(J,L) - 1.0D+00
      F=DABS(CC)
      IF(FMAX.GT.F)GO TO 4
      LBIG=L
      JBIG=J
      BIG=PHI(J,L)
4     CONTINUE
5     DO 1 J=1,NP
      PHI(J,L)=X(J)
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RESCAL                                                 RESC0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         RESC0002
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),  RESC0003
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,       RESC0004
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,  RESC0005
     3JDUM,IHOLD(90)                                                    RESC0006
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5), RESC0007
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)                      RESC0008
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,        RESC0009
     1  C(201,5) ,W(201,5,5)  ,S(201,5)                                 RESC0010
      ABS(ZZ)=DABS(ZZ)                                                  RESC0011
      SQRT(ZZ)=DSQRT(ZZ)                                                RESC0012
      BNORM=0.0                                                         RESC0013
      DO 1 J=1,NP                                                       RESC0014
      DO 1 L=1,NGP                                                      RESC0015
    1 BNORM=BNORM+PHI(J,L)*PHI(J,L)                                     RESC0016
      BNORM=SQRT(BNORM)                                                 RESC0017
      DNORM=ANORM/BNORM                                                 RESC0018
      DO 2 J=1,NP                                                       RESC0019
      DO 2 L=1,NGP                                                      RESC0020
    2 PHI(J,L)=PHI(J,L)*DNORM                                           RESC0021
      RK2=BNORM/ANORM                                                   RESC0022
      IF(KEEP)3,3,4                                                     RESC0023
    4 BIG=ABS(PHI(JBIG,LBIG)-BIG)                                       RESC0024
    3 CONTINUE                                                          RESC0025
      RETURN                                                            RESC0026
      END                                                               RESC0027
```

```
      SUBROUTINE ADJUST
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      ALPHA=1.1
      GO TO (1,2,3),IOP
    1 EFFK=(1.0+ALPHA*(1.0-RK2))*EFFK
      RETURN
    2 CT=1.0+ALPHA*(1.0-RK2)
    5 DO 6 J=1,NRVARY
      JJ=IRVARY(J)
      TH(JJ)=CT*TH(JJ)
    6 CONTINUE
      RETURN
    3 IF (ITO.EQ.0) GO TO 7
      CC=1.0+((CC-1.0)/CC)*((1.0-RK2)/(RK2-RK1))
      GO TO 8
    7 CC=1.1
    8 DO 9 J=1,NRG
      CONC(MCODE,J)=CC*CONC(MCODE,J)
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ERROR(N)                                               ERR00001
      IMPLICIT REAL*8 (A-H,O-Z)                                         ERR00002
      COMMON /ERR/IERR                                                  ERR00003
      WRITE (6,1) N                                                     ERR00004
    1 FORMAT ('1      ERROR STOP NUMBER',I5)                            ERR00005
      IERR=1                                                            ERR00006
      RETURN                                                            ERR00007
      END                                                               ERR00008
```

```
      SUBROUTINE EDIT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),
     1  SIGGG(10,5,5)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      WRITE (6,991)
      WRITE (6,996)
      WRITE (6,997) (I,IHOLD(I),AHOLD(I),I=1,ITO)
      GO TO (1,2,3),IOP
    1 EFFK1=1.0/EFFK
      WRITE (6,993) ((I,J,PHL(I,J),I=1,NP),J=1,NGP)
      WRITE (6,992) EFFK1
      RETURN
 2    WRITE (6,992)RK2
      WRITE (6,995) (J,TH(J),J=1,NRG)
      WRITE (6,993) ((I,J,PHL(I,J),I=1,NP),J=1,NGP)
      RETURN
 3    WRITE (6,992)RK2
      WRITE (6,994) MVARY,(J,CONC(MCODE,J),J=1,NRG)
      WRITE (6,993) ((I,J,PHL(I,J),I=1,NP),J=1,NGP)
      RETURN
 991  FORMAT ('1',20X,'PROGRAM EDIT')
  992 FORMAT (///'  K EFFECTIVE = ',F10.6)
  993 FORMAT (///' I,J,PHL(I,J) . . . I=SPACE POINT, J=GROUP'/6(I5,I3,D1
```

```
   12.5))
994  FORMAT(///' CRITICAL CONCENTRATION OF MATERIAL',I5,/(' REGION =',
   1 I5,10X,'CONCENTRATION =',F10.7))
995  FORMAT(///' CRITICAL SIZE',/('THICKNESS OF REGION',I5,'  =',F10.5,
   1  '    CM'))
 996 FORMAT (///' OUTER ITERATION   NUMBER OF INNER ITERATIONS   EIGENV
   1ALUE',//)
 997 FORMAT (I7,20X,I5,15X,F10.6)
     END
```

```
      SUBROUTINE AJOINT                                                 AJOI0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         AJOI0002
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),  AJOI0003
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,       AJOI0004
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,  AJOI0005
     3JDUM,IHOLD(90)                                                    AJOI0006
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5), AJOI0007
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)                      AJOI0008
      COMMON /ERR/IERR                                                  AJOI0009
      ABS(ZZ)=DABS(ZZ)                                                  AJOI0010
      IAJ=1                                                             AJOI0011
      RK1=0.0                                                           AJOI0012
      TK=0.0                                                            AJOI0013
      BIG=1000.0                                                        AJOI0014
      KEEP=0                                                            AJOI0015
      ITI=0                                                             AJOI0016
    2 CALL SOLVE                                                        AJOI0017
      CALL RESCAL                                                       AJOI0018
      ITI=ITI+1                                                         AJOI0019
      IF(ITI.GT.ITMAXI) CALL ERROR(1)                                   AJOI0020
      IF(IERR.NE.0) RETURN                                              AJOI0021
      IF(ABS(RK2-TK).LT.EPS1) GO TO 6                                   AJOI0022
      TK=RK2                                                            AJOI0023
      GO TO 2                                                           AJOI0024
    6 IF (BIG.LT.EPS3) GO TO 100                                        AJOI0025
      BIG=0.0                                                           AJOI0026
      KEEP=1                                                            AJOI0027
      GO TO 2                                                           AJOI0028
  100 RETURN                                                            AJOI0029
      END                                                               AJOI0030
```

```
      SUBROUTINE AEDIT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(15),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,
     1  C(201,5) ,W(201,5,5)  ,S(201,5)
      WRITE (6,991)
      WRITE (6,993) ((I,J,PHI(I,J),I=1,NP),J=1,NGP)
  991 FORMAT ('1',20X,'ADJOINT EDIT')
  993 FORMAT(///' I,J,PHIA(I,J)      I=SPACE POINT, J=GROUP',/6(I5,I3,D12
     1.5))
      RETURN
      END
```

```
      SUBROUTINE WINI
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLTM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT, IDMAT(10)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,IBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON/DELTA/ THSA(10,5),THNSF(10,5),THD(10,5),THST(10,5,5),
     1 THTRP(10,5),THSTT(10,5),DSAM(10,5),DNSFM(10,5),DDM(10,5),
     2 DSTM(10,5,5),DTRPM(10,5),DSTTM(10,5),THSF(10,5),DSFM(10,5),
     3 SFU(10,5),SCU(10,5),SUP(10,5),POWED(10),CONCP(10),VNO
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),
     1  SIGGG(10,5,5)
      COMMON/KSWY/ SB(10,5),PPU(10),PU(10),PDU(10),PRS(10),URN(10),
     1 URC(10),SD(10),DOPL(10)
      COMMON/CONV/ CRMA(30),NPR,KNA,NCR,IDNA
      COMMON/DELFI/ IP,IU
      COMMON/ITER/ NIT
      NCE=NCR+1
      IF(KNA.EQ.1) GO TO 2
      IF(NIT.NE.0) GO TO 2
      READ(5,2500) VNO,NPR,NCR
 2500 FORMAT(F10.0,2I5)
      IF(NPR.NE.2) GO TO 1
      READ(5,2520) IDNA
 2520 FORMAT(I5)
 1    READ(5,2510) IP,IU
 2510 FORMAT(2I5)
      READ(5,600) CONCP(IP),CONCP(IU)
 600  FORMAT(2F15.0)
```

```
      WRITE(6,610)
      WRITE(6,611)  CONCP(IP),CONCP(IU)
610   FORMAT(///5X,' CONCENTRATION OF PURE MATERIALS')
611   FORMAT(///2F15.8)
      DO 9 I=1,NRG
      DO 9 J=1,NGP
      THSA(I,J)=0.0
      THSF(I,J)=0.0
      THNSF(I,J)=0.0
      THTRP(I,J)=0.0
      THSTT(I,J)=0.0
      DSAM(I,J)=0.0
      DSFM(I,J)=0.0
      DNSFM(I,J)=0.0
      DTRPM(I,J)=0.0
      DSTTM(I,J)=0.0
      SFU(I,J)=0.0
      SCU(I,J)=0.0
      SUP(I,J)=0.0
      SB(I,J)=0.0
      DO 9 K=1,NGP
      THST(I,J,K)=0.0
      DSTM(I,J,K)=0.0
9     CONTINUE
      DO 11 I=1,NGP
      DO 11 K=1,NRG
      THSA(K,I)=THSA(K,I)+CONCP(IP)*SIGC(IP,I)
      THSA(K,I)=THSA(K,I)-CONCP(IU)*SIGC(IU,I)
      THSF(K,I)=THSF(K,I)+CONCP(IP)*SIGF(IP,I)
      THSF(K,I)=THSF(K,I)-CONCP(IU)*SIGF(IU,I)
      THNSF(K,I)=THNSF(K,I)+CONCP(IP)*SIGF(IP,I)*XNU(IP,I)
      THNSF(K,I)=THNSF(K,I)-CONCP(IU)*SIGF(IU,I)*XNU(IU,I)
      THTRP(K,I)=THTRP(K,I)+CONCP(IP)*SIGTR(IP,I)
      THTRP(K,I)=THTRP(K,I)-CONCP(IU)*SIGTR(IU,I)
      DO 11 L=1,NGP
      THST(K,I,L)=THST(K,I,L)+CONCP(IP)*SIGGG(IP,I,L)
```

```
      THST(K,I,L)=THST(K,I,L)-CONCP(IU)*SIGGG(IU,I,L)                    WINI0073
      THSTT(K,I)=THSTT(K,I)+THST(K,I,L)                                  WINI0074
 11   CONTINUE                                                           WINI0075
 2    DO 13 I=1,NGP                                                      WINI0076
      DO 13 K=1,NRG                                                      WINI0077
      DDM(K,I)=-DTRPM(K,I)/(3.0*(STR(K,I)*STR(K,I)))                     WINI0078
 13   THD(K,I)=-THTRP(K,I)/(3.0*(STR(K,I)*STR(K,I)))                     WINI0079
      DO 60 I=1,NGP                                                      WINI0080
      DO 61 K=1,NCR                                                      WINI0081
      SFU(K,I)= CONCP(IU)*SIGF(IU,I)                                     WINI0082
      SCU(K,I)= CONCP(IU)*(SIGC(IU,I)-SIGF(IU,I))                        WINI0083
      SUP(K,I)=SCU(K,I)+CONCP(IP)*SIGC(IP,I)                             WINI0084
      SB(K,I)=CONC(IU,K)*(SIGC(IU,I)-SIGF(IU,I))-CONC(IP,K)*SIGC(IP,I)   WINI0085
 61   CONTINUE                                                           WINI0086
      DO 60 K=NCE,NRG                                                    WINI0087
      SFU(K,I)= CONCP(IU)*SIGF(IU,I)                                     WINI0088
      SUP(K,I)=0.0                                                       WINI0089
      SB(K,I)=CONC(IU,K)*(SIGC(IU,I)-SIGF(IU,I))                         WINI0090
      SCU(K,I)=CONCP(IU)*(SIGC(IU,I)-SIGF(IU,I))                         WINI0091
 60   CONTINUE                                                           WINI0092
      DO 10 K=1,NCR                                                      WINI0093
 10   SD(K)=CONC(IU,K)*SIGC(IU,5)                                        WINI0094
      RETURN                                                             WINI0095
      END                                                                WINI0096
```

```
      SUBROUTINE  ISCHIS                                                ISCH0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         ISCH0002
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),  ISCH0003
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),  ISCH0004
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),   ISCH0005
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),           ISCH0006
     4 SGG(10,5,5),DI(10,5)                                             ISCH0007
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),  ISCH0008
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,       ISCH0009
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,  ISCH0010
     3JDUM,IHOLD(90)                                                    ISCH0011
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5), ISCH0012
     1CONC(10,10),D(10,5),XR(10,5),CC,CT, IDMAT(10)                     ISCH0013
      COMMON /ERR/IERR                                                  ISCH0014
      COMMON /MICX/ SIGC(10,5) ,SIGTR(10,5) ,XNU(10,5) ,SIGF(10,5),     ISCH0015
     1  SIGGG(10,5,5)                                                   ISCH0016
      COMMON /FLUX/ PHI(201,5) ,ANORM,BNORM,A(201,5) ,B(201,5) ,        ISCH0017
     1  C(201,5) ,W(201,5,5)  ,S(201,5)                                 ISCH0018
      COMMON/KSWY/ SB(10,5),PPU(10),PU(10),PDU(10),PRS(10),URN(10),     ISCH0019
     1 URC(10),SD(10),DOPL(10)                                          ISCH0020
      COMMON/DELTA/ THSA(10,5),THNSF(10,5),THD(10,5),THST(10,5,5),      ISCH0021
     1 THTRP(10,5),THSTT(10,5),DSAM(10,5),DNSFM(10,5),DDM(10,5),        ISCH0022
     2 DSTM(10,5,5),DTRPM(10,5),DSTTM(10,5),THSF(10,5),DSFM(10,5),      ISCH0023
     3 SFU(10,5),SCU(10,5),SUP(10,5),POWED(10),CONCP(10),VNO            ISCH0024
      COMMON/ITER/ NIT                                                  ISCH0025
      COMMON/CONV/ CRMA(30),NPR,KNA,NCR,IDNA                            ISCH0026
      DIMENSION BRA(30),SPOL(30),DOPC(30)                               ISCH0027
      EFFK1=1.0/EFFK                                                    ISCH0028
      IF(KNA.EQ.1) GO TO 10                                             ISCH0029
      WRITE (6,992) EFFK1                                               ISCH0030
  992 FORMAT (///'  K EFFECTIVE = ',F10.6)                              ISCH0031
      GO TO 11                                                          ISCH0032
 10   WRITE(6,993) EFFK1                                                ISCH0033
 993  FORMAT(///' K EFFECTIVE OF VOIDED CORE =',F10.6)                  ISCH0034
      M=NIT+1                                                           ISCH0035
      SPOL(M)=EFFK1                                                     ISCH0036
```

```
      IF(NIT.EQ.0) GO TO 11                                             ISCH0037
      IF(DABS(SPOL(M)-SPOL(M-1)).LT.0.00001)  CALL EXIT                 ISCH0038
   11 CALL WINI                                                         ISCH0039
      IF(NIT.NE.0) GO TO 12                                             ISCH0040
      AKTIS(1)=0.0                                                      ISCH0041
      DO 8 J=2,NP                                                       ISCH0042
      AKTIS(J)=AKTIS(J-1)+STS                                           ISCH0043
    8 CONTINUE                                                          ISCH0044
   12 BR=0.0                                                            ISCH0045
      BREX=0.0                                                          ISCH0046
      DOP=0.0                                                           ISCH0047
      BRU=0.0                                                           ISCH0048
      POWERT=0.0                                                        ISCH0049
      FSDIN=0.0                                                         ISCH0050
      MA=0.0                                                            ISCH0051
      NPT(1)=NPT(1)+1                                                   ISCH0052
      STSI=STS*0.33333333333333                                         ISCH0053
      DO 1 L=1,NRG                                                      ISCH0054
      TOTP(L)=0.0                                                       ISCH0055
      URN(L)=0.0                                                        ISCH0056
      URC(L)=0.0                                                        ISCH0057
      PU(L)=0.0                                                         ISCH0058
      PPU(L)=0.0                                                        ISCH0059
      PRS(L)=0.0                                                        ISCH0060
      PDU(L)=0.0                                                        ISCH0061
      POWED(L)=0.0                                                      ISCH0062
      SYLO=0.0                                                          ISCH0063
      SYLOM=0.0                                                         ISCH0064
      FISIO=0.0                                                         ISCH0065
      FISIOM=0.0                                                        ISCH0066
      FSDIO=0.0                                                         ISCH0067
      IF(L.EQ.1)GO TO 2                                                 ISCH0068
      MA=MA+NPT(L-1)                                                    ISCH0069
      K=NPT(L)-2+MA                                                     ISCH0070
      N=MA                                                              ISCH0071
      GO TO 3                                                           ISCH0072
```

```
2     K=NPT(1)-2                                                        ISCH0073
      N=1                                                               ISCH0074
3     DO 4 I=1,NGP                                                      ISCH0075
      TEMP=0.0                                                          ISCH0076
      SYL=0.0                                                           ISCH0077
      FISI=0.0                                                          ISCH0078
      FISIM=0.0                                                         ISCH0079
      FSDI=0.0                                                          ISCH0080
      DO 5 J=N,K,2                                                      ISCH0081
      TEMP=TEMP+PHL(J,I)*AKTIS(J)+4.0*PHL(J+1,I)*AKTIS(J+1)+PHL(J+2,I)  ISCH0082
     1*AKTIS(J+2)                                                       ISCH0083
      SYL=SYL+PHL(J,I)*PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,I)*PHI(J+1,I)*     ISCH0084
     1AKTIS(J+1) +PHL(J+2,I)*PHI(J+2,I)*AKTIS(J+2)                      ISCH0085
5     CONTINUE                                                          ISCH0086
      DO 6 M=1,NGP                                                      ISCH0087
      FIS=0.0                                                           ISCH0088
      FSD=0.0                                                           ISCH0089
      DO 7 J=N,K,2                                                      ISCH0090
      FIS=FIS+PHL(J,M)* PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,M)* PHI(J+1,I)*   ISCH0091
     1AKTIS(J+1)+PHL(J+2,M)* PHI(J+2,I)*AKTIS(J+2)                      ISCH0092
      FSD=FSD+PHL(J,M)* PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,M)* PHI(J+1,I)*   ISCH0093
     1AKTIS(J+1)+PHL(J+2,M)* PHI(J+2,I)*AKTIS(J+2)                      ISCH0094
7     CONTINUE                                                          ISCH0095
      FISM=FIS*DNSFM(L,M)                                               ISCH0096
      FIS=FIS*THNSF(L,M)                                                ISCH0097
      FISI=FISI+FIS                                                     ISCH0098
      FISIM=FISIM+FISM                                                  ISCH0099
      FSD=FSD*SNUF(L,M)                                                 ISCH0100
      FSDI=FSDI+FSD                                                     ISCH0101
6     CONTINUE                                                          ISCH0102
      FISI=FISI*SPECT(I)                                                ISCH0103
      FISIM=FISIM*SPECT(I)                                              ISCH0104
      FISIO=FISIO+FISI                                                  ISCH0105
      FISIOM=FISIOM+FISIM                                               ISCH0106
      FSDI=FSDI*SPECT(I)                                                ISCH0107
      FSDIO=FSDIO+FSDI                                                  ISCH0108
```

```
      URN(L)=URN(L)+TEMP*SFU(L,I)
      PPU(L)=PPU(L)+TEMP*THSF(L,I)
      BR=BR+TEMP*SB(L,I)
      IF(L.GT.NCR) GO TO 250
      BRIN=BR
      GO TO 260
250   BREX=BREX+TEMP*SB(L,I)
260   PU(L)=PU(L)-TEMP*SUP(L,I)
      URC(L)=URC(L)-TEMP*SCU(L,I)
      BRU=BRU+TEMP*SCU(L,I)
      TOTP(L)=TOTP(L)+TEMP*SIGFM(L,I)
      POWED(L)=POWED(L)+PHL(N,I)*SIGFM(L,I)
      PRS(L)=PRS(L)+PHL(N,I)*SFU(L,I)*VNO
      PDU(L)=PDU(L)+PHL(N,I)*THSF(L,I)
      SYLM=SYL*DSAM(L,I)
      IF(I.NE.5) GO TO 45
      IF(L.GT.NCR) GO TO 45
      DOP=DOP+SYL*SD(L)
      DOPL(L)=SYL
45    SYL=SYL*THSA(L,I)
      SYLO=SYLO+SYL
      SYLOM=SYLOM+SYLM
4     CONTINUE
      DOPL(L)=DOPL(L)*STSI
      TOTP(L)=TOTP(L)*STSI
      SYLI(L)=SYLO*STSI
      SYLIM(L)=SYLOM*STSI
      POWERT=POWERT+TOTP(L)
      URC(L)=URC(L)*STSI
      PU(L)=PU(L)*STSI
      FISIT(L)=FISIO*STSI*EFFK
      FISITM(L)=FISIOM*STSI*EFFK
      FSDIT(L)=FSDIO*STSI
      FSDIN=FSDIN+FSDIT(L)
      URN(L)=URN(L)*STSI
      PPU(L)=PPU(L)*STSI
```

```
 1     CONTINUE                                                        ISCH0145
       BR=BR*STSI/POWERT                                               ISCH0146
       BRIN=BRIN*STSI/POWERT                                           ISCH0147
       BREX=BREX*STSI/POWERT                                           ISCH0148
       DOP=DOP*STSI/FSDIN                                              ISCH0149
       BRU=BRU*STSI                                                    ISCH0150
       PNORM=100.0/POWERT                                              ISCH0151
       BRU=BRU*PNORM                                                   ISCH0152
       IF(KNA.EQ.1) GO TO 210                                          ISCH0153
       WRITE(6,521) BR                                                 ISCH0154
       WRITE(6,522) BRIN                                               ISCH0155
       WRITE(6,523) BREX                                               ISCH0156
       IF(NPR.NE.1) GO TO 200                                          ISCH0157
       IF(NIT.EQ.0) GO TO 210                                          ISCH0158
       BRA(NIT)=BR                                                     ISCH0159
       IF(NIT.EQ.1) GO TO 210                                          ISCH0160
       IF(DABS(BRA(NIT)-BRA(NIT-1)).LT.0.00001)  CALL EXIT             ISCH0161
       GO TO 210                                                       ISCH0162
 200   IF(NPR.NE.3) GO TO 210                                          ISCH0163
       NI=NIT+1                                                        ISCH0164
       DOPC(NI)=DOP                                                    ISCH0165
 210   MA=0.0                                                          ISCH0166
       DO 14 L=1,NRG                                                   ISCH0167
       TMETO =0.0                                                      ISCH0168
       TMETOM=0.0                                                      ISCH0169
       IF(L.EQ.1) GO TO 15                                             ISCH0170
       MA=MA+NPT(L-1)                                                  ISCH0171
       K=NPT(L)-2+MA                                                   ISCH0172
       N=MA                                                            ISCH0173
       GO TO 16                                                        ISCH0174
 15    K=NPT(1)-2                                                      ISCH0175
       N=1                                                             ISCH0176
       IB=NGP-1                                                        ISCH0177
 16    DO 17 I=1,IB                                                    ISCH0178
       IA=I+1                                                          ISCH0179
       TMETA=0.0                                                       ISCH0180
```

```
      TMETAM=0.0
      DO 18 M=IA,NGP
      TMET=0.0
      DO 19 J=N,K,2
      TMET=TMET+PHL(J,I)*( PHI(J,I)- PHI(J,M))*AKTIS(J)+4.0*PHL(J+1,I)*
     1( PHI(J+1,I)- PHI(J+1,M))*AKTIS(J+1)+PHL(J+2,I)*
     1( PHI(J+2,I)- PHI(J+2,M))*AKTIS(J+2)
   19 CONTINUE
      TMETM=TMET*DSTM(L,I,M)
      TMET=TMET*THST(L,I,M)
      TMETA=TMETA+TMET
      TMETAM=TMETAM+TMETM
   18 CONTINUE
      TMETO =TMETO +TMETA
      TMETOM=TMETOM+TMETAM
   17 CONTINUE
      TMETOL(L)=TMETO *STSI
      TMETLM(L)=TMETOM*STSI
   14 CONTINUE
      STSIZ=(0.5/STS)
      MA=0.0
      DO 20 L=1,NRG
      ALKGE(L)=0.0
      ALKGEM(L)=0.0
      IF(L.EQ.1) GO TO 21
      MA=MA+NPT(L-1)
      K=NPT(L)-3+MA
      N=MA+1
      GO TO 22
   21 K=NPT(1)-3
      N=2
   22 KA=K+2
      KB=K+3
      DO 23 I=1,NGP
      ALKG=0.0
      DO 24 J=N,KA
```

```
          A(J,I)=(PHL(J+1,I)-PHL(J-1,I))*STSIZ
          B(J,I)=(PHI(J+1,I)- PHI(J-1,I))*STSIZ
   24     CONTINUE
          DO 32 J=N,K,2
          ALKG=ALKG+    A(J,I)*    B(J,I)*AKTIS(J)+4.0*    A(J+1,I)*
         1    B(J+1,I)*AKTIS(J+1)+    A(J+2,I)*    B(J+2,I)*AKTIS(J+2)
   32     CONTINUE
          IF(L.EQ.1) GO TO 25
          GRPH1(L,I)=DI(L-1,I)*GRPH2(L-1,I)/DI(L,I)
          GRPHA1(L,I)=DI(L-1,I)*GRPHA2(L-1,I)/DI(L,I)
          GO TO 40
   25     GRPH1(1,I)=0.0
          GRPHA1(1,I)=0.0
              A(1,I)=0.0
              B(1,I)=0.0
   40     AD1=0.0
          AD2=0.0
          ADA1=0.0
          ADA2=0.0
          DO 41 M=1,NGP
          CR=SGG(L,M,I)+SPECT(I)*SNUF(L,M)*EFFK
          CRA=SGG(L,I,M)+SPECT(M)*SNUF(L,I)*EFFK
          AD1=AD1+CR*PHL(KB,M)
          AD2=AD2+CR*PHL(KA,M)
          ADA1=ADA1+CRA*PHI(KB,M)
          ADA2=ADA2+CRA*PHI(KA,M)
   41     CONTINUE
          CR=SA(L,I)+SR(L,I)
          AF1=CR*PHL(KB,I)
          AF2=CR*PHL(KA,I)
          AFA1=CR*PHI(KB,I)
          AFA2=CR*PHI(KA,I)
          GRPH2(L,I)=AKTIS(KA)*A(KA,I)/AKTIS(KB)+0.5*STS*((AF1-AD1)+
         1 AKTIS(KA)*(AF2-AD2)/AKTIS(KB))/DI(L,I)
          GRPHA2(L,I)=AKTIS(KA)*B(KA,I)/AKTIS(KB)+0.5*STS*((AFA1-ADA1)+
         1 AKTIS(KA)*(AFA2-ADA2)/AKTIS(KB))/DI(L,I)
```

```
      ALKG=ALKG*STSI
      ALKG=ALKG+(GRPH1(L,I)*GRPHA1(L,I)*AKTIS(MA)+A(N,I)*B(N,I)*
     1 AKTIS(N))*0.5*STS
      ALKG=ALKG+(GRPH2(L,I)*GRPHA2(L,I)*AKTIS(KB)+A(KA,I)*B(KA,I)*
     1 AKTIS(KA))*0.5*STS
      ALKGM=ALKG*DDM(L,I)
      ALKG=ALKG*THD(L,I)
      ALKGE(L)=ALKGE(L)+ALKG
      ALKGEM(L)=ALKGEM(L)+ALKGM
   23 CONTINUE
      SYLI(L)=SYLI(L)/FSDIN
      SYLIM(L)=SYLIM(L)/FSDIN
      FISIT(L)=FISIT(L)/FSDIN
      FISITM(L)=FISITM(L)/FSDIN
      TMETOL(L)=TMETOL(L)/FSDIN
      TMETLM(L)=TMETLM(L)/FSDIN
      ALKGE(L)=ALKGE(L)/FSDIN
      ALKGEM(L)=ALKGEM(L)/FSDIN
      TOTP(L)=TOTP(L)*PNORM
      URN(L)=URN(L)*PNORM
      URC(L)=URC(L)*PNORM
      PPU(L)=PPU(L)*PNORM
      PU(L)=PU(L)*PNORM
      PRS(L)=PRS(L)*PNORM
      PDU(L)=PDU(L)*PNORM
      POWED(L)=POWED(L)*PNORM
      IF(KNA.EQ.1) GO TO 20
      WRITE(6,506) L,POWED(L)
   20 CONTINUE
      NPT(1)=NPT(1)-1
      DO 70 J=1,NP
      DO 70 I=1,NGP
      PHL(J,I)=PHL(J,I)*PNORM
   70 CONTINUE
      POWERT=100.0
      WRITE(6,504) POWERT
```

```
504   FORMAT(///' TOTAL POWER=',F15.7)
506   FORMAT('0REGION',I3,10X,'POWER DENSITY=',1PD15.7)
521   FORMAT(///' BREEDING GAIN =',F15.7)
522   FORMAT(///' INTERNAL BREEDING GAIN =', F15.7)
523   FORMAT(///' EXTERNAL BREEDING GAIN =', F15.7)
      RETURN
      END
```

```
      SUBROUTINE BASE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON/COWE/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
     1 VUR(13,5),NOP(10),NBD(10),NOPT,NRE
      COMMON/GREKO/ U2L(13),U2R(13),UVL(13,5),UVR(13,5),V2R(13,5,5),
     1 DU2L(13),DU2R(13),DUUL(13),DUUR(13),V2L(13,5,5),VVL(13,5,5),
     2 DVUSR(13,5),VVR(13,5,5),DV2R(13,5),DV2L(13,5),DVVL(13,5),
     3 DVVR(13,5),UVSR(13,5),UVSL(13,5),VUSR(13,5),VUSL(13,5),
     4 DVUSL(13,5),DUVL(13,5),DVUR(13,5),DVUL(13,5),DUVR(13,5),
     5 DUVSR(13,5),DUVSL(13,5),G(99,99)
C
C     NON-ZERO PRODUCTS
C
      DO 69 K=1,NRG
      NOP(K)=2
      HA(K)=0.5*TH(K)
   69 CONTINUE
 1000 STSI=STS*0.33333333333333
      NOPT=0
      DO 23 K=1,NRG
      NOPT=NOPT+NOP(K)
   23 CONTINUE
      NF=NOPT+1
      DO 5 K=1,NF
      U2L(K)=0.0
      U2R(K)=0.0
      DU2L(K)=0.0
```

```
      DU2R(K)=0.0                                                  BASE0037
      DUUL(K)=0.0                                                  BASE0038
      DUUR(K)=0.0                                                  BASE0039
      UUL(K)=0.0                                                   BASE0040
      UUR(K)=0.0                                                   BASE0041
      DO 5 I=1,NGP                                                 BASE0042
      VUL(K,I)=0.0                                                 BASE0043
      VUR(K,I)=0.0                                                 BASE0044
      UVL(K,I)=0.0                                                 BASE0045
      UVR(K,I)=0.0                                                 BASE0046
      DVUSR(K,I)=0.0                                               BASE0047
      DVUSL(K,I)=0.0                                               BASE0048
      DUVSR(K,I)=0.0                                               BASE0049
      DUVSL(K,I)=0.0                                               BASE0050
      DV2R(K,I)=0.0                                                BASE0051
      DV2L(K,I)=0.0                                                BASE0052
      DVVL(K,I)=0.0                                                BASE0053
      DVVR(K,I)=0.0                                                BASE0054
      UVSR(K,I)=0.0                                                BASE0055
      UVSL(K,I)=0.0                                                BASE0056
      VUSR(K,I)=0.0                                                BASE0057
      VUSL(K,I)=0.0                                                BASE0058
      DUVL(K,I)=0.0                                                BASE0059
      DUVR(K,I)=0.0                                                BASE0060
      DVUR(K,I)=0.0                                                BASE0061
      DVUL(K,I)=0.0                                                BASE0062
      DO 5 L=1,NGP                                                 BASE0063
      V2R(K,I,L)=0.0                                               BASE0064
      V2L(K,I,L)=0.0                                               BASE0065
      VVL(K,I,L)=0.0                                               BASE0066
      VVR(K,I,L)=0.0                                               BASE0067
    5 CONTINUE                                                     BASE0068
      NRE=NRG-1                                                    BASE0069
      NBD(1)=NOP(1)+1                                              BASE0070
      DO 1 K=2,NRG                                                 BASE0071
      NBD(K)=NBD(K-1)+NOP(K)                                       BASE0072
```

```
1     CONTINUE
      U2R(1)=HA(1)*HA(1)*(13.0/35.0-2.0/7.0)
      UUR(1)=9.0*HA(1)*HA(1)/140.0
      DU2R(1)=1.2-0.6
      DUUR(1)=-0.6
      N=1
      R=0.0
      DO 2 K=1,NRG
      M=NOP(K)-1
      DO 3 J=1,M
      R=R+HA(K)
      N=N+1
      U2L(N)=(2.0*HA(K)*HA(K)/7.0)+13.0*HA(K)*(R-HA(K))/35.0
      U2R(N)=-(2.0*HA(K)*HA(K)/7.0)+13.0*HA(K)*(R+HA(K))/35.0
      UUR(N)=(9.0/140.0)*HA(K)*HA(K)+(9.0/70.0)*HA(K)*R
      UUL(N)=(-9.0/140.0)*HA(K)*HA(K)+(9.0/70.0)*HA(K)*R
      DU2L(N)=(6.0/5.0)*(R-HA(K))/HA(K)+0.6
      DU2R(N)=(6.0/5.0)*(R+HA(K))/HA(K)-0.6
      DUUR(N)=-0.6-1.2*R/HA(K)
      DUUL(N)=0.6-1.2*R/HA(K)
3     CONTINUE
      N=N+1
      R=R+HA(K)
2     CONTINUE
      R=0.0
      DO 4 K=1,NRE
      R=R+NOP(K)*HA(K)
      N=NBD(K)
      U2L(N)=(2.0*HA(K)*HA(K)/7.0)+13.0*HA(K)*(R-HA(K))/35.0
      U2R(N)=-(2.0*HA(K+1)*HA(K+1)/7.0)+13.0*HA(K+1)*(R+HA(K+1))/35.0
      UUL(N)=UUR(N-1)
      UUR(N)=UUL(N+1)
      DU2R(N)=1.2*(R+HA(K+1))/HA(K+1)-0.6
      DU2L(N)=1.2*(R-HA(K))/HA(K)+0.6
      DUUL(N)=DUUR(N-1)
      DUUR(N)=DUUL(N+1)
```

```
4     CONTINUE                                                          BASE0109
      N=1                                                               BASE0110
      R=0.0                                                             BASE0111
      DO 10 K=1,NRG                                                     BASE0112
      M=NOP(K)-1                                                        BASE0113
      DO 11 J=1,M                                                       BASE0114
      R=R+HA(K)                                                         BASE0115
      N=N+1                                                             BASE0116
      DO 12 I=1,NGP                                                     BASE0117
      DO 13 L=I,NGP                                                     BASE0118
      V2R(N,I,L)=((R+HA(K))/105.0-HA(K)/168.0)*HA(K)*HA(K)*HA(K)/       BASE0119
     1 (DI(K,I)*DI(K,L))                                                BASE0120
      V2L(N,I,L)=((R-HA(K))/105.0+HA(K)/168.0)*HA(K)*HA(K)*HA(K)/       BASE0121
     1 (DI(K,I)*DI(K,L))                                                BASE0122
      VVR(N,I,L)=(-R/140.0-HA(K)/280.0)*HA(K)*HA(K)*HA(K)/(DI(K,I)*     BASE0123
     1 DI(K,L))                                                         BASE0124
      VVL(N,I,L)=(-R/140.0+HA(K)/280.0)*HA(K)*HA(K)*HA(K)/(DI(K,I)*     BASE0125
     1 DI(K,L))                                                         BASE0126
13    CONTINUE                                                          BASE0127
      UVSR(N,I)=((R+HA(K))*11.0/210.0-HA(K)/28.0)*HA(K)*HA(K)/DI(K,I)   BASE0128
      UVSL(N,I)=(-(R-HA(K))*11.0/210.0 -HA(K)/28.0)*HA(K)*HA(K)/DI(K,I) BASE0129
      UVR(N,I)=(-13.0*R/420.0-HA(K)/70.0)*HA(K)*HA(K)/DI(K,I)           BASE0130
      UVL(N,I)=(13.0*R/420.0-HA(K)/70.0)*HA(K)*HA(K)/DI(K,I)            BASE0131
      VUSL(N,I)=UVSL(N,I)                                               BASE0132
      VUSR(N,I)=UVSR(N,I)                                               BASE0133
      VUR(N,I)=(13.0*R/420.0+HA(K)/60.0)*HA(K)*HA(K)/DI(K,I)            BASE0134
      VUL(N,I)=(-13.0*R/420.0+HA(K)/60.0)*HA(K)*HA(K)/DI(K,I)           BASE0135
      DUVSR(N,I)=(R+HA(K))*0.1/DI(K,I)                                  BASE0136
      DUVSL(N,I)=-(R-HA(K))*0.1/DI(K,I)                                 BASE0137
      DVUSL(N,I)=DUVSL(N,I)                                             BASE0138
      DVUSR(N,I)=DUVSR(N,I)                                             BASE0139
      DUVR(N,I)=0.1*R/DI(K,I)                                           BASE0140
      DUVL(N,I)= -DUVR(N,I)                                             BASE0141
      DVUR(N,I)=-0.1*(R+HA(K))/DI(K,I)                                  BASE0142
      DVUL(N,I)=0.1*(R-HA(K))/DI(K,I)                                   BASE0143
      DV2R(N,I)=((R+HA(K))*2.0/15.0-0.1*HA(K))*HA(K)/(DI(K,I)*DI(K,I))  BASE0144
```

```
      DV2L(N,I)=((R-HA(K))*2.0/15.0+0.1*HA(K))*HA(K)/(DI(K,I)*DI(K,I))     BASE0145
      DVVL(N,I)=(HA(K)/(DI(K,I)*DI(K,I)))*(-1.0*R/30.0+HA(K)/60.0)         BASE0146
      DVVR(N,I)=(HA(K)/(DI(K,I)*DI(K,I)))*(-1.0*R/30.0-HA(K)/60.0)         BASE0147
   12 CONTINUE                                                             BASE0148
   11 CONTINUE                                                             BASE0149
      N=N+1                                                                BASE0150
      R=R+HA(K)                                                            BASE0151
   10 CONTINUE                                                             BASE0152
      DO 14 I=1,NGP                                                        BASE0153
      DO 15 L=I,NGP                                                        BASE0154
      V2L(N,I,L)=((R-HA(K))/105.0+HA(K)/168.0)*HA(K)*HA(K)*HA(K)           BASE0155
     1 /(DI(K,I)*DI(K,L))                                                  BASE0156
      VVL(N,I,L)=VVR(N-1,I,L)                                              BASE0157
   15 CONTINUE                                                             BASE0158
      DV2L(N,I)=((R-HA(K))*2.0/15.0+0.1*HA(K))*HA(K)/(DI(K,I)*DI(K,I))     BASE0159
      DVVL(N,I)=DVVR(N-1,I)                                                BASE0160
      VUL(N,I)=UVR(N-1,I)                                                  BASE0161
      DVUL(N,I)=DUVR(N-1,I)                                                BASE0162
   14 CONTINUE                                                             BASE0163
      R=0.0                                                                BASE0164
      DO 16 K=1,NRE                                                        BASE0165
      R=R+NOP(K)*HA(K)                                                     BASE0166
      N=NBD(K)                                                             BASE0167
      DO 17 I=1,NGP                                                        BASE0168
      DO 18 L=I,NGP                                                        BASE0169
      V2R(N,I,L)=((R+HA(K+1))/105.0-HA(K+1)/168.0)*HA(K+1)*HA(K+1)*HA(     BASE0170
     1 K+1)/(DI(K+1,I)*DI(K+1,L))                                          BASE0171
      V2L(N,I,L)=((R-HA(K))/105.0+HA(K)/168.0)*HA(K)*HA(K)*HA(K)/          BASE0172
     1 (DI(K,I)*DI(K,L))                                                   BASE0173
      VVL(N,I,L)=VVR(N-1,I,L)                                              BASE0174
      VVR(N,I,L)=VVL(N+1,I,L)                                              BASE0175
   18 CONTINUE                                                             BASE0176
      UVSR(N,I)=((R+HA(K+1))*11.0/210.0-HA(K+1)/28.0)*HA(K+1)*HA(K+1)/     BASE0177
     1 DI(K+1,I)                                                           BASE0178
      UVSL(N,I)=(-(R-HA(K))*11.0/210.0 -HA(K)/28.0)*HA(K)*HA(K)/DI(K,I)    BASE0179
      UVR(N,I)=VUL(N+1,I)                                                  BASE0180
```

```
      UVL(N,I)=VUR(N-1,I)
      VUSL(N,I)=UVSL(N,I)
      VUSR(N,I)=UVSR(N,I)
      VUR(N,I)=UVL(N+1,I)
      VUL(N,I)=UVR(N-1,I)
      DUVSR(N,I)=(R+HA(K+1))*0.1/DI(K+1,I)
      DUVSL(N,I)=-(R-HA(K))*0.1/DI(K,I)
      DUVR(N,I)=DVUL(N+1,I)
      DVUSL(N,I)=DUVSL(N,I)
      DVUSR(N,I)=DUVSR(N,I)
      DVVL(N,I)=(HA(K)/(DI(K,I)*DI(K,I)))*(-1.0*R/30.0+HA(K)/60.0)
      DVVR(N,I)=(HA(K+1)/(DI(K+1,I)*DI(K+1,I)))*(-1.0*R/30.0-HA(K+1)
     1/60.0)
      DUVL(N,I)=DVUR(N-1,I)
      DVUR(N,I)=DUVL(N+1,I)
      DVUL(N,I)=DUVR(N-1,I)
      DV2R(N,I)=((R+HA(K+1))*2.0/15.0-0.1*HA(K+1))*HA(K+1)/(DI(K+1,I)
     1 *DI(K+1,I))
      DV2L(N,I)=((R-HA(K))*2.0/15.0+0.1*HA(K))*HA(K)/(DI(K,I)*DI(K,I))
   17 CONTINUE
   16 CONTINUE
      DO 19 I=1,NGP
      UVR(1,I)=VUL(2,I)
      DUVR(1,I)=DVUL(2,I)
   19 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  BIGMAT                                                BIGM0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         BIGM0002
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),BIGM0003
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),  BIGM0004
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),   BIGM0005
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),           BIGM0006
     4 SGG(10,5,5),DI(10,5)                                             BIGM0007
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),  BIGM0008
     1NGP,NRG,NMAT,NGFOM,JBCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,       BIGM0009
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,  BIGM0010
     3JDUM,IHOLD(90)                                                    BIGM0011
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5), BIGM0012
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)                      BIGM0013
      COMMON/COWE/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),   BIGM0014
     1 VUR(13,5),NOP(10),NBD(10),NOPT,NRE                               BIGM0015
      COMMON/GREKO/ U2L(13),U2R(13),UVL(13,5),UVR(13,5),V2R(13,5,5),    BIGM0016
     1 DU2L(13),DU2R(13),DUUL(13),DUUR(13),V2L(13,5,5),VVL(13,5,5),     BIGM0017
     2 DVUSR(13,5),VVR(13,5,5),DV2R(13,5),DV2L(13,5),DVVL(13,5),        BIGM0018
     3 DVVR(13,5),UVSR(13,5),UVSL(13,5),VUSR(13,5),VUSL(13,5),          BIGM0019
     4 DVUSL(13,5),DUVL(13,5),DVUR(13,5),DVUL(13,5),DUVR(13,5),         BIGM0020
     5 DUVSR(13,5),DUVSL(13,5),G(99,99)                                 BIGM0021
      COMMON/ATHENS/ BU(3),BV(3),OLU(5),OLV(5,5),ADU(5),ADV(5),DDU(5),  BIGM0022
     1 DDV(5),TU(13,5),TV(13,5),TUL(13,5),TUR(13,5),TVL(13,5),TVR(13,5),BIGM0023
     2       DU(13,5),DV(13,5),DUL(13,5),DUR(13,5),DVL(13,5),DVR(13,5)  BIGM0024
      DIMENSION F(99,11),LW(99),MW(99)                                  BIGM0025
C                                                                       BIGM0026
C     MATRICES CONSISTING THE DIAGONAL ELEMENTS OF THE BIG MATRIX       BIGM0027
C                                                                       BIGM0028
      DO 1 K=1,NRG                                                      BIGM0029
      DO 1 I=1,NGP                                                      BIGM0030
      SNUF(K,I)=SNUF(K,I)*EFFK                                          BIGM0031
 1    CONTINUE                                                          BIGM0032
      NAGN=2*NRG+1                                                      BIGM0033
      NOM=2*NOPT*NGP-1                                                  BIGM0034
      DO 24 M=1,NOM                                                     BIGM0035
      DO 29 K=1,NAGN                                                    BIGM0036
```

```
      F(M,K)=0.0                                                        BIGM0037
29    CONTINUE                                                          BIGM0038
      DO 24 N=1,NOM                                                     BIGM0039
      G(M,N)=0.0                                                        BIGM0040
24    CONTINUE                                                          BIGM0041
      K=0                                                               BIGM0042
      DO 25 I=1,NGP                                                     BIGM0043
      NA=1                                                              BIGM0044
      N=NA+K                                                            BIGM0045
      M=1+K                                                             BIGM0046
      CR=SPECT(I)*SNUF(1,I)-SR(1,I)-SA(1,I)                             BIGM0047
      IF(I.EQ.1) GO TO 100                                              BIGM0048
      G(M,N)=-DI(1,I)*0.6+CR*U2R(1)                                     BIGM0049
      G(M,N+1)=DI(1,I)*0.6+CR*UUR(1)                                    BIGM0050
      G(M,N+2)=-DI(1,I)*DUVR(1,I)+CR*UVR(1,I)                           BIGM0051
      M=M+1                                                             BIGM0052
      G(M,N)=-DI(1,I)*DUUL(2)+CR*UUL(2)                                 BIGM0053
      G(M,N+1)=-DI(1,I)*(DU2R(2)+DU2L(2))+CR*(U2R(2)+U2L(2))            BIGM0054
      G(M,N+2)=-DI(1,I)*(DUVSR(2,I)+DUVSL(2,I))+CR*(UVSR(2,I)+UVSL(2,I))BIGM0055
      G(M,N+3)=-DI(1,I)*DUUR(2)+CR*UUR(2)                               BIGM0056
      G(M,N+4)=-DI(1,I)* DUVR(2,I)+CR*UVR(2,I)                          BIGM0057
      M=M+1                                                             BIGM0058
      G(M,N)=-DI(1,I)*DVUL(2,I)+CR*VUL(2,I)                             BIGM0059
      G(M,N+1)=-DI(1,I)*(DVUSR(2,I)+DVUSL(2,I))+CR*(VUSL(2,I)+VUSR(2,I))BIGM0060
      G(M,N+2)=-DI(1,I)*(DV2R(2,I)+DV2L(2,I))+CR*(V2R(2,I,I)+V2L(2,I,I))BIGM0061
      G(M,N+3)=-DI(1,I)*DVUR(2,I)+CR*VUR(2,I)                           BIGM0062
      G(M,N+4)=-DI(1,I)*DVVR(2,I)+CR*VVR(2,I,I)                         BIGM0063
      NA=NA+1                                                           BIGM0064
      GO TO 101                                                         BIGM0065
100   G(M,N)  =-DI(1,I)*(DU2R(2)+DU2L(2))+CR*(U2R(2)+U2L(2))            BIGM0066
      F(M,N)= DI(1,I)*DUUL(2)-CR*UUL(2)                                 BIGM0067
      G(M,N+1)=-DI(1,I)*(DUVSR(2,I)+DUVSL(2,I))+CR*(UVSR(2,I)+UVSL(2,I))BIGM0068
      G(M,N+2)=-DI(1,I)*DUUR(2)+CR*UUR(2)                               BIGM0069
      G(M,N+3)=-DI(1,I)* DUVR(2,I)+CR*UVR(2,I)                          BIGM0070
      M=M+1                                                             BIGM0071
      F(M,N)= DI(1,I)*DVUL(2,I)-CR*VUL(2,I)                             BIGM0072
```

```
      G(M,N)  =-DI(1,I)*(DVUSR(2,I)+DVUSL(2,I))+CR*(VUSL(2,I)+VUSR(2,I))        BIGM0073
      G(M,N+1)=-DI(1,I)*(DV2R(2,I)+DV2L(2,I))+CR*(V2R(2,I,I)+V2L(2,I,I))        BIGM0074
      G(M,N+2)=-DI(1,I)*DVUR(2,I)+CR*VUR(2,I)                                   BIGM0075
      G(M,N+3)=-DI(1,I)*DVVR(2,I)+CR*VVR(2,I,I)                                 BIGM0076
 101  L=2                                                                       BIGM0077
      DO 26 NR=2,NRG                                                            BIGM0078
      DO 27 JJ=1,2                                                              BIGM0079
      JI=2-JJ                                                                   BIGM0080
      M=M+1                                                                     BIGM0081
      L=L+1                                                                     BIGM0082
      N=NA+K                                                                    BIGM0083
      J=NR-JI                                                                   BIGM0084
      CRL=SPECT(I)*SNUF(J,I)-SR(J,I)-SA(J,I)                                    BIGM0085
      CRR=SPECT(I)*SNUF(NR,I)-SR(NR,I)-SA(NR,I)                                 BIGM0086
      G(M,N)=-DI(J,I)*DUUL(L)+CRL*UUL(L)                                        BIGM0087
      G(M,N+1)=-DI(J,I)*DUVL(L,I)+CRL*UVL(L,I)                                  BIGM0088
      G(M,N+2)=-DI(J,I)*DU2L(L)-DI(NR,I)*DU2R(L)+CRL*U2L(L)+CRR*U2R(L)          BIGM0089
      G(M,N+3)=-DI(J,I)*DUVSL(L,I)-DI(NR,I)*DUVSR(L,I)+CRL*UVSL(L,I)+           BIGM0090
     1 CRR*UVSR(L,I)                                                            BIGM0091
      G(M,N+4)=-DI(NR,I)*DUUR(L)+CRR*UUR(L)                                     BIGM0092
      G(M,N+5)=-DI(NR,I)*DUVR(L,I)+CRR*UVR(L,I)                                 BIGM0093
      M=M+1                                                                     BIGM0094
      G(M,N)=-DI(J,I)*DVUL(L,I)+CRL*VUL(L,I)                                    BIGM0095
      G(M,N+1)=-DI(J,I)*DVVL(L,I)+CRL*VVL(L,I,I)                                BIGM0096
      G(M,N+2)=-DI(J,I)*DVUSL(L,I)-DI(NR,I)*DVUSR(L,I)+CRL*VUSL(L,I)+           BIGM0097
     1 CRR*VUSR(L,I)                                                            BIGM0098
      G(M,N+3)=-DI(J,I)*DV2L(L,I)-DI(NR,I)*DV2R(L,I)+CRL*V2L(L,I,I)+            BIGM0099
     1 CRR*V2R(L,I,I)                                                           BIGM0100
      G(M,N+4)=-DI(NR,I)*DVUR(L,I)+CRR*VUR(L,I)                                 BIGM0101
      G(M,N+5)=-DI(NR,I)*DVVR(L,I)+CRR*VVR(L,I,I)                               BIGM0102
      NA=NA+2                                                                   BIGM0103
      IF(NR.EQ.NRG) GO TO 28                                                    BIGM0104
 27   CONTINUE                                                                  BIGM0105
 26   CONTINUE                                                                  BIGM0106
 28   M=M+1                                                                     BIGM0107
      J=NRG                                                                     BIGM0108
```

```
      N=NA+K
      L=L+1
      CR=SPECT(I)*SNUF(J,I)-SR(J,I)-SA(J,I)
      G(M,N)=-DI(J,I)*DUUL(L)+CR*UUL(L)
      G(M,N+1)=-DI(J,I)*DUVL(L,I)+CR *UVL(L,I)
      G(M,N+2)=-DI(J,I)*(DU2L(L)+DU2R(L))+CR*(U2L(L)+U2R(L))
      G(M,N+3)=-DI(J,I)*(DUVSL(L,I)+DUVSR(L,I))+CR*(UVSL(L,I)+UVSR(L,I))
      G(M,N+4)=-DI(J,I)*DUVR(L,I)+CR*UVR(L,I)
      M=M+1
      G(M,N)=-DI(J,I)*DVUL(L,I)+CR *VUL(L,I)
      G(M,N+1)=-DI(J,I)*DVVL(L,I)+CR *VVL(L,I,I)
      G(M,N+2)=-DI(J,I)*(DVUSL(L,I)+DVUSR(L,I))+CR*(VUSL(L,I)+VUSR(L,I))
      G(M,N+3)=-DI(J,I)*(DV2L(L,I)+DV2R(L,I))+CR*(V2L(L,I,I)+V2R(L,I,I))
      G(M,N+4)=-DI(J,I)*DVVR(L,I)+CR*VVR(L,I,I)
      M=M+1
      N=N+2
      L=L+1
      G(M,N)=-DI(J,I)*DVUL(L,I)+CR *VUL(L,I)
      G(M,N+1)=-DI(J,I)*DVVL(L,I)+CR *VVL(L,I,I)
      G(M,N+2)=-DI(J,I)*DV2L(L,I)+CR*V2L(L,I,I)
      IF(I.EQ.1) GO TO 102
      K=K+2*NOPT
      GO TO 25
  102 K=K+2*NOPT-1
   25 CONTINUE
C
C     MATRICES CONSISTING THE ABOVE THE DIAGONAL ELEMENTS OF THE BIG
C     MATRIX
C
      NGPE=NGP-1
      IA=0
      DO 55 L=1,NGPE
      KW=2*NOPT*L-1
      IB=L+1
      DO 56 I=IB,NGP
      M=IA+1
```

```
      NA=1                                                              BIGM0145
      N=NA+KW                                                           BIGM0146
      CR=SPECT(L)*SNUF(1,I)                                             BIGM0147
      IF(L.EQ.1) GO TO 103                                              BIGM0148
      G(M,N)=CR*U2R(1)                                                  BIGM0149
      G(M,N+1)=CR*UUR(1)                                                BIGM0150
      G(M,N+2)=CR*UVR(1,I)                                              BIGM0151
      M=M+1                                                             BIGM0152
  103 G(M,N)=CR*UUL(2)                                                  BIGM0153
      G(M,N+1)=CR*(U2R(2)+U2L(2))                                       BIGM0154
      G(M,N+2)=CR*(UVSR(2,I)+UVSL(2,I))                                 BIGM0155
      G(M,N+3)=CR*UUR(2)                                                BIGM0156
      G(M,N+4)=CR*UVR(2,I)                                              BIGM0157
      M=M+1                                                             BIGM0158
      G(M,N)=CR*VUL(2,L)                                                BIGM0159
      G(M,N+1)=CR*(VUSR(2,L)+VUSL(2,L))                                 BIGM0160
      G(M,N+2)=CR*(V2R(2,L,I)+V2L(2,L,I))                               BIGM0161
      G(M,N+3)=CR*VUR(2,L)                                              BIGM0162
      G(M,N+4)=CR*VVR(2,L,I)                                            BIGM0163
      NA=NA+1                                                           BIGM0164
      K=2                                                               BIGM0165
      DO 57 NR=2,NRG                                                    BIGM0166
      DO 58 JJ=1,2                                                      BIGM0167
      JI=2-JJ                                                           BIGM0168
      J=NR-JI                                                           BIGM0169
      K=K+1                                                             BIGM0170
      N=NA+KW                                                           BIGM0171
      M=M+1                                                             BIGM0172
      CRL=SPECT(L)*SNUF(J,I)                                            BIGM0173
      CRR=SPECT(L)*SNUF(NR,I)                                           BIGM0174
      G(M,N)=CRL*UUL(K)                                                 BIGM0175
      G(M,N+1)=CRL*UVL(K,I)                                             BIGM0176
      G(M,N+2)=CRL*U2L(K)+CRR*U2R(K)                                    BIGM0177
      G(M,N+3)=CRL*UVSL(K,I)+CRR*UVSR(K,I)                              BIGM0178
      G(M,N+4)=CRR*UUR(K)                                               BIGM0179
      G(M,N+5)=CRR*UVR(K,I)                                             BIGM0180
```

```
      M=M+1                                                    BIGM0181
      G(M,N)=CRL*VUL(K,L)                                      BIGM0182
      G(M,N+1)=CRL*VVL(K,L,I)                                  BIGM0183
      G(M,N+2)=CRL*VUSL(K,L)+CRR*VUSR(K,L)                     BIGM0184
      G(M,N+3)=CRL*V2L(K,L,I)+CRR*V2R(K,L,I)                   BIGM0185
      G(M,N+4)=CRR*VUR(K,L)                                    BIGM0186
      G(M,N+5)=CRR*VVR(K,L,I)                                  BIGM0187
      NA=NA+2                                                  BIGM0188
      IF(NR.EQ.NRG) GO TO 59                                   BIGM0189
   58 CONTINUE                                                 BIGM0190
   57 CONTINUE                                                 BIGM0191
   59 M=M+1                                                    BIGM0192
      J=NRG                                                    BIGM0193
      K=K+1                                                    BIGM0194
      N=NA+KW                                                  BIGM0195
      CR=SPECT(L)*SNUF(J,I)                                    BIGM0196
      G(M,N)=CR*UUL(K)                                         BIGM0197
      G(M,N+1)=CR*UVL(K,I)                                     BIGM0198
      G(M,N+2)=CR*(U2L(K)+U2R(K))                              BIGM0199
      G(M,N+3)=CR*(UVSL(K,I)+UVSR(K,I))                        BIGM0200
      G(M,N+4)=CR*UVR(K,I)                                     BIGM0201
      M=M+1                                                    BIGM0202
      G(M,N)=CR*VUL(K,L)                                       BIGM0203
      G(M,N+1)=CR*VVL(K,L,I)                                   BIGM0204
      G(M,N+2)=CR*(VUSL(K,L)+VUSR(K,L))                        BIGM0205
      G(M,N+3)=CR*(V2L(K,L,I)+V2R(K,L,I))                      BIGM0206
      G(M,N+4)=CR*VVR(K,L,I)                                   BIGM0207
      M=M+1                                                    BIGM0208
      N=N+2                                                    BIGM0209
      K=K+1                                                    BIGM0210
      G(M,N)=CR*VUL(K,L)                                       BIGM0211
      G(M,N+1)=CR*VVL(K,L,I)                                   BIGM0212
      G(M,N+2)=CR*V2L(K,L,I)                                   BIGM0213
      KW=KW+2*NOPT                                             BIGM0214
   56 CONTINUE                                                 BIGM0215
      IF(L.EQ.1) GO TO 104                                     BIGM0216
```

```
      IA=IA+2*NOPT                                                      BIGM0217
      GO TO 55                                                          BIGM0218
  104 IA=IA+2*NOPT-1                                                    BIGM0219
   55 CONTINUE                                                          BIGM0220
C                                                                       BIGM0221
C     MATRICES CONSISTING THE BELOW THE DIAGONAL ELEMENTS OF THE BIG    BIGM0222
C     MATRIX                                                            BIGM0223
C                                                                       BIGM0224
      KW=0                                                              BIGM0225
      DO 65 L=1,NGPE                                                    BIGM0226
      IA=2*NOPT*L-1                                                     BIGM0227
      IB=L+1                                                            BIGM0228
      DO 66 I=IB,NGP                                                    BIGM0229
      M=IA+1                                                            BIGM0230
      NA=1                                                              BIGM0231
      N=NA+KW                                                           BIGM0232
      CR=SGG(1,L,I)+SPECT(I)*SNUF(1,L)                                  BIGM0233
      IF(L.EQ.1) GO TO 105                                              BIGM0234
      G(M,N)=CR*U2R(1)                                                  BIGM0235
      G(M,N+1)=CR*UUR(1)                                                BIGM0236
      G(M,N+2)=CR*UVR(1,L)                                              BIGM0237
      M=M+1                                                             BIGM0238
      G(M,N)=CR*UUL(2)                                                  BIGM0239
      G(M,N+1)=CR*(U2R(2)+U2L(2))                                       BIGM0240
      G(M,N+2)=CR*(UVSR(2,L)+UVSL(2,L))                                 BIGM0241
      G(M,N+3)=CR*UUR(2)                                                BIGM0242
      G(M,N+4)=CR*UVR(2,L)                                              BIGM0243
      M=M+1                                                             BIGM0244
      G(M,N)=CR*VUL(2,I)                                                BIGM0245
      G(M,N+1)=CR*(VUSR(2,I)+VUSL(2,I))                                 BIGM0246
      G(M,N+2)=CR*(V2R(2,L,I)+V2L(2,L,I))                               BIGM0247
      G(M,N+3)=CR*VUR(2,I)                                              BIGM0248
      G(M,N+4)=CR*VVR(2,L,I)                                            BIGM0249
      NA=NA+1                                                           BIGM0250
      GO TO 106                                                         BIGM0251
  105 G(M,N)  =CR*UUR(1)                                                BIGM0252
```

```
      F(M,N)=-CR*U2R(1)                                                 BIGM0253
      G(M,N+1)=CR*UVR(1,L)                                              BIGM0254
      M=M+1                                                             BIGM0255
      F(M,N)=-CR*UUL(2)                                                 BIGM0256
      G(M,N)   =CR*(U2R(2)+U2L(2))                                      BIGM0257
      G(M,N+1)=CR*(UVSR(2,L)+UVSL(2,L))                                 BIGM0258
      G(M,N+2)=CR*UUR(2)                                                BIGM0259
      G(M,N+3)=CR*UVR(2,L)                                              BIGM0260
      M=M+1                                                             BIGM0261
      F(M,N)=-CR*VUL(2,I)                                               BIGM0262
      G(M,N)   =CR*(VUSR(2,I)+VUSL(2,I))                                BIGM0263
      G(M,N+1)=CR*(V2R(2,L,I)+V2L(2,L,I))                               BIGM0264
      G(M,N+2)=CR*VUR(2,I)                                              BIGM0265
      G(M,N+3)=CR*VVR(2,L,I)                                            BIGM0266
 106  K=2                                                               BIGM0267
      DO 67 NR=2,NRG                                                    BIGM0268
      DO 68 JJ=1,2                                                      BIGM0269
      JI=2-JJ                                                           BIGM0270
      J=NR-JI                                                           BIGM0271
      K=K+1                                                             BIGM0272
      N=NA+KW                                                           BIGM0273
      M=M+1                                                             BIGM0274
      CRL=SGG(J,L,I)+SPECT(I)*SNUF(J,L)                                 BIGM0275
      CRR=SGG(NR,L,I)+SPECT(I)*SNUF(NR,L)                               BIGM0276
      G(M,N)=CRL*UUL(K)                                                 BIGM0277
      G(M,N+1)=CRL*UVL(K,L)                                             BIGM0278
      G(M,N+2)=CRL*U2L(K)+CRR*U2R(K)                                    BIGM0279
      G(M,N+3)=CRL*UVSL(K,L)+CRR*UVSR(K,L)                              BIGM0280
      G(M,N+4)=CRR*UUR(K)                                               BIGM0281
      G(M,N+5)=CRR*UVR(K,L)                                             BIGM0282
      M=M+1                                                             BIGM0283
      G(M,N)=CRL*VUL(K,I)                                               BIGM0284
      G(M,N+1)=CRL*VVL(K,L,I)                                           BIGM0285
      G(M,N+2)=CRL*VUSL(K,I)+CRR*VUSR(K,I)                              BIGM0286
      G(M,N+3)=CRL*V2L(K,L,I)+CRR*V2R(K,L,I)                            BIGM0287
      G(M,N+4)=CRR*VUR(K,I)                                             BIGM0288
```

```
      G(M,N+5)=CRR*VVR(K,L,I)                                           BIGM0289
      NA=NA+2                                                           BIGM0290
      IF(NR.EQ.NRG) GO TO 69                                            BIGM0291
   68 CONTINUE                                                          BIGM0292
   67 CONTINUE                                                          BIGM0293
   69 M=M+1                                                             BIGM0294
      J=NRG                                                             BIGM0295
      CR=SGG(J,L,I)+SPECT(I)*SNUF(J,L)                                  BIGM0296
      K=K+1                                                             BIGM0297
      N=NA+KW                                                           BIGM0298
      G(M,N)=CR*UUL(K)                                                  BIGM0299
      G(M,N+1)=CR*UVL(K,L)                                              BIGM0300
      G(M,N+2)=CR*(U2L(K)+U2R(K))                                       BIGM0301
      G(M,N+3)=CR*(UVSL(K,L)+UVSR(K,L))                                 BIGM0302
      G(M,N+4)=CR*UVR(K,L)                                              BIGM0303
      M=M+1                                                             BIGM0304
      G(M,N)=CR*VUL(K,I)                                                BIGM0305
      G(M,N+1)=CR*VVL(K,L,I)                                            BIGM0306
      G(M,N+2)=CR*(VUSL(K,I)+VUSR(K,I))                                 BIGM0307
      G(M,N+3)=CR*(V2L(K,L,I)+V2R(K,L,I))                               BIGM0308
      G(M,N+4)=CR*VVR(K,L,I)                                            BIGM0309
      M=M+1                                                             BIGM0310
      N=N+2                                                             BIGM0311
      K=K+1                                                             BIGM0312
      G(M,N)=CR*VUL(K,I)                                                BIGM0313
      G(M,N+1)=CR*VVL(K,L,I)                                            BIGM0314
      G(M,N+2)=CR*V2L(K,L,I)                                            BIGM0315
      IA=IA+2*NOPT                                                      BIGM0316
   66 CONTINUE                                                          BIGM0317
      IF(L.EQ.1) GO TO 107                                              BIGM0318
      KW=KW+2*NOPT                                                      BIGM0319
      GO TO 65                                                          BIGM0320
  107 KW=KW+2*NOPT-1                                                    BIGM0321
   65 CONTINUE                                                          BIGM0322
      DO 95 M=1,NOM                                                     BIGM0323
      DO 95 N=1,NOM                                                     BIGM0324
```

```
      G(M,N)=0.01*G(M,N)
   95 CONTINUE
      CALL WENDO
C
C     RIGHT HAND SIDE MATRIX
C
      LB=NRG-2
      K=-1
      DO 81 I=1,NGP
      IF(I.EQ.1) GO TO 116
      F(1+K,2)=TU(1,I)
      F(1+K,3)=DU(1,I)
  116 F(2+K,2)=TU(2,I)
      F(3+K,2)=TV(2,I)
      F(4+K,2)=TUL(3,I)
      F(5+K,2)=TVL(3,I)
      F(2+K,3)=DU(2,I)
      F(3+K,3)=DV(2,I)
      F(4+K,3)=DUL(3,I)
      F(5+K,3)=DVL(3,I)
      N=3
      J=2
      L=4+K
      DO 150 LA=1,LB
      J=J+2
      M=J+1
      F(L,J)=TUR(N,I)
      F(L+1,J)=TVR(N,I)
      F(L+2,J)=TU(N+1,I)
      F(L+3,J)=TV(N+1,I)
      F(L+4,J)=TUL(N+2,I)
      F(L+5,J)=TVL(N+2,I)
C
      F(L,M)=DUR(N,I)
      F(L+1,M)=DVR(N,I)
      F(L+2,M)=DU(N+1,I)
```

```
      F(L+3,M)=DV(N+1,I)
      F(L+4,M)=DUL(N+2,I)
      F(L+5,M)=DVL(N+2,I)
      L=L+4
      N=N+2
 150  CONTINUE
      J=J+2
      M=J+1
      F(L,J)=TUR(N,I)
      F(L+1,J)=TVR(N,I)
      F(L+2,J)=TU(N+1,I)
      F(L+3,J)=TV(N+1,I)
      F(L+4,J)=TVL(N+2,I)
C
      F(L,M)=DUR(N,I)
      F(L+1,M)=DVR(N,I)
      F(L+2,M)=DU(N+1,I)
      F(L+3,M)=DV(N+1,I)
      F(L+4,M)=DVL(N+2,I)
      K=2*NOPT+K
 81   CONTINUE
      DO 94 M=1,NOM
      DO 94 N=1,NAGN
      F(M,N)=0.01*F(M,N)
 94   CONTINUE
      CALL DMINV(G,NOM,DT,LW,MW)
C
      N=NOM
      M=NOM
      L= NAGN
      CALL ELIZA(G,F,WK,N,M,L)
      RETURN
      END
```

```
      SUBROUTINE  WENDO
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM, IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON/COWE/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
     1 VUR(13,5),NOP(10),NBD(10),NOPT,NRE
      COMMON/ATHENS/ BU(3),BV(3),OLU(5),OLV(5,5),ADU(5),ADV(5),DDU(5),
     1 DDV(5),TU(13,5),TV(13,5),TUL(13,5),TUR(13,5),TVL(13,5),TVR(13,5),
     2        DU(13,5),DV(13,5),DUL(13,5),DUR(13,5),DVL(13,5),DVR(13,5)
      COMMON/DELTA/ THSA(10,5),THNSF(10,5),THD(10,5),THST(10,5,5),
     1 THTRP(10,5),THSTT(10,5),DSAM(10,5),DNSFM(10,5),DDM(10,5),
     2 DSTM(10,5,5),DTRPM(10,5),DSTTM(10,5),THSF(10,5),DSFM(10,5),
     3 SFU(10,5),SCU(10,5),SUP(10,5),POWED(10),CONCP(10),VNO
C
C
C     ***   ***
      DO 1 L=1,NRG
      DO 1 I=1,NGP
      SR(L,I)=SR(L,I)+SA(L,I)
    1 CONTINUE
C     ***   ***
      STSI=STS*0.33333333333333
      NPT(1)=NPT(1)+1
      L=1
      K=0.5*(NPT(1)-1)-1+0.5
      HSQ=HA(1)*HA(1)
      N1=K+2
```

```
      DO 40 I=1,NGP                                                     WEND0037
      OLU(I)=0.0                                                        WEND0038
      DO 41 J=1,K,2                                                     WEND0039
      DO 42 JA=1,3                                                      WEND0040
      JI=J+JA-1                                                         WEND0041
      ARG(JA)=AKTIS(N1)-AKTIS(JI)                                       WEND0042
      BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(1))*AKTIS(JI)    WEND0043
   42 CONTINUE                                                          WEND0044
      J1=J+1                                                            WEND0045
      J2=J+2                                                            WEND0046
      OLU(I)=OLU(I)+PHL(J,I)*BU(1)+4.0*PHL(J1,I)*BU(2)+PHL(J2,I)*BU(3)  WEND0047
   41 CONTINUE                                                          WEND0048
      OLU(I)=OLU(I)*STSI                                                WEND0049
   40 CONTINUE                                                          WEND0050
      DO 43 I=1,NGP                                                     WEND0051
      ADU(I)=0.0                                                        WEND0052
      DDU(I)=0.0                                                        WEND0053
      DO 44 M=1,NGP                                                     WEND0054
      PROS=(THD(1,I)/DI(1,I))*(SGG(1,M,I)+SPECT(I)*SNUF(1,M))-THST(1,M  WEND0055
     1 ,I)-SPECT(I)*THNSF(1,M)*EFFK                                     WEND0056
      PROD=(DDM(1,I)/DI(1,I))*(SGG(1,M,I)+SPECT(I)*SNUF(1,M))-DSTM(1,M  WEND0057
     1 ,I)-SPECT(I)*DNSFM(1,M)*EFFK                                     WEND0058
      ADU(I)=ADU(I)+PROS*OLU(M)                                         WEND0059
      DDU(I)=DDU(I)+PROD*OLU(M)                                         WEND0060
   44 CONTINUE                                                          WEND0061
      AF   =THSA(1,I)+THSTT(1,I)-(THD(1,I)/DI(1,I))*SR(1,I)             WEND0062
      DF   =DSAM(1,I)+DSTTM(1,I)-(DDM(1,I)/DI(1,I))*SR(1,I)             WEND0063
      TU(1,I)=AF   *OLU(I)+ADU(I)                                       WEND0064
      DU(1,I)=DF   *OLU(I)+DDU(I)                                       WEND0065
   43 CONTINUE                                                          WEND0066
      KK=1                                                              WEND0067
      MA=0                                                              WEND0068
      DO 45 L=1,NRG                                                     WEND0069
      KK=KK+1                                                           WEND0070
      HSQ=HA(L)*HA(L)                                                   WEND0071
      IF(L.EQ.1) GO TO 46                                               WEND0072
```

```
      MA=MA+NPT(L-1)                                                    WEND0073
      K=NPT(L)-2+MA                                                     WEND0074
      N=MA                                                              WEND0075
      K1=MA+0.5*NPT(L)-2+0.5                                            WEND0076
      N1=K1+2                                                           WEND0077
      K2=K+2                                                            WEND0078
      GO TO 47                                                          WEND0079
46    K=NPT(1)-2                                                        WEND0080
      N=1                                                               WEND0081
      K1=(NPT(1)-1)*0.5-1+0.5                                           WEND0082
      N1=K1+2                                                           WEND0083
      K2=K+2                                                            WEND0084
47    DO 48 I=1,NGP                                                     WEND0085
      OLU(I)=0.0                                                        WEND0086
      DO 2 M=1,NGP                                                      WEND0087
2     OLV(M,I)=0.0                                                      WEND0088
      DO 49 J=N,K1,2                                                    WEND0089
      DO 50 JA=1,3                                                      WEND0090
          JI=J+JA-1                                                     WEND0091
      ARG(JA)=AKTIS(JI)-AKTIS(N)                                        WEND0092
      BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(L))*AKTIS(JI)    WEND0093
      BV(JA)=(ARG(JA)*ARG(JA)/(DI(L,I)*HA(L)))*(-1.0+ARG(JA)/HA(L))*    WEND0094
     1 AKTIS(JI)                                                        WEND0095
50    CONTINUE                                                          WEND0096
      J1=J+1                                                            WEND0097
      J2=J+2                                                            WEND0098
      OLU(I)=OLU(I)+PHL(J,I)*BU(1)+4.0*PHL(J1,I)*BU(2)+PHL(J2,I)*BU(3)  WEND0099
      DO 49 M=1,NGP                                                     WEND0100
      OLV(M,I)=OLV(M,I)+PHL(J,M)*BV(1)+4.0*PHL(J1,M)*BV(2)+PHL(J2,M)*   WEND0101
     1  BV(3)                                                           WEND0102
49    CONTINUE                                                          WEND0103
      DO 51 J=N1,K,2                                                    WEND0104
      DO 52 JA=1,3                                                      WEND0105
      JI=J+JA-1                                                         WEND0106
      ARG(JA)=AKTIS(K2)-AKTIS(JI)                                       WEND0107
      BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(L))*AKTIS(JI)    WEND0108
```

```
      BV(JA)=(ARG(JA)*ARG(JA)/(DI(L,I)*HA(L)))*(1.0-ARG(JA)/HA(L))*
     1 AKTIS(JI)
52    CONTINUE
      J1=J+1
      J2=J+2
      OLU(I)=OLU(I)+PHL(J,I)*BU(1)+4.0*PHL(J1,I)*BU(2)+PHL(J2,I)*BU(3)
      DO 51 M=1,NGP
      OLV(M,I)=OLV(M,I)+PHL(J,M)*BV(1)+4.C*PHL(J1,M)*BV(2)+PHL(J2,M)*
     1  BV(3)
51    CONTINUE
      OLU(I)=OLU(I)*STSI
      DO 48 M=1,NGP
      OLV(M,I)=OLV(M,I)*STSI
48    CONTINUE
      DO 53 I=1,NGP
      ADU(I)=0.0
      DDU(I)=0.0
      ADV(I)=0.0
      DDV(I)=0.0
      DO 54 M=1,NGP
      PROS=(THD(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-THST(L,M
     1 ,I)-SPECT(I)*THNSF(L,M)*EFFK
      PROD=(DDM(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-DSTM(L,M
     1 ,I)-SPECT(I)*DNSFM(L,M)*EFFK
      ADU(I)=ADU(I)+PROS*OLU(M)
      DDU(I)=DDU(I)+PROD*OLU(M)
      ADV(I) =ADV(I)+PROS*OLV(M,I)
      DDV(I) =DDV(I)+PROD*OLV(M,I)
54    CONTINUE
      AF    =THSA(L,I)+THSTT(L,I)-(THD(L,I)/DI(L,I))*SR(L,I)
      DF    =DSAM(L,I)+DSTTM(L,I)-(DDM(L,I)/DI(L,I))*SR(L,I)
      TU(KK,I)=AF    *OLU(I)+ADU(I)
      DU(KK,I)=DF    *OLU(I)+DDU(I)
      TV(KK,I)=AF*OLV(I,I)+ ADV(I)
      DV(KK,I)=DF *OLV(I,I)+DDV(I)
53    CONTINUE
```

```
      KK=KK+1                                                           WENDO145
   45 CONTINUE                                                          WENDO146
      NPT(1)=NPT(1)-1                                                   WENDO147
      K=NPT(1)*0.5+1+0.5                                                WENDO148
      DO 89 L=1,NRE                                                     WENDO149
      KK=NBD(L)                                                         WENDO150
      HSQ=HA(L)*HA(L)                                                   WENDO151
      N=K                                                               WENDO152
      K=K+(NPT(L)+NPT(L+1))*0.5+0.5                                     WENDO153
      K1=N+NPT(L)*0.5-2+0.5                                             WENDO154
      N1=K1+2                                                           WENDO155
      K2=K-2                                                            WENDO156
      DO 85 I=1,NGP                                                     WENDO157
      OLU(I)=0.0                                                        WENDO158
      DO 3 M=1,NGP                                                      WENDO159
    3 OLV(M,I)=0.0                                                      WENDO160
      DO 86 J=N,K1,2                                                    WENDO161
      DO 87 JA=1,3                                                      WENDO162
      JI=J+JA-1                                                         WENDO163
      ARG(JA)=AKTIS(JI)-AKTIS(N)                                        WENDO164
      BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(L))*AKTIS(JI)    WENDO165
      BV(JA)=(ARG(JA)*ARG(JA)/(DI(L,I)*HA(L)))*(-1.0+ARG(JA)/HA(L))*    WENDO166
     1 AKTIS(JI)                                                        WENDO167
   87 CONTINUE                                                          WENDO168
      J1=J+1                                                            WENDO169
      J2=J+2                                                            WENDO170
      OLU(I)=OLU(I)+PHL(J,I)*BU(1)+4.0*PHL(J1,I)*BU(2)+PHL(J2,I)*BU(3)  WENDO171
      DO 86 M=1,NGP                                                     WENDO172
      OLV(M,I)=OLV(M,I)+PHL(J,M)*BV(1)+4.0*PHL(J1,M)*BV(2)+PHL(J2,M)*   WENDO173
     1  BV(3)                                                           WENDO174
   86 CONTINUE                                                          WENDO175
      OLU(I)=OLU(I)*STSI                                                WENDO176
      DO 85 M=1,NGP                                                     WENDO177
      OLV(M,I)=OLV(M,I)*STSI                                            WENDO178
   85 CONTINUE                                                          WENDO179
      DO 88 I=1,NGP                                                     WENDO180
```

```
      ADU(I)=0.0                                                        WEND0181
      DDU(I)=0.0                                                        WEND0182
      ADV(I)=0.0                                                        WEND0183
      DDV(I)=0.0                                                        WEND0184
      DO 60 M=1,NGP                                                     WEND0185
      PROS=(THD(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-THST(L,M  WEND0186
     1 ,I)-SPECT(I)*THNSF(L,M)*EFFK                                     WEND0187
      PROD=(DDM(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-DSTM(L,M  WEND0188
     1 ,I)-SPECT(I)*DNSFM(L,M)*EFFK                                     WEND0189
      ADU(I)=ADU(I)+PROS*OLU(M)                                         WEND0190
      DDU(I)=DDU(I)+PROD*OLU(M)                                         WEND0191
      ADV(I) =ADV(I)+PROS*OLV(M,I)                                      WEND0192
      DDV(I) =DDV(I)+PROD*OLV(M,I)                                      WEND0193
60    CONTINUE                                                          WEND0194
      AF    =THSA(L,I)+THSTT(L,I)-(THD(L,I)/DI(L,I))*SR(L,I)            WEND0195
      DF    =DSAM(L,I)+DSTTM(L,I)-(DDM(L,I)/DI(L,I))*SR(L,I)            WEND0196
      TUL(KK,I)=AF*OLU(I) +ADU(I)                                       WEND0197
      DUL(KK,I)=DF*OLU(I) +DDU(I)                                       WEND0198
      TVL(KK,I)=AF*OLV(I,I)+ADV(I)                                      WEND0199
      DVL(KK,I)=DF*OLV(I,I)+DDV(I)                                      WEND0200
88    CONTINUE                                                          WEND0201
      HSQ=HA(L+1)*HA(L+1)                                               WEND0202
      DO 61 I=1,NGP                                                     WEND0203
      OLU(I)=0.0                                                        WEND0204
      DO 7 M=1,NGP                                                      WEND0205
7     OLV(M,I)=0.0                                                      WEND0206
      DO 62 J=N1,K2,2                                                   WEND0207
      DO 63 JA=1,3                                                      WEND0208
      JI=J+JA-1                                                         WEND0209
      ARG(JA)=AKTIS(K)-AKTIS(JI)                                        WEND0210
      BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(L+1))*AKTIS(JI)  WEND0211
      BV(JA)=(ARG(JA)*ARG(JA)/(DI(L+1,I)*HA(L+1)))*(1.0-ARG(JA)/HA(L+1))WEND0212
     1 *AKTIS(JI)                                                       WEND0213
63    CONTINUE                                                          WEND0214
      J1=J+1                                                            WEND0215
      J2=J+2                                                            WEND0216
```

```
      OLU(I)=OLU(I)+PHL(J,I)*BU(1)+4.0*PHL(J1,I)*BU(2)+PHL(J2,I)*BU(3)
      DO 62 M=1,NGP
      OLV(M,I)=OLV(M,I)+PHL(J,M)*BV(1)+4.0*PHL(J1,M)*BV(2)+PHL(J2,M)*
     1  BV(3)
62    CONTINUE
      OLU(I)=OLU(I)*STSI
      DO 61 M=1,NGP
      OLV(M,I)=OLV(M,I)*STSI
61    CONTINUE
      DO 64 I=1,NGP
      ADU(I)=0.0
      DDU(I)=0.0
      ADV(I)=0.0
      DDV(I)=0.0
      DO 90 M=1,NGP
      PROS=(THD(L+1,I)/DI(L+1,I))*(SGG(L+1,M,I)+SPECT(I)*SNUF(L+1,M))-
     1 THST(L+1,M,I)-SPECT(I)*THNSF(L+1,M) *EFFK
      PROD=(DDM(L+1,I)/DI(L+1,I))*(SGG(L+1,M,I)+SPECT(I)*SNUF(L+1,M))-
     1 DSTM(L+1,M,I)-SPECT(I)*DNSFM(L+1,M)*EFFK
      ADU(I)=ADU(I)+PROS*OLU(M)
      DDU(I)=DDU(I)+PROD*OLU(M)
      ADV(I) =ADV(I)+PROS*OLV(M,I)
      DDV(I) =DDV(I)+PROD*OLV(M,I)
90    CONTINUE
      AF    =THSA(L+1,I)+THSTT(L+1,I)-(THD(L+1,I)/DI(L+1,I))*SR(L+1,I)
      DF    =DSAM(L+1,I)+DSTTM(L+1,I)-(DDM(L+1,I)/DI(L+1,I))*SR(L+1,I)
      TUR(KK,I)=AF    *OLU(I)+ADU(I)
      DUR(KK,I)=DF    *OLU(I)+DDU(I)
      TVR(KK,I)=AF*OLV(I,I)+ADV(I)
      DVR(KK,I)=DF *OLV(I,I)+DDV(I)
64    CONTINUE
89    CONTINUE
      L=NRG
      N=NP-NPT(L)*0.5+0.5
      K=NP-2
      HSQ=HA(L)*HA(L)
```

```
      DO 70 I=1,NGP
      DO 8 M=1,NGP
8     OLV(M,I)=0.0
      DO 71 J=N,K,2
      DO 72 JA=1,3
      JI=J+JA-1
      ARG(JA)=AKTIS(JI)-AKTIS(N)
      BV(JA)=(ARG(JA)*ARG(JA)/(DI(L,I)*HA(L)))*(-1.0+ARG(JA)/HA(L))*
     1 AKTIS(JI)
72    CONTINUE
      J1=J+1
      J2=J+2
      DO 71 M=1,NGP
      OLV(M,I)=OLV(M,I)+PHL(J,M)*BV(1)+4.0*PHL(J1,M)*BV(2)+PHL(J2,M)*
     1  BV(3)
71    CONTINUE
      DO 70 M=1,NGP
      OLV(M,I)=OLV(M,I)*STSI
70    CONTINUE
      DO 73 I=1,NGP
      ADV(I)=0.0
      DDV(I)=0.0
      DO 74 M=1,NGP
      PROS=(THD(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-THST(L,M
     1 ,I)-SPECT(I)*THNSF(L,M)*EFFK
      PROD=(DDM(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-DSTM(L,M
     1 ,I)-SPECT(I)*DNSFM(L,M)*EFFK
      ADV(I) =ADV(I)+PROS*OLV(M,I)
      DDV(I) =DDV(I)+PROD*OLV(M,I)
74    CONTINUE
      AF    =THSA(L,I)+THSTT(L,I)-(THD(L,I)/DI(L,I))*SR(L,I)
      DF    =DSAM(L,I)+DSTTM(L,I)-(DDM(L,I)/DI(L,I))*SR(L,I)
      TVL(NBD(L),I)=AF*OLV(I,I)+ADV(I)
      DVL(NBD(L),I)=DF*OLV(I,I)+DDV(I)
73    CONTINUE
      RETURN
```

END

WEND0289

```
      SUBROUTINE  ELIZA(G,A,C,N,M,L)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),G(1),C(1)
      IR=0
      IK=-M
      DO 92 K=1,L
      IK=IK+M
      DO 92 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      C(IR)=0.0
      DO 92 I=1,M
      JI=JI+N
      IB=IB+1
   92 C(IR)=C(IR)+G(JI)*A(IB)
      RETURN
      END
```

```
      SUBROUTINE DMINV(A,N,D,L,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),L(1),M(1)
C
C     D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
 35   I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
```

```
      HOLD=-A(JK)                                                       DMIN0037
      A(JK)=A(JI)                                                       DMIN0038
   40 A(JI) =HOLD                                                       DMIN0039
   45 IF(BIGA) 48,46,48                                                 DMIN0040
   46 D=0.0                                                             DMIN0041
      RETURN                                                            DMIN0042
   48 DO 55 I=1,N                                                       DMIN0043
      IF(I-K) 50,55,50                                                  DMIN0044
   50 IK=NK+I                                                           DMIN0045
      A(IK)=A(IK)/(-BIGA)                                               DMIN0046
   55 CONTINUE                                                          DMIN0047
      DO 65 I=1,N                                                       DMIN0048
      IK=NK+I                                                           DMIN0049
      HOLD=A(IK)                                                        DMIN0050
      IJ=I-N                                                            DMIN0051
      DO 65 J=1,N                                                       DMIN0052
      IJ=IJ+N                                                           DMIN0053
      IF(I-K) 60,65,60                                                  DMIN0054
   60 IF(J-K) 62,65,62                                                  DMIN0055
   62 KJ=IJ-I+K                                                         DMIN0056
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            DMIN0057
   65 CONTINUE                                                          DMIN0058
      KJ=K-N                                                            DMIN0059
      DO 75 J=1,N                                                       DMIN0060
      KJ=KJ+N                                                           DMIN0061
      IF(J-K) 70,75,70                                                  DMIN0062
   70 A(KJ)=A(KJ)/BIGA                                                  DMIN0063
   75 CONTINUE                                                          DMIN0064
      D=D*BIGA                                                          DMIN0065
      A(KK)=1.0/BIGA                                                    DMIN0066
   80 CONTINUE                                                          DMIN0067
      K=N                                                               DMIN0068
  100 K=(K-1)                                                           DMIN0069
      IF(K) 150,150,105                                                 DMIN0070
  105 I=L(K)                                                            DMIN0071
      IF(I-K) 120,120,108                                               DMIN0072
```

```
108 JQ=N*(K-1)                                      DMIN0073
    JR=N*(I-1)                                      DMIN0074
    DO 110 J=1,N                                    DMIN0075
    JK=JQ+J                                         DMIN0076
    HOLD=A(JK)                                      DMIN0077
    JI=JR+J                                         DMIN0078
    A(JK)=-A(JI)                                    DMIN0079
110 A(JI) =HOLD                                     DMIN0080
120 J=M(K)                                          DMIN0081
    IF(J-K) 100,100,125                             DMIN0082
125 KI=K-N                                          DMIN0083
    DO 130 I=1,N                                    DMIN0084
    KI=KI+N                                         DMIN0085
    HOLD=A(KI)                                      DMIN0086
    JI=KI-K+J                                       DMIN0087
    A(KI)=-A(JI)                                    DMIN0088
130 A(JI) =HOLD                                     DMIN0089
    GO TO 100                                       DMIN0090
150 RETURN                                          DMIN0091
    END                                             DMIN0092
```

```
      SUBROUTINE BASINT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON/COWE/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
     1 VUR(13,5),NOP(10),NBD(10),NOPT,NRE
C
C     INTEGRALS OF BASE POLYNOMIALS
C
      UUR(1)=HA(1)*HA(1)*0.15
      N=1
      R=0.0
      DO 1 K=1,NRG
      M=NOP(K)-1
      DO 2 J=1,M
      R=R+HA(K)
      N=N+1
      DO 3 I=1,NGP
      VUL(N,I)=-HA(K)*HA(K)*((R-HA(K))/(12.0*DI(K,I))+0.05*HA(K)/
     1 DI(K,I))
      VUR(N,I)=HA(K)*HA(K)*((R+HA(K))/(12.0*DI(K,I))-0.05*HA(K)/
     1 DI(K,I))
    3 CONTINUE
      UUL(N)=HA(K)*(0.35*HA(K)+0.5*(R-HA(K)))
      UUR(N)=HA(K)*(-0.35*HA(K)+0.5*(R+HA(K)))
    2 CONTINUE
      N=N+1
      R=R+HA(K)
    1 CONTINUE
```

```
      DO 4 I=1,NGP
      VUL(N,I)=-HA(K)*HA(K)*((R-HA(K))/(12.0*DI(K,I))+0.05*HA(K)/
     1 DI(K,I))
4     CONTINUE
      R=0.0
      DO 5 K=1,NRE
      R=R+NOP(K)*HA(K)
      N=NBD(K)
      DO 6 I=1,NGP
      VUL(N,I)=-HA(K)*HA(K)*((R-HA(K))/(12.0*DI(K,I))+0.05*HA(K)/
     1 DI(K,I))
      VUR(N,I)=HA(K+1)*HA(K+1)*((R+HA(K+1))/(12.0*DI(K+1,I))-0.05*
     1 HA(K+1)/DI(K+1,I))
6     CONTINUE
      UUL(N)=HA(K)*(0.35*HA(K)+0.5*(R-HA(K)))
      UUR(N)=HA(K+1)*(-0.35*HA(K+1)+0.5*(R+HA(K+1)))
5     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LINPRO
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/POWER/ SIGFM(10,5) ,AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
     1FSDIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
     2GRPH2(10,5) ,GRPHA1(10,5) ,GRPHA2(10,5) ,ALKGE(10),GRPH1(10,5),
     3 STS,PHL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
     4 SGG(10,5,5),DI(10,5)
      COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
     1NGP,NRG,NMAT,NGEOM,JBCL,JBCR,NFG,JAD,NP,NPT(10),IOP,NRVARY,
     2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
     3JDUM,IHOLD(90)
      COMMON /MACX/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
     1CONC(10,10),D(10,5),XR(10,5),CC,CT,IDMAT(10)
      COMMON/KSWY/ SB(10,5),PPU(10),PU(10),PDU(10),PRS(10),URN(10),
     1 URC(10),SD(10),DOPL(10)
      COMMON/DELTA/ THSA(10,5),THNSF(10,5),THD(10,5),THST(10,5,5),
     1 THTRP(10,5),THSTT(10,5),DSAM(10,5),DNSFM(10,5),DDM(10,5),
     2 DSTM(10,5,5),DTRPM(10,5),DSTTM(10,5),THSF(10,5),DSFM(10,5),
     3 SFU(10,5),SCU(10,5),SUP(10,5),POWED(10),CONCP(10),VNO
      COMMON/COWE/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
     1 VUR(13,5),NOP(10),NBD(10),NOPT,NRE
      COMMON/CONV/ CRMA(30),NPR,KNA,NCR,IDNA
      COMMON/DELFI/ IP,IU
      COMMON/ITER/ NIT
      DIMENSION CA(13,5,13),CB(13,5,13)
      DIMENSION UO(10),AS(14,17),CS(17),BS(14),P(14),XX(14),Y(14),
     1  PE(14),E(200),KO(6),JH(14)
      REAL*4 X(17)
C
      IF(KNA.EQ.1) GO TO 350
      IF(NIT.NE.0) GO TO 1500
      NVC=NCR
      NAR=1
      NEQ=3*NCR+2
      NAV=4*NCR+1
      READ(5,1000) (UO(L),L=1,NCR)
```

```
      READ(5,2000) PDL,THUO
 1000 FORMAT(7F10.0)
 2000 FORMAT(2F10.0)
      WRITE(6,5500)  (UO(L),L=1,NCR)
      WRITE(6,5600) PDL,THUO
 5600 FORMAT(2F15.7)
 1500 CALL BASINT
      NAGN=2*NRG+1
      DO 15 J=1,NAV
      DO 14 I=1,NEQ
      AS(I,J)=0.0
 14   CONTINUE
      CS(J)=0.0
 15   CONTINUE
      DO 16 I=1,NEQ
      BS(I)=0.0
 16   CONTINUE
      NF=NOPT+1
      CA(1,1,1)=1.0
      DO 100 M=2,NAGN
 100  CA(1,1,M)=0.0
      K=1
      DO 90 I=1,NGP
      IF(I.EQ.1) GO TO 91
      DO 92 M=1,NAGN
 92   CA(1,I,M)=WK(K,M)
      K=K+1
 91   DO 93 J=2,NOPT
      DO 94 M=1,NAGN
      CA(J,I,M)=WK(K,M)
 94   CONTINUE
      K=K+1
      DO 95 M=1,NAGN
      CB(J,I,M)=WK(K,M)
 95   CONTINUE
      K=K+1
```

```
93    CONTINUE
      DO 96 M=1,NAGN
      CB(NF,I,M)=WK(K,M)
96    CONTINUE
      K=K+1
90    CONTINUE
      DO 97 I=1,NGP
      DO 97 M=1,NAGN
      CB(1,I,M)=0.0
      CA(NF,I,M)=0.0
97    CONTINUE
      IF(NAR.EQ.2) GO TO 60
      LL=2
      LU=2*NCR
      GO TO 61
60    LL=2*NCR+3
      LU=2*NRG+1
61    DO 20 M=1,NGP
      COM=SIGFM(1,M)*UUR(1)
      COB=SB(1,M)*UUR(1)
      J=1
      AS(1,1)=AS(1,1)+COM*CA(1,M,1)
      CS(1)=CS(1)+COB*CA(1,M,1)
      DO 27 I=LL,LU,2
      J=J+1
      AS(1,J)=AS(1,J)+COM*CA(1,M,I)
      CS(J)=CS(J)+COB*CA(1,M,I)
27    CONTINUE
      N=1
      DO 22 L=1,NRG
      DO 23 JJ=1,2
      N=N+1
      JI=JJ-1
      NR=L+JI
      COM1=SIGFM(L,M)*UUL(N)+SIGFM(NR,M)*UUR(N)
      COM2=SIGFM(L,M)*VUL(N,M)+SIGFM(NR,M)*VUR(N,M)
```

```
      COB1=SB(L,M)*UUL(N)+SB(NR,M)*UUR(N)
      COB2=SB(L,M)*VUL(N,M)+SB(NR,M)*VUR(N,M)
      J=1
      AS(1,1)=AS(1,1)+COM1*CA(N,M,1)+COM2*CB(N,M,1)
      CS(1)=CS(1)+COB1*CA(N,M,1)+COB2*CB(N,M,1)
      DO 24 I=LL,LU,2
      J=J+1
      AS(1,J)=AS(1,J)+COM1*CA(N,M,I)+CCM2*CB(N,M,I)
      CS(J)=CS(J)+COB1*CA(N,M,I)+COB2*CB(N,M,I)
24    CONTINUE
      IF(L.EQ.NRG) GO TO 25
23    CONTINUE
22    CONTINUE
25    N=N+1
      COM=SIGFM(L,M)*VUL(N,M)
      COB=SB(L,M)*VUL(N,M)
      J=1
      AS(1,J)=AS(1,J)+CCM*CB(N,M,J)
      CS(J)=CS(J)+COB*CB(N,M,J)
      DO 26 I=LL,LU,2
      J=J+1
      AS(1,J)=AS(1,J)+COM*CB(N,M,I)
      CS(J)=CS(J)+COB*CB(N,M,I)
26    CONTINUE
20    CONTINUE
      WRITE(6,900) (CS(J),J=1,NAV)
900   FORMAT(8D15.7)
      LM=NVC+1
      BS(1)=BS(1)+AS(1,1)*PHL(1,1)
      DO 21 J=2,LM
      I=J-1
      BS(1)=BS(1)+AS(1,J)*UO(I)
21    CONTINUE
      IF(NAR.EQ.2) GO TO 62
      DO 63 J=2,LM
      L=J-1
```

```
      AS(1,J)=AS(1,J)+PPU(L)
      CS(J)=CS(J)+PU(L)
      BS(1)=BS(1)+PPU(L) *UO(L)
      PERT   =FISIT(L)-SYLI(L)-TMETOL(L)-ALKGE(L)
      AS(2,J)=PERT
      BS(2)=BS(2)+UO(L)*PERT
   63 CONTINUE
      GO TO 64
   62 L=NCR
      DO 30 J=2,LM
      L=L+1
      M=J-1
      AS(1,J)=AS(1,J)-URN(L)
      CS(J)=CS(J)+URC(L)
      BS(1)=BS(1)-URN(L)*UO(M)
   30 CONTINUE
C
   64 L=0
      N=-1
      IF(NAR.EQ.2) GO TO 70
      LN=2+NCR
      LI=3
      GO TO 71
   70 LN=1+NCR
      LI=2
   71 DO 40 I=LI,LN
      L=L+1
      N=N+2
      DO 40 M=1,NGP
      J=1
      AS(I,J)=AS(I,J)+SIGFM(L,M)*CA(N,M,J)
      DO 40 K=LL,LU,2
      J=J+1
      AS(I,J)=AS(I,J)+SIGFM(L,M)*CA(N,M,K)
   40 CONTINUE
      DO 28 I=LI,LN
```

```
      BS(I)=BS(I)+AS(I,1)*PHL(1,1)
      DO 28 J=2,LM
      L=J-1
      BS(I)=BS(I)+AS(I,J)*UO(L)
   28 CONTINUE
      IF(NAR.EQ.2) GO TO 65
      DO 66 I=3,LN
      J=I-1
      L=J-1
      AS(I,J)=AS(I,J)+PDU(L)
   66 CONTINUE
      DO 67 L=1,NCR
      I=2+L
      BS(I)=PDL-PRS(L)+BS(I)
   67 CONTINUE
      GO TO 68
   65 DO 45 L=1,NCR
      I=1+L
      BS( I)=BS(I)+PDL-POWED(L)
   45 CONTINUE
   68 K=LN+1
      KA=LN+NVC
      L=0
      DO 50 I=K,KA
      L=L+1
      J=L+1
      AS(I,J)=1.0
      BS(I)=UO(L)-THUO
      M=I+NVC
      AS(M,J)=1.0
      BS(M)=UO(L)+THUO
   50 CONTINUE
      GO TO(201,201,203),NPR
  203 CS(1)=0.0
      CS(2)=TH(1)*TH(1)
      R1=0.0
```

```
      R2=TH(1)
      DO 85 K=2,NCR
      J=K+1
      R1=R1+TH(K-1)
      R2=R2+TH(K)
      CS(J)=R2*R2-R1*R1
 85   CONTINUE
C
C     SLACK VARIABLES
C
 201  J=NVC+1
      DO 55 I=LI,LN
      J=J+1
      AS(I,J)=1.0
 55   CONTINUE
      K=LN+1
      KA=LN+NVC
      M=NEQ+1
      L=NAV+1
      DO 56 I=K,KA
      J=J+1
      AS(I,J)=-1.0
      M=M-1
      L=L-1
      AS(M,L)=1.0
 56   CONTINUE
      IF(NPR.NE.1) GO TO 332
      DO 57 J=1,NAV
      CS(J)=-CS(J)
 57   CONTINUE
 332  IF(NPR.NE.2) GO TO 335
      KNA=1
      RETURN
 350  I=NCR+1
      L=0
      DO 340 J=2,I
```

```
      L=L+1
      PERT   =FISIT(L)-SYLI(L)-TMETOL(L)-ALKGE(L)
340   CS(J)=PERT
      CS(1)=0.0
335   II=0
      MX=NEQ
      NN=NAV
      CALL SIMPLE(II,MX,NN,AS,BS,CS,KO,X,P,JH,XX,Y,PE,E)
      KNA=0
      IF(NPR.NE.3) GO TO 334
      CM=0.0
      I=NCR+1
      DO 86 J=2,I
      CM=CM+CS(J)*X(J)
86    CONTINUE
      CM=0.0313881267*CM
      WRITE(6,5900) CM
334   WRITE(6,6000) KO(1)
6000  FORMAT(///' FEASIBILITY=',I2)
5900  FORMAT(///' CRITICAL MASS  IN KG =', 1PD15.7)
5500  FORMAT(10D12.4)
      M=NVC+1
      L=0
      DO 80 J=2,M
      L=L+1
      UO(L)=X(J)
80    CONTINUE
      WRITE(6,5000) (L,UO(L),L=1,NCR)
5000  FORMAT(' REGION', I5,10X,'FISSILE VOLUME FRACTION=',D15.7)
      DO 81 K=1,NCR
      CONC(IP,K)=CONCP(IP)*UO(K)
      CONC(IU,K)=CONCP(IU)*(0.35-UO(K))
81    CONTINUE
      NIT=NIT+1
      WRITE(6,6100) NIT
6100  FORMAT(///' NUMBER OF ITERATIONS=',I5)
```

```
      IF(KO(1).EQ.1) CALL EXIT
      GO TO(301,301,303),NPR
  303 CRMA(NIT)=CM
      IF(DABS(CRMA(NIT)-CRMA(NIT-1)).LT.0.001)  CALL EXIT
  301 RETURN
      END
```

```
      SUBROUTINE SIMPLE(INFLAG,MX,NN,A,B,C,KO,KB,P,JH,X,Y,PE,E)
      IMPLICIT REAL*8 (A-H,O-Z)
C  AUTOMATIC SIMPLEX        REDUNDANT EQUATIONS CAUSE INFEASIBILITY
      DIMENSION B(1),C(1),P(1),X(1),Y(1),PE(1),E(1)
      REAL*4 XX
      INTEGER INFLAG,MX,NN,KO(6),KB(1),JH(1)
      EQUIVALENCE (XX,LL)
      DIMENSION A(14,17)
      INTEGER I,IA,INVC,IR,ITER,J,JT,K,KBJ,L,LL,M,M2,MM,N
      INTEGER NCUT,NPIV,NUMVR,NVER
      LOGICAL FEAS,VER,NEG,TRIG,KQ,ABSC
C
C          SET INITIAL VALUES, SET CONSTANT VALUES
      ITER = 0
      NUMVR = 0
      NUMPV = 0
      M = MX
      N = NN
      TEXP  =  .000015259
      NCUT  =  4*M + 200
      NVER  =  M*.5  +  5
      M2 = M*M
      FEAS = .FALSE.
      IF  (INFLAG.NE.0)  GO TO 1400
C* 'NEW'    START  PHASE ONE WITH SINGLETON BASIS
      DO 1402  J = 1,N
        KB(J) = 0
        KQ = .FALSE.
        DO 1403  I = 1,M
          IF (A(I,J).EQ.0.0)  GO TO 1403
          IF (KQ.OR.A(I,J).LT.0.0) GO TO 1402
          KQ = .TRUE.
 1403   CONTINUE
        KB(J) = 1
 1402 CONTINUE
 1400 DO 1401  I = 1,M
```

```
        JH (I) = -1
 1401 CONTINUE
C* 'VER'    CREATE INVERSE FROM 'KB' AND 'JH'      (STEP 7)
 1320 VER = .TRUE.
      INVC = 0
      NUMVR   =  NUMVR  +1
      TRIG = .FALSE.
      DO 1101  I = 1,M2
        E(I) = 0.0
 1101 CONTINUE
      MM=1
      DO 1113  I = 1,M
        E(MM) = 1.0
        PE(I)  =  0.0
        X(I) =  B(I)
        IF (JH(I) .NE.0) JH(I) = -1
        MM = MM + M + 1
 1113 CONTINUE
C                FORM INVERSE
      DO 1102  JT = 1,N
        IF (KB(JT).EQ.0)  GO TO 1102
        GO TO 600
C 600   CALL JMY
C                           CHOOSE PIVOT
 1114   TY = 0.0
        KQ = .FALSE.
        DO 1104 I = 1,M
          IF (JH(I).NE.-1.OR.DABS(Y(I)).LE.TPIV) GO TO 1104
          IF (KQ) GO TO 1116
          IF (X(I).EQ.0.)  GO TO 1115
      IF     (DABS(Y(I)/X(I)).LE.TY)  GO TO 1104
        TY =DABS(Y(I)/X(I))
          GO TO 1118
 1115     KQ = .TRUE.
          GO TO 1117
 1116     IF (X(I).NE.0..OR.DABS(Y(I)).LE.TY)  GO TO 1104
```

```
 1117     TY =DABS(Y(I))
 1118     IR = I
 1104   CONTINUE
        KB(JT) = 0
C                       TEST PIVOT
        IF (TY.LE.0.)    GO TO 1102
C                       PIVOT
        GO TO 900
C 900   CALL PIV
 1102 CONTINUE
C                  RESET ARTIFICIALS
      DO 1109  I = 1,M
        IF (JH(I).EQ.-1)  JH(I) = 0
        IF (JH(I).EQ.0)  FEAS = .FALSE.
 1109 CONTINUE
 1200 VER = .FALSE.
C                 ***          PERFORM ONE ITERATION          ***
C* 'XCK'    DETERMINE FEASIBILITY                    (STEP 1)
      NEG = .FALSE.
      IF (FEAS)  GO TO 500
         FEAS= .TRUE.
         DO 1201   I = 1,M
        IF (X(I).LT.0.0)  GO TO 1250
        IF (JH(I).EQ.0)  FEAS = .FALSE.
 1201 CONTINUE
C* 'GET'    GET APPLICABLE PRICES                    (STEP 2)
      IF (.NOT.FEAS)  GO TO 501
  500 DO 503 I = 1,M
        P(I) = PE(I)
        IF (X(I).LT.0.)  X(I) = 0.
  503 CONTINUE
      ABSC = .FALSE.
      GO TO 599
 1250 FEAS = .FALSE.
      NEG  = .TRUE.
  501 DO 504  J = 1, M
```

```
        P(J) = 0.
  504 CONTINUE
      ABSC = .TRUE.
      DO 505  I = 1,M
        MM = I
        IF (X(I).GE.0.0)  GO TO 507
        ABSC = .FALSE.
        DO 508 J = 1,M
          P(J) = P(J) + E(MM)
          MM = MM + M
  508   CONTINUE
        GO TO 505
  507   IF (JH(I).NE.0) GO TO 505
        IF (X(I).NE.0.) ABSC = .FALSE.
        DO 510  J = 1,M
          P(J) = P(J) - E(MM)
          MM = MM + M
  510   CONTINUE
  505 CONTINUE
C* 'MIN'    FIND MINIMUM REDUCED COST                (STEP 3)
  599 JT = 0
      BB  =  0.0
      DO 701  J =1,N
        IF (KB(J).NE.0)   GO TO 701
        DT = 0.0
        DO 303 I = 1,M
          DT = DT + P(I) * A(I,J)
  303   CONTINUE
        IF (FEAS)  DT = DT + C(J)
        IF  (ABSC)  DT = -DABS(DT)
        IF (DT.GE.BB)  GO TO 701
        BB = DT
        JT = J
  701 CONTINUE
C  TEST FOR NO PIVOT COLUMN
      IF (JT.LE.0)  GO TO 203
```

```
C  TEST FOR ITERATION LIMIT EXCEEDED
      IF (ITER.GE.NCUT)  GO TO 160
      ITER = ITER +1
C* 'JMY'    MULTIPLY INVERSE TIMES A(.,JT)          (STEP 4)
  600 DO 610  I= 1,M
        Y(I) = 0.0
  610 CONTINUE
      LL = 0
      COST = C(JT)
      DO 605   I= 1,M
        AIJT = A(I,JT)
        IF (AIJT.EQ.0.) GO TO 602
        COST = COST + AIJT * PE(I)
        DO  606  J = 1,M
          LL = LL + 1
          Y(J) = Y(J) + AIJT * E(LL)
  606   CONTINUE
        GO TO 605
  602   LL = LL + M
  605 CONTINUE
C         COMPUTE PIVOT TOLERANCE
      YMAX = 0.0
      DO 620  I  =  1,M
        YMAX = DMAX1(DABS(Y(I)),YMAX )
  620 CONTINUE
      TPIV  =  YMAX * TEXP
C             RETURN TO INVERSION ROUTINE, IF INVERTING
      IF (VER)  GO TO 1114
C        COST TOLERANCE CONTROL
      RCOST = YMAX/BB
      IF (TRIG.AND.BB.GE.-TPIV)  GO TO 203
      TRIG = .FALSE.
      IF (BB.GE.-TPIV)  TRIG = .TRUE.
C* 'ROW'    SELECT PIVOT ROW                        (STEP 5)
C AMONG EQS. WITH X=0, FIND MAXIMUM Y  AMONG ARTIFICIALS, OR, IF NONE,
C  GET MAX POSITIVE Y(I) AMONG REALS.
```

```
      IR = 0
      AA = 0.0
      KQ = .FALSE.
      DO  1050  I =1,M
        IF (X(I).NE.0.0.OR.Y(I).LE.TPIV)  GO TO 1050
        IF (JH(I).EQ.0)  GO TO 1044
       IF (KQ) GO TO 1050
 1045   IF (Y(I).LE.AA)  GO TO 1050
        GO TO 1047
 1044   IF (KQ) GO TO 1045
        KQ = .TRUE.
 1047   AA = Y(I)
        IR = I
 1050 CONTINUE
      IF (IR.NE.0)  GO TO 1099
      AA = 1.0E+20
C                FIND MIN. PIVOT AMONG POSITIVE EQUATIONS
      DO 1010  I  = 1,M
        IF (Y(I).LE.TPIV.OR.X(I).LE.0.0.OR.Y(I)*AA.LE.X(I) ) GO TO 1010
        AA = X(I)/Y(I)
        IR = I
 1010 CONTINUE
      IF (.NOT.NEG)  GO TO 1099
C  FIND PIVOT AMONG NEGATIVE EQUATIONS, IN WHICH X/Y IS LESS THAN THE
C MINIMUM X/Y IN THE POSITIVE EQUATIONS, THAT HAS THE LARGEST ABSF(Y)
         BB = - TPIV
      DO 1030  I = 1,M
        IF (X(I).GE.0..OR.Y(I).GE.BB.OR.Y(I)*AA.GT.X(I) )  GO TO 1030
        BB = Y(I)
        IR = I
 1030 CONTINUE
C  TEST FOR NO PIVOT ROW
 1099 IF  (IR.LE.0)  GO TO 207
C* 'PIV'    PIVOT ON (IR,JT)                    (STEP d)
      IA = JH(IR)
      IF (IA.GT.0)  KB(IA) = 0
```

```
 900   NUMPV=NUMPV+1
       JH(IR) = JT
       KB(JT) = IR
       YI = -Y(IR)
       Y(IR) = -1.0
       LL = 0
C                                   TRANSFORM INVERSE
       DO  904  J = 1,M
         L = LL + IR
         IF (E(L).NE.0.0)  GO TO 905
         LL = LL + M
         GO TO 904
  905    XY = E(L) / YI
         PE(J) = PE(J) + COST * XY
         E(L) = 0.0
         DO 906  I = 1,M
           LL  =  LL + 1
 906   CONTINUE
  904  CONTINUE
C                                   TRANSFORM X
       XY  =  X(IR) / YI
       DO 908  I  = 1, M
         XOLD = X(I)
         X(I) = XOLD + XY * Y(I)
         IF (.NOT.VER.AND.X(I).LT.0..AND.XOLD.GE.0.)  X(I) = 0.
  908  CONTINUE
       Y(IR) = -YI
       X(IR) = -XY
       IF (VER) GO TO 1102
       IF (NUMPV.LE.M) GO TO 1200
C  TEST FOR INVERSION ON THIS ITERATION
       INVC   = INVC   +1
       IF (INVC.EQ.NVER) GO TO 1320
       GO  TO  1200
C*  END OF ALGORITHM, SET EXIT VALUES              ***
  207  IF (.NOT.FEAS.OR.RCOST.LE.-1000.)  GO TO 203
```

```
C              INFINITE SOLUTION                                      SIMP0253
      K  =  2                                                         SIMP0254
      GO TO 250                                                       SIMP0255
C            PROBLEM IS CYCLING                                       SIMP0256
  160 K  =  4                                                         SIMP0257
      GO TO 250                                                       SIMP0258
C            FEASIBLE OR INFEASIBLE SOLUTION                          SIMP0259
  203 K  =  0                                                         SIMP0260
  250 IF (.NOT.FEAS)  K =    + 1                                      SIMP0261
      DO 1399  J = 1,N                                                SIMP0262
        XX  =  0.0                                                    SIMP0263
        KBJ  =  KB(J)                                                 SIMP0264
        IF (KBJ.NE.0)  XX = X(KBJ)                                    SIMP0265
        KB(J) = LL                                                    SIMP0266
 1399 CONTINUE                                                        SIMP0267
      KO(1) = K                                                       SIMP0268
      KO(2) = ITER                                                    SIMP0269
      KO(3) = INVC                                                    SIMP0270
      KO(4) = NUMVR                                                   SIMP0271
      KO(5) = NUMPV                                                   SIMP0272
      KO(6) = JT                                                      SIMP0273
      RETURN                                                          SIMP0274
      END                                                             SIMP0275
```

REFERENCES

1. Hasnain, S. A. and D. Okrent, "On the Design and Management of Fast Reactor Blankets", Nucl. Sci. Eng., 9, 314 (1961).

2. Perks, M. A. and R. M. Lord, "Effects of Axial and Radial Blanket Design on Breeding and Economics", ANL-6792 (October, 1963).

3. Evans, P. V., Ed., "Fast Breeder Reactors. Proceedings of the London Conference on Fast Breeder Reactors organized by the British Nuclear Energy Society. May 1966", Pergamon Press, Oxford, 1967.

4. Ott, K. O., "An Improved Definition of the Breeding Ratio for Fast Reactors", Trans. Amer. Nucl. Soc., Vol. 12, No. 2, 719 (1969).

5. Spinrad, B. I., "On the Definition of Breeding", ANL-6122 (October, 1959).

6. Goertzel, G., "Minimum Critical Mass and Flat Flux", J. Nucl. Energy, 2, 193 (1956).

7. Kochurov, B. P., "Minimum Critical Mass at Limited Uranium Concentration", Soviet Atomic Energy, Vol. 20, No. 3, 267 (1966).

8. Goldschmidt, P. and J. Quenon, "Minimum Critical Mass in Fast Reactors with Bounded Power Density", Nucl. Sci. Eng., 39, 311 (1970).

9. Zaritskaya, T. S., and A. P. Rudik, "Using L. S. Pontryagin's Maximum Principle in Minimum-Critical-Size and Maximum-Power Reactor Problems", Soviet Atomic Energy, Vol. 22, No. 1, 5 (1967).

10. Rosztoczy, Z. R. and L. E. Weaver, "Optimum Reactor Shutdown Program for Minimum Xenon Buildup", Nucl. Sci. Eng., 20, 318 (1964).

11. Roberts, J. J. and P. Smith, "Time Optimal Solution to the Reactivity-Xenon Shutdown Problem", Nucl. Sci. Eng., 22, 470 (1965).

12. Ash, M., "Application of Dynamic Programming to Optimal Shutdown Control", Nucl. Sci. Eng., 24, 77 (1966).

13. Wall, I. and H. Fenech, "The Application of Dynamic Programming to Fuel Management Optimization", Nucl. Sci. Eng., 22, 285 (1965).

14. Gandini, A., M. Salvatores and G. Sena, "Use of Generalized Perturbation Methods for Optimization of Reactor Design", J. Nucl. Energy, 23, 469 (1969).

15. Purica, I., M. Pavelescou and V. Anton, "Optimizing the Initial Conversion Ratio of Fast Breeders", JPRS 48331, Vol. 2, (July, 1969).

16. Hadley, G., "Linear Programming", Addison-Wesley, Reading, Massachusetts, 1962.

17. Lamarsh, J. R., "Introduction to Nuclear Reactor Theory", Addison-Wesley, Reading, Massachusetts, 1966.

18. Rohan, P. E., "Comparisons of Transport and Diffusion Theory Calculations of Performance Characteristics for Large Fast Reactors", University of Illinois, Urbana, 1970.

19. Kang, C. M., Tentative title: "Finite Element Methods for Space-Time Reactor Problems", Ph.D. Thesis, Department of Nuclear Engineering, M.I.T., Expected publication date: September, 1971.

20. Loewenstein, W. B. and G. W. Main, "Fast Reactor Shape Factors and Shape-Dependent Variables", ANL-6403 (November, 1961).

21. "Proceedings of the Conference on Safety, Fuels and Core Design in Large, Fast Power Reactors", ANL-7120 (October, 1965).

22. "Proceedings of the International Conference on Sodium Technology and Large Fast Reactor Design", ANL-7520 (November, 1968).

23. Hansen, H. E. and W. H. Roach, "Six and Sixteen Group Cross Sections for Fast and Intermediate Critical Assemblies", LAMS-2543 (December, 1961).

24. Shapiro, M. M., "Minimum Critical Mass in Variable Density and Epithermal Reactors", Nucl. Sci. Eng., 10, 159 (1961).

25. Devooght, J., "Restricted Minimum Critical Mass", Nucl. Sci. Eng., 5, 190 (1959).

26. Hofmann, P. L. and H. Hurwitz, "Application of Minimum Loading Conditions to Enriched Lattices", Nucl. Sci. Eng., 2, 461 (1957).

27. Otsuka, M., "Fuel-Importance Function and Minimum Critical Mass", Nucl. Sci. Eng., 18, 514 (1964).

28. Monroe, C. W., "Sodium Void Coefficient of Reactivity. A Review and Study of Analytical Work", UNC-5123 (May, 1965).

29. Hummel, H. H. and D. Okrent, "Reactivity Coefficients in Large Fast Power Reactors", American Nuclear Society, 1970.

30. "Large Fast Reactor Design Study", ACNP-64503 (January, 1964).

31. Gass, S. I., "Linear Programming: Methods and Applications", 2nd ed. McGraw-Hill, New York, 1964.

32. Dantzig, G. B., "Linear Programming and Extensions", Princeton Univ. Press, Princeton, New Jersey, 1963.

33. Weaver, L. E., "Reactor Dynamics and Control", Americal Elsevier, New York, 1968.