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
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
Atoms 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>4</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>5</td>
</tr>
<tr>
<td>List of Figures</td>
<td>8</td>
</tr>
<tr>
<td>List of Tables</td>
<td>9</td>
</tr>
<tr>
<td>Chapter 1. Introduction</td>
<td>11</td>
</tr>
<tr>
<td>1.1 The Problem</td>
<td>11</td>
</tr>
<tr>
<td>1.2 The Breeding Ratio and Breeding Gain</td>
<td>14</td>
</tr>
<tr>
<td>1.3 Optimization Techniques</td>
<td>16</td>
</tr>
<tr>
<td>1.4 Report Outline</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 2. The Optimization Method</td>
<td>20</td>
</tr>
<tr>
<td>2.1 Mathematical Statement of the Problem</td>
<td>20</td>
</tr>
<tr>
<td>2.2 The Linearized Form of the Breeding</td>
<td>24</td>
</tr>
<tr>
<td>Optimization Problem</td>
<td></td>
</tr>
<tr>
<td>2.3 Solution of the Linearized Multigroup</td>
<td>30</td>
</tr>
<tr>
<td>Diffusion Equations</td>
<td></td>
</tr>
<tr>
<td>2.4 The Iterative Scheme</td>
<td>32</td>
</tr>
<tr>
<td>2.5 Remarks</td>
<td>34</td>
</tr>
<tr>
<td>2.6 Summary</td>
<td>35</td>
</tr>
<tr>
<td>Chapter 3. Core Optimization</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>36</td>
</tr>
<tr>
<td>3.2 Breeding Optimization</td>
<td>36</td>
</tr>
<tr>
<td>3.3 Critical Mass Optimization</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Sodium Void Reactivity Optimization</td>
<td>47</td>
</tr>
<tr>
<td>3.5 Summary</td>
<td>52</td>
</tr>
<tr>
<td>Chapter 4. Blanket Optimization</td>
<td>56</td>
</tr>
<tr>
<td>4.1 The Effect of Blanket Moderation</td>
<td>57</td>
</tr>
<tr>
<td>4.2 The Effect of the Reflector Composition</td>
<td>64</td>
</tr>
<tr>
<td>Chapter 5. Conclusions and Recommendations</td>
<td>70</td>
</tr>
<tr>
<td>5.1 Conclusions</td>
<td>70</td>
</tr>
<tr>
<td>5.2 Recommendations for Future Work</td>
<td>72</td>
</tr>
<tr>
<td>Appendix A. Bibliography</td>
<td>76</td>
</tr>
<tr>
<td>Appendix B. Linear Programming and Linearization</td>
<td>80</td>
</tr>
<tr>
<td>B.1 Linear Programming</td>
<td>80</td>
</tr>
<tr>
<td>B.2 Linearization</td>
<td>81</td>
</tr>
<tr>
<td>Appendix C. The Method of Piecewise Polynomials, and Integrals of Piecewise Polynomials</td>
<td>85</td>
</tr>
<tr>
<td>C.1 The Method of Piecewise Polynomials</td>
<td>85</td>
</tr>
<tr>
<td>C.2 Integrals of Piecewise Polynomials</td>
<td>89</td>
</tr>
</tbody>
</table>
Appendix D. The Computer Program Greko

D.1 Introduction 95
D.2 Input 97
D.3 Output 100
D.4 Listing 102

References 188
LIST OF FIGURES

2.1 Schematic Representation of LMFBR Cylindrical Geometry ........................................... 21
C.1 The Cubic Piecewise Polynomials $w_k$ and $v_{k,i}$ ............................................................. 87
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Dimensions of Reactor No. 1</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>Reactor Composition</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Five-Group Cross Section Set Structure</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Fissile Composition and Breeding Gain as a Function of Linear Programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Iteration Number for Reactor No. 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Peak Power Densities for Reactor No. 1</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>Fissile Composition and Breeding Gain as a Function of Linear Programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Iteration Number for Reactor No. 1 and Different Starting Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>Dimensions of Reactor No. 2</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>Optimum Configuration of Reactor No. 2</td>
<td>46</td>
</tr>
<tr>
<td>3.9</td>
<td>Effect of Blanket Reflector on Breeding Gain</td>
<td>48</td>
</tr>
<tr>
<td>3.10</td>
<td>Fissile Composition and Critical Mass as a Function of Linear Programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Iteration Number for Reactor No. 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td>3.11</td>
<td>Fissile Composition and k-effective of Sodium Voided Reactor as a Function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>of Linear Programming Iteration Number for Reactor No. 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>55</td>
</tr>
<tr>
<td>Table No.</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.1</td>
<td>Dimensions of Reactor used in Blanket Studies</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Reactor Composition for BeO Moderated Blanket</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Reactor Composition for Na Moderated Blanket</td>
<td>61</td>
</tr>
<tr>
<td>4.4</td>
<td>The Breeding Gain as a Function of Moderator Concentration in the Blanket</td>
<td>62</td>
</tr>
<tr>
<td>4.5</td>
<td>The Breeding Gain as a Function of the Reflector Material and Blanket Thickness</td>
<td>65</td>
</tr>
<tr>
<td>4.6</td>
<td>The Breeding Gain as a Function of BeO Reflector Properties</td>
<td>67</td>
</tr>
<tr>
<td>4.7</td>
<td>The Effect of Resonance Self-Shielding on Breeding Gain</td>
<td>68</td>
</tr>
</tbody>
</table>
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 UO₂ 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)

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

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

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

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

$$b_0 = \frac{R_238 + \gamma_0 R_239 + \gamma_1 R_240 + \gamma_2 R_241}{R_239 + \gamma_0 R_240 + \gamma_1 R_241 + \gamma_2 R_242}$$

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

$$\gamma_i = \frac{N_i}{\sum_i N_i}, \text{ } i = \text{Pu}^{240}, \text{Pu}^{241}, \text{Pu}^{242}$$

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 $^{238}\text{U}$ 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 non-
linear, 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 recom-
mendations 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)

\[
\nabla D_i(r) \nabla \phi_i(r) - \Sigma_{a,i}(r) \phi_i(r) - \sum_{h=i+1}^{N} \Sigma_{(i-h)}(r) \phi_i(r) + \\
\sum_{h=1}^{N} \sum_{(h-i)} \phi_h(r) + \chi_i \varepsilon \sum_{h=1}^{N} \phi_h(r) = 0
\]

(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 \(i\)

\(\varepsilon_h\) = number of neutrons released per fission occurring in group \(h\)

\(\varepsilon_{f,h}\) = 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

\[
P(r) = \sum_{i=1}^{N} \left( \varepsilon_{f,i} \phi_i(r) + [N_0 - \varepsilon_{f,i}] \varepsilon_{fr,i} \phi_i(r) \right)
\]

(2.2)

where

\(\varepsilon_{f,i}\) = volume fraction of the fissile material

\(\varepsilon_{m}\) = volume fraction of the moderating material
\( \Sigma_{fs,i} \) = macroscopic fission cross section of pure fissile material for group \( i \)

\( \Sigma_{fr,i} \) = 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

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

(2.3)

where

\( t_f \) = outer reactor radius

The breeding gain as defined in Section 1.2 is written as

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

(2.4)

where

\( \Sigma_{fr,i} \) = macroscopic capture cross section of pure fertile material for group \( i \)

\( \Sigma_{fs,a,i} \) = 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
   \[
   P(r) \leq p = \text{const.}
   \]  
   (2.5)
3. The total thermal power
   \[
   W = \text{const.}
   \]  
   (2.6)
4. The sum of fissile and moderator volume fractions
   \[
   u_m + u_f \leq N_0 = \text{const.}
   \]  
   (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

\[
BG = \frac{2\pi}{W} \left\{ - \int_0^{t_f} u_f(r) \sum_{i=1}^{N} \left( \Sigma_{Y, i}^{fr} + \Sigma_{a, i}^{fs} \right) \phi_i^0(r) r dr - \int_0^{t_f} u_m(r) \sum_{i=1}^{N} \frac{\Sigma_{Y, i}^{fr} \phi_i^0(r)}{r} r dr + \int_0^{t_f} \left[ (N_0 - u_f^0(r) - u_m^0(r)) \Sigma_{Y, i}^{fr} - u_f^0(r) \Sigma_{a, i}^{fs} \right] \phi_i^*(r) r dr + \right.
\]

\[
\left. \int_0^{t_f} N_0 \sum_{i=1}^{N} \frac{\Sigma_{Y, i}^{fr} \phi_i^0(r)}{r} r dr \right\}
\]

(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

\[
\phi_i^*(r) = \phi_i(r) - \phi_i^0(r)
\]

(2.9)
2. Linearized multigroup diffusion equations

\[
\frac{1}{r} \frac{d}{dr} \left[ r b_i^0(r) \frac{d}{dr} \phi_i^*(r) \right] - \Sigma_{a,i}^0(r) \phi_i^*(r) - \sum_{h=i+1}^{N} \Sigma_{(i-h)}^0(r) \phi_i^*(r) + \\
i-1
\sum_{h=1}^{N} \Sigma_{(h-i)}^0(r) \phi_i^*(r) + \chi_i \Sigma_{h=1}^{N} \nu_{f,h}^0(r) \phi_h^* + \\
[u_f(r) - u_f^0(r)] - \left[ \Sigma_{a,i}^fr - \Sigma_{a,i}^{fr} \right] \phi_i^0(r) - \sum_{h=i+1}^{N} \left[ \Sigma_{(i-h)}^{fs} - \Sigma_{(i-h)}^{fr} \right] \phi_i^0(r) + \\
i-1
\sum_{h=1}^{N} \left[ \Sigma_{(h-i)}^{fs} - \Sigma_{(h-i)}^{fr} \right] \phi_h^0(r) + \chi_i \sum_{h=1}^{N} \left[ \nu_{f,h}^{fr} - \nu_{f,h}^{fr} \right] \phi_h^0(r) - \\
[u_f(r) - u_f^0(r)] - \left[ \Sigma_{a,i}^{fr} - \Sigma_{a,i}^{fr} \right] \phi_i^0(r) - \sum_{h=i+1}^{N} \left[ \Sigma_{(i-h)}^{fr} - \Sigma_{(i-h)}^{fr} \right] \phi_i^0(r) + \\
i-1
\sum_{h=1}^{N} \left[ \Sigma_{(h-i)}^{fr} - \Sigma_{(h-i)}^{fr} \right] \phi_h^0(r) + \chi_i \sum_{h=1}^{N} \left[ \nu_{f,h}^{fr} - \nu_{f,h}^{fr} \right] \phi_h^0(r) - \\
\frac{\Sigma_{tr,i}^{fr} \Sigma_{tr,i}^0 \frac{1}{r} \frac{d}{dr} \left[ r \frac{d \phi_i^0(r)}{dr} \right]}{3 \Sigma_{tr,i}^0} = 0
\]

(2.10)
where

\[ \Sigma_{tr,i} = \text{macroscopic transport cross section for group } i \]

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

3. **Linearized total thermal power**

\[
W = \int_0^{t_f} u_f(r) \left( \Sigma \left[ \Sigma_{fs,i} - \Sigma_{fr,i} \right] \phi_i^0(r) \right) dr - \int_0^{t_f} u_m(r) \left( \Sigma \Sigma_{fr,i} \phi_i^0(r) \right) dr + \int_0^{t_f} \left( \sum_{i=1}^{N} \Sigma_{fs,i} \phi_i^*(r) \right) dr + \int_0^{t_f} \left( \sum_{i=1}^{N} \Sigma_{fr,i} \phi_i^0(r) \right) dr
\]

(2.11)

4. **Linearized power density**

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

(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):

$$
\int_{0}^{t_f} -[u_f(r)-u_0^f(r)] N \sum_{i=1}^{fs} \sum_{r_{tr,i}} -\sum_{r_f,i} \frac{\nabla \phi_i^0(r) \nabla \psi_i^0(r)}{2} r dr +
$$

$$
\int_{0}^{t_f} [u_f(r)-u_0^f(r)] N \sum_{i=1}^{fs} \sum_{a, i} \phi_i^0(r) \psi_i^0(r) r dr +
$$

$$
\int_{0}^{t_f} [u_f(r)-u_0^f(r)] \frac{N}{k} \sum_{h=1}^{l_f} \sum_{i=1}^{fs} \sum_{r_{f,h}} -\sum_{r_{f,h}, i} \frac{\nabla \phi_i^0(r) \nabla \psi_i^0(r)}{2} r dr -
$$

$$
\int_{0}^{t_f} [u_m(r)-u_0^m(r)] N \sum_{i=1}^{m} \sum_{r_{tr,i}} -\sum_{r_f,i} \frac{\nabla \phi_i^0(r) \nabla \psi_i^0(r)}{2} r dr +
$$

$$
\int_{0}^{t_f} [u_m(r)-u_0^m(r)] N \sum_{i=1}^{m} \sum_{a, i} \phi_i^0(r) \psi_i^0(r) r dr +
$$

$$
\int_{0}^{t_f} [u_m(r)-u_0^m(r)] \frac{N}{k} \sum_{h=i+1}^{l_f} \sum_{r_{f,h}} -\sum_{r_{f,h}, i} \frac{\nabla \phi_i^0(r) \nabla \psi_i^0(r)}{2} r dr -
$$

$$
\int_{0}^{t_f} [u_m(r)-u_0^m(r)] \frac{N}{k} \sum_{i=1}^{m} \sum_{r_{f,h}} -\sum_{r_{f,h}, i} \frac{\nabla \phi_i^0(r) \nabla \psi_i^0(r)}{2} r dr = 0 \quad (2.13)
$$
where

\[ \psi_i \quad = \text{adjoint flux for group } i \]
\[ k \quad = \text{k-effective} \]

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 \( \text{BG} \) (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. \[
   0 < u_f, 0 < u_m, u_m + u_f < N_0 = \text{const.} \tag{2.16}
   \]

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

$$L \phi^* = f(u_f^*, u_m^*)$$ \hspace{1cm} (2.17)

where $L$ is the multigroup diffusion matrix operator and

$$u_f^* = u_f^0 - u_f, \quad u_m^* = u_m^0 - u_m$$ \hspace{1cm} (2.18)

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

$$M \phi^* = f(u_f^*, u_m^*)$$ \hspace{1cm} (2.19)

Equations (2.19) can be solved by inversion of the matrix $M$. On the other hand even for 5 neutron groups and 100 mesh points $M$ 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
\begin{equation}
\phi_i^* \approx \phi_i^* = \sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_{k,i}
\end{equation}

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

\begin{equation}
\int_V (L_i \phi_i^*) w_k \, dV = \int_V f_i(u_f^*, u_m^*) w_k \, dV
\end{equation}

\begin{equation}
\int_V (L_i \phi_i^*) v_{k,i} \, dV = \int_V f_i(u_f^*, u_m^*) v_{k,i} \, dV
\end{equation}

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

\begin{equation}
A a = g(u_f^*, u_m^*, a_{11}),
\end{equation}

where \( a_{11} \) is the coefficient of the polynomial \( w_1 \) in Eq. (2.20) for \( i=1 \), and the components of the vectors \( u_f^*, u_m^* \) are given by
\[ u_{f,j}^* = u_{f,j} - u_{f,j}^0, \quad u_{m,j}^* = u_{m,j} - u_{m,j}^0, \quad j=1, R \]  \hspace{1cm} (2.24)

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

\[ a = A^{-1} \hat{g} \]  \hspace{1cm} (2.25)

For \( n \) mesh intervals and \( N \) neutron groups the order of the matrix \( A \) is equal to \( 2nN-1 \). The method of piecewise polynomials, compared to the finite difference technique, gives a very good approximation to \( \phi_1^* \) with only a few mesh intervals, \( n \). Since the order of matrix \( A \) is a function of the number of mesh intervals, \( n \), the method of piecewise polynomials gives a smaller matrix \( A \) than the finite difference technique for the same accuracy in \( \phi_1^* \). Thus for \( N = 5 \) and \( n = 10 \) the order of \( A \) is \( 2 \times 10 \times 5 - 1 = 99 \). For the same accuracy in \( \phi_1^* \) 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 \( u_{f,j}^0 \) and \( u_{m,j}^0 \) 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

\[
0 < u_{f,j} - \varepsilon_f < u_{f,j} + \varepsilon_f, \quad 0 < u_{m,j} - \varepsilon_m < u_{m,j} + \varepsilon_m,
\]

\((j = 1, \ldots, R)\) \hspace{1cm} (2.26)

The parameters \( \varepsilon_f, \varepsilon_m \) are constants such that \( u_{f,j} \) and \( u_{m,j} \) remain close enough to \( u_{f,j}^0 \) and \( u_{m,j}^0 \) 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 \( u_{f,j}^0 \) and \( u_{m,j}^0 \). To advance the solution the following iterative scheme is devised. If \( u_{f,j}^{(1)} \) and \( u_{m,j}^{(1)} \) is the solution given by Linear Programming, the problem is re-linearized about \( u_{f,j}^{(1)}, u_{m,j}^{(1)} \) and Linear Programming is again applied, while the relations

\[
0 < u_{f,j} - \varepsilon_f < u_{f,j} + \varepsilon_f, \quad 0 < u_{m,j} - \varepsilon_m < u_{m,j} + \varepsilon_m,
\]

\((j = 1, \ldots, R)\) \hspace{1cm} (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 $u_x$ and $u_m$; 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

<table>
<thead>
<tr>
<th>Region</th>
<th>Inner Radius</th>
<th>Outer Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>1</td>
<td>0.00 cm</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>62.64 cm</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>90.48 cm</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>111.36 cm</td>
</tr>
<tr>
<td>Radial Blanket</td>
<td>5</td>
<td>128.76 cm</td>
</tr>
</tbody>
</table>

*Extrapolated outer boundary
### TABLE 3.2

Reactor Composition

<table>
<thead>
<tr>
<th>Material</th>
<th>Core (v/o)</th>
<th>Blanket (v/o)</th>
<th>Atomic or Molecular density (for pure materials) $\text{cm}^{-3} \times 10^{-24}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>50 v/o</td>
<td>50 v/o</td>
<td>0.025410</td>
</tr>
<tr>
<td>Fe</td>
<td>15 v/o</td>
<td>15 v/o</td>
<td>0.084870</td>
</tr>
<tr>
<td>PuO$_2$</td>
<td>$&gt;35$ v/o</td>
<td>--</td>
<td>0.025189</td>
</tr>
<tr>
<td>UO$_2$</td>
<td>$&gt;35$ v/o</td>
<td>35 v/o</td>
<td>0.024444</td>
</tr>
</tbody>
</table>
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:

\[
p = \frac{P \times 2\pi H \times W}{W} \times \frac{100}{W} \times \frac{w}{w} = 2.30267 \frac{W}{cm^2}
\]

where

\[P = \text{power density upper limit} = 550 \, \text{w/cm}^3\]

\[H = \text{reactor height} = 100 \, \text{cm}\]

\[W = \text{normalized total power} = 100 \, \text{w}\]

\[W = \text{total thermal power} = 1500 \times 10^6 \, \text{w}\]

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 \(A\) (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.

\[3.2 \text{ BREEDING OPTIMIZATION}\]

The purpose of this section is to present the results obtained for the Breeding Optimization Problem. In Table 3.4, the results
<table>
<thead>
<tr>
<th>Group</th>
<th>Neutron Energy in Mev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.400 - $\infty$</td>
</tr>
<tr>
<td>2</td>
<td>0.400-1.400</td>
</tr>
<tr>
<td>3</td>
<td>0.100-0.400</td>
</tr>
<tr>
<td>4</td>
<td>0.017-0.100</td>
</tr>
<tr>
<td>5</td>
<td>0.000-0.017</td>
</tr>
</tbody>
</table>
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 multi-group diffusion equations are solved to express $\phi_i^*$ as a function of $u_f$ and $u_m$; 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

$$BG = \frac{2\pi \int_0^{t_f N} \sum_{i=1}^{t_f} [(N_0 - u_f) \Sigma_{r,i} + \Sigma_{s,a,i} u_f] \phi_i \, rdr}{2\pi \int_0^{t_f N} \sum_{i=1}^{t_f} \Sigma_{f,i} \phi_i \, rdr}$$

(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>
<thead>
<tr>
<th>Iteration Number</th>
<th>PuO$_2$</th>
<th>v/o</th>
<th>Breeding Gain*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.41200</td>
<td>3.41200</td>
<td>3.41200</td>
</tr>
<tr>
<td>2</td>
<td>3.40670</td>
<td>3.53833</td>
<td>3.21200</td>
</tr>
<tr>
<td>3</td>
<td>3.38110</td>
<td>3.69036</td>
<td>3.01200</td>
</tr>
<tr>
<td>4</td>
<td>3.35800</td>
<td>3.82934</td>
<td>2.81200</td>
</tr>
<tr>
<td>5</td>
<td>3.33607</td>
<td>3.95874</td>
<td>2.61200</td>
</tr>
<tr>
<td>6</td>
<td>3.31556</td>
<td>4.07905</td>
<td>2.41200</td>
</tr>
<tr>
<td>7</td>
<td>3.29832</td>
<td>4.17795</td>
<td>2.24362</td>
</tr>
<tr>
<td>8</td>
<td>3.29680</td>
<td>4.16995</td>
<td>2.32654</td>
</tr>
<tr>
<td>9</td>
<td>3.29543</td>
<td>4.16177</td>
<td>2.40826</td>
</tr>
<tr>
<td>10</td>
<td>3.29407</td>
<td>4.15375</td>
<td>2.48842</td>
</tr>
<tr>
<td>11</td>
<td>3.29277</td>
<td>4.14585</td>
<td>2.56699</td>
</tr>
<tr>
<td>12</td>
<td>3.29146</td>
<td>4.13812</td>
<td>2.64417</td>
</tr>
<tr>
<td>13</td>
<td>3.29017</td>
<td>4.13053</td>
<td>2.71992</td>
</tr>
<tr>
<td>14</td>
<td>3.28885</td>
<td>4.12313</td>
<td>2.79443</td>
</tr>
<tr>
<td>15</td>
<td>3.28765</td>
<td>4.11576</td>
<td>2.86731</td>
</tr>
<tr>
<td>16</td>
<td>3.28642</td>
<td>4.10857</td>
<td>2.93906</td>
</tr>
<tr>
<td>17</td>
<td>3.28521</td>
<td>4.10151</td>
<td>3.00954</td>
</tr>
<tr>
<td>18</td>
<td>3.28402</td>
<td>4.09457</td>
<td>3.07881</td>
</tr>
<tr>
<td>19</td>
<td>3.27854</td>
<td>4.09062</td>
<td>3.03854</td>
</tr>
<tr>
<td>20</td>
<td>3.27801</td>
<td>4.08658</td>
<td>3.07689</td>
</tr>
<tr>
<td>21</td>
<td>3.27801</td>
<td>4.08662</td>
<td>3.07676</td>
</tr>
</tbody>
</table>

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

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

*Net production of Pu<sup>239</sup> 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

<table>
<thead>
<tr>
<th>Region</th>
<th>Inner Radius</th>
<th>Outer Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.00 cm</td>
<td>55.68 cm</td>
</tr>
<tr>
<td>2</td>
<td>55.68 cm</td>
<td>80.04 cm</td>
</tr>
<tr>
<td>3</td>
<td>80.04 cm</td>
<td>97.44 cm</td>
</tr>
<tr>
<td>4</td>
<td>97.44 cm</td>
<td>111.36 cm</td>
</tr>
<tr>
<td>Radial Blanket</td>
<td>111.36 cm</td>
<td>156.60 cm *</td>
</tr>
</tbody>
</table>

TABLE 3.8
Optimum Configuration of Reactor No. 2

<table>
<thead>
<tr>
<th>Region</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuO₂ v/o</td>
<td>3.23751</td>
<td>3.72338</td>
<td>5.01528</td>
<td>0.50175</td>
</tr>
<tr>
<td>Peak Power Density</td>
<td>2.30267</td>
<td>2.30267</td>
<td>2.30267</td>
<td>0.29742</td>
</tr>
</tbody>
</table>

*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

\[ M_c = \frac{A \times M_{Pu}}{N_A} \int_0^{t_f} 2\pi r u_f(r)dr \]  

(3.3)

where

- \( A \) = atom density of Pu in PuO₂
- \( M_{Pu} \) = atomic weight of Pu
- \( N_A \) = Avogadro's number
TABLE 3.9

Effect of Blanket Reflector on Breeding Gain

<table>
<thead>
<tr>
<th>Reactor</th>
<th>Breeding Gain of Unreflected Reactor</th>
<th>Breeding Gain after addition of BeO Reflector*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>Homogeneous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. 1</td>
<td>0.405686</td>
<td>0.170841</td>
</tr>
<tr>
<td>Optimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. 1</td>
<td>0.345045</td>
<td>0.255540</td>
</tr>
<tr>
<td>Optimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. 2</td>
<td>0.377648</td>
<td>0.204880</td>
</tr>
</tbody>
</table>

* 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

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>Region</th>
<th>PuO$_2$ v/o</th>
<th>Critical Mass in kg x 10$^{-1}$ per cm core height</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>3.41200</td>
<td>3.41200</td>
<td>3.41200</td>
</tr>
<tr>
<td>2</td>
<td>3.40556</td>
<td>3.54010</td>
<td>3.21200</td>
</tr>
<tr>
<td>3</td>
<td>3.38058</td>
<td>3.69092</td>
<td>3.01200</td>
</tr>
<tr>
<td>4</td>
<td>3.35716</td>
<td>3.83033</td>
<td>2.81200</td>
</tr>
<tr>
<td>5</td>
<td>3.33521</td>
<td>3.95964</td>
<td>2.61200</td>
</tr>
<tr>
<td>6</td>
<td>3.31461</td>
<td>4.08004</td>
<td>2.41200</td>
</tr>
<tr>
<td>7</td>
<td>3.29748</td>
<td>4.17807</td>
<td>2.24549</td>
</tr>
<tr>
<td>8</td>
<td>3.29674</td>
<td>4.16940</td>
<td>2.32623</td>
</tr>
<tr>
<td>9</td>
<td>3.29536</td>
<td>4.16123</td>
<td>2.40797</td>
</tr>
<tr>
<td>10</td>
<td>3.29400</td>
<td>4.15322</td>
<td>2.48816</td>
</tr>
<tr>
<td>11</td>
<td>3.29266</td>
<td>4.14535</td>
<td>2.56682</td>
</tr>
<tr>
<td>12</td>
<td>3.29134</td>
<td>4.13762</td>
<td>2.64401</td>
</tr>
<tr>
<td>13</td>
<td>3.29005</td>
<td>4.13004</td>
<td>2.71979</td>
</tr>
<tr>
<td>14</td>
<td>3.28877</td>
<td>4.12259</td>
<td>2.79418</td>
</tr>
<tr>
<td>15</td>
<td>3.28751</td>
<td>4.11528</td>
<td>2.86724</td>
</tr>
<tr>
<td>16</td>
<td>3.28628</td>
<td>4.10809</td>
<td>2.93901</td>
</tr>
<tr>
<td>17</td>
<td>3.28506</td>
<td>4.10103</td>
<td>3.00951</td>
</tr>
<tr>
<td>18</td>
<td>3.28386</td>
<td>4.09409</td>
<td>3.07880</td>
</tr>
<tr>
<td>19</td>
<td>3.27152</td>
<td>4.09379</td>
<td>3.08245</td>
</tr>
<tr>
<td>20</td>
<td>3.27747</td>
<td>4.08592</td>
<td>3.07626</td>
</tr>
<tr>
<td>21</td>
<td>3.27746</td>
<td>4.08594</td>
<td>3.07623</td>
</tr>
</tbody>
</table>
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 \( u_f^0(r) \) 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

\[
\frac{1}{k_v} - \frac{1}{k_p} = \int_0^{t_f} \left( -u_f^* \sum_{i=1}^N \left( \Sigma_{tr,i}^{fs} - \Sigma_{tr,i}^{fr} \right) \nabla \phi_i \nabla \psi_i \right) r dr + \\
\int_0^{t_f} u_f \sum_{i=1}^N \left( \Sigma_{a,i}^{fs} - \Sigma_{a,i}^{fr} \right) \phi_i \psi_i r dr + \\
\int_0^{t_f} u_f^* \sum_{i=1}^N \sum_{h=i+1}^N \left\{ \left[ \Sigma_{i-h}^{fs} - \Sigma_{i-h}^{fr} \right] \phi_i (\psi_i - \psi_h) \right\} r dr + \\
- \frac{1}{k_v} \int_0^{t_f} u_f^* \sum_{i=1}^N \sum_{h=1}^N \left[ \nu_{fs}^{fs,h} - \nu_{fs}^{fr,h} - \nu_{fr}^{fr,h} - \nu_{fr}^{fs,h} \right] \chi_i \phi_i \psi_i r dr
\]

(3.4)

where

\( k_v \) = k-effective of voided reactor

\( k_p \) = k-effective of voided reactor after the fissile fuel
perturbation

and

\[
u_f^* = u_f - u_f^0 \quad (3.5)
\]
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

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>Region</th>
<th>PuO₂ v/o</th>
<th>k-effective of Sodium Voided Reactor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3.41200</td>
<td>1.06523</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.40556</td>
<td>1.06465</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3.38058</td>
<td>1.06401</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3.35716</td>
<td>1.06325</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3.33521</td>
<td>1.06241</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>3.31461</td>
<td>1.06151</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>3.29748</td>
<td>1.06064</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>3.29674</td>
<td>1.06045</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>3.29536</td>
<td>1.06027</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3.29400</td>
<td>1.06009</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>3.29266</td>
<td>1.05990</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>3.29134</td>
<td>1.05971</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>3.29005</td>
<td>1.05952</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>3.28877</td>
<td>1.05932</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>3.28751</td>
<td>1.05913</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>3.28628</td>
<td>1.05893</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>3.28506</td>
<td>1.05873</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>3.27500</td>
<td>1.05765</td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>3.27923</td>
<td>1.05764</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Region</th>
<th>Region</th>
<th>Inner Radius</th>
<th>Outer Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>1</td>
<td>0.00 cm</td>
<td>62.64 cm</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>62.64 cm</td>
<td>90.48 cm</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>90.48 cm</td>
<td>111.36 cm</td>
</tr>
<tr>
<td>Radial Blanket</td>
<td>4</td>
<td>111.36 cm</td>
<td>160.08 cm</td>
</tr>
<tr>
<td>Reflector</td>
<td>5</td>
<td>160.08 cm</td>
<td>206.48 cm</td>
</tr>
</tbody>
</table>

*Extrapolated outer boundary*
TABLE 4.2
Reactor Composition for BeO Moderated Blanket

<table>
<thead>
<tr>
<th>Material</th>
<th>Core Regions</th>
<th>Blanket</th>
<th>Reflector</th>
<th>Atomic or Molecular Density for Pure Materials cm(^{-3}) x10(^{-24})</th>
</tr>
</thead>
<tbody>
<tr>
<td>PuO(_2)</td>
<td>3.2775 v/o 4.0859 v/o 3.0763 v/o</td>
<td>-</td>
<td>-</td>
<td>0.025189</td>
</tr>
<tr>
<td>UO(_2)</td>
<td>31.7225 v/o 30.9141 v/o 31.9237 v/o</td>
<td>-</td>
<td>55 v/o</td>
<td>0.024444</td>
</tr>
<tr>
<td>BeO</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.071270</td>
</tr>
<tr>
<td>Na</td>
<td>50 v/o</td>
<td>50 v/o</td>
<td>50 v/o</td>
<td>30 v/o</td>
</tr>
<tr>
<td>Fe</td>
<td>15 v/o</td>
<td>15 v/o</td>
<td>15 v/o</td>
<td>15 v/o 100 v/o</td>
</tr>
<tr>
<td>Material</td>
<td>Core Regions</td>
<td>Blanket</td>
<td>Reflector</td>
<td>Density for Pure Materials ( \text{cm}^{-3} \times 10^{-24} )</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>---------</td>
<td>------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>( \text{PuO}_2 )</td>
<td>3.2775 v/o</td>
<td>4.0859 v/o</td>
<td>3.0763 v/o</td>
<td>-</td>
</tr>
<tr>
<td>( \text{UO}_2 )</td>
<td>31.7225 v/o</td>
<td>30.9141 v/o</td>
<td>31.9237 v/o</td>
<td>( &gt;85 \text{ v/o} )</td>
</tr>
<tr>
<td>Na</td>
<td>50 v/o</td>
<td>50 v/o</td>
<td>50 v/o</td>
<td>-</td>
</tr>
<tr>
<td>Fe</td>
<td>15 v/o</td>
<td>15 v/o</td>
<td>15 v/o</td>
<td>15 v/o</td>
</tr>
</tbody>
</table>
TABLE 4.4

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

<table>
<thead>
<tr>
<th>Case</th>
<th>Moderator v/o</th>
<th>(^{238}\text{U}) v/o</th>
<th>Breeding Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>75</td>
<td>0.340401</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>65</td>
<td>0.341137</td>
</tr>
<tr>
<td>3</td>
<td>30*</td>
<td>55</td>
<td>0.342077</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>45</td>
<td>0.343326</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>35</td>
<td>0.345091</td>
</tr>
</tbody>
</table>

BeO Moderator

<table>
<thead>
<tr>
<th>Case</th>
<th>Moderator v/o</th>
<th>(^{238}\text{U}) v/o</th>
<th>Breeding Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>55</td>
<td>0.342077</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>50</td>
<td>0.344532</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>45</td>
<td>0.347181</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>35</td>
<td>0.353354</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>25</td>
<td>0.361465</td>
</tr>
<tr>
<td>11</td>
<td>5**</td>
<td>50</td>
<td>0.344557</td>
</tr>
<tr>
<td>12</td>
<td>5***</td>
<td>50</td>
<td>0.343183</td>
</tr>
</tbody>
</table>

* The volume fractions of Na and \(\text{UO}_2\) of this row are representative of typical fast reactor blanket designs

** \(\sigma_{\text{BeO}}^{(n,2n)} = 0.0\)

*** \(\sigma_{\text{BeO}}^{\text{down-scattering}} = 0.0\)
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 \(^{238}\text{U}\) 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 \(^{238}\text{U}\) 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

<table>
<thead>
<tr>
<th>Blanket Thickness cm</th>
<th>Breeding Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>BeO Reflector</td>
<td></td>
</tr>
<tr>
<td>16.24</td>
<td>0.344334</td>
</tr>
<tr>
<td>32.48</td>
<td>0.342144</td>
</tr>
<tr>
<td>48.72</td>
<td>0.342076</td>
</tr>
<tr>
<td>Graphite Reflector</td>
<td></td>
</tr>
<tr>
<td>16.24</td>
<td>0.343837</td>
</tr>
<tr>
<td>32.48</td>
<td>0.342133</td>
</tr>
<tr>
<td>48.72</td>
<td>0.342076</td>
</tr>
<tr>
<td>Iron Reflector</td>
<td></td>
</tr>
<tr>
<td>16.24</td>
<td>0.343804</td>
</tr>
<tr>
<td>32.48</td>
<td>0.342196</td>
</tr>
<tr>
<td>48.72</td>
<td>0.342077</td>
</tr>
<tr>
<td>No Reflector</td>
<td></td>
</tr>
<tr>
<td>32.48</td>
<td>0.341873</td>
</tr>
<tr>
<td>48.72</td>
<td>0.342071</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Reflector</th>
<th>Breeding Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>No Reflector</td>
<td>0.341873</td>
</tr>
<tr>
<td>BeO with $\sigma_{\text{down-scatt}}$ = 0.0</td>
<td>0.342354</td>
</tr>
<tr>
<td>BeO with $\sigma_{n,2n}$ = 0.0</td>
<td>0.342146</td>
</tr>
<tr>
<td>BeO</td>
<td>0.342144</td>
</tr>
<tr>
<td>U(^{238}) cross sections</td>
<td>Breeding Gain</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>Unshielded</td>
<td>0.342144</td>
</tr>
<tr>
<td>Shielded</td>
<td>0.346069</td>
</tr>
</tbody>
</table>
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.
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 $^{238}\text{U}$ in the blanket the neutron moderation by Na is not enough to offset the loss in breeding due to reduction of the $^{238}\text{U}$ concentration and consequently the breeding gain decreases as the Na concentration increases; (b) when BeO replaces $^{238}\text{U}$ 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 $^{238}\text{U}$ concentration; for any other BeO concentration the neutron moderation is not enough to offset breeding losses due to reduction of the $^{238}\text{U}$ 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.
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

$$H_0(r) = \frac{\pi}{B_{0,z}^2(r)}$$  \hspace{1cm} (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
$$I = \int_{V} u_{p} dV$$  \hspace{1cm} (5.2)

where

$$u_{p} = \text{volume fraction of control poison.}$$

As discussed in Chapter 2 the solution of the linearized multi-group 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 multi-group 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.


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.


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


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.


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 $x$ with non-negative components which satisfies the relations

$$Ax \geq b,$$

(B.1)
and maximizes or minimizes the function

\[ I = c^T x, \]  \hspace{1cm} (B.2)

where the matrix \( A \), and the vectors \( b \) and \( c \) are all independent of \( x \).

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 \( u(t) \) which maximizes (minimizes) the functional

\[ I = \int_{t_i}^{t_f} L(x, u, t) dt + S[x(t_f), t_f], \]  \hspace{1cm} (B.3)

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

\[ \frac{dx}{dt} = f(x, u, t) \]  \hspace{1cm} (B.4)

The terminal point \( t_f \) may be fixed or free, the terminal state \( x(t_f) \) may be fixed, completely free, or specified by a set of equations of the form
\[ h[x(t_f), t_f] = 0 \]  

(B.5)

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

\[ \phi(t, x, u) \leq 0 \]  

(B.6)

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

\[
\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_j, u_1, u_2, \ldots, u_k, t) \tag{B.7}
\]

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 \( x^0 \), \( u^0 \).

This series expansion is given by the relation

\[
\frac{\partial f_i}{\partial x_1} (x_1 - x_1^0) + \ldots + \frac{\partial f_i}{\partial x_j} (x_j - x_j^0) + \frac{\partial f_i}{\partial u_k} (u_k - u_k^0) + \ldots + \frac{\partial^2 f_i}{\partial u_1^2} (u_1 - u_1^0) + \ldots + \frac{\partial^2 f_i}{\partial u_k^2} (u_k - u_k^0) + \text{higher-order terms}, \tag{B.8}
\]
where the derivatives are evaluated at

\[ x_0, \ldots, x_j, u_1, \ldots, u_k \]

If changes in \( x \) and \( u \) from the solution \( x^0, u^0 \) are designated as \( x^* \) and \( u^* \), defined by the relations

\[ x^* = x - x^0, \quad u^* = u - u^0, \quad (B.9) \]

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

\[
 f_i(x_1, \ldots, x_j, u_1, \ldots, u_k, t) = f_i(x_1^0, \ldots, x_j^0, u_1^0, \ldots, u_k^0, t) + \]

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

higher-order terms. \quad (B.10)

Since

\[
 \frac{dx_i}{dt} = f_i(x_1, \ldots, x_j, u_1, \ldots, u_k, t),
\]

and

\[
 \frac{dx_i^0}{dt} = f_i(x_1^0, \ldots, x_j^0, u_1^0, \ldots, u_k^0, t),
\]

with \( x^* \) and \( u^* \) sufficiently close to \( x^0 \) and \( u^0 \), a first-order approximation to Eq. (B.7) is given by the relation
\[
\frac{dx_i^*}{dt} = \frac{\partial f_i^*}{\partial x_1} x_1^* + \ldots + \frac{\partial f_i^*}{\partial x_j} x_j^* + \frac{\partial f_i^*}{\partial u_{l1}} u_{l1}^* + \ldots + \frac{\partial f_i^*}{\partial u_{k1}} u_{k1}^* \quad (B.11)
\]

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

$$\phi_i \approx \phi_i = \sum_{k=1}^{n} a_{k,i} w_k + \sum_{k=1}^{n} \beta_{k,i} v_k, (C.1)$$

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

$$w_k = \begin{cases} \frac{(r-r_{k-1})^2}{3(h_-)} - \frac{(r-r_{k-1})^3}{2(h_-)}, & r \in [r_{k-1}, r_k] \\ \frac{(r+1-r)_{k+1}^2}{3(h_+)} - \frac{(r+1-r)_{k+1}^3}{(h_+)^2}, & r \in [r_k, r_{k+1}] \\ 0 \text{ otherwise} \end{cases}$$

$$v_k = \begin{cases} \frac{(r-r_{k-1})^2}{3(h_-)} - \frac{(r-r_{k-1})^3}{2(h_-)}, & r \in [r_{k-1}, r_k] \\ \frac{(r+1-r)_{k+1}^2}{3(h_+)} - \frac{(r+1-r)_{k+1}^3}{(h_+)^2}, & r \in [r_k, r_{k+1}] \\ 0 \text{ otherwise} \end{cases}$$

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

\[ \frac{d w_k}{dr} = 0 \text{ at } k-1, k, k+1 \]

\[ \frac{d v_{k,i}}{dr} = 0 \text{ at } k-1, k+1 \] (C.4)

\[ D_{i-} \frac{d v_{k,i}}{dr} = D_{i+} \frac{d v_{k,i}}{dr} \text{ at } k \]
FIG. C.1 THE CUBIC PIECEWISE POLYNOMIALS $w_k$ AND $v_{k,i}$
\[ w_k = 0, \quad v_{k,i} = 0 \text{ at } k-1, k+1 \]

\[ w_k = 1, \quad v_{k,i} = 0 \text{ at } k \]

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

\[ \frac{d\phi_i}{dr} \bigg|_0 = 0, \quad \phi_i(t_f) = 0, \quad (C.5) \]

we define

\[ \beta_{i,i} = 0, \quad a_{n,i} = 0 \quad (C.6) \]

The multigroup diffusion equations can be written in the form

\[ L_i \phi_i = 0 \quad (C.7) \]

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

\[ \int_V (L_i \phi_i) w_k \, dV = 0, \quad (C.8) \]

\[ \int_V (L_i \phi_i) v_{k,i} \, dV = 0 \quad (C.9) \]
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)

\[
|\phi_i - \phi_i^*| \leq k(h)^4,
\]

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

\[
\int_V (L_i \phi_i) w_k dV = \int_V f_i (u_m^*, u_m^*) w_k dV,
\]

(C.11)
and

\[
\int_{V} (L \phi_{i}) v_{k,i} \, dV = \int_{V} f_{i} (u_{f}^{*}, u_{m}^{*}) v_{k,i} \, dV \quad \text{(C.12)}
\]

or

\[
\int_{V} \left[ \sum_{k=1}^{n} a_{k,i} w_{k} + \sum_{k=1}^{n} \beta_{k,i} v_{k,i} \right] \, w_{k} \, dV =
\int_{V} f_{i} (u_{f}^{*}, u_{m}^{*}) \, w_{k} \, dV \quad \text{(C.13)}
\]

and

\[
\int_{V} \left[ \sum_{k=1}^{n} a_{k,i} w_{k} + \sum_{k=1}^{n} \beta_{k,i} v_{k,i} \right] \, v_{k,i} \, dV =
\int_{V} f_{i} (u_{f}^{*}, u_{m}^{*}) \, v_{k,i} \, dV \quad \text{(C.14)}
\]

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):
\[
\int w_k w_k \, rdr = \frac{2}{7} (h_-^2 - h_+^2) + \frac{13}{35} (h_r \cdot k-1 + h_r \cdot k+1)
\]

\[
\int w_k w_{k+1} \, rdr = \frac{9}{140} h_+^2 + \frac{9}{70} h_r \cdot i
\]

\[
\int w_k w_{k-1} \, rdr = -\frac{9}{140} h_-^2 + \frac{9}{70} h_r \cdot i
\]

\[
\int v_k, v_k \, rdr = \frac{1}{105} \left[ r_{k+1} \frac{h_+^3}{D_{1+} D_{j+}} + r_{k-1} \frac{h_-^3}{D_{1-} D_{j-}} \right] + \frac{1}{168} \left[ \frac{h_-^4}{D_{1-} D_{j-}} - \frac{h_+^4}{D_{1+} D_{j+}} \right]
\]

\[
\int v_k, v_{k+1} \, rdr = -\frac{1}{140} r_k \frac{h_+^3}{D_{1+} D_{j+}} - \frac{1}{280} \frac{h_+^4}{D_{1+} D_{j+}}
\]

\[
\int v_k, v_{k-1} \, rdr = -\frac{1}{140} r_k \frac{h_-^3}{D_{1-} D_{j-}} + \frac{1}{280} \frac{h_-^4}{D_{1-} D_{j-}}
\]

\[
\int w_k v_k, \, rdr = -\frac{11}{210} (r_{k-1} \frac{h_-^2}{D_{1-}} - r_{k+1} \frac{h_+^2}{D_{1+}}) - \frac{1}{28} (\frac{h_-^3}{D_{1-}} + \frac{h_+^3}{D_{1+}})
\]
\[
\int \frac{w_{k,v} v_{k+1,i} \, r \, d\rho}{v} = -\frac{13}{420} \frac{r_k h_+^2}{D_{1+}} - \frac{h_+^3}{70 D_{1+}}
\]

\[
\int \frac{w_{k,v} v_{k-1,i} \, r \, d\rho}{v} = \frac{13}{420} \frac{r_k h_-^2}{D_{1-}} - \frac{1}{70} \frac{h_-^3}{D_{1-}}
\]

\[
\int \frac{d w_k}{d \rho} \times \frac{d w_k}{d \rho} \, r \, d\rho = \frac{6}{5} \frac{r_{k-1}}{h_-} + 0.6 + \frac{6}{5} \frac{r_{k+1}}{h_+} - 0.6^*
\]

\[
\int \frac{d w_k}{d \rho} \times \frac{d w_{k+1}}{d \rho} \, r \, d\rho = -\frac{6r_k}{5h_+} - 0.6
\]

\[
\int \frac{d w_k}{d \rho} \times \frac{d w_{k-1}}{d \rho} \, r \, d\rho = -\frac{6r_k}{5h_-} + 0.6
\]

\[
\int \frac{d v_{k,i}}{d \rho} \times \frac{d v_{k,i}}{d \rho} \, r \, d\rho = \frac{2}{15} \left( -\frac{r_{k-1} h_-^2}{D_{1-}^2} + \frac{r_{k+1} h_+^2}{D_{1+}^2} \right) + \frac{1}{10} \left( \frac{h_-^2}{D_{1-}^2} - \frac{h_+^2}{D_{1+}^2} \right)
\]

*The first two terms come from integration to the left of point \( k \) and the last two from integration to the right of point \( k \).*
The solution of the linearized multigroup diffusion equations (Section 2.3) gives the coefficients $a_{k,i}$ and $b_{k,i}$ 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:
\[ w_k \, r \, d r = \frac{7h^2}{20} + 0.5 r_{k-1} h_- \]
\[ w_{k+1} \, r \, d r = -\frac{7h^2}{20} + 0.5 r_{k+1} h_+ \]
\[ v_{k,i} \, r \, d r = -\frac{h^2}{12D_{i-}} - \frac{h^3}{20D_{i-}} \]
\[ v_{k+1,i} \, r \, d r = \frac{r_{k+1} h^2}{12D_{i+}} - \frac{h^3}{20D_{i+}} \]

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 \(2 \times l\) where \(l\) 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

\[
4 \times \text{NRG} \times \text{NGP} - 1
\]

where:

- \(\text{NRG} = \) number of regions
- \(\text{NGP} = \) 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 \(\epsilon\) (Eqs. 2.26, 2.27). The value of the parameter \(\epsilon\) is chosen such that the \(u_{f,j}^\circ\), \((j = 1,R)\) remain close enough to \(u_{f,j}^0\) (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 \(\epsilon\) 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} \times 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 \times 10^{-4})
<table>
<thead>
<tr>
<th>Card</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS2</td>
<td>Convergence criterion on eigenvalue in outer iteration (recommended value $1.0 \times 10^{-5}$)</td>
<td></td>
</tr>
<tr>
<td>EPS3</td>
<td>Convergence criterion for flux (recommended value $1.0 \times 10^{-8}$)</td>
<td></td>
</tr>
<tr>
<td>ITMAX0</td>
<td>Maximum number of outer iterations (typical value 10)</td>
<td></td>
</tr>
<tr>
<td>ITMAXI</td>
<td>Maximum number of inner iterations (typical value 20)</td>
<td></td>
</tr>
</tbody>
</table>

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

Card #7 FORMAT (16I5)

<table>
<thead>
<tr>
<th>MMM</th>
<th>Material ID number</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>= 0, non-fissionable material</td>
</tr>
<tr>
<td></td>
<td>= 1, fissionable material</td>
</tr>
</tbody>
</table>

Card #8 FORMAT (7G10.0)

<table>
<thead>
<tr>
<th>SIGC(JJ,J), J=1, NGP</th>
<th>Total microscopic absorption cross section of material JJ in group J (capture + fission), barns</th>
</tr>
</thead>
</table>

Card #9 FORMAT (7G10.0)

<table>
<thead>
<tr>
<th>SIGTR(JJ,J), J=1, NGP</th>
<th>Microscopic transport cross section of material JJ in group J, barns</th>
</tr>
</thead>
</table>

Card #10 FORMAT (7G10.0)

<table>
<thead>
<tr>
<th>XNU(JJ,J), J=1, NGP</th>
<th>Fission neutron yield, $\nu$, of material JJ in group J</th>
</tr>
</thead>
</table>
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  Microscopic scattering cross section
                            K=1, NGP  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:
        = 1, Breeding Optimization
        = 2, Sodium Void Reactivity Optimization
        = 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

```
** ** ** ** ** ** ** ** ** ** ** **
PROGRAM GPFFKD
** ** ** ** ** ** ** ** ** ** ** **
MAIN PROGRAM
IMPLICIT REAL*8(A-H,0-Z)
COMMON/POWER/SIGEM(1*,5),AKTIS(201),TOPP(10),SYL1(10),FISIT(10),
1FSDT(1*),TMETLM(10),SYL2M(10),FISITM(10),TMETLM(10),ALKGEM(10),
2GPH2(10,5),GPHN(1*,5),GPHA2(10,5),ALKGEN(10),GPH1(10,5),
3ST,SPE1(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
4SGG(10,5,5),DI(10,5)
COMMON /FLUX/ PHI(201,5),ANORM,RNORM,A(201,5),B(201,5),
1C(201,5),W(201,5,5),S(201,5)
COMMON /CTRL/ EPS1,EPS2,EPS3,EFFK,T(1*),RK1,RK2,BIG,AHOLD(90),
1NGP,NGS,NGMAT,NGEM,JACL,JACR,NED,JAC,JNP,NPT(10),IOP,NRVAR,
2IRVAR(90),MVAX,ITMAXO,ITMAX1,ITM,ENE,KEEPM,CODE,LBIG,JBIG,IAJ,
3JOUN,KTHL(90)
COMMON /MACX/ XACT(5),X(11,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
1CRHCT(1*,1*),DT(10,5),XR(10,5),CC,CT,IDMAT(1*)
COMMON /ERR/IERR
COMMON/CONV/CMA(30),NPR,KNA,NCR,ION
COMMON/ITER//NIT
DIMENSION CMA(30),NPR,KNA,NCR,ION
APS(77)=DAMS(77)
KNA=0
NIT=150
IERR=0
RX1=0.0
TK=.0
BIG=1.0E3
IAJ=0
EEFK=1.0
KEEP=0
CC=1.0
ITP=1
```
ITI=0
IF(NIT.NE.0) GO TO 1510
IF(KNA.EQ.1) GO TO 1510
CALL INDATA
1510 IF(IERR.NE.0) GO TO 9999
CALL MACROX
CALL FLUXIN
IF(IERR.NE.0) GO TO 9999
1 CALL XSECT
7 CALL TRIGTA
5 CALL WEIGHT
2 CALL SOLVE
CALL RESCAL
ITI=ITI+1
IF(ITI.GT.ITMAX) CALL ERROR(1)
IF(IERR.NE.0) GO TO 9999
IF(ABS(PK2-TK).LT.EPS1) GO TO 3
TK=PK2
GO TO 2
3 IF(ABS(RK2-1.0).LT.EPS2) GO TO 6
CALL ADJUST
IHOLD(I) = ITI
AHOLD(I) = K
RK1=KK
IT0=IT0+1
IF(IT0.GT.ITMAX) CALL ERROR(6)
IF(IERR.NE.0) GO TO 9999
ITI=0
GO TO (5,7,1),100
6 IF(BIG.LT.EPS3) GO TO 100
BIG=C0
KEEP=1
GO TO 2
100 IF(JAD.EQ.0) GO TO 9999
DO 10 J=1,NP
DO 10 I=1,NGP
\[ \Phi(J', I) = \Phi(J, I) \]

10 CONTINUE
CALL ADJOIN
CALL ISCHIS
IF(NPR.NE.2) GO TO 25
IF(KNA.EQ.1) GO TO 1531
IF(NIT.NE.3) GO TO 1530
DO 20 K=1,NCR
20 CONC(K)=CONC(IDNA,K)
25 IF(NIT.NE.3) GO TO 1530
CALL EDIT
CALL AEDIT
1530 CALL BASE
CALL RGMAT
1531 CALL LINPRO
IF(NPR.NE.2) GO TO 1541
IF(KNA.EQ.1) GO TO 1545
DO 15 K=1,NCR
15 CONC(IDNA,K)=CONC(K)
GO TO 1541
1545 DO 21 K=1,NCR
21 CONC(IDNA,K)=CONC(K)
1540 GO TO 1500
9999 CALL EXIT
END
SUBROUTINE INDATA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /FLUX/ PHIC(2:1,5),ANDIM, BNORM, A(2:1,5), R(2:1,5),
1 C(2:1,5), M(2:1,5), S(2:1,5)
COMMON /CTRL/ EPS1, EPS2, EPS3, EFFK, TH(1:5), R1K, R2K, RIG, AHULD(90),
1 NGP, NRG, NMAT, NGEOM, JBCCL, JBCR, NFRG, JAC, NP, NPT(1:2), IOP, NRVARY,
2 R2VARY(20), MVARY, ITMAXJ, ITMAXI, IT1, IT2, KEP, ICODE, LBIC, JBIG, IAJ,
3 JDOM, JHOLD(90)
COMMON /MACX/ SPECT(5), XA(1:5), XNlFF(1:5), XTR(1:5), XGG(1:5,5),
1 CONC(1:1,1), 0(1:5), XR(1:5), CC, CT, IDMAT(1:5)
COMMON /MICX/ SIGC(1:5), SIGTR(1:5), XNU(1:5), SIGF(1:5),
1 SIGGG(1:5,5)
COMMON /ERR/ IFRR
REAL TITLE
DIMENSION TITLE(20)
NGEOM=2
JBCCL=1
JBCR=5
IOP=1
READ (5,9901) (TITLE(J),J=1,20)
WRITE (5,9901) (TITLE(J),J=1,20)
READ (5,9902) NGP, NRG, NMAT
WRITE (6,9931) 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 (6,999)
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,IMAT(J),J=1,NMAT)
READ (5,9912) EPS1,EPS2,EPS3,ITMAX3,ITMAX1
DO 3 I=1,NMAT
READ (5,992) MM,MT,M2
DO 4 J=1,NMAT
JJ=J
IF (MM.EQ.IDMAT(J)) GO TO 5
4 CONTINUE
CALL ERROR (2)
IF (IERB.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)=.
SIGF(JJ,J)=.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
WRITE (6,997) (SIGC(I,J),J=1,NGP)
WRITE (6,998) (SIGTR(I,J),J=1,NGP)
WRITE (6,999) (SIGF(I,J),J=1,NGP)
WRITE (6,9911) (XNU(I,J),J=1,NGP)
WRITE (6,9918) (SIGTR(I,J),J=1,NGP)
DO 10 K=1,NGP
WRITE (6,9919) K,(SIGGG(I,K,J),J=1,NGP)
10 CONTINUE
991 FORMAT (29X,20A4)
992 FORMAT (16I5)
993 FORMAT (///' NUMBER OF ENERGY GROUPS =',11C,///' NUMBER OF REGIONS
1=',11C,///' NUMBER OF MATERIALS =',11C,///)
994 FORMAT (///' PROBLEM GEOMETRY = SLAB')
995 FORMAT (///' PROBLEM GEOMETRY = CYLINDER')
996 FORMAT (///' PROBLEM GEOMETRY = SPHERE')
997 FORMAT (791:1)
998 FORMAT (///1X,'GROUP',1LX,'LOWER ENERGY BOUND',/10X,'-----',10X,
1'------------------1',//(1C,13,1G0X,015,5))
999 FORMAT(15,E3,4E15.1,/'E15.2,,'2F15.6)
990 FORMAT (///1X,'MATERIAL',4X,'REGION / CONCENTRATION',//)
991 FORMAT (2A4)
992 FORMAT(/1X,12,8(I2,,'F13.10'),/10X,8(I4,,'F15.10'))
993 FORMAT (/// REGION / NUMBER OF MESH POINTS',/10(I5,,'F13.10'))
994 FORMAT (/// FISSION SPECTRUM',/(9F15.10))
995 FORMAT (1'CROSS SECTIONS FOR MATERIAL',110,///)
996 FORMAT (' CAPTURE CROSS SECTION',/(8F15.10))
997 FORMAT (' FISSION CROSS SECTION',/(8F15.10))
998 FORMAT (' TRANSPORT CROSS SECTION',/(8F15.10))
999 FORMAT (' TRANSFER CROSS SECTION FROM GROUP',15,/(8F15.10))
991 FORMAT (' NU',/(8F15.6))
9912 FORMAT (3F15.0,15) 
9913 FORMAT(/// REGION / REGION THICKNESS IN CM',/6(5X,12,,'F8.4))
RETURN
END
SUBROUTINE MACROX
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/POWER/ SIGFM(15,5),AKTIS(21),TUTP(10),SYLI(10),FISIT(10),
IFDT(1),TMEFLM(10),SYLM(10),FISITM(10),TMEFLM(10),ALKGEM(10),
2GRPH2(10,5),GRPH2(10,5),GRPH2(10,5),GRPH2(10,5),GRPH2(10,5),
3STS,PHL(2,15),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
4SIGG(10,5,5),DI(10,5)
COMMON /CNTRL/ EPS1,EPS2,EPS3,FFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NGR,NMAT,NGEM,JBCF,JBCF,NFG,JAC,NP,NPT(1),IOP,IRVARY,
2IRVARY(90),MVAR,ITYMAX,ITMAXI,IT0,ITR,KEEP,MCODE,LEIG,JBIG,IAJ,
3JNOM, IHOLD(90)
COMMON /MACX/ SPECT(5),XAI(10,5),XNUF(10,5),TXR(10,5),XGG(10,5,5),
1CONC(10,5),D(10,5),XK(10,5),CC,CT,IMAT(10)
COMMON /MICX/ SIGC(10,5),SIGTR(10,5),XNUF(10,5),SIGF(10,5),
1SIGGG(10,5,5)
COMMON /FLUX/ PHI(201,5),ANKKM, BNORM,A(201,5),B(201,5),
1C(201,5),W(201,5,5),S(201,5)
DO 4 I=1,NGR
DO 4 J=1,NGP
SP(I,J)=0.0
XAI(I,J)=0.0
SIGFM(I,J)=0.0
XNUF(I,J)=0.0
TXR(I,J)=0.0
DO 4 K=1,NGP
XGG(I,J,K)=0.0
CONTINU
DO 3 J=1,NMAT
DO 3 I=1,NGP
DO 3 K=1,NGR
XAI(K,I)=XAI(K,I)+CONC(J,K)*SIGC(J,I)
SIGFM(K,I)=SIGFM(K,I)+CONC(J,K)*SIGC(J,I)
XNUF(K,I)=XNUF(K,I)+CONC(J,K)*SIGF(J,I)*XNUF(J,I)
TXR(K,I)=TXR(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
DO 2 I=1, NRG
DO 2 J=1, NGP
2 XGG(I,J,J)=0
DO 5 K=1, NRG
DO 5 I=1, NGP
SA(K,I)=XA(K,I)
SNUF(K,I)=XNUF(K,I)
STR(K,I)=XTR(K,I)
DI(K,I)=1.0/(3.0*XSTR(K,I))
DO 5 L=1, NGP
SGG(K,I,L)=XGG(K,I,L)
SK(K,I)=SK(K,I)+XGG(K,I,L)
5 CONTINUE
RETURN
END
SUBROUTINE FLUXIN

IMPLICIT REAL*8 (A-H,O-Z)

COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),NCGP,NKG,NGMAT,NGEOM,JACCL,JACF,NFG,JAC,JP,NPT(1T),IOP,NRVARY,
1 TNVAR(97),MVARY,ITMAX,ITMAXI,ITD,ITI,KEEP,MCODE,LIBG,JBIG,IAJ,
2 IQUAM,IPHIND(90)

COMMON /MACX/ SPEC1(5),X(1:5),XNUL(1:5),XTR(1:5),XGG(1:5,5)

COMMON /FLUX/ PHI(201,5),ANORM,BNORM,A(201,5),B(201,5)

COMMON /IERR/IERR

SQR(07)=DSQR(77)

NP=1

DO 1 J=1,NRC

NP=NP+NPT(J)

1 DO 2 L=1,NGP

DO 2 J=1,NGP

2 PHI(J,L)=1.0

3 ANORM=SQR(1,J,AC)*NP*NGP

GO TO (6,6,5),IOP

5 DO 3 J=1,NMAT

JJ=J

IF (IDMAT(J).EQ.MVAPY) GO TO 4

3 CONTINUE

CALL ERROR(4)

IF (IERR.NE.0) RETURN

4 MCODE=JJ

6 CONTINUE

RETURN

END
SUBROUTINE XSEC
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTPL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NGR,NMAT,NGEM,JBCJ,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVAR/
2Y(90),MVARY,ITMAX,ITMAXI,ITR,KEEP,MCODE,LBIG,JBIG,IAJ,
3JNUX,HCLD(90)
COMMON /MACX/ SPECT(5),XA(15,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
1CONC(15,10),O(15,5),XR(10,5),CC,CT,DMAT(10)
COMMON /MICX/ SIGC(10,5),SIGG(10,5,5),XNU(10,5),SIGF(10,5),
1SICGG(10,5,5)
COMMON /FLUX/ PHI(201,5),ANORM,BNORM,A(201,5),B(201,5),
1C(201,5),W(201,5,5),S(201,5)
DIMENSION F(10)

DO 1 K=1,NKG
Z=(CC-1.0)/CC
1 CONTINUE
DO 1 K=1,NKG
DO 1 I=1,NGP
XA(K,I)=XA(K,I)+F(K)*SIGC(MCODE,I)
1 CONTINUE
DO 1 K=1,NKG
DO 1 I=1,NGP
XNUF(K,I)=XNUF(K,I)+F(K)*SIGF(MCODE,I)*XNU(MCODE,I)
1 CONTINUE
DO 1 K=1,NKG
DO 1 I=1,NGP
XTR(K,I)=XTR(K,I)+F(K)*SIGTR(MCODE,I)
1 CONTINUE
DO 1 K=1,NKG
DO 1 I=1,NGP
XR(I,J)=XR(I,J)+XGG(I,J,K)
1 CONTINUE
RETURN
SUBROUTINE TRIDIA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
NGP,NRG,NMAT,NGEDM,JBCL,JBCR,NFG,JDAD,NP (10),IDP,NVARV,
IRVARV(90),MVARV,ITMAXD,ITMAX!,IT?,ITI,KEEP, MCD,EB, LBIG,JBIG,IAJ,
3JDUM, IHOLD(90)
COMMON /MAX/ SPECT(5),XA(10,5),XNUF (10,5),XTR (10,5),XGG (10,5,5),
1CONC(10,10),O(10,5),XR(10,5),CJ,CT, IDMAT(10)
COMMON /FLUX/ PHI(201,5),ANORM, BNORM, A(201,5), R(201,5),
1C(201,5), W(201,5,5) , S(201,5)
DO 3 L=1,NGP
DO 3 J=1,NGP
          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 (NGEDM,EQ.1) RP=1.0
IF (NGEDM,EQ.2) RP=R
IF (NGEDM,EQ.3) RP=P*R
DO 2 L=1,NGP
A(JJ,L)=RP*K(K,L)*HI
C(JJ,L)=A(JJ,L)
Z=A(JJ,L)+((R-H/2.0)**(NGEDM-1))*XR(K,L)*H*0.5
Z1=A(JJ,L)+((R+H/2.0)**(NGEDM-1))*XR(K,L)*H*0.5
R(JJ,L)=B(JJ,L)-Z
R(JJ+1,L)=B(JJ+1,L)-Z1
2 CONTINUE
RWR=RWR+TH(K)
1 CONTINUE
IF (JBCLE.Q.EQ.,C) GO TO 4
DO 5 L=1,NGP
   A(1,L)=2.0*A(1,L)
   B(1,L)=-A(1,L)
GO TO 6
5 DO 7 L=1,NGP
   B(1,L)=1.0
7 A(1,L)=.
9 IF (JBCP.EQ.,0) GO TO 9
   PP = RWR ** (NGECM-1)
DO 9 L=1,NGP
9 C(NP-1,L)=2.0*C(NP-1,L)
   B(NP,L) = 2.0*B(NP,L)-P*XR(NRG,L)*H
GO TO 10
E DO 11 L=1,NGP
   B(NP,L)=1.0
11 C(NP-1,L)=0.0
10 CONTINUE
RETURN.
END
SUBROUTINE WEIGHT

IMPLICIT REAL*8 (A-H,O-Z)

COMMON /CTRL/ EPS1,EPS2,EPS3,EFFK,TH(I1),RK1,RK2,BIG,AHOLD(9C),
1,NGP,NRG,NMAT,NGEOM,JBCCL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
2,NRVARY(9C),MVARY,ITMAXD,ITMAXI,ITO,IT,KEEP,MCODE,LBIG,JBIG,IAJ,
3,UNDUM,THOLD(9C)

COMMON /MACX/ SPECT(5),XA(15,5),XNUF(13,5),XTR(13,5),XGG(13,5,5),
1,CC(13,10),CP(13,10),CC,CT,IDMAT(1C)

COMMON /FLUX/ PHI(2C1,5),ANORM,BNORM,A(2C1,5),B(2C1,5),
1,C(2C1,5),w(2C1,5,5),s(2C1,5)

JJ=1

PWR=F,0

DO 1 I=1,NGP

H=TH(I)/NPT(I)

JMAX=NPT(I)-1

R=RWR

DO 2 J=1,JMAX

JJ=JJ+1

R=R+H

IF (NGFOM.EQ.1) RP=1.F

IF (NGFOM.EQ.2) RP=P

IF (NGFOM.EQ.3) RP=P*R

F=H*RP

DO 3 L=1,NGP

DO 3 K=1,NGP

W(JJ,K,L)=(XGG(I,K,L)+EFFK*SPECT(L)*XNUF(I,K))*F

CONTINUE

IF (I.EQ.NRG) GO TO 1

JJ=JJ+1

K=K+H

RP=RP*(NGEOM-1)

H2=TH(I+1)/NPT(I+1)

F=H*5*R

DO 4 L=1,NGP

DO 4 K=1,NGP

W(JJ,K,L)=((XGG(I,K,L)+EFFK*SPECT(L)*XNUF(I,K))*H+(XGG(I+1,K,L)+
1  EFFK*SPFCT(L)*XNUF(I+1,K)*H2/F
   RWR=RWR+TH(1)
1  CONTINUE
   H1=TH(1)/NPT(1)
   H2=TH(ARG)/NPT(NRG)
   RWP=RWR+TH(NRG)
   RP=RWR**((NGEOM-1))
   F1=H1*(1/10P)
   F2=H2*RP
   DO 5 L=1,NGP
   DO 5 K=1,NGP
   IF (JBCL) 6,6,7
6  W(1,K,L)=.0
   GO TO 8
7  W(1,K,L)=.0
8  IF (JBCR) 9,9,10
9  W(NP,K,L)=.0
   GO TO 5
10  W(NP,K,L)=(XGG(NRG,K,L)+EFFK*SPFCT(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,RIG,AHOLD(90),
1NGP,NRG,NGEM,JBCLR,JRCNFG,JAD,NP,NPT(10),IOP,NRVARY,
2IRVARY(90),MVARV,ITMAX0,ITMAXI,ITO,ITI,KEEP,MODE,LBIG,JBIG,IAJ,
3JDOM, IHOLD(90)
COMMON /MACX/ SPET(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.
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,DO,DU,Y,N)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(1),DL(1),DO(1),DU(1),Y(1),WA(2N),GA(2N)
WA(1)=DU(1)/DO(1)
GA(1)=Y(1)/DO(1)
DO 1 K=2,N
   T1=1.0/(DO(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)
   CONTINUE
2 CONTINUE
RETURN
END
SUBROUTINE SOLVE
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTPL/ EPS1, EPS2, EPS3, FFFK, TH(10), RK1, RK2, BIG, AHOOLD(90),
1NGP, NRG, NMAT, NGEO, J3CL, JRCF, NFG, JAC, NP, NPT(10), IOP, NRVARY,
2NRVARY(90), MVAR, ITMAXD, ITMAII, ITU, ITJ, KEEP, MCODE, LBIG, JBIG, IAJ,
3JDOM, IHOL(97)
COMMON /FLUX/ PHI(201,5), A0RM, BNGRM, T(201,5), B(201,5),
1C(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.D0+CO
F=ABS(CC)
IF (FMAX.GT.F) GO TO 4
LBIG=L
BIG=J
BIG=PHI(J,L)
4 CONTINUE
5 DO 1 J=1, NP
PHI(J,L)=X(J)
1 CONTINUE
RETURN
END
SUBROUTINE RESCAL
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTRL/ FPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NRG,NMAT,NGEOM,JBCJ,JBCR,NGF,JAC,NP,NPT(10),IOP,NRVARY,
2IRVARY(90),MVARY,ITMAX),ITMAXJ,ITG.ITI,KEEP,MCODE,LBIG,IBIG,IAJ,
3JDUM,IHOLD(90)
COMMON /MACX/ SPECT(5),XA(1,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
1CONC(1,1,..),D(1,5),XR(10,5),CC,CT,IMAT(10)
COMMON /FLUX/ PHI(21,5),ANORM,BNORM,A(201,5),B(201,5),
1 C(201,5),N(2,1,5,5) ,S(2,1,5)
ABS(ZZ)=DABS(ZZ)
SQR(2)=DQR(ZZ)
BNORM=1
DO 1 J=1,NP
DO 1 L=1,NGP
1 BNORM=BNORM+PHI(J,L)*PHI(J,L)
BNORM=SQR(BNORM)
DNORM=ANORM/BNORM
DO 2 J=1,NP
DO 2 L=1,NGP
2 PHI(J,L)=PHI(J,L)*DNORM
RK2=BNORM/ANORM
IF(KEEP)3,3,4
4 SIG=ABS(PHI(J,BIG,BIG)-BIG)
3 CONTINUE
RETURN
END
SUBROUTINE ADJUST
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(1),RK1,RK2,BIG,AHOLD(90),
1NGP,NRG,NMAT,NGEOM,JBC7,JBCR,NFG,JAC,NP,NPT(19),IOP,NRVARY,
2TRVARY(90),MVARY,ITMAXJ,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
3JDUM,IOHOLD(90)
COMMON /MACX/ SPECT(5),XA(1:,5),XNUF(1,5),XTR(1,5),XG(10,5,5),
1CONC(10,10),D(10,5),XR(10,5),CC,CT,TMAT(16)
ALPHA=1.1
GO TO (1,2,3,1OP
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
2 IF (ITO.EQ.0) GO TO 7
CC=1.0+((CC-1.0)/CC)*((1.0-RK2)/(RK2-RK1))
GO TO 8
7 CC=1.0
8 DO 9 J=1,NRG
CONC(MCODE,J)=CC*CONC(MCODE,J)
9 CONTINUE
RETURN
END
SUBROUTINE ERROR(N)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /ERR/IERR
WRITE (6,1) N
1 FORMAT ('1 ERROR STOP NUMBER',15)
IERR=1
RETURN
END
SUBROUTINE EDIT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/POWER/ SIGF(10,5),AKTI$201),TUTP(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),GRPHI(10,5),
3STS,PHEL(201,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
4SGG(10,5),SIGG(10,5),TH(10),PHEL(10,5),XGG(10,5),
5COMMON/CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NRG,NMAT,NGE0M,JBCL,JBCL,NAC,NNP,NPT(10),IOP,NVARY,
2IRVARY(90),MVARY,ITMAX0,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
3JNUM,IOHLD(90),
COMMON/MACX/ SPECT(5),XAI(5),XNUF(10,5),XTR(10,5),XGG(10,5),
1CONC(10,5),P(10,5),XR(10,5),CC,CT,IMMAT(10),
COMMON/MICX/ SIGC(10,5),SIGTR(10,5),XNUF(10,5),SIGF(10,5),
1SIGG(10,5),
COMMON/FLUX/ PHI(201,5),ANORM,BNORM,AV(201,5),B(201,5),
1WRITE (6,991)
WRITE (6,996)
WRITE (6,997) (I,IOHLD(I),AHOLD(I),I=1,ITO)
GO TO (1,2,3),IOP
1EFFK=1.0/EFFK
WRITE (6,993) ((1,PHEL(I,J),I=1,NGP),J=1,NGP)
WRITE (6,992) EFFK1
RETURN
2WRITE (6,992) RK2
WRITE (6,995) (J,TH(J),J=1,NGP)
WRITE (6,993) ((1,PHEL(I,J),I=1,NGP),J=1,NGP)
RETURN
3WRITE (6,992) RK2
WRITE (6,994) MVARY,(I,CONC(MCODE,J),J=1,NGP)
WRITE (6,993) ((1,PHEL(I,J),I=1,NGP),J=1,NGP)
RETURN
991 FORMAT ('I',20X,'PROGRAM EDIT')
992 FORMAT ('I','K EFFECTIVE = ',F10.6)
993 FORMAT ('I',J,PHEL(I,J)...I=SPACE POINT, J=GROUP/6(I5,I3,I1
12.5))
994 FORMAT(///' CRITICAL CONCENTRATION OF MATERIAL',IS,(/' REGION =',
1 IS,10X,'CONCENTRATION =',F10.7))
995 FORMAT(///' CRITICAL SIZE',//('THICKNESS OF REGION',IS,/' =',F10.5,
1 ' CM'))
996 FORMAT (///' OUTER ITERATION NUMBER OF INNER ITERATIONS EIGENV
1 ALUE',//)
997 FORMAT (17,20X,IS,15X,F10.6)
END
SUBROUTINE AJOINT
IMPLICIT REAL*8 (A-H, O-Z)
COMMON /CNTRL/ EPS1, EPS2, EPS3, EFK, TH(10), RK1, RK2, BIG, AHOLD(90),
1NGP, NRG, NMAT, NGEOM, JBCL, J3CK, NFG, JAD, NP, NPT(1), IOP, NRVARY,
2IRVARY(95), MVAR, ITMAXO, ITMAXI, TO, ITI, KEEP, MCODE, LBIG, JRIG, IAJ,
3JDNJ, IHDLD(90)
COMMON /MAXX/ SPECT(5), XA(10, 5), XNUF(10, 5), XTR(10, 5), XGG(10, 5, 5),
1ONC(10, 10), O(10, 5), XR(10, 5), CC, CT, IOMAT(10)
COMMON /ERR/ IERR
ABS(zz)=DABS(zz)
IAJ=1
RK1=0.3
TK=0.3
BIG=10000.0
KEEP=0
ITI=0
2 CALL SOLVE
CALL RESCAL
ITI=ITI+1
IF( ITI.GT.ITMAXI ) CALL ERROR(1)
IF( IERR .NE. 0 ) RETURN
IF( ABS(RK2-TK).LT.EPS1 ) GO TO 6
TK=RK2
GO TO 2
6 IF( BIG.LT.EPS3 ) GO TO 100
BIG=0.3
KEEP=1
GO TO 2
100 RETURN
END
SUBROUTINE AEDIT
IMPLICIT REAL*8 (A-H,O-Z)
COMMUN /CTRL/ EPS1,EPS2,EPS3,FFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NRG,MMAT,NGEM,J3CL,J3CR,NFG,JAQ,NP,NPT(15),IOP,NRVAR,
2IRVAR(9),MVARY,ITMAX0,ITMAX1,IT0,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
3JDOM,THOLD(90)
COMMUN /MACX/ SPECT(5),XA(1C,5),XNUF(10,5),XTR(1C,5),XGG(1C,5,5),
1CONC(1,1),D(1,5),XR(10,5),CC,CT,IGMAT(10)
COMMUN /FLUX/ PHI(2C1,5),ANORM,BNORM,A(2C1,5),B(2C1,5),
1C(2C1,5),W(2,1,5,5),S(2C1,5)
WRITE (6,991)
WRITE (6,993) ((I,J,PHI(I,J)),I=1,NGP),J=1,NGP)
FORMAT ('',2X,'ADJOIN EDIT')
FORMAT ('///',I,J,PHIA(I,4),1=SPACE POINT, J=GROUP',/6(15,13,012
1.5))
RETURN
END
SUBROUTINE WINI

IMPLICIT REAL*8 (A-H,O-Z)

COMMON/POWER/ SIFG(1,5),AKTIS(201),TOTP(10),SILI(10),FISIT(10),
ISDIIT(1,15),TMETOL(10),SPLYM(10),FISITM(10),TMETLM(10),ALKGEM(10),
1GRPH1(10,5),GRPHA1(10,5),GRPHA2(10,5),ALKGE(10),GRPH1(10,5),
2GRPH2(10,5),GRPHA1(10,5),GRPHA2(10,5),ALKGE(10),GRPH1(10,5),
3GRPH3(10,5),GRPHA1(10,5),GRPHA2(10,5),ALKGE(10),GRPH1(10,5),
4GRPH4(10,5),DI(10,5),

COMMON/AXCI/ SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
1CON(1,15),DI(15,5),XP(15,5),CC,CT,IDMAT(1,)

COMMON/CNTRL/ EPS1, EPS2, EPS3, EFKK, TH(10), RK1, RK2, BIO, AHOLD(90),
1NGP, NRG, NMAT, NGEOM, JBC, JBC, NCG, JAC, NP, NPT(10), IOP, NRVAR,
2LRVAR(99), MVAR, ITMA0, ITMA1, IT0, ITI, KEEP, MCODE, LG1, JRG, IA1, 
3JONJ, IT0LH(99)

COMMON/DDELTA/ THSA(10,5), THSF(10,5), THD(10,5), THST(10,5,5),
1THPP(10,5), THSTT(10,5), DSNM(10,5), DSNM(10,5),
2DSFM(10,5,5), DSTTPM(10,5), DSTM(10,5), THSF(10,5), DSFM(10,5),
3SFU(10,5), SCU(10,5), SUP(10,5), POSED(10), CONCP(10), VNO

COMMON/ACONC/ SIGG(10,5), SIGTR(10,5), XNUF(10,5), SIGF(10,5),
1SIGGG(10,5),5

COMMON/KCNSY/ SB(10,5), PPU(10), PU(10), PJU(10), PRS(10), URN(10),
1URC(10), SD(10), DPL(10)

COMMON/CMX/ CMRX(30), NPR, KNA, NCR, ILNA

COMMON/DELFI/ IP, IU

COMMON/ITER/ NIT

NGE=NCR+1

IF(KNA,F0,1) GO TO 2
IF(NIT,NE,2) GO TO 1
READ(5,2500) VNO,NPR,NCR

2502 FORMAT(F10.5,215)

IF(NPR,NE,2) GO TO 1
READ(5,2520) IDNA

2522 FORMAT(F15.5)

1 READ(5,2520) IP, IU

2510 FORMAT(215)

READ(5,6) CONCP(IP), CONCP(IU)

601 FORMAT(2E15.5)
WRITE(6,61)
WRITE(6,61) CONCP(IP),CONCP(IU)
61: FORMAT('///5X,1 CONCENTRATION OF PURF MATERIALS')
611 FORMAT(///2F15.3)
DO 9 I=1,NGP
DO 9 J=1,NGP
THSA(I,J)=-
THSF(I,J)=-
THNSF(I,J)=
THTRP(I,J)=.0
THSTT(I,J)=-
DSAM(I,J)=-
DSFM(I,J)=0.
DNSFM(I,J)=0.
OTPRM(I,J)=0.
OSTFM(I,J)=0.
SFU(I,J)=.0
SCU(I,J)=.0
SUP(I,J)=.0
SB(I,J)=.1
DO 9 K=1,NGP
THST(I,J,K)=-.1
OSTM(I,J,K)=.0
CONTINUE
DO 11 I=1,NGP
DO 11 K=1,NGP
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)*SIGG(IP,I)*XNU(IP,I)
THNSF(K,I)=THNSF(K,I)-CONCP(IU)*SIGG(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)
DG 11 L=1,NGP
THST(K,I,L)=THST(K,I,L)+CCNCP(IP)*SIGG(IP,I,L)
\[ \text{THST}(K, I, L) = \text{THST}(K, I, L) - \text{CONCP}(IU) \times \text{SIGG}(IU, I, L) \]

\[ \text{THSTT}(K, I) = \text{THST}(K, I) + \text{THST}(K, I, L) \]

11 CONTINUE

2 DO 13 I = 1, NGP
3 DO 13 K = 1, NRG
4 \[ \text{DDM}(K, I) = -\text{DTRPM}(K, I) / (3.0 \times (\text{STR}(K, I) \times \text{STR}(K, I))) \]
5 13 THP(K, I) = -\text{THTRP}(K, I) / (3.0 \times (\text{STR}(K, I) \times \text{STR}(K, I)))

60 CONTINUE

61 DO 6 I = 1, NGP
7 DO 6 K = 1, NRG
8 \[ \text{SFU}(K, I) = \text{CONCP}(IU) \times \text{SIGF}(IU, I) \]
9 \[ \text{SCU}(K, I) = \text{CONCP}(IU) \times (\text{SIGC}(IU, I) - \text{SIGF}(IU, I)) \]
10 \[ \text{SUP}(K, I) = \text{SCU}(K, I) + \text{CONCP}(IP) \times \text{SIGC}(IP, I) \]
11 \[ \text{SB}(K, I) = \text{CONC}(IU, K) \times (\text{SIGC}(IU, I) - \text{SIGF}(IU, I)) - \text{CONC}(IP, K) \times \text{SIGC}(IP, I) \]

61 CONTINUE

63 DO 10 K = 1, NRG
7 SD(K) = CONC(IU, K) \times \text{SIGC}(IU, I)
8 RETURN
9 END
SUBROUTINE ISCHIS
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/POWER/ SGFM(10,5), AKTI(21), TOTP(10), SYLI(10), FISIT(10),
LSODIT(10), TMETOL(T10), SYLIM(10), FISITM(10), TMETLM(10), ALKGEM(10),
2GRPH2(10,5), GRPH(10,5), GRPH2(10,5), ALKG(10), GRPH1(10,5),
3STS, PH(21,5), SR(10,5), SA(10,5), SNUF(10,5), STR(10,5),
4SGG(10,5,5),01(10,5),
COMMON/CNTRL/ FPS1, EPS2, EPS3, EFFK, TH(10), RK1, RK2, BIG, AHOLD(90),
1NGP, NRG, NMAT, NGEQ4, JBC, JCR, NF, JAC, NP, NPT(10), IOP, NRVARY,
2RVA(90), MVAR, ITMAX, ITMAXI, IT, ITI, IC, KMOD, LBIG, JBIG, IAJ,
3AJUM, IHOLD(90),
COMMON/MAX/ SPECT(5), XA(10,5), XNUF(10,5), XTR(10,5), XGG(10,5,5),
1THC(10,10), 01(10,5), XR(10,5), CC, CT, IDM(10),
COMMON/ERR/ IERR
COMMON/MIC/ SIGC(10,5), SIGTR(10,5), XNU(10,5), SIGF(10,5),
1SGGF(10,5,5)
COMMON/FLUX/ PHI(201,5), ANORM, BNNRM, A(201,5), B(201,5),
1C(201,5), W(201,5,5), S(201,5),
COMMON/KSWV/ SB(10,5), PPU(10), PUT(10), PDU(10), PRS(10), URN(10),
1URC(10), 0C(10), DOPL(10)
COMMON/DELTAF/ THSA(10,5), THNSF(10,5), THD(10,5), THST(10,5,5),
1THPP(10,5), THST(10,5), DSA(10,5), DNSF(10,5), DCM(10,5),
2ST(10,5,5), DTRP(10,5), USTM(10,5), THSF(10,5), DSFM(10,5),
3SFU(10,5), SCU(10,5), SUP(10,5), POED(10), CONCP(10), VNO
COMMON/ITER/ NIT
COMMON/CONV/ CRMA(30), VRK, KNA, NCR, IDNA
DIMENSION RRA(30), SPOL(30), DOPLC(30)
EFFK1=1.0/EFFK
IF(KNA.EQ.1) GO TO 13
WRITE (6,992) EFFK1
992 FORMAT (/// 'K EFFECTIVE = ',F9.6)
GO TO 11
10 WRITE(6,993) EFFK1
993 FORMAT (/// 'K EFFECTIVE OF VOITED CORE = ',F9.6)
14=NIT+1
SPOL(4)=EFFK1
IF (NIT .EQ. 0) GO TO 11
IF (DABS (SPOL (M) - SPOL (M-1)) .LT. .00001) CALL EXIT

11 CALL WINI
IF (NIT .NF. 0) GO TO 12
AKTIS (1) = 0.0
DO 8 J = 2, NP
AKTIS (J) = AKTIS (J-1) + STS
CONTINUE

8 BR = 0.0
BREX = 0.0
DOP = 0.0
BRU = 0.0
POWERT = 0.0
FSDIN = 0.0
NPT (1) = NPT (1) + 1
STSI = STS * 0.33333333333333
DO 1 L = 1, NRG
TOTP (L) = 0.0
URN (L) = 0.0
URC (L) = 0.0
PU (L) = 0.0
PPU (L) = 0.0
PRS (L) = 0.0
PDU (L) = 0.0
POWERD (L) = 0.0
SYL = 1.0
SYLCM = 0.0
FISIO = 0.0
FISIOE = 0.0
FSDIO = 0.0
IF (L .EQ. 1) GO TO 2
MA = MA + NPT (L-1)
K = NPT (L) - 2 + MA
L = MA
GO TO 3
2 \(K=NPT(1)-2\)
3 \(N=1\)
4 DO 4 I=1,NGP
5 \(TEMP=.\)
6 \(SYL=C.\)
7 \(FIST=\_\)
8 \(FISIM=\_\)
9 \(FSDI=\_\)
10 DO 5 J=N,K,2
11 \(TEMP=TEMP+PHL(J,I)*AKTIS(J)+4.0*PHL(J+1,I)*AKTIS(J+1)+PHL(J+2,I)*AKTIS(J+2)\)
12 \(SYL=SYL+PHL(J,I)*PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,I)*PHI(J+1,I)*AKTIS(J+1)+PHL(J+2,I)*PHI(J+2,I)*AKTIS(J+2)\)
13 \(CONTINUE\)
14 DO 6 M=1,NGP
15 \(FIS=0\)
16 \(FSD=\_\)
17 DO 7 J=N,K,2
18 \(FIS=FIS+PHL(J,M)*PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,M)*PHI(J+1,I)*AKTIS(J+1)+PHL(J+2,M)*PHI(J+2,I)*AKTIS(J+2)\)
19 \(FSD=FSD+PHL(J,M)*PHI(J,I)*AKTIS(J)+4.0*PHL(J+1,M)*PHI(J+1,I)*AKTIS(J+1)+PHL(J+2,M)*PHI(J+2,I)*AKTIS(J+2)\)
20 \(CONTINUE\)
21 \(FISM=FISM*NSFM(L,M)\)
22 \(FIS=FIS*TNSF(L,M)\)
23 \(FISI=FISI+FIS\)
24 \(FISIM=FISIM+FISIM\)
25 \(FSD=FSD*SNUF(L,M)\)
26 \(FSDI=FSDI+FSD\)
27 \(CONTINUE\)
28 \(FISM=FISM*SPECT(I)\)
29 \(FISIM=FISIM*SPECT(I)\)
30 \(FISIO=FISIO+FISI\)
31 \(FISIM=FISIM+FISIM\)
32 \(FSDI=FSDI*SPECT(I)\)
33 \(FSDI=FSDI+FSDI\)
UPN(L) = URN(L) + TEMP*SF(L,I)
PPU(L) = PPU(L) + TEMP*THSF(L,I)
PP = 3F + TEMP*SB(L,I)
IF(L . GT. NCR) GO TO 25
BRN = BR
GO TO 26
25
BREX = BREX + TEMP*SB(L,I)
26
DNL = DNL - TEMP*SUP(L,I)
URC(L) = URC(L) - TEMP*SCUL(L,I)
RPUI = BRUI + TEMP*SCUL(L,I)
TTP(L) = TTP(L) + TEMP*SIGMI(L,I)
POWER(L) = POWER(L) + PHL(N,I)*SIGMI(L,I)
PSL = PSL + PHL(N,I)*SUP(L,I)*VNO
DNL = DNL + PHN(L,I)*THSF(L,I)
SYLM = SYLDM + SYL
IF(L . LE. 5) GO TO 45
IF(L . GT. NCR) GO TO 45
DOPL(L) = SYL
45
SYL = SYL + THSA(L,I)
SYLO = SYLO + ST
SYLM = SYLDM + SYL
CONTINUE
DOPL(L) = DOPL(L) + STSI
TTP(L) = TTP(L) + STSI
SYLI(L) = SYLO + STSI
SYLIM(L) = SYLDM + STSI
POWER = POWER + TTP(L)
URC(L) = URC(L) + STSI
PU(L) = PU(L) + STSI
FISIT(L) = FISI0 + STSI*EFFK
FISITM(L) = FISI0M + STSI*EFFK
FSDIT(L) = FSDTO + STSI
FSDIN = FSDIN + FSDIT(L)
URN(L) = URN(L) + STSI
PPU(L) = PPU(L) + STSI
CONTINUE
BR=BR*STSI/POWFR
BRIN=BRIN*STSI/POWFR
BR=BR*STSI/POWFR
DOP=DOP*STSI/FSDIN
BRU=BRU*STSI
PNorm=1.0 /POWFR
BRU=BRU*PNorm
IF(KNA.EQ.1) GO TO 210
WRITE(6,521) BR
WRITE(6,522) BRIN
WRITE(6,523) BRFX
IF(NPR.NE.1) GO TO 200
IF(NIT.EQ.1) GO TO 210
NRA(NIT)=3R
IF(NIT.EQ.1) GO TO 210
IF(NABS(BRA(NIT)-BRA(NIT-1)).LT.3.00001) CALL EXIT
GO TO 210
IF(NPR.NE.3) GO TO 213
NI=NIT+1
DOPC(NI)=DOP
150 MA=0.0
DO 14 L=1,NRG
TMETO = 3.0
T4ETOM=2.0
IF(L.EQ.1) GO TO 15
MA=MA+NPT(L-1)
K=NPT(L)-2+MA
N=MA
GO TO 16
15 K=NPT(1)-2
N=1
IA=NGP-1
16 DO 17 I=1,IA
IA=I+1
TMETA=0.0
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)
CONTINUE

TMFTM=TMET*DSTM(LIM)
TMET=TMET*THST(LIM)
TMFTA=TMETM+TMET
TMFTAM=TMETAM+TMETM

CONTINUE

TMETO =TMETM +TMET
TMETOM=TMETOM+TMETM
CONTINUE

TMELOL(L)=TMETO *STSI
TMETLM(L)=TMETOM*STSI
CONTINUE

STSI=(-.5/STS)
MA=0.0
DO 20 L=1,NRG
ALKG(L)=0.0
ALKGFM(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

KA=K+2
KS=K+3
DO 23 I=1,NGP
ALKG=0.0
DO 24 J=N,KA

CONTINUE
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(I,1) = 0.0
GRPHA1(1,1) = 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,1) + SPECT(I) * SNUF(L,M) * EFFK
  CRA = SGG(L,M,1) + SPFCT(M) * SNUF(L,M) * EFFK
  AD1 = AD1 + CRA * PHL(KB,M)
  AD2 = AD2 + CRA * PHL(KA,M)
  ADA1 = ADA1 + CRA * PHI(KB,M)
  ADA2 = ADA2 + CRA * PHI(KA,M)
41 CONTINUE
CR = SA(L,1) + SR(L,1)
AF1 = CR * PHL(KB,1)
AF2 = CR * PHL(KA,1)
AFA1 = CR * PHI(KB,1)
AFA2 = CR * PHI(KA,1)
GRPH2(L,1) = AKTIS(KA) * A(KA,1) / AKTIS(KB) + 0.5 * STS * ((AF1 - AD1) +
  1 AKTIS(KA) *(AF2 - AD2) / AKTIS(KB)) / DI(L,1)
GRPHA2(L,1) = AKTIS(KA) * B(KA,1) / AKTIS(KB) + 0.5 * STS * ((AFA1 - ADA1) +
  1 AKTIS(KA) *(AFA2 - ADA2) / AKTIS(KB)) / DI(L,1)
ALKG=ALKG*STSI
ALKG=ALKG+(GRPHA1(L,I)*GRPHA1(L,I)*AKTIS(MA)+A(N,I)*B(N,I))
1 AKTIS(N))=0.5*STSI
ALKG=ALKG+(GRPHA2(L,I)*GRPHA2(L,I)*AKTIS(KB)+A(KA,I)*B(KA,I))
1 AKTIS(KA))=0.5*STSI
ALKG=ALKG*DDM(L,I)
ALKG=ALKG*THD(L,I)
ALKG=ALKG*THD(L,I)
ALKG=ALKG*ALKG
ALKGEM(L)=ALKGEM(L)+ALKG

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
PRSL(L)=PRSL(L)*PNORM
PDU(L)=PDU(L)*PNORM
POWED(L)=POWED(L)*PNORM
IF(KNA.EQ.1) GO TO 20
WRITE(6,5)6 L,POWED(L)

CONTINUE
NPT(I)=NPT(I)-1
DO 70 J=1,NP
DO 70 I=1,NGP
PHL(J,I)=PHL(J,I)*PNORM
7) CONTINUE
POWERT=100.*0
WRITE(6,5,4) POWERT
504 FORMAT(/// TOTAL POWER=', F15.7)
506 FORMAT('REGION', I3, 1X, '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),
1 FSDIT(10), TMETOL(10), SYLIM(10), FISITM(10), TMETLM(10), ALKGEM(10),
2 GRPH2(10,5), GRPHA1(10,5), GRPHA2(10,5), ALKG(10), GRPH1(10,5),
3 STS, PHL(201,5), SR(10,5), SA(10,5), SNUF(10,5), STR(10,5),
4 SG(10,5,5), DI(10,5)
COMMON/CNTRL/ EPS1, EPS2, EPS3, EFFK, TH(10), RK1, RK2, BIG, AHOME(90),
1 NGP, NRG, NMAT, NGEOM, JBLCK, JBCN, NFG, JAC, NP, NPNT(10), IOP, NRVARY,
2 IRVARY(90), NRVARY, ITMAX, ITMAXO, ITMAXI, ITO, ITI, KEP, MCODE, LIG, JBIG, IAJ,
3 JDIUM, IHOME(90)
COMMON/COWE/ HA(10), ARG(3), WK(99,11), UUL(13), UUR(13), VUL(13,5),
1 VUR(13,5), NOP(K), NBD(10), NOPT, NRE
COMMON/GREK0/ U2L(13), U2R(13), UVL(13,5), UVR(13,5), V2L(13,5),
1 V2R(13,5), V2UL(13), V2UR(13), V2L(13,5), V2R(13,5),
2 DVUSR(13,5), VVR(13,5), DV2R(13,5), DV2L(13,5), DVVL(13,5),
3 DVVR(13,5), UVSL(13,5), VUSR(13,5), VUSL(13,5),
4 DVUSL(13,5), DUVL(13,5), DVUR(13,5), DUVL(13,5),
5 DVUSL(13,5), DUVL(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
1600 STS=STS*U .33333333333333
NOPT=0
DO 23 K=1,NRG
NOPT=NOPT+NOP(K)
23 CONTINUE
NF=NOPT+1
DI 5 K=1,NF
U2L(K)=0.0
U2R(K)=0.0
U2L(K)=0.0
DU2R(K)=0.0
DUUL(K)=0.0
DUUR(K)=0.0
UUL(K)=0.0
UUR(K)=0.0
DO 5 I=1,NGP
VUL(K,I)=0.0
VUR(K,I)=0.0
ULV(K,I)=0.0
ULV(K,I)=0.0
DVUSR(K,I)=0.0
DVUSL(K,I)=0.0
DVUSR(K,I)=0.0
DVUSL(K,I)=0.0
DV2R(K,I)=0.0
DV2L(K,I)=0.0
DVVL(K,I)=0.0
DVVR(K,I)=0.0
UVSR(K,I)=0.0
UVSL(K,I)=0.0
VUSP(K,I)=0.0
VUSL(K,I)=0.0
DUVL(K,I)=0.0
DUVR(K,I)=0.0
DVUR(K,I)=0.0
DVUL(K,I)=0.0
DO 5 L=1,NGP
V2R(K,I,L)=0.0
V2L(K,I,L)=0.0
VVL(K,I,L)=0.0
VVR(K,I,L)=0.0
CONTINUE
NRE=NRG-1
NBD(1)=NOP(1)+1
DO 1 K=2,NRG
NBD(K)=NBD(K-1)+NOP(K)
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) * (R - HA(K)) / HA(K) + 0.6
UUL(N) = (9.0 / 140.0) * (R + HA(K)) / HA(K) + 0.6
DU2R(N) = 1.2 * (R + HA(K)) / HA(K) - 0.6
DU2L(N) = 1.2 * (R - HA(K)) / HA(K) + 0.6
DUUL(N) = DUUR(N - 1)
DUUR(N) = DUUL(N + 1)
CONTINUE
N = N + 1
R = R + HA(K)
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) * HA(K) / 7.0) + 13.0 * HA(K + 1) * (R + HA(K + 1)) / 35.0
UUR(N) = UUR(N - 1)
UUL(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)
CONTINUE
N=1
R=0
DO 10 K=1, NRG
M=NP(K)-1
DO 11 J=1, M
R=R+HA(K)
N=N+1
DO 12 I=1, NGP
DO 13 L=I, NGP
V2R(N, I, L)=(\(R+HA(K)) \times 105.0 - HA(K)/168.0\) \times HA(K) \times HA(K) \times HA(K)/
1 (DI(K, I) \times DI(K, L))
V2L(N, I, L)=(\(R-HA(K)) \times 105.0 + HA(K)/168.0\) \times HA(K) \times HA(K) \times HA(K)/
1 (DI(K, I) \times DI(K, L))
VVR(N, I, L)=(-P/140.0C-HA(K)/280.0) \times HA(K) \times HA(K) \times HA(K)/(DI(K, I) *
1 DI(K, L))
VVL(N, I, L)=(-R/140.0C+HA(K)/280.0) \times HA(K) \times HA(K) \times HA(K)/(DI(K, I) *
1 DI(K, L))
CONTINUE
UVSR(N, I)=((R+HA(K))*11.0/210.0 - HA(K)/28.0) \times HA(K) \times HA(K)/DI(K, I)
UVSL(N, I)=(-R/420.0C+HA(K)/280.0) \times HA(K) \times HA(K)/DI(K, I)
UVR(N, I)=(-13.0*R/420.0C-HA(K)/70.0)*HA(K)*HA(K)/DI(K,I)
UVL(N, I)=(-13.0*R/420.0C-HA(K)/70.0)*HA(K)*HA(K)/DI(K,I)
VUSR(N, I)=UVSR(N, I)
VUSR(N, I)=UVSR(N, I)
VUR(N, I)=(-13.0*R/420.0C+HA(K)/60.0) \times HA(K) \times HA(K)/DI(K, I)
VUL(N, I)=(-13.0*R/420.0C+HA(K)/60.0) \times HA(K) \times HA(K)/DI(K, I)
DUVSR(N, I)=(R+HA(K))*0.1/DI(K, I)
DUVSL(N, I)=(-R-HA(K))*0.1/DI(K, I)
DVUSL(N, I)=DUVSL(N, I)
DVUSR(N, I)=DUVSR(N, I)
DVUR(N, I)=0.1*R/DI(K, I)
DVUL(N, I)= -DVUR(N, I)
DVVR(N, I)=0.1*(R+HA(K))/DI(K, I)
DVVL(N, I)=0.1*(R-HA(K))/DI(K, I)
DV2R(N, I)=((R+HA(K))*2.0/15.0 - 0.1*HA(K))*HA(K)/(DI(K, I)*DI(K, I))
DV2L(N,I) = ((R-HA(K))*2.0/15.0+0.1*HA(K))*HA(K)/(DI(K,I)*DI(K,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)/(DI(K,I)*DI(K,I)))*(-1.0*R/30.0-HA(K)/60.0)

CONTINUE
N=N+1
R=R+HA(K)

CONTINUE
DO 14 I=1,NGP
DO 15 L=I,NGP
V2L(N,I,L) = ((R-HA(K))/105.0+HA(K)/168.0)*HA(K)*HA(K)*HA(K)/
1/(DI(K,I)*DI(K,L))
VVL(N,I,L) = VVR(N-1,I,L)
VUL(N,I,L) = UVR(N+1,I,L)

14 CONTINUE
R=.
R=R+NOP(K)*HA(K)
N=NBK(K)
DO 17 I=1,NGP
DO 18 L=I,NGP
V2R(N,I,L) = ((R+HA(K+1))/105.0+HA(K+1)/168.0)*HA(K+1)*HA(K+1)*HA(K+1)/
1/(DI(K+1,I)*DI(K+1,L))
V2L(N,I,L) = ((R-HA(K))/105.0+HA(K)/168.0)*HA(K)*HA(K)*HA(K)/
1/(DI(K,I)*DI(K,L))
VVL(N,I,L) = VVR(N-1,I,L)
VUR(N,I,L) = UVR(N+1,I,L)

18 CONTINUE
UV2R(N,I) = ((R+HA(K+1))/11.0/210.0-HA(K+1)/28.0)*HA(K+1)*HA(K+1)/
1/(DI(K+1,I))
UVL(N,I) = (-1.0-R-HA(K))*11.0/210.0-HA(K)/28.0)*HA(K)*HA(K)/DI(K,I)
UVR(N,I) = VUL(N+1,I)
UVL(N,I) = VUR(N-1,I)
VUSL(N,I) = UVSL(N,I)
VUSR(N,I) = UUSR(N,I)
VUR(N,I) = ULR(N+1,I)
VUL(N,I) = UVR(N-1,I)
OUVR(N,I) = (R+HA(K+1)) * 0.1 / DI(K+1, I)
OUVS(N,I) = -(R-HA(K)) * 0.1 / DI(K, I)
OUVR(N,I) = DVUL(N+1,I)
OUVR(N,I) = DVVR(C-I)
DUVL(N,I) = DVUL(N+1,I)
DVVR(N,I) = ((R+HA(K+1)) * 0.1 / DI(K+1, I) + (R-HA(K) * 0.1 / DI(K, I)) * (-1.0 * R / 30.0 + HA(K) / 60.0))
1/ 60.0)

CONTINUE

DO 19 I=1,NGP
  UVR(1,I) = VUL(2,I)
  OUVR(1,I) = DVUL(2,I)
CONTINUE
RETURN
END
SUBROUTINE BIGMAT
IMPLICIT REAL*8(A-H,O-Z)
COMMON/POWER/SIGFM(10,5),AKTIS(201),TOTP(10),SYLI(10),FISIT(10),
IFSIT(10),TMETOL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
2GRPHA1(10,5),GRPHA2(10,5),GRPHA3(10,5),ALKGEM(10),GRPHA(10,5),
3STS,PHL(21,5),SR(10,5),SA(10,5),SNUF(10,5),STR(10,5),
4SGG(11C,5,5),DI(10,5)
COMMON/CNTRL/FPS1,EPS2,EPS3,EFFK,TH1(10),RK1,RK2,BIG,AHOLD(90),
1NGP,NRG,NMAT,NGFOM,J3CL,JBCR,NFG,JAC,NP,NPT(10),IOP,NRVARY,
2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,BBIG,JBIG,IAJ,
3JDUM,IHOLD(90)
COMMON/MACX/SPECT(5),XA(10,5),XNUF(10,5),XTR(10,5),XGG(10,5,5),
1CONC(10,5),JR(10,5),CC,CT,IMAT(10)
COMMON/COOE/HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
1VUR(13,5),NOD(10),NBD(10),NOPT,NRE
COMMON/GREKO/U2L(13),U2R(13),UVL(13,5),UVR(13,5),V2R(13,5,5),
1DU2L(13),DU2R(13),DVUL(13),DVUR(13),V2L(13,5,5),VV(13,5,5),
2DVUSR(13,5),VVR(13,5,5),DV2R(13,5),DV2L(13,5),DVVL(13,5),
3DVVR(13,5),DVUSR(13,5),DVSL(13,5),DUVL(13,5),DVUR(13,5),
4DVUL(13,5),DVUL(13,5),DVUR(13,5),DVVL(13,5),DVUL(13,5),
5DVVR(13,5),DVUSR(13,5),DVSL(13,5),G(99,99)
COMMON/ATHENS/BU(3),BV(3),OLU(5),OLV(5,5),ADU(5),ADV(5),DDU(5),
1DDV(5),TU(13,5),TV(13,5),TUL(13,5),TUR(13,5),TVL(13,5),TVR(13,5),
2DV(13,5),DV(13,5),DV(13,5),DV(13,5),DV(13,5),DV(13,5),DV(13,5)
DIMENSION F(99,11),LW(99),MW(99)

C      MATRICES CONSISTING THE DIAGONAL ELEMENTS OF THE BIG MATRIX
C
DO 1 K=1,NRG
DO 1 I=1,NGP
SNUF(K,I)=SNUF(K,I)*EFFK
1 CONTINUE
NAGN=2*NRG+1
NOM=2*NOPT*NGP-1
DO 24 M=1,NOM
DO 29 K=1,NAGN
F(M,K) = 0

CONTINUE

DO 24 N = 1, NOM
G(M,N) = 0

CONTINUE

K = N

DO 25 I = 1, NGP
NA = 1
N = NA + K
M = I + K

CR = SPECT(I) * SNUF(1, I) - SR(1, I) - SA(1, I)

IF (I.EQ.1) GO TO 100

G(M,N) = DI(1, I) * 0.6 + CR * U2R(1)
G(M,N+1) = DI(1, I) * 0.6 + CR * UUR(1)
G(M,N+2) = DI(1, I) * DUVR(1, I) + CR * UVR(1, I)

M = M + 1

G(M,N) = DI(1, I) * DUUL(2) + CR * UUL(2)
G(M,N+1) = DI(1, I) * (DU2R(2) + DU2L(2)) + CR * (U2R(2) + U2L(2))
G(M,N+2) = DI(1, I) * (DUSR(2, 1) + DUSL(2, I)) + CR * (USR(2, I) + USL(2, I))
G(M,N+3) = DI(1, I) * DUR(2) + CR * UUR(2)
G(M,N+4) = DI(1, I) * DUVR(2, I) + CR * UVR(2, I)

M = M + 1

G(M,N) = -DI(1, I) * DVUL(2, I) + CR * VUL(2, I)
G(M,N+1) = -DI(1, I) * (DV2R(2, I) + DV2L(2)) + CR * (V2R(2, I) + V2L(2))
G(M,N+2) = -DI(1, I) * (DVUSR(2, 1, I) + DVUSL(2, I, I)) + CR * (VUSR(2, I, I) + VUSL(2, I, I))
G(M,N+3) = -DI(1, I) * DUR(2, I) + CR * UUR(2, I)
G(M,N+4) = -DI(1, I) * DUVR(2, I) + CR * UVR(2, I)

NA = NA + 1

GO TO 101

100 G(M,N) = -DI(1, I) * (DU2R(2) + DU2L(2)) + CR * (U2R(2) + U2L(2))
F(M,N) = DI(1, I) * DUUL(2) - CR * UUL(2)
G(M,N+1) = -DI(1, I) * (DUUSR(2, I) + DUUSL(2, I)) + CR * (VUSR(2, I) + USL(2, I))
G(M,N+2) = -DI(1, I) * DUR(2) + CR * UUR(2)
G(M,N+3) = -DI(1, I) * DUVR(2, I) + CR * UVR(2, I)

M = M + 1

F(M,N) = DI(1, I) * DVUL(2, I) - CR * VUL(2, I)
G(M,N) = -DI(1,1) * (DVUSR(2,1) + DVUSL(2,1)) + CR * (VUSL(2,1) + VUSR(2,1))
G(M,N+1) = -DI(1,1) * (DV2R(2,1) + DV2L(2,1)) + CR * (V2R(2,1) + V2L(2,1))
G(M,N+2) = -DI(1,1) * DVVR(2,1) + CR * VVR(2,1,1)
G(M,N+3) = -DI(1,1) * DVVR(2,1) + CR * VVR(2,1,1)

L = 2
DO 26 NR = 2, NRG
DO 27 JJ = 1, 2
JI = 2 - JJ
M = M + 1
L = L + 1
N = NA + K
J = NR - JI
CRL = SPECT(I) * SNUF(J, I) - SP(J, I) - SA(J, I)
CRR = SPECT(I) * SNUF(NR, I) - SP(NR, I) - SA(NR, I)
G(M,N) = -DI(J, I) * DUUL(L) + CRL * UUL(L)
G(M,N+1) = -DI(J, I) * DUVL(L, I) + CRL * UVL(L, I)
G(M,N+2) = -DI(J, I) * DU2L(L) - DI(NR, I) * DU2R(L) + CRL * U2L(L) + CRR * U2R(L)
G(M,N+3) = -DI(J, I) * DU2L(L, I) - DI(NR, I) * DU2R(L, I) + CRL * U2L(L, I) + CRR * U2R(L, I)

1 CRR = VUSR(L, I)
G(M,N+4) = -DI(NR, I) * DUUR(L, I) + CRR * UUR(L, I)
G(M,N+5) = -DI(NR, I) * DUVR(L, I) + CRR * UVR(L, I)

M = M + 1
G(M,N) = -DI(J, I) * DUUL(L, I) + CRL * UUL(L, I)
G(M,N+1) = -DI(J, I) * DUVL(L, I) + CRL * UVL(L, I)
G(M,N+2) = -DI(J, I) * DU2L(L, I) - DI(NR, I) * DU2R(L, I) + CRL * U2L(L, I) + CRR * U2R(L, I)
G(M,N+3) = -DI(J, I) * DU2L(L, I) - DI(NR, I) * DU2R(L, I) + CRL * U2L(L, I) + CRR * U2R(L, I)

1 CRR = VUSR(L, I)
G(M,N+4) = -DI(NR, I) * DUUR(L, I) + CRR * UUR(L, I)
G(M,N+5) = -DI(NR, I) * DUVR(L, I) + CRR * UVR(L, I)

M = M + 1
J = NRG

26 CONTINUE
27 CONTINUE
28 IF(NR .EQ. NRG) GO TO 28
\begin{verbatim}
N=NA+K
L=L+1
CR=SPECT(I)*SNUF(JI)-SR(J,I)-SA(JI)
G(M,N)=-DI(JI)*DUUL(L)+CR*UUL(L)
G(M,N+1)=-DI(JI)*DUVL(LI)+CR*VUL(L,I)
G(M,N+2)=-DI(JI)*(DU2L(L)+DU2R(L))+CR*(U2L(L)+U2R(L))
G(M,N+3)=-DI(JI)*(DUVSL(L,I)+DUVSRI(L,I))+CR*(UVSL(L,I)+UVSR(L,I))
G(M,N+4)=-DI(JI)*DUVR(LI)+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)
G(M,N+2)=-DI(J,I)*(DVUL(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)+V2R(L,I))
G(M,N+4)=-DI(J,I)*DVVR(L,I)+CR*VVR(L,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)
G(M,N+2)=-DI(J,I)*(DVUL(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)+V2R(L,I))
G(M,N+4)=-DI(J,I)*DVVR(L,I)+CR*VVR(L,I)
IF(I.EQ.1) GO TO 112
K=K+2*NOPT
GO TO 55
112 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
56 CONTINUE
55 CONTINUE

\end{verbatim}
NA=1
N=NA+KW
CR=SPECT(L)*SNUF(1,1)
IF(L.EQ.1) GO TO 103
G(M,N)=CR*U2R(1)
G(M,N+1)=CR*UUR(1)
G(M,N+2)=CR*UVR(1,1)
M=M+1
103 G(M,N)=CR*UUL(2)
G(M,N+1)=CR*(U2R(2)+U2L(2))
G(M,N+2)=CR*(UVSR(2,1)+UVSL(2,1))
G(M,N+3)=CR*UUR(2)
G(M,N+4)=CR*UVR(2,1)
M=M+1
G(M,N)=CR*VUL(2,L)
G(M,N+1)=CR*(VUSR(2,L)+VUSL(2,L))
G(M,N+2)=CR*(V2R(2,L,1)+V2L(2,L,1))
G(M,N+3)=CR*VUR(2,L)
G(M,N+4)=CR*VVR(2,L)
NA=NA+1
K=2
DO 57 NR=2,NRG
DO 58 JJ=1,2
JI=2-JJ
J=NR-JI
K=K+1
N=NA+KW
M=M+1
CRL=SPECT(L)*SNUF(J,1)
CRR=SPECT(L)*SNUF(NR,1)
G(M,N)=CRL*UUL(K)
G(M,N+1)=CRL*UVL(K,1)
G(M,N+2)=CRL*U2L(K)+CRR*U2R(K)
G(M,N+3)=CRL*UVSL(K,1)+CRR*UVSR(K,1)
G(M,N+4)=CRR*UUR(K)
G(M,N+5)=CRR*UVR(K,1)
M=M+1
G(M,N)=CRL*VUL(K,L)
G(M,N+1)=CRL*VVL(K,L,I)
G(M,N+2)=CRL*VUSL(K,L)+CRR*VUSR(K,L)
G(M,N+3)=CRL*V2L(K,L,I)+CRR*V2R(K,L,I)
G(M,N+4)=CRR*VUR(K,L)
G(M,N+5)=CRR*VVR(K,L,1)
NA=NA+2
IF(NR.E.NRG) GO TO 59

CONTINUE
CONTINUE

M=M+1
J=NRG
K=K+1
N=NA+KW
CR=SPECT(L)*SNJF(J, I)
G(M,N)=CR*UUL(K)
G(M,N+1)=CR*UVL(K, I)
G(M,N+2)=CR*(U2L(K)+U2R(K))
G(M,N+3)=CR*(UVSL(K, I)+UVSR(K, I))
G(M,N+4)=CR*UVR(K, I)
M=M+1
G(M,N)=CR*VUL(K,L)
G(M,N+1)=CR*VVL(K,L,I)
G(M,N+2)=CR*(VUSL(K,L)+VUSR(K,L))
G(M,N+3)=CR*(V2L(K,L,I)+V2R(K,L,I))
G(M,N+4)=CR*VVR(K,L, I)
M=M+1
N=N+2
K=K+1
G(M,N)=CR*VUL(K,L)
G(M,N+1)=CR*VVL(K,L,I)
G(M,N+2)=CR*V2L(K,L,I)
KW=KW+2*NOPT

CONTINUE
IF(L.EQ.1) GO TO 104
IA = IA + 2 * NOPT
GO TO 55

104 IA = IA + 2 * NOPT - 1
55 CONTINUE

C
C MATRICES CONSISTING THE BELOW THE DIAGONAL ELEMENTS OF THE BIG
C MATRIX
C
KW = 0
DO 65 L = 1, NGPE
IA = 2 * NOPT * L - 1
IA = IA + 1
DO 66 I = IB, NGP
M = IA + 1
N = NA + KW
CR = SGG(1, L, I) * SPECT(I) * SNUF(1, L)
IF (L .EQ. 1) GO TO 105
G(M, N) = CR * U2R(1)
G(M, N + 1) = CR * UUR(1)
G(M, N + 2) = CR * UVR(1, L)
M = M + 1
G(M, N) = CR * UUL(2)
G(M, N + 1) = CR * (U2R(2) + U2L(2))
G(M, N + 2) = CR * (UVSR(2, L) + UVSL(2, L))
G(M, N + 3) = CR * UUR(2)
G(M, N + 4) = CR * UVR(2, L)
M = M + 1
G(M, N) = CR * VUL(2, I)
G(M, N + 1) = CR * (UVSR(2, I) + UVSL(2, I))
G(M, N + 2) = CR * (V2R(2, L, I) + V2L(2, L, I))
G(M, N + 3) = CR * VUR(2, I)
G(M, N + 4) = CR * VVR(2, L, I)
NA = NA + 1
GO TO 105

105 G(M, N) = CR * UUR(1)
\[ F(M+N) = -CR \times U2R(1) \]
\[ G(M,N+1) = CR \times UVR(1,L) \]
\[ M = M + 1 \]
\[ F(M,N) = -CR \times UUL(2) \]
\[ G(M,N+1) = CR \times (U2R(2) + U2L(2)) \]
\[ G(M,N+2) = CR \times U2R(2) \]
\[ G(M,N+3) = CR \times UVR(2,L) \]
\[ M = M + 1 \]
\[ F(M,N) = -CR \times VUL(2, I) \]
\[ G(M,N+1) = CR \times (VUSR(2, I) + VUSL(2, I)) \]
\[ G(M,N+2) = CR \times VUR(2, I) \]
\[ G(M,N+3) = CR \times VVR(2, I) \]

\( K = 2 \)
\[ DO \ 67 \ NR=2, NRG \]
\[ DO \ 68 \ JJ=1,2 \]
\[ JI=2-JJ \]
\[ J=NR-JI \]
\[ K=K+1 \]
\[ N=NA+KW \]
\[ M=M+1 \]
\[ CRL=SGG(j, L, I) + \text{SPECT}(I) \times \text{SNUF}(J, L) \]
\[ CRR=SGG(NR, L, I) + \text{SPECT}(I) \times \text{SNUF}(NR, L) \]
\[ G(M,N)=CRL \times UUL(K) \]
\[ G(M,N+1)=CRL \times UVR(K,L) \]
\[ G(M,N+2)=CRL \times U2L(K)+CRR \times U2R(K) \]
\[ G(M,N+3)=CRL \times UVR(K,L)+CRR \times U2R(K) \]
\[ G(M,N+4)=CRR \times U2L(K)+CRR \times UVSL(K,L) \]
\[ G(M,N+5)=CRR \times UVR(K,L) \]
\[ M=M+1 \]
\[ G(M,N)=CRL \times VUL(K, I) \]
\[ G(M,N+1)=CRL \times VVL(K,L, I) \]
\[ G(M,N+2)=CRL \times VUSL(K,I)+CRR \times VUSR(K,I) \]
\[ G(M,N+3)=CRL \times V2L(K,L, I)+CRR \times V2R(K,L, I) \]
\[ G(M,N+4)=CRR \times V2L(K,L, I) \]
G(M,N+5)=CRR*VVR(K,L,I)
NA=NA+2
IF(NR.EQ.NRG) GO TO 69
68 CONTINUE
67 CONTINUE
69 M=M+1
J=NRG
CR=SGG(J,L,I)+SPECT(I)*SNUF(J,L)
K=K+1
N=NA+KW
G(M,N)=CR*UUL(K)
G(M,N+1)=CR*UVL(K,L)
G(M,N+2)=CR*(U2L(K)+U2R(K))
G(M,N+3)=CR*(UVSL(K,L)+UVSR(K,L))
G(M,N+4)=CR*UVR(K,L)
M=M+1
G(M,N)=CR*UUL(K,L)
G(M,N+1)=CR*UVL(K,L)
G(M,N+2)=CR*(V2L(K,L,I)+V2R(K,L,I))
G(M,N+3)=CR*(VUSR(K,L,I))
G(M,N+4)=CR*VVR(K,L,I)
M=M+1
N=N+2
K=K+1
G(M,N)=CR*VUL(K,L)
G(M,N+1)=CR*VVL(K,L,I)
G(M,N+2)=CR*(UVSL(K,L,I)+UVSR(K,L,I))
G(M,N+3)=CR*(V2L(K,L,I)+V2R(K,L,I))
IA=IA+2*NOPT
66 CONTINUE
IF(L.EQ.1) GO TO 107
KW=KW+2*NOPT
GO TO 65
107 KW=KW+2*NOPT-1
65 CONTINUE
DO 95 M=1,NOM
DO 95 N=1,NOM
**RIGHT HAND SIDE MATRIX**

```plaintext
G(M,N) = 0.01 * G(M,N)

CONTINUE
CALL WENDO

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) = DLVL(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)
F(L, M) = DUR(N, I)
F(L + 1, M) = DVR(N, I)
F(L + 2, M) = DU(N + 1, I)
```

```
BIGM0325
BIGM0326
BIGM0327
BIGM0328
BIGM0329
BIGM0330
BIGM0331
BIGM0332
BIGM0333
BIGM0334
BIGM0335
BIGM0336
BIGM0337
BIGM0338
BIGM0339
BIGM0340
BIGM0341
BIGM0342
BIGM0343
BIGM0344
BIGM0345
BIGM0346
BIGM0347
BIGM0348
BIGM0349
BIGM0350
BIGM0351
BIGM0352
BIGM0353
BIGM0354
BIGM0355
BIGM0356
BIGM0357
BIGM0358
BIGM0359
BIGM0360
```
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/ SIGMF(10,5), AKTIS(201), TOTP(10), SYLI(10), FISIT(10),
1 FSDIT(10), TMEQ(10), SYLM(10), FISITM(10), TMDTL(1C), ALKGEM(10),
2 GRPH2(10,5), GRPHA1(10,5), GRPHA2(1C,5), ALKGEM(10), GRPH1(10,5),
3 STSPHL(201,5), SR(10,5), SA(10,5), SNUF(10,5), STR(1C,5),
4 SGG(10,5), DI(10,5)
COMMON/CNTRL/ FPS1, EPS2, EPS3, EFFK, TH(10), RK1, RK2, BIG, AHQNL(90),
1 NPG, NRG, NMT, NGEOM, JBC, JBCR, JAD, NP, NPT(10), IO, NRVARY,
2 NRVARY(90), NRVARY, HTRV, ITMAXD, ITMAX, IT0, ITI, KIP, MCODE, LBIQ, JBQ, IAQ,
3 JQDUM, IQHOL(90)
COMMON/MACX/ SPECT(5), XA(10,5), XNUMF(1,5), XTR(10,5), XGG(10,5,5),
1 IGNC(10,10), D(10,5), XR(10,5), CC, CT, IDMNT(10)
COMMON/COWF/ HA(10), ARG(3), WK(99,11), UUL(13), UUR(13), VUL(13,5),
1 VUR(13,5), NQ(10), NBD(10), NQPNT, NRE
COMMON/ATHENS/ BU(3), BV(3), OL(5), OLV(5,5), ADU(5), ADV(5), DDU(5),
1 NDV(5), TV(13,5), TUL(13,5), TUL(13,5), TUR(13,5), TVR(13,5),
2 DUL(13,5), DVL(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), TSTST(10,5), DSNM(10,5), CNSM(10,5), CDN(10,5),
2 DSTM(10,5,5), DPNM(10,5), DSTM(10,5), THS(10,5), DFSM(10,5),
3 SFU(10,5), SCU(10,5), SUP(10,5), POW(10,5), CONCP(10), VNO
C
C
C
*** ***
DO 1 L=1, NRG
DO 1 I=1, NPG
SR(L,I)=SR(L,I)+SA(L,I)
1 CONTINUE
C
C
C
*** ***
STSI=STSI*,33333333333333
NPT(1)=NPT(1)+1
L=1
K=0.5*(NPT(1)-1)-0.5
HSQ=HA(N,1)*HA(1)
N1=L+2
DO 40 I=1,NGP
OLU(I)=0.0
DO 41 J=1,K,2
DO 42 JA=1,3
JI=J+JA-1
ARG(JA)=AKTIS(N1)-AKTIS(JI)
BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*(3.0-2.0*ARG(JA)/HA(1))*AKTIS(JI)
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)
CONTINUE
OLU(I)=OLU(I)*STSI
CONTINUE
DO 43 I=1,NGP
ADU(I)=0.0
DDU(I)=0.0
DO 44 M=1,NGP
PROS=(THD(1,I)/DI(1,I))*(SGG(1,M,I)+SPECT(I)*SNUF(1,M))-THST(1,M
1,I)*SPECT(I)*THNSF(1,M)*EFFK
PROD=(DDM(1,I)/DI(1,I))*(SGG(1,M,I)+SPECT(I)*SNUF(1,M))-DSTM(1,M
1,I)*SPECT(I)*DNSFM(1,M)*EFFK
ADU(I)=ADU(I)+PROS*OLU(M)
DDU(I)=DDU(I)+PROD*OLU(M)
CONTINUE
AF =THSA(1,I)+THSTT(1,I)-(THD(1,I)/DI(1,I))*SR(1,I)
DF =DSAM(1,I)+DSTTM(1,I)-(DDM(1,I)/DI(1,I))*SR(1,I)
TU(1,I)=AF *OLU(I)+ADU(I)
DU(1,I)=DF *OLU(I)+DDU(I)
CONTINUE
KK=1
MA=0
DO 45 L=1,NRG
KK=KK+1
HSQ=HA(L)*HA(L)
IF(L.EQ.1) GO TO 46
MA = MA + NPT(L - 1)
K = NPT(L) - 2 + MA
N = MA
K1 = MA + 0.5 * NPT(L) - 2 + 0.5
N1 = K1 + 2
K2 = K + 2
GO TO 47
46
K = NPT(1) - 2
N = 1
K1 = (NPT(1) - 1) * 0.5 - 1 + 0.5
N1 = K1 + 2
K2 = K + 2
47 DO 48 I = 1, NGP
OLU(I) = C.0
DO 2 M = 1, NGP
2 OLV(M, I) = 0.0
DO 49 J = N, K1, 2
DO 50 JA = 1, 3
JI = J + JA - 1
ARG(JA) = AKTIS(JI) - AKTIS(N)
BU(JA) = (ARG(JA) * ARG(JA) / HSQ) * (3.0 - 2.0 * ARG(JA) / HA(L)) * AKTIS(JI)
BV(JA) = (ARG(JA) * ARG(JA) / (DI(L, I) * HA(L))) * (-1.0 + ARG(JA) / HA(L)) * AKTIS(JI)
CONTINUE
50 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 49 M = 1, NGP
OLV(M, I) = OLV(M, I) + PHL(J, M) * BV(1) + 4.0 * PHL(J1, M) * BV(2) + PHL(J2, M) * BV(3)
1 BV(3)
49 CONTINUE
DO 51 J = N1, K, 2
DO 52 JA = 1, 3
JI = J + JA - 1
ARG(JA) = AKTIS(K2) - AKTIS(JI)
BU(JA) = (ARG(JA) * ARG(JA) / HSQ) * (3.0 - 2.0 * ARG(JA) / HA(L)) * AKTIS(JI)
BV(JA) = (ARG(JA) * ARG(JA) / (DI(L, I) * HA(L))) * (1.0 - ARG(JA) / HA(L)) * 
1.0

AKTIS(JI)

CONTINUE
J1 = J + 1
J2 = J + 2

OLU(J, I) = OLU(J, I) + PHL(J1, I) * BV(1) + 4.0 * PHL(J2, I) * BV(2) + PHL(J2, I) * BV(3)

DO 51 M = 1, NGP
OLV(M, I) = OLV(M, I) + PHL(J1, M) * BV(1) + 4.0 * PHL(J1, M) * BV(2) + PHL(J2, M) * 
BV(3)

CONTINUE

OLU(J, I) = OLU(J, I) * STSI

DO 48 M = 1, NGP
OLV(M, I) = OLV(M, I) * STSI

CONTINUE

DO 53 I = 1, NGP
ADU(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) - SPECT(I) * SNUF(L, M) * THST(L, M) - THD(L, I) / DI(L, I) * SGG(L, M, I) * SPECT(I) * SNUF(L, M) - DSTM(L, M)

1, I) = SPECT(I) * DNSTF(L, M) * EFFK
PROD = (DDM(L, I) / DI(L, I)) * SGG(L, M, I) * SPECT(I) * SNUF(L, M) - DSTM(L, M)

ADU(I) = ADU(I) + PROS * OLU(M)
ADV(I) = ADV(I) + PROS * OLU(M)

CONTINUE

AF = THSA(L, I) + THSTT(L, I) - (THD(L, I) / DI(L, I)) * SR(L, I)

TU(KK, I) = AF * OLU(I) + ADU(I)
DU(KK, I) = DF * OLU(I) + DDU(I)
TU(KK, I) = AF * OLU(I, I) + ADV(I)
DU(KK, I) = DF * OLV(I, I) + DDV(I)

CONTINUE
45 KK=KK+1
CCONTINUE
NPT(1)=NPT(1)-1
K=NPT(1)*0.5+1+0.5
DO 89 L=1,NRE
KK=NBD(L)
HSQ=HA(L)*HA(L)
N=K
K=K+(NPT(L)+NPT(L+1))*0.5+0.5
K1=N+NPT(L)*0.5-2+0.5
N1=K1+2
K2=K-2
DO 85 I=1,NGP
OLU(I)=0.0
DO 3 M=1,NGP
OLV(M,I)=0.0
DO 86 J=N,K1,2
DO 87 JA=J1,J2,1
ARG(JA)=AKTIS(JI)-AKTIS(N)
BU(JA)=(ARG(JA)*ARG(JA)/HSQ)*((3.0-2.0*ARG(JA)/HA(L))*AKTIS(JI)
BV(JA)=(ARG(JA)*ARG(JA)/(DI(L,I)*HA(L)))*(-1.0+ARG(JA)/HA(L))*
1 AKTIS(JI)
87 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 86 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)
86 CONTINUE
OLU(I)=OLU(I)*STSI
DO 85 M=1,NGP
OLV(M,I)=OLV(M,I)*STSI
85 CONTINUE
DO 88 I=1,NGP
ADU(I) = C.0
DDU(I) = C.0
ADV(I) = 0.0
DOV(I) = 0.0
DO 60 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, I) * EFFK
ADU(I) = ADU(I) + PROS * OLU(M)
DDU(I) = DDU(I) + PROD * OLU(M)
ADV(I) = ADV(I) + PROS * OLV(M, I)
DOV(I) = DOV(I) + PROD * OLV(M, I)
60 CONTINUE
AF = THSA(L, I) + THSTT(L, I) - (THD(L, I) / DI(L, I)) * SR(L, I)
DF = DSAAM(L, I) + DSTTM(L, I) - (DDM(L, I) / DI(L, I)) * SR(L, I)
TUL(KK, I) = AF * OLU(I) + ADU(I)
DUL(KK, I) = DF * OLU(I) + DDU(I)
TVL(KK, I) = AF * OLV(I, I) + ADV(I)
DVL(KK, I) = DF * OLV(I, I) + DOV(I)
88 CONTINUE
HSQ = HA(L + 1) * HA(L + 1)
DO 61 I = 1, NGP
OLU(I) = C.0
DO 7 M = 1, NGP
OLV(M, I) = 0.0
7 J = N1, K2, 2
DO 63 JA = 1, 3
JI = J + JA - 1
ARG(JA) = AKTIS(K) - AKTIS(JI)
BUI(JA) = (ARG(JA) * ARG(JA) / HSQ) * (3.0 - 2.0 * ARG(JA) / HA(L + 1)) * AKTIS(JI)
BU(JA) = (ARG(JA) * ARG(JA) / (DI(L + 1, I) * HA(L + 1))) * (1.0 - ARG(JA) / HA(L + 1)) * AKTIS(JI)
1 CONTINUE
J1 = J + 1
J2 = J + 2
CLU(I) = OLU(I) + PHL(J1, 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) + 4.4 * PHL(J, M) * BV(2) + PHL(J, M) * 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

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
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)=G.0
DDV(I)=G.0
DO 74 M=1,NGP
PROS=(THD(L,I)/DI(L,I))*(SGG(L,M,I)+SPECT(I)*SNUF(L,M))-THST(L,I)
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)+DSTM(L,I)-(DDM(L,I)/DI(L,I))*SR(L,I)
TVL(NBU(L,I))=AF*OLV(L,I)+ADV(I)
DVL(NBD(L,I))=DF*OLV(L,I)+DDV(I)
73 CONTINUE
RETURN
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
TB=IK
C(IR)=0.0
DO 92 I=1,M
JI=JI+N
TB=TB+1
92 C(IR)=C(IR)+G(JI)*A(TB)
RETURN
END
SUBROUTINE DMINV(A,N,D,L,M)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(1),L(1),M(1)
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+1
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)
A(JI)=HOLD
30 A(JI) =HOLD
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)
A(JK)=A(JI)
47 A(JI)=HOLD
45 IF (BIGA) 48, 46, 48
46 D=0.0
  RETURN
48 DO 55 I=1,N
  IF (I-K) 50, 55, 50
50 IK=NK+I
  A(IK)=A(IK)/(-BIGA)
55 CONTINUE
DO 65 I=1,N
  IK=NK+I
  HOLD=A(IK)
  IJ=I-N
  DO 65 J=1,N
    TJ=IJ+N
    IF (I-K) 60, 65, 60
60 IF (J-K) 62, 65, 62
62 KJ=IJ-I+K
    A(IJ)=HOLD*A(KJ)+A(IJ)
65 CONTINUE
KJ=K-N
75 DO 75 J=1,N
    KJ=KJ+N
    IF (J-K) 70, 75, 70
70 A(KJ)=A(KJ)/BIGA
75 CONTINUE
D=D*BIGA
A(KK)=1.0/BIGA
80 CONTINUE
K=N
100 K=(K-1)
105 IF (K) 150, 150, 105
105 I=L(K)
    IF (I-K) 120, 120, 108
108  JQ=N*(K-1)
    JR=N*(I-1)
    DO 110  J=1,N
    JK=JQ+J
    HOLD=A(JK)
    JI=JR+J
    A(JK)=-A(JI)
110  A(JI)=HOLD

120  J=M(K)
    IF(J-K) 120 100,125

125  KI=K-N
    DO 130  I=1,N
    KI=KI+N
    HOLD=A(KI)
    JI=KI-K+J
    A(KI)=-A(JI)
130  A(JI)=HOLD
    GO TO 130
150  RETURN
END
SUBROUTINE BASINT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(9C),
1NGP,NRG,NMAT,NGEOM,JBCX,JBCR,NFG,IAJ,NPT(10),IOP,NRVARY,
2IRRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,ITI,KEEP,MCODE,LBIG,JBIG,IAJ,
3JDUH,IHOLD(9C)
COMMON/POWER/ SIGFM(10,5),AKTIS(20),TOTP(10),S(YL1(10),FISIT(10),
1FSIT(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(21,5),SR(10,5),SA(10,5),ALKNF(10,5),STR(10,5),
4 SGG(10,5,5),DI(10,5)
COMMON/COVE/ HA(10),ARG(3),WK(99,11),UUL(13),VUL(13),UUR(13),VUL(13,5),
1 VUR(13,5),NOP(10),NBO(10),NOPTNRE
C
C INTEGRALS OF BASE POLYNOMIALS
C
UUR(1)=HA(1)*HA(1)*G.15
N=1
R=G.0
DO 1 K=1,NGP
M=NOP(K)-1
DO 2 J=1,M
R=R+HA(K)
1 CONTINUE
N=N+1
R=P+HA(K)
CONTINUE
N=N+1
N=1

INTEGRALS OF BASE POLYNOMIALS

UUR(1)=HA(1)*HA(1)*G.15
N=1
R=G.0
DO 1 K=1,NGP
M=NOP(K)-1
DO 2 J=1,M
R=R+HA(K)
1 CONTINUE
N=N+1
R=P+HA(K)
CONTINUE
N=N+1
N=1
DO 4 I=1,NGP
  VUL(N,I)=-HA(K)*HA(K)*((R-HA(K))/(12.)*DI(K,I))+0.05*HA(K)/
  DI(K,I))
  CONTINUE
R=0.0
DO 5 K=1,NRF
  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.)*DI(K,I))+0.05*HA(K)/
    DI(K,I))
    VUR(N,K)=HA(K+1)*HA(K+1)*((R+HA(K+1))/(12.0*DI(K+1,I)))-0.05*
    HA(K+1)/DI(K+1,I))
    CONTINUE
6
UUL(N)=HA(K)*(.35*HA(K)+0.5*(R-HA(K)))
UUR(N)=HA(K+1)*(-.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),
1FSIT1(10),TMETL(10),SYLIM(10),FISITM(10),TMETLM(10),ALKGEM(10),
2GRPH2(10,5),GRPHAI(10,5),GRPHA2(10,5),ALKGE(10),GRPH1(10,5),
3STS,PHL(201,5),SR(10,5),SA1(10,5),SNUF(10,5),STR(10,5),
4SGG(10,5,5),DI(10,5)
COMMON/CNTRL/ EPS1,EPS2,EPS3,EFFK,TH(10),RK1,RK2,BIG,AHOLD(90),
1INGP,NGP,NGTM,NMAT,NR,CR,JAC,NC,JAC,NGP,NRT(10),IDP,NRVARY,
2IRVARY(90),MVARY,ITMAXO,ITMAXI,ITO,IT1,KEEP,MCODE,LBIG,JBIG,IAJ,
3JDUM,IPHOLD(90)
COMMON/MACX/ SPECT(5),XA(10,5),XNUF(10,5),XR(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),
1URC(10),SD(10),DOPL(10)
COMMON/DELTA/ THSA(10,5),THNSF(10,5),THO(10,5),THST(10,5,5),
1THTRP(10,5),THSTT(10,5),DSAM(10,5),DNSFM(10,5),DDM(10,5),
2DSTM(10,5,5),DTRPM(10,5),DSTM(10,5),THSF(10,5),DSFM(10,5),
3SFM(10,5),SCU(10,5),SUP(10,5),PORED(10),CONCP(10),VNO
COMMON/CONW/ HA(10),ARG(3),WK(99,11),UUL(13),UUR(13),VUL(13,5),
1VUR(13,5),NOP(10),NAD(10),NOPT,NRE
COMMON/CONV/ CRMA(30),NPR,KNA,NCR,INCA
COMMON/DF/ IP,RU
COMMON/ITER/ NI
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),
1PE(14),E(200),K0(6),JH(14)
RFAL*4 X(17)

IF(KNA.EQ.1) GO TO 350
IF(NIT.NE.0) GO TO 1500
NVC=NCR
NAR=1
NEO=3*NCR+2
NAV=4*NCR+1
READ(5,1000) (UO(L),L=1,NCR)
READ(5,2000) PDL,THU

1600 FORMAT(7F10.0)
2000 FORMAT(2F10.0)
   WRITE(6,5500) (U0(L),L=1,NCR)
   WRITE(6,5600) PDL,THU
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
         CONTINUE
      CS(J)=0.0
      CONTINUE
   DO 16 I=1,NEQ
      BS(I)=0.0
      CONTINUE
16 CONTINUE
   NF=NOPT+1
   CA(1,1,1)=1.0
   DO 100 M=2,NAGN
      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
            CA(1,I,M)=WK(K,M)
            K=K+1
         92 CONTINUE
      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
CONTINUE
DO 96 M=1,NAGN
CB(NFI,M)=WK(K,M)
96 CONTINUE
K=K+1
97 CONTINUE
DO 97 I=1,NGP
DO 97 M=1,NAGN
CB(I,M)=WK(K,M)
CA(NF, I, M)=0.0
97 CONTINUE
IF(NAR.EQ.2) GO TO 60
LL=2
LU=2*NGP+1
GO TO 61
60 LL=2*NCR+3
LU=2*NRG+1
61 DO 2C 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,1)
CS(J)=CS(J)+COB*CA(1,M,1)
27 CONTINUE
N=1
DO 22 L=1,NGP
DO 23 J=1,2
N=N+1
JI=JJ-1
NR=L+JI
CM1=SIGFM(L,M)*UUL(N)+SIGFM(NR, M)*UUR(N)
CM2=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, J) = AS(1, J) + COM1 * CA(N, M, 1) + COM2 * CB(N, M, 1)
CS(J) = CS(J) + COB1 * CA(N, M, 1) + COB2 * CB(N, M, 1)
DO 24 I = LLU, 2
J = J + 1
AS(1, J) = AS(1, J) + COM1 * CA(N, M, I) + COM2 * 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)
COR = SB(L, M) * VUL(N, M)
J = 1
AS(1, J) = AS(1, J) + COM1 * CB(N, M, 1) + COM2 * CB(N, M, 1)
CS(J) = CS(J) + COB1 * CB(N, M, 1) + COB2 * CB(N, M, 1)
DO 26 I = LLU, 2
J = J + 1
AS(1, J) = AS(1, J) + COM1 * CB(N, M, I) + COM2 * CB(N, M, I)
CS(J) = CS(J) + COB1 * CB(N, M, I) + COB2 * CB(N, M, I)
26 CONTINUE
20 CONTINUE
WRITE (6, 900) (CS(J), J = 1, NAV)
900 FORMAT (BD15.7)
LM = NVC + 1
RS(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) - TMFTOL(L) - ALKGE(L)
AS(2,J)=PERT
BS(2)=BS(2)+UO(L)*PERT
CONTINUE
GO TO 64
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)
CONTINUE
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)
CONTINUE
DO 28 I=LI,LN
BS(I) = PS(I) + AS(I,1) * PHL(1,1)
DO 28 J = 2, L, M
L = J - 1
BS(I) = BS(I) + AS(I, J) * UO(L)
28 CONTINUE
IF (NAP.EQ.2) GO TO 65
DO 66 I = 3, L, N
J = I - 1
L = J - 1
AS(I, J) = AS(I, J) + PDU(L)
66 CONTINUE
GO TO 68
65 DO 45 L = 1, NCR
I = 1 + L
BS(I) = PDL - PRS(L) + BS(I)
45 CONTINUE
K = LN + 1
KA = LN + NVC
L =
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
273 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 = L1, 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
PFRT = FISIT(L) - SYLI(L) - TMFTOL(L) - ALKGE(L)

CS(J) = PERT
CS(1) = 0.0

II = 0
MX = NEQ
NN = NAV
CALL SIMPLE(TI, MX, NN, AS, BS, CS, KU, X, P, JH, XX, Y, PE, E)
KNA = C
IF(NPR .NE. 3) GO TO 334
CM = C .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=', 12)
5900 FORMAT(' CRITICAL MASS IN KG = ', 1PD15.7)
5500 FORMAT('REGION', 15, IX, 'FISSILE VOLUME FRACTION=', D15.7)

DO 81 J = 2, MU
L = L + 1
UO(L) = X(J)

80 CONTINUE
WRITE(6, 5000) (L, UO(L), L = 1, NCR)

5000 FORMAT('REGION', 15, IX, '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(K0(1).EQ.1) CALL EXIT
GO TO(301,301,303),NPR
203 CRMA(NIT)=CM
IF(ABS(CRMA(NIT)-CRMA(NIT-1)).LT.1.E-01) CALL EXIT
301 RETURN
FND
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(I), C(I), P(I), X(I), Y(I), PE(I), E(I)
REAL*4 XX
INTEGER INFLAG, MX, NN, KO(6), KB(I), JH(I)
EQUIVALENCE (XX, LL)
DIMENSION A(14, 17)
INTEGER I, IA, INVCP, 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
NMPV = 0
M = MX
N = NN
TFXP = .000015259
NCUT = 4*M + 200
NVER = M*.5 + 5
M2 = M*M
FEAS = .FALSE.
IF (INFLAG.NE.' ') GO TO 1400
C* 'NEW' START PHASE ONE WITH SINGLECN 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

SIMPO001
SIMPO002
SIMPO003
SIMPO004
SIMPO005
SIMPO006
SIMPO007
SIMPO008
SIMPO009
SIMPO010
SIMPO011
SIMPO012
SIMPO013
SIMPO014
SIMPO015
SIMPO016
SIMPO017
SIMPO018
SIMPO019
SIMPO020
SIMPO021
SIMPO022
SIMPO023
SIMPO024
SIMPO025
SIMPO026
SIMPO027
SIMPO028
SIMPO029
SIMPO030
SIMPO031
SIMPO032
SIMPO033
SIMPO034
SIMPO035
SIMPO036
JH (I) = -1
1401 CONTINUE
C* 'VFR' CREATE INVERSE FROM 'KB' AND 'JH' (STEP 7)
1320 VER = .TRUE.
INVC = 0
NUMVR = NUMVR +1
TRIG = .FALSE.
DO 1101 I = 1,M
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 CALL JMY
C CHOOSE PIVOT
C
1114 TY = (0.0)
KQ = .FALSE.
DO 1104 I = 1,M
IF (JH(I) .NE. -1.0) DABS(Y(I)) .LE. TPIV) GO TO 1104
IF (KQ) GO TO 1116
IF (X(I) .EQ. 0.0) 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. 0.0) DABS(Y(I)) .LE. TY) GO TO 1104

SIMP0037
SIMP0038
SIMP0039
SIMP0040
SIMP0041
SIMP0042
SIMP0043
SIMP0044
SIMP0045
SIMP0046
SIMP0047
SIMP0048
SIMP0049
SIMP0050
SIMP0051
SIMP0052
SIMP0053
SIMP0054
SIMP0055
SIMP0056
SIMP0057
SIMP0058
SIMP0059
SIMP0060
SIMP0061
SIMP0062
SIMP0063
SIMP0064
SIMP0065
SIMP0066
SIMP0067
SIMP0068
SIMP0069
SIMP0070
SIMP0071
SIMP0072
1117    TY = DABS(Y(I))
1118    IR = I
1104   CONTINUE
      KRIJT) = (       TFST PIVOT
      IF (TY.LE.C.) GO TO 1102
      PIVOT
      GO TO 900
C 900   CALL PIV
1102 CONTINUE
C       RESER 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 (.NOT.FEAS) GO TO 500
      FEAS = .TRUE.
      DO 1201 I = 1,M
      IF (X(I).LT.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
      ARSC = .FALSE.
      GO TO 599
1250 FEAS = .FALSE.
      NEG = .TRUE.
501 DO 504 J = 1, M
P(J) = 0.

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

CONTINUE

GO TO 505

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

CONTINUE

505 continue

C* 'MIN' FIND MINIMUM REDUCED COST

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)

CONTINUE

IF (FEAS) DT = DT + C(J)

IF (ABSC) DT = -DARS(DT)

IF (DT.GE.BB) GO TO 701

BB = DT

JT = J

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 = 
   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)
602 CONTINUE
   GO TO 605
605 CONTINUE
C COMPUTE PIVOT TOLERANCE
   YMAX = .A
DO 626 I = 1,M
   YMAX = DMAX1(DABS(Y(I)),YMAX)
626 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=1., 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 (IP.NE.0) GO TO 1099
   AA = 1.9E+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
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.0.0 .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)
   IA = JH(IR)
   IF (IA.GT.0) KB(IA) = 0
NUMPV = NUMPV + 1
JH(IR) = JT
KB(JT) = IR
YI = -Y(IR)
Y(IR) = -1.0
LL = 0

DO 904 J = 1, M
   L = LL + IR
   IF (F(L).NE.'-') 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

DO 908 I = 1, M
   XOLD = X(I)
   X(I) = XOLD + XY * Y(I)
   IF (.NOT.VER.AND.X(I).LT.L..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
INFINITE SOLUTION

K = 2
GO TO 25

PROBLEM IS CYCLING

K = 4
GO TO 250

FEASIBLE OR INFEASIBLE SOLUTION

K = 0
250 IF (.NOT.FEAS) K = +1

DO 1399 J = 1,N
    XX = J
    KBJ = KB(J)
    IF (KBJ.NE.0) XX = X(KBJ)
    KB(J) = LL
1399 CONTINUE

KO(1) = K
KO(2) = ITER
KO(3) = INVC
KO(4) = NUMVR
KO(5) = NUMPV
KO(6) = JT
RETURN
END
REFERENCES


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


