JAERI 1256

JAERI 1256

FEM-BABEL

A Computer Program for Solving Three-Dimensional Neutron Diffusion Equation by the Finite Element Method

March 1978

日本原子力研究所

Japan Atomic Energy Research Institute

編集兼発行　日本原子力研究所
印　　刷　共同印刷株式会社

JAERI 1256

FEM-BABEL

A Computer Program for Solving Three-Dimensional Neutron Diffusion Equation by the Finite Element Method

Takeharu ISE, Toshio YAMAZAKI* and Yasuaki NAKAHARA

Tokai Research Establishment, Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

Received November 29, 1977

The finite element method has been applied to solve accurately the multi-dimensional neutron diffusion equation on a modern computer. A new computer program FEM-BABEL has been developed by adopting the solution algorithm based on the Galerkin-type scheme. This three-dimensional program makes use of the combination of prism- and box-shaped elements to simulate reactor geometries efficiently. The successive over-relaxation method is adopted to solve the system equation and the inner iterations are accelerated using the coarse mesh rebalancing technique.

Numerical calculations have demonstrated the present finite element method has advantages over the finite difference method for solving realistic three-dimensional problems in view of computing cost.

Keywords

Finite Element Method, Neutron Diffusion Equation, Three-Dimensional Calculation, Prism-Shaped Element, Box-Shaped Element, Galerkin-Type Approximation, Computer Program, Successive Over-Relaxation, Coarse Mesh Rebalancing Technique.

---

* Sumitomo Heavy Industries, Ltd.

FEM-BABEL
有限要素法による3次元中性子拡散方程式の解法プログラム

日本原子力研究所　東海研究所

伊　勢　武　治・山　崎　俊　雄*・中　原　康　明

1977年11月29日受理

多次元中性子拡散方程式を大型電子計算機により精度良く解く為に有限要素法の応用が計られ，ガラキン型の数値解法アルゴリズムと計算プログラム FEM-BABEL の開発がなされた．この3次元用プログラムは三角柱と四角柱要素との組み合わせを用いており，体系方程式の解法としては SOR 法，中性子束の反復計算の加速法としては粗メッシュ再釣合い法が採用されている．実在体系についての数値計算では，差分法プログラムより計算機の使用コストの点で優位性が示された．

* 住友重機工業株式会社

Contents

目　　次

1. Introduction

The finite element method (FEM)[1),2)] has been proved to be a very useful tool for discretizing practical differential equations[3)]. Since Ohnishi advocated the applicability of FEM to reactor physics calculations[4)], FEM has been developed to solve neutron diffusion[5)–16)] and transport equations.[17)–23)] It offers many other possibilities than the discretization by the finite difference method (FDM). For this, two reasons may be mentioned; The first is that with elements of appropriate shape the method can easily be applied to represent complicated geometric structures flexibly (geometrical flexibility).[5),6),18)] The second is that within the elements the unknown functions can be approximated by using interpolation polynomials of any desired degree (higher order approximation).[7)–10),15),23)] Thus it is possible to obtain very precise answers by using FEM.

However, one must of course pay for these advantages; One is the complexity of the solution algorithm introduced from the variability in the element shape and in the approximation order. The resulting equation with an irregularly occupied matrix is to be solved by using a complicated method, mostly the direct method.[24)] Even with this method, computer programs for solving a large system of matrix equations require a sophisticated data management system. For this problem, however, we can find nowadays promising solution procedures.[25)] Another is that the user must specify a large amount of input data inherent in the finite element method. To facilitate this, finite element mesh generating programs are being developed, so that it will become sufficient for the user to specify a relatively small amount of data.[26)–28)]

Three-dimensional diffusion calculation is the most practical way to take account of all essential features of realistic nuclear design problems. Among various kinds of three-dimensional finite elements, like the tetrahedron, hexahedron, quadrilateral prism, triangular prism (prism-shaped element), rectangular prism (box-shaped element) and so on,[29)] the prism- or box-shaped element is considered to be most appropriate from the viewpoint of computing time and computer storage.[30),31)] It is reported that calculations using the terahedron element[32)] could not converge for a practical problem within reasonable computing cost.[33)] Thus we have here chosen prism- and box-shaped elements in our computer program.

For realistic three-dimensional problems, a million unknowns may be needed to solve the system equations accurately and the coefficient matrices have generally sparse and irregular structure. Accordingly, the iterative method rather than the direct one shall be used to solve such large matrix equations from its potentiality by taking account of the next generation computer, because the iterative method requests less severe storage requirements and seems more efficient for most problems of large size. However, the implementation and effectiveness of the iterative methods have not been clear in the finite element approach to practical problems.[25)] Thus we here adopt the successive over-relaxation method in our computer program.

The remainder of this report is arranged as follows. In Chapter 2, we present the formalism of the finite element method for the multi-dimensional neutron diffusion equation. Chapter 3 describes the three-dimensional computer program developed here according to the formalism in Chapter 2. This chapter is intended to read also as user's manual for our program. Chapter 4 gives the verification of the program by solving a problem for which the exact solution is known. In addition, the applicabilities are demonstrated through a realistic large problem of a pressurized water reactor.

2. Solution of the Neutron Diffusion Equation by the Finite Element Method

We start to express the multigroup neutron diffusion equation in the form of the Galerkin approximation.[1] Next we construct concrete expressions of the basis functions (called the shape function in engineering) for triangular[5], rectangular[6], prism-shaped and box-shaped finite elements. It is noted that the numerical integrations over these elements are shown to be performed analytically. Finally, we derive the matrix expression of the system equations by the Galerkin approximation and generate the concrete expression of the coefficient matrices.

2.1 Multigroup Neutron Diffusion Equation and Finite Element

In the general multigroup formalism, the neutron diffusion equation is represented by a coupled system of differential equations on the scalar flux, ϕ:

$$-\nabla D_g(\boldsymbol{r})\nabla\phi_g(\boldsymbol{r})+\Sigma_{r,g}(\boldsymbol{r})\phi_g(\boldsymbol{r}) = \sum_{\substack{g'=1\\(g'\neq g)}}^{G} \Sigma_{s,g'g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r})+\sum_{g'=1}^{G}\frac{\chi_g}{K_{\mathrm{eff}}}(\nu\Sigma_f)_{g'}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r}),$$

$$g=1,2,\cdots G, \qquad \text{for} \quad r\in\Omega, \qquad (1)$$

where the notations are defined as follows:

g	the energy group index,
ϕ_g	the flux in the g-th energy group ($\mathrm{cm^{-2}\,sec^{-1}}$),
D_g	the diffusion constant (cm),
$\Sigma_{r,g}$	the total removal cross section ($\mathrm{cm^{-1}}$),
$\Sigma_{s,g'g}$	the scattering cross section from g' into g ($\mathrm{cm^{-1}}$),
χ_g	the fission source spectrum normalized as $\sum_{g=1}^{G}\chi_g=1.0$,
K_{eff}	the effective multiplication factor,
ν_g	the average number of neutrons produced by fissions induced in group g,
$\Sigma_{f,g}$	the fission cross section ($\mathrm{cm^{-1}}$).

By denoting the external boundary of a domain Ω by $\partial_e\Omega$, the general form of boundary condition associated with Eq. (1) is described as

$$a(r)\left(D_g\frac{\partial\phi_g}{\partial n}\right)(\boldsymbol{r})+b(r)\phi_g(\boldsymbol{r}) = 0, \qquad r\in\partial_e\Omega, \qquad (2)$$

where $\partial/\partial n$ represents the outward normal derivative at $\partial_e\Omega$ with unit vector n, and $a=0$ shows the free boundary, $b=0$ the reflective boundary, and $a>0$ and $b>0$ the extrapolated boundaries. If a reactor is composed of a finite number of subdomains, each of which is characterized by a specific material property, then the bulk coefficients $D_g(\boldsymbol{r})$, $\Sigma_{r,g}(\boldsymbol{r})$, $\Sigma_{g'g}(\boldsymbol{r})$, $(\nu\Sigma_f)_g(\boldsymbol{r})$, as well as the boundary coefficients $a(r)$ and $b(r)$, are constant throughout each subdomain. In addition to Eq. (2), the usual interface conditions with respect to neutron flux and current must be satisfied. That is, if the domain Ω is composed of matrials having interfaces $\partial_i\Omega$, then the interface conditions are

$$\phi_g(\boldsymbol{r}) \quad \text{and} \quad \left(D_g\frac{\partial\phi}{\partial n}\right)(\boldsymbol{r}) \quad \text{are continuous across} \quad \partial_i\Omega. \qquad (3)$$

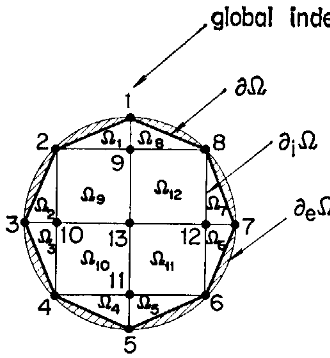


Fig. 1 Partition of a polygon into subdomains, or trianglar and rectangular elements and their global node indices.

For more explanation, the domain Ω is assumed a bounded set in the two-dimensional Euclidean space and Ω is the union of a finite number N of contiguous subdomain Ω_i what is called the finite element:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \tag{4}$$

As shown in **Fig. 1**, a two-dimensional reactor can be divided into $N=12$ triangular or rectangular subdomains. The presence of a black absorber or hole subdomain is not allowed in the interior of the reactor domain Ω. The boundary $\partial\Omega$ of Ω, which is the union of the exterior boundary $\partial_e\Omega$ and the interfaces $\partial_i\Omega$:

$$\partial\Omega = \partial_e\Omega \cup \partial_i\Omega\,, \tag{5}$$

is assumed to be piecewise rectilinear. The cross-hatched region in **Fig. 1** must be as small as possible for a good approximation to the virtual external boundary. In later use, it is defined that

$$\bar{\Omega} = \Omega \cup \partial\Omega\,. \tag{6}$$

Now to solve the generalized eigenvalue problem described by Eqs.(1)–(3) for the effective multiplication factor K_{eff} and the corresponding positive eigenfunction, $\{\phi_g(r);\ g=1, 2, \dots, G\}$ we adopt the usual outer iteration (also called the source iteration or the power iteration) procedure. That is, starting with a positive but otherwise arbitrary estimate $\{\phi_g^{(0)}\}$ for group fluxes and a positive estimated $K_{\text{eff}}^{(0)}$ for the effective multiplication factor, we generate successive estimates $\{\phi_g^{(n)}\}$ and $K_{\text{eff}}^{(n)}$, $n=1, 2, \dots,$ according to the following scheme:

$$-\nabla D_g(r)\nabla\phi_g^{(n)}(\boldsymbol{r})+\Sigma_{r,g}(\boldsymbol{r})\phi_g^{(n)}(\boldsymbol{r}) = \sum_{g'(<g)} \Sigma^d_{s,g'g}(\boldsymbol{r})\phi_{g'}^{(n)}(\boldsymbol{r})+\sum_{g'(>g)} \Sigma^u_{s,g'g}(\boldsymbol{r})\phi_{g'}^{(n-1)}(\boldsymbol{r})$$
$$+\frac{1}{K_{\text{eff}}^{(n-1)}}\chi_g(\boldsymbol{r})\sum_{g'=1}^{G}(\nu\Sigma_f)_{g'}(\boldsymbol{r})\phi_{g'}^{(n-1)}(\boldsymbol{r})\,, \quad \text{for}\quad g=1,2,\cdots G\,, \tag{7}$$

and

$$K_{\text{eff}}^{(n)} = K_{\text{eff}}^{(n-1)}\frac{\langle\phi^{(n)},\,\phi^{(n)}\rangle}{\langle\phi^{(n)},\,\phi^{(n-1)}\rangle}\,, \tag{8}$$

where $<,>$ denotes any appropriate inner product. The scattering cross section $\Sigma_{s,g'g}$ is parted into the down-scattering cross section $\Sigma^d_{s,g'g}$ $(g'<g)$ and the up-scattering cross section $\Sigma^u_{s,g'g}$ $(g'>g)$.

At each step of this outer iteration procedure, we thus solve G uncoupled self-adjoint elliptic boundary value problems of the form:

$$-\nabla D(\boldsymbol{r})\nabla\phi(\boldsymbol{r})+\Sigma(\boldsymbol{r})\phi(\boldsymbol{r})=f(\boldsymbol{r}), \qquad \text{for} \quad \boldsymbol{r}\in\Omega, \tag{9}$$

where f is a known function. The unknown ϕ is subject to the same boundary conditions, Eqs. (2) and (3). The conservation of the self-adjoint character provides the Galerkin-type approximation procedure for the solution of the elliptic boundary value problem as described in the following.

The matrix expression of Eq. (7) in the form of Eq. (9) is given by

$$[-\nabla D(\boldsymbol{r})\nabla+\Sigma(\boldsymbol{r})]\phi^{(n)}(\boldsymbol{r})=F^{(n-1)}(\boldsymbol{r}), \tag{10}$$

where

$$\Sigma=\Sigma_r-\Sigma^d, \tag{10a}$$

$$F^{(n-1)}(\boldsymbol{r})=\left[\Sigma^u(\boldsymbol{r})+\frac{1}{K_{\text{eff}}^{(n-1)}}\chi(\boldsymbol{r})S(\boldsymbol{r})\right]\phi^{(n-1)}(\boldsymbol{r}), \tag{10b}$$

$$\Sigma_r=\begin{pmatrix}\Sigma_{r,1} & & & \\ & \Sigma_{r,2} & & 0 \\ & & \ddots & \\ 0 & & & \Sigma_{r,G}\end{pmatrix}, \tag{10c}$$

$$D=\begin{pmatrix}D_1 & & & \\ & D_2 & & 0 \\ & & \ddots & \\ 0 & & & D_G\end{pmatrix}, \tag{10d}$$

$$\Sigma^d=\begin{pmatrix}0 & & & & \\ \Sigma_{s,12} & 0 & & 0 & \\ \Sigma_{s,13} & \Sigma_{s,23} & 0 & & \\ \vdots & \vdots & & \ddots & \\ \Sigma_{s,1G} & \Sigma_{s,2G} & \cdots & \Sigma_{s,G-1,G} & 0\end{pmatrix}, \tag{10e}$$

$$\Sigma^u=\begin{pmatrix}0 & \Sigma_{s,21} & \Sigma_{s,31} & \cdots & \Sigma_{s,G1} \\ & 0 & \Sigma_{s,32} & \cdots & \Sigma_{s,G2} \\ & & 0 & & \vdots \\ & 0 & & \ddots & \Sigma_{s,G,G-1} \\ & & & & 0\end{pmatrix}, \tag{10f}$$

$$\chi=(\chi_1,\chi_2,\cdots\chi_G)^T, \tag{10g}$$

$$S=((\nu\Sigma_f)_1,(\nu\Sigma_f)_2,\cdots(\nu\Sigma_f)_G)^T, \tag{10h}$$

$$\phi=(\phi_1,\phi_2,\cdots\phi_G)^T, \tag{10i}$$

in which $(\,.\,)^T$ is the transposed vector.

2.2 Galerkin-Type Approximation

We proceed here as usual on a Galerkin-type approximate procedure.[5] The symbol x denotes a point on the two-dimensional Euclidean plane. Let $L_2(\Omega)$ be the Hilbert space of functions which are square integrable over Ω. The inner product on $L_2(\Omega)$ is expressed by

$$(u,v)=\int_\Omega u(x)v(x)dx, \tag{11}$$

and the norm by

$$\|u\| = \sqrt{(u, u)}\,. \tag{12}$$

we define $D(L)$ as the set of functions in $L_2(\Omega)$, which have the following properties:

(i) they are twice continuously differentiable in Ω,

(ii) the functions and their first derivatives are continuous on $\partial\Omega$,

(iii) they satisfy the boundary conditions Eqs. (2) and (3).

Thus our problem is to find the function $u \in D(L)$ which satisfies the following equation:

$$(L+\Sigma)u = f, \qquad f \in L_2(\Omega)\,, \tag{13}$$

where we define the linear differential operator L by

$$(Lu)(x) = -\frac{\partial}{\partial x}\cdot D(x)\frac{\partial}{\partial x}u\,, \quad \text{on domain} \quad D(L)\,, \tag{13a}$$

and the multiplicative operator Σ by

$$(\Sigma u)(x) = \Sigma(x)u(x)\,, \quad \text{on} \quad L_2(\Omega)\,. \tag{13b}$$

Furthermore, let $W_2^1(\Omega)$ denote the Hilbert space of all elements of $L_2(\Omega)$ that have generalized derivatives of the first degree in $L_2(\Omega)$ (which is called Sobolev space). The inner product in $W_2^1(\Omega)$ is defined by

$$\langle u, v\rangle_{1,2} = \int_\Omega \left(uv + \frac{\partial u}{\partial x}\cdot\frac{\partial v}{\partial x}\right)dx\,, \tag{14}$$

and the norm by

$$\|u\|_{1,2} = \sqrt{\langle u, u\rangle_{1,2}}\,. \tag{15}$$

If $u \in D(L)$ and all the coefficients in Eq. (13) are smooth, then it can be shown that the solution of Eq. (13) is equivalent to find the function $u \in W_2^1(\Omega)$ which satisfies the following equations:

$$a(u, v) = (f, v)\,, \qquad \text{for all} \quad v \in W_2^1(\Omega)\,, \tag{16}$$

where

$$a(u, v) = \int_\Omega \left(D(x)\frac{\partial u}{\partial x}\cdot\frac{\partial v}{\partial x} + \Sigma(x)u(x)v(x)\right)dx + \int_{\partial_e'\Omega} \frac{b(x)}{a(x)}u(x)v(x)dS\,, \tag{16a}$$

and

$$(f, v) = \int_\Omega f(x)v(x)dx\,, \qquad \text{for any} \quad v \in W_2^1(\Omega) \tag{16b}$$

and

$$\partial_e'\Omega = \{x \mid x \in \partial_e\Omega,\, a(x) > 0\}\,. \tag{16c}$$

The expression (16) is called the weak form of Galerkin approximation.

Here we shall define the Galerkin-type approximation to the present problem. The weak form of the original problem is given by

$$a(\phi, \psi) = (F, \psi)\,, \qquad \text{for all} \quad \psi \in W_2^1(\Omega)\,, \tag{17}$$

where

$$a(\phi, \psi) = \int_\Omega [D\nabla\phi\nabla\psi + \Sigma\phi\psi]dV + \int_{\partial_e'\Omega} \frac{b}{a}\phi\psi dS\,, \tag{17a}$$

$$\partial_e'\Omega = \{r \mid r \in \partial_e\Omega, a(r) > 0\} \tag{17b}$$

and

$$(F, \psi) = \int_\Omega F\psi dV. \tag{17c}$$

Let M_N be any finite-dimensional subspace of $W_2^1(\Omega)$ as shown also in **Fig. 1**, and then our aim is to solve the following approximate problem having a unique solution $\hat{\phi}$ in M_N,

$$a(\hat{\phi}, \psi) = (F, \psi), \qquad \text{for all} \quad \psi(r) \in M_N. \tag{18}$$

With any subspace, there is a finite partition of M_N into polygonal subdomains of element shape, like triangles and/or rectangles. If we define a polynomial function $\hat{\phi}$ with degree m as an element of the subspace in each polygonal subdomain of the partition, then each polynomial can be uniquely determined by its behavior within its associated subdomain. This approximate procedure is commonly referred to as the finite element method.

In order to uniquely determine $\hat{\phi} \in M_N$ from its local behavior, we need ν data in each subdomain if a polynomial of degree m in two variables (for two-dimensional space) has ν degrees of freedom. These data can be the values of the function or of its derivatives at a certain number of points in the subdomain. Polynomials which are obtained in this manner are called interpolating polynomials or interpolants. In the following study we will assume that only function values, and no derivative values, are used to determine the polynomials. This type of interpolating polynomials is commonly referred to as the Lagrange-type polynomial. Thus, in order to uniquely determine a Lagrange interpolant over a subdomain we need ν reference points, which we shall take at the nodes of a grid over the subdomain: vertex nodes, equally spaced nodes on each of all the sides of the subdomain, and interior nodes. Since we associate here one degree of freedom with each node, the dimension of the finite-dimensional subspace M_N is precisely N or the total number of nodes in Ω.

2.3 Construction of Basis Functions

We make here a choice of a finite-dimensional subspace and construct a basis for it. With each node (global index i; $i=1, 2, \ldots N$) in Ω we now associate a basis u_i which has the minimum support. That is, u_i vanishes outside the union of the finite elements (triangles and/or rectangles) to which the i-th node belongs. Furthermore, u_i assumes the value 1 at the i-th node and the value 0 at all other nodes within its support. It is easily verified that u_i is continuous across inter-element boundaries, so $u \in M_N$ implies $u \in C(\Omega)$, in which $C(\Omega)$ means a class of functions continuous over Ω.

Now we construct the basis function $u_i(r)$. The solution $\hat{\phi}_g(r)$ of the approximate problem described by Eq. (18) will be of the form:

$$\hat{\phi}_g(\boldsymbol{r}) = \sum_{i=1}^{N} q_{gi} u_i(\boldsymbol{r}), \tag{19}$$

or the matrix expression is

$$\hat{\phi}(\boldsymbol{r}) = \boldsymbol{q}\boldsymbol{u}(\boldsymbol{r}), \tag{19a}$$

where

$$\boldsymbol{q} = \begin{pmatrix} q_{11} & q_{12} \cdots q_{1N} \\ q_{21} & q_{22} \cdots q_{2N} \\ \vdots & \vdots & \vdots \\ q_{G1} & q_{G2} \cdots q_{GN} \end{pmatrix}, \tag{19b}$$

$$\boldsymbol{u} = (u_1, u_2, \cdots u_N)^T. \tag{19c}$$

The coefficients q_{gi}, i=1, 2, . . . N, in the expansion (19) represent the values of $\hat{\phi}_g(\boldsymbol{r})$ at the node, that is,

$$\hat{\phi}_g(\boldsymbol{r}_i) = q_{gi}, \quad \text{for} \quad i = 1, 2, \cdots N. \tag{20}$$

Our goal is now to determine the basis $\boldsymbol{u}$ in terms of the values of the nodal parameter $\boldsymbol{q}$. The coefficient $\boldsymbol{q}$ is called the generalized coordinate in the finite element terminology.

We use triangle and rectangle for two-dimensional domain (prism and box for three-dimensional domain) as the shapes of the finite elements, as shown in **Fig. 2** (in **Fig. 3**). The nodal indices shown in **Figs. 2** and **3** are called the local indices written later. Their indices are numbered in counterclockwise order on the plane and the order does not generally coincide with the global indices (written as i or j).

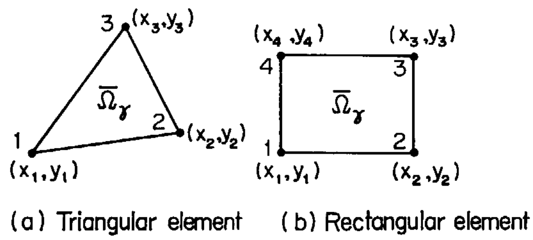


Fig. 2 Local node indices and their coordinates for two-dimensional finite elements.

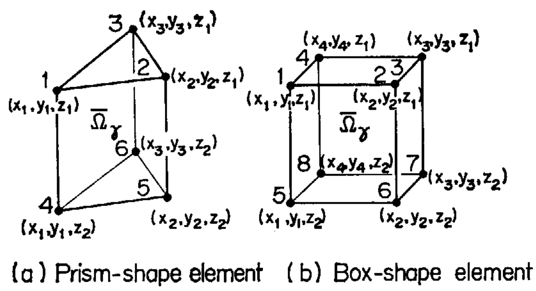


Fig. 3 Local node indices and their coordinates for three-dimensional finite elements.

The solution $\hat{\phi}^\gamma$ has the general form expressed with a polynomial of x and y for two-dimensional element (a polynomial of x, y and z for three-dimensional element):

$$\hat{\phi}_g^\gamma(x, y, z) = \sum_{k=1}^{\nu} a_{gk}^\gamma P_k(x, y, z), \tag{21}$$

or in matrix form by

$$\hat{\phi}^\gamma(x, y, z) = \boldsymbol{a}^\gamma \boldsymbol{P}(x, y, z), \tag{21a}$$

where ν is, for instance, 3 for triangular element, 4 for rectangular element, 6 for prism-shaped element, and 8 for box-shaped element if applying the first degree polynomials. Then the basis function $\boldsymbol{P}(x, y, z)$ is expressed by

$$\boldsymbol{P}(x, y, z) = \boldsymbol{P}(x, y) = \begin{cases} (1, x, y)^T, & \text{for triangular element}, \\ (1, x, y, xy)^T, & \text{for rectangular element}, \end{cases} \tag{21b}$$

$$P(x,y,z) = \begin{cases} (1,x,y,z,xz,yz)^T, & \text{for prism-shaped element}, \\ (1,x,y,z,xy,yz,zx,xyz)^T, & \text{for box-shaped element}. \end{cases} \tag{21c}$$

Here, let $\bar{\Omega}_\gamma$ be the subdomain to which the i-th node (global index) belongs and let $i(k)$ be the index corresponding to the k-th local index. On each subdomain $\bar{\Omega}_\gamma$, the basis function $\boldsymbol{u}_i$ is represented by, say $\boldsymbol{u}_i^\gamma$:

$$\boldsymbol{u}_i(x,y,z) = u_{i(k)}^\gamma(x,y,z), \qquad \text{for} \quad (x,y,z) \in \bar{\Omega}_\gamma \quad \text{and} \quad \gamma \in \Gamma_i, \tag{22}$$

where Γ_i is the set of suffixes γ on $\bar{\Omega}_\gamma$ to which the i-th node belongs. Using Eq. (22), Eq. (19) is rewritten as

$$\hat{\phi}_g^\gamma(x,y,z) = \sum_{k=1}^{\nu} q_{gk}^\gamma u_{i(k)}^\gamma(x,y,z), \tag{23}$$

or in matrix form by

$$\hat{\phi}^\gamma = \boldsymbol{q}^\gamma \boldsymbol{u}_i^\gamma. \tag{23a}$$

Now, we express the basis function $\boldsymbol{u}$ by the polynomial $\boldsymbol{P}(x, y, z)$. First, we define the following functional:

$$L_k[\hat{\phi}_g^\gamma] = \hat{\phi}_g^\gamma(x_k, y_k, z_k), \qquad \text{for} \quad k = 1, 2, \cdots \nu. \tag{24}$$

From Eq. (20), we must have the identity,

$$L_k[\hat{\phi}_g^\gamma] = q_{gk}^\gamma, \qquad \text{for} \quad k = 1, 2, \cdots \nu, \tag{25}$$

or in matrix form by

$$\boldsymbol{L}(\hat{\phi}^\gamma)^T = (\boldsymbol{q}^\gamma)^T, \tag{25a}$$

where $\boldsymbol{L}^T = (L_1, L_2, \ldots L_\nu)$.

Putting Eq. (21) into Eq. (23), we obtain

$$(\boldsymbol{q}^\gamma)^T = (\boldsymbol{L}\boldsymbol{P}^T)(\boldsymbol{a}^\gamma)^T \equiv \boldsymbol{C}(\boldsymbol{a}^\gamma)^T. \tag{26}$$

The matrix $\boldsymbol{C}$ is non-singular, so that

$$\boldsymbol{a}^\gamma = \boldsymbol{q}^\gamma(\boldsymbol{C}^{-1})^T, \tag{27}$$

where

$$\boldsymbol{C} = \boldsymbol{L}\boldsymbol{P}^T(x,y,z). \tag{27a}$$

Putting Eq. (25) into Eq. (21), the result is given by

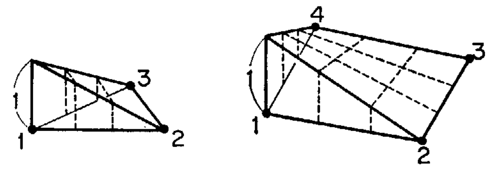
$$\hat{\phi}^\gamma = \boldsymbol{q}^\gamma(\boldsymbol{C}^{-1})^T\boldsymbol{P}. \tag{28}$$

Using Eq. (28) together with Eq. (23), we obtain finally the expression about $\boldsymbol{u}_i^\gamma$ (x, y, z) as follows:

$$u_{i(k)}^\gamma(x,y,z) = \sum_{\ell=1}^{\nu} (C^{-1})_{\ell k} P_\ell(x,y,z), \tag{29}$$

or in matrix form by

$$\boldsymbol{u}_i^\gamma(x,y,z) = (\boldsymbol{C}^{-1})\boldsymbol{P}(x,y,z). \tag{29a}$$



(a) Triangular element (b) Rectangular element

Fig. 4 Basis functions (at the first node) of linear Lagrange-type interpolant.

Here, on $\bar{\Omega}_\gamma$, $\boldsymbol{u}_i^\gamma$ is the unique Lagrange-type interpolant which has the value 1 at the i-th node and value 0 at all other nodes. Linear Lagrange-type interpolants are shown in **Fig. 4**. The concrete expressions of $\boldsymbol{u}_i^\gamma\ (x, y, z)$ for two- and three-dimensional finite elements are then represented, by writing $\boldsymbol{u}_{i(k)}^\gamma$ as $\boldsymbol{u}_k^\gamma$, as follows:

$$\left.\begin{aligned} u_{(1)}^\gamma(x,y) &= \frac{1}{J_\gamma}\{x_2y_3-x_3y_2+x(y_2-y_3)+y(x_3-x_2)\} \\ u_{(2)}^\gamma(x,y) &= \frac{1}{J_\gamma}\{x_3y_1-x_1y_3+x(y_3-y_1)+y(x_1-x_3)\} \\ u_{(3)}^\gamma(x,y) &= \frac{1}{J_\gamma}\{x_1y_2-x_2y_1+x(y_1-y_2)+y(x_2-x_1)\} \end{aligned}\right\}, \quad \text{for triangular element}, \tag{30}$$

where

$$J_\gamma = x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2) \neq 0,$$

and

$$\left.\begin{aligned} u_{(1)}^\gamma(x,y) &= \frac{1}{K_\gamma}(x-x_2)(y-y_2) \\ u_{(2)}^\gamma(x,y) &= \frac{1}{K_\gamma}(x-x_1)(y_2-y) \\ u_{(3)}^\gamma(x,y) &= \frac{1}{K_\gamma}(x-x_1)(y-y_1) \\ u_{(4)}^\gamma(x,y) &= \frac{1}{K_\gamma}(x-x_2)(y_1-y) \end{aligned}\right\}, \quad \text{for rectangular element}, \tag{30a}$$

where

$$K_\gamma = (x_2-x_1)(y_2-y_1) \not\equiv 0.$$

The expressions for the prism- and box-shaped elements are written by using the above expressions of the triangular and rectangular elements, respectively:

$$\left.\begin{aligned} u_{(k)}^\gamma(x,y,z) &= \frac{z-z_2}{z_1-z_2}u_{(k)}^\gamma(x,y) \\ u_{(k+3)}^\gamma(x,y,z) &= \frac{z_1-z}{z_1-z_2}u_{(k)}^\gamma(x,y) \end{aligned}\right\}, \quad k=1,2,3, \quad \text{for prism-shaped element}, \tag{31}$$

and

$$\left.\begin{aligned} u_{(k)}^\lambda(x,y,z) &= \frac{z-z_2}{z_1-z_2}u_{(k)}^\gamma(x,y) \\ u_{(k+4)}^\gamma(x,y,z) &= \frac{z_1-z}{z_1-z_2}u_{(k)}^\gamma(x,y) \end{aligned}\right\}, \quad k=1,2,3,4, \quad \text{for box-shaped element}. \tag{31a}$$

These four finite elements were already referred to in **Figs. 2** and **3**.

2.4 Solution Algorithm in Galerkin Approximation

We start to determine the nodal parameters $\boldsymbol{q}$ of Eq. (19) by adopting the basis u_i $(i=1, 2, \ldots N)$ constructed in the previous section as the trial function. Equation (17) is rewritten as

$$\boldsymbol{a}(\phi_g^{(n)}, u_i) = (\boldsymbol{F}^{(n-1)}, u_i) + \boldsymbol{K}_{gi}^{(n)}, \qquad \text{for} \quad i = 1, 2, \cdots N, \tag{32}$$

where n is the outer iteration index and

$$\boldsymbol{a}(\phi_g^{(n)}, u_i) = \int_\Omega [D_g \nabla \phi_g^{(n)} \cdot \nabla u_i + \Sigma_{r,g} \phi_g^{(n)} \cdot u_i] dV + \int_{\partial_e \Omega} \frac{b}{a} \phi_g^{(n)} \cdot u_i dS, \tag{32a}$$

$$\boldsymbol{K}_{gi}^{(n)} = \int_\Omega [\Sigma^d \phi^{(n)}]_g u_i dV. \tag{32b}$$

Moreover, the fission source term $\boldsymbol{F}$ in Eq. (32) is expanded with the basis u_i as follows:

$$F_g^{(n)}(x, y, z) = \sum_{i=1}^{N} f_{gi}^{(n)} u_i(x, y, z), \tag{32c}$$

or in matrix form,

$$\boldsymbol{F}^{(n)}(x, y, z) = \boldsymbol{f}^{(n)} \boldsymbol{u}(x, y, z). \tag{32d}$$

Substituting these expressions into Eq. (32), we obtain the following linear system equations for the generalized coordinates:

$$\sum_{j=1}^{N} q_{gi}^{(n)} a(u_j, u_i) = \sum_{j=1}^{N} f_{gi}^{(n-1)}(u_j, u_i) + K_{gi}^{(n)}, \qquad \text{for} \quad i = 1, 2, \cdots N, \tag{33}$$

or in matrix form,

$$(\boldsymbol{q}^{(n)} \boldsymbol{A})_{gi} = (\boldsymbol{f}^{(n-1)} \boldsymbol{B} + \boldsymbol{K}^{(n)})_{gi}, \qquad \text{for} \quad i = 1, 2, \cdots N, \tag{33a}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the following symmetric matrices:

$$\boldsymbol{A} = \begin{pmatrix} a(u_1, u_1), & a(u_1, u_2), & \cdots a(u_1, u_N) \\ a(u_2, u_1), & a(u_2, u_2), & \cdots a(u_2, u_N) \\ \vdots & \vdots & \vdots \\ a(u_N, u_1), & a(u_N, u_2), & \cdots a(u_N, u_N) \end{pmatrix}, \tag{34}$$

$$\boldsymbol{B} = \begin{pmatrix} (u_1, u_1), & (u_1, u_2), & \cdots (u_1, u_N) \\ (u_2, u_1), & (u_2, u_2), & \cdots (u_2, u_N) \\ \vdots & \vdots & \vdots \\ (u_N, u_1), & (u_N, u_2), & \cdots (u_N, u_N) \end{pmatrix}, \tag{35}$$

and $\boldsymbol{f}$ is represented by using Eq. (10b) as

$$\boldsymbol{f}^{(n-1)} = \left[\Sigma^u + \frac{1}{K_{\text{eff}}^{(n-1)}} \chi \boldsymbol{S} \right] \boldsymbol{q}^{(n-1)}. \tag{36}$$

Consequently, we obtain the following equation from Eq. (32) is to determine the nodal parameters:

$$\boldsymbol{q}_{gi}^{(n)} = \left\{ \left[\left(\Sigma^u + \frac{1}{K_{\text{eff}}^{(n-1)}} \chi \boldsymbol{S} \right) \boldsymbol{q}^{(n-1)} \boldsymbol{B} + \boldsymbol{K}^{(n)} \right] \boldsymbol{A}^{-1} \right\}_{gi}, \qquad i = 1, 2, \cdots N,$$

$$\text{for} \quad g = 1, 2, \cdots G, \tag{37}$$

and then $K_{\rm eff}$ is expressed by the following equation if Σ^u is zero,

$$K_{\rm eff}^{(n)} = K_{\rm eff}^{(n-1)} \frac{\sum_{g=1}^{G}\sum_{g'=1}^{G} \chi_g(\nu\Sigma_f)_{g'}(\hat{\phi}_g^{(n)}, \hat{\phi}_{g'}^{(n)})}{\sum_{g=1}^{G}\sum_{g'=1}^{G} \chi_g(\nu\Sigma_f)_{g'}(\hat{\phi}_g^{(n)}, \hat{\phi}_{g'}^{(n-1)})} \equiv K_{\rm eff}^{(n-1)} \frac{\boldsymbol{S}^T\boldsymbol{q}^{(n)}\boldsymbol{B}(\boldsymbol{q}^{(n)})^T\chi}{\boldsymbol{S}^T\boldsymbol{q}^{(n-1)}\boldsymbol{B}(\boldsymbol{q}^{(n)})^T\chi}\,. \tag{38}$$

The solution $\hat{\phi}$ is thus obtained through Eq. (19).

2.5 Generation of the Coefficients in the Approximate Equation

We give here the concrete expressions of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{K}$. Though the descriptions are mainly made for two-dimensional elements, the expressions for three-dimensional elements are easily derived with the help of the results for two-dimensional elements as shown afterwards.

We begin to describe $\boldsymbol{B}$. An element B_{ij} in the symmetric matrix $\boldsymbol{B}$ is given by,

$$B_{ij} = \int_\Omega u_i u_j dxdy\,. \tag{39}$$

Therefore the value of B_{ij} is non-zero only when the global node indices i and j belong to the same sub-domain Ω_γ according to the property of the basis function $\boldsymbol{u}$. Consequently B_{ij} is expressed by

$$B_{ij} = \sum_{\gamma\in\Gamma_{ij}} B^\gamma_{ij(k\ell)}\,, \tag{40}$$

where

$$B^\gamma_{ij(k\ell)} = \int_{\Omega_\gamma} u^\gamma_{i(k)} u^\gamma_{j(\ell)} dxdy \tag{40a}$$

and Γ_{ij} denotes the set of indices (γ) of the elements ($\bar{\Omega}_\gamma$) which belong to the intersection of the domains of u_i and u_j. The meaning of indices $i(k)$ and $j(\ell)$ was already described in Eq. (22).

We start on the triangular element. In order to most efficiently calculate $B^\gamma_{ij(k\ell)}$, we map the triangle Ω_γ onto a standard canonical triangle, say T_0. Let Ω_γ have the vertices, (x_1, y_1), (x_2, y_2) and (x_3, y_3), and let T_0 have the vertices (0, 0), (1, 0) and (0, 1). This mapping is easily performed by linear transformations $x=a_1+b_1\xi+c_1\eta$ and $y=a_2+b_2\xi+c_2\eta$. Then by solving these equations about the constants a_1, b_1, . . . , we obtain the following equations:

$$\begin{aligned} x &= x_1+(x_2-x_1)\xi+(x_3-x_1)\eta \\ y &= y_1+(y_2-y_1)\xi+(y_3-y_1)\eta\,, \end{aligned} \tag{41}$$

and the inverse mapping is given by

$$\begin{aligned} \xi &= \frac{1}{J_\gamma}\{(-x_1y_3+x_3y_1)+(y_3-y_1)x+(x_1-x_3)y\} \\ \eta &= \frac{1}{J_\gamma}\{(x_1y_2-x_2y_1)+(y_1-y_2)x+(x_2-x_1)y\}\,, \end{aligned} \tag{42}$$

where

$$J_\gamma = \begin{vmatrix} \dfrac{\partial x}{\partial \xi}, & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi}, & \dfrac{\partial y}{\partial \eta} \end{vmatrix} = (x_1-x_2)(y_1-y_3)-(x_1-x_3)(y_1-y_2)\,, \tag{42a}$$

which is the Jacobian or the functional determinant.

After the linear transformation (written with tilde ~), we obtain the following equations:

$$\tilde{C} = \begin{pmatrix} 1 & \xi_1 & \eta_1 \\ 1 & \xi_2 & \eta_2 \\ 1 & \xi_3 & \eta_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \tag{43}$$

$$\tilde{C}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \tag{44}$$

and

$$P = \begin{pmatrix} 1 \\ \xi \\ \eta \end{pmatrix}. \tag{45}$$

Accordingly, the basis function $\tilde{u}^{\tau}_{i(k)}\ (\xi, \eta)$ corresponding to Eq. (29) is given by

$$\tilde{u}_{i(k)}(\xi, \eta) = a_k + b_k\xi + c_k\eta\,, \qquad \text{for} \quad k = 1, 2, 3\,, \tag{46}$$

where

$$\begin{aligned} a_1 &= 1\,, & b_1 &= -1\,, & c_1 &= -1\,, \\ a_2 &= 0\,, & b_2 &= 1\,, & c_2 &= 0\,, \\ a_3 &= 0\,, & b_3 &= 0\,, & c_3 &= 1\,. \end{aligned} \tag{46a}$$

By using these expressions, Eq. (40) leads to the following expression:

$$\begin{aligned} B^{\tau}_{ij(k\ell)} &= \int_{\Omega_\tau} u^{\tau}_{i(k)} u^{\tau}_{j(\ell)} dxdy \\ &= \int_{T0} \tilde{u}_{i(k)} \tilde{u}_{j(\ell)} |J_\tau| d\xi d\eta \\ &= \int_0^1 d\xi \int_0^{1-\xi} \tilde{u}_{i(k)} \tilde{u}_{j(\ell)} |J_\tau| d\eta \\ &= \frac{1}{24} |(x_1-x_2)(y_1-y_3)-(x_1-x_3)(y_1-y_2)| \\ &\quad \times (12a_k a_\ell + 2b_k b_\ell + 2c_k c_\ell + 4\overline{a_k b_\ell} + \overline{b_k c_\ell} + 4\overline{c_\ell a_k})\,, \qquad \text{for} \quad k, \ell = 1, 2, 3\,, \end{aligned} \tag{47}$$

where $\overline{a_k b_\ell} = a_k b_\ell + a_\ell b_k$.

Thus, after some algebraic maniplations we obtain the final expression about $\boldsymbol{B}$ as follows:

$$B^{\tau} = \frac{J_\tau}{24} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \qquad \text{for triangular element}\,, \tag{48}$$

where J_τ was already defined by Eq. (42a). Similarly, we obtain

$$B^{\tau} = \frac{K_\tau}{36} \begin{pmatrix} 4 & 2 & 1 & 2 \\ & 4 & 2 & 1 \\ \text{symmet.} & & 4 & 2 \\ & & & 4 \end{pmatrix}, \qquad \text{for rectangular element}\,, \tag{49}$$

where $K_\tau = (x_1 - x_2)\ (y_1 - y_2)$.

Next, we derive the expressions for three-dimensional elements. Since the three-dimensional interpolant Eq. (31) is factorized into the axial and planar components:

$$u^r_{i(k)}(x,y,z)=f_{i(k)}(z)u^r_{i(k)}(x,y)\,, \tag{50}$$

the expressions B^r corresponding to Eqs. (48) and (49) for the three-dimensional elements are represented by

$$\boldsymbol{B}^r_{ij(k\ell)}=\int_{z_1}^{z_2}f_{i(k)}(z)f_{j(\ell)}(z)dz\cdot\int_{\Omega_r}u^r_{i(k)}(x,y)u^r_{j(\ell)}(x,y)dxdy\,. \tag{51}$$

As the result of planar integral in the right hand side of Eq. (51) has already been obtained as Eq. (48) or (49), the concrete expressions of Eq. (51) are easily written as follows:

$$\boldsymbol{B}^r=\frac{J'_r}{144}\begin{pmatrix}\boldsymbol{U} & \boldsymbol{V}\\ \boldsymbol{V} & \boldsymbol{U}\end{pmatrix},\qquad \text{for prism-shaped element}\,, \tag{52}$$

where

$$\boldsymbol{U}=\begin{pmatrix}4 & 2 & 2\\ & 4 & 2\\ \text{symmet.} & & 4\end{pmatrix}, \tag{52a}$$

$$\boldsymbol{V}=\begin{pmatrix}2 & 1 & 1\\ & 2 & 1\\ \text{symmet.} & & 2\end{pmatrix}. \tag{52b}$$

and

$$J'_r=(z_2-z_1)\cdot J_r\,. \tag{52c}$$

$$\boldsymbol{B}^r=\frac{K'_r}{216}\begin{pmatrix}\boldsymbol{U}' & \boldsymbol{V}'\\ \boldsymbol{V}' & \boldsymbol{U}'\end{pmatrix},\qquad \text{for box-shaped element}\,, \tag{53}$$

where

$$\boldsymbol{U}'=\begin{pmatrix}8 & 4 & 2 & 4\\ & 8 & 4 & 2\\ & & 8 & 4\\ \text{symmet.} & & & 8\end{pmatrix}, \tag{53a}$$

$$\boldsymbol{V}'=\begin{pmatrix}4 & 2 & 1 & 2\\ & 4 & 2 & 1\\ & & 4 & 2\\ \text{symmet.} & & & 4\end{pmatrix}, \tag{53b}$$

and

$$K'_r=(z_2-z_1)K_r\,. \tag{53c}$$

The expressions of J_r and K_r in the above equations were already described in Eqs. (42a) and (49), respectively.

We now describe the concrete expression of the symmetric matrix $\boldsymbol{A}$, Eq. (34) The matrix element is expressed with the help of Eq. (32a) as

$$A_{ij}=\int_{\Omega}[D\nabla u_i\cdot\nabla u_j+\Sigma_r u_i u_j]dxdy\,, \tag{54}$$

where the last term for the boundary condition on the right hand side of Eq. (32a) is excluded because Eq. (32a) can be solved for the generalized coordinate $\boldsymbol{q}$ for the natural boundary conditions since they are satisfied automatically in the Galerkin approximation. For the non-natural conditions, the q's must be constrained to satisfy the conditions.

The value of A_{ij} is non-zero only if the domain to which u_i and u_j belong is common on the finite element. By assuming that reactor parameters D, Σ_r, . . . are constant within a finite element, Eq. (54) is rewritten as

$$A_{ij} = \sum_{\gamma \in \Gamma_{ij}} A^{\gamma}_{ij(k\ell)}\,, \tag{55}$$

$$A^{\gamma}_{ij(k\ell)} = DQ^{\gamma}_{ij(k\ell)} + \Sigma_r B^{\gamma}_{ij(k\ell)} \tag{56}$$

and

$$Q^{\gamma}_{ij(k\ell)} = \int_{\Omega_\gamma} \nabla u^{\gamma}_{i(k)} \nabla u^{\gamma}_{j(\ell)} dxdy\,. \tag{57}$$

The values of $B^{\gamma}_{ij(k\ell)}$ for triangular, rectangular, prism and box finite elements were already given by Eqs. (48), (49), (52) and (53), respectively.
The $Q^{\gamma}_{k\ell}$'s (writing $Q^{\gamma}_{ij(k\ell)}$ as $Q^{\gamma}_{k\ell}$ for simplicity) are obtained through the linear transformation in a way similar to obtaining $B^{\gamma}_{ij(k\ell)}$.

For the two-dimensional triangular element, we use the identity:

$$\begin{aligned}\frac{\partial}{\partial x} &= \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \\ &= \frac{1}{J_\gamma}\left[(y_3-y_1)\frac{\partial}{\partial \xi} + (y_1-y_2)\frac{\partial}{\partial \eta}\right]\end{aligned} \tag{58}$$

and

$$\frac{\partial}{\partial y} = \frac{1}{J_\gamma}\left[(x_1-x_3)\frac{\partial}{\partial \xi} + (x_2-x_1)\frac{\partial}{\partial \eta}\right]. \tag{58b}$$

Thus we obtain

$$\begin{aligned}Q^{\gamma}_{k\ell} &= \frac{1}{J^2_\gamma}\int_{T_0}[\{b_k(y_3-y_1)+c_k(y_1-y_2)\}\cdot\{b_\ell(y_3-y_1)+c_\ell(y_1-y_2)\} \\ &\quad + \{(b_k(x_1-x_3)+c_k(x_2-x_1)\}\cdot\{b_\ell(x_1-x_3)+c_\ell(x_2-x_1)\}]J_\gamma d\xi d\eta \\ &= \frac{1}{J_\gamma}W_{k\ell}\int_0^1 d\xi\int_0^{1-\xi} d\eta = \frac{1}{2J_\gamma}W_{k\ell}\,, \qquad \text{for } k, \ell = 1, 2, 3\,,\end{aligned} \tag{59}$$

where

$$\begin{aligned}W_{k\ell} &= b_k b_\ell[(y_3-y_1)^2+(x_1-x_3)^2] + c_k c_\ell[(y_1-y_2)^2+(x_2-x_1)^2] \\ &\quad + b_k c_\ell[(y_3-y_1)(y_1-y_2)+(x_1-x_3)(x_2-x_1)]\,,\end{aligned} \tag{59a}$$

or by rearranging, we get

$$Q_\gamma = \frac{1}{2J_\gamma}\begin{pmatrix} \sum_\alpha(\alpha_2-\alpha_3)^2, & -\sum_\alpha(\alpha_1-\alpha_3)(\alpha_2-\alpha_3), & -\sum_\alpha(\alpha_1-\alpha_2)(\alpha_3-\alpha_2) \\ & \sum_\alpha(\alpha_3-\alpha_1)^2, & -\sum_\alpha(\alpha_3-\alpha_1)(\alpha_2-\alpha_1) \\ \text{symmetric} & & \sum_\alpha(\alpha_1-\alpha_2)^2 \end{pmatrix},$$

$$\text{for triangular element}\,, \tag{60}$$

where $\sum_\alpha$ denotes the summation over x and y like

$$\sum_\alpha(\alpha_2-\alpha_3)^2 = (x_2-x_3)^2+(y_2-y_3)^2\,.$$

Similarly, we obtain the expression for the rectangular element as follows:

$$Q^r = \frac{1}{6K_r}\begin{pmatrix} 2X+2Y, & X-2Y, & -X-Y, & -2X+Y \\ & 2X+2Y, & -2X+Y, & -X-Y \\ & & 2X+2Y, & X-2Y \\ \text{symmetric} & & & 2X+2Y \end{pmatrix}, \quad \text{for rectangular element} \tag{61}$$

where $X=(x_2-x_1)^2$ and $Y=(y_2-y_1)^2$.

We can obtain $\boldsymbol{Q}^r$ for the three-dimensional elements in a similar manner to $\boldsymbol{B}^r$. The $Q^r_{k\ell}$ is given by

$$\begin{aligned} Q^r_{k\ell} &= \int_{z_1}^{z_2} dz f_{i(k)}(z) f_{j(\ell)}(z) \cdot \int_{\Omega_r}\left[\frac{\partial u^r_{i(k)}(x,y)}{\partial x}\frac{\partial u^r_{j(\ell)}(x,y)}{\partial x} + \frac{\partial u^r_{i(k)}(x,y)}{\partial y}\frac{\partial u^r_{j(\ell)}(x,y)}{\partial y}\right]dxdy \\ &+ \int_{z_1}^{z_2} dz \frac{df_{i(k)}(z)}{dz}\frac{df_{j(\ell)}(z)}{dz}\int_{\Omega_r} u^r_{i(k)}(x,y)u^r_{j(\ell)}(x,y)dxdy \\ &= Q^{r(2)}_{k\ell}\int_{z_1}^{z_2} f_{i(k)}(z) f_{j(\ell)}(z)dz + B^{r(2)}_{k\ell}\int_{z_1}^{z_2}\frac{df_{i(k)}}{dz}\frac{df_{j(\ell)}}{dz}dz\,, \end{aligned} \tag{62}$$

where $Q^{r(2)}_{k\ell}$ and $B^{r(2)}_{k\ell}$ are respectively $Q^r_{k\ell}$ and $B^r_{k\ell}$ for the two-dimensional elements already described.

Consequently, we have the following concrete expressions for the three-dimensional elements:

$$\boldsymbol{Q}^r = \frac{1}{24J_r'}\begin{pmatrix} \boldsymbol{U} & \boldsymbol{V} \\ \boldsymbol{V} & \boldsymbol{U} \end{pmatrix}, \quad \text{for prism-shaped element}, \tag{63}$$

where

$$\boldsymbol{U} = \begin{pmatrix} 4ZQ_{11}+2J_r^2, & 4ZQ_{12}+J_r^2, & 4ZQ_{13}+J_r^2 \\ & 4ZQ_{22}+2J_r^2, & 4ZQ_{23}+J_r^2 \\ \text{symmetric} & & 4ZQ_{33}+2J_r^2 \end{pmatrix}, \tag{63a}$$

$$\boldsymbol{V} = \begin{pmatrix} 2ZQ_{11}-2J_r^2, & 2ZQ_{12}-J_r^2, & 2ZQ_{13}-J_r^2 \\ & 2ZQ_{22}-2J_r^2, & 2ZQ_{23}-J_r^2 \\ \text{symmetric} & & 2ZQ_{33}-2J_r^2 \end{pmatrix}, \tag{63b}$$

and $Z=(z_2-z_1)^2$, and Q_{ij} $(i, j=1, 2, 3)$ is given by Eq. (60) without the factor $1/(2J_r)$.

In addition,

$$\boldsymbol{Q}^r = \frac{1}{36K_r'}\begin{pmatrix} \boldsymbol{U}' & \boldsymbol{V}' \\ \boldsymbol{V}' & \boldsymbol{U}' \end{pmatrix}, \quad \text{for box-shaped element},$$

where

$$\boldsymbol{U}' = \begin{pmatrix} 4Z(X+Y)+4XY, & 2Z(X-2Y)+2XY, & -2Z(X+Y)+XY, & -2Z(2X-Y)+2XY \\ & 4Z(X+Y)+4XY, & -2Z(2X-Y)+2XY, & -2Z(X+Y)+XY \\ & & 4Z(X+Y)+4XY, & 2Z(X-2Y)+2XY \\ \text{symmetric} & & & 4Z(X+Y)+4XY \end{pmatrix}, \tag{64a}$$

$$\boldsymbol{V}' = \begin{pmatrix} 2Z(X+Y)-4XY, & Z(X-2XY)-2XY, & -Z(X+Y)-XY, & -Z(2X-Y)-2XY \\ & 2Z(X+Y)-4XY, & -Z(2X-Y)-2XY, & -Z(X+Y)-XY \\ & & 2Z(X+Y)-4XY, & Z(X-2Y)-2XY \\ \text{symmetric} & & & 2Z(X+Y)-4XY \end{pmatrix}, \tag{64b}$$

where $X=(x_2-x_1)^2$, $Y=(y_2-y_1)^2$ and $Z=(z_2-z_1)^2$.

With these expressions $\boldsymbol{B}^r$ and $\boldsymbol{Q}^r$, we obtain $\boldsymbol{A}^r$ from

$$A^{\gamma} = DQ^{\gamma} + \Sigma_r B^{\gamma} \tag{56a}$$

Finally we describe the scattering source matrix $\boldsymbol{K}$ in Eq. (37). We start from the two-dimensional elements. The matrix element K_{gi} is written as

$$\begin{aligned} K_{gi} &= \int_{\Omega_\gamma} (\Sigma^d \hat{\phi})_g u_i dxdy \\ &= \sum_{g'=1}^{g-1} \int_{\Omega_\gamma} \Sigma^d_{g'g}(x, y) \hat{\phi}_g \cdot u_i dxdy \\ &= \sum_{g'=1}^{g-1} \sum_{j=1}^{N} q_{g'j} \int_{\Omega_\gamma} \Sigma^d_{g'g}(x, y) u_i u_j dxdy \\ &= \sum_{g'=1}^{g-1} \sum_{j=1}^{N} \Sigma^d_{g'g} q_{g'j} B_{ji} , \end{aligned} \tag{65}$$

where

$$\begin{aligned} B_{ij} &= \int_{\Omega_\gamma} u_i u_j dxdy \\ &= \sum_{\gamma \in \Gamma_{ij}}{}' B^{\gamma}_{ij(k\ell)} . \end{aligned} \tag{65a}$$

Thus we obtain the following matrix expression:

$$(\boldsymbol{K})_{gi} = (\Sigma^d \boldsymbol{q} \boldsymbol{B}^{\gamma})_{gi} , \qquad \text{for two-dimensional elements} , \tag{65b}$$

and the concrete expressions of $\boldsymbol{B}^{\gamma}$ were already described by Eqs. (48) and (49).

We obtain the same expression for the three-dimensional elements in a similar way as for the two-dimensional elements,

$$K_{gi} = \sum_{g'=1}^{g-1} \sum_{j=1}^{N} \Sigma^d_{g'g} q_{g'j} B_{ji} , \tag{66}$$

where

$$B_{ji} = \int_{\Omega_\gamma} u_j(x, y, z) u_i(x, y, z) dxdydz . \tag{66a}$$

Thus,

$$(\boldsymbol{K})_{gi} = (\Sigma^d \boldsymbol{q} \boldsymbol{B}^{\gamma})_{gi} . \tag{66b}$$

where the concrete expressions of $\boldsymbol{B}^{\gamma}$ wer already described by Eqs. (52) and (53).

3. Three-Dimensional Computer Program FEM-BABEL

The program is all written in the FORTRAN-IV language for implementing on the FACOM 230/75 operating system. Significant features are summarized as follows;

(i) Arbitrary combination of the prism-and box-shaped elements is adopted for simulating reactor geometry. Use of two types of the elements will give more geometrical flexibility and save the computer storage by taking account of the symmetry appropriate to the geometry.

(ii) Successive over-relaxation (SOR) method is adopted for solving the system equation having a large and sparse coefficient matrix. Taking advantage of the feature (i), the data transmission is performed on each *x-y* plane and then the point SOR is applied successively to the plane.

(iii) Inner iterations are accelerated by using the coarse mesh rebalancing technique and the power iterations to solve eigenvalue problems are accelerated by adopting the extrapolation by SOR.

(iv) Use of free field FIDO input form, complete restarting procedure, automatic mesh generation routine and so on will give users a help to prepare the input data more easily.

(v) Any down-scattering of neutrons is allowed, but up-scattering and region-dependent fission spectrum are not permitted.

(vi) Free and reflective boundary conditions can be imposed but logarithmic boundary condition can not be.

(vii) FEM-BABEL has special mesh generator program for PWR calculations.[28]

3.1 Solution of the System Equation by the Relaxation Method

In this section, we derive the matrix expression of the total system and then describe the equation two-dimensionally because of the use of the prism-and box-shaped elements. That is, it is shown that the equation is solved by applying SOR successively to $x-y$ planar layers.

The system equation to be solved is rewritten from the result of Galerkin approximation in the following form:

$$[H]^g \cdot \phi^g = S^g, \qquad \text{for} \quad g = 1, 2, \cdots G, \tag{67}$$

where

$$[H]^g = D^g[Q]^\gamma + \Sigma_r^g[B]^\gamma \quad (\text{cf. Eq. (56)}), \tag{67a}$$

$$S^g = \frac{1}{K_{\text{eff}}}\chi^g \sum_{g'} [F]^{\gamma,g'}\phi^{g'} + \sum_{g'} [K]^{\gamma,g'g}\phi^{g'}, \tag{67b}$$

$$[F]^{\gamma,g'} = (\nu\Sigma_f)^{g'}[B]^\gamma, \tag{67c}$$

$$[K]^{\gamma,g'g} = \Sigma_s^{g'g}[B]^\gamma, \tag{67d}$$

and $[Q]^\gamma$ and $[B]^\gamma$ are the integrals over finite elements, Q^γ and B^γ, already shown in Chapter 2 (see also the following Eqs. (68) and (68a)) and γ is the supercript which shows the prism- or box-shaped element.

The local node indices of both the elements are ordered on $x-y$ planar layers as shown in **Fig. 5.** Since the three-dimensional $[Q]^\gamma$ and $[B]^\gamma$ have already been described by two-dimensional ones in

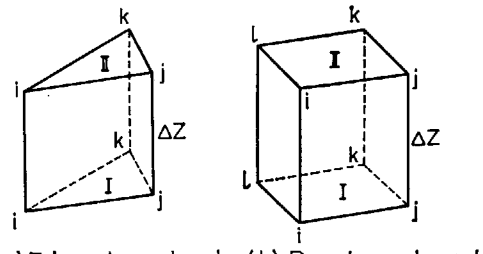


Fig. 5 Local node indices on x-y planar layers.

Chapter 2, the matrices in Eq. (67) are written by the submatrices having planar indices:

$$[\boldsymbol{Q}]^{r} = \begin{pmatrix} Q_I & P_I \\ P_I & Q_I \end{pmatrix} \tag{68}$$

and

$$[\boldsymbol{B}]^{r} = \begin{pmatrix} B_I & D_I \\ D_I & B_I \end{pmatrix}. \tag{68a}$$

Consequently we obtain the expressions for $[\boldsymbol{H}]$, $[\boldsymbol{F}]$ and $[\boldsymbol{K}]$ as follows:

$$[\boldsymbol{H}] = \begin{pmatrix} A_I & C_I \\ C_I & A_{II} \end{pmatrix}, \tag{69}$$

$$[\boldsymbol{F}] = \begin{pmatrix} F_I & G_I \\ G_I & F_{II} \end{pmatrix}, \tag{69a}$$

and

$$[\boldsymbol{K}] = \begin{pmatrix} S_I & R_I \\ R_I & S_{II} \end{pmatrix}, \tag{69b}$$

where

$$\begin{aligned} A_I &= A_{II} = DQ_I + \Sigma_r B_I, \\ C_I &= DP_I + \Sigma_r D_I, \\ F_I &= F_{II} = (\nu\Sigma_f)B_I, \\ G_I &= (\nu\Sigma_f)D_I, \\ S_I &= S_{II} = \Sigma_S B_I, \\ R_I &= \Sigma_S D_I, \end{aligned} \tag{69c}$$

In addition, Q_I, P_I, B_I and D_I are expressed by two-dimensional submatrices as shown in Chapter 2:

$$P_I = \Delta z \cdot \boldsymbol{Q}^r - \frac{1}{\Delta z} \cdot \boldsymbol{B}^r, \tag{70}$$

$$Q_I = 2 \cdot \Delta z \boldsymbol{Q}^r + \frac{1}{\Delta z} \cdot \boldsymbol{B}^r, \tag{70a}$$

$$B_I = 2 \cdot D_I, \tag{70b}$$

$$D_I = \frac{1}{6} \cdot \Delta z \cdot \boldsymbol{B}^r, \tag{70c}$$

where $\boldsymbol{B}^r$ and $\boldsymbol{Q}^r$ are the two-dimensional ones shown in Eqs. (48) (49), (60) and (61).

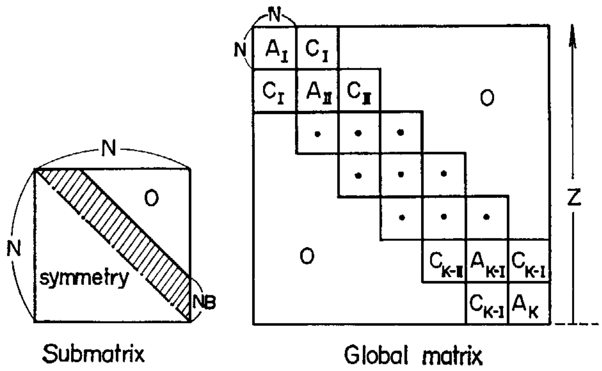


Fig. 6 Structure of the global matrix.

Our ultimate object is to solve Eq. (67). Here we investigate the property of the coefficient matrices. The submatrices are symmetric banded matrices as follows:

$$[A] = A(NB, N)\,, \tag{71}$$

$$[C] = C(NB, N)\,, \tag{71a}$$

where NB and N stand respectively for the half band width and the total dimension as shown in **Fig. 6.**

The global matrix $[\boldsymbol{H}]$ is composed of these submatrices and has the tridiagonal structure shown also in **Fig. 6.** Considering the structure and the rather large dimensions of the global matrix and the submatrices, it is pertinent to solve Eq. (67) by the successive over-relaxation method (SOR)[34] that has less storage limitations to computers.[25]

The inner iterations with SOR are performed in the core memory successively for each planar layer. That is, if t is the inner iteration index, i and j nodal indices and I plane index,

$$\phi_{i,I}^{(t+1)} = \phi_{i,I}^{(t)} + \beta(\phi^* - \phi_{i,I}^{(t)})\,, \tag{72}$$

where

$$\phi^* = A_I(1, i)^{-1}(S_{i,I} - X_i - Y_i - Z_i)\,, \tag{72a}$$

$$X_i = \sum_{j=2}^{NB} A_I(j, L_j)\phi_{L_j,I}^{(t+1)} + \sum_{j=2}^{NB} A_I(j, i)\phi_{M_j,I}^{(t)}\,, \tag{72b}$$

$$Y_i = \sum_{j=2}^{NB} C_{I-1}(j, L_j)\phi_{L_j,I-1}^{(t+1)} + \sum_{j=1}^{NB} C_{I-1}(j, i)\phi_{M_j,I-1}^{(t+1)}\,, \tag{72c}$$

$$Z_i = \sum_{j=2}^{NB} C_I(j, L_j)\phi_{L_j,I+1}^{(t)} + \sum_{j=1}^{NB} C_I(j, i)\phi_{M_j,I+1}^{(t)}\,, \tag{72d}$$

$$\text{for} \quad L_j = i-(j-1) \quad (L_j \geq 1)\,, \qquad M_j = i+(j-1) \quad (M_j \leq N)\,,$$

and $\phi_{L_j,I}^{(t)}$ is the neutron flux for node L_j within plane I at t-th iteration. The β is the relaxation factor and it is approved by the Ostrowski theorem that the iterations converge for $0<\beta<2$.[34]

3.2 Acceleration Techniques and Convergence Criteria

The outer iterations are accelerated by the extrapolation of the SOR because of less storage requirements for computers. If P_i is the fission source at a nodal point i and P^* the value normalized by the eigenvalue (K_{eff}), we obtain

$$P_i = \frac{1}{K_{\text{eff}}} P_i^* \tag{73}$$

where

$$K_{\text{eff}} = \int_{\text{reactor}} P_i^* \, dV \quad \text{and} \quad P_i^* = \sum_{g=1}^{G} (\nu\Sigma_f)_g \phi_i^g$$

If n is the outer iteration index and $(P'_i)^{(n+1)}$ the point fission source calculated by the n-th flux, then we obtain

$$P_i^{(n+1)} = P_i^{(n)} + \beta_s((P_i')^{(n+1)} - P_i^{(n)}), \tag{74}$$

and by calculating

$$K_{\text{eff}}^{(n+1)} = \int P_i^{(n+1)} dV,$$

we get finally

$$(P_i^*)^{(n+1)} = \frac{1}{K_{\text{eff}}^{(n+1)}} \cdot P_i^{(n+1)}.$$

The $(P_i^*)^{(n+1)}$ is the extrapolated fission source for the next outer iteration. This acceleration is adopted after the 4-th outer iteration and continues until the effect of acceleration satisfies the criterion ERR$\leq 10\times$ EPS1, where ERR is $|K_{\text{eff}}^{(n)} - K_{\text{eff}}^{(n+1)}|/K_{\text{eff}}^{(n+1)}$ and EPS1 the input for the outer interation criterion written later in Section 3.7.

The inner iterations are accelerated by the coarse mesh rebalancing technique.[35] This technique is adopted neither for the one coarse mesh region nor for the case when satisfying the following condition. That is, let K be the number of the coarse mesh regions and f_k the rebalancing factor for the region k, then the condition is given by

$$|1.0 - f_{\max}| \leq 0.01, \tag{75}$$

where $f_{\max} = \max\{f_1, f_2, \cdots, f_K\}$

The algorithm of the coarse mesh rebalancing technique is as follows. If ψ_i is the accelerated point flux, it is given by

$$\psi_i = \phi_i \sum_{k=1}^{K} f_k R_k, \qquad \text{for} \quad 1 \leq i \leq N, \tag{76}$$

where

$$R_k = \begin{cases} 1, & \text{for } i \in k, \\ 0, & \text{for } i \notin k. \end{cases}$$

The rebalancing factor $\{f\}$ is calculated as follows. The weight function $\{w\}$ is defined by

$$W_j = \begin{cases} \phi_j, & \text{for } j \in k, \\ 0, & \text{for } j \notin k, \end{cases} \qquad \text{for} \quad 1 \leq j \leq N. \tag{77}$$

Multiplying Eq. (67) by W_j, putting the resulting residual to zero and substituting Eq. (76) into the obtained equation, we get

$$[H]^*\{f\} = \{S\}^*, \tag{78}$$

where

$$H_{kl}^{*}=\sum_{i}\sum_{j}\phi_i H_{ij}\phi_j\,,\qquad \text{for}\quad i\in k, j\in \ell\,, \tag{78a}$$

and

$$S_k^{*}=\sum_{i}\phi_i S_i\,. \tag{78b}$$

Equation (78) is solved by the direct method (Gaussian elimination)[36].

Determination of the band width is also important for solving Eq. (67), because it affects the computer storage and computation time. If the nodal indices are numbered on plane as shown in **Fig. 5-(b)** for instance, then the half band width NBAND (=NB of Eq. (71)) is determined by

$$\text{NBAND}=\max\{B_1, B_2, \cdots, B_n, \cdots, B_{\text{NELEM}}\}\,, \tag{79}$$

where NELEM is the number of finite elements on the plane and

$$B_n=\max\{\alpha_i, \alpha_j, \alpha_k, \alpha_\ell\}\,, \tag{79a}$$

in which

$$\alpha_i=|i-j|+1\,,\qquad \text{for cyclic } i, j, k \text{ and } \ell\,. \tag{79b}$$

If the user mistakes input for NBAND, the edited print for input data messages the correct NBAND calculated by the above algorithm.

Finally we describe the convergence criteria. The program recognizes and stops when the following two convergence criteria are satisfied. The inner iterations continue until the number of iterations reaches a given inner maximum (NIMAX) or the following criterion is satisfied in SOR iteration t for an energy group:

$$\left|\frac{\phi^{(t+1)}-\phi^{(t)}}{\phi^{(t)}}\right|\leqq \text{EPS 2 (input)}\,. \tag{80}$$

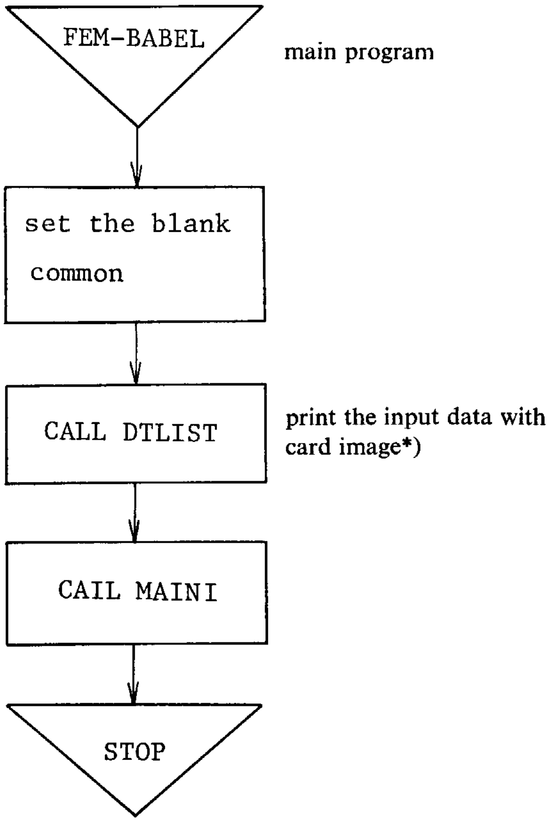
After the above conditions are satisfied, the calculation is transmitted to next energy group.

On the other hand the outer iterations continue until the number of iterations reaches a given maximum (NOMAX) or the following criterion is satisfied for the eigenvalue K_{eff} in the outer iteration n,

$$\left|\frac{K_{\text{eff}}^{(n+1)}-K_{\text{eff}}^{(n)}}{K_{\text{eff}}^{(n+1)}}\right|\leqq \text{EPS 1 (input)}\,. \tag{80b}$$

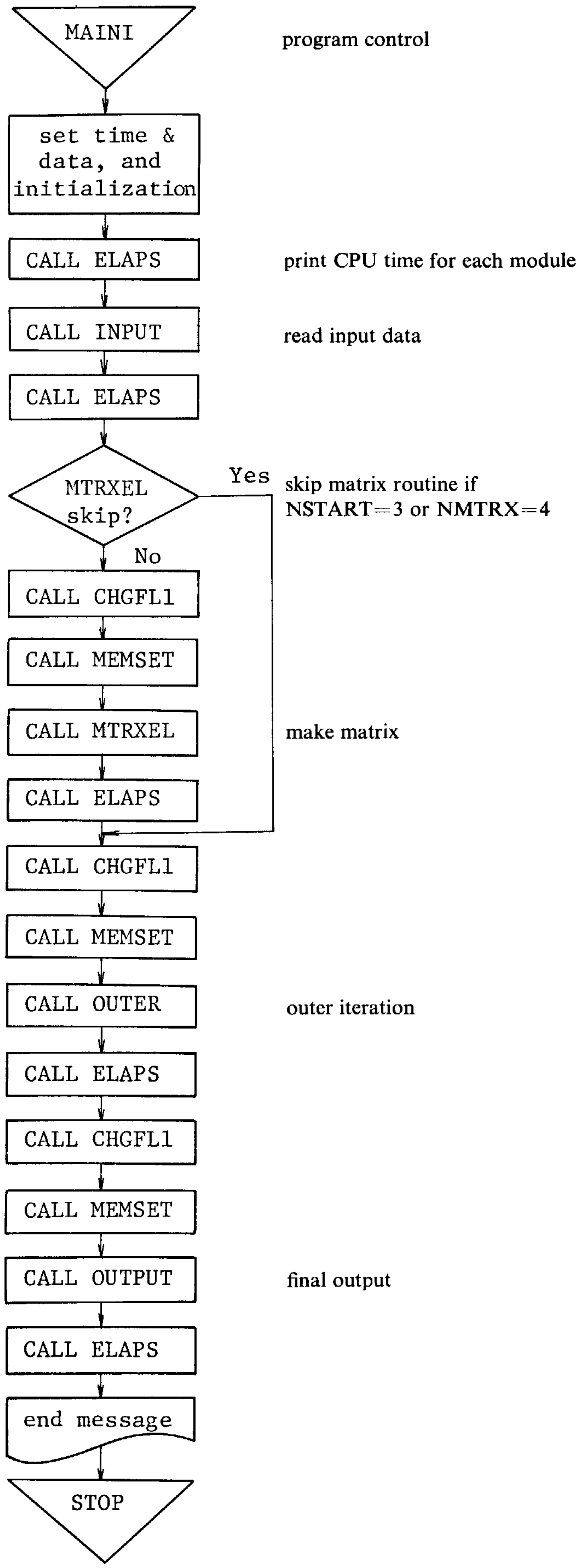
3.3 Flow Diagrams of Programs

i) FEM-BABEL

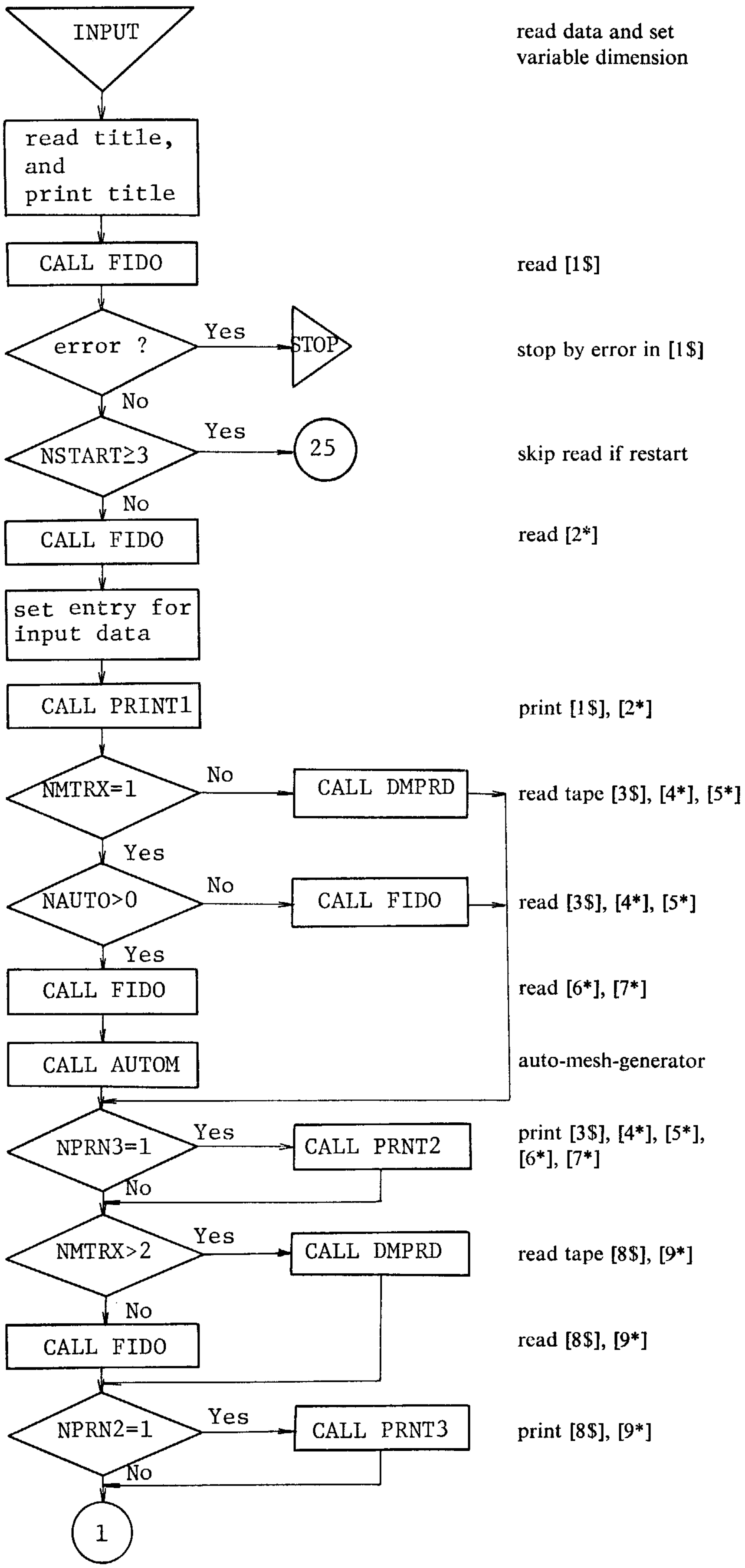


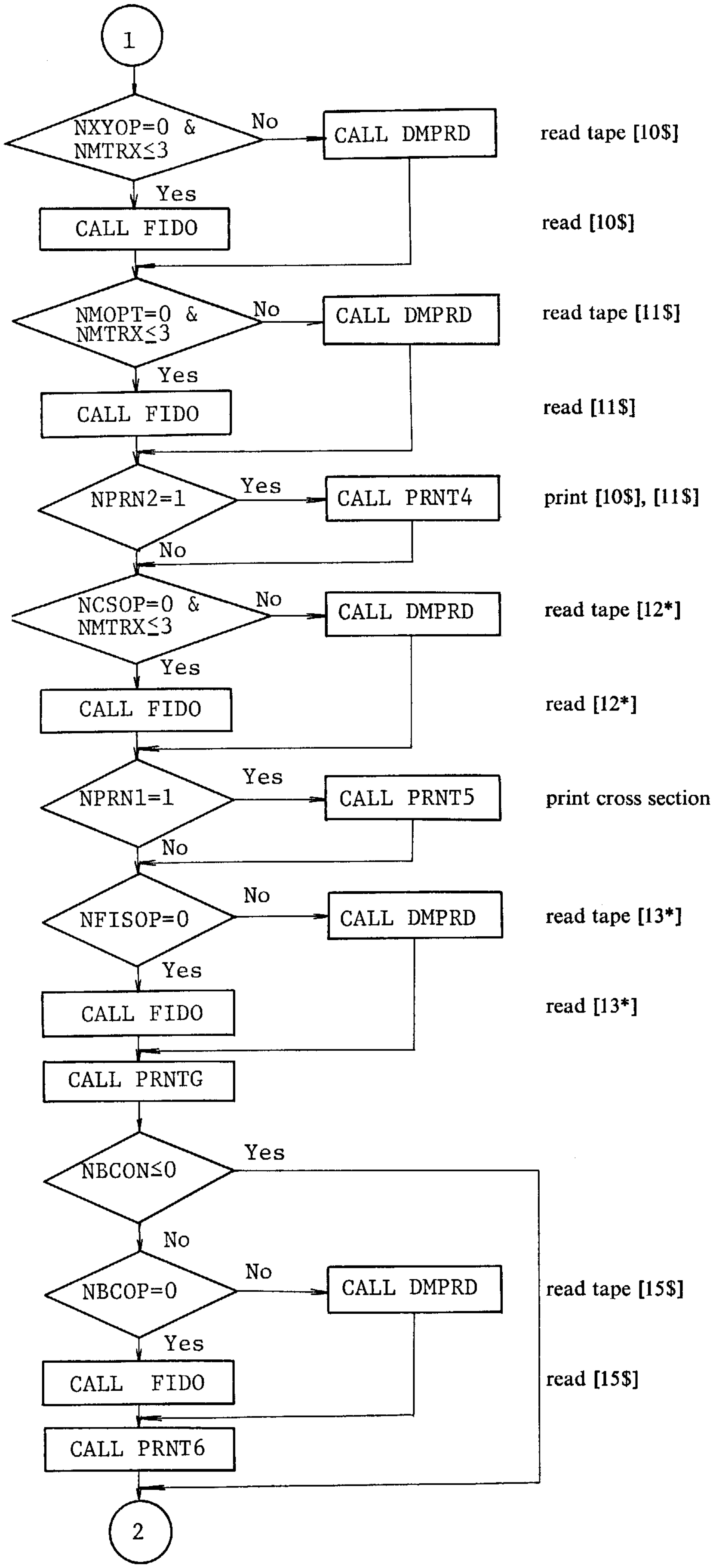
*) The program which prints the data with the very card image after reading, was programed by H. Ryufuku and registered already in JSSL[37].

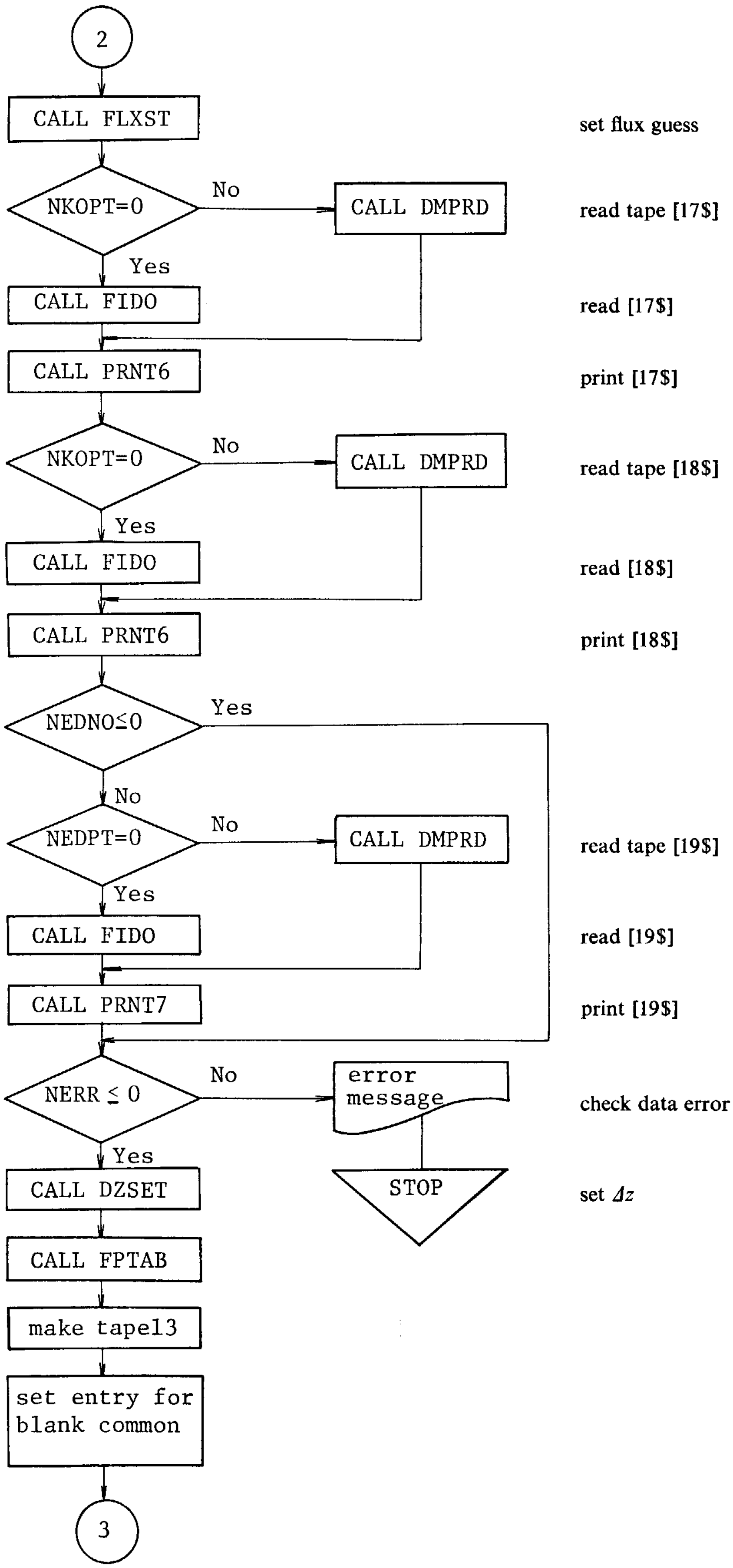
ii) MAINI

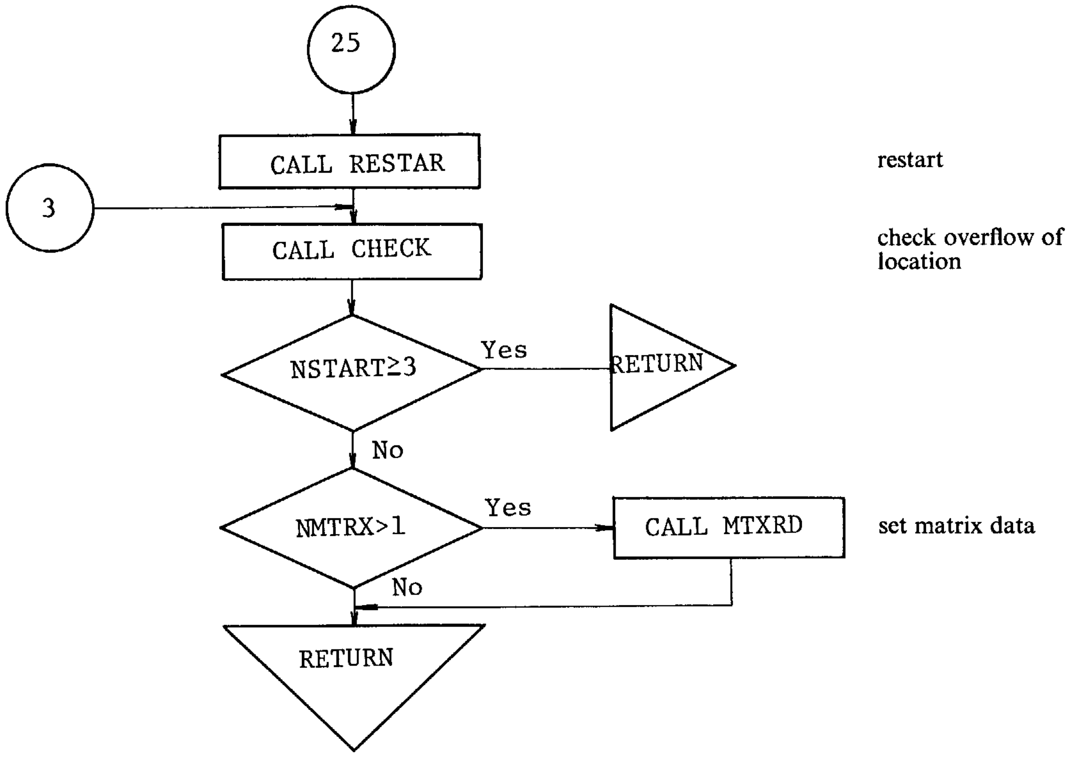

MAINI
program control
set time & data, and initialization
CALL ELAPS
print CPU time for each module
CALL INPUT
read input data
CALL ELAPS
MTRXEL skip?
Yes
skip matrix routine if NSTART=3 or NMTRX=4
No
CALL CHGFL1
CALL MEMSET
CALL MTRXEL
make matrix
CALL ELAPS
CALL CHGFL1
CALL MEMSET
CALL OUTER
outer iteration
CALL ELAPS
CALL CHGFL1
CALL MEMSET
CALL OUTPUT
final output
CALL ELAPS
end message
STOP

iii) INPUT

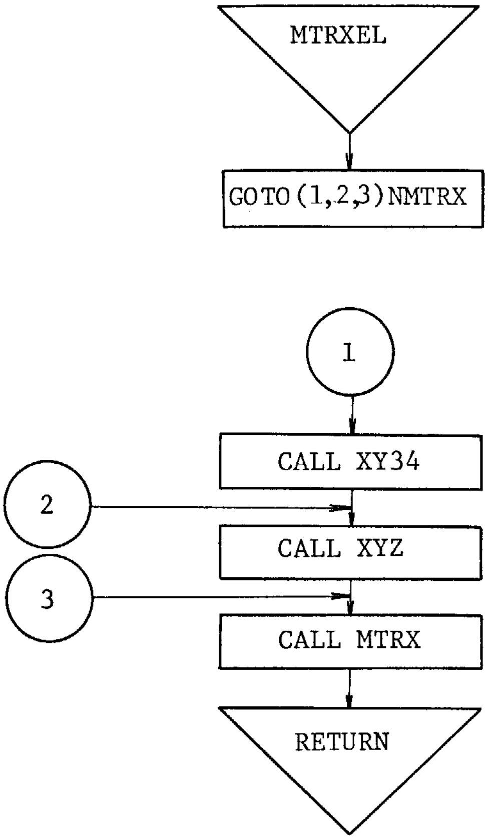

INPUT
read data and set variable dimension
read title, and print title
CALL FIDO
read [1$]
error ?
Yes
STOP
stop by error in [1$]
No
NSTART≥3
Yes
25
skip read if restart
No
CALL FIDO
read [2*]
set entry for input data
CALL PRINT1
print [1$], [2*]
NMTRX=1
No
CALL DMPRD
read tape [3$], [4*], [5*]
Yes
NAUTO>0
No
CALL FIDO
read [3$], [4*], [5*]
Yes
CALL FIDO
read [6*], [7*]
CALL AUTOM
auto-mesh-generator
NPRN3=1
Yes
CALL PRNT2
print [3$], [4*], [5*], [6*], [7*]
No
NMTRX>2
Yes
CALL DMPRD
read tape [8$], [9*]
No
CALL FIDO
read [8$], [9*]
NPRN2=1
Yes
CALL PRNT3
print [8$], [9*]
No
1


1
NXYOP=0 &
NMTRX≤3
No
CALL DMPRD
read tape [10$]
Yes
CALL FIDO
read [10$]
NMOPT=0 &
NMTRX≤3
No
CALL DMPRD
read tape [11$]
Yes
CALL FIDO
read [11$]
NPRN2=1
Yes
CALL PRNT4
print [10$], [11$]
No
NCSOP=0 &
NMTRX≤3
No
CALL DMPRD
read tape [12*]
Yes
CALL FIDO
read [12*]
NPRN1=1
Yes
CALL PRNT5
print cross section
No
NFISOP=0
No
CALL DMPRD
read tape [13*]
Yes
CALL FIDO
read [13*]
CALL PRNTG
NBCON≤0
Yes
No
NBCOP=0
No
CALL DMPRD
read tape [15$]
Yes
CALL FIDO
read [15$]
CALL PRNT6
2


2
CALL FLXST
set flux guess
NKOPT=0
No
CALL DMPRD
read tape [17$]
Yes
CALL FIDO
read [17$]
CALL PRNT6
print [17$]
NKOPT=0
No
CALL DMPRD
read tape [18$]
Yes
CALL FIDO
read [18$]
CALL PRNT6
print [18$]
NEDNO≤0
Yes
No
NEDPT=0
No
CALL DMPRD
read tape [19$]
Yes
CALL FIDO
read [19$]
CALL PRNT7
print [19$]
NERR ≤ 0
No
error message
check data error
Yes
CALL DZSET
STOP
set Δz
CALL FPTAB
make tape13
set entry for blank common
3

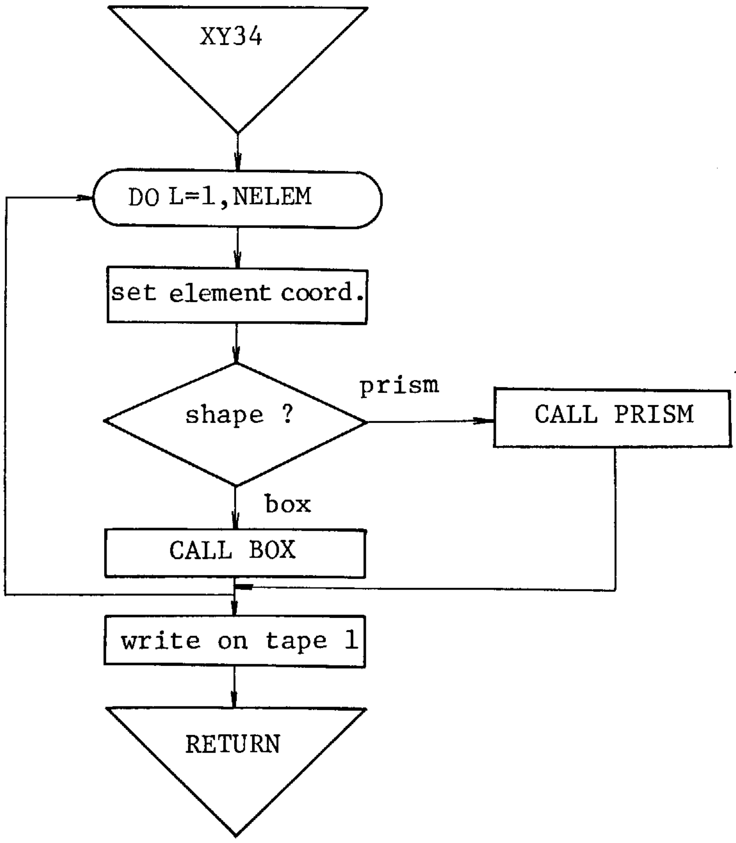

25
CALL RESTAR
restart
3
CALL CHECK
check overflow of location
NSTART≥3
Yes
RETURN
No
NMTRX>1
Yes
CALL MTXRD
set matrix data
No
RETURN

iv) MTRXL



control the routine making matrices

v) XY34



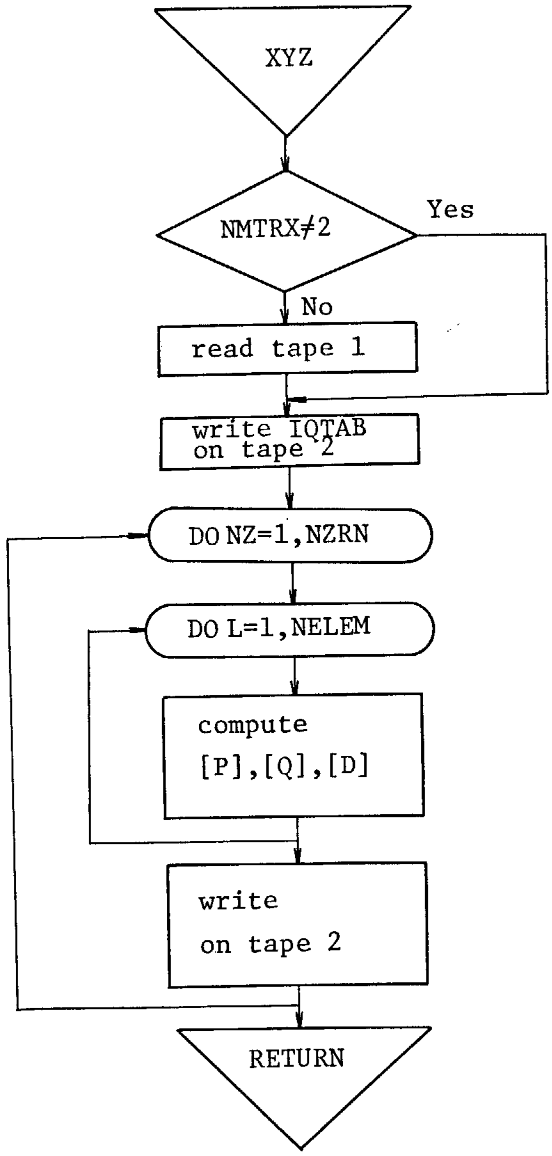
make two-dimensional $[B]^r$, $[Q]^r$

prism element

box element

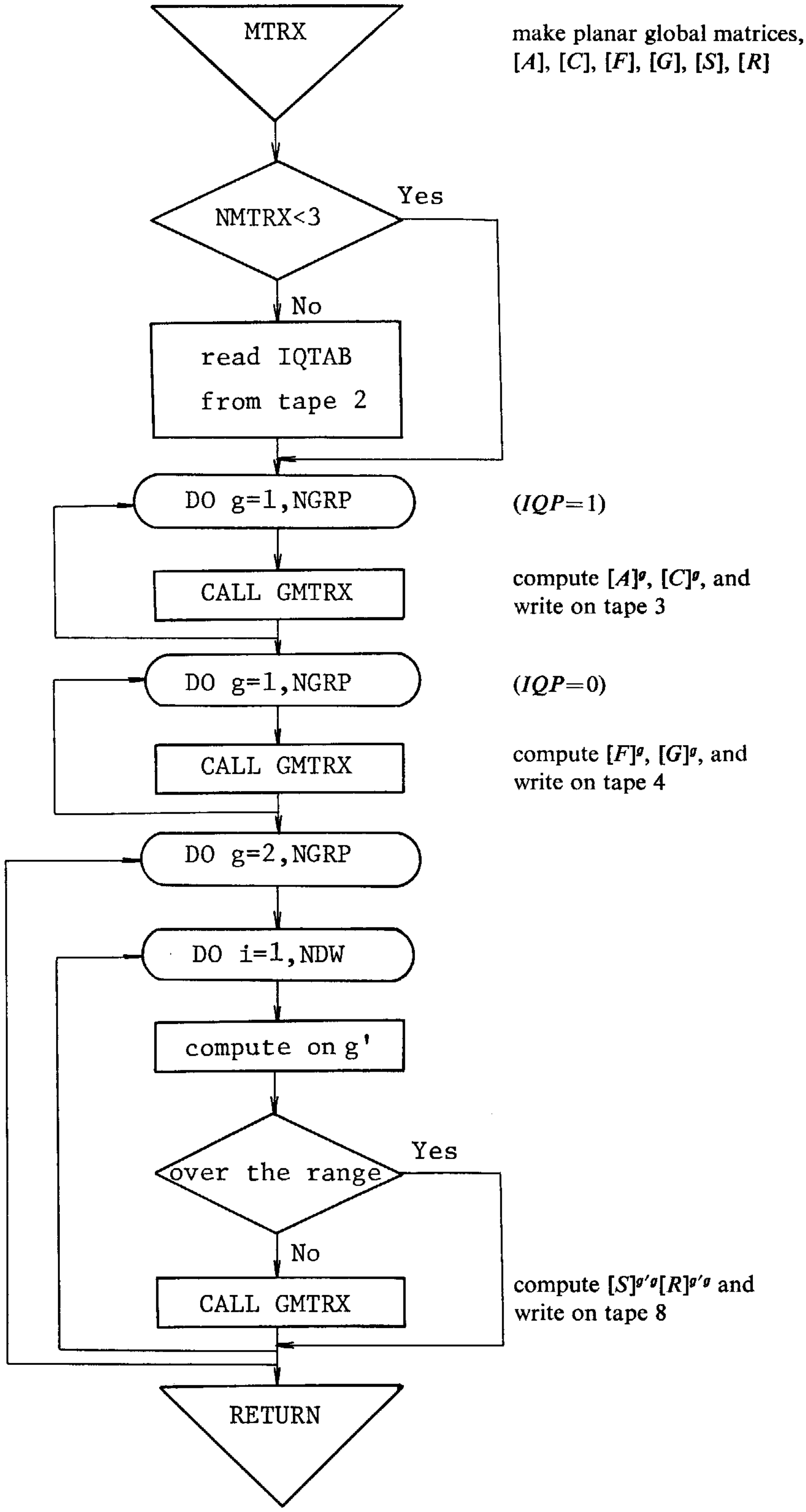
$[B]^r$, $[Q]^r$

vi) XYZ

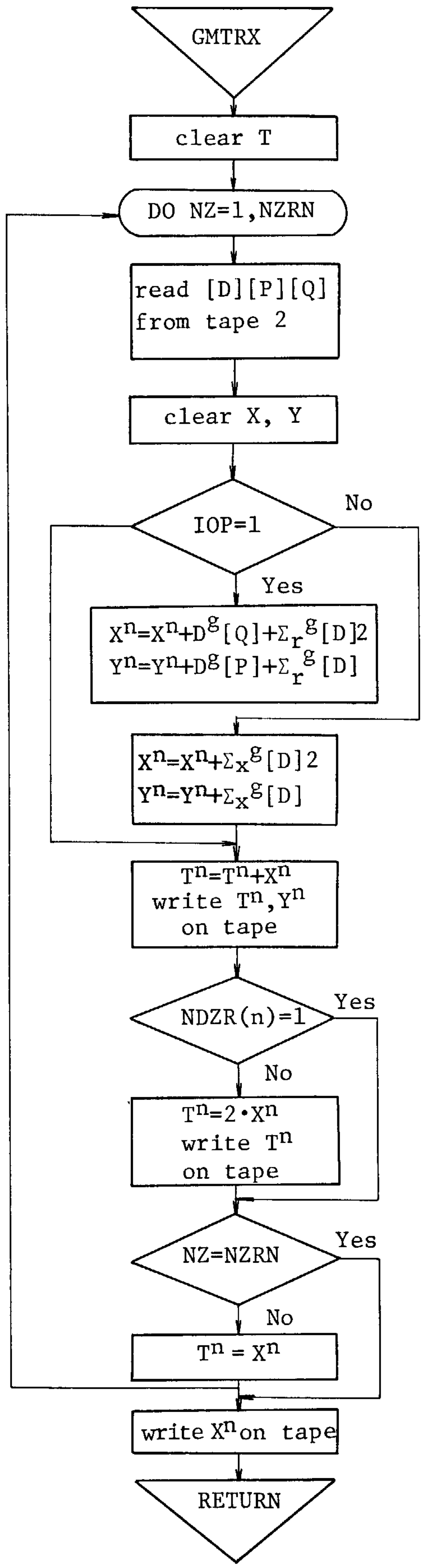


make three-dimensional
[*P*], [*Q*], [*D*] and write on tape

vii) MTRX


MTRX
make planar global matrices,
[A], [C], [F], [G], [S], [R]
NMTRX<3
Yes
No
read IQTAB
from tape 2
DO g=1,NGRP
(IQP=1)
CALL GMTRX
compute $[A]^g$, $[C]^g$, and
write on tape 3
DO g=1,NGRP
(IQP=0)
CALL GMTRX
compute $[F]^g$, $[G]^g$, and
write on tape 4
DO g=2,NGRP
DO i=1,NDW
compute on g'
over the range
Yes
No
CALL GMTRX
compute $[S]^{g'g}[R]^{g'g}$ and
write on tape 8
RETURN

viii) GMTRX



make planar sub-global matrices and write on tape

compute $[A]^g[C]^g$

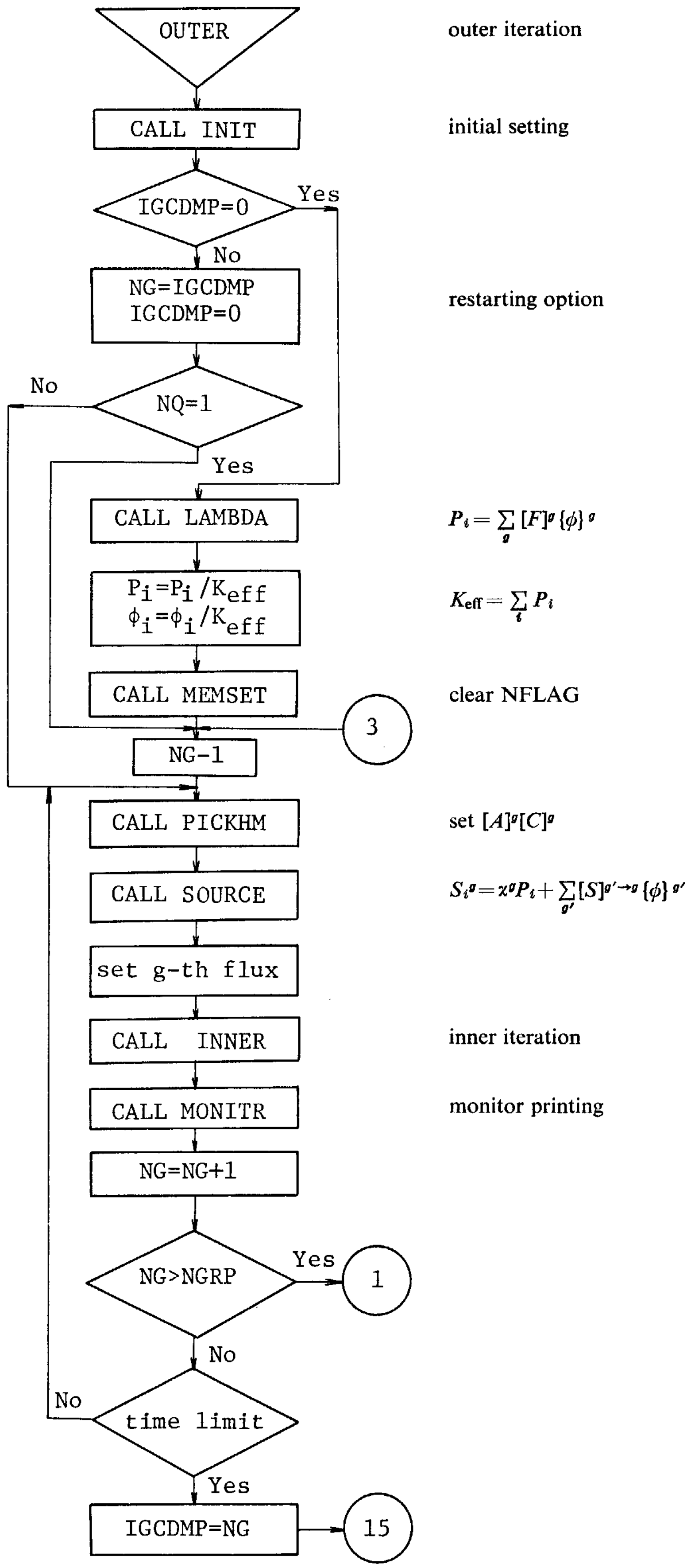
compute $[F]^g$, $[G]^g$ or $[S]^{g'g}$ $[R]^{g'g}$

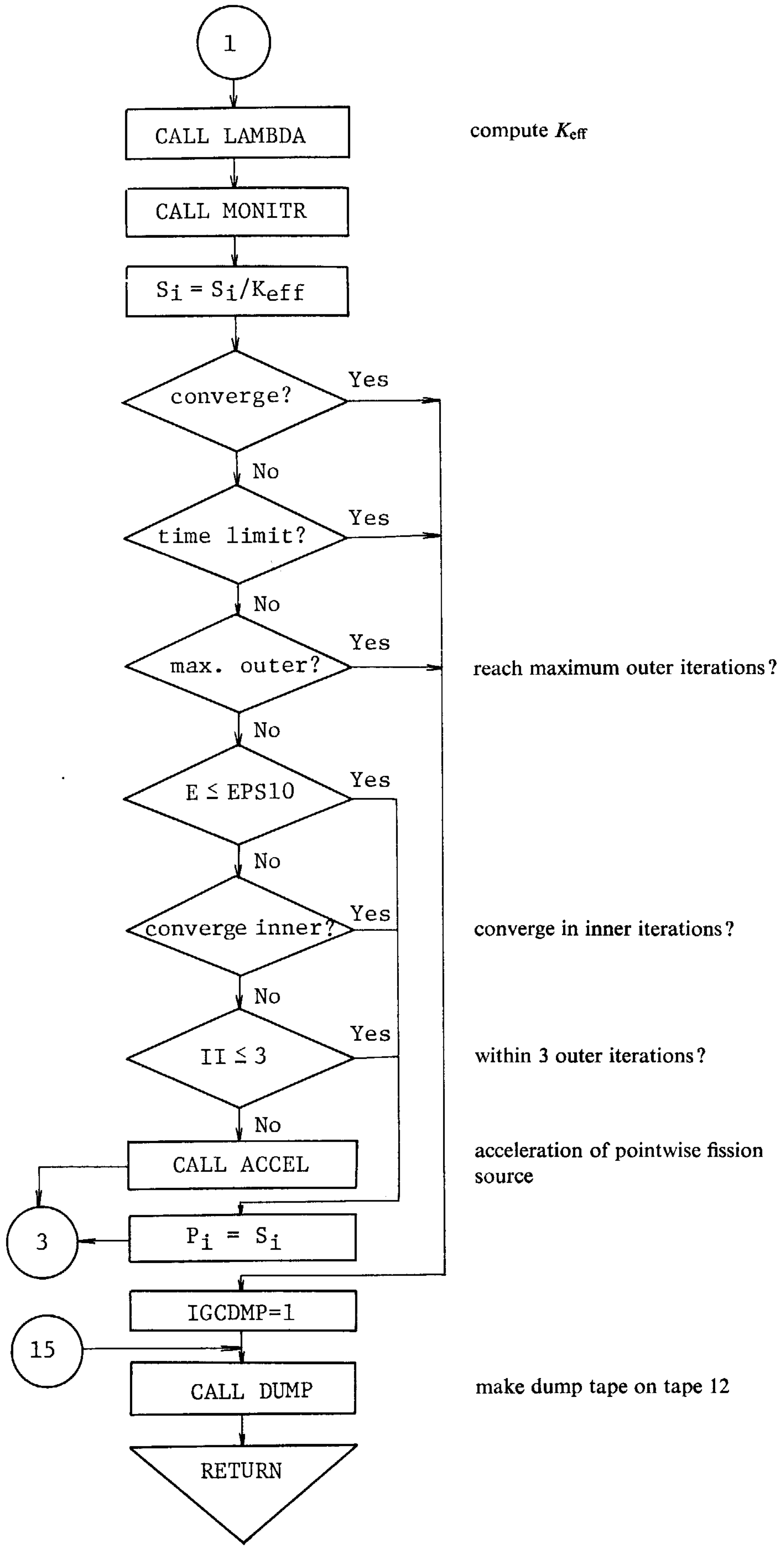
X_{NZ}^{NZ-1}, Y_{NZ}

X_{NZ}^{NZ} (within region)

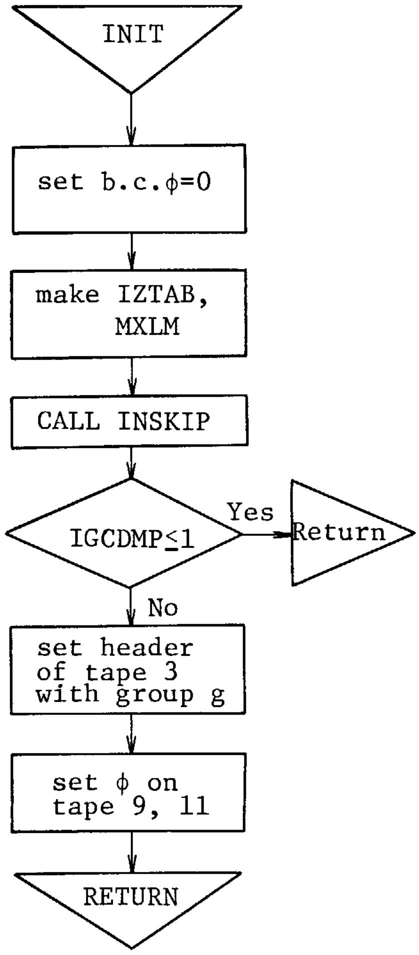
X_{last}

ix) OUTER


OUTER
outer iteration
CALL INIT
initial setting
IGCDMP=0
Yes
No
NG=IGCDMP
IGCDMP=0
restarting option
No
NQ=1
Yes
CALL LAMBDA
$P_i = \sum_g [F]^g \{\phi\}^g$
$P_i = P_i / K_{eff}$
$\phi_i = \phi_i / K_{eff}$
$K_{eff} = \sum_i P_i$
CALL MEMSET
clear NFLAG
3
NG-1
CALL PICKHM
set $[A]^g [C]^g$
CALL SOURCE
$S_i^g = \chi^g P_i + \sum_{g'} [S]^{g' \to g} \{\phi\}^{g'}$
set g-th flux
CALL INNER
inner iteration
CALL MONITR
monitor printing
NG=NG+1
NG>NGRP
Yes
1
No
No
time limit
Yes
IGCDMP=NG
15


1
CALL LAMBDA
compute K_{eff}
CALL MONITR
$S_i = S_i/K_{eff}$
converge?
Yes
No
time limit?
Yes
No
max. outer?
Yes
reach maximum outer iterations?
No
E ≤ EPS10
Yes
No
converge inner?
Yes
converge in inner iterations?
No
II ≤ 3
Yes
within 3 outer iterations?
No
CALL ACCEL
acceleration of pointwise fission source
3
$P_i = S_i$
IGCDMP=1
15
CALL DUMP
make dump tape on tape 12
RETURN

x) INIT

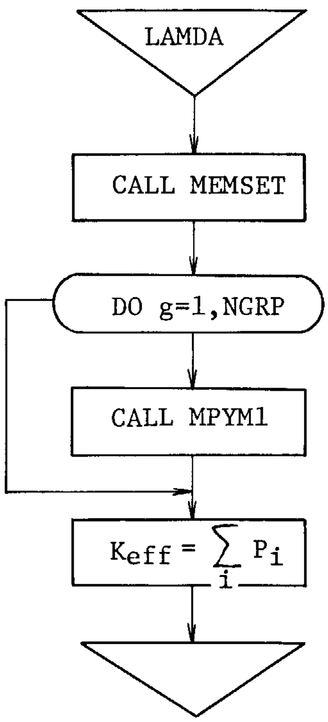


initial set before outer iteration and boundary condition (b.c.)

skip records of inscattering matrix on tape

set *g* at restart

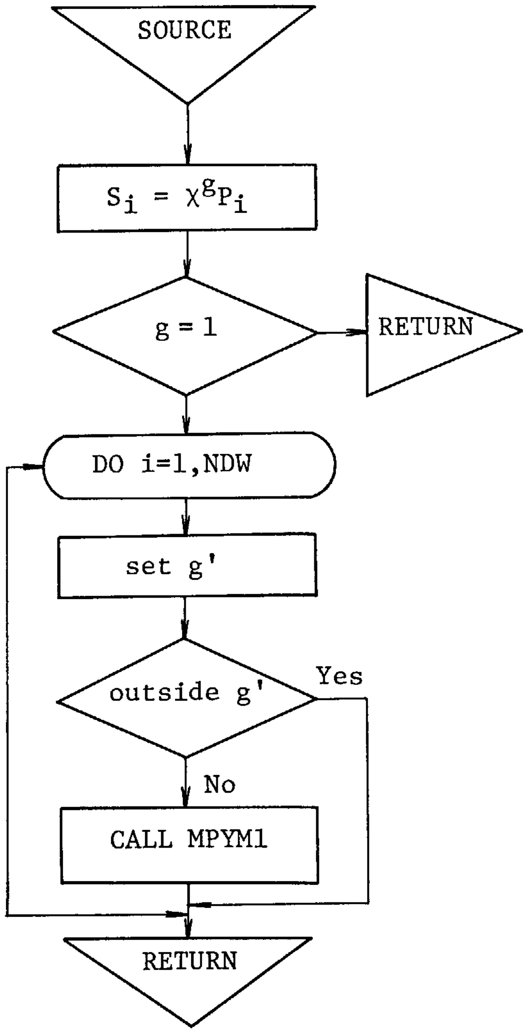
xi) LAMDA



calculate fission source and K_{eff}

$P_i=\sum_g [F]^g \{\phi\}^g$

xii) SOURCE

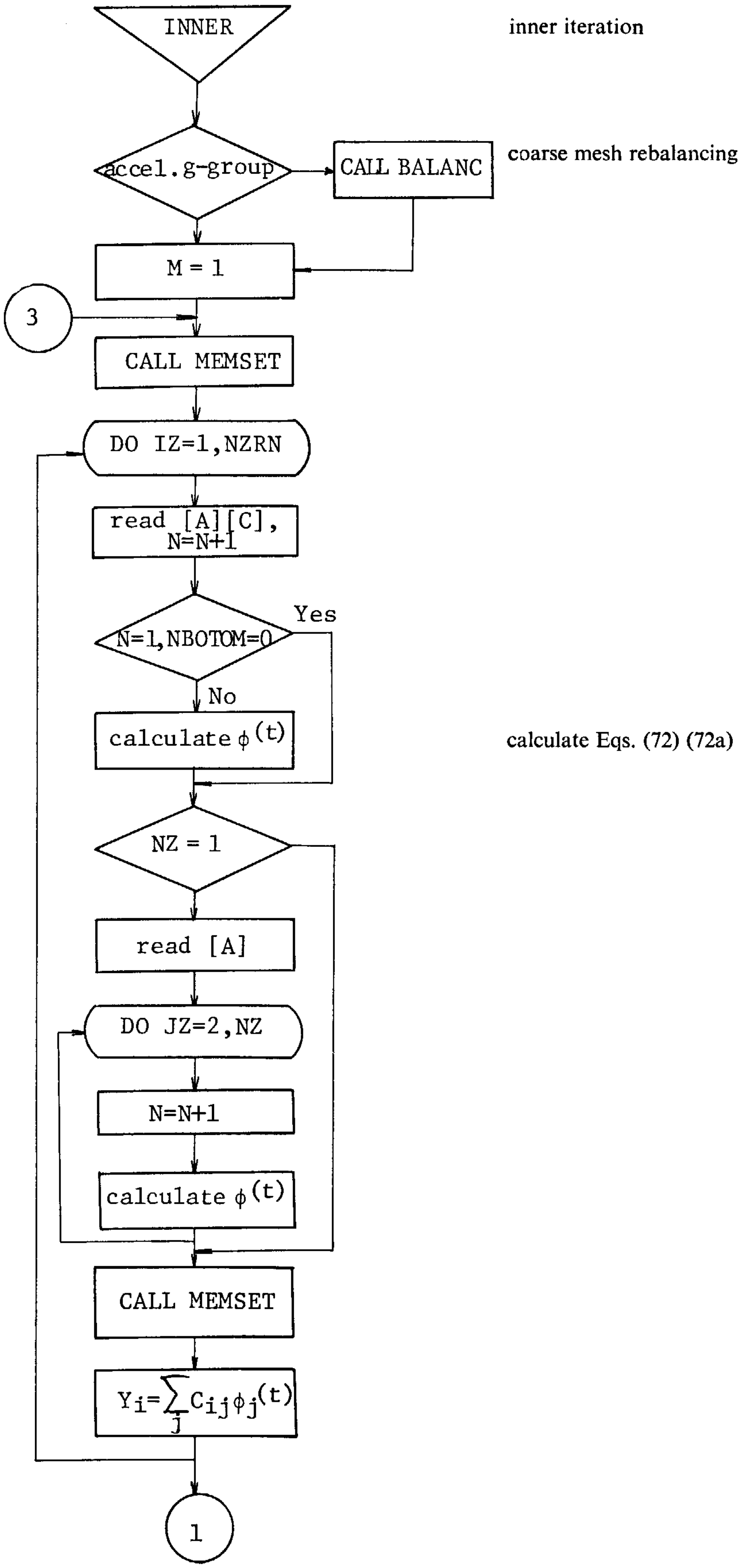


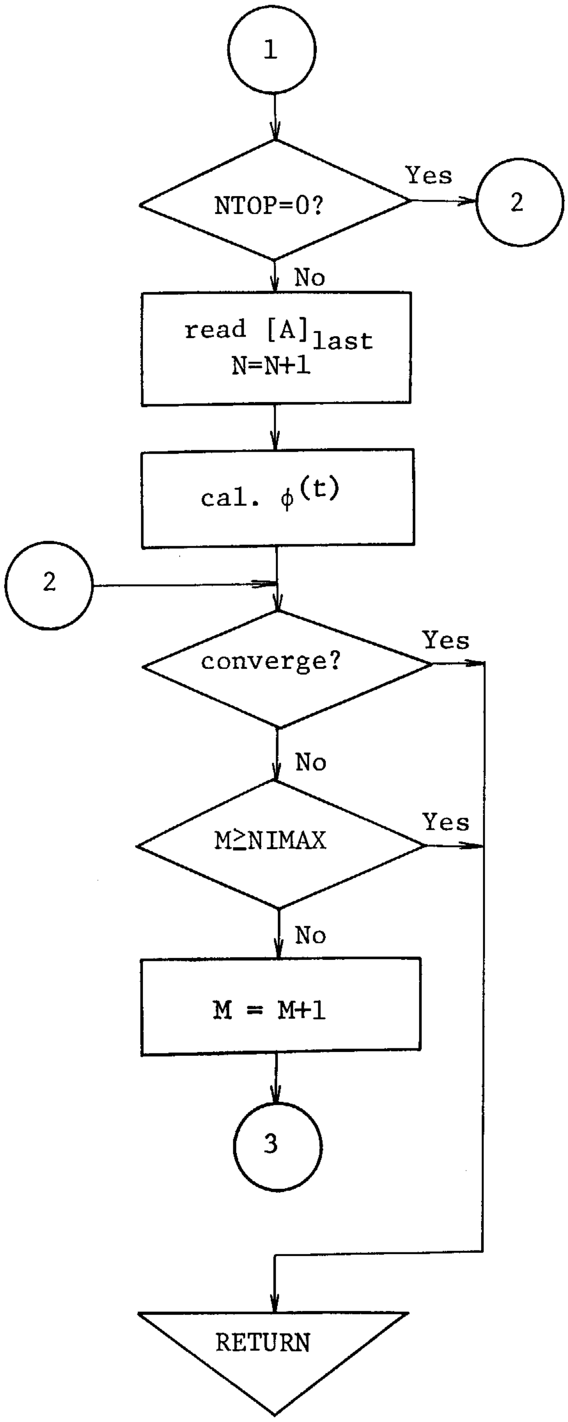
calculate g-th source term

fission source

source per iteration

xiii) INNER

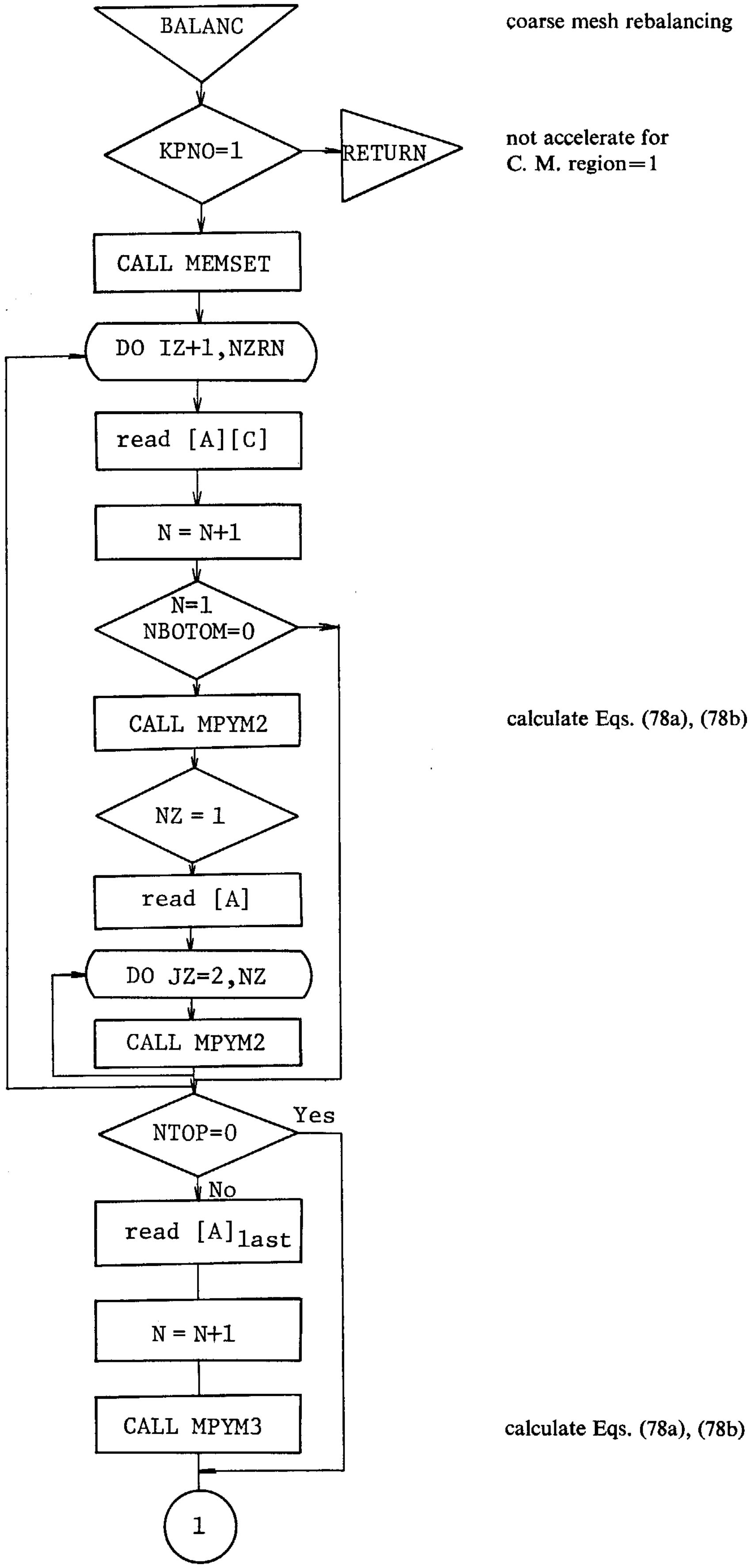

INNER
inner iteration
accel.g-group
CALL BALANC
coarse mesh rebalancing
M = 1
3
CALL MEMSET
DO IZ=1,NZRN
read [A][C], N=N+1
N=1,NBOTOM=0
Yes
No
calculate φ(t)
calculate Eqs. (72) (72a)
NZ = 1
read [A]
DO JZ=2,NZ
N=N+1
calculate φ(t)
CALL MEMSET
Yi=∑j Cij φj(t)
1

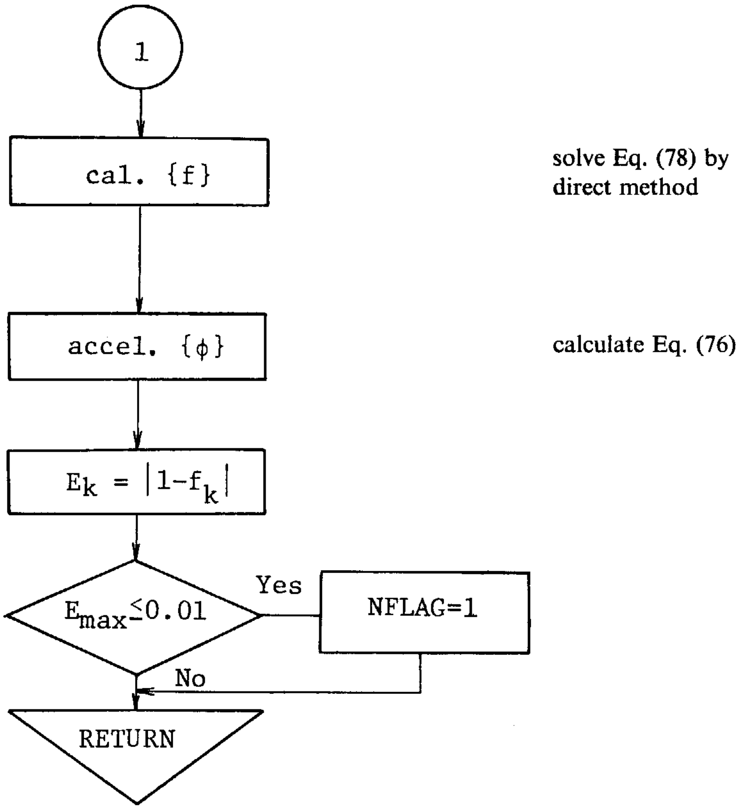


calculate Eqs. (72), (72a)

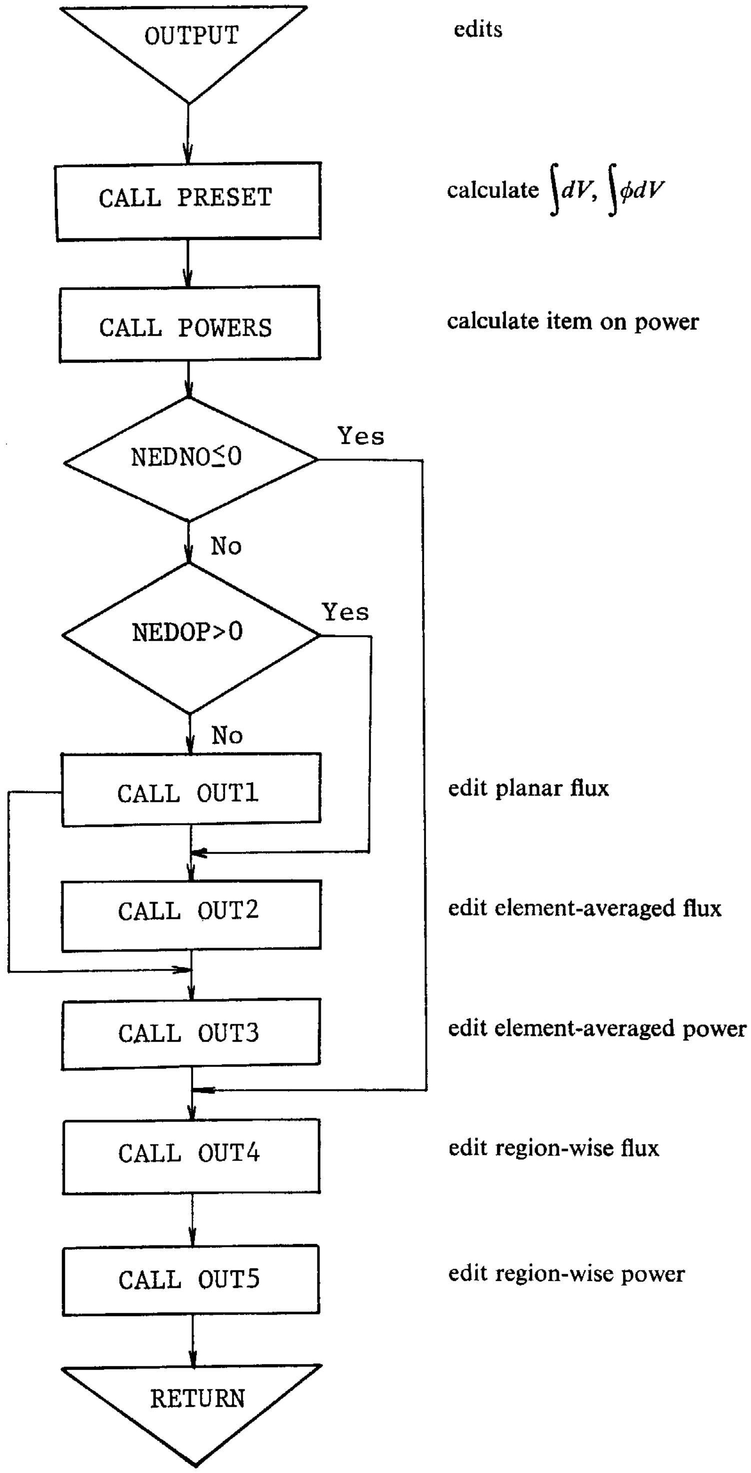
check inner max.

xiv) BALANC


BALANC
coarse mesh rebalancing
KPNO=1
RETURN
not accelerate for
C. M. region=1
CALL MEMSET
DO IZ+1,NZRN
read [A][C]
N = N+1
N=1
NBOTOM=0
CALL MPYM2
calculate Eqs. (78a), (78b)
NZ = 1
read [A]
DO JZ=2,NZ
CALL MPYM2
NTOP=0
Yes
No
read [A]last
N = N+1
CALL MPYM3
calculate Eqs. (78a), (78b)
1


1
cal. {f}
solve Eq. (78) by direct method
accel. {φ}
calculate Eq. (76)
Ek = |1-fk|
Emax≤0.01
Yes
NFLAG=1
No
RETURN

xx) OUTPUT


OUTPUT
edits
CALL PRESET
calculate $\int dV$, $\int \phi dV$
CALL POWERS
calculate item on power
NEDNO≤0
Yes
No
NEDOP>0
Yes
No
CALL OUT1
edit planar flux
CALL OUT2
edit element-averaged flux
CALL OUT3
edit element-averaged power
CALL OUT4
edit region-wise flux
CALL OUT5
edit region-wise power
RETURN

3.4 Descriptions of FIDO Input Form

The FIDO input format used in the FEM-BABEL has been widely used since it was developed at ORNL and has been adopted in the neutron transport programs ANISN, DOT[38] and so on. The input form is specially devised to allow the entering or modifying large data arrays with minimum labor. The options for repeated data and for symmetric data are especially of advantage to the finite element computer program. Accordingly, users may easily prepare data cards compactly if they are familiar with the FIDO format.

Fixed Field Input

Each card is divided into six 12-column data fields, each of which is divided into three subfields, as shown in **Fig. 7.** Three subfields are always composed of 2, 1, and 9 columns, respectively. To begin with the first array of a data block, an array originator is placed in any field on a card as follows:

Subfield 1: An integer array identifier (<100) specifying the data array to read.

Subfield 2: An array type indicator,
"$" if the array is integer data,
"*" if the array is real data,

Subfield 3: Blank.

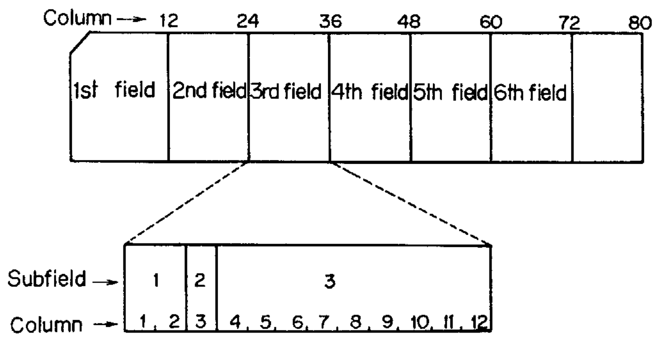


Fig. 7 Preparation of a data card in the FIDO format.

Data are then placed in successive fields until the required number of entries has been entered. In entering the data, it is convenient to think of a "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at the first location of the array by entering the array originator field. The pointer subsequently moves according to the data operator chosen. Blank fields neither cause any data modification nor move the pointer.

Data field

It has the following form:

Subfield 1: The data numerator (an integer <100). This entry is referred to as N_1 in the following explanation.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to read in F9.0 format. It will be converted to an integer if the array is a "$" array or if a special array operator such as "Q" is used. Note that an exponent is permissible but not required. Likewise, a decimal is permissible but not required. If no decimal is supplied, it is assumed to be immediately to the left of the exponent, if any, and otherwise to the right of the last column. This entry is referred to as N_3 in the following description.

Data operators and their effect on the array are as follows:

"Blank" indicates a single entry of data. The data entry in the 3rd subfield is entered in the location indicated by the pointer, and then the pointer is advanced by one. However, an entirely blank field is ignored.

"+" or "−" indicates exponentiation. The data entry in the 3rd field is multiplied by $10^{\pm N_1}$, where where N_1 is the data numerator in the 1st subfield, and the sign is indicated by the data operator itself. The pointer is advanced by one. In a case where an exponent is needed, this option allows the entering of more significant figures than the blank option.

"R" indicates that the data entry is to be repeated N_1 times. The pointer is advanced by N_1.

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry in the 3rd subfield is entered, followed by N_1 interpolated entries equally spaced between that value and the data entry found in the 3rd subfield of the next non-blank field. The pointer is thus advanced by (N_1+1). The field following an "I" field is processed normally according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by (N_1+N_3). A sequence of (N_1+N_3) is repeated one time only and the pointer is advanced by (N_1+N_3). This feature is especially valuable for geometry specification such as regions.

"N" indicates an inverse repetition of sequence of numbers analogously to "Q" except that the sequence is repeated in the inverse order.

"M" has the same effect as "N" but the sign of each entry in the sequence is reversed in addition.

"S" indicates that the pointer is to skip N_1 positions leaving those array positions unchanged. If the 3rd subfield is non-blank, then data entry is entered following the skip and the pointer is advanced by (N_1+1).

"A" moves the pointer to the position N_3 specified in the 3rd subfield.

"F" fills the remainder of the array with the datum entered in the 3rd subfield.

"E" skips over the remainder of the array. The array length criterion is always satisfied by an "E", no matter how many entries have been specified. No more entries to an array may be given following an "E" except that data entry may be restarted with an "A".

"Z" causes (N_1+N_3) locations to be set to 0. The pointer is advanced by (N_1+N_3).

The reading of data to an array is terminated when a new array originator field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given and a flag is set to abort later the execution of the problem. The FIDO then continues with the next array if an array originator was read but otherwise it returns control to the calling program.

A block termination consists of a field having "T" in the 2nd subfield. All entries following "T" on a card are ignored and control is returned from FIDO to the calling program.

Special Field Input

If the user follows the array identifier in the array originator field with the character "U", "V", "W" or "X" in the 2nd subfield, the input format can be selected by the user. If one of these characters is specified, the FORMAT explained below must be supplied in columns 1–72 of the next card. Then the data for the entire array must follow on successive cards. If the array data do not fill the last card, the remainder must be left blank. In a field with "U", "V", "W" or "X" the 3rd subfield must be left blank.

"U" inputs the following cards in the format (6E12.5).

"V" inputs the following cards in the format 4 (1X, E16.9, 1X).

"W" inputs the variable format data with (18A4) in the next card. In the following cards the data are read according to the specified format.

"X" inputs the data in the following cards according to the variable format already read as type "W".

3.5 Use of Restarting Procedures

The FEM-BABEL cannot run different problems successively. However, the restart procedure is designed to effect the continuation of the iterative process terminated in a previous run by saving the dump tape (tape 12; restarting procedure). In the restarting procedure, one can use optional data by taking out of the dump tape, for an example, the nuclear cross section data and/or the pointwise flux, etc.

The restart is specified by NSTART=3 or 4 in the input card "1$" and the calculation is continued following the last iteration in the previous run.

A. Complete restarting with NSTART=3

User should input "title" card and "1$" card only and specify "NSTART=3" in the "1$" card. All the data (the fluxes, the node points, the nuclear cross sections etc.) are read to be used as input data and/or initial guess fluxes from the dump tape.

B. Modified restarting with NSTART=4

This restarting procedure is to modify partial data which does not change the program flow and then to restart. One should input "title" card, "1$" card, and "2*" card. The data in the last run is taken for unmodified data. The data which can be modified are for the following items (see Section 3.7): 12., 13., 22., 23., 24., 25., 26., 27., 33., in "1$" card and 1., 2., 3., 4., 5., 6., 7. in "2*" card.

3.6 I/O File Unit Requirements

The contents in each tape file unit are described in the following, including those in the dump tape.

<table>
<tr><th>Unit No.</th><th>Contents</th></tr>
<tr><td>1</td><td>Geometrical matrix data for each element on $x-y$ plane;
record 1: (($B_T(i, \ell)$, $Q_T(i, \ell)$, i=1, 10), IQTAB (ℓ), ℓ=1, NELEM),
record 2: ($\Delta S(\ell)$, ℓ=1, NELEM)</td></tr>
<tr><td>2</td><td>Geometrical 3-dimensional matrix data for each z mesh and region;
record 1: (IQTAB (ℓ), ℓ=1, NELEM),
record 2~record NZRN+1: (($D(i, \ell)$, $P(i, \ell)$, $Q(i, \ell)$, i=1, 10), ℓ=1, NELEM)</td></tr>
<tr><td>3</td><td>$x-y$ plane global matrix data of every z region for each energy group, that is, A^g (NBAND, NPOINT), C^g (NBAND, NPOINT);
<table>
<tr><td colspan="10">← $g = 1$ →</td><td>← $g = 2$</td></tr>
<tr><td>A_1^0</td><td>C_1</td><td>A_1^1</td><td>A_2^2</td><td>C_2</td><td>A_2^2</td><td>…</td><td>A_{NZ}^{NZ-1}</td><td>C_{NZ}</td><td>A_{NZ}^{NZ}</td><td>A_{last}</td><td>…</td></tr>
<tr><td colspan="3">← $NZ = 1$ →</td><td colspan="3">← $NZ = 2$ →</td><td></td><td colspan="3">← $NZ = NZRN$ →</td><td></td><td></td></tr>
</table>
It is noted that there is no A_{NZ}^{NZ} when the number of meshes within a region, NZ is equal to 1.</td></tr>
</table>

4	F^g (NBAND, NPOINT), G^g (NBAND, NPOINT), and fission source term. The same form as for file unit 3.
5	Card input
6	Print output
7	Not used
8	Scattering term in the same form as for file unit 3: $S^{g'\to g}$ (NBAND, NPOINT), $R^{g'\to g}$ (NBAND, NPOINT) up to maximum NDW in the following order. |← 1 →|← 2 →| … |← g →| | S, R | S, R | S, R | · · · | S, R | · · · | S, R | $1\to 2$ $1\to 3$ $2\to 3$ $1\to g(\leq \text{NDW}+1)$ $g-1\to g$
9	Point-wise fluxes are written for each energy group and this file is used also as the external tape for initial flux guess; record 1~record NGRP: ((FLUX (i, j), i=1, NPOINT), j=1, NZMAX)
10	Temporary file. It is used at the inner iterations for $[A]^g$ and $[C]^g$.
11	Temporary file for fluxes (the same contents as in file unit 9)
12	Dump tape. See 17. and 18. in the card B in Section 3.7
13	It is used for restarting or constant data at editing.
14	It is used only as the external tape for the geometrical $x-y$ element data. For instance, it is directly read from the edit file of LOOM-P[28]

No.	Record	Contents of the dump tape	Description
1	1	M12 (22)	Flag for reading the dump tape
2	1	A (1)-A (260)	Entry table and input constants
3	1	"3$", "4*", "5*"	Geometrical data of x-y plane
4	1	"8$", "9*"	Geometrical data of z mesh
5	1	"10$"	Regional data on x-y plane
6	1	"11$"	Material table
7	1	"12*"	Cross section data
8	1	"13*"	χ data
9	1	"15$"	Data of the zero flux boundary conditions; it is skipped if NBCON≤0
10	1~NGRP	FLUX (NPOINT, NZMAX)	fluxes
11	1	"17$"	x-y planar data for coarse mesh rebalancing
12	1	"18$"	z mesh data for coarse mesh rebalancing
13	1	"19$"	Data for editing flux
14	1	PS (NPOINT, NZMAX)	Point fission source
15	1	DZRN (NZRN)	Δz for each axial region; record 1 of file unit 13
16*)	1	MFPI1 (NXYRN)	Point table for each region; record 2 of file unit 13
17*)	1	MFPI2 (MPOINT)	Point table for each region; record 3 of file unit 13
18	1	NFLAG (NGRP)	Flag for convergence performance of g-th group
19	1	B^r, Q^r, IQTAB	Record 1 of file unit 1
	2	ΔS	Record 2 of file unit 1
20	1	IQTAB	Record 1 of file unit 2
	2~NZRN+1	D, P, Q	Record 2 of file unit 2
21	1~MXLM	A, C	Copy of file unit 3
22	1~MXLM	F, G	Copy of file unit 4
23	1~MXLS	S, R	Copy of file unit 8

*) If NEDOP>0, it does not have these contents.

3.7 Input Specifications

The input data array for FEM-BABEL is read using the FIDO input system described in Section 3.4 except for "title" card. The data arrays are organized into blocks, which are terminated by a "T" delimiter as explained already. The FEM-BABEL does not permit to run the successive cases of 3-dimensional calculations, since they need too much computer time even on today's computers. However, the code has the various restarting functions in order to save the computer time as described in Section 3.5. An input example is shown in Appendix together with necessary control cards for the execution.

A. Title card

one card; format (18A4)

B. Integer data

1$ [33 parameters]

1. NGRP number of energy groups
2. NDW maximum number of groups for down-scatterings
3. NPOINT number of node points on x-y plane
4. NZMAX number of node points along z direction
5. NELEM number of elements on x-y plane
6. NXYRN number of geometrical regions on x-y plane
7. NZRN number of geometrical regions along z direction
8. MTT number of materials
9. NBCON number of node points with the zero flux boundary condition on x-y plane
10. NBOTOM bottom boundary condition;
    0: zero flux,
    1: reflective
11. NTOP top boundary condition;
    0: zero flux,
    1: reflective
12. NIMAX inner iteration maximum allowed for energy group
13. NOMAX outer iteration maximum for execution stop
14. NBAND half bandwidth in the global matrix (see Section 3.2)
15. NKXY number of coarse mesh rebalancing regions on x-y plane
16. NKZ number of coarse mesh rebalancing regions along z direction
17. NSTART starting option for initial guess for flux;
    0: flat,
    1: guess read from the external tape (file unit 9),
    2: guess read from the dump tape (file unit 12),
    3: complete restarting from the dump tape (file unit 12),
    4: modified restarting from the dump tape (file unit 12)
18. NMTRX option for matrix calculation;
    1: calculate all the matrices for a new case,
    2: read the geometrical matrix on x-y plane from the dump tape (file unit 12),
    3: read the 3-dimensional geometrical matrix from the dump tape (file unit 12),
    4: read all the global matrices from the dump tape (file unit 12)

19. NPRN1 print option for the material cross sections;
 0: no print,
 1: print
20. NPRN2 print option for regional data;
 0: no print,
 1: print
21. NPRN3 print option for elements and coordinates;
 0: no print,
 1: print
22. NXYOP input option for regional data on *x-y* plane;
 0: by cards,
 1: from the dump tape
23. NMOPT input option for material number data;
 0: by cards,
 1: from the dump tape
24. NCSOP input option for nuclear cross section data;
 0: by cards
 1: from the dump tape
25. NFISOP input option for χ data;
 0: by cards,
 1: from the dump tape
26. NBCOP input option for zero flux boundary condition data;
 0: by cards,
 1: from the dump tape
27. NKOPT input option for the coarse mesh rebalancing region data;
 0: by cards,
 1: from the dump tape
28. NAUTO option for auto-mesh generating routine (on *x-y* plane);
 0: not used (input by cards),
 1: generate grid meshes all composed of right angle triangles,
 2: generate grid meshes all composed of rectangles,
 3: read from the external tape (file unit 14)
29. NXP1 number of node points along *x* direction for auto-mesh generating routine
30. NYP1 number of node points along *y* direction for auto-mesh generating routine
31. NEDOP edit option for flux;
 0: edit the point-wise fluxes on *x-y* planes,
 1: edit the element-averaged fluxes on *x-y* planes,
 2: edit the element-averaged fluxes on *z* meshes
32. NEDNO number of edit fluxes for *x-y* planes or *z* meshes; the number of *z* meshes with *x-y* planes for edit *x-y* plane fluxes, or the number of *x-y* points with *z* meshes for edit *z* mesh fluxes
33. NEDPT input option for parameter 32, NEDNO;
 0: by cards,
 1: from the dump tape

"T" terminator

C. Floating point data

2* [7 parameters] (input for NSTART$\neq$3 in "1$" card)

1. EPS1 criterion for outer iteration convergence (K_{eff})
2. EPS2 criterion for inner iteration convergence (point-wise flux)
3. SORF over-relaxation factor β due to SOR method; $1.0 \leq \text{SORF} \leq 2.0$
4. POWER operating power level in megawatts for normalizing fluxes
5. TIME CPU execution time limit in minutes
6. FA1 coefficient on geometrical symmetry of a nuclear reactor for power-normalized fluxes (such as 1/FA1-reactor core)
7. FA2 number of nuclear fissions per watt-sec ($1/K_f$, see also Section 3.9)

"T" terminator

D. Geometrical data

point data on x-y plane (input for NMTRX=1 and NAUTO=0 "1$" card)

3$ NELNO (4, NELEM)
input the number of the node points which compose an element in anticlockwise, put NELNO (3, NE)=NELNO (4, NE) for a triangular element

4* PX (NPOINT)
input x-coordinate on each node point

5* PY (NPOINT)
input y-coordinate on each node point

"T" terminator

data for auto-mesh generating routine (input for NMTRX=1 and NAUTO>0)

6* XNODE (NXP1)
distance from center to each mesh-division in x-coordinate

7* YNODE (NYP1)
distance from center to each mesh-division in y-coordinate

"T" terminator

z mesh data (input for NMTRX$\leq$2)

8$ NDZR (NZRN)
number of divisions in each region in z-direction (from bottom)

9* ZNODE (NZRN)
region width in each region in z-direction (from bottom)

"T" terminator

regional data on x-y plane (input for NXYOP=0 and NMTRX$\leq$3)

10$ NXYGN (NELEM)
assign the region number on each element

"T" terminator

E. Material data

material table (input for NMOPT=0 and NMTRX$\leq$3 in "1$" card)

11$ NMRGN (NXYRN, NZRN)
assign the material number on each region (from bottom)

"T" terminator

cross section data (input for NCSOP=0 and NMTRX$\leq$3)

12* CS (IHM, NGRP, MTT)

input macroscopic cross sections on each energy group for every material; within a group, the order of data in cross section tables is:

Position	Entry
1	Σ_f^g
2	D^g
3	$\nu\Sigma_f^g$
4	Σ_a^g
5	$\Sigma_s^{g\to g+1}$
6	$\Sigma_s^{g\to g+2}$
.	.
.	.
.	.
IHM	$\Sigma_s^{g\to g+NDW}$

where NDW is the number of groups for down-scatterings

"T" terminator

F. Fission spectrum (input for NFISOP=0 in "1$" card)

13* AKAI (NGRP)

input the energy-wise χ^g in order of g=1, 2, . . . NGRP

"T" terminator

G. Data for the zero flux boundary condition on x-y plane (input for NBCOP=0 and NBCON>0 in "1$" card)

15$ NBPOT (NBCON)

input all the numbers of node points with the zero flux boundary condition

"T" terminator

H. Data for coarse mesh rebalancing region (input for NKOPT=0 in "1$" card)

x-y plane data

17$ KRPNT (NPOINT)

assign the coarse mesh rebalancing region number to each node point

"T" terminator

z mesh data

18$ KZRN (NZRN)

assign the coarse mesh rebalancing region number to each region (from bottom)

"T" terminator

I. Edit data (input for NEDNO>0 and NEDPT=0 in "1$" card)

19$ NEDTB (NEDNO)

position table of edited fluxes or powers for specifying;
the node point numbers in the z direction for NEDOP=0 (from bottom), or
the mesh (element) numbers in z direction for NEDOP=1 (from bottom), or
the element numbers on x-y plane for NEDOP=2.
It is noted that the power in z meshes is edited even for NEDOP=0 at power edit.

"T" terminator

3.8 Operating Instructions

Since the variable dimension in the blank common is adopted in FEM-BABEL, users must determine the amount of data storage required for a problem in the program in order to use the computer efficiently as well as to prevent the storage from overflowing.

The required amount of data storage space, MEMORY, is given by MEMORY=PL+BLK, where PL, the program size which contains the labeled common, is 41 kilowords on the FACOM 230/75 operating system. The BLK, the length of blank common, is given by

BLK=INPUT+max {MTRXEL, OUTER, OUTPUT},

where INPUT, MTRXEL, OUTER and OUTPUT are the lengths of blank common in their modules, respectively. Usually user can refer to the memory arrangement printed at the beginning of the edit as shown in **Table 1.** Otherwise, one can calculate the length of blank common in each module as follows:

INPUT=260+6×NELEM+NPOINT (3+NZMAX)+NZRN (4+NXYRN)+NGRP+NBCON+ NEDNO+IHM×NGRP×MTT,

MTRXEL=31×NELEM+3×NBAND×NPOINT,

OUTER=KPNO×KPNO1+2×NPOINT (NZMAX+NBAND+1)+2×NGRP,

OUTPUT=NELEM (3×NZMAX−2)+NXYRN (1+NZRN (7+2×NGRP))+NGRP.

Table 1 Memory arrangement edited as an example on FACOM 230/75 computer

Module name	Required size (words)	Allowed size (words)
INPUT	25125	100000
MTRXEL	86584	100000
OUTER	98844	100000
OUTPT	82508	100000

3.9 Edits

Input edit

Following the title, the edit prints the input data B., C., D., E., F., H., and I. explained in Section 3.7. The table for the memory arrangement is printed at the end.

Output edit

It begins with editing the convergence performance (on K_{eff} and flux) and the calculation time for each outer iteration. Additional edits are described in the following (i=node point, g=energy group, e=element and r=region),

i) point-wise fluxes normalized by power for each region ($\phi_i^{*,g}$);

$$\phi_i^{*,g} = \alpha \phi_i^g ,$$

where

$$\alpha = \frac{\mathrm{POWER} \times 10^6 \times \mathrm{FA2}}{\mathrm{FA1} \times \int_{\mathrm{core}} \sum_g \Sigma_f^g \phi^g dV}$$

(see "2*" in Section 3.7 for POWER, FA1 and FA2).

ii) point-wise powers (PP_i);

$$PP_i = \sum_g \Sigma_f^g \phi_i^{*,g} .$$

iii) element-averaged fluxes and powers ($\bar{\phi}_e^{*,g}$, $\overline{PP}_e$);

$$\bar{\phi}_e^{*,g} = \frac{\int_e \phi^{*,g} dV}{\int_e dV} ,$$

$$\overline{PP}_e = \frac{\int_e PPdV}{\int_e dV} .$$

iv) region-wise edit;

regional volume: $V_r = \sum_{e \in r} \int_e dV$,

regional power: $PR_r = \sum_{e \in r} \int_e PPdV$,

regional volume-averaged flux: $AF_r^g = \frac{1}{V_r} \sum_{e \in r} \int_e \phi^{*,g} dV$,

regional maximum flux: $FM_r^g = \max\{\phi_i^{*,g} : i \in r\}$,

regional maximum power: $PM_r = \max\{PP_i : i \in r\}$,

regional flux peaking factor: $PFF_r^g = \frac{FM_r^g}{AF_r^g}$,

regional power peaking factor: $PFP_r = PM_r \cdot \frac{V_r}{PR_r}$.

v) edit for total system

system volume: $V = \sum_r V_r$,

system power: $POW = \sum_r PR_r$,

system flux peaking factor: $PFF = FMAX \frac{V}{\int_{\text{system}} \phi^{*,g} dV}$

system power peaking factor: $PFP = PMAX \frac{V}{POW}$,

where $FMAX$ and $PMAX$ are the maximum point flux and the maximum point power in the system, respectively. In addition, the element numbers and the mesh numbers in z direction are edited for maximum point powers.

The descriptions about error message are omitted as they are self-explanatory in the code.

4. Program Application and Comparison with Other Methods

The main purpose of this chapter is to establish the reliability of FEM-BABEL. For this purpose we have treated in Section 4.1 a homogeneous cubic reactor problem which is exactly solvable and in Section 4.2 a modified IAEA three-dimensional problem for comparison with the finite difference calculations. Comparisons have been made for the eigenvalues, power distributions, convergences of outer and inner iterations, computer storage requirements and computation time.

In discussion of the power distributions, the element-averaged powers (see Chapter 3.9) are used since the finite difference program CITATION[39] adopts the central difference scheme. Program sizes are 41 kilowords for FEM-BABEL and 60 kilowords for CITATION on the FACOM 230/75 operating system. The calculations are performed with 3 and 4 inner interactions per outer iteration for the 1st and 2nd problems, respectively, for both the programs. Material constants used for these test problems are listed in **Table 2** (see also **Fig. 12** for the corresponding material number of the IAEA problem).

Table 2 Material constants for the test problems

Problem	Material	Energy group	$\Sigma_{f,g}$ (cm^{-1})	D_g (cm)	$\nu\Sigma_{f,g}$ (cm^{-1})	$\Sigma_{a,g}$ (cm^{-1})	$\Sigma_{s,g,g+1}$ (cm^{-1})	χ_g	Comment
Exact problem[5]	1	1	1.2334×10^{-2}	2.6800	3.0834×10^{-2}	1.3785×10^{-2}	4.0792×10^{-2}	0.575	Core
		2	1.0080×10^{-2}	1.5788	2.5200×10^{-2}	1.4496×10^{-2}	0.0	0.425	
IAEA problem[33]	1	1	0.0	1.50	0.0	0.01	0.02	—	Fuel 1
		2	0.056	0.40	0.135	0.08	0.0		
	2	1	0.0	1.50	0.0	0.01	0.02	1.0	Fuel 2
		2	0.056	0.40	0.135	0.085	0.0	0.0	
	3	1	0.0	1.50	0.0	0.01	0.02	—	Fuel 2 +absorber
		2	0.056	0.4	0.135	0.13	0.0		
	4	1	0.0	2.0	0.0	0.0	0.04	—	Reflector
		2	0.0	0.3	0.0	0.01	0.0		
	5	1	0.0	2.0	0.0	0.0	0.04	—	Reflector +absorber
		2	0.0	0.3	0.0	0.055	0.0		

4.1 Verification of the Program through Exact Solution

In order to verify the computer program FFM-BABEL, it will be best to deal with an analytically solvable problem. We therefore consider here a homogeneous cubic reactor in a two-energy-group model. The reactor configuration is illustrated in **Fig. 8.** In this case, the multigroup diffusion equation is exactly solvable. Now, set $\Omega=\{(x, y, z): 0<x<L, 0<y<L, 0<z<L\}$, and the system of Eq. (1) takes the form:

$$
\begin{aligned}
-D_1\Delta\phi_1+\Sigma_{r,1}\phi_1 &= \frac{1}{K_{\text{eff}}}\chi_1[(\nu\Sigma_f)_1+(\nu\Sigma_f)_2\phi_2]\,, \\
-D_2\Delta\phi_2+\Sigma_{r,2}\phi_2 &= \Sigma_{s,12}\phi_1+\frac{1}{K_{\text{eff}}}\chi_2[(\nu\Sigma_f)_1\phi_1+(\nu\Sigma_f)_2\phi_2]\,,
\end{aligned}
\qquad (81)
$$

for $(x, y, z)\in\Omega$, with the boundary conditions

$$\frac{\partial}{\partial x}\phi_g(0,x,z)=\frac{\partial}{\partial y}\phi_g(x,0,z)=\frac{\partial}{\partial z}\phi_g(x,y,0)\,, \tag{81a}$$

$$\phi_g(L,x,z)=\phi_g(x,L,z)=\phi_g(x,y,L)=0\,, \tag{81b}$$

for $(x, y, z)\in\partial\Omega$, and $g=1, 2$.

If we set $B^2=3\,(\pi/2L)^2$ and $E_g=D_gB^2+\Sigma_{r,g}$ for $g=1, 2$, then the multiplication factor $K_{\rm eff}$ is given by the expression

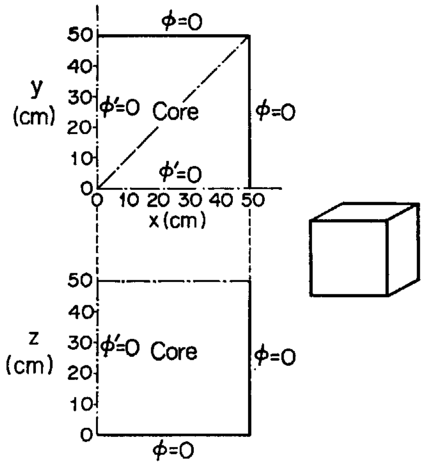


Fig. 8 Reactor configuration of the exact problem.

Table 3 Comparison between the numerical and analytical values of the multiplication factor for the exact problem

	Analytical value	Mesh size					
		2 cm		5 cm		10 cm	
		$K_{\rm eff}$	Relative error	$K_{\rm eff}$	Relative error	$K_{\rm eff}$	Relative error
FEM-BABEL	1.335506	1.33537*	0.010%	1.33473*	0.058%	1.33211*	0.25%
		—	—	1.33478	0.054%	—	—
CITATION		1.33562	0.009%	1.33623	0.054%	1.33842	0.22%

* With octant symmetric configuration

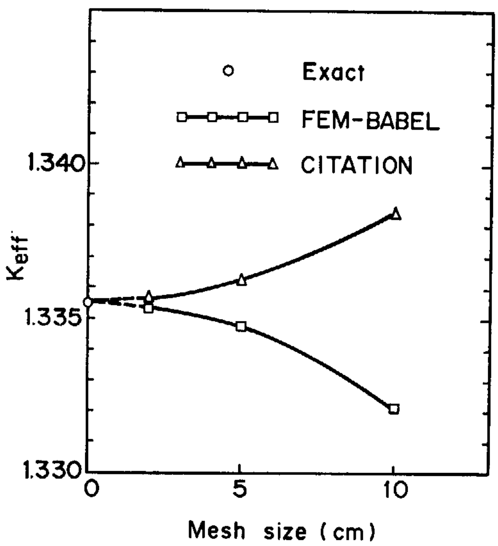


Fig. 9 Comparison of $K_{\rm eff}$ between FEM-BABEL and CITATION as a function of mesh refinements.

Table 4 Comparison between the numerical and analytical values of the power distribution for the mesh size of 2 cm

	x (cm)												
	1.0	5.0	9.0	13.0	17.0	21.0	25.0	29.0	33.0	37.0	41.0	45.0	49.0
Analytical	1.0	0.9882	0.9608	0.9182	0.8612	0.7905	0.7075	0.6132	0.5093	0.3973	0.2791	0.1565	0.03143
FEM-BABEL	1.0	0.9879	0.9605	0.9180	0.8610	0.7903	0.7073	0.6131	0.5092	0.3972	0.2791	0.1565	0.03142
	—	0.03%	0.03%	0.02%	0.02%	0.03%	0.03%	0.02%	0.02%	0.03%	0.00%	0.00%	0.03%
CITATION	1.0	0.9877	0.9605	0.9178	0.8610	0.7903	0.7072	0.6130	0.5091	0.3972	0.2790	0.1565	0.03141
	—	0.05%	0.03%	0.04%	0.02%	0.03%	0.04%	0.03%	0.04%	0.03%	0.03%	0.00%	0.06 %

Percent values are the fractional deviations from the analytic solution

$$K_{\text{eff}} = \frac{\chi_1(\nu\Sigma_f)_1 E_2 + \chi_1(\nu\Sigma_f)_2 \Sigma_{s,12} + \chi_2(\nu\Sigma_f)_2 E_1}{E_1 E_2} \tag{82}$$

and the unnormalized flux corresponding to K_{eff} is written as follows:

$$\phi_g(x, y, z) = A_g \cos\left(\frac{\pi}{2L}x\right)\cdot\cos\left(\frac{\pi}{2L}y\right)\cdot\cos\left(\frac{\pi}{2L}z\right), \qquad \text{for} \quad g = 1.2 \tag{83}$$

where

$$A_1 = \alpha A_2\,,$$

$$\alpha = \frac{\chi_1 E_2}{\chi_2 E_1 + \chi_1 \Sigma_{s,12}}\,.$$

If the flux is normalized to one fission in the entire reactor, then

$$A_2 = \frac{(1/8)\cdot B^2}{\alpha(\Sigma_f)_1 + (\Sigma_f)_2}\,.$$

For the program check of FEM-BABEL, we used this exact solution and the finite difference solution from CITATION.[39] The finite difference calculations were performed for a quarter symmetric configuration for a sequence of mesh sizes of 2 cm, 5 cm, 10 cm. On the other hand, the finite element calculations were carried out for an octant symmetric configuration for the same mesh size sequence and for a quarter symmetric configuration with the mesh size of 5 cm.

From a comparison of the effective multiplications with the exact value given in **Table 3,** the finite element solutions are seen to come as near to the exact value as the finite difference solutions. As illustrated in **Fig. 9,** we also note the finite element solutions approach the exact value from smaller values as the mesh

Table 5 Comparison of the numerical and analytical values of the power distribution for the mesh size of 5 cm

		x (cm)									
		2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5	42.5	47.5
Analytical		1.0	0.9754	0.9267	0.8553	0.7628	0.6515	0.5241	0.3839	0.2342	0.07870
FEM-BABEL	Octant	1.0	0.9742	0.9254	0.8541	0.7617	0.6506	0.5234	0.3834	0.2339	0.07860
		—	0.12%	0.14%	0.14%	0.14%	0.14%	0.13%	0.13%	0.13%	0.13%
	Quarter	1.0	0.9754	0.9268	0.8553	0.7628	0.6515	0.5241	0.3839	0.2342	0.07870
		—	0.00%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CITATION		1.0	0.9754	0.9267	0.8553	0.7627	0.6515	0.5241	0.3840	0.2342	0.07869
		—	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.03%	0.00%	0.01%

Table 6 Comparison between the numerical and analytical values of the power distribution for the mesh size of 10 cm

	x (cm)				
	5.0	15.0	25.0	35.0	45.0
Analytical	1.0	0.9021	0.7159	0.4597	0.1584
FEM-BABEL	1.0	0.8979	0.7119	0.4574	0.1576
	—	0.47%	0.56%	0.50%	0.51%
CITATION	1.0	0.9021	0.7160	0.4596	0.1584
	—	0.00%	0.01%	0.02%	0.00%

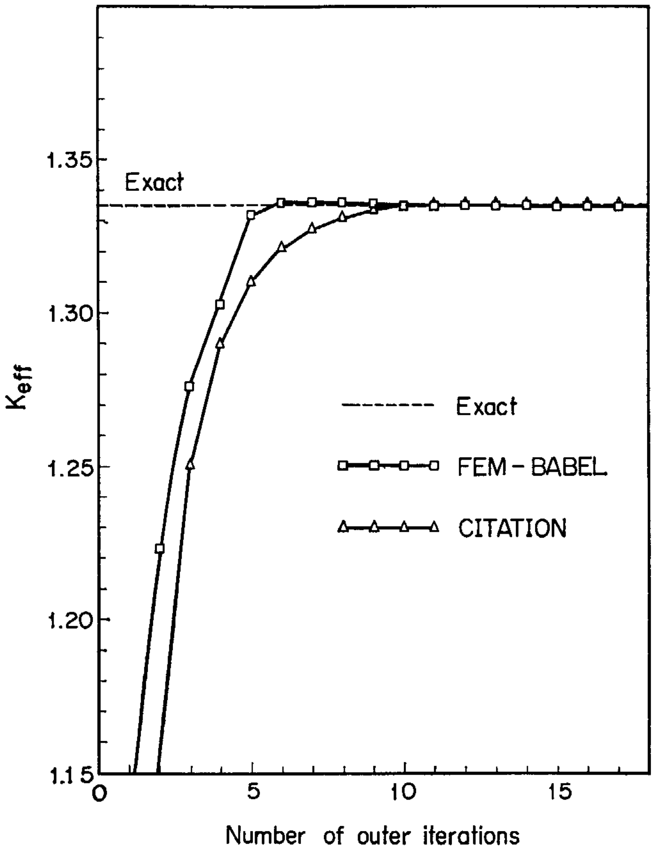


Fig. 10 Comparison of K_{eff}'s iterative perfromance for mesh size of 5 cm.

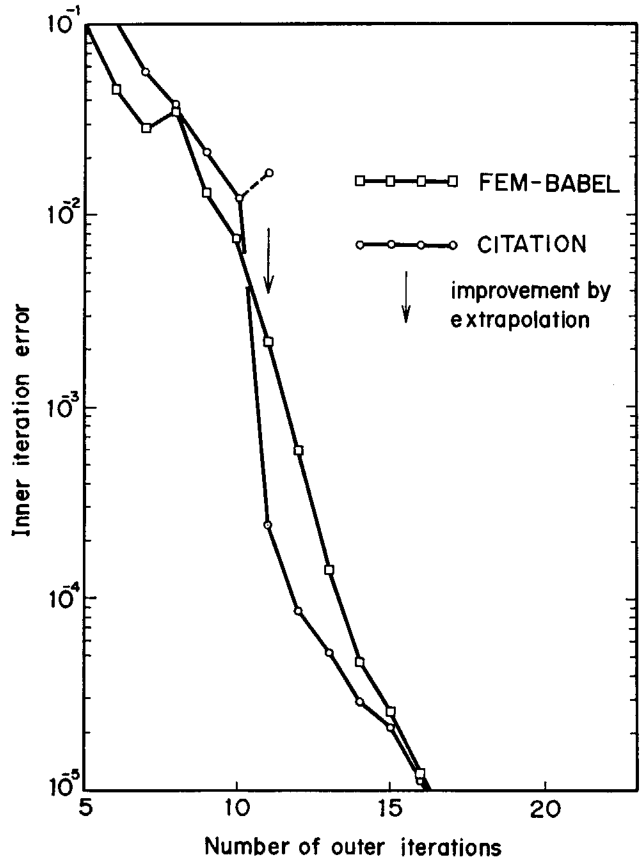


Fig. 11 Comparison of the inner iterative performance for mesh size of 5 cm.

Table 7 Comparison of the outer and inner errors for various numbers of outer iterations

	Mesh size (cm)	7th outer iteration			10th outer iteration			15th outer iteration			last outer iteration		
		K_{eff}	Outer error	Inner error	K_{eff}	Outer error	Inner error	K_{eff}	Outer error	Inner error	K_{eff}	Outer error	Inner error
FEM-BABEL	2	1.2524	2.9×10^{-2}	9.4×10^{-1}	1.3140	1.1×10^{-2}	5.1×10^{-2}	1.3367	6.8×10^{-4}	3.8×10^{-2}	1.33537	(32) 1.7×10^{-6}	8.6×10^{-6}
	5	1.33535	4.9×10^{-4}	5.7×10^{-2}	1.33470	9.5×10^{-5}	3.9×10^{-2}	1.33473	1.9×10^{-6}	7.4×10^{-4}	1.33473	(20) 4.5×10^{-8}	7.4×10^{-6}
	10	1.3311	2.3×10^{-3}	2.8×10^{-2}	1.33184	1.4×10^{-5}	2.4×10^{-3}	1.33210	2.6×10^{-6}	3.6×10^{-5}	1.33211	(20) 1.8×10^{-6}	8.5×10^{-6}
CITATION	2	1.3237	8.1×10^{-3}	1.2	1.3328	1.2×10^{-3}	3.7×10^{-2}	1.3352	2.4×10^{-4}	2.1×10^{-3}	1.33562	(22) $2.\ \times10^{-6}$	7.1×10^{-6}
	5	1.3275	6.4×10^{-3}	6.7×10^{-2}	1.33620	1.9×10^{-3}	1.3×10^{-2}	1.33623	$1.\ \times10^{-6}$	1.7×10^{-5}	1.33624	(17) $1.\ \times10^{-6}$	7.5×10^{-6}
	10	1.3291	6.5×10^{-3}	7.2×10^{-2}	1.33836	1.7×10^{-5}	3.2×10^{-3}	1.33842	$1.\ \times10^{-6}$	3.4×10^{-5}	1.33842	(18) $1.\ \times10^{-6}$	9.5×10^{-6}

Figures in parentheses are the numbers of the outer iterations at which the results have converged.

sizes decrease, while the finite differences solutions approach from larger values. From comparisons between the power distributions for various mesh sizes given in **Tables 4** to **6,** it is here ascertained that both the numerical solutions agree well with analytical results within adequate precision.

Table 8 Comparison of the computation times and storage requirements

	Mesh size					
	2 cm		5 cm		10 cm	
	Storage (words)	CPU time (sec)	Storage (words)	CPU time (sec)	Storage (words)	CPU time (sec)
FEM-BABEL	50669*	1775	5594* 10337	54 92	1469*	10
CITATION	158585	342	15228	38	4709	25

* For octant symmetric configuration

Table 9 Comparison of iterative performance at the same error range in case of mesh size of 5 cm

	Error range* (%)	K_{eff} at the range	Outer iterations	Computation time (sec.)	Relative ratio
FEM-BABEL	0.060	1.33470	6	96	1.0
CITATION	0.052	1.33620	10	22	1.4

* Analytical value ($K_{eff}=1.335506$) is taken as reference one.

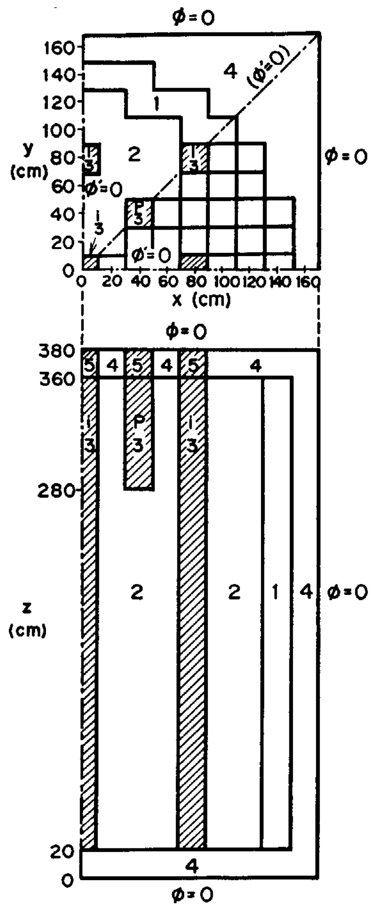


Fig. 12 Reactor configuration of the IAEA problem.

Next we investigate the convergence mechanism of the outer and inner iterations. **Figure 10** presents a comparison of the eigenvalue convergence as a function of the number of outer iterations between the finite element and finite difference calculations for mesh size of 5 cm. It is of interest to note that the finite element calculations indicate rather higher convergence than the finite difference ones. Inner iteration error for the finite element calculations given in **Fig. 11** is seen to decay rather smoothly and fast, even though these calculations use the simple SOR. On the contrary, the finite difference calculations show that the speed of error decay is drastically increased in the midway of convergence, for it comes from the fact that CITATION adopts a sophisticated technique like the flux extrapolation for the inner iterations. Considering together with the results summarized in **Table 7,** we may infer that both the outer and inner iterations of the finite element calculations converge in the same rate as the finite difference calculations. Comparison of computation times and storage requirements between two programs in **Table 8** indicates that the differences are resulted from those between their data processing procedures. In addition, from the iterative performance in **Table 9,** we find that FEM-BABEL has rather less computing cost. From the above mentioned results, we may conclude that FEM-BABEL calculations give proper solutions and are utilizable.

4.2 Calculation of a Pressurized Water Reactor

In this section we discuss the efficiency of FEM-BABEL for a real scale problem, which comes from a slight modification (for reason of consistency to the boundary condition of the finite difference method) of the three-dimensional IAEA water reactor problem[33] illustrated in **Fig. 12,** by comparing with the finite difference calculations. A comparison is performed for mesh size of 5 cm on x-y plane and 10cm in z direction up to 34 outer iterations due to saving computation time. The reference value for comparison

Table 10 Comparison of the multiplication factor and convergence between FEM-BABEL and CITATION for the IAEA problem

		FEM-BABEL	CITATION
10th outer iteration	K_{eff}	1.02261 (0.58%)	1.02075 (0.76%)
	Outer error	4.5×10^{-4}	7.4×10^{-4}
	Inner error	4.7	3.4×10^{-1}
	CPU time (min)	44.2	9.41
20th outer iteration	K_{eff}	1.02510 (0.34%)	1.02416 (0.43%)
	Outer error	1.6×10^{-4}	2.2×10^{-4}
	Inner error	3.5×10^{-2}	1.6×10^{-1}
	CPU time (min)	85.7	19.0
30th outer iteration	K_{eff}	1.02635 (0.22%)	1.02572 (0.28%)
	Outer error	9.6×10^{-5}	1.2×10^{-4}
	Inner error	2.5×10^{-2}	2.7×10^{-1}
	CPU time (min)	128.7	28.5
34th outer iteration	K_{eff}	1.02670 (0.19%)	1.02614 (0.24%)
	Outer error	8.0×10^{-5}	9.7×10^{-5}
	Inner error	2.2×10^{-5}	1.6×10^{-2}
	CPU time (min)	146.3	32.2

Percent values in K_{eff} are the fractional deviations from the reference value K_{eff}= 1.028615 calculated by CITATION with 63 outer iterations, outer error of $1.\times10^{-6}$, inner error of 6.1×10^{-4} and CPU time of 59 min.

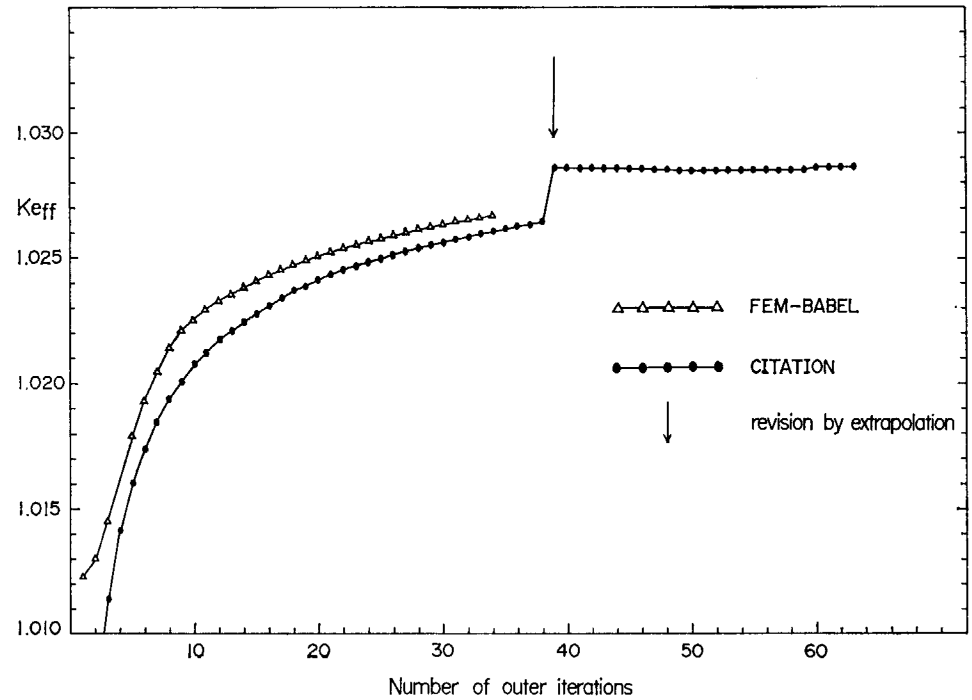


Fig. 13 Comparison of convergence performance of K_{eff} between FEM-BABEL and CITATION.

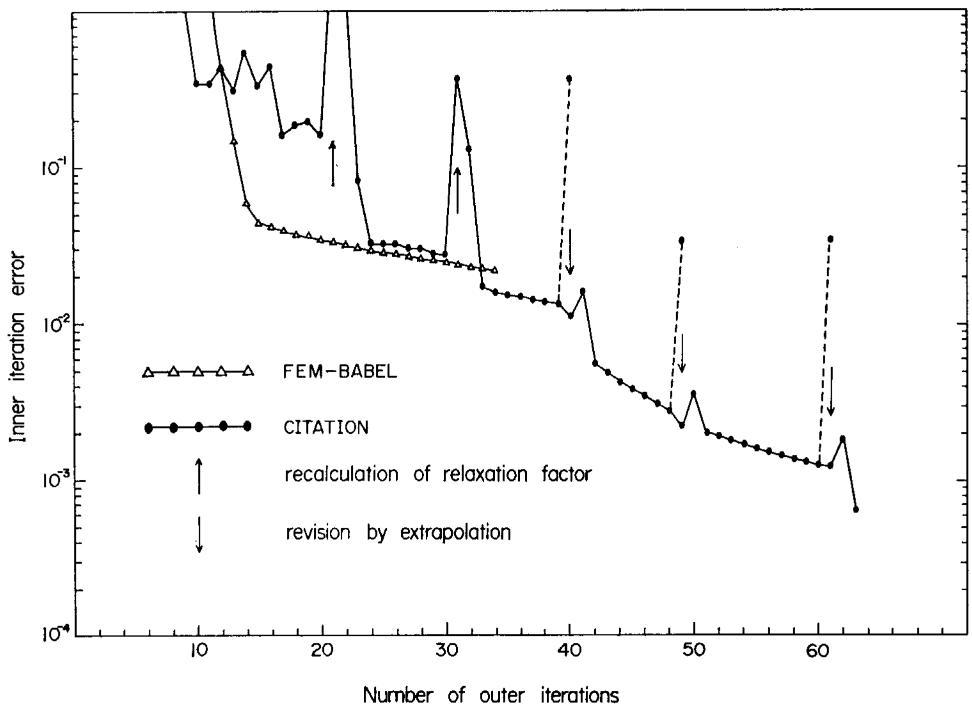


Fig. 14 Comparison of performance of inner iterations between FEM-BABEL and CITATION.

Table 11 Comparison of iterative performance at the same error range

	Error range* (%)	K_{eff} at the range	Outer iterations	Computation time (min.)	Relative ratio
FEM-BABEL	0.24	1.02615	28	120.	1.0
CITATION	0.24	1.02614	34	32.	0.27

* See about the reference value the margin below Table 10.

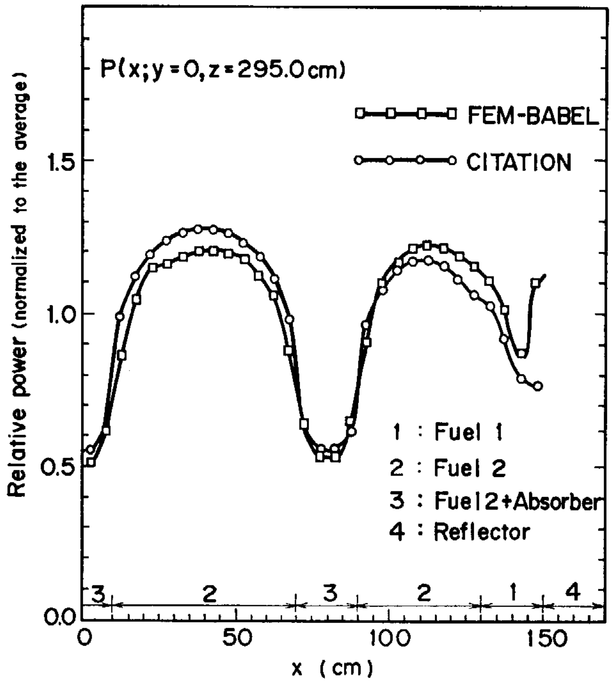


Fig. 15 Comparison of radial power distribution between FEM-BABEL and CITATION.

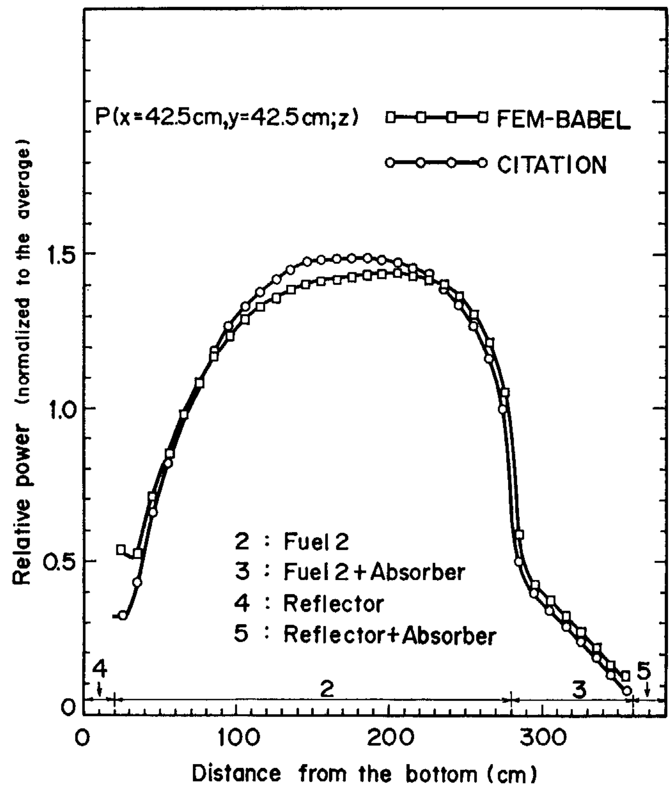


Fig. 16 Comparison of axial power distribution between FEM-BABEL and CITATION.

was obtained from the CITATION calculation for the same mesh size, when converged within the outer error of 1.0×10^{-4} and inner error of 1.0×10^{-3}.

The results on iterative performance given in **Table 10, Figs. 13** and **14** show that the finite element calculations converge in similar rates for the outer and inner errors as in the finite difference ones except for the timing comparison. The locally better performance in the finite different calculations seen in **Figs. 13**

Table 12 Comparison of the solution techniques and assumptions for the IAEA problem

Program size	FEM-BABEL	CITATION
	41 kilowords	61 kilowords
Solution method	SOR for every 4 inner iterations with $\beta=1.6$	ADI for every 4 inner iterations with adaptive β
Acceleration for inner iterations	use only SOR but not coarse mesh rebalancing technique	SOR+adaptive extrapolation
Acceleration for outer iterations	fixed extrapolation by SOR with $\beta_s=1.7$	Chebyshev extrapolation
Storage requirements and data processing	126 kilowords; only planar data in core memory	455 kilowords; all data in core memory
CPU time (see Table 11)	120 min for 28 outer iterations	32 min for 34 outer iterations

and **14** is due to the adaptive extrapolation technique as already described in the previous section. On outer iterations the finite element calculations indicate rather better performance than the finite difference ones, although the former doesn't adopt such sophisticated technique as the latter does. A comparison of the timing performance at the same error range is shown in **Table 11.**

Comparisons of power distributions illustrated in **Fig. 15** for x-y plane and **Fig. 16** for z direction show that within tolerable errors the finite element results agree globally with the finite difference ones except near the core-reflector interface. It may be inferred from a sense of reactor physics that in this point of view the finite element calculations are more reasonable than the finite difference ones.

Finally we present in **Table 12** the comparison of the solution technique and the assumption between FEM-BABEL and CITATION. The FEM-BABEL requires solely a smaller core storage and then can easily perform realistic three-dimensional calculations without being afraid of the computer restriction. We can compute using FEM-BABEL the present problem with the same core storage (for reason of adopting the plane-like SOR in FEM-BABEL: see also Section 3.1) for finer meshes in the z direction, for instance, mesh size of 5 cm. However, CITATION in our version needs so much core storage in three-dimensional problem that it cannot calculate this problem. It seems that FEM-BABEL is slower in iterations, but we consider that the iterations are substantially fast from the fact that in the core storage the disk data transfer rate is about 10 times slower than the multiply rate.

5. Conclusions

Through the numerical calculations of the exact problem and a realistic large problem, it can be concluded that FEM-BABEL is acceptable especially for the analysis of a pracitcal problem from viewpoint of reasonable computing cost. It is seen that the iteration method is very effective also for the finite element calculations. In the present calculations, we did not use the coarse mesh rebalancing technique, which is known to have an effect for analysis of practical problems (it is reported that the computation time is reduced to one fourth by adopting this technique for a two-dimensional TRIGA reactor calculation[40]). By using this technique the calculation time will be reduced to some extent compared with the present calculation.

Some improvements to the iteration method have been reported so far.[41),42] Data handlings like the concurrent iterations (solving simultaneously several planar layers in the core storage) and the parallel processing will improve the efficiency of the method.[25]

Another question for the finite element calculation is the difficulty of mesh generation for practical problems. To FEM-BABEL, the mesh generation program[28] already developed by us gives the users a help to prepare a large amount of data.

Acknowledgement

The FEM-BABEL has been programed by Mr. Atsuhiro Kubo of the Century Research Center Co. Authors are indebted to Dr. Takumi Asaoka for critical reading of this manuscript. They also wish to express many thanks to other members of the study group of the finite element method for their discussions and encouragements: to Mr. Toichiro Fujimura of our laboratory and Mr. Tadahiro Ohnishi of Hitachi Ltd.

This work was performed in partial cooperation with Sumitomo Heavy Industries, Ltd.

References

1) Strang G. and Fix G. J.: "An Analysis of the Finite Element Method," Prentice-Hall (1971).
2) Zienkiewicz O. C.: "The Finite Element Method in Engineering and Science," McGraw-Hill (1971).
3) Mitchell A. R. and Wait R.: "Finite Element Method in Partial Differential Equations," John Wiley & Sons (1977).
4) Ohnishi T.: "Application of the Finite Element Solution Techniques to Neutron Diffusion and Transport Equations," CONF-710302, Vol. II, 273 (1971).
5) Kaper H. G., Leaf G. K. and Lindeman A. J.: "A timing comparison study for some high order finite element approximation procedures and a low order finite difference approximation procedures for the numerical solution of the multigroup neutron diffusion equation," *Nucl. Sci. Eng.*, **49**, 27 (1972).
6) Semenza L. A., Lewis E. E. and Rossow E. C.: "The application of the finite element method to the multigroup neutron diffusion equation," *Nucl. Sci. Eng.*, **47**, 302 (1972).
7) Kang C. M. and Hansen K. F.: "Finite element methods for space-time reactor analysis," *Nucl. Sci. Eng.*, **51**, 456 (1973).
8) Deppe L. O. and Hansen K. F.: "Application of the finite element method to two-dimensional diffusion problems," *Nucl. Sci. Eng.*, **54**, 456 (1974).
9) Schmidt F. A. R. and Frank H. P.: "Power reactor calculations with the finite element program FEM 2D," *Nucl. Sci. Eng.*, **56**, 431 (1975).
10) Franke H. P., Sapper E. and Schmidt F. A. R.: "Two and three-dimensional reactor physics calculations with the finite element method," *Atomkernenergie*, **26**, 158 (1975).
11) Yang S. and Henry A. F.: "A finite element synthesis method," *Nucl. Sci. Eng.*, **59**, 63 (1976).
12) Antila M. and Silvennoinen P.: "Coarse mesh finite element analysis of the core power distribution in hexagonal geometry," *Kernenergie*, **19**, 218 (1976).
13) Biswas D., Ram K. S. and Rao S. S.: "Application of 'natural coordinate system' in the finite element solution of multigroup neutron diffusion equation," *Annals of Nuclear Energy*, **3**, 465 (1976).
14) Hennart J. P. and Mund E. H.: "Singularities in the finite element approximation of two-dimensional diffusion problems," *Nucl. Sci. Eng.*, **62**, 55 (1977).
15) Tomlinson E. T. and Robinson J. C.: "Solution of the finite element diffusion and P_1 equations by iteration," *Nucl. Sci. Eng.*, **63**, 167 (1977).
16) Komoriya H. and Walters W. F.: "The energy-dependent finite element method for two-dimensional diffusion problems," *Nucl. Sci. Eng.*, **64**, 576 (1977).
17) Miller Jr. W. F., Lewis E. E. and Rossow E. E.: "The application of phase-space finite elements to the one-dimensional neutron transport equation," *Nucl. Sci. Eng.*, **51**, 148 (1973).
18) Reed W. H., Hill T. R., Brinkley F. W. and Lathrop K. D.: "TRIPLET: A Two-Dimensional, Multigroup, Triangular Mesh, Planar Geometry, Explicit Transport Code," LA-5428-MS (1973).
19) Kaper H. G., Leaf G. K. and Lindeman A. J.: "Formulation of a Rit-Galerkin type procedure for the approximate solution of the neutron transport equation," *J. Math. Anal. Appl.*, **50**, 42 (1975).
20) Ackroyd R. T.: "The Finite Element Method for Neutron Transport: 1. the Even Parity Equation for Neutron Transport and its Maximum Variational Principles," TRG Report 2705 (R) (1975).
21) Lewis E. E., Miller Jr. W. F. and Henry T. P.: "A two-dimensional finite element method for integral neutron transport calculations," *Nucl. Sci. Eng.*, **58**, 203 (1975).
22) Martin W. R. and Duderstadt J. J.: "Finite element solutions of the neutron transport equation with applications to strong heterogeneities," *Nucl. Sci. Eng.*, **62**, 371 (1977).
23) Fujimura T., Tsutsui T., Horikami K., Nakahara Y. and Ohnishi T.: "Application of finite element method to two-dimensional multi-group neutron transport equation in cylindrical geometry," *J. Nucl. Sci. Technol.*, **14**, 541 (1977).
24) Rose D. J. and Willoughby R. A. (eds.): "Sparse Matrices and their Applications," Plenum Press (1972).
25) Hageman L. A.: "The Solution of Linear Equations Resulting from Finite Element Discretizations of Multi-Dimensional Boundary Value Problems," WAPD-TM-1209 (1975).

26) Hutula D. N. and Zeiler S. M.: "MESH3D: A Three-Dimensional Finite Element Mesh Generator Program for Eight-Node Isoparametric Elements," WAPD-TM-1079 (1973).
27) Jones R. E.: "The QMESH Mesh Generation Package," SAND-75-5884 (1975).
28) Ise T. and Yamazaki T.: "LOOM-P: a Finite Element Mesh Generation Program with On-Line Graphic Display," JAERI-M 7119 (1977).
29) Desai C. S. and Abel J. F.: "Introduction to the Finite Element Method," Van Nostrand (1972).
30) Abu-Shumays I. K. and Hageman L. A.: "Development and Comparison of Practical Discretization Methods for the Neutron Diffusion Equation over General Quadrilateral Partitions," CONF-750413, I-117 (1976).
31) Vondy D. R. and Fowler T. B.: "On the Difference Equations to Represent Neutorn Transport with Diffusion Theory," CONF-760622-8 (1976).
32) Franke H.P.: "Untersuchungen zur Numerischen Lœsung Dreidimensionaler Stationarer Diffusionsgleichungen nach der Mettode der Finite Elemente," IKE 4-53 (1976).
33) Misfeldt I.: "Solution of the Multigroup Neutron Diffusion Equations by the Finite Element Method," RISO-M-1809 (1975).
34) Varga R. A.: "Matrix Iterative Analysis," Prentice-Hall (1962).
35) Froelich R.: "A Theoretical Foundation for Coarse Mesh Variational Techniques," Intern. Conf. Research Reactor Utilization and Reactor Mathematics, Mexico, 219 (1967).
36) Tewarson R. P.: "Sparse Matrices," Academic Press (1973).
37) Fujimura T., Nishida T. and Asai K. (eds.): "Manual on JSSL (JAERI Scientific Subroutine Library)," JAERI-M 7102 (1977) [In Japanese]
38) Mynatt F. R., Engle Jr. W. W. et al.: "The DOT III, Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280 (1973).
39) Fowler T. B., Vondy D. R. and Cunningham G. W.: "Nuclear Reactor Core Analysis Code: CITATION," ORNL-TM-2496, Rev. 2 (1971).
40) Froelich R.: "Flux Synthesis Methods versus Difference Approximation Methods for the Efficient Determination of Neutron Flux Distributions in Fast and Thermal Reactors," IAEA-SM-154/14 in "Numerical Reactor Calculations," IAEA (1972).
41) Conrad V. and Wallach Y.: "A faster SSOR algorithm," *Numer. Math.*, **27**, 371 (1977).
42) Concus P., Golub G. H. and O'Leary D. P.: "A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations," LBL-4604 (1975).

Appendix

We present here the list of input cards together with control cards, for FEM-BABEL calculations on the representative problems shown in the present report. Readers can therefore ascertain the results given in the text.

A-1 Input cards set up for the quarter symmetric geometry with mesh size of 5 cm for the exact problem
A-2 Input cards set up for a new case of the IAEA problem
A-3 Input cards set up for a restart case of the IAEA problem

A-1 Input cards set up for the quarter symmetric geometry with mesh size of 5 cm for the exact problem

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

¥NQ         P666,                                                            /
                                                                    T.5/TIME 15M
                                                                    C.3/CORE 192
                                                                    W.1/PAGE  80
                                                                    P.0/PCH    0
¥GJOB     S521130,T,ISE,431,11,FEMBABEL                                /JOB-CARD
*HFORT
C       PROGRAM  FEM-BABEL
        COMMON     IA( 70000)
        CALL CHGFLO(M,N)                                            BABEL  7
C                            * FACOM *                              BABEL  9
        IA(1)= 70000                                                IA
C                                                                   BABEL 12
        CALL DTLIST
        CALL MAIN1                                                  BABEL 13
        STOP                                                        BABEL 14
        END                                                         BABEL 15
*HLIEDRUN RFNAME=J1223,BABELA,    EDIT=YES,         SIZE=30
*TPDISK F01,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F02,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F03,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F04,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F08,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F09,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F10,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F11,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F12,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F13,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F14,DISP=DELETE,RSIZE=900,BSIZE=6300
*DATA
 2 GROUPS 3 DIM,,BARE REACTOR,DX=5CM,QUARTER SYMMET,,ALL SQUARE MESHES
 1*                                                                  1*
          2         1        121        11        100            1  1- 6
          1         1         21         0          1            3  7-12
        300        13          1         1          0            1 13-18
          1         1          1         0          0            0 19-24
          0         0          0         2         11           11 25-30
          0         3          0  T                                31- 33
 2*                                                                  2*
    1.0 -05   1.0 -05    1.7       1.0       3.0       8.0          1- 6
    3.3 +10  T                                                      7- 9
 6*                                                                  6*
 9I  0.0       50.0
 7*                                                                  7*
 9I  0.0       50.0       T
 8*                                                                  8*
         10
 9*                                                                  9*
   50.0      T
10*                                                                 10*
  F       1  T
11*                                                                 11*
  F       1  T
12*                                                                 12*
   1.2334-02  2.6800    3.0834-02  1.3785-02  4.0792-02

   1.0080-02  1.5788    2.5200-02  1.4496-02    0.0       T
13*                                                                 13*
    0.575      0.425     T
15*                                                                 15*NBPOT
 8I        11       110 9I       111         121  T
17*
  F       1  T
18*
  F       1  T
19*
          1         5         10  T
¥JEND
```

A-2 Input cards set up for a new case of the IAEA problem

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8


¥NO        B333,                                                          /
                                                                    T.7/TIME 60M
                                                                    C.4/CORE 256
                                                                    W.1/PAGE  80
                                                                    P.0/PCH    0
                                                                   ,LRG/
¥GJOB    S521130,T,ISE,431,11,FEMBABEL                                  /JOB-CARD
*HFORT
C       PROGRAM  FEM-BABEL
        COMMON     IA(140000)
        CALL CHGFLO(M,N)                                                BABEL   7
C                          * FACOM *                                    BABEL   9
        IA(1)=140000
C                                                                       BABEL  12
        CALL DTLIST
        CALL MAIN1                                                      BABEL  13
        STOP                                                            BABEL  14
        END                                                             BABEL  15
        SUBROUTINE ACCEL(P,S)
        DIMENSION P(1),S(1)
        COMMON     A(1)
        EQUIVALENCE (A(142),NPALL),(A(128),RAMDA),(A(253),SORF)
        T=0.0
        DO 1 I=1,NPALL
        P(I)=P(I)+1.70*(S(I)-P(I))
      1 T=T+P(I)
        DO 2 I=1,NPALL
      2 P(I)=P(I)/T
        RETURN
        END
*HLIEDRUN  RFNAME=J1223,BABELA,  EDIT=YES,UPDT=YES,SIZE=30             BABELA
*TPDISK F01,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F02,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F03,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F04,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F08,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F09,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F10,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F11,DISP=DELETE,RSIZE=900,BSIZE=6300
*TAPE  F12,J1223,B333,NEW,010215
*TPDISK F13,DISP=DELETE,RSIZE=900,BSIZE=6300
*TPDISK F14,DISP=DELETE,RSIZE=900,BSIZE=6300
*DATA
 3D-IAEA(MODIFIED) FOR COMPARISON WITH CITATION, DX=5CM DZ=10CM
 1*                 2            1           630          39          595 1 -05
          5         4            5            35           0            0 6 -11
          6       100           36             1           1            0 12-17
          1 3R      1 6R         0             0           0            0 18-30
          1         2            0  T                                     31-33
 2*            1.0 -04     1.0 -03           1.6       3800.        55.0  1 - 5
          8.    3.265+10  T
 3*
          1         2            3             3
          2         4            5             3
          3         5            6             6
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

           4           7           8           5
           5           8           9           6
           6           9          10          10
           7          11          12           8
           8          12          13           9
           9          13          14          10
          10          14          15          15
          11          16          17          12
          12          17          18          13
          13          18          19          14
          14          19          20          15
          15          20          21          21
          16          22          23          17
          17          23          24          18
          18          24          25          19
          19          25          26          20
          20          26          27          21
          21          27          28          28
          22          29          30          23
          23          30          31          24
          24          31          32          25
          25          32          33          26
          26          33          34          27
          27          34          35          28
          28          35          36          36
          29          37          38          30
          30          38          39          31
          31          39          40          32
          32          40          41          33
          33          41          42          34
          34          42          43          35
          35          43          44          36
          36          44          45          45
          37          46          47          38
          38          47          48          39
          39          48          49          40
          40          49          50          41
          41          50          51          42
          42          51          52          43
          43          52          53          44
          44          53          54          45
          45          54          55          55
          46          56          57          47
          47          57          58          48
          48          58          59          49
          49          59          60          50
          50          60          61          51
          51          61          62          52
          52          62          63          53
          53          63          64          54
          54          64          65          55
          55          65          66          66
          56          67          68          57
          57          68          69          58
          58          69          70          59
```

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

59	70	71	60
60	71	72	61
61	72	73	62
62	73	74	63
63	74	75	64
64	75	76	65
65	76	77	66
66	77	78	78
67	79	80	68
68	80	81	69
69	81	82	70
70	82	83	71
71	83	84	72
72	84	85	73
73	85	86	74
74	86	87	75
75	87	88	76
76	88	89	77
77	89	90	78
78	90	91	91
79	92	93	80
80	93	94	81
81	94	95	82
82	95	96	83
83	96	97	84
84	97	98	85
85	98	99	86
86	99	100	87
87	100	101	88
88	101	102	89
89	102	103	90
90	103	104	91
91	104	105	105
92	106	107	93
93	107	108	94
94	108	109	95
95	109	110	96
96	110	111	97
97	111	112	98
98	112	113	99
99	113	114	100
100	114	115	101
101	115	116	102
102	116	117	103
103	117	118	104
104	118	119	105
105	119	120	120
106	121	122	107
107	122	123	108
108	123	124	109
109	124	125	110
110	125	126	111
111	126	127	112
112	127	128	113
113	128	129	114

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

114	129	130	115
115	130	131	116
116	131	132	117
117	132	133	118
118	133	134	119
119	134	135	120
120	135	136	136
121	137	138	122
122	138	139	123
123	139	140	124
124	140	141	125
125	141	142	126
126	142	143	127
127	143	144	128
128	144	145	129
129	145	146	130
130	146	147	131
131	147	148	132
132	148	149	133
133	149	150	134
134	150	151	135
135	151	152	136
136	152	153	153
137	154	155	138
138	155	156	139
139	156	157	140
140	157	158	141
141	158	159	142
142	159	160	143
143	160	161	144
144	161	162	145
145	162	163	146
146	163	164	147
147	164	165	148
148	165	166	149
149	166	167	150
150	167	168	151
151	168	169	152
152	169	170	153
153	170	171	171
154	172	173	155
155	173	174	156
156	174	175	157
157	175	176	158
158	176	177	159
159	177	178	160
160	178	179	161
161	179	180	162
162	180	181	163
163	181	182	164
164	182	183	165
165	183	184	166
166	184	185	167
167	185	186	168
168	186	187	169

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8
```

169	187	188	170
170	188	189	171
171	189	190	190
172	191	192	173
173	192	193	174
174	193	194	175
175	194	195	176
176	195	196	177
177	196	197	178
178	197	198	179
179	198	199	180
180	199	200	181
181	200	201	182
182	201	202	183
183	202	203	184
184	203	204	185
185	204	205	186
186	205	206	187
187	206	207	188
188	207	208	189
189	208	209	190
190	209	210	210
191	211	212	192
192	212	213	193
193	213	214	194
194	214	215	195
195	215	216	196
196	216	217	197
197	217	218	198
198	218	219	199
199	219	220	200
200	220	221	201
201	221	222	202
202	222	223	203
203	223	224	204
204	224	225	205
205	225	226	206
206	226	227	207
207	227	228	208
208	228	229	209
209	229	230	210
210	230	231	231
211	232	233	212
212	233	234	213
213	234	235	214
214	235	236	215
215	236	237	216
216	237	238	217
217	238	239	218
218	239	240	219
219	240	241	220
220	241	242	221
221	242	243	222
222	243	244	223
223	244	245	224

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

       224       245       246       225
       225       246       247       226
       226       247       248       227
       227       248       249       228
       228       249       250       229
       229       250       251       230
       230       251       252       231
       231       252       253       253
       232       254       255       233
       233       255       256       234
       234       256       257       235
       235       257       258       236
       236       258       259       237
       237       259       260       238
       238       260       261       239
       239       261       262       240
       240       262       263       241
       241       263       264       242
       242       264       265       243
       243       265       266       244
       244       266       267       245
       245       267       268       246
       246       268       269       247
       247       269       270       248
       248       270       271       249
       249       271       272       250
       250       272       273       251
       251       273       274       252
       252       274       275       253
       253       275       276       276
       254       277       278       255
       255       278       279       256
       256       279       280       257
       257       280       281       258
       258       281       282       259
       259       282       283       260
       260       283       284       261
       261       284       285       262
       262       285       286       263
       263       286       287       264
       264       287       288       265
       265       288       289       266
       266       289       290       267
       267       290       291       268
       268       291       292       269
       269       292       293       270
       270       293       294       271
       271       294       295       272
       272       295       296       273
       273       296       297       274
       274       297       298       275
       275       298       299       276
       276       299       300       300
       277       301       302       278
       278       302       303       279
```

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

279	303	304	280
280	304	305	281
281	305	306	282
282	306	307	283
283	307	308	284
284	308	309	285
285	309	310	286
286	310	311	287
287	311	312	288
288	312	313	289
289	313	314	290
290	314	315	291
291	315	316	292
292	316	317	293
293	317	318	294
294	318	319	295
295	319	320	296
296	320	321	297
297	321	322	298
298	322	323	299
299	323	324	300
300	324	325	325
301	326	327	302
302	327	328	303
303	328	329	304
304	329	330	305
305	330	331	306
306	331	332	307
307	332	333	308
308	333	334	309
309	334	335	310
310	335	336	311
311	336	337	312
312	337	338	313
313	338	339	314
314	339	340	315
315	340	341	316
316	341	342	317
317	342	343	318
318	343	344	319
319	344	345	320
320	345	346	321
321	346	347	322
322	347	348	323
323	348	349	324
324	349	350	325
325	350	351	351
326	352	353	327
327	353	354	328
328	354	355	329
329	355	356	330
330	356	357	331
331	357	358	332
332	358	359	333
333	359	360	334

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

          334       360       361       335
          335       361       362       336
          336       362       363       337
          337       363       364       338
          338       364       365       339
          339       365       366       340
          340       366       367       341
          341       367       368       342
          342       368       369       343
          343       369       370       344
          344       370       371       345
          345       371       372       346
          346       372       373       347
          347       373       374       348
          348       374       375       349
          349       375       376       350
          350       376       377       351
          351       377       378       378
          352       379       380       353
          353       380       381       354
          354       381       382       355
          355       382       383       356
          356       383       384       357
          357       384       385       358
          358       385       386       359
          359       386       387       360
          360       387       388       361
          361       388       389       362
          362       389       390       363
          363       390       391       364
          364       391       392       365
          365       392       393       366
          366       393       394       367
          367       394       395       368
          368       395       396       369
          369       396       397       370
          370       397       398       371
          371       398       399       372
          372       399       400       373
          373       400       401       374
          374       401       402       375
          375       402       403       376
          376       403       404       377
          377       404       405       378
          378       405       406       406
          379       407       408       360
          380       408       409       381
          381       409       410       382
          382       410       411       383
          383       411       412       384
          384       412       413       385
          385       413       414       386
          386       414       415       387
          387       415       416       388
          388       416       417       389
```

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

389	417	418	390
390	418	419	391
391	419	420	392
392	420	421	393
393	421	422	394
394	422	423	395
395	423	424	396
396	424	425	397
397	425	426	398
398	426	427	399
399	427	428	400
400	428	429	401
401	429	430	402
402	430	431	403
403	431	432	404
404	432	433	405
405	433	434	406
406	434	435	435
407	436	437	408
408	437	438	409
409	438	439	410
410	439	440	411
411	440	441	412
412	441	442	413
413	442	443	414
414	443	444	415
415	444	445	416
416	445	446	417
417	446	447	418
418	447	448	419
419	448	449	420
420	449	450	421
421	450	451	422
422	451	452	423
423	452	453	424
424	453	454	425
425	454	455	426
426	455	456	427
427	456	457	428
428	457	458	429
429	458	459	430
430	459	460	431
431	460	461	432
432	461	462	433
433	462	463	434
434	463	464	435
435	464	465	465
436	466	467	437
437	467	468	438
438	468	469	439
439	469	470	440
440	470	471	441
441	471	472	442
442	472	473	443
443	473	474	444

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

444	474	475	445
445	475	476	446
446	476	477	447
447	477	478	448
448	478	479	449
449	479	480	450
450	480	481	451
451	481	482	452
452	482	483	453
453	483	484	454
454	484	485	455
455	485	486	456
456	486	487	457
457	487	488	458
458	488	489	459
459	489	490	460
460	490	491	461
461	491	492	462
462	492	493	463
463	493	494	464
464	494	495	465
465	495	496	496
466	497	498	467
467	498	499	468
468	499	500	469
469	500	501	470
470	501	502	471
471	502	503	472
472	503	504	473
473	504	505	474
474	505	506	475
475	506	507	476
476	507	508	477
477	508	509	478
478	509	510	479
479	510	511	480
480	511	512	481
481	512	513	482
482	513	514	483
483	514	515	484
484	515	516	485
485	516	517	486
486	517	518	487
487	518	519	488
488	519	520	489
489	520	521	490
490	521	522	491
491	522	523	492
492	523	524	493
493	524	525	494
494	525	526	495
495	526	527	496
496	527	528	528
497	529	530	498
498	530	531	499

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8
```

499	531	532	500
500	532	533	501
501	533	534	502
502	534	535	503
503	535	536	504
504	536	537	505
505	537	538	506
506	538	539	507
507	539	540	508
508	540	541	509
509	541	542	510
510	542	543	511
511	543	544	512
512	544	545	513
513	545	546	514
514	546	547	515
515	547	548	516
516	548	549	517
517	549	550	518
518	550	551	519
519	551	552	520
520	552	553	521
521	553	554	522
522	554	555	523
523	555	556	524
524	556	557	525
525	557	558	526
526	558	559	527
527	559	560	528
528	560	561	561
529	562	563	530
530	563	564	531
531	564	565	532
532	565	566	533
533	566	567	534
534	567	568	535
535	568	569	536
536	569	570	537
537	570	571	538
538	571	572	539
539	572	573	540
540	573	574	541
541	574	575	542
542	575	576	543
543	576	577	544
544	577	578	545
545	578	579	546
546	579	580	547
547	580	581	548
548	581	582	549
549	582	583	550
550	583	584	551
551	584	585	552
552	585	586	553
553	586	587	554

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

         554       587       588       555
         555       588       589       556
         556       589       590       557
         557       590       591       558
         558       591       592       559
         559       592       593       560
         560       593       594       561
         561       594       595       595
         562       596       597       563
         563       597       598       564
         564       598       599       565
         565       599       600       566
         566       600       601       567
         567       601       602       568
         568       602       603       569
         569       603       604       570
         570       604       605       571
         571       605       606       572
         572       606       607       573
         573       607       608       574
         574       608       609       575
         575       609       610       576
         576       610       611       577
         577       611       612       578
         578       612       613       579
         579       613       614       580
         580       614       615       581
         581       615       616       582
         582       616       617       583
         583       617       618       584
         584       618       619       585
         585       619       620       586
         586       620       621       587
         587       621       622       588
         588       622       623       589
         589       623       624       590
         590       624       625       591
         591       625       626       592
         592       626       627       593
         593       627       628       594
         594       628       629       595
         595       629       630       630
 4*                 0.        5.        5.       10.       10.
         10.       15.       15.       15.       15.       20.
         20.       20.       20.       20.       25.       25.
         25.       25.       25.       25.       30.       30.
         30.       30.       30.       30.       30.       35.
         35.       35.       35.       35.       35.       35.
         35.       40.       40.       40.       40.       40.
         40.       40.       40.       40.       45.       45.
         45.       45.       45.       45.       45.       45.
         45.       45.       50.       50.       50.       50.
         50.       50.       50.       50.       50.       50.
         50.       55.       55.       55.       55.       55.
         55.       55.       55.       55.       55.       55.
```

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

55.	60.	60.	60.	60.	60.
60.	60.	60.	60.	60.	60.
60.	60.	65.	65.	65.	65.
65.	65.	65.	65.	65.	65.
65.	65.	65.	65.	70.	70.
70.	70.	70.	70.	70.	70.
70.	70.	70.	70.	70.	70.
70.	75.	75.	75.	75.	75.
75.	75.	75.	75.	75.	75.
75.	75.	75.	75.	75.	80.
80.	80.	80.	80.	80.	80.
80.	80.	80.	80.	80.	80.
80.	80.	80.	80.	85.	85.
85.	85.	85.	85.	85.	85.
85.	85.	85.	85.	85.	85.
85.	85.	85.	85.	90.	90.
90.	90.	90.	90.	90.	90.
90.	90.	90.	90.	90.	90.
90.	90.	90.	90.	90.	95.
95.	95.	95.	95.	95.	95.
95.	95.	95.	95.	95.	95.
95.	95.	95.	95.	95.	95.
95.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	105.	105.
105.	105.	105.	105.	105.	105.
105.	105.	105.	105.	105.	105.
105.	105.	105.	105.	105.	105.
105.	105.	110.	110.	110.	110.
110.	110.	110.	110.	110.	110.
110.	110.	110.	110.	110.	110.
110.	110.	110.	110.	110.	110.
110.	115.	115.	115.	115.	115.
115.	115.	115.	115.	115.	115.
115.	115.	115.	115.	115.	115.
115.	115.	115.	115.	115.	115.
115.	120.	120.	120.	120.	120.
120.	120.	120.	120.	120.	120.
120.	120.	120.	120.	120.	120.
120.	120.	120.	120.	120.	120.
120.	120.	125.	125.	125.	125.
125.	125.	125.	125.	125.	125.
125.	125.	125.	125.	125.	125.
125.	125.	125.	125.	125.	125.
125.	125.	125.	125.	130.	130.
130.	130.	130.	130.	130.	130.
130.	130.	130.	130.	130.	130.
130.	130.	130.	130.	130.	130.
130.	130.	130.	130.	130.	130.
130.	135.	135.	135.	135.	135.
135.	135.	135.	135.	135.	135.
135.	135.	135.	135.	135.	135.
135.	135.	135.	135.	135.	135.
135.	135.	135.	135.	135.	140.

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

          140.      140.      140.      140.      140.      140.
          140.      140.      140.      140.      140.      140.
          140.      140.      140.      140.      140.      140.
          140.      140.      140.      140.      140.      140.
          140.      140.      140.      140.      145.      145.
          145.      145.      145.      145.      145.      145.
          145.      145.      145.      145.      145.      145.
          145.      145.      145.      145.      145.      145.
          145.      145.      145.      145.      145.      145.
          145.      145.      145.      145.      150.      150.
          150.      150.      150.      150.      150.      150.
          150.      150.      150.      150.      150.      150.
          150.      150.      150.      150.      150.      150.
          150.      150.      150.      150.      150.      150.
          150.      150.      150.      150.      150.      155.
          155.      155.      155.      155.      155.      155.
          155.      155.      155.      155.      155.      155.
          155.      155.      155.      155.      155.      155.
          155.      155.      155.      155.      155.      155.
          155.      155.      155.      155.      155.      155.
          155.      160.      160.      160.      160.      160.
          160.      160.      160.      160.      160.      160.
          160.      160.      160.      160.      160.      160.
          160.      160.      160.      160.      160.      160.
          160.      160.      160.      160.      160.      160.
          160.      160.      160.      160.      165.      165.
          165.      165.      165.      165.      165.      165.
          165.      165.      165.      165.      165.      165.
          165.      165.      165.      165.      165.      165.
          165.      165.      165.      165.      165.      165.
          165.      165.      165.      165.      165.      165.
          165.      165.      170.      170.      170.      170.
          170.      170.      170.      170.      170.      170.
          170.      170.      170.      170.      170.      170.
          170.      170.      170.      170.      170.      170.
          170.      170.      170.      170.      170.      170.
          170.      170.      170.      170.      170.      170.
          170.
 5*                   0.        0.        5.        0.        5.
           10.        0.        5.       10.       15.        0.
            5.       10.       15.       20.        0.        5.
           10.       15.       20.       25.        0.        5.
           10.       15.       20.       25.       30.        0.
            5.       10.       15.       20.       25.       30.
           35.        0.        5.       10.       15.       20.
           25.       30.       35.       40.        0.        5.
           10.       15.       20.       25.       30.       35.
           40.       45.        0.        5.       10.       15.
           20.       25.       30.       35.       40.       45.
           50.        0.        5.       10.       15.       20.
           25.       30.       35.       40.       45.       50.
           55.        0.        5.       10.       15.       20.
           25.       30.       35.       40.       45.       50.
           55.       60.        0.        5.       10.       15.
           20.       25.       30.       35.       40.       45.
```

....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

50.	55.	60.	65.	0.	5.
10.	15.	20.	25.	30.	35.
40.	45.	50.	55.	60.	65.
70.	0.	5.	10.	15.	20.
25.	30.	35.	40.	45.	50.
55.	60.	65.	70.	75.	0.
5.	10.	15.	20.	25.	30.
35.	40.	45.	50.	55.	60.
65.	70.	75.	80.	0.	5.
10.	15.	20.	25.	30.	35.
40.	45.	50.	55.	60.	65.
70.	75.	80.	85.	0.	5.
10.	15.	20.	25.	30.	35.
40.	45.	50.	55.	60.	65.
70.	75.	80.	85.	90.	0.
5.	10.	15.	20.	25.	30.
35.	40.	45.	50.	55.	60.
65.	70.	75.	80.	85.	90.
95.	0.	5.	10.	15.	20.
25.	30.	35.	40.	45.	50.
55.	60.	65.	70.	75.	80.
85.	90.	95.	100.	0.	5.
10.	15.	20.	25.	30.	35.
40.	45.	50.	55.	60.	65.
70.	75.	80.	85.	90.	95.
100.	105.	0.	5.	10.	15.
20.	25.	30.	35.	40.	45.
50.	55.	60.	65.	70.	75.
80.	85.	90.	95.	100.	105.
110.	0.	5.	10.	15.	20.
25.	30.	35.	40.	45.	50.
55.	60.	65.	70.	75.	80.
85.	90.	95.	100.	105.	110.
115.	0.	5.	10.	15.	20.
25.	30.	35.	40.	45.	50.
55.	60.	65.	70.	75.	80.
85.	90.	95.	100.	105.	110.
115.	120.	0.	5.	10.	15.
20.	25.	30.	35.	40.	45.
50.	55.	60.	65.	70.	75.
80.	85.	90.	95.	100.	105.
110.	115.	120.	125.	0.	5.
10.	15.	20.	25.	30.	35.
40.	45.	50.	55.	60.	65.
70.	75.	80.	85.	90.	95.
100.	105.	110.	115.	120.	125.
130.	0.	5.	10.	15.	20.
25.	30.	35.	40.	45.	50.
55.	60.	65.	70.	75.	80.
85.	90.	95.	100.	105.	110.
115.	120.	125.	130.	135.	0.
5.	10.	15.	20.	25.	30.
35.	40.	45.	50.	55.	60.
65.	70.	75.	80.	85.	90.
95.	100.	105.	110.	115.	120.

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

          125.         130.         135.         140.           0.           5.
           10.          15.          20.          25.          30.          35.
           40.          45.          50.          55.          60.          65.
           70.          75.          80.          85.          90.          95.
          100.         105.         110.         115.         120.         125.
          130.         135.         140.         145.           0.           5.
           10.          15.          20.          25.          30.          35.
           40.          45.          50.          55.          60.          65.
           70.          75.          80.          85.          90.          95.
          100.         105.         110.         115.         120.         125.
          130.         135.         140.         145.         150.           0.
            5.          10.          15.          20.          25.          30.
           35.          40.          45.          50.          55.          60.
           65.          70.          75.          80.          85.          90.
           95.         100.         105.         110.         115.         120.
          125.         130.         135.         140.         145.         150.
          155.           0.           5.          10.          15.          20.
           25.          30.          35.          40.          45.          50.
           55.          60.          65.          70.          75.          80.
           85.          90.          95.         100.         105.         110.
          115.         120.         125.         130.         135.         140.
          145.         150.         155.         160.           0.           5.
           10.          15.          20.          25.          30.          35.
           40.          45.          50.          55.          60.          65.
           70.          75.          80.          85.          90.          95.
          100.         105.         110.         115.         120.         125.
          130.         135.         140.         145.         150.         155.
          160.         165.           0.           5.          10.          15.
           20.          25.          30.          35.          40.          45.
           50.          55.          60.          65.          70.          75.
           80.          85.          90.          95.         100.         105.
          110.         115.         120.         125.         130.         135.
          140.         145.         150.         155.         160.         165.
          170.   T
8*                  2           26            8            2
9*              2.0 +01     2.6 +02      8.0 +01      2.0 +01   T
10*
3R         518R         2 6R         2            3 6R         2 2R         3
6R         2 3R         3 6R         2 4R         350R         2 2R         5
12R        2            5 2R         512R         2 2R         5 2R         5
12R        2 3R         5 2R         512R         2 4R         514R         2
5R         114R         2 6R         114R         2 7R         114R         2
8R         1 6R         212R         1 5R         4 6R         212R         1
6R         4 6R         212R         1 7R         4 6R         212R         1
8R         410R         117R         410R         118R         410R         1
19R        410R         120R         4   F        4   T
11*
           4            4            4            4            4
           1            2            2            4            3
           1            2            3            4            3
           4            4            5            4            5   T
12*
       0.00         1.5          0.00         0.01         0.02                  F1
       0.056        0.4          0.135        0.08         0.0
       0.00         1.5          0.0          0.01         0.02                  F2
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

      0.056       0.4         0.135       0.085       0.0
      0.00        1.5         0.00        0.01        0.02                F2+CR
      0.056       0.4         0.135       0.13        0.0
      0.00        2.0         0.00        0.00        0.04                REFL.
      0.00        0.3         0.00        0.01        0.0
      0.00        2.0         0.00        0.0         0.04                REFL+CR
      0.00        0.3         0.00        0.055       0.0
  T
13*               1.0         0.0         T
15*         331       596         630     T
17*           F         1   T
18*           F         1   T
19*                    10          30     T
*JEND
```

A-3 Input cards set up for a restart case of the IAEA problem

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

¥NO         B333.                                                          /
                                                                      C.4/CORE 256
                                                                      W.1/PAGE  80
                                                                      P.0/PCH    0
                                                                      T.7/TIME 60M
                                                                      .LRG/
¥GJOB    S521130,T,ISE,431,11,FEMBABEL                                  /JOB-CARD
¥HFORT
C     PROGRAM  FEM-BABEL
      COMMON    IA(140000)
      CALL CHGFLO(M,N)
C                      * FACOM *
      IA(1)=140000
C
      CALL DTLIST
      CALL MAIN1
      STOP
      END
      SUBROUTINE ACCEL(P,S)
      DIMENSION P(1),S(1)
      COMMON    A(1)
      EQUIVALENCE (A(142),NPALL),(A(128),RAMDA),(A(253),SORF)
      T=0.0
      DO 1 I=1,NPALL
      P(I)=P(I)+1.50*(S(I)-P(I))
    1 T=T+P(I)
      DO 2 I=1,NPALL
    2 P(I)=P(I)/T
      RETURN
      END
¥HLIEDRUN  RFNAME=J1223,BABELA,  EDIT=YES,UPDT=YES,SIZE=30              BABELA
¥TPDISK F01,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F02,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F03,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F04,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F08,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F09,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F10,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F11,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TAPE  F12,J1223,B333,NEW,010215
¥TPDISK F13,DISP=DELETE,RSIZE=900,BSIZE=6300
¥TPDISK F14,DISP=DELETE,RSIZE=900,BSIZE=6300
¥DATA
 3D-IAEA(MODIFIED) FOR COMPARISON WITH CITATION, DX=5CM DZ=10CM  RESTART
 1¥                 2           1           630         39          595 1 -05
           5        4           5            35          0            0 6 -11
           6      100          36             1          1            4 12-17
           4 3R     1 6R        0             0          0            0 18-30
           1        2           0  T                                    31-33
 2*           1.0 -04     1.0 -03           1.6       3800.        55.0 1 - 5
          8.    3.265+10   T
¥JEND
```