AAEC/E307

AAEC/E307

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

ZHEX – A THREE DIMENSIONAL DIFFUSION CODE FOR HEXAGONAL, Z GEOMETRY

by

G. DOHERTY

March 1974

ISBN 0 642 99624 5

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

ZHEX - A THREE DIMENSIONAL DIFFUSION CODE FOR HEXAGONAL, Z GEOMETRY

by

G. DOHERTY

ABSTRACT

A description is given of ZHEX, a multigroup, finite difference diffusion code for hexagonal-z geometry. The code was written for an IBM360/50 run in MVT mode using HASP and employs dynamic storage allocation, direct access data sets and some double precision arithmetic. The basic calculation block contains one plane of mesh points for one group so that core storage requirements are independent of the numbers of groups and axial planes. With the present overlay structure a 250 MW fast reactor requires 246K and a 1,000 MW fast reactor 342K. Calculation times are given for some realistic sample problems.

National Library of Australia card number and ISBN 0 642 99624 5

The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA–INIS–12 (INIS: Manual for Indexing) and IAEA–INIS–13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.

COMPUTER CALCULATIONS; FAST REACTORS; FINITE DIFFERENCE METHOD; HEXAGONAL CONFIGURATION; IBM COMPUTERS; MULTIGROUP THEORY; NEUTRON DIFFUSION EQUATION; NEUTRON FLUX; NEUTRON LEAKAGE; NEUTRONS; PERFORMANCE; THREE-DIMENSIONAL CALCULATIONS; Z CODES

CONTENTS

1. INTRODUCTION

The program ZHEX was written to examine the feasibility of performing multigroup, three dimensional, finite difference diffusion calculations for fast reactors of current design. The AAEC computing environment consists of an IBM360/50 with 1152K bytes of processor storage, run in MVT mode (using HASP) with a maximum program partition of 600K bytes. Direct access storage comprises 12 AMPEX disc drives with capacity and transfer rates similar to an IBM2314 drive and an average access time of 32 milliseconds.

Subassemblies in a fast reactor core are hexagonal in cross section and the diffusion equations for a horizontal plane must in consequence be solved on either a hexagonal or a triangular mesh. Based on a rough figure of 2 MW per subassembly, the number of subassemblies in a 1000 MW core will be approximately 500. If each hexagon is divided into six equilateral triangles and the flux computed at the centroid of each triangle, the total number of mesh points per plane will be 3000 and the number per subassembly 6. Alternatively, if the flux is evaluated at the corners and centre of each hexagon, the total number per plane will be about 1700 and the number per subassembly 7 - an impressive saving for a comparable mesh representation. The disadvantage of the hexagon corner and centre mesh is that further subdivision is impossible should the basic mesh prove inadequate, while for the triangular mesh further subdivision presents no conceptual difficulties. To give the program some possibility of running on our present computer, the hexagon corner and centre mesh has been chosen. Variable axial mesh intervals are allowed in the program and the fluxes are evaluated at planes on the edges of mesh intervals (rather than at the centre of the interval). The program is designed to accommodate a variable number of energy groups, but upscattering is not permitted as the code is specifically oriented towards fast reactors.

The basic solution process for a group is a plane-by-plane relaxation with individual optimum relaxation parameters for each plane and for the axial sweep through all planes. Within the plane each iteration sweeps from the centre outwards, solving separately for the isolated hexagon centres and simultaneously for all points on the outer boundary of each ring of hexagons. The amount of computation involved in the latter is roughly twice that required when the normal forward elimination, back substitution algorithm is applied to a vector of the same length. A reduction in the number of inner iterations is provided by a coarse mesh rebalance which is applied once on entry to each group.

As the program is limited to downscattering group structures, an outer iteration consists of one pass through the groups. Fission source and eigenvalue convergence is accelerated by the use of a modified Chebyshev extrapolation procedure developed by Pollard (1973).

The axial boundary conditions imposed on the flux at the lower and upper boundaries are respectively

$$\frac{d\phi}{dz} = \frac{\phi}{\lambda d_1} \quad , \quad \frac{d\phi}{dz} = - \frac{\phi}{\lambda d_2}$$

where λ is the transport mean free path of the region adjacent to the boundary, and

d_1 and d_2 are input parameters.

Reflective boundaries are signalled by d = 0 in the input. The boundary condition in each horizontal plane is that the flux goes to zero on a circular boundary r = R. The plane is divided into concentric hexagonal rings of subassemblies containing in turn 1, 6, 12, 18 .... subassemblies (Figure 1). The program requires the outermost ring to be completely filled and also that a material be specified for the volume between the outermost ring and the extrapolation boundary r = R. The precise method of representing the boundary is discussed in detail in Section 2. It is felt that the method used is a reasonable compromise between realistic physical representation and convenience of implementation in the program.

2. FINITE DIFFERENCE EQUATIONS

As indicated in the introduction, the fluxes are computed at the corners and centres of hexagons on planes which lie on the boundaries of mesh intervals. Thus, if there are NZ axial mesh intervals the fluxes will be computed on NZ + 1 planes. Figures 2 and 3 show the layout of mesh points and planes followed in the program. Figure 4 shows the volume ABCDEF associated with a point of the mesh (for convenience the central point labelled 1 has been taken), and dimensions of importance.

The diffusion equation is

$$- \nabla \cdot D\nabla\phi + \Sigma_R\phi = S \qquad \ldots(1)$$

where D is the diffusion coefficient

Σ_R is the removal cross section

ϕ is flux density

S is source density

For the point 1 in plane i (coordinate z_i) we integrate over the hexagonal prism with cross section ABCDEF bounded by the planes $z = z_i - 0.5\ \Delta z_{i-1}$,

$z = z_i + 0.5\,\Delta z_{i+1}$ and apply Green's theorem in the usual manner to obtain

$$D_{12i}\,\frac{\phi_{1i} - \phi_{2i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{13i}\,\frac{\phi_{1i} - \phi_{3i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{14i}\,\frac{\phi_{1i} - \phi_{4i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{15i}\,\frac{\phi_{1i} - \phi_{5i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{16i}\,\frac{\phi_{1i} - \phi_{6i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{17i}\,\frac{\phi_{1i} - \phi_{7i}}{h}\cdot\frac{h}{\sqrt{3}}\cdot\frac{\Delta z_{i-1} + \Delta z_i}{2}$$

$$+\ D_{1+i}\,\frac{\phi_{1i} - \phi_{1i+1}}{\Delta z_1}\cdot\frac{\sqrt{3}\,h^2}{2}$$

$$+\ D_{1-i}\,\frac{\phi_{1i} - \phi_{1i-1}}{\Delta z_{i-1}}\cdot\frac{\sqrt{3}\,h^2}{2}$$

$$+\ \Sigma_{R1i}\,\phi_{1i}\,\frac{\sqrt{3}\,h^2}{2}\,\frac{\Delta z_i + \Delta z_{i-1}}{2}$$

$$= S\,\frac{\sqrt{3}\,h^2}{2}\,\frac{\Delta z_i + \Delta z_{i-1}}{2} \qquad \ldots(2)$$

The first six terms represent the leakage out of faces AB, BC, CD, DE, EF and FA respectively. Each is based on a linear approximation for the flux between mesh points. On the face AB the gradient $\nabla\phi \approx \frac{\phi_2 - \phi_1}{h}\,\underset{\sim}{n}$ where $\underset{\sim}{n}$ is the normal to the face. The diffusion coefficient D_{12i} is the surface weighted average diffusion coefficient for the different materials bordering the line $1 \rightarrow 2$:

$$D_{12i} = \frac{\Delta z_{i-1}\,D_{Ai-1} + \Delta z_i\,D_{Ai} + \Delta z_{i-1}\,D_{Bi-1} + \Delta z_i\,D_{Bi}}{2(\Delta z_{i-1} + \Delta z_i)} \qquad \ldots(3)$$

where D_{Ai-1} = D for material in mesh volume below A

D_{Ai} = D for material in mesh volume above A, etc.

Σ_R is a volume weighted average over constituent materials.

The seventh and eighth terms are the leakage out of the top and bottom faces respectively:

$$D_{1+i} = \frac{D_{Ai} + D_{Bi} + D_{Ci} + D_{Di} + D_{Ei} + D_{Fi}}{6} \qquad \ldots(4)$$

The ninth term is the loss of neutrons (removals) by absorption and downscatter from the volume. The right hand side is the production of neutrons within the volume. Thus Equation (2) may be viewed as a statement of neutron conservation in the volume element:

leakage + removals = production .

Similar equations hold for all internal mesh points. Mesh points on the periphery have different equations which incorporate the boundary conditions we impose on the flux. Thus for the bottom plane (i=1)

$$D_{121}\ \frac{\phi_{11} - \phi_{21}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_1}{2}$$

$$+\ D_{131}\ \frac{\phi_{11} - \phi_{31}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_1}{2}$$

$$+\ D_{141}\ \frac{\phi_{11} - \phi_{41}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_1}{2}$$

$$+\ - - - - - -$$

$$+\ D_{171}\ \frac{\phi_{11} - \phi_{17}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_1}{2}$$

$$+\ D_{1+1}\ \frac{\phi_{11} - \phi_{12}}{\Delta z_1} \cdot \frac{\sqrt{3}\, h^2}{2}$$

$$+\ D_{1+1}\ \frac{\phi_{11}}{d_1 \lambda} \cdot \frac{\sqrt{3}\, h^2}{2}$$

$$+\ \Sigma_{11}\, \phi_{11}\, \frac{\sqrt{3}\, h^2}{2} \cdot \frac{\Delta z_1}{2} = S\, \frac{\sqrt{3}\, h^2}{2} \cdot \frac{\Delta z_1}{2} \qquad \ldots(5)$$

The equation for the top plane is similar to that for the bottom plane except that the boundary condition is included in the upleakage rather than the down.

A typical radial peripheral point is illustrated in Figure 5. The volume around the mesh point is filled with material which is specified to surround the last ring of hexagons. The effect of the material is confined to its contribution to the peripheral volume elements and to providing a D for the outwards leakage. Thus a large reflector would be imperfectly represented and should be included by specifying a further ring or rings of hexagons to improve the representation. Equation (6) for point 44 of Figures 2 and 5 is given as an example.

$$
\begin{aligned}
& D_{44,73,i} \frac{\phi_{44,i} - \phi_{73,i}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_m \frac{\phi_{44,i}}{p_1} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_m \frac{\phi_{44,i}}{p_2} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_m \frac{\phi_{44,i}}{p_3} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_{44,45,i} \frac{\phi_{44,i} - \phi_{45,i}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_{44,32,i} \frac{\phi_{44,i} - \phi_{32,i}}{h} \cdot \frac{h}{\sqrt{3}} \cdot \frac{\Delta z_{i-1} + \Delta z_i}{2} \\
+ \; & D_{44,+,i} \frac{\phi_{44,i} - \phi_{44,i+1}}{\Delta z_i} \frac{\sqrt{3}\, h^2}{2} \\
+ \; & D_{44,-,i} \frac{\phi_{44,i} - \phi_{44,i-1}}{\Delta z_i} \cdot \frac{\sqrt{3}\, h^2}{2} \\
+ \; & \Sigma_{R44,i} \phi_{44,i} \frac{\sqrt{3}\, h^2}{2} \frac{\Delta z_i + \Delta z_{i-1}}{2} \\
= \; & S \frac{\sqrt{3}\, h^2}{2} \frac{\Delta z_i + \Delta z_{i-1}}{2}
\end{aligned}
\qquad \ldots (6)
$$

3. SOLUTION OF THE EQUATIONS

For a general point j in plane i we may write the finite difference equation for a particular group in the form

$$\sum_k \alpha_{jki}\,\phi_{ki} + \beta_{ji}\,\phi_{j\,i+1} + \gamma_{ji}\,\phi_{j\,i-1} = S_{ji} \qquad \ldots(7)$$

S_{ji} is the source density of neutrons in the volume element surrounding the point (j,i). It is composed of a fission source (if non-zero) and (excluding group 1) downscattering from higher groups. When solving the equations for a particular group, the set of S_{ji} are regarded as fixed. The sum over k in Equation (7) includes k=j and the six neighbours of j.

To show the plane-by-plane iteration process more clearly Equation (7) is written in the form

$$\sum_k \alpha_{jki}\,\phi_{ki} = S_{ji} - \beta_{ji}\,\phi_{j\,i+1} - \gamma_{ji}\,\phi_{j\,i-1} \qquad \ldots(8)$$

On iteration m+1 a Gauss-Seidel iteration would be written

$$\sum_k \alpha_{jki}\,\phi^{m+1}_{ki} = S_{ji} - \beta_{ji}\,\phi^{m}_{ji+1} - \gamma_{ji}\,\phi^{m+1}_{j\,i-1} \qquad \ldots(9)$$

where the superscript indicates the iteration to which the flux belongs.

For the purpose of exposition we assume that Equation (9) is solved explicitly (though in fact an iterative procedure is also employed for it). With this assumption, the plane-by-plane iteration matrix is irreducible, block tridiagonal, and block diagonally dominant. Therefore over-relaxation may be employed, using well proven methods to determine the optimum over-relaxation parameter w (Appendix A).

Instead of obtaining ϕ^{m+1}_{ji} directly from Equation (9) the following procedure is used:

$$\sum_k \alpha_{jki}\,\psi_{ki} = S_{ji} - \beta_{ji}\,\phi^{m}_{j\,i+1} - \gamma_{ji}\,\phi^{m+1}_{j\,i-1} \qquad \ldots(10)$$

$$\phi^{m+1}_{ji} = w\,\psi_{ji} + (1-w)\,\phi^{m}_{ji} \qquad \ldots(11)$$

Using this strategy, the largest amount of data needed to hold in core at any one time is that required to solve Equations (10) and (11) for one plane and one group. Very large problems can, therefore, be accommodated.

When solving Equation (10) iteratively, the right hand side may be regarded as constant. Collecting terms and dropping unnecessary subscripts and superscripts, Equation (10) may be written

$$\sum_k \alpha_{jk} \psi_k = T_j \qquad \ldots(12)$$

or in matrix notation as

$$A \psi = T \qquad \ldots(13)$$

To simplify the presentation ψ is partitioned in the following way:

partition 1	internal points of ring 1
partition 2	external points of ring 1
partition 3	internal points of ring 2
partition 4	external points of ring 2

Thus for the example in Figure 2

$$\psi^T = (\psi_1 \,|\, \psi_2 \to \psi_7 \,|\, \psi_8 \to \psi_{13} \,|\, \psi_{14} \to \psi_{31} \,|\, \psi_{32} \to \psi_{43} \,|\, \psi_{44} \to \psi_{73}) \ldots(14)$$

Now, because the sum over k in Equation (12) includes only j and its six nearest neighbours, with this partitioning A becomes a well-organised sparse matrix which is irreducible and diagonally dominant. In the example (Figure 2):

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} & 0 & 0 \\ 0 & A_{32} & A_{33} & A_{34} & 0 & 0 \\ 0 & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} \\ 0 & 0 & 0 & A_{54} & A_{55} & A_{56} \\ 0 & 0 & 0 & A_{64} & A_{65} & A_{66} \end{pmatrix} \qquad \ldots(15)$$

The diagonal submatrices A_{ii} are, of course, square. Those relating to internal points (viz A_{11}, A_{33}, A_{55}) are diagonal because the centres of hexagons are not immediate neighbours. Those relating to external points (A_{22}, A_{44}, A_{66}) are almost tridiagonal, having the two extra non-zero entries characteristic of a periodic system.

Thus, in the example

$$A_{22} = \begin{pmatrix} \alpha_{22} & \alpha_{23} & 0 & 0 & 0 & \alpha_{27} \\ \alpha_{32} & \alpha_{33} & \alpha_{34} & 0 & 0 & 0 \\ 0 & \alpha_{43} & \alpha_{44} & \alpha_{45} & 0 & 0 \\ 0 & 0 & \alpha_{54} & \alpha_{55} & \alpha_{56} & 0 \\ 0 & 0 & 0 & \alpha_{65} & \alpha_{66} & \alpha_{67} \\ \alpha_{72} & 0 & 0 & 0 & \alpha_{76} & \alpha_{77} \end{pmatrix} \quad \ldots(16)$$

An efficient algorithm based on elimination can be used to solve systems of equations involving this type of matrix (see Appendix C). The amount of computing required is roughly double that of the forward elimination, back substitution algorithm for tridiagonal matrices. Numerical stability is good, even for the rather long strings of mesh points towards the outer boundary of a large power reactor. The block matrix A of Equation (15) gives rise to a 2-cyclic iteration matrix allowing the normal formulae of Appendix A to be applied in the determination of an optimum over-relaxation factor. The submatrices $A_{i,i-2}$, $A_{i,i+2}$ (i even) are extremely sparse - for example A_{64} contains only 12 non-zero entries in a total of 720. A, therefore, behaves very much like a block tridiagonal matrix.

Solution of Equation (13) by over-relaxation using the matrix partition of Equation (15) appears reasonably efficient, though we are not in a position to compare it with alternative techniques. It corresponds to the radial sweep from the centre outwards which is commonly employed in r,z diffusion codes.

The steps of an inner iteration are in summary:

- Read source S, plane 1, store in T
- Read flux plane 2, compute leakage to plane 1, add to T
- Solve for new plane 1 flux
- Set T for plane i = leakage from plane i-1
- Read S plane i, add to T
- Read flux plane i+1, compute leakage to plane i, add to T
- Solve for new plane i flux

REPEAT i = 2,NZ

- Set T for plane NZ+1 = leakage from plane NZ
- Read S plane NZ+1, add to T
- Solve for new flux, plane NZ+1
- Test for convergence.

4. COARSE MESH REBALANCING

Spatial rebalancing is applied before inner iterations for a group are commenced. The user must specify a set of coarse subdivisions of the reactor

to take advantage of the accelerated convergency which the method offers. In a fast reactor containing a central core, axial and radial blankets and reflector, the physical boundaries between the different material compositions provide a suitable coarse mesh over which to rebalance. In large commercial reactors further subdivisions of the various regions may be desirable.

The regions are specified to be adjacent rings of channels grouped together and adjacent axial planes grouped together. For example, in the sample problem rings 1 and 2 may be lumped together, ring 3 kept separate, and the axial dimension cut into 3 to give a total of 6 coarse regions.

As will be seen, the mathematics of the procedure is rather general and not restricted to the shape demanded here (a hexagonal doughnut) or even to contiguous spatial regions. There are intuitive reasons for believing that with the inner iteration scheme 60 degree slices might give better convergence. It is also possible to rebalance during inner iterations instead of only on the first iteration. The choice has been governed by the ease with which the scheme adopted can be fitted into the rather complicated overall solution strategy.

The starting point for the coarse mesh rebalance procedure is Equation (7) which is restated for continuity .

$$\sum_k \alpha_{jki}\,\phi_{ki} + \beta_{ji}\,\phi_{j,i+1} + \gamma_{ji}\,\phi_{j,i-1} = S_{ji} \qquad \ldots(17)$$

At the end of a set of inner iterations the flux ϕ^o_{ji} will be a good approximation to the true solution of Equation (17). When next this group is encountered the source will have changed to S^1_{ji} giving rise to a new solution ϕ^1_{ji}. A collection of disjoint sets R_n of mesh points is defined such that the union of the sets contains all the mesh points in the problem. (These sets are, of course, the coarse mesh rebalance regions.) It is assumed that ϕ^1_{ji} which is the solution of

$$\sum_k \alpha_{jki}\,\phi^1_{ki} + \beta_{ji}\,\phi^1_{j,i+1} + \gamma_{j,i}\,\phi^1_{j,i-1} = S^1_{ji} \qquad \ldots(18)$$

takes the form

$$\phi^1_{ji} = \varepsilon_n\,\phi^0_{ji} \quad \text{for } (j,i) \text{ in } R_n \qquad \ldots(19)$$

Equation (19) is substituted into Equation (18) and equations for points (j,i) which comprise R_n are collected together to obtain a set of equations of the form

$$\sum_n L_{mn}\ \varepsilon_n = S^1_m \qquad \ldots(20)$$

where $$S^1_m = \sum_{(j,i)\ \text{in}\ R_m} S^1_{ji} \qquad \ldots(21)$$

and $$L_{mn} = \sum_{\substack{(j,i)\ \text{in}\ R_m \\ (k,i)\ \text{in}\ R_n}} \alpha_{jki}\ \phi^0_{ki} + \sum_{\substack{(j,i)\ \text{in}\ R_m \\ (j,i+1)\ \text{in}\ R_n}} \beta_{ji}\ \phi^0_{j,i+1} + \sum_{\substack{(j,i)\ \text{in}\ R_m \\ (j,i-1)\ \text{in}\ R_n}} \gamma_{ji}\ \phi^0_{j,i-1} \qquad \ldots(22)$$

The dimension of the matrix equation, though at the discretion of the user, will probably be small. In the program the matrix is row normalised to avoid precision difficulties and the equations are then solved using the Gauss-Jordan elimination routine SID (J.P. Pollard, unpublished AAEC report). The vector ϕ^1_{ji} computed from Equation (19) is then used as the starting point for the next round of iterations.

The form of the matrix L_{mn} is, of course, determined by the disjoint sets R_n which are used to partition the core volume. Using the example of 2 radial and 3 axial cuts, L takes the form

$$L = \begin{pmatrix} L_{11} & L_{12} & L_{13} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & L_{24} & 0 & 0 \\ L_{31} & 0 & L_{33} & L_{34} & L_{35} & 0 \\ 0 & L_{42} & L_{43} & L_{44} & 0 & L_{46} \\ 0 & 0 & L_{53} & 0 & L_{55} & L_{56} \\ 0 & 0 & 0 & L_{64} & L_{65} & L_{66} \end{pmatrix} \qquad \ldots(23)$$

where ε_1 = factor for radial cut 1, axial cut 1

ε_2 = factor for radial cut 2, axial cut 1

ε_3 = factor for radial cut 1, axial cut 2 ... etc.

The rebalance procedure can be formulated in terms of a variational principle which effectively disguises its essential simplicity. Its efficiency is difficult to establish theoretically, particularly if it is applied every few inner iterations, but there are examples in the literature of the substantial savings in computation which result from its use. In this program

it is used once per group, before inner iterations commence, to fit in with the overall strategy involving disc storage of fluxes and coefficients. Problems of convergence with modified inner iteration schemes therefore do not arise.

5. OUTER ITERATION STRATEGY

The outer iteration scheme we have employed follows that of POW which is described in detail by Pollard (1973) and here we merely summarise the computational steps. If A is the matrix which operates on a fission source density vector ψ to produce the next generation vector ϕ, the principal eigenvalue and eigenvector of

$$A\psi = \lambda \psi \qquad \ldots(24)$$

are sought.

Suppose $\lambda_1 > |\lambda_2| \geqslant$ rest of the eigenvalues, and that ψ_1, ψ_2 are the associated eigenvectors, then the unextrapolated procedure may be written

$$\phi^{n+1} = A\,\psi^n \qquad \ldots(25)$$

$$h^{n+1} = (V,\ \phi^{n+1}) \qquad \ldots(26)$$

$$\psi^{n+1} = \frac{1}{h^{n+1}}\,\phi^{n+1} \qquad \ldots(27)$$

The vector V in the inner product of Equation (26) is a volume vector so that the renormalisation of Equation (27) produces one neutron per iteration. The equation sequence (25) to (27) gives the simple power method which would eventually yield λ_1 and ψ_1. The dominance ratio d defined by

$$d = \lambda_2/\lambda_1 \qquad \ldots(28)$$

is estimated during this phase.

Suppose the starting guess ψ^0 has the form

$$\psi^0 = \sum_{1}^{m} b_m\ \psi_m \qquad \ldots(29)$$

where $$A\psi_m = \lambda_m\ \psi_m \qquad \ldots(30)$$

The dominance ratio d is estimated by assuming that, after a number of iterations have been performed, ψ^n can be adequately represented by a two term expansion

$$\psi^n \approx \frac{\lambda_1^n b_1 \psi_1}{h^1 h^2 \text{ --- } h^n} + \frac{\lambda_2^n b_2 \psi_2}{h^1 h^2 \text{ --- } h^n} \qquad \ldots(31)$$

If ψ_i^n is element number i of the vector ψ^n, the ratio of elements on successive iterations r_i^n can be written

$$r_i^n = \frac{\psi_i^n}{\psi_i^{n-1}} = \frac{\lambda_1 [1 + d^n B_i]}{h^n [1 + d^{n-1} B_i]} \qquad \ldots(32)$$

where $$B_i = b_2 \psi_{2i} / b_1 \psi_{1i} \qquad \ldots(33)$$

After some elementary algebra

$$d = \frac{h^{n-1} h^n}{\lambda_1^2} r_i^{n-1} r_j^{n-1} \frac{r_i^n - r_j^n}{r_i^{n-1} - r_j^{n-1}} \qquad \ldots(34)$$

In addition to the approximation embodied in Equation (31), the further assumption

$$\frac{h^{n-1} h^n}{\lambda_1^2} = 1 \qquad \ldots(35)$$

is made to reduce Equation (34) to a form suitable for the estimation of d

$$d = r_i^{n-1} r_j^{n-1} \frac{r_i^n - r_j^n}{r_i^{n-1} - r_j^{n-1}} \qquad \ldots(36)$$

Power iterations are continued until a value of d (less than unity) is obtained to 10 per cent on successive iterations. If d is computed to be less than 0.1 rapid convergence is already assured and normal power iterations are continued. For d in the range (0.1, 1.0) the program begins tentative Chebyshev extrapolation, at the same time attempting to further refine its estimate of d.

The modification to the equation sequence (25) to (27) is as follows :

$$\phi^n = A \psi^{n-1} \qquad \ldots(37)$$

$$h^n = (r, \phi^n) \qquad \ldots(38)$$

$$\psi^n = \alpha_n \beta \phi^n + \alpha_n(1 - \beta h^n)\psi^{n-1} - (\alpha_n - 1)\psi^{n-2} \quad \ldots(39)$$

where $$\beta = \frac{2}{2\lambda_1 - \lambda_2} \quad \ldots(40)$$

$$\alpha_0 = 1 \quad \ldots(41)$$

$$\alpha_1 = 0.5/[0.5 - \gamma \alpha_0] \quad \ldots(42)$$

$$\alpha_n = 1/[1 - \gamma \alpha_{n-1}] \quad \ldots(43)$$

and $$\gamma = \frac{\lambda_2^2}{4(2\lambda_1 - \lambda_2)^2} \quad \ldots(44)$$

The usual Chebyshev extrapolation procedure is similar. When λ_1 is the dominant eigenvalue and the remainder of the eigenvalues of A are contained in the interval $(0,\lambda_2)$ the equation sequence (37) to (44) is repeated except for

$$\psi^n = \frac{\alpha_n \beta \lambda_1}{h^n} \phi^n + \alpha_n(1 - \beta \lambda_1)\psi^{n-1} - (\alpha_n - 1)\psi^{n-2} \quad \ldots(45)$$

A discussion of the use of modified Chebyshev extrapolation is given by Pollard (1973). The main implication of its use is that the set of polynomials Q_n for which ψ^n may be written

$$\psi^n = \sum_{m=1}^{m} Q_n(\lambda_m)\, b_m\, \psi_m \quad \ldots(46)$$

satisfy the recurrence relation

$$Q_n(\lambda) = \alpha_n(\beta \lambda + 1 - \beta h^n)\, Q_{n-1}(\lambda) - (\alpha_n - 1)\, Q_{n-2}(\lambda) \quad \ldots(47)$$

rather than

$$Q_n(\lambda) = \alpha_n(\beta \lambda + 1 - \beta \lambda_1)\, Q_{n-1}(\lambda) - (\alpha_n - 1)\, Q_{n-2}(\lambda) \quad \ldots(48)$$

for the conventional Chebyshev polynomials.

Applying analysis similar to that outlined for the power iterations we arrive at a formula for estimating the dominance ratio:

$$\frac{r_i^n - r_j^n}{r_i^{n-1} - r_j^{n-1}} = \frac{Q_{n-1}(h^n)\, Q_{n-1}(h^n d)}{Q_{n-1}(h^n)\, Q_{n-2}(h^n d)} \qquad \ldots(49)$$

where $r_i^n = \phi_i^n / h^n \psi_i^{n-1}$ $\qquad \ldots(50)$

Equation (49) is solved by the chord method for d. Low order (8) extrapolation is continued until an estimate of d has been obtained to 0.5 per cent after which high order (16) extrapolation is permitted and further attempts to refine the estimate of d are abandoned.

6. CODE PERFORMANCE

To give a realistic assessment of the utility of the code, test calculations were performed on a reactor similar in size and composition to an early PFR design (Frame *et al* 1966). The reactor consists of 10 rings of channels (271 in all), each hexagon being 14.2 cm across the flats. Control rod locations were assumed to be occupied by fuel for simplicity. Symmetry about the axial midplane was assumed - in situations where this assumption cannot be used the computation times will be double those quoted.

Calculations were made using a 6-group subset of the Hansen Roach 16-group cross section set. The generation of the 6-group macroscopic cross section library was performed using the AUS system (J.P. Pollard *et al*, AAEC unpublished report). The present version of ZHEX reads cross sections only from AUS cross section data sets, but the coding required to generalise this step is inconsequential.

The first calculation was made using 3-axial mesh intervals (4 planes) each of roughly 22 cm. This calculation provided a guide to what might be expected from a more realistic calculation. The numbers of interest which emerged are summarised in Table 1.

TABLE 1

SUMMARY OF 3-AXIAL MESH, 6-GROUP, SIMPLIFIED PFR CALCULATION

Dominance ratio	0.82
Typical eigenvalue for plane iteration	0.65
Typical eigenvalue for between-plane iteration	0.65
Typical outer iteration IBM360/50 computer time (min)	3
Number of outers to converge to 10^{-4} in source	19

The second calculation was made using 9-axial mesh intervals (10 planes) giving an axial separation between mesh points of about 7.5 cm which is comparable with the radial separation. Results are summarised in Table 2.

TABLE 2

SUMMARY OF 9-AXIAL MESH, 6-GROUP, SIMPLIFIED PFR CALCULATION

Dominance ratio	0.83
Typical eigenvalue for plane iteration	0.28
Typical eigenvalue for between-plane iteration	0.82
Typical outer iteration IBM360/50 computer time (min)	8
Number of outers to converge to 10^{-4} in source	19

A number of points of interest emerge from a comparison of Tables 1 and 2.

- The dominance ratio in the two calculations is similar. The second eigenvalue corresponds to a radial mode and is essentially determined by the physical size of the reactor so the close correspondence is not surprising.
- The eigenvalues of the iteration matrix for a single plane decrease as the axial mesh is refined, which is due to the increasing contribution of the leakage to neighbouring planes in the overall loss term.
- The eigenvalues of the between-plane iteration matrix increase as the mesh is refined because the leakage to neighbouring planes occurs in the off-diagonal elements of the matrix.
- The ratio of times is roughly proportional to the ratio of the number of mesh planes, suggesting that the overall eigenvalue of the group iteration matrix is not strongly dependent on the axial mesh spacing in the region of interest. Obviously, an indefinite axial mesh refinement would be accompanied by a dramatic increase in eigenvalue.

Starting from a flux guess flat in core and breeder regions and discontinuous at the interfaces, the 6-group, 9 axial mesh problem requires 150 minutes to converge to 10^{-4} in source vector. Substantial savings can result from starting with a better flux guess, as one might do in a burnup sequence, or converging the source vector less tightly, which may well be adequate for many applications.

For a change in group structure the computation time should exhibit a

roughly linear dependence on the number of groups. The additional scattering computation will be compensated by a reduction in the eigenvalues of the group iteration matrix as removals increase.

A further calculation was performed with three axial mesh intervals for an idealised power reactor of 1000 MW containing 16 rings of channels. The number of mesh points per plane increases from 871 to 2247 and, because the central core is somewhat larger, the dominance ratio could be expected to be appreciably worse. Results for this calculation are summarised in Table 3.

TABLE 3

SUMMARY OF 3-AXIAL MESH, 6-GROUP, SIMPLIFIED 1,000 MW REACTOR CALCULATION

Dominance ratio	0.81
Typical eigenvalue for plane iteration	0.66
Typical eigenvalue for between-plane iteration	0.64
Typical outer iteration IBM360/50 computer time (min)	8
Number of outers to converge to 10^{-4} in source	19

The dominance ratio is not appreciably worse because the problem attempted has $\pi/3$ symmetry in both solution and flux guess. The method of solution retains this symmetry and thereby exclues the possibility of a higher dominance ratio. Even with the present dominance ratio, a 9-axial mesh, 6-group calculation for this reactor would require about 6 hours to converge to 10^{-4} in source.

7. CONCLUSIONS

The times presented in the previous section provide a rough framework in which the possible role of a direct finite difference three dimensional flux solution can be assessed. The speed is adequate to establish benchmark solutions against which simpler reactor representations and approximate solution techniques can be evaluated. A limitation on the use of the program in this mode is the indivisibility of the hexagonal mesh adopted for the plane. Results quoted by Adamson, Mann & Perks (1969) for the two dimensional code TRIAG (which uses a triangular mesh capable of further subdivision) show that errors of up to 2 per cent in channel ratings are possible in regions of rapidly changing flux such as core/breeder interfaces. The larger errors of up to 14 per cent in control rod worths would be more serious if left unattended, but can be eliminated by a modification of control rod cross sections, taking account of the reactor mesh representation, as is usually done in thermal reactor design.

The direct use of ZHEX in routine calculations would be expensive. A

change in the source convergence criterion from 10^{-4} to 10^{-3} would save roughly 25 per cent of the times quoted. Better flux guesses which would be available in a burnup sequence, or for a perturbation calculation involving control rod movement, could give a comparable saving. A 6-group ZHEX calculation on a 1000 MW reactor with 18 axial planes, starting from a reasonable flux guess and converging to 10^{-3} in source would require about 6 hours on an IBM360/50 or 12 minutes on an IBM370/165. This compares with quoted times of 2 minutes on an IBM370/165 for a similar 2-group AGR calculation using the tailored design code CRISP described by Askew, Anderson & Pearson (1972). With refinements, ZHEX could be used for detailed design check calculations or following the history of an operating reactor. The slow running time clearly limits its application in the wider field of exploring various operating and refuelling alternatives.

Because of the labour involved, a comparison of the inner iteration scheme used in ZHEX with other possible choices has not been performed. It was felt that the scheme adopted was reasonably efficient and that the computing times would be representative of direct three dimensional solution methods. Comparison with experiment is considered to be beyond the scope of this report.

8. REFERENCES

Adamson, J., Mann, J.E. & Perks, M.A. (1969) - Considerations of detail in physics design and fuel management of fast power breeder reactors. BNES Conference on the Physics of Fast Reactor Operation and Design, London.

Askew, J.R., Anderson, D.W. & Pearson, K.G. (1972) - Methods for three dimensional fuel management studies on high temperature reactors. AEEW-R832.

Frame, A.G., Hutchinson, W.G., Laithwaite, J.M. & Parker, H.F. (1966) - Design of the prototype fast reactor. BNES Conference on Fast Breeder Reactors, London.

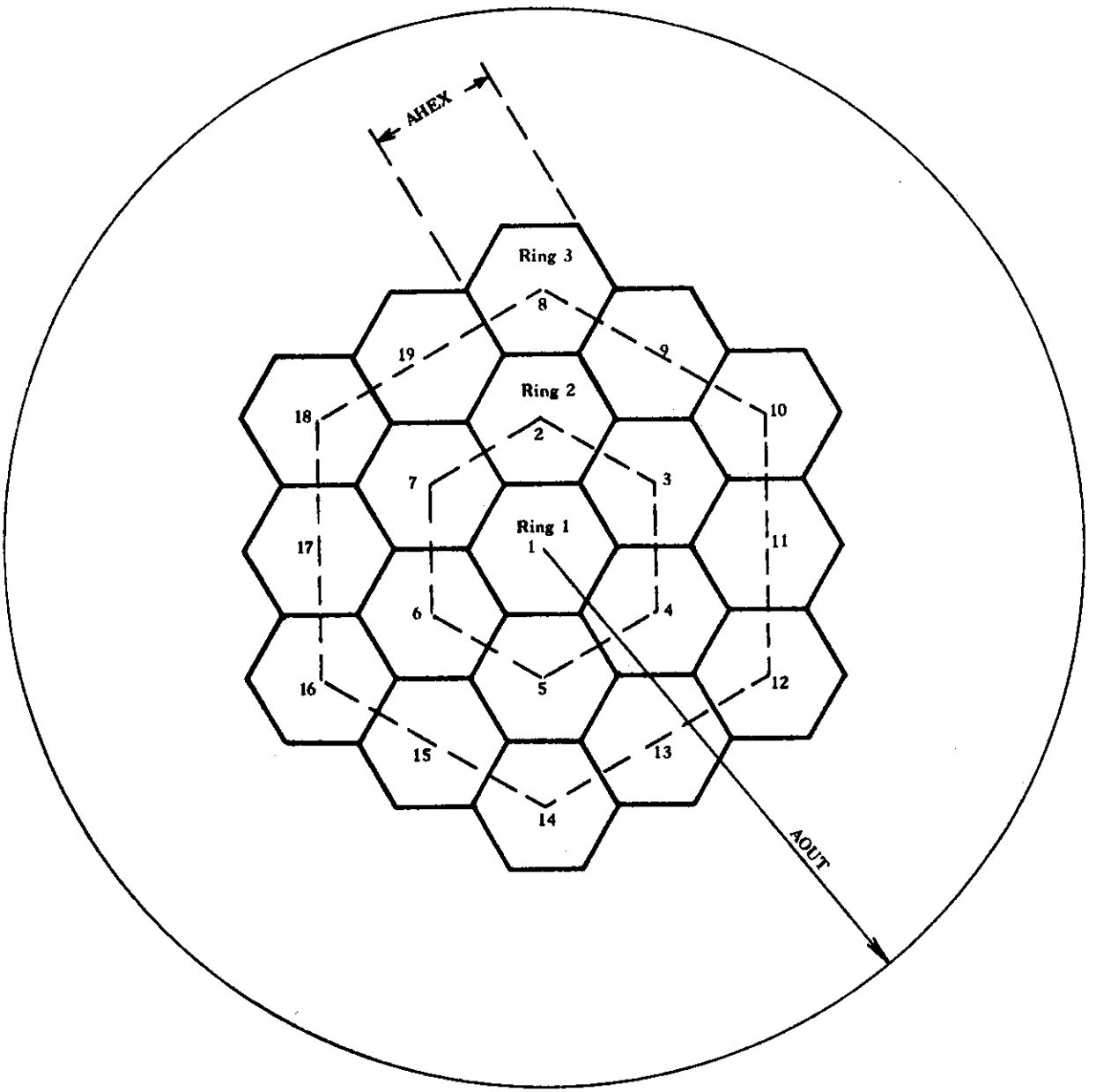
Hopkins, D. & Oakes, D.B. (1968) - The two dimensional, multigroup diffusion code, GOG. AEEW-R532.

Pollard, J.P. (1973) - Numerical methods used in neutronics calculations. Ph.D Thesis, University of NSW.

Varga, R.S. (1962) - Matrix Iterative Analysis. Prentice Hall, New Jersey.

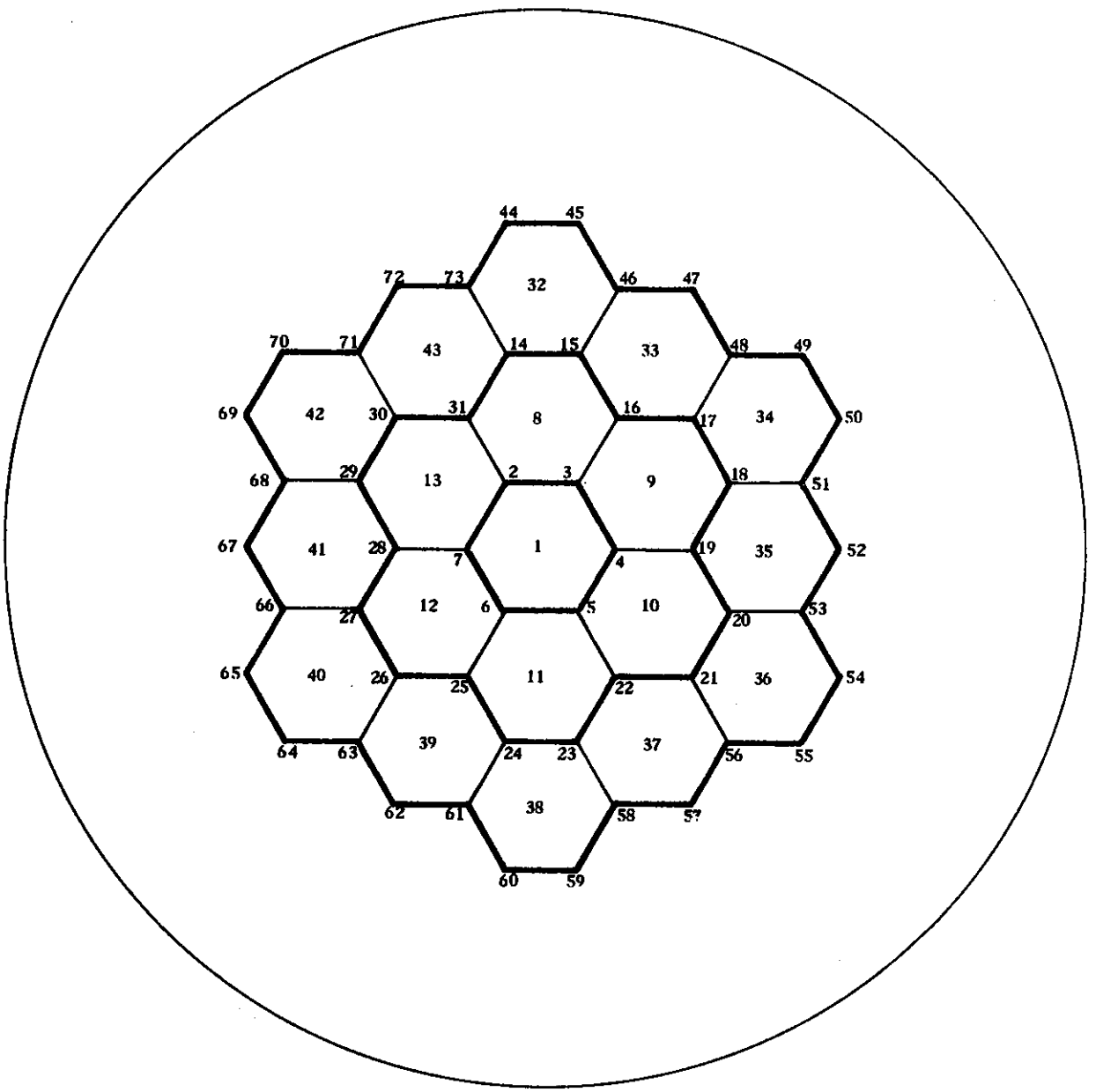
9. ACKNOWLEDGEMENT

The author wishes to thank Mr. J. Pollard for his advice and assistance in the development of this program. The optimum relaxation parameter search, outer iteration acceleration and rebalance procedure of the two dimensional code POW (Pollard 1973) form the basis of those features in ZHEX.



FLUX IS ASSUMED TO GO TO ZERO ON r = AOUT

FIGURE 1. CHANNEL LAYOUT IN A PLANE (NRING = 3)



POINTS CONNECTED BY HEAVY LINES ARE COMPUTED SIMULTANEOUSLY IN THE FLUX SOLUTION PROCESS

FIGURE 2. MESH POINTS IN A PLANE (NRING = 3)

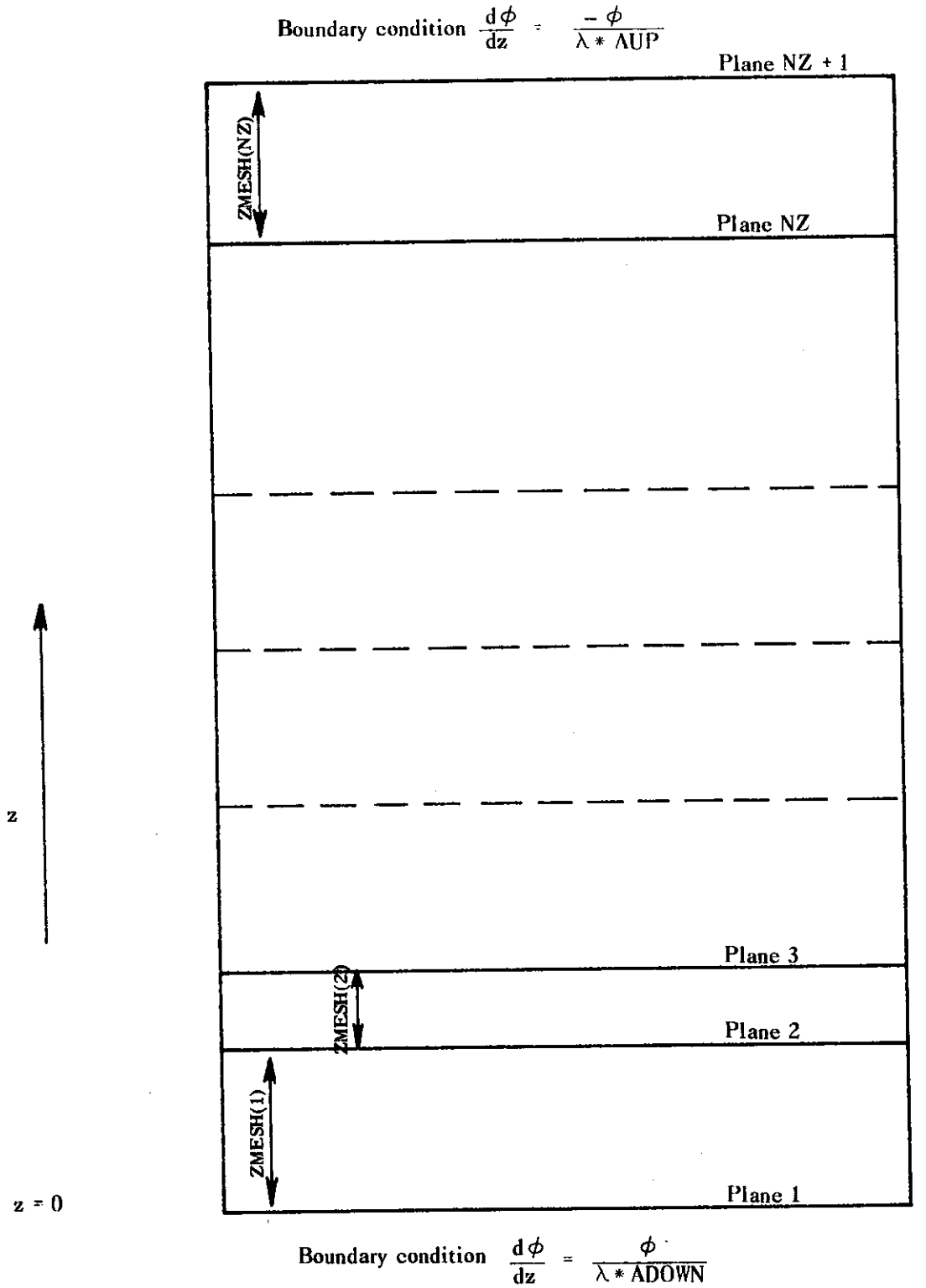


FIGURE 3. PLANE LAYOUT

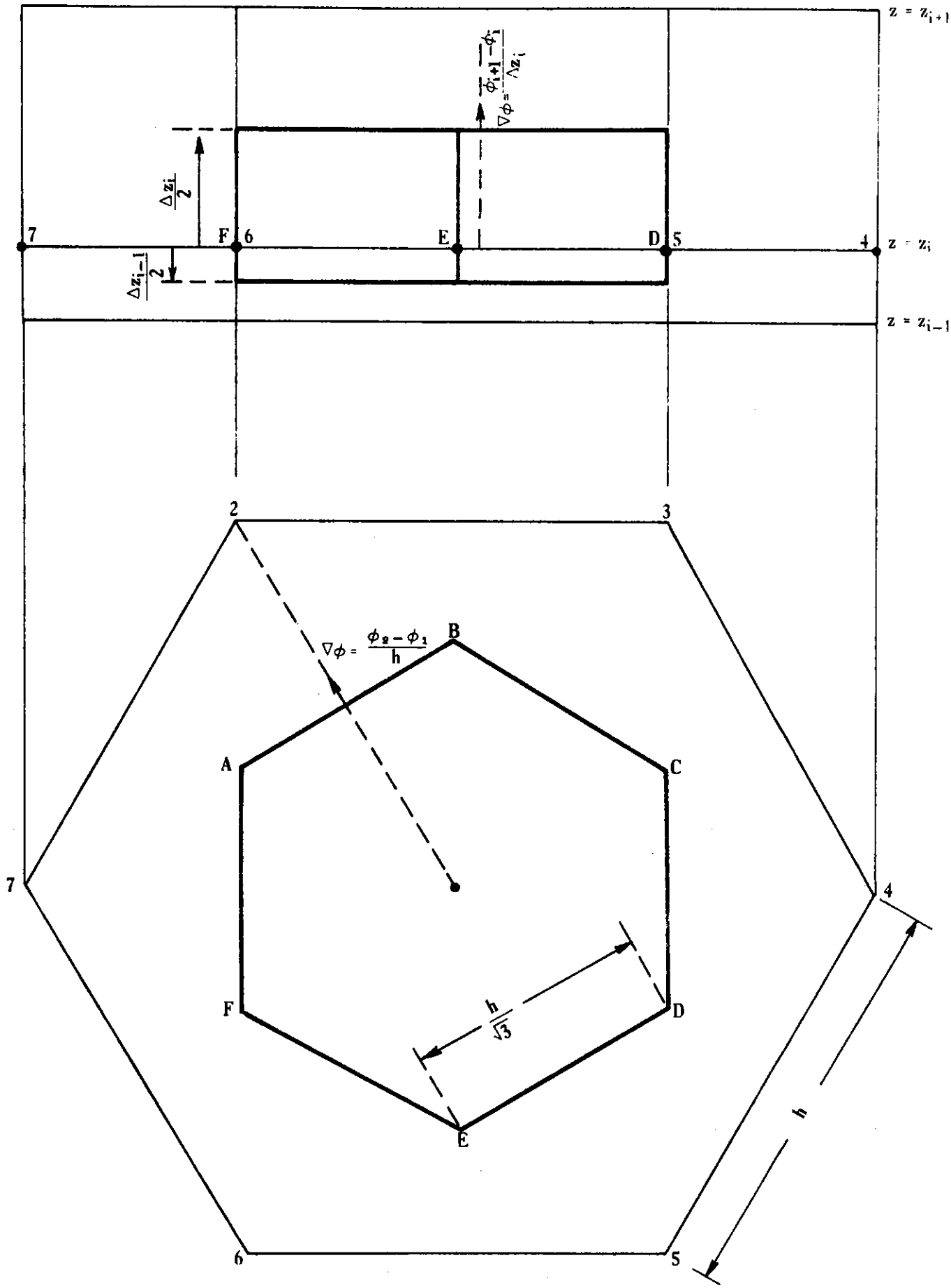


FIGURE 4. VOLUME ELEMENT ABOUT A MESH POINT

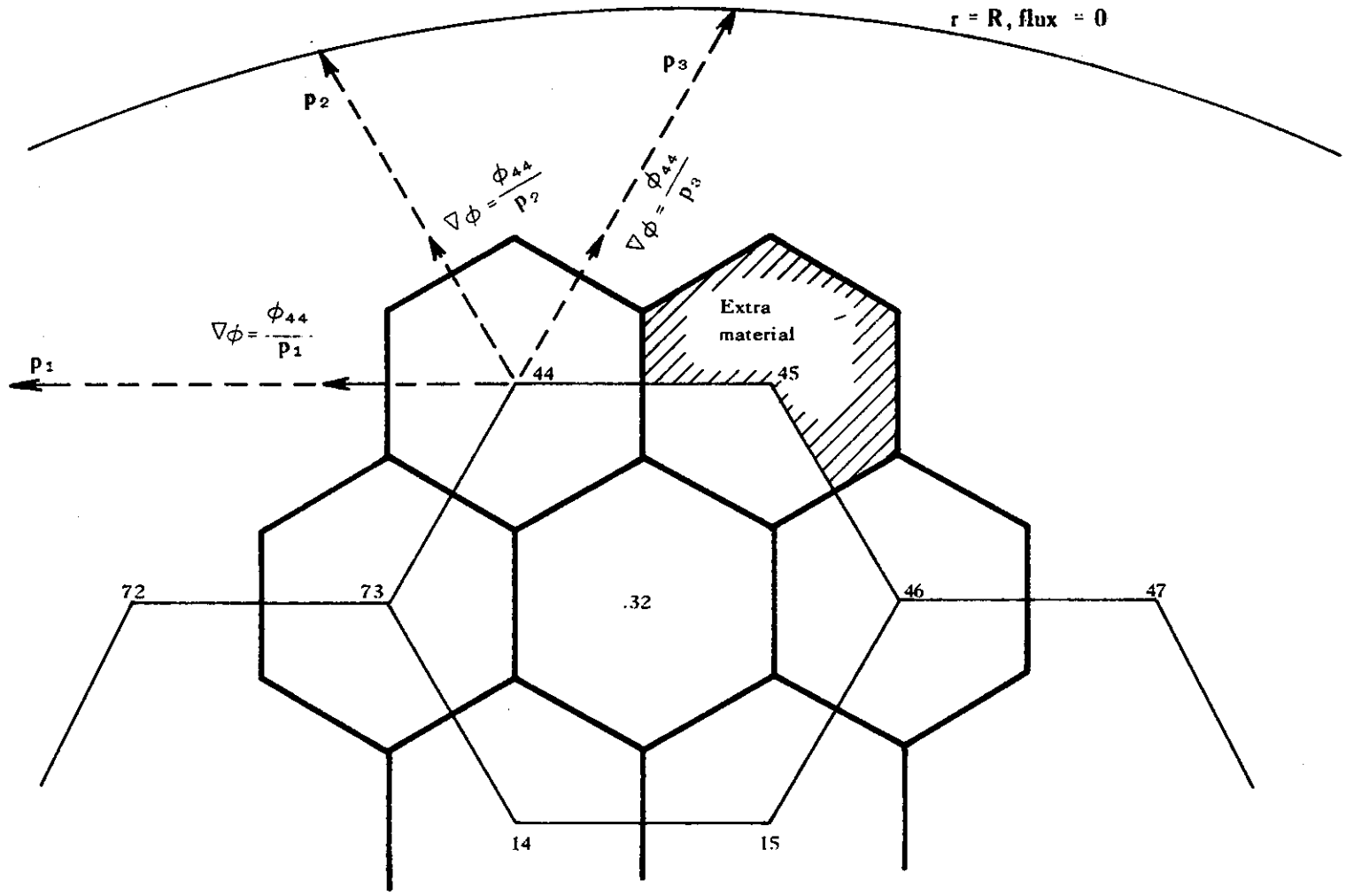


FIGURE 5. OUTER BOUNDARY VOLUME ELEMENT

APPENDIX A

ESTIMATION OF OPTIMUM OVER-RELAXATION PARAMETERS

A matrix iterative scheme may be written in the form

$$\phi_{n+1} = A\,\phi_n + b \qquad \ldots(A1)$$

where the subscript denotes the iteration number.

If the true solution is ϕ then

$$\phi = A\,\phi + b \qquad \ldots(A2)$$

Subtracting these two equations

$$\phi_{n+1} - \phi = A(\phi_n - \phi) \qquad \ldots(A3)$$

The rate of convergence of the scheme is therefore determined by the eigenvalues of A. The optimum relaxation parameter for an over-relaxation scheme also depends on the largest eigenvalue of A (Varga 1962). In this program the elegant method of Pollard (1973) for estimating the largest eigenvalue is followed.

Similarly to Equation (A3), we can easily derive

$$\phi_{n+1} - \phi_{n+2} = A(\phi_n - \phi_{n+1}) \qquad \ldots(A4)$$

After some iterations it is assumed that $\phi_n - \phi_{n+1}$ can be approximated by

$$\phi_n - \phi_{n+1} \approx z_1 + z_2 \qquad \ldots(A5)$$

where $$A\,z_1 = \lambda_1\,z_1 \qquad \ldots(A6)$$

$$A\,z_2 = \lambda_2\,z_2 \qquad \ldots(A7)$$

and $\lambda_1 \geqslant \lambda_2 \geqslant$ remaining eigenvalues of A. $\qquad \ldots(A8)$

Write

$$C_1 = \phi_n - \phi_{n+1} = z_1 + z_2 \qquad \ldots(A9)$$

$$C_2 = A(\phi_n - \phi_{n+1}) = \lambda_1 z_1 + \lambda_2 z_2 \qquad \ldots(A10)$$

$$C_3 = A^2(\phi_n - \phi_{n+1}) = \lambda_1^2 z_1 + \lambda_2^2 z_2 \qquad \ldots(A11)$$

$$C_4 = A^3(\phi_n - \phi_{n+1}) = \lambda_1^3 z_1 + \lambda_2^3 z_2 \qquad \ldots(A12)$$

If it is supposed that λ_1, λ_2 are the roots of a quadratic equation

$$x^2 - px + q = 0 \qquad \ldots(A13)$$

then

$$C_3 - C_2 p + C_1 q = 0 \qquad \ldots(A14)$$

$$C_4 - C_3 p + C_2 q = 0 \qquad \ldots(A15)$$

In fact Equations (A14) and (A15) ought to be true element by element for the length of the vectors C_1, C_2, etc. In the program scalars C_1, C_2, C_3, C_4 are used which are the sums of the elements of the vectors defined in Equations (A9) to (A12) to obtain unique λ estimates from each iteration.

$$p = \frac{C_4 C_1 - C_2 C_3}{C_3 C_1 - C_2^2} \qquad \ldots(A16)$$

$$q = \frac{p C_3 - C_4}{C_2} \qquad \ldots(A17)$$

$$\lambda_1 = \frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q} \qquad \ldots(A18)$$

$$\lambda_2 = \frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q} \qquad \ldots(A19)$$

Estimates of λ_1 can be obtained from Equation (A18) on successive iterations and the differences in the estimates largely reflect the quality of the approximation in Equation (A5). The advantage of Pollard's procedure is that the information necessary to determine an optimum over-relaxation parameter can be accumulated during the course of solution with an un-optimised w. This represents a significant saving over a procedure discussed by Varga (1962), and implemented in the program GOG (Hopkins & Oakes 1968) for example, whereby the eigenvalue information is obtained by solving the artificial problem with the vector b of Equation (A1) set to zero.

If μ is an eigenvalue of the Jacobi iteration matrix which is cyclic of index p, and λ is an eigenvalue of the over-relaxation matrix with arbitrary w, then the relation (Varga 1962)

$$(\lambda + w - 1)^p = \lambda^{b-1} w^b \mu^p \qquad \ldots(A20)$$

In particular for $p = 2$,

$$(\lambda + w - 1)^2 = \lambda\, w^2\, \mu^2 \qquad \ldots(A21)$$

and the optimum over-relaxation parameter w_b is

$$w_b = \frac{2}{1 + \sqrt{1 - \mu^2}} \qquad \ldots(A22)$$

Reliable estimates of w_b are obtained in the program during the course of solution with $w = 1$. Once w_b is sufficiently well determined iterations proceed with $w = w_b$.

APPENDIX B

ELIMINATION METHOD FOR PERIODIC SYSTEMS

A solution to the matrix equation

$$Ax = b \qquad \ldots(B1)$$

is sought, where A is tridiagonal except for the elements a_{1n}, a_{n1}.

Thus for a 4 x 4 problem A has the form

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ a_{41} & 0 & a_{43} & a_{44} \end{pmatrix} \qquad \ldots(B2)$$

The pre-inversion stage involves a series of operations to obtain A^1 in the form

$$A^1 = \begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} \\ 0 & a_{22}^1 & a_{23}^1 & a_{24}^1 \\ 0 & 0 & a_{33}^1 & a_{34}^1 \\ 0 & 0 & 0 & a_{44}^1 \end{pmatrix} \qquad \ldots(B3)$$

These tranformations take the form

$$p_i = a_{i+1,\,i}/a_{i,i}$$

$$q_i = a_{n,i}/a_{i,i}$$

$$a_{i+1,j} = a_{i+1,j} - p_i\, a_{i,j} \qquad \text{(all j)}$$

$$a_{n,j} = a_{n,j} - q_i\, a_{i,j} \qquad \text{(all j)}$$

The vectors p_i, q_i i = 1,n-1 and the matrix A^1 are stored.

For each new vector b we compute b^1 using the same transformations, i.e.

$$b_{i+1} = b_{i+1} - p_i\, b_i$$

$$b_n = b_n - q_i\, b_i$$

and then solve the upper triangular system

$$A^1x = b^1$$

If the diagonals of A^1 are stored as inverses the arithmetic per point in the solution process consists of 5 multiplications and 4 subtractions.

The matrix A in the problem is diagonally dominant and double precision arithmetic has been used in this part of the program so precision difficulties are not expected. Both in prior external testing and in the test problems run so far, the algorithm has performed satisfactorily.

APPENDIX C

INPUT DATA

Data: (read with SCAN except for the title card which must be distinct)

NRING	number of complete rings of hexagons making up a horizontal plane (if ≤ 0 job restarts from tape on FT20 and requires no further card input)
NZ	number of axial mesh points
NRR	number of ring subdivisions for rebalance
NZZ	number of axial subdivisions for rebalance
NMATT	number of material specification cards required to specify the material layout (refer to material layout cards NC1 etc.)
ACC	convergence tolerance for eigenvalue and fission source vector
JGEOM	>0 implies geometry specification follows on subsequent cards (a copy will be made on unit 13) ≤ 0 implies the geometry is to be read from an AUS library for geometry on unit 13
TITLE CARD	(read and written 20A4)
AHEX	distance across the flats of basic hexagon
(ZMESH(I),I=1,NZ)	dimensions of each axial mesh interval from bottom plane upwards
AOUT	an extrapolated outer boundary on which the flux is assumed to be zero for the calculation of leakage in the horizontal plane
ADOWN, AUP	extrapolation distance for the lower and upper planes respectively. If d is the relevant distance, the boundary condition is actually applied in the form $\frac{d\phi}{dx} = -\frac{\phi}{\lambda d}$ where λ is the transport mean free path of the material inside the boundary (d = 0.71 is frequently used; d = 0.0 is interpreted as a reflective boundary).
NC1,NC2,NZ1,NZ2,MAT	A material layout card implying that the volumes identified by channels NC1 to NC2 inclusive, and mesh intervals NZ1 to NZ2 inclusive are occupied by material

number MAT. Layout directives are executed sequentially in the order presented and multiple specification for volumes are permitted (to simplify this sequence of the input). Channels are labelled from the centre outwards, each ring containing the following number of channels 1, 6, 12, 18, 24 --- respectively. NC1=0 is taken to refer to the material surrounding the outermost ring of complete hexagons. The final layout card contains NC1=-1 while NC2 etc. are absent.

Table 4 contains a summary of channel and mesh point information.

(MRR(I),I=1,NRR) ring subdivisions for rebalance

(MZZ(I),I=1,NZZ) axial subdivisions for rebalance

Rebalance procedure involves a renormalisation of all the fluxes in a region so that for the region

absorption + leakage = source

The total number of coarse regions in the rebalance is NRR* NZZ. The code expects

MRR(NRR) = NRING

and MZZ(NZZ) = NZ+1

RATIO, NR1, NR2, NZ1, NZ2

The flux guess routine expects a ratio core flux/reflector flux and two ring numbers NR1, NR2 and axial mesh numbers NZ1, NZ2 to define the core. If RATIO≤0. the code will attempt to read a flux from unit 19 which is assumed contains a dump from a previous ZHEX run on a similar system. In such a problem NR1, NR2, NZ1, NZ2 are not punched.

TABLE C1

MESH POINT SUMMARY

Ring	Channels	Central points	Outside corner points
1	1 → 1	1 → 1	2 → 7
2	2 → 7	8 → 13	14 → 31
3	8 → 19	32 → 43	44 → 73
4	20 → 37	74 → 91	92 → 133
5	38 → 61	134 → 157	158 → 211
6	62 → 91	212 → 241	242 → 307
7	92 → 127	308 → 343	344 → 421
8	128 → 169	422 → 463	464 → 553
9	170 → 217	554 → 601	602 → 703
10	218 → 271	704 → 757	758 → 871
11	272 → 331	872 → 931	932 → 1057
12	332 → 397	1058 → 1123	1124 → 1261
13	398 → 469	1262 → 1333	1334 → 1483
14	470 → 547	1484 → 1561	1562 → 1723
15	548 → 631	1724 → 1807	1808 → 1981
16	632 → 721	1982 → 2061	2062 → 2247