PNC TN9410 96-277

An Investigation of Discretization Errors for Mesh Centered Finite Difference Approximations

October 1996

POWER REACTOR AND NUCLEAR FUEL DEVELOPMENT CORPORATION
O-ARAI ENGINEERING CENTER

複製又はこの資料の入手については，下記にお問い合わせください。
〒311-13　茨城県東茨城郡大洗町成田町4002
動力炉・核燃料開発事業団
大洗工学センター　システム開発推進部・技術管理室

Enquires about copyright and reproduction should be addressed to: Technology Management Section O-arai Engineering Center, Power Reactor and Nuclear Fuel Development Corporation 4002 Narita-cho, O-arai-machi, Higashi-Ibaraki, Ibaraki-ken, 311-13, Japan

動力炉・核燃料開発事業団（Power Reactor and Nuclear Fuel Development Corporation）

PNC TN9410 96-277
October, 1996

ABSTRACT

J K Fletcher*

In this paper eigenvalue mesh dependence is investigated for mesh centered finite difference approximations to the diffusion equation. The well known mesh squared dependence is quantified giving the following formula, in rectangular geometry, for δk the change in eigenvalue k

$$\frac{\delta k}{k} = \frac{1}{6} \frac{-\sum_{ijk} V_{ijk}\phi^{*}_{ijk}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\alpha_i^2\frac{\partial^2}{\partial x^2}+\beta_j^2\frac{\partial^2}{\partial y^2}+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk} + \sum_{\substack{mn,\, l\,\text{boundary}\\ perm(l,m,n)}} \phi^{*}_{lmn} D_{l\pm\frac{1}{2}mn}\, w_l^2\left(\frac{\partial^2}{\partial w^2_{(m)}}+\frac{\partial^2}{\partial w^2_{(n)}}\right)\phi_{lmn}}{\sum_{ijk} V_{ijk}\phi^{*}_{ijk}\frac{(\nu\sigma_f)_{ijk}}{k}\phi_{ijk}}$$

with ϕ_{ijk} and ϕ^{*}_{ijk} the regular and adjoint fluxes at mesh region ijk of volume V_{ijk}. $\nu\sigma_f, \sigma_s$ and σ_r are the production, scatter and removal cross sections respectively. Meshes in the x,y and z directions are denoted by α_i, β_j and γ_k. $w_{(m)}$ implies a mesh parallel to an outer boundary and *perm* means permute the indices so that all boundaries are included. $D_{l\pm\frac{1}{2}mn}$ is the diffusion coefficient at the boundary and the $\pm$ indicates whether the boundary is in the positive or negative axis direction from the meshpoint. Similar expressions are then found for triangular and hexagonal geometries

The second part of the paper describes a method of significantly reducing mesh errors in diffusion theory finite difference codes. Essentially approximations to higher derivatives involving flux values at mesh points are used to generate a source which eliminates second order errors. The approach has been implemented in XYZ and Triangular-Z geometries and after a description of the technique results are presented for a series of test problems showing that almost zero mesh values can be obtained with the correction process.

* Core Physics Research Section, Advanced Technology Division
PNC O-arai Japan.

PNC TN9410 96-277
1 9 9 6 年 1 0 月

有限差分近似法における
メッシュ誤差低減の研究

J. K. フレッチャー*

要 旨

中性子拡散方程式において、中心メッシュ有限差分近似を用いた場合の、固有値のメッシュ依存性を研究した成果を報告する。直交座標系における有限メッシュの、固有値に対するエラー δk は下式として導出されたが、これは従来経験的に用いてきたメッシュサイズの2乗則を定量的に表現している。

$$\frac{\delta k}{k}=\frac{1}{6}\frac{-\sum_{ijk}V_{ijk}\phi^{*}_{ijk}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\alpha_i^2\frac{\partial^2}{\partial x^2}+\beta_j^2\frac{\partial^2}{\partial y^2}+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk}+\sum_{\substack{mn,\,l\,boundary\\ perm(l,m,n)}}\phi^{*}_{lmn}D_{l\pm\frac{1}{2}mn}w_l^2\left(\frac{\partial^2}{\partial w^2_{(m)}}+\frac{\partial^2}{\partial w^2_{(n)}}\right)\phi_{lmn}}{\sum_{ijk}V_{ijk}\phi^{*}_{ijk}\frac{(\nu\sigma_f)_{ijk}}{k}\phi_{ijk}}$$

ここで、ϕ_{ijk} と ϕ^{*}_{ijk} は各々ボリューム V_{ijk} をもつ領域 ijk での中性子束と随伴中性子束である。$\nu\sigma_f,\sigma_s$ 及び σ_r は各々、中性子生成、散乱及び除去断面積であり、x,y 及び z におけるメッシュサイズは α_i,β_j 及び γ_k で表している。$w_{(m)}$ は外側境界におけるメッシュの平行成分方向を表す。*perm*は、全ての境界が含まれるように添字を交換する意味である。

$D_{l\pm\frac{1}{2}mn}$ は 境界での拡散係数であり、±はメッシュ点からの正負方向を表す。同様の表式は、三角メッシュと六角メッシュ形状についても導出できた。

本報告の後半では、この定式化により、拡散理論でのメッシュサイズ誤差が非常に低減する方法を示す。ここではメッシュ点での中性子束を含む高次微分近似が、2次エラーを低減したソース項の生成のために用いられた。このアプローチはXYZと三角-Z形状に適用され、テストコードによる検証の結果、ほぼゼロメッシュサイズと同等の値が得られることが判明した。

---

* 動燃 大洗工学センター 基盤技術開発部 炉心技術開発室 国際特別研究員

CONTENTS

Page

ABSTRACT I

CONTENTS III

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 DERIVATION OF TERMS IN THE DIFFUSION EQUATION .. 2

CHAPTER 3 EVALUATING TAYLOR'S SERIES COEFFICIENTS 5

CHAPTER 4 TREATMENT OF BOUNDARIES 6

CHAPTER 5 DERIVATION OF THE SECOND ORDER MESH CORRECTION 6

CHAPTER 6 SIMPLIFICATION OF THE MESH CORRECTION FORMULA 8

CHAPTER 7 OTHER GEOMETRIES 10

7.1 TRIANGULAR-Z GEOMETRY 10

7.2 HEXAGONAL-Z GEOMETRY 14

CHAPTER 8 DERIVATION OF THE FINITE DIFFERENCE EQUATIONS FOR SOLUTION 15

CHAPTER 9 THE SECOND ORDER MESH CORRECTION IN DIFFERENCE FORM 17

CHAPTER 10 TRIANGULAR-Z GEOMETRY 19

CHAPTER 11 STANDARD FINITE DIFFERENCE SOLUTION 21

CHAPTER 12 TEST PROBLEMS AND RESULTS 28

12.1 Small FBR Problem 28
12.2 Large FBR Problem 30
12.3 Hexagonal FBR Problem 30

CHAPTER 13 DISCUSSION 31

CHAPTER 14 CONCLUSIONS 33

REFERENCES 34

FIGURE 1 ILLUSTRATION OF FINITE DIFFERENCE MESH IN XYZ GEOMETRY 35

FIGURE 2 TRIANGULAR-Z COORDINATES IN THE XY PLANE 36

FIGURE 3 LOCAL COORDINATES FOR TRIANGULAR-Z 37

FIGURE 4 HEXAGONAL-Z COORDINATES 38

FIGURE 5 LOCATION OF POINTS TO BE USED IN EVALUATING $\frac{\partial^2}{\partial s^2}$. 39

FIGURE 6 TEST PROBLEM 1 SMALL FBR 40

FIGURE 7 TEST PROBLEM 2 LARGE FBR 41

FIGURE 8 TEST PROBLEM 3 HEXAGONAL FBR 42

FIGURE 9 CONTROL ROD POSITIONS FOR PROBLEM 3 43

APPENDIX 1 DATA INPUT TO THE PNCD CODE 44

APPENDIX 2 EXAMPLE OF INPUT TO PNCD 46

APPENDIX 3 CROSS SECTION DATA FOR THE XYZ GEOMETRY TEST 48

APPENDIX 4 CROSS SECTION DATA FOR THE TRIANGULAR-Z CASE 50

APPENDIX 5 FORTRAN LISTING OF PNCD CODE 53

CHAPTER 1 INTRODUCTION

For many years it has been known that eigenvalues calculated by finite difference approximations show a significant dependence on mesh size and that the variation is linear with mesh interval squared[1,2] as long as the grid is not too coarse. In this work the approximations made when deriving the standard finite difference equations will be studied with a view to finding expressions for the effect on eigenvalue and then means sought to reduce them in mesh centered finite difference solutions of the diffusion equation.

The method used is based on the Taylor expansion of a continuous and differentiable function. However as is well known[3] flux differentials of higher order than the first are singular across material interfaces and this phenomenon contributes significantly to the mesh dependence of the eigenvalue as will be seen in the following sections. Also such singular behavior implies that care must be exercised in deciding whether derivatives across interfaces can be neglected and when using Taylor's expansion.

As indicated in the Title for this work mesh centered finite difference approximations only will be investigated mainly because the higher order terms take a fairly straightforward form, as will be seen later.

In Chapter 2 the equations for XYZ geometry are derived followed by a brief discussion of error term evaluation in Chapter 3. Modifications for boundaries follow in Chapter 4 and the second order eigenvalue correction is derived in Chapter 5. A method of simplifying the eigenvalue correction formula is then presented and equivalent formulæ for Triangular-Z and Hexagonal-Z geometries are derived in Chapter 7.

These first chapters consist of theoretical work and although there is some satisfaction in quantifying the second order mesh effect on eigenvalue, practical applications are not obvious. The remainder of the work looks at a means of modifying the standard finite difference solution of the diffusion equation so that mesh effects are included up to second order. Basically the next higher derivatives in the standard mesh centered finite difference method[3] have been retained in the

numerical approximation, evaluated using flux values from the previous iteration and added in as an extra source term in the numerical solution technique utilized.

In Chapter 8 the equations for solution in XYZ geometry are derived, with the numerical representation of mesh correction terms forming Chapter 9. The equivalent work for Triangular-Z geometry follows. A diffusion theory code was written to carry out the test work and details of the line overrelaxation solution used may be seen in Chapter 11 leading on to results for various test problems in Chapter 12. Finally DISCUSSION and CONCLUSIONS sections summarize what has been achieved by the work.

CHAPTER 2 DERIVATION OF TERMS IN THE DIFFUSION EQUATION

Figure 1 shows the normal mesh centered approximation for XYZ geometry. In this a regular grid is imposed on the problem and planes defined by the grid coincide with material boundaries. Points are then located at cell or mesh box centers and, using these, approximations made to the diffusion equation so leading to a set of difference relations usually solved by iterative methods(3).

Derivations depend on the Taylor expansion which may be written as

$$\phi(x+\delta x, y+\delta y, z+\delta z) = \phi(x,y,z) + \sum_{n=1}^{\infty} \sum_{\substack{i,j,k \\ i+j+k=n}} \frac{\delta x^i \delta y^j \delta z^k}{i!j!k!} \frac{\partial^n \phi(x,y,z)}{\partial x^i \partial y^j \partial z^k} \qquad (1),$$

this expression enables the series to be expressed compactly as

$$\phi(x+\delta x, y+\delta y, z+\delta z) = e^{\delta x \frac{\partial}{\partial x} + \delta y \frac{\partial}{\partial y} + \delta z \frac{\partial}{\partial z}} \phi(x,y,z) \qquad (2)$$

which will be used later.

The multigroup diffusion equation in the form

$$\nabla D \nabla \phi + \left(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r\right)\phi = 0 \qquad (3),$$

will be studied where ϕ is the flux column vector, D the diffusion coefficient and $\nu\sigma_f, \sigma_s$ and σ_r the production, scatter and removal(absorption plus scatter out of the group) cross section matrices respectively with k the eigenvalue. Dependence on the space variables x,y,z is understood. Clearly for clarity the one group case may be considered when ϕ and other terms in equation (3) are

scalars and probably it is useful to consider this form for the derivations which follow then convert back to the matrix expression in the final formulæ. Further conditions are that the flux and current, $-D\nabla\phi$, are continuous everywhere. Also at external boundaries the condition $\mathbf{n}.D\nabla\phi = -\lambda\phi$ applies where λ is a supplied constant. and $\mathbf{n}$ the outward normal.

The diffusion equation is then integrated over each mesh region yielding typically

$$\sum_{l=1,6}\int_{S_l} D\nabla\phi.\mathbf{dS} + \int_{v_{ijk}} (\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)\phi dV = 0 \qquad (4)$$

where l denotes the mesh box boundaries in some order.

Using Taylor's expansion the volume term for mesh region i,j,k takes the form

$$\int_{v_{ijk}} (\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk}(\phi_{ijk} + (x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y} + z\frac{\partial}{\partial z})\phi_{ijk}$$
$$+\frac{1}{2}(x^2\frac{\partial^2}{\partial x^2} + 2xy\frac{\partial^2}{\partial x\partial y} + 2xz\frac{\partial^2}{\partial x\partial z} + y^2\frac{\partial^2}{\partial y^2} + 2yz\frac{\partial^2}{\partial y\partial z} + z^2\frac{\partial^2}{\partial z^2})\phi_{ijk} + \ldots)dV \qquad (5)$$

where terms above second order have been neglected and the subscript on the brackets round the cross sections signifies value in the mesh box.

On performing the integration, because the point is at the center of the mesh box odd order contributions vanish giving

$$(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk} V_{ijk}(\phi_{ijk} + \frac{1}{24}(\alpha_i^2\frac{\partial^2}{\partial x^2} + \beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2})\phi_{ijk} + \ldots) \qquad (6)$$

when using the mesh intervals $\alpha_i, \beta_j, \gamma_k$ as specified in Figure 1 with V_{ijk} the cell volume.

For the surface leakage term just the i to $i+1$ contribution will be derived as the other five directions follow the same procedure. From Figure 1 the integral to be evaluated is

$$\int_{S_{ij}} D\frac{\partial\phi^-(y,z)}{\partial x}dydz \qquad (7).$$

with the minus superscript on the flux denoting the i,j,k side of cell face, this distinction being necessary as interfaces where material changes cause discontinuities in higher order derivatives because of current($-D\nabla\phi$) continuity. Again using Taylor's expansion

$$\phi_{ijk} = \phi^-(y,z) - \frac{\alpha_i}{2}\frac{\partial\phi^-(y,z)}{\partial x} - (y\frac{\partial}{\partial y} + z\frac{\partial}{\partial z})\phi^-(y,z)$$
$$+(\frac{\alpha_i^2}{8}\frac{\partial^2}{\partial x^2} + \frac{\alpha_i}{2}y\frac{\partial^2}{\partial x\partial y} + \frac{\alpha_i}{2}z\frac{\partial^2}{\partial x\partial z} + \frac{y^2}{2}\frac{\partial^2}{\partial y^2} + yz\frac{\partial^2}{\partial y\partial z} + \frac{z^2}{2}\frac{\partial^2}{\partial z^2})\phi^-(y,z) + \ldots \quad (8)$$

and in terms of $\phi^-_{i+\frac{1}{2}jk}$ the value at the center of the face, after further use of Taylor's theorem there results

$$\phi_{ijk} = \phi^-(y,z) - \frac{\alpha_i}{2}\frac{\partial\phi^-(y,z)}{\partial x} - (y\frac{\partial}{\partial y} + z\frac{\partial}{\partial z})\phi^-_{i+\frac{1}{2}jk}$$
$$+(\frac{\alpha_i^2}{8}\frac{\partial^2}{\partial x^2} + \frac{\alpha_i}{2}y\frac{\partial^2}{\partial x\partial y} + \frac{\alpha_i}{2}z\frac{\partial^2}{\partial x\partial z} - \frac{y^2}{2}\frac{\partial^2}{\partial y^2} - yz\frac{\partial^2}{\partial y\partial z} - \frac{z^2}{2}\frac{\partial^2}{\partial z^2})\phi^-_{i+\frac{1}{2}jk} + \ldots \quad (9)$$

yielding after integration over the cell face of area $S_{kj} = \beta_j\gamma_k$

$$S_{kj}\phi_{ijk} = \int_{S_{kj}}(\phi^-(y,z) - \frac{\alpha_i}{2}\frac{\partial}{\partial x}\phi^-(y,z))dS + \frac{S_{kj}}{8}(\alpha_i^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} + \ldots \quad (10).$$

Similarly for cell $i+1,j,k$

$$S_{kj}\phi_{i+1jk} = \int_{S_{kj}}(\phi^+(y,z) + \frac{\alpha_{i+1}}{2}\frac{\partial}{\partial x}\phi^+(y,z))dS + \frac{S_{kj}}{8}(\alpha_{i+1}^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^+_{i+\frac{1}{2}jk} + \ldots (11) .$$

At the boundary the flux and current are continuous so that $\phi^-(y,z) = \phi^+(y,z)$ and $D_{ijk}\frac{\partial\phi^-(y,z)}{\partial x} = D_{i+1jk}\frac{\partial\phi^+(y,z)}{\partial x}$ and by subtracting the above two equations then eliminating $\frac{\partial\phi^+(y,z)}{\partial x}$ the relation below is found

$$D_{ijk}\int_{S_{jk}}\frac{\partial\phi^-(y,z)}{dx}dS = \frac{2D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}\alpha_{i+1} + D_{i+1jk}\alpha_i)}((\phi_{i+1jk} - \phi_{ijk}) + \frac{1}{8}(\alpha_i^2\frac{\partial^2\phi^-_{i+\frac{1}{2}jk}}{\partial x^2} - \alpha_{i+1}^2\frac{\partial^2\phi^+_{i+\frac{1}{2}jk}}{\partial x^2}) + \ldots) \quad (12)$$

on remembering that derivatives parallel to the interface are continuous. In the standard formulation(3) the first terms in expression (6) and the right hand side of equation (12) are used to approximate to the diffusion equation integrated over the mesh box and for all i,j,k form the matrix equation to evaluate ϕ_{ijk}, the other higher order contributions being neglected.

CHAPTER 3 EVALUATING TAYLOR'S SERIES COEFFICIENTS

The exponential expression (2) provides a convenient means of evaluating higher order errors. The procedure adopted in the foregoing Chapter may be summarized, for the left hand side of the i to $i+1$ face and using the compact formula (2) as

$$\phi_{ijk} = \phi^{-}(y,z) - \delta x \phi^{-}(y,z) + (e^{-\delta x} + e^{\delta y + \delta z}(\delta x - 1))\phi^{-}_{i+\frac{1}{2}jk} \tag{13}$$

since the all terms except the first two were expressed as values at the midpoint of the cell surface and δx has been used to signify $x\frac{\partial}{\partial x}$ with a similar interpretation for the other variables.
Thus the nth order term will be

$$(-)^n \frac{\delta x^n}{n!} + \frac{\delta x(\delta y + \delta z)^{n-1}}{(n-1)!} - \frac{(\delta y + \delta z)^n}{n!} \tag{14}$$

For example the third order at the inner right hand cell boundary, after integration, takes the interesting form

$$\frac{S_{kj}\alpha_i}{48}\left(-\alpha_i^2 \frac{\partial^2}{\partial x^2} + \beta_j^2 \frac{\partial^2}{\partial y^2} + \gamma_k^2 \frac{\partial^2}{\partial z^2}\right)\frac{\partial \phi^{-}_{i+\frac{1}{2}jk}}{\partial x} \tag{15}$$

which is a simple result derived straightforwardly, given the complexity of the expansions utilized. Using this approach higher orders can be accurately and conveniently found.

Similarly an expression using mesh center values comes from

$$\phi_{ijk} = \phi^{-}(y,z) - \delta x \phi^{-}(y,z) + (1 + e^{\delta x + \delta y + \delta z}(\delta x - 1))\phi_{ijk} \tag{16}$$

giving an n'th order term,

$$\frac{\delta x(\delta x + \delta y + \delta z)^{n-1}}{(n-1)!} - \frac{(\delta x + \delta y + \delta z)^n}{n!} \tag{17}$$

which again can be easily evaluated. The difficulty with these expressions is their representation in difference form, given that discontinuities are found in higher derivatives.

CHAPTER 4 TREATMENT OF BOUNDARIES

At an outer boundary there is no adjoining mesh region to enable formula (12) to be derived. However substituting the usual diffusion theory boundary condition

$$\mathbf{n}.D\nabla\phi = -\lambda\phi \tag{18}$$

where λ is a supplied constant and $\mathbf{n}$ the outward normal to the surface, in the previously derived relation

$$S_{kj}\phi_{ijk} = \int_{S_{kj}} (\phi^-(y,z) - \frac{\alpha_i}{2}\frac{\partial}{\partial x}\phi^-(y,z))dS + \frac{S_{kj}}{8}(\alpha_i^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} + \ldots \tag{19}$$

leads to

$$D_{ijk}\int_{S_{kj}} \frac{\partial}{\partial x}\phi^-(y,z)dS = \frac{S_{kj}D_{ijk}\lambda}{(D_{ijk} + \frac{\alpha_i}{2}\lambda)}(-\phi_{ijk} + \frac{1}{8}(\alpha_i^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} + \ldots \tag{20}$$

CHAPTER 5 DERIVATION OF THE SECOND ORDER MESH CORRECTION

Let $D_{i+\frac{1}{2}jk} = \dfrac{2D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}\alpha_{i+1} + D_{i+1jk}\alpha_i)}$ or $\dfrac{D_{ijk}\lambda S_{kj}}{(D_{ijk} + \frac{\alpha_i}{2}\lambda)}$ as appropriate with similar expressions for j and k directions then on substituting formulæ (6) and (12) in relation (4) the difference equation for cell i,j,k becomes

$$\begin{aligned}
&D_{i+\frac{1}{2}jk}(\phi_{i+1jk} - \phi_{ijk}) + D_{i-\frac{1}{2}jk}(\phi_{i-1jk} - \phi_{ijk}) + D_{ij+\frac{1}{2}k}(\phi_{ij+1k} - \phi_{ijk}) + D_{ij-\frac{1}{2}k}(\phi_{ij-1k} - \phi_{ijk}) \\
&+ D_{ijk+\frac{1}{2}}(\phi_{ijk+1} - \phi_{ijk}) + D_{ijk-\frac{1}{2}}(\phi_{ijk-1} - \phi_{ijk}) + (\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk}V_{ijk}\phi_{ijk} = \\
&\quad -\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk}(\alpha_i^2\frac{\partial^2}{\partial x^2} + \beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial y^2})\phi_{ijk} \\
&\quad + \frac{1}{8}\begin{pmatrix} D_{i+\frac{1}{2}jk}\frac{\partial^2}{\partial x^2}(\alpha_{i+1}^2\phi^+_{i+\frac{1}{2}jk} - \alpha_1^2\phi^-_{i+\frac{1}{2}jk}) - D_{i-\frac{1}{2}jk}\frac{\partial^2}{\partial x^2}(\alpha_i^2\phi^+_{i-\frac{1}{2}jk} - \alpha_{i-1}^2\phi^-_{i-\frac{1}{2}jk}) \\ +D_{ij+\frac{1}{2}k}\frac{\partial^2}{\partial y^2}(\beta_{j+1}^2\phi^+_{ij+\frac{1}{2}k} - \beta_j^2\phi^-_{ij+\frac{1}{2}k}) - D_{ij-\frac{1}{2}k}\frac{\partial^2}{\partial y^2}(\beta_j^2\phi^+_{ij-\frac{1}{2}k} - \beta_{j-1}^2\phi^-_{ij-\frac{1}{2}k}) \\ +D_{ijk+\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma_{k+1}^2\phi^+_{ijk+\frac{1}{2}} - \gamma_k^2\phi^-_{ijk+\frac{1}{2}}) - D_{ijk-\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma_k^2\phi^+_{ijk-\frac{1}{2}} - \gamma_{k-1}^2\phi^-_{ijk-\frac{1}{2}}) \end{pmatrix}
\end{aligned} \tag{21}$$

recalling that at boundary mesh regions fluxes outside the system are omitted with the differential terms becoming, as a consequence of equation (20), say for the right hand x limit

$$-D_{i+\frac{1}{2}jk}(\alpha_i^2\frac{\partial^2}{\partial x^2}-\frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2}+\gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} \qquad (22).$$

The term on the left hand side of relation (21) is the standard finite difference approximation to the diffusion equation(3). Let ϕ^*_{ijk} be the row vector adjoint solution so that

$$\begin{aligned} &D_{i+\frac{1}{2}jk}(\phi^*_{i+1jk}-\phi^*_{ijk})+D_{i-\frac{1}{2}jk}(\phi^*_{i-1jk}-\phi^*_{ijk})+D_{ij+\frac{1}{2}k}(\phi^*_{ij+1k}-\phi^*_{ijk})+D_{ij-\frac{1}{2}k}(\phi^*_{ij-1k}-\phi^*_{ijk}) \\ &+D_{ijk+\frac{1}{2}}(\phi^*_{ijk+1}-\phi^*_{ijk})+D_{ijk-\frac{1}{2}}(\phi^*_{ijk-1}-\phi^*_{ijk})+\phi^*_{ijk}(\frac{\nu\sigma_f}{k_1}+\sigma_s-\sigma_r)_{ijk}V_{ijk}=0 \end{aligned} \qquad (23).$$

On multiplying the above by the flux on the right then subtracting from Equation (21) multiplied by the adjoint, after summing over all subscripts the following results

$$\begin{aligned} &(\frac{1}{k}-\frac{1}{k_1})\sum_{ijk}V_{ijk}\phi^*_{ijk}(\nu\sigma_f)_{ijk}\phi_{ijk}= \\ &\quad -\sum_{ijk}\phi^*_{ijk}\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}(\alpha_i^2\frac{\partial^2}{\partial x^2}+\beta_j^2\frac{\partial^2}{\partial y^2}+\gamma_k^2\frac{\partial^2}{\partial y^2})\phi_{ijk} \\ &+\sum_{ijk}\frac{\phi^*_{ijk}}{8}\begin{pmatrix} D_{i+\frac{1}{2}jk}\frac{\partial^2}{\partial x^2}(\alpha^2_{i+1}\phi^+_{i+\frac{1}{2}jk}-\alpha_i^2\phi^-_{i+\frac{1}{2}jk})-D_{i-\frac{1}{2}jk}\frac{\partial^2}{\partial x^2}(\alpha_i^2\phi^+_{i-\frac{1}{2}jk}-\alpha^2_{i-1}\phi^-_{i-\frac{1}{2}jk}) \\ +D_{ij+\frac{1}{2}k}\frac{\partial^2}{\partial y^2}(\beta^2_{j+1}\phi^+_{ij+\frac{1}{2}k}-\beta_j^2\phi^-_{ij+\frac{1}{2}k})-D_{ij-\frac{1}{2}k}\frac{\partial^2}{\partial y^2}(\beta_j^2\phi^+_{ij-\frac{1}{2}k}-\beta^2_{j-1}\phi^-_{ij-\frac{1}{2}k}) \\ +D_{ijk+\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma^2_{k+1}\phi^+_{ijk+\frac{1}{2}}-\gamma_k^2\phi^-_{ijk+\frac{1}{2}})-D_{ijk-\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma_k^2\phi^+_{ijk-\frac{1}{2}}-\gamma^2_{k-1}\phi^-_{ijk-\frac{1}{2}}) \end{pmatrix} \\ &\qquad +boundary\ \ contributions \end{aligned} \qquad (24).$$

The above gives a formula for the second order change in eigenvalue due to mesh effects but not in a convenient form as interface fluxes occur.

CHAPTER 6 SIMPLIFICATION OF THE MESH CORRECTION FORMULA

The first term on the right in relation (24) cannot really be simplified but the leakage contribution appears unwieldy and worth investigating. For clarity only the x coordinate will be dealt with, as the other two axes have the same treatment. In the large bracket the interface values may be taken at the appropriate cell mesh point to second order. $\frac{\partial^2 \phi_{ijk}}{\partial x^2}$ terms can then be collected to give

$$\frac{1}{8}\sum_{ijk}\left((D_{i+\frac{1}{2}jk}(\phi^*_{i+1jk}-\phi^*_{ijk})+D_{i-\frac{1}{2}jk}(\phi^*_{i-1jk}-\phi^*_{ijk}))\right)\alpha_i^2\frac{\partial^2\phi_{ijk}}{\partial x^2} \qquad (25).$$

The expression in brackets is just the adjoint x axis leakage component as shown in equation (23). Now, since derivatives parallel to boundaries are continuous across those boundaries, add in

$$\begin{aligned}&(D_{ij+\frac{1}{2}k}(\phi^*_{ij+1k}-\phi^*_{ijk}))\alpha_i^2\frac{\partial^2\phi_{ij+\frac{1}{2}k}}{\partial x^2}+D_{ij-\frac{1}{2}k}(\phi^*_{ij-1k}-\phi^*_{ijk})\alpha_i^2\frac{\partial^2\phi_{ij-\frac{1}{2}k}}{\partial x^2}\\&+(D_{ijk+\frac{1}{2}}(\phi^*_{ijk+1}-\phi^*_{ijk}))\alpha_i^2\frac{\partial^2\phi_{ijk+\frac{1}{2}}}{\partial x^2}+D_{ijk-\frac{1}{2}}(\phi^*_{ijk-1}-\phi^*_{ijk})\alpha_i^2\frac{\partial^2\phi_{ijk-\frac{1}{2}}}{\partial x^2}\end{aligned} \qquad (26)$$

to point ijk. As derivatives parallel to the interface are continuous the superscript on the flux can be omitted. Clearly the foregoing procedure will not affect the value of formula (24) as zero is being added to it, with values from adjacent mesh points canceling. Then, to second order, the interface flux terms can be replaced by ijk values and expression (23) becomes with the added terms

$$\frac{1}{8}\sum_{ijk}\begin{pmatrix}D_{i+\frac{1}{2}jk}(\phi^*_{i+1jk}-\phi^*_{ijk})+D_{i-\frac{1}{2}jk}(\phi^*_{i-1jk}-\phi^*_{ijk})\\+D_{ij+\frac{1}{2}k}(\phi^*_{ij+1k}-\phi^*_{ijk})+D_{ij-\frac{1}{2}k}(\phi^*_{ij-1k}-\phi^*_{ijk})\\+D_{ijk+\frac{1}{2}}(\phi^*_{ijk+1}-\phi^*_{ijk})+D_{ijk-\frac{1}{2}}(\phi^*_{ijk-1}-\phi^*_{ijk})\end{pmatrix}\alpha_i^2\frac{\partial^2\phi_{ijk}}{\partial x^2}. \qquad (28).$$

The brackets contain the adjoint leakage term of the standard approximation in relation (23).

To complete the process so that by using equation (23) the adjoint leakage contribution can be replaced by minus the volume term, a modification must be made at boundaries. Thus for example at the outer y limit

$$-\frac{1}{8}D_{iJ+\frac{1}{2}k}\phi^{*}_{iJk}\alpha_i^2\frac{\partial^2\phi_{iJk}}{\partial x^2} \tag{29}$$

is required, on assuming there are J mesh points in that direction. Hence the whole expression becomes

$$\begin{aligned}-\frac{1}{8}\sum_{ijk}V_{ijk}\phi^{*}_{ijk}\left(\frac{\nu\sigma_f}{k_1}+\sigma_s-\sigma_r\right)_{ijk}\alpha_i^2\frac{\partial^2\phi_{ijk}}{\partial x^2}\\ +\frac{1}{6}\sum_{ik,(j\text{ on boundary})}\alpha_i^2 D_{ij\pm\frac{1}{2}k}\phi^{*}_{ijk}\frac{\partial^2\phi_{ijk}}{\partial x^2}+\frac{1}{6}\sum_{ij,(k\text{ on boundary})}\alpha_i^2 D_{ijk\pm\frac{1}{2}}\phi^{*}_{ijk}\frac{\partial^2\phi_{ijk}}{\partial x^2}\end{aligned} \tag{30}$$

where the last two summations imply the y and z axis boundaries respectively. The factor $\frac{1}{6}$ results because of the term already present as shown in expression (20) and the $\pm$ becomes minus at the axis origin and plus for an outer limit.

Finally if $k_1 = k+\delta k$ then

$$\frac{\delta k}{k}=\frac{1}{6}\frac{-\sum_{ijk}V_{ijk}\phi^{*}_{ijk}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\alpha_i^2\frac{\partial^2}{\partial x^2}+\beta_j^2\frac{\partial^2}{\partial y^2}+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk}+\sum_{\substack{mn,(l\text{ on boundary})\\ perm(l,m,n)}}\phi^{*}_{lmn}D_{l\pm\frac{1}{2}mn}w_l^2\left(\frac{\partial^2}{\partial w^2_{(m)}}+\frac{\partial^2}{\partial w^2_{(n)}}\right)\phi_{lmn}}{\sum_{ijk}V_{ijk}\phi^{*}_{ijk}\frac{(\nu\sigma_f)_{ijk}}{k}\phi_{ijk}} \tag{31}$$

on adding in the existing term in relation (6) and neglecting the difference between k and k_1 in the denominator. The term $perm(l,m,n)$ means permute over the indices in the bracket and $w_{(n)}$ denotes the mesh at the boundary in the n direction parallel to the mesh cell face with the plus or minus sign in the diffusion coefficient taken at outer and inner boundaries respectively

Although the flux is the zero mesh value, to second order the values calculated from the usual approximation can be used and to a similar accuracy, from formula (12) for $i-1,i$ and $i,i+1$,

$$\alpha_i^2\frac{\partial^2}{\partial x^2}\phi_{ijk}=\alpha_i D^{-1}_{ijk}\left(D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})+D_{i-\frac{1}{2}jk}(\phi_{i-1jk}-\phi_{ijk})\right) \tag{32}$$

on using the diffusion coefficients defined for equation (21) so that the expression can be evaluated to second order with fluxes from the standard finite difference solution.

Further, as has been shown by many workers(1,2), if meshes are reduced by a factor then again to second order, δk is reduced by that factor squared for practical problems where the leakage is small and the second term in the numerator of relation (31) is ignorable.

In deriving the above, superficially, the leakage terms in equation (24) appear to be regroupable in third order contributions, and of course do cancel if there is no material interface. However care has to be exercised since differentiation by x in the x direction increases the order of singularity at material interfaces and hence, as explained in the Introduction, such terms would not be neglectable since Taylor's expansion is not valid across the boundary.

CHAPTER 7 OTHER GEOMETRIES

Two other lattice geometries have been investigated Triangular-Z where triangles are equilateral and Hexagonal-Z as these are commonly used in fast reactor studies. Normally in the x,y plane these variables are replaced by s,t and u directions as shown in Figures 2 and 3 since meshpoints at the centroid of triangles and center of hexagons lie on these axes. Firstly Triangular-Z will be dealt with.

7.1 TRIANGULAR-Z GEOMETRY

From Figure 2 the transformations in the x,y plane are clearly

$$y=-t+\frac{s}{2}=-t+\frac{u}{2},\quad x=s\frac{\sqrt{3}}{2}=-u\frac{\sqrt{3}}{2}\qquad t=-y+\frac{x}{\sqrt{3}},\quad s=\frac{2x}{\sqrt{3}},\quad u=-\frac{2x}{\sqrt{3}}$$

and as, from differential geometry theory, or by direct evaluation

$$(x\frac{\partial}{\partial x}+y\frac{\partial}{\partial y})^n=(s\frac{\partial}{\partial s}+t\frac{\partial}{\partial t})^n, \tag{33}$$

for the shaded triangle in Figure 2 the second order error terms in the plane, with a the side length, are

$$\frac{1}{2}\int_V(x^2\frac{\partial^2}{\partial x^2}+xy\frac{\partial^2}{\partial x\partial y}+\frac{\partial^2}{\partial y^2})\phi_{ijk}dxdy=\frac{\sqrt{3}}{4}\int_{-\frac{a}{\sqrt{3}}}^{0}\int_0^{\frac{a}{2\sqrt{3}}+\frac{t}{2}}(s^2\frac{\partial^2}{\partial s^2}+st\frac{\partial^2}{\partial s\partial t}+t^2\frac{\partial^2}{\partial t^2})\phi_{ijk}dsdt \tag{34}$$

since the Jacobian, $\begin{Vmatrix}\frac{\partial x}{\partial s} & \frac{\partial x}{\partial t}\\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t}\end{Vmatrix}=\frac{\sqrt{3}}{2}$ and integral limits are interdependent.

By standard integration methods

$$\frac{\sqrt{3}}{4}\int_{-\frac{a}{\sqrt{3}}}^{0}\int_{0}^{\frac{a}{2\sqrt{3}}+\frac{t}{2}} s^2\frac{\partial^2\phi_{ijk}}{\partial s^2}dsdt=\frac{\sqrt{3}}{4}\frac{a^4}{864}\frac{\partial^2\phi_{ijk}}{\partial s^2} \qquad (35)$$

and

$$\frac{\sqrt{3}}{4}\int_{-\frac{a}{\sqrt{3}}}^{0}\int_{0}^{\frac{a}{2\sqrt{3}}+\frac{t}{2}} t^2\frac{\partial^2\phi_{ijk}}{\partial s^2}dsdt=\frac{\sqrt{3}}{4}\frac{a^4}{216}\frac{\partial^2\phi_{ijk}}{\partial t^2} \qquad (36).$$

Similar expressions are found for the other five triangles bounded by the s,t,u coordinates and cross terms vanish because of the relation between s,t and u.

As each sub-triangle appears twice with the same variables, the volume term for the plane comes to

$$A\frac{5a^2}{432}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)\phi_{ijk} \qquad (37)$$

where $A=\frac{\sqrt{3}a^2}{4}$, the area of the mesh triangle.

Clearly integrals in the z direction are unaffected and the whole volume term takes a familiar form from xyz geometry (see expression (6)),

$$\frac{V_{ijk}}{24}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\frac{5a^2}{18}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk} \qquad (38).$$

Again a procedure paralleling the previous work is adopted for the leakage contribution although in the derivation some modification of the coordinate transformation will be employed. First using the local (v,w) coordinates shown in Figure 3

$$v=s-\frac{t}{2}=s-\frac{u}{2},\quad w=-\frac{\sqrt{3}t}{2}=\frac{\sqrt{3}u}{2}$$

and

$$s=v-\frac{w}{\sqrt{3}},\quad t=-\frac{2w}{\sqrt{3}},\quad u=\frac{2w}{\sqrt{3}}.$$

With these local coordinates the leakage term in the s direction may be taken directly from equation (12) by substitution of v and w for x and y respectively and recalling from Figure 2 that the mesh length in terms of the triangle side a is $\frac{a}{\sqrt{3}}$

$$D_{ijk}\int_{S_{jk}}\frac{\partial\phi^{-}(w,z)}{\partial v}dS=\frac{2\sqrt{3}D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}a+D_{i+1jk}a)}(\phi_{i+1jk}-\phi_{ijk}+\frac{a^2}{24}(\frac{\partial^2\phi^{-}_{i+\frac{1}{2}jk}}{\partial v^2}-\frac{\partial^2\phi^{+}_{i+\frac{1}{2}}}{\partial v^2})+\ldots) \quad (39),$$

and at boundaries

$$D_{ijk}\int_{S_{kj}}\frac{\partial}{\partial v}\phi^{-}(y,z)dS=\frac{S_{kj}D_{ijk}\lambda}{(D_{ijk}+\frac{a}{2\sqrt{3}}\lambda)}(-\phi_{ijk}+\frac{1}{24}(a^2\frac{\partial^2}{\partial v^2}-a^2\frac{\partial^2}{\partial w^2}-\gamma_k^2\frac{\partial^2}{\partial z^2})\phi^{-}_{i+\frac{1}{2}jk}+\ldots \quad (40)$$

but

$$\frac{\partial}{\partial v}=\frac{\partial s}{\partial v}\frac{\partial}{\partial s}+\frac{\partial t}{\partial v}\frac{\partial}{\partial t}=\frac{\partial}{\partial s},\qquad \frac{\partial^2}{\partial v^2}=\frac{\partial^2}{\partial s^2} \quad (41)$$

so that the difference equation may be written after repeating the procedure for the other two sides of the triangle

$$\begin{aligned}
&D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})+D_{i-\frac{1}{2}}(\phi_{i-1jk}-\phi_{ijk})+D_{ij\pm\frac{1}{2}k}(\phi_{ij\pm1k}-\phi_{ijk})\\
&+D_{ijk+\frac{1}{2}}(\phi_{ijk+1}-\phi_{ijk})+D_{ijk-\frac{1}{2}}(\phi_{ijk-1}-\phi_{ijk})+(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}V_{ijk}\phi_{ijk}=\\
&\quad-\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}(\frac{5a^2}{18}(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})+\gamma_k^2\frac{\partial^2}{\partial z^2})\phi_{ijk}\\
&+\frac{1}{8}\left(\begin{array}{l}\frac{a^2}{3}\left(D_{i+\frac{1}{2}jk}\frac{\partial^2}{\partial s^2}(\phi^{+}_{i+\frac{1}{2}jk}-\phi^{-}_{i+\frac{1}{2}jk})-D_{i-\frac{1}{2}jk}\frac{\partial^2}{\partial u^2}(\phi^{+}_{i-\frac{1}{2}jk}-\phi^{-}_{i-\frac{1}{2}jk})+D_{ij\pm\frac{1}{2}k}\frac{\partial^2}{\partial t^2}(\phi^{+}_{ij\pm\frac{1}{2}k}-\phi^{-}_{ij\pm\frac{1}{2}k})\right)\\
+D_{ijk+\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma^2_{k+1}\phi^{+}_{ijk+\frac{1}{2}}-\gamma^2_k\phi^{-}_{ijk+\frac{1}{2}})-D_{ijk-\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma^2_k\phi^{+}_{ijk-\frac{1}{2}}-\gamma^2_{k-1}\phi^{-}_{ijk-\frac{1}{2}})\end{array}\right)
\end{aligned} \quad (42)$$

where± is used to denote whether the triangle points in the positive or negative y direction in an obvious manner.

The procedure of the rectangular case is now repeated, First multiplying by the adjoint then adding a third order term to enable the adjoint flux leakage expression to be formed at each mesh point and finally substituting for it with minus the central contribution.

However because the coordinate axes in (s,t,u) are not orthogonal some care has to be exercised in finding a second differential which is continuous across the triangle boundaries. Clearly, from Figure 3, $\frac{\partial^2}{\partial w^2}$ is since the coordinate is parallel to the triangle side. Now by the chain rule for

differentiation

$$\frac{\partial^2}{\partial t^2}=\frac{1}{4}\frac{\partial^2}{\partial v^2}+\frac{\sqrt{3}}{2}\frac{\partial^2}{\partial v\partial w}+\frac{3}{4}\frac{\partial^2}{\partial w^2}$$

and

$$\frac{\partial^2}{\partial u^2}=\frac{1}{4}\frac{\partial^2}{\partial v^2}-\frac{\sqrt{3}}{2}\frac{\partial^2}{\partial v\partial w}+\frac{3}{4}\frac{\partial^2}{\partial w^2}$$

so that

$$\frac{\partial^2}{\partial w^2}=\frac{1}{3}\left(2(\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})-\frac{\partial^2}{\partial s^2}\right) \qquad (43)$$

on using the second of relations (41)

By interchanging the variables continuous derivatives across the other sides can be obtained in terms of (s,t,u) in an obvious manner.

Now equation (42) is multiplied by the adjoint solution of the standard approximation and the same procedure as XYZ geometry followed to give

$$(\frac{1}{k}-\frac{1}{k_1})\sum_{ijk}V_{ijk}\phi^*_{ijk}(\nu\sigma_f)_{ijk}\phi_{ijk}=$$

$$-\sum_{ijk}\phi^*_{ijk}\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}\left(\frac{5a^2}{18}(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})+\gamma_k^2\frac{\partial^2}{\partial y^2}\right)\phi_{ijk}$$

$$+\sum_{ijk}\frac{\phi^*_{ijk}}{8}\left(\begin{array}{l}\frac{a^2}{3}\left(D_{i+\frac{1}{2}jk}\frac{\partial^2}{\partial s^2}(\phi^+_{i+\frac{1}{2}jk}-\phi^-_{i+\frac{1}{2}jk})-D_{i-\frac{1}{2}jk}\frac{\partial^2}{\partial u^2}(\phi^+_{i-\frac{1}{2}jk}-\phi^-_{i-\frac{1}{2}jk})+D_{ij\pm\frac{1}{2}k}\frac{\partial^2}{\partial y^2}(\phi^+_{ij\pm\frac{1}{2}k}-\phi^-_{ij\pm\frac{1}{2}k})\right)\\ +D_{ijk+\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\phi^+_{ijk+\frac{1}{2}}-\phi^-_{ijk+\frac{1}{2}})-D_{ijk-\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\phi^+_{ijk-\frac{1}{2}}-\phi^-_{ijk-\frac{1}{2}})\end{array}\right)$$

$$+\ \textit{boundary contributions} \qquad (44).$$

Clearly any factor may multiply expression (43) which is then multiplied by the adjoint gradient and added in, however unity will be seen to be the best. Thus in the s direction the quantity

$$\frac{a^2}{24}D_{i+\frac{1}{2}jk}(\phi^*_{i+1jk}-\phi^*_{ijk})\frac{1}{3}\left(2(\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})-\frac{\partial^2}{\partial s^2}\right)\phi_{i+\frac{1}{2}jk} \qquad (45)$$

is added to terms with subscript ijk and subtracted from $i+1jk$ and of course this will not alter the sum. Repeating this procedure at the other sides with appropriate interchange of coordinates gives for the second flux derivative with respect to s, since derivatives at the interface may be replaced by values at the mesh point without loss of second order accuracy,

$$\frac{a^2}{36}\begin{pmatrix} D_{i+\frac{1}{2}jk}(\phi^*_{i+1jk}-\phi^*_{ijk})+D_{i-\frac{1}{2}jk}(\phi^*_{i-1jk}-\phi^*_{ijk})+D_{ij\pm\frac{1}{2}k}(\phi^*_{ij\pm1k}-\phi^*_{ijk}) \\ +D_{ijk+\frac{1}{2}}(\phi^*_{ijk+1}-\phi^*_{ijk})+D_{ijk-\frac{1}{2}}(\phi^*_{ijk-1}-\phi^*_{ijk}) \end{pmatrix}\frac{\partial^2\phi_{ijk}}{\partial s^2} \qquad (46)$$

where the z axis contribution is obviously the same as the rectangular geometry case.

Repeating the above for the other directions in the plane and replacing the adjoint leakage with the volume term gives finally the eigenvalue mesh effect as

$$\frac{\delta k}{k}=\frac{1}{6}\frac{-\sum_{ijk}V_{ijk}\phi^*_{ijk}(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}(\frac{17a^2}{72}(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})+\gamma_k^2\frac{\partial^2}{\partial z^2})\phi_{ijk}+\sum_{(lmn)\,l\ on\ boundary}\phi^*_{lmn}D_{l+\frac{1}{2}mn}w_l^2(\varepsilon_m\frac{\partial^2}{\partial w^2_{(m)}}+\varepsilon_n\frac{\partial^2}{\partial w^2_{(n)}}+\varepsilon_p\frac{\partial^2}{\partial w^2_{(p)}})\phi_{lmn}}{\sum_{ijk}V_{ijk}\phi^*_{ijk}\frac{(\nu\sigma_f)_{ijk}}{k}\phi_{ijk}} \qquad (47)$$

where ε_i is unity for the z variable and $\frac{1}{3}$ otherwise and (lmn) implies permuting the indices so all boundaries are included. Subscript p does not occur in the diffusion coefficient $D_{l+\frac{1}{2}mn}$, however the meaning is clear in that the original enumeration of meshpoints has been retained but because three axes have been utilised in the xy plane a third second order derivative is involved.

7.2 HEXAGONAL-Z GEOMETRY

Hexagonal geometry uses the same coordinates in the plane as triangular geometry as illustrated in Figure 4. Volume integrals now take the form, with a the distance across hexagon flats,

$$\frac{\sqrt{3}}{4}\int_{-\frac{a}{3}}^{0}dt\int_{-t}^{\frac{a}{2}+\frac{t}{2}}s^2ds=\frac{\sqrt{3}}{96}a^4\left(\frac{19}{108}\right) \qquad (48)$$

and

$$\frac{\sqrt{3}}{4}\int_{-\frac{a}{3}}^{0}t^2dt\int_{-t}^{\frac{a}{2}+\frac{t}{2}}ds=\frac{\sqrt{3}}{96}a^4\left(\frac{1}{27}\right) \qquad (49).$$

The above integrals appear 4 times for each variable giving a final volume term, since the hexagon area is $\frac{\sqrt{3}a^2}{2}$, of

$$\frac{V_{ijk}}{24}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\frac{23a^2}{54}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk} \quad (50).$$

Again adding the term (43) at *ijk* but now for six directions then taking values at that meshpoint which does not alter the second order accuracy, using the definitions of the Triangular-Z case but with $\varepsilon_n=\frac{3}{4}$ for coordinates other than z the final formula obtained is

$$\frac{\delta k}{k}=\frac{1}{6}\frac{-\sum_{ijk}V_{ijk}\phi^*_{ijk}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\frac{131a^2}{216}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk}+\sum_{(lmn)l\ on\ boundary}\phi^*_{lmn}D_{l+\frac{1}{2}mn}w_l^2\left(\varepsilon_m\frac{\partial^2}{\partial w^2_{(m)}}+\varepsilon_n\frac{\partial^2}{\partial w^2_{(n)}}+\varepsilon_p\frac{\partial^2}{\partial w^2_{(p)}}\right)\phi_{lmn}}{\sum_{ijk}V_{ijk}\phi^*_{ijk}\frac{(\nu\sigma_f)_{ijk}}{k}\phi_{ijk}} \quad (51).$$

CHAPTER 8 DERIVATION OF THE FINITE DIFFERENCE EQUATIONS FOR SOLUTION

As indicated in Chapter 2, Figure 1 shows the normal mesh centered approximation for XYZ geometry and derivations of difference equations depend on the Taylor expansion which may be written as

$$\phi(x+\delta x,y+\delta y,z+\delta z)=\phi(x,y,z)+\sum_{n=1}^{\infty}\sum_{\substack{i,j,k\\ i+j+k=n}}\frac{\delta x^i\delta y^j\delta z^k}{i!j!k!}\frac{\partial^n\phi(x,y,z)}{\partial x^i\partial y^j\partial z^k} \quad (1).$$

Where equations are repeated from earlier in the text the original numbering will be used so that the full derivation or more detailed description can be conveniently found.

Using the definitions of Chapter 2 the diffusion equation in the form

$$\nabla D\nabla\phi+\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)\phi=0 \quad (3),$$

will be solved that is where ϕ is the column vector of fluxes, D the diffusion coefficient diagonal matrix and $\nu\sigma_f$, σ_s and σ_r the production, scatter and removal (absorption plus scatter out of the group) cross section matrices respectively with k the eigenvalue. Dependence on the space variables x,y,z is understood. Also the flux ϕ and current, $-D\nabla\phi$, are continuous everywhere in the problem and at external boundaries the relation $\mathbf{n}.D\nabla\phi = -\lambda\phi$ is imposed where $\mathbf{n}$ is the outward normal to the surface, these restrictions being the usual diffusion theory subsidiary continuity conditions(3).

Equation (3) is then integrated over each mesh region yielding typically

$$\sum_{l=1,6}\int_{S_l} D\nabla\phi.\mathbf{dS} + \int_{v_{ijk}} (\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)\phi dV = 0 \qquad (4)$$

where l denotes the mesh box boundaries in some order.

Using Taylor's expansion the volume term for mesh region i,j,k takes the form shown in equation (6)

$$(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk} V_{ijk}(\phi_{ijk} + \frac{1}{24}(\alpha_i^2\frac{\partial^2}{\partial x^2} + \beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2})\phi_{ijk} + \ldots) \qquad (6)$$

when using the mesh intervals $\alpha_i, \beta_j, \gamma_k$ specified in Figure 1 with V_{ijk} the cell volume. The leakage term from points (i,j,k) to $(i+1,j,k)$ approximates as equation (12) repeated below

$$D_{ijk}\int_{S_{jk}} \frac{\partial\phi^-(y,z)}{dx}dS = \frac{2D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}\alpha_{i+1} + D_{i+1jk}\alpha_i)}(\phi_{i+1jk} - \phi_{ijk} + \frac{1}{8}(\alpha_i^2\frac{\partial^2\phi^-_{i+\frac{1}{2}jk}}{\partial x^2} - \alpha_{i+1}^2\frac{\partial^2\phi^+_{i+\frac{1}{2}jk}}{\partial x^2}) + \ldots) \qquad (12)$$

or at an external boundary from Chapter 4

$$D_{ijk}\int_{S_{kj}} \frac{\partial}{\partial x}\phi^-(y,z)dS = \frac{S_{kj}D_{ijk}\lambda}{(D_{ijk} + \frac{\alpha_i}{2}\lambda)}(-\phi_{ijk} + \frac{1}{8}(\alpha_i^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} + \ldots (20).$$

CHAPTER 9 THE SECOND ORDER MESH CORRECTION IN DIFFERENCE FORM

As before set $D_{i+\frac{1}{2}jk} = \dfrac{2D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}\alpha_{i+1} + D_{i+1jk}\alpha_i)}$ or $\dfrac{D_{ijk}\lambda S_{kj}}{(D_{ijk} + \frac{\alpha_i}{2}\lambda)}$ at a boundary with similar expressions for j and k directions then the difference equation for cell i,j,k becomes on substituting relations (12) and (6) in equation (3)

$$D_{i+\frac{1}{2}jk}(\phi_{i+1jk} - \phi_{ijk}) + D_{i-\frac{1}{2}jk}(\phi_{i-1jk} - \phi_{ijk}) + D_{ij+\frac{1}{2}k}(\phi_{ij+1k} - \phi_{ijk}) + D_{ij-\frac{1}{2}k}(\phi_{ij-1k} - \phi_{ijk})$$

$$+ D_{ijk+\frac{1}{2}}(\phi_{ijk+1} - \phi_{ijk}) + D_{ijk-\frac{1}{2}}(\phi_{ijk-1} - \phi_{ijk}) + (\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk} V_{ijk}\phi_{ijk} =$$

$$-\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r)_{ijk}(\alpha_i^2\frac{\partial^2}{\partial x^2} + \beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial y^2})\phi_{ijk}$$

$$+\frac{1}{8}\begin{pmatrix} D_{i+\frac{1}{2}jk}\dfrac{\partial^2}{\partial x^2}(\alpha_{i+1}^2\phi^+_{i+\frac{1}{2}jk} - \alpha_1^2\phi^-_{i+\frac{1}{2}jk}) - D_{i-\frac{1}{2}jk}\dfrac{\partial^2}{\partial x^2}(\alpha_i^2\phi^+_{i-\frac{1}{2}jk} - \alpha_{i-1}^2\phi^-_{i-\frac{1}{2}jk}) \\ +D_{ij+\frac{1}{2}k}\dfrac{\partial^2}{\partial y^2}(\beta_{j+1}^2\phi^+_{ij+\frac{1}{2}k} - \beta_j^2\phi^-_{ij+\frac{1}{2}k}) - D_{ij-\frac{1}{2}k}\dfrac{\partial^2}{\partial y^2}(\beta_j^2\phi^+_{ij-\frac{1}{2}k} - \beta_{j-1}^2\phi^-_{ij-\frac{1}{2}k}) \\ +D_{ijk+\frac{1}{2}}\dfrac{\partial^2}{\partial z^2}(\gamma_{k+1}^2\phi^+_{ijk+\frac{1}{2}} - \gamma_k^2\phi^-_{ijk+\frac{1}{2}}) - D_{ijk-\frac{1}{2}}\dfrac{\partial^2}{\partial z^2}(\gamma_k^2\phi^+_{ijk-\frac{1}{2}} - \gamma_{k-1}^2\phi^-_{ijk-\frac{1}{2}}) \end{pmatrix} \quad (21)$$

recalling that at boundary mesh regions, fluxes outside the system are omitted with the differential terms becoming, as a consequence of equation (20), say for the right hand x-axis limit

$$-D_{i+\frac{1}{2}jk}(\alpha_i^2\frac{\partial^2}{\partial x^2} - \frac{1}{3}(\beta_j^2\frac{\partial^2}{\partial y^2} + \gamma_k^2\frac{\partial^2}{\partial z^2}))\phi^-_{i+\frac{1}{2}jk} \quad (22).$$

The left hand side of equation (21) forms the standard mesh centered finite difference approximation, so it seems sensible, since the mesh correction is a small effect, to use the right hand side as a perturbative term and incorporate it in the outer iteration fission source evaluation (see Chapter 11 for details of the flux solution). Consequently difference approximations for the terms on the right hand side are required. Straightforwardly from Taylor's theorem, to second order,

$$\frac{\partial^2\phi_{ijk}}{\partial x^2} = \frac{1}{\alpha_i}\left(\frac{\partial\phi^-_{i+\frac{1}{2}jk}}{\partial x} - \frac{\partial\phi^+_{i-\frac{1}{2}jk}}{\partial x}\right) \quad (52)$$

and since, for example $D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})$ in equation (21) is an approximation to $D\frac{\partial\phi}{\partial x}$ at the interface then neglecting higher orders

$$\frac{\partial\phi^-_{i+\frac{1}{2}jk}}{\partial x}=\frac{D_{i+\frac{1}{2}jk}}{D_{ijk}}(\phi_{i+1jk}-\phi_{ijk}),\qquad \frac{\partial\phi^+_{i-\frac{1}{2}jk}}{\partial x}=\frac{D_{i-\frac{1}{2}jk}}{D_{ijk}}(\phi_{ijk}-\phi_{i-1jk}) \tag{53}$$

and hence

$$\begin{aligned}\frac{\partial^2\phi_{ijk}}{\partial x^2}&=\frac{1}{\alpha_i D_{ijk}}(D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})+D_{i-\frac{1}{2}jk}(\phi_{ijk}-\phi_{i-1jk}))\\&=\Delta_x^2\phi_{ijk}\end{aligned} \tag{54}$$

say, with similar expressions for the other second order derivatives. Thus the term on the right hand side of equation (21) becomes in difference form

$$\begin{aligned}&-\frac{V_{ijk}}{24}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}(\alpha_i^2\Delta_x^2+\beta_j^2\Delta_y^2+\gamma_k^2\Delta_z^2)\phi_{ijk}\\&+\frac{1}{8}\begin{pmatrix}D_{i+\frac{1}{2}jk}\Delta_x^2(\alpha_{i+1}^2\phi_{i+1jk}-\alpha_i^2\phi_{ijk})-D_{i-\frac{1}{2}jk}\Delta_x^2(\alpha_i^2\phi_{ijk}-\alpha_{i-1}^2\phi_{i-1jk})\\+D_{ij+\frac{1}{2}k}\Delta_y^2(\beta_{j+1}^2\phi_{ij+1k}-\beta_j^2\phi_{ijk})-D_{ij-\frac{1}{2}k}\Delta_y^2(\beta_j^2\phi_{ijk}-\beta_{j-1}^2\phi_{ij-1k})\\+D_{ijk+\frac{1}{2}}\Delta_z^2(\gamma_{k+1}^2\phi_{ijk+1}-\gamma_k^2\phi_{ijk})-D_{ijk-\frac{1}{2}}\Delta_z^2(\gamma_k^2\phi_{ijk}-\gamma_{k-1}^2\phi_{ijk-1})\end{pmatrix}.\end{aligned} \tag{55}$$

Since there are only values at mesh points in the above it can be evaluated without difficulty and provides the required source term to correct for second order mesh effects, bearing in mind the modifications at external boundaries from equation (22) in which, to second order, meshpoint values can be used without loss of accuracy.

CHAPTER 10 TRIANGULAR-Z GEOMETRY

Triangular-Z geometry assumes equilateral triangles in the x,y plane so that it is natural as noted in Chapter 7 to consider axes s,t,u parallel to the lines defining this triangular mesh. From Figure 2 the transformations in the x,y plane to these new axes are clearly

$$y=-t+\frac{s}{2}=-t+\frac{u}{2},\quad x=s\frac{\sqrt{3}}{2}=-u\frac{\sqrt{3}}{2}\qquad t=-y+\frac{x}{\sqrt{3}},\quad s=\frac{2x}{\sqrt{3}},\quad u=-\frac{2x}{\sqrt{3}}$$

and as in Chapter 7.1 the volume term becomes

$$\frac{V_{ijk}}{24}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\frac{5a^2}{18}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk}\qquad(38)$$

with leakage along the s axis taking the form

$$D_{ijk}\int_{S_{jk}}\frac{\partial\phi^-(w,z)}{\partial s}dS=\frac{2\sqrt{3}D_{ijk}D_{i+1jk}S_{kj}}{(D_{ijk}a+D_{i+1jk}a)}\left(\phi_{i+1jk}-\phi_{ijk}+\frac{a^2}{24}\left(\frac{\partial^2\phi^-_{i+\frac{1}{2}jk}}{\partial s^2}-\frac{\partial^2\phi^+_{i+\frac{1}{2}}}{\partial s^2}\right)+\ldots\right)\quad(40)$$

so that the difference equation may be written after repeating the procedure for the other two sides of the triangle

$$\begin{aligned}
&D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})+D_{i-\frac{1}{2}}(\phi_{i-1jk}-\phi_{ijk})+D_{ij\pm\frac{1}{2}k}(\phi_{ij\pm1k}-\phi_{ijk})\\
&+D_{ijk+\frac{1}{2}}(\phi_{ijk+1}-\phi_{ijk})+D_{ijk-\frac{1}{2}}(\phi_{ijk-1}-\phi_{ijk})+\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}V_{ijk}\phi_{ijk}=\\
&\quad-\frac{V_{ijk}}{24}\left(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r\right)_{ijk}\left(\frac{5a^2}{18}\left(\frac{\partial^2}{\partial s^2}+\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2}\right)+\gamma_k^2\frac{\partial^2}{\partial z^2}\right)\phi_{ijk}\\
&+\frac{1}{8}\left(\begin{array}{l}\frac{a^2}{3}\left(D_{i+\frac{1}{2}jk}\frac{\partial^2}{\partial s^2}(\phi^+_{i+\frac{1}{2}jk}-\phi^-_{i+\frac{1}{2}jk})-D_{i-\frac{1}{2}jk}\frac{\partial^2}{\partial u^2}(\phi^+_{i-\frac{1}{2}jk}-\phi^-_{i-\frac{1}{2}jk})+D_{ij\pm\frac{1}{2}k}\frac{\partial^2}{\partial t^2}(\phi^+_{ij\pm\frac{1}{2}k}-\phi^-_{ij\pm\frac{1}{2}k})\right)\\
+D_{ijk+\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma^2_{k+1}\phi^+_{ijk+\frac{1}{2}}-\gamma^2_k\phi^-_{ijk+\frac{1}{2}})-D_{ijk-\frac{1}{2}}\frac{\partial^2}{\partial z^2}(\gamma^2_k\phi^+_{ijk-\frac{1}{2}}-\gamma^2_{k-1}\phi^-_{ijk-\frac{1}{2}})\end{array}\right)\quad(42)
\end{aligned}$$

where $\pm$ is used to denote whether the triangle points in the positive or negative y direction in an obvious manner.

At boundaries say in the s direction

$$D_{ijk}\int_{S_{kj}}\frac{\partial}{\partial s}\phi^{-}(y,z)dS=\frac{S_{kj}D_{ijk}\lambda}{(D_{ijk}+\frac{a}{2\sqrt{3}}\lambda)}(-\phi_{ijk}+\frac{S_{kj}}{24}(a^2\frac{\partial^2}{\partial s^2}-a^2\frac{\partial^2}{\partial w^2}-\gamma_k^2\frac{\partial^2}{\partial z^2})\phi^{-}_{i+\frac{1}{2}jk}+\ldots \tag{40}$$

However because the coordinate axes in (s,t,u) are not orthogonal some care has to be exercised in finding the second differential, $\frac{\partial^2}{\partial w^2}$, in terms of known values. The appropriate expression is

$$\frac{\partial^2}{\partial w^2}=\frac{1}{3}\left(2(\frac{\partial^2}{\partial t^2}+\frac{\partial^2}{\partial u^2})-\frac{\partial^2}{\partial s^2}\right) \tag{41}$$

as obtained previously in Chapter 7.

By interchanging the variables continuous derivatives for the other sides can be obtained in terms of (s,t,u) in an obvious manner. In approximating to the (s,t,u) second derivatives values along the appropriate axes are taken. Thus for s with the triangle pointing upwards

$$\frac{\partial\phi^{-}_{i+\frac{1}{2}jk}}{\partial s}=\frac{D_{i+\frac{1}{2}jk}}{D_{ijk}}(\phi_{i+1j+1k}-\phi_{ijk}),\qquad \frac{\partial\phi^{+}_{i-\frac{1}{2}jk}}{\partial s}=\frac{D_{i-\frac{3}{2}jk}}{2D_{ijk}}(\phi_{i1jk}-\phi_{i-2j-1k})$$

with $D_{i-\frac{3}{2}j-1k}$ defined in the usual way between points ijk and $i-2\,j-1k$ as shown in Figure 5. Initially some approximations to second derivatives used nearest neighbors but none gave satisfactory results for test problems. It seems curious that the point along the axis through a vertex of the triangle should give a good second derivative approximation when it is more or less isolated in leakage terms from triangle (i,j,k). Hence

$$\begin{aligned}\frac{\partial^2\phi_{ijk}}{\partial x^2}&=\frac{1}{a\sqrt{3}D_{ijk}}(2D_{i+\frac{1}{2}jk}(\phi_{i+1j+1k}-\phi_{ijk})+D_{i-\frac{3}{2}jk}(\phi_{ijk}-\phi_{i-2j-1k}))\\&=\Delta_s^2\phi_{ijk}\end{aligned} \tag{56}$$

where $D_{i-\frac{3}{2}jk}$ is the averaged diffusion coefficient between the two triangles shown in Figure 5.

The difference equation then becomes

$$D_{i+\frac{1}{2}jk}(\phi_{i+1jk}-\phi_{ijk})+D_{i-\frac{1}{2}jk}(\phi_{i-1jk}-\phi_{ijk})+D_{ij\pm\frac{1}{2}k}(\phi_{ij\pm1k}-\phi_{ijk})$$
$$+D_{ijk+\frac{1}{2}}(\phi_{ijk+1}-\phi_{ijk})+D_{ijk-\frac{1}{2}}(\phi_{ijk-1}-\phi_{ijk})+(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}V_{ijk}\phi_{ijk}=$$
$$-\frac{V_{ijk}}{24}(\frac{\nu\sigma_f}{k}+\sigma_s-\sigma_r)_{ijk}(\frac{5a^2}{18}(\Delta_s^2+\Delta_t^2+\Delta_u^2)+\gamma_k^2\Delta_z^2)\phi_{ijk}$$
$$+\frac{1}{24}\begin{pmatrix} D_{i+\frac{1}{2}jk}a^2(\Delta_s^2(\phi_{i+1jk}-\phi_{ijk})-D_{ij\pm\frac{1}{2}k}a^2\Delta_t^2(\phi_{ijk}-\phi_{ij\pm1k}) \\ +D_{i-\frac{1}{2}jk}a^2\Delta_u^2(\phi_{i-1jk}-\phi_{ijk}) \\ +3D_{ijk+\frac{1}{2}}\Delta_z^2(\gamma_{k+1}^2\phi_{ijk+1}-\gamma_k^2\phi_{ijk})-3D_{ijk-\frac{1}{2}}\Delta_z^2(\gamma_k^2\phi_{ijk}-\gamma_{k-1}^2\phi_{ijk-1}) \end{pmatrix} \quad (57).$$

When the boundary term of relation (40) is needed then meshpoint values of derivatives as usual can be used to second order accuracy.

CHAPTER 11. STANDARD FINITE DIFFERENCE SOLUTION

A code to test the mesh correction procedure has been developed called PNCD which primarily solves the standard mesh centered diffusion equation, that is relation (21) with the right hand side set to zero. The well known algorithm has been used(3) of source iteration in which, beginning with a flux guess, the fission source is calculated and new fluxes evaluated by solving the difference equation below for the flux ϕ_{ijk}^{gn} for each energy group g in turn beginning at the highest, where the superscript n denotes iteration number

$$D^g_{i+\frac{1}{2}jk}(\phi^{gn+1}_{i+1jk}-\phi^{gn+1}_{ijk})+D^g_{i-\frac{1}{2}jk}(\phi^{gn+1}_{i-1jk}-\phi^{gn+1}_{ijk})+D^g_{ij+\frac{1}{2}k}(\phi^{gn+1}_{ij+1k}-\phi^{gn+1}_{ijk})+D^g_{ij-\frac{1}{2}k}(\phi^{gn+1}_{ij-1k}-\phi^{gn+1}_{ijk})$$
$$+D^g_{ijk+\frac{1}{2}}(\phi^{gn+1}_{ijk+1}-\phi^{gn+1}_{ijk})+D^g_{ijk-\frac{1}{2}}(\phi^{gn+1}_{ijk-1}-\phi^{gn+1}_{ijk})-(\sigma_r)_{gijk}V_{ijk}\phi^{gn+1}_{ijk}+\sum_{g'}(\sigma_s^d)_{g'gijk}V_{ijk}\phi^{g'n+1}_{ijk}= \quad (58)$$
$$-\sum_{g'}\left((\frac{\nu\sigma_f}{k^n})_{g'ijk}\chi_g+(\sigma_s^u)_{g'gijk}\right)V_{ijk}\phi^{g'n}_{ijk}$$

with σ_s^d and σ_s^u are the down and upscatter parts of the scatter matrix, the reason for the separation being the group by group nature of the solution process. χ_g is the fission spectrum and the $g'g$ subscript implies

the scatter from group g' to g. Equation (58) is the standard approximation from expression (21) in group notation.

The eigenvalue k^n is then straightforwardly updated by a source balancing process as follows,

$$k^{n+1} = \left(\frac{\sum_{ijk} \left(\sum_{g} (\nu\sigma_f)_{gijk} \phi_{ijk}^{gn+1} \right) \left(\sum_{g} (\nu\sigma_f)_{gijk} \phi_{ijk}^{gn+1} \right)}{\sum_{ijk} \left(\sum_{g} (\nu\sigma_f)_{gijk} \phi_{ijk}^{gn+1} \right) \left(\sum_{g} (\nu\sigma_f)_{gijk} \phi_{ijk}^{gn} \right)} \right) k^n \tag{59}$$

where if the new source is too high k^{n+1} increases and decreases otherwise. The iterative process continues until eigenvalue and flux are changing by less than preset accuracy criteria between successive calculations

To solve equation (58) for ϕ_{ijk}^{gn+1} a process called line overrelaxation is invoked(3). This is a well established technique which uses the simple structure of the difference equations to converge the calculation rapidly. For completeness a brief description follows and for clarity the group identifier will be omitted since the process is the same for each group.

Let

$$\Phi_{ij}^{n+1} = \begin{pmatrix} \phi_{ij1}^{n+1} \\ \phi_{ij2}^{n+1} \\ \cdot \\ \cdot \\ \phi_{ijK}^{n+1} \end{pmatrix} \tag{60}$$

where there are K points in the z direction then equation (58) may be written in the matrix form

$$\mathbf{D}_{i-1j}\Phi_{i-1j}^{n+1} + \mathbf{D}_{ij-1}\Phi_{ij-1}^{n+1} + \mathbf{E}_{ij}\Phi_{ij}^{n+1} + \mathbf{D}_{i+1j}\Phi_{i+1j}^{n+1} + \mathbf{D}_{ij+1}\Phi_{ij+1}^{n+1} = \mathbf{S}_{ij}^{'n} \tag{61}$$

with, typically,

$$\mathbf{D}_{i-1j} = \begin{pmatrix} D_{i-\frac{1}{2}j1} & 0 & . & . & 0 \\ 0 & D_{i-\frac{1}{2}j2} & & & \\ . & & & & \\ . & & & & \\ 0 & & & & D_{i-\frac{1}{2}jK} \end{pmatrix} \tag{62}$$

and $\mathbf{E}_{jk}$ takes the tri-diagonal form

$$\mathbf{E}_{jk} = \begin{pmatrix} E_{ij1} & D_{ij\frac{1}{2}} & 0 & & 0 \\ D_{ij\frac{1}{2}} & E_{ij2} & D_{ij\frac{3}{2}} & & \\ & & & & \\ & & & & \\ 0 & & & D_{ijK-\frac{1}{2}} & E_{ijK} \end{pmatrix} \tag{63}$$

with

$$E_{ijk} = -D_{i-\frac{1}{2}jk} - D_{i+\frac{1}{2}jk} - D_{ij-\frac{1}{2}k} - D_{ij+\frac{1}{2}k} - D_{ijk-\frac{1}{2}} - D_{ijk+\frac{1}{2}} - (\sigma_r)_{ijk} \tag{64}.$$

The matrices defined in expressions 33 and 34 may be considered as blocks of global matrices $\mathbf{D}$ and $\mathbf{E}$ defined by

$$\mathbf{D} = \left(\mathbf{D}_{ij}\right) \tag{65}$$

and

$$\mathbf{E} = \left(\mathbf{E}_{ij}\right) \tag{66}.$$

$$\mathbf{S}'^{n}_{jk} = -\sum_{g'} \left(\left(\frac{\nu\sigma_f}{k^n}\right)_{g'ijk} \phi^{g'n}_{ijk} \chi_g + (\sigma^{u}_{s})_{g'gijk} \phi^{g'n}_{ijk} + (\sigma^{d}_{s})_{g'gijk} \phi^{g'n+1}_{ijk} \right) \tag{67}$$

where downscatter terms use the latest flux values as the calculation begins at $g=1$, the highest energy, this being a natural way to proceed as downscatter dominates the scattering process.

As already mentioned equation (61) is solved iteratively by line overrelaxation so, for convenience, denote the number of inner iterations by m and rewrite the relation in the form

$$\mathbf{D}_{i-1j}(\Phi^{n+1}_{i-1j})^{m+1}+\mathbf{D}_{ij-1}(\Phi^{n+1}_{ij-1})^{m+1}+\mathbf{E}_{ij}(\Phi^{n+1}_{ij})^{m+1}+\mathbf{D}_{i+1j}(\Phi^{n+1}_{i+1j})^{m}+\mathbf{D}_{ij+1}(\Phi^{n+1}_{ij+1})^{m}=\mathbf{S}^{n}_{ij} \quad (68)$$

where latest values have been used whenever possible and the solution begins at $\mathbf{i}=\mathbf{j}=1$ then proceeds along the $\mathbf{i}$ index first followed by $\mathbf{j}$. The iteration scheme is known as Gauss-Seidel and clearly

$$\Phi^{n}_{ij}=(\Phi^{n+1}_{ij})^{0} \quad (69).$$

A rather less efficient method uses values from the previous m iteration for all $\mathbf{D}_{ij}$ that is

$$\mathbf{E}(\Phi^{n+1})^{m+1}=-\mathbf{D}(\Phi^{n+1})^{m}+\mathbf{S}^{n} \quad (70)$$

in an obvious notation and is known as the Jacobi solution.

The technique of overrelaxation involves using a factor ω and calculating

$$(\Phi^{n+1}_{ij})^{m+1*}=\omega(\Phi^{n+1}_{ij})^{m+1}+(1-\omega)(\Phi^{n+1}_{ij})^{m} \quad (71)$$

so that the inner iteration scheme becomes

$$\mathbf{D}_{i-1j}(\Phi^{n+1}_{i-1j})^{m+1*}+\mathbf{D}_{ij-1}(\Phi^{n+1}_{ij-1})^{m+1*}+\mathbf{E}_{ij}\frac{\left((\Phi^{n+1}_{ij})^{m+1*}+(\omega-1)(\Phi^{n+1}_{ij})^{m}\right)}{\omega}+\mathbf{D}_{i+1j}(\Phi^{n+1}_{i+1j})^{m}+\mathbf{D}_{ij+1}(\Phi^{n+1}_{ij+1})^{m}=\mathbf{S}^{n}_{ij} \quad (72).$$

Denoting the m'th iteration error by $\varepsilon^{m+1}_{ij}=(\Phi^{n+1}_{ij})^{m+1*}-(\Phi^{n+1}_{ij})^{m*}$ then by subtraction of iterations $m+1$ and m

$$\mathbf{D}_{i-1j}\varepsilon^{m+1}_{i-1j}+\mathbf{D}_{ij-1}\varepsilon^{m+1}_{ij-1}+\mathbf{E}_{ij}\frac{\varepsilon^{m+1}_{ij}}{\omega}=-\mathbf{E}_{ij}\frac{(\omega-1)\varepsilon^{m}_{ij}}{\omega}-\mathbf{D}_{i+1j}\varepsilon^{m}_{i+1j}-\mathbf{D}_{ij+1}\varepsilon^{m}_{ij+1} \quad (73)$$

and if X_{ij} is an eigenvector element of the system having K components as in definition (60) for the flux, with eigenvalue λ then

$$\lambda(\mathbf{D}_{i-1j}X_{i-1j}+\mathbf{D}_{ij-1}X_{ij-1}+\mathbf{E}_{ij}\frac{X_{ij}}{\omega})=-\mathbf{E}_{ij}\frac{(\omega-1)X_{ij}}{\omega}-\mathbf{D}_{i+1j}X_{i+1j}-\mathbf{D}_{ij+1}X_{ij+1} \quad (74).$$

Finally by modifying the eigenvector to $X_{ij} = \lambda^{\frac{i+j}{2}} X'_{ij}$ there results

$$(\mathbf{D}_{i-1j}X'_{i-1j} + \mathbf{D}_{ij-1}X'_{ij-1})\lambda^{\frac{i+j+1}{2}} + \mathbf{E}_{ij}\frac{X'_{ij}\lambda^{\frac{i+j}{2}+1}}{\omega} = -\mathbf{E}_{ij}\frac{(\omega-1)X'_{ij}\lambda^{\frac{i+j}{2}}}{\omega} - (\mathbf{D}_{i+1j}X'_{i+1j} - \mathbf{D}_{ij+1}X'_{ij+1})\lambda^{\frac{i+j+1}{2}} \tag{75}$$

or in terms of the global matrices defined by relations (65 and 66)

$$(\frac{\sqrt{\lambda}}{\omega} + \frac{(\omega-1)}{\omega\sqrt{\lambda}})X' = -\mathbf{E}^{-1}\mathbf{D}X' \tag{76}.$$

Now if η is an eigenvalue of $-\mathbf{E}^{-1}\mathbf{D}$ which from expression (70) is the Jacobi matrix, then by definitions (62) and (63) the matrix is positive with zero diagonal and row elements sum to unity at most so eigenvalues will be of modulus unity or less. The principle is now to select a value of ω which minimizes λ the error eigenvalue of the iterative process specified by equation (72) thus reducing the error more rapidly as can be seen from relations (73) and (74).

If $\omega = 1$ then $\sqrt{\lambda} = \eta$. From relation (71) this is the standard Gauss-Seidel and clearly shows an advantage over the Jacobi solution. Now ω is chosen to minimize λ for η_{max} the highest Jacobi eigenvalue.

Since, using equation (76)

$$\eta_{max} = \left(\frac{\sqrt{\lambda}}{\omega} + \frac{(\omega-1)}{\omega\sqrt{\lambda}}\right) \tag{77}$$

then

$$\lambda - \omega\eta_{max}\sqrt{\lambda} + (\omega-1) = 0 \tag{78}$$

and

$$\sqrt{\lambda} = \frac{\omega\eta_{max} \pm \sqrt{\omega^2\eta_{max}^2 - 4(\omega-1)}}{2} \tag{79}.$$

From equation (79) on taking the positive square root since this gives Gauss Seidel for $\omega = 1$ and differentiating

$$\frac{1}{\sqrt{\lambda}}\frac{d\lambda}{d\omega}=\frac{1}{2}\left(\eta_{\max}+\frac{\omega\eta_{\max}^2-2}{\sqrt{\eta_{\max}^2\omega^2-4(\omega-1)}}\right) \tag{80}$$

which is negative for $\omega=0$ so that initially λ decreases as ω increases. If the derivative of the second term on the right is negative then it becomes more negative and reduces λ more rapidly. Now

$$\frac{d}{d\omega}\left(\frac{\omega\eta_{\max}^2-2}{\sqrt{\eta_{\max}^2\omega^2-4(\omega-1)}}\right)=\left(\frac{-4(1-\eta_{\max}^2)}{(\eta_{\max}^2\omega^2-4(\omega-1))^{\frac{3}{2}}}\right) \tag{81}$$

which is negative as required since $\eta_{\max}\leq 1$. However eventually with increasing ω the gradient in equation (80) becomes infinite, then has an imaginary component and increases with ω so the optimum value of ω to give a minimum λ is found when the square root is zero, that is

$$\omega=\frac{2}{1+\sqrt{1-\eta_{\max}^2}} \tag{82}$$

Other error eigenvalues, λ_i say take the form

$$\sqrt{\lambda_i}=\frac{\eta_i+\sqrt{\eta_i^2-\eta_{\max}^2}}{1+\sqrt{1-\eta_{\max}^2}} \tag{83}$$

thus having a small imaginary part but being reduced in modulus.

The ω factors are found for each energy group by beginning with a constant vector then carrying out the iterative process of equation (72). The vector converges to the highest error eigenvector for the Gauss-Seidel scheme and with sufficient accuracy say at iteration M then with $\mathbf{e}_m$ the vector with components ε_{ij}^m

$$\eta_{\max}^2=\left(\frac{\mathbf{e}_M.\mathbf{e}_M}{\mathbf{e}_{M-1}.\mathbf{e}_{M-1}}\right)^{\frac{1}{2}} \tag{84}$$

on recalling that this iterative process gives the eigenvalues of the Jacobi procedure squared. Usually when changes between successive estimates of $\eta_{\max}^n$ are less than 0.0005 convergence is considered sufficiently accurate.

The tri-diagonal matrix defined by formulation (63) of course implies a solution of simultaneous equations

$$\mathbf{E}_{ij}(\Phi_{ij}^{n+1})^{m+1}=\mathbf{S}_{ij}^{*} \tag{85}$$

where the right hand side is the sum of all other terms in expression (68).

This may be expanded in a simplified notation based on matrix (63)

$$\begin{pmatrix} E_1 & D_1 & 0 & 0 & 0 & . & 0 \\ C_2 & E_2 & D_2 & 0 & 0 & . & 0 \\ 0 & C_3 & E_3 & 0 & 0 & . & 0 \\ 0 & & & & & & \\ . & & & & & & \\ 0 & 0 & 0 & & & C_K & E_K \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \\ \\ \phi_K \end{pmatrix} = S \qquad (86)$$

Forward substitution and backward differencing are used to solve equation (86). In this the matrix $\mathbf{E}_{ij}$ is expressed as the product of two matrices, lower and upper triangular respectively, so that the unknown vector Φ_{ij} can be found in a two stage process without a lengthy elimination process.

Consider

$$\begin{pmatrix} E_1' & 0 & 0 & 0 & 0 & . & 0 & 0 \\ C_2 & E_2' & 0 & 0 & 0 & . & 0 & 0 \\ 0 & C_3 & E_3' & 0 & 0 & . & 0 & 0 \\ 0 & & & & & & & \\ . & & & & & & & \\ 0 & 0 & 0 & & & & C_K & E_K' \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_3 \\ \\ \\ x_K \end{pmatrix} = S \qquad (87)$$

and

$$\begin{pmatrix} 1 & D_1' & 0 & 0 & 0 & . & 0 \\ 0 & 1 & D_2' & 0 & 0 & . & 0 \\ 0 & 0 & 1 & D_3' & 0 & . & 0 \\ 0 & & & & & & \\ . & & & & & & \\ 0 & 0 & 0 & & & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \\ \\ \phi_K \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \\ \\ x_K \end{pmatrix} \qquad (88)$$

The first system can be solved easily beginning at x_1 while the second uses substitution beginning at ϕ_K. This will be equivalent to the original system (86) if the product of the matrices in relations (87) and (88) gives the same elements. The product is

$$\begin{pmatrix} E_1' & E_1'D_1' & 0 & 0 & 0 & . & . & 0 & 0 \\ C_2' & E_2'+C_2'D_1' & E_2'D_2' & 0 & 0 & & & & \\ 0 & C_3' & E_3'+C_3'D_2' & E_3'D_3' & 0 & & & & \\ . & & & & & & & & \\ . & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & . & C_K' & E_K'+C_K'D_{K-1}' & \end{pmatrix} \tag{89}$$

so that in general if

$$C_i' = C_i, \quad E_i' = E_i - C_i'D_{i-1}', \quad D_i' = \frac{D_i}{E_i'} \tag{90}$$

then equation (86) will be solved by the efficient double sweep method.

When setting up the coefficients these adjustments are carried out and the revised values stored for use in the solution process.

The foregoing provides an outline of the method of solving the diffusion equation and data input is specified in Appendix 1 with an example in Appendix 2.

To incorporate the mesh correction term is straightforward after the evaluation of the fission source at iteration n in equation (61) the extra terms of expression (55) are added in using the current flux values. First the standard solution is converged then the calculation continues with the correction term included until convergence is obtained a second time.

Triangular-Z geometry uses the same difference equations except that there are only three leakage terms in the xy plane as can be seen in equation (42) so that the same solution technique can be used.

CHAPTER 12 TEST PROBLEMS AND RESULTS

Three fast reactor problems have been used from the 3-dimensional neutron benchmark exercise carried out in 1990(4). Two are in XYZ geometry and the third in Triangular-Z

121 Small FBR Problem

The problem is illustrated in Figure 6 and cross section data given in Appendix 3. As shown two cases have been considered control rod out and half in. For each case solutions were obtained for the mesh specified and then halved. Because the dependence of eigenvalue on mesh squared was proved in Part 1 the variation of k with mesh may be written with h

representing mesh size and a a constant

$$k_1 = k_0 + ah^2 \tag{91}$$

and for the mesh halved

$$k_{\frac{1}{2}} = k_0 + \frac{ah^2}{4} \tag{92}$$

where k_0 is the zero mesh value so that on eliminating a

$$k_0 = \frac{4k_{\frac{1}{2}} - k_1}{3} \tag{93}$$

Tables 1 and 2 show results obtained for the rod out and rod half in cases respectively and the good values given by the mesh correction term.

Table 1 k_{eff} Values for the Small FBR Rod Out Case

	full mesh	half mesh	zero mesh
no mesh correction	0.970307	0.969204	0.968836
mesh corrected	0.968897	0.968806	

Table 2 k_{eff} Values for the Small FBR Rod Half In Case

	full mesh	half mesh	zero mesh
no mesh correction	0.957161	0.954891	0.954134
mesh corrected	0.954712	0.954140	

12.2 Large FBR Problem

Details of the problem are shown in Figure 7 showing the very heterogeneous structure with internal breeder

Table 3 k_{eff} Values for the Large FBR Rod Out Case

	full mesh	half mesh	zero mesh
no mesh correction	0.970683	0.966615	0.965259
mesh corrected	0.966485	0.965135	

Here the mesh effect is quite large and the corrected value being 0.000124 lower than the extrapolate suggests that the full mesh is too coarse thus containing significant higher order error terms so that the mesh squared dependence rule is not strictly applicable.

12.3 Hexagonal FBR Problem

The problem is shown in Figure 8 and control rod configuration in Figure 9 with the cross section data given in Appendix 4. As can be seen from the Figures the reactor is extremely heterogeneous and this is reflected in the k_{eff} values shown in Tables 4,5 and 6 for the three control rod positions. The correction formula on the halved mesh performs well even on the rod inserted case which has a mesh effect on k_{eff} of around 2.0% for the full mesh model.

As in the previous problem it seems that the full mesh is somewhat outside the range of applicability for the mesh squared rule. However the problem is extremely heterogeneous and maybe a finer mesh is required to obtain a very accurate correction.

Table 4 k_{eff} Values for the Hexagonal FBR Rod Out Case

	full mesh	half mesh	zero mesh
no mesh correction	1.078300	1.072279	1.070272
mesh corrected	1.071235	1.069542	

Table 5 k_{eff} Values for the Hexagonal FBR Rod Half In Case

	full mesh	half mesh	zero mesh
no mesh correction	0.970527	0.959776	0.956190
mesh corrected	0.959521	0.955590	

Table 6 k_{eff} Values for the Hexagonal FBR Rod Fully In Case

	full mesh	half mesh	zero mesh
no mesh correction	0.872014	0.856737	0.851645
mesh corrected	0.857078	0.850902	

CHAPTER 13 DISCUSSION

Formulas (31), (47) and (51) give a relationship between mesh and eigenvalue depending only on the regular and adjoint fluxes for rectangular, triangular-Z and hexagonal-Z geometries respectively. Such expressions are interesting in that they quantify the mesh squared dependency of the eigenvalue k and give some idea of the relative error between the various geometries. However such formulas are not likely to have a practical application as the tendency is to run two cases of a problem, one with mesh halved, then extrapolate to the zero mesh value assuming the mesh squared dependence and use the fluxes from the finer mesh case for editing

As mentioned in the previous paragraph some indication of relative mesh error can be deduced for the various geometries. To make such a comparison the triangular expression should be converted to mesh interval rather than length of the triangle side. From Figure 2 it can be seen that

$$\text{mesh interval}=\frac{a}{\sqrt{3}}$$

so the constant multiplying the (s,t,u) term in formula (19) becomes $\frac{17}{24}$ and hence rectangular and triangular give roughly the same error for the same mesh. However, usually the hexagonal mesh will be three times the triangular so in terms of the triangular equivalent

the constant in expression (20) is $\frac{131}{17}$ suggesting an error higher by a factor of around 8 in the plane, although of course there are 6 times fewer mesh points. Since it is not possible to reduce a hexagonal mesh i.e. hexagons cannot be nested the foregoing indicates that hexagonal geometry finite difference results should be treated with some caution.

The mesh effect appears as two contributions, a central term always positive on assuming second order derivatives and the cross section in brackets have the same sign whereas the boundary leakage looks like a negative effect by similar reasoning. Thus it may be possible to obtain a zero mesh effect but of course fluxes would be seriously in error The foregoing reasoning is certainly true in one group since second order derivatives have the same sign but is somewhat conjectural for the multigroup case and based on experience(2) that large problems with small leakage invariably give an eigenvalue higher than the zero mesh value.

The point may be clarified by assuming a solution of the form $e^{i\kappa.r}$ for the flux and substituting in Equation (3) so that κ must be a solution of the determinant equation

$$\left\| D\kappa^2 - \left(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r\right) \right\| = 0.$$

Thus the flux takes the form below, for group g,

$$\sum_j \iiint_{\kappa_j^2=\kappa_x^2+\kappa_y^2+\kappa_z^2} a_{gj}(\kappa_x, \kappa_y, \kappa_z) e^{i\kappa_j.r} d\kappa_x d\kappa_y d\kappa_z$$

where $a_{gj}(\kappa_x, \kappa_y, \kappa_z)$ depends on continuity conditions at the mesh region boundary. From this expression it does not seem clear that second order derivatives for a group will have the opposite sign to

$$\left(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r\right)$$

or that the sum over groups in formula (31) will have the required property.

Reference 2 gives a relation (expression 20 page 406) in which derivatives are 4th order and the leading factor is $\frac{1}{24}$ rather than $\frac{1}{6}$ as derived in this paper. However the expression uses a buckling formulation which is related to $3\sigma_t\left(\frac{\nu\sigma_f}{k} + \sigma_s - \sigma_r\right)$ where σ_t is the total cross section and only leakage effects appear to be

considered. Thus if say the leakage contribution in formula (24) has both numerator and denominator multiplied by 3 to give a buckling equivalent expression, the factor $\frac{1}{24}$ appears giving agreement with the value in reference 2.

CHAPTER 14 CONCLUSIONS

Formulæ (31), (47) and (51) give compact expressions specifying the second order mesh dependency of k for rectangular, Triangular-Z and Hexagonal-Z geometries respectively when using mesh centered finite differences. It is conjectured for the multigroup case that there is a positive volume term and a negative boundary contribution. This seems to be borne out by experience(2) where for small leakage problems the eigenvalue is always overestimated.

A method has been proposed for reducing mesh errors by including second order effects as a source term. The approach has been shown to work well on three test problems including both XYZ and Triangular-Z geometries. For halved mesh cases all eigenvalues with the correction term included give differences between corrected and extrapolated values ranging from 0.01% in XYZ geometry to 0.07% for the difficult hexagonal FBR problem. It is likely that the mesh squared rule does not strictly hold to this accuracy for meshes of around 7.5cm and the corrected values may be nearer the zero mesh eigenvalues than the extrapolate. Although as mentioned in Chapter 12.3 the problem is extremely heterogeneous and may need a finer mesh to give a more accurate correction value To test these views requires the use of further mesh division which makes the problem too large for the computer facilities available at present.

The approach for testing purposes used a specially written diffusion theory code, a FORTRAN listing being given in Appendix 5, but the method is quite general and could be incorporated in any mesh centered finite difference code.

A further benefit arises because the mesh squared dependency of k_{eff} for sufficiently small meshes has been demonstrated theoretically in Part 1 of this report so no reference to other calculational methods is needed to validate the technique as zero mesh values can be derived to good accuracy by extrapolation.

The method will enable flux transport effects to be calculated accurately by comparison with Monte Carlo. Thus for design work where diffusion theory is still widely used it will be possible to obtain close to zero mesh values for reaction rates so enabling

discrepancies due to transport theory only to be evaluated.

Future work will concentrate on applying the method in the design area and assessing its value for general use.

REFERENCES

1Kato Y, Takeda T & Takeda S.
A Coarse Mesh Correction of the Finite Difference Method for Neutron Diffusion Calculations.
Nuc. Sci. & Eng. **61** 127-141 (1976)

2 Kato Y
Spatial Mesh Effects in Three Dimensional x-y-z Neutron Diffusion Calculations in Fast Breeders Reactors
Nuc. Sci. & Eng. **104** 402-411 (1990)

3. Fletcher J. K.
MARC/PN. A Computer Code to Solve the Multigroup Neutron Transport Equation.
RTS-002(88) AEA Technology (1988)

4. Takeda T,Ikeda H
3-D Neutron Transport Benchmarks.
NEACRP-L-330 (1991)

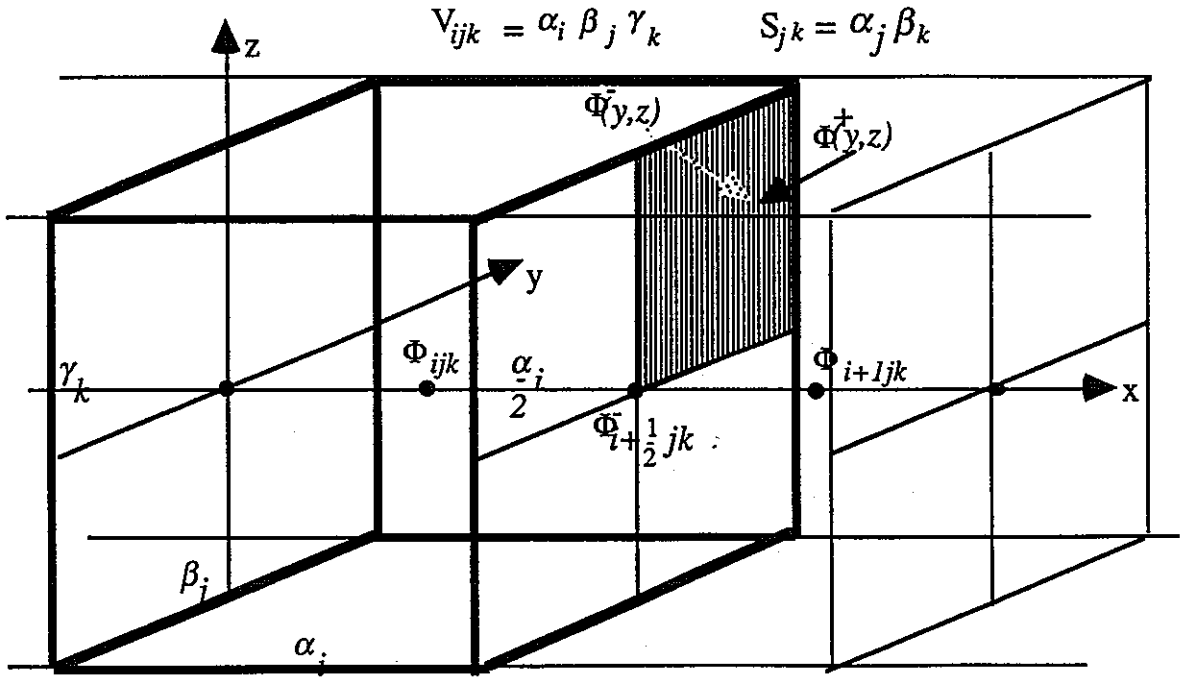


Figure 1 . Illustration of Finite Difference Mesh in XYZ Geometry

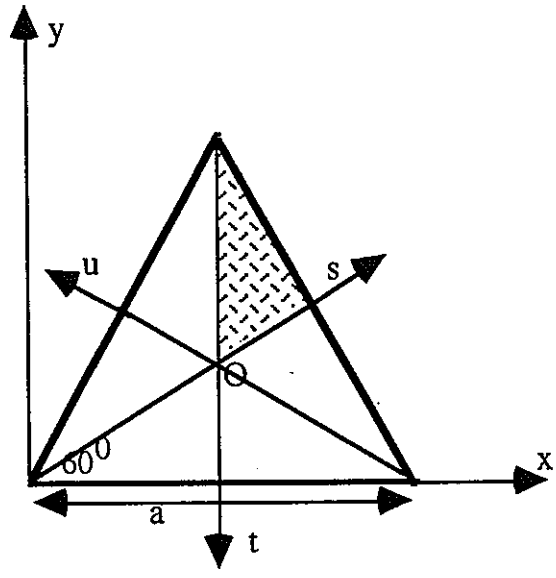


Figure 2 Triangular-Z Coordinates in the XY Plane

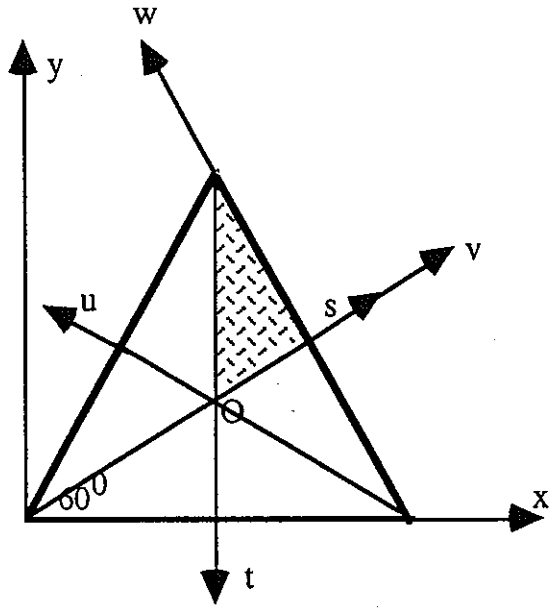


Figure 3. Local Coordinates for Triangular-Z Geometry

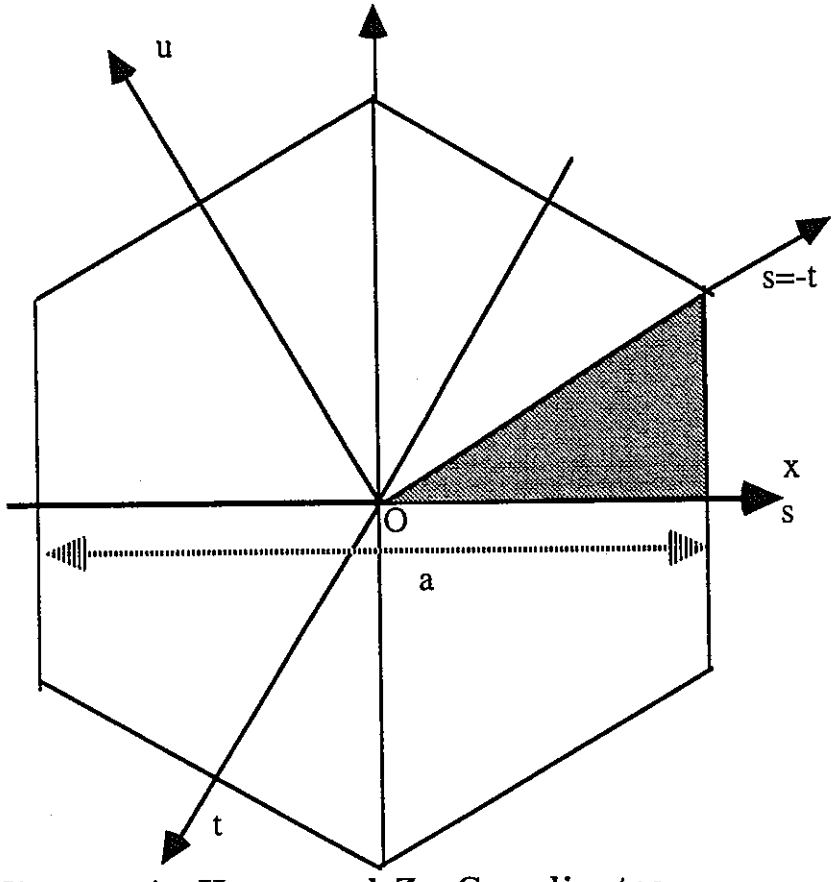


Figure 4 Hexagonal-Z Coordinates

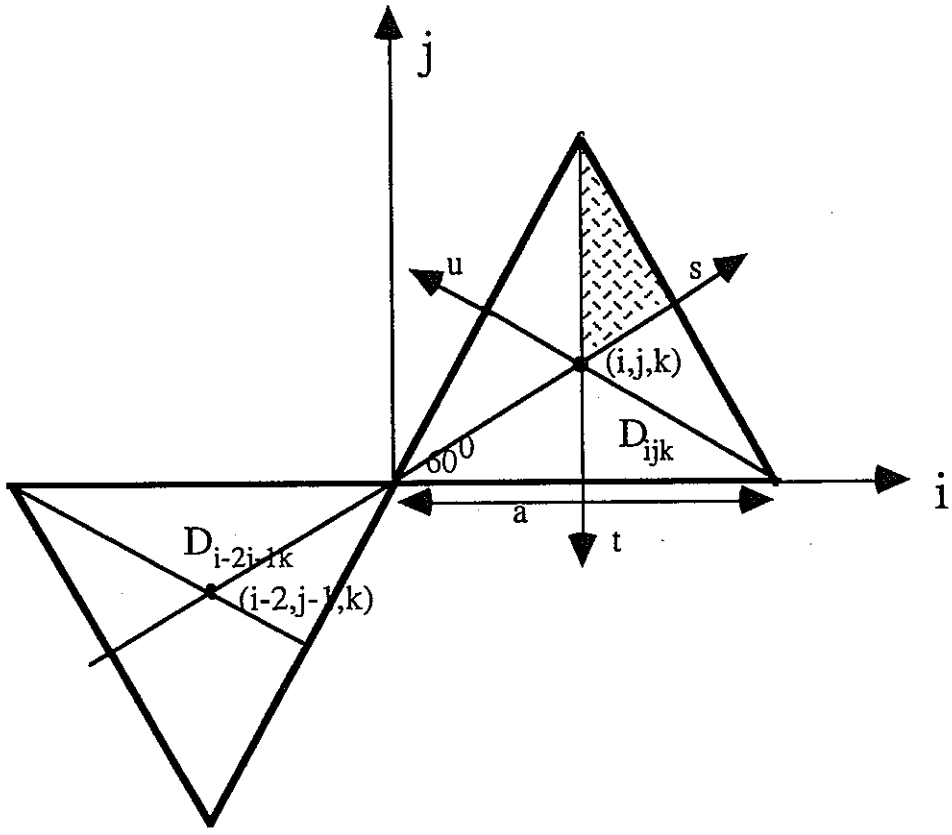


Figure 5 Location of Points to be Used in Evaluating $\frac{\partial^2}{\partial s^2}$

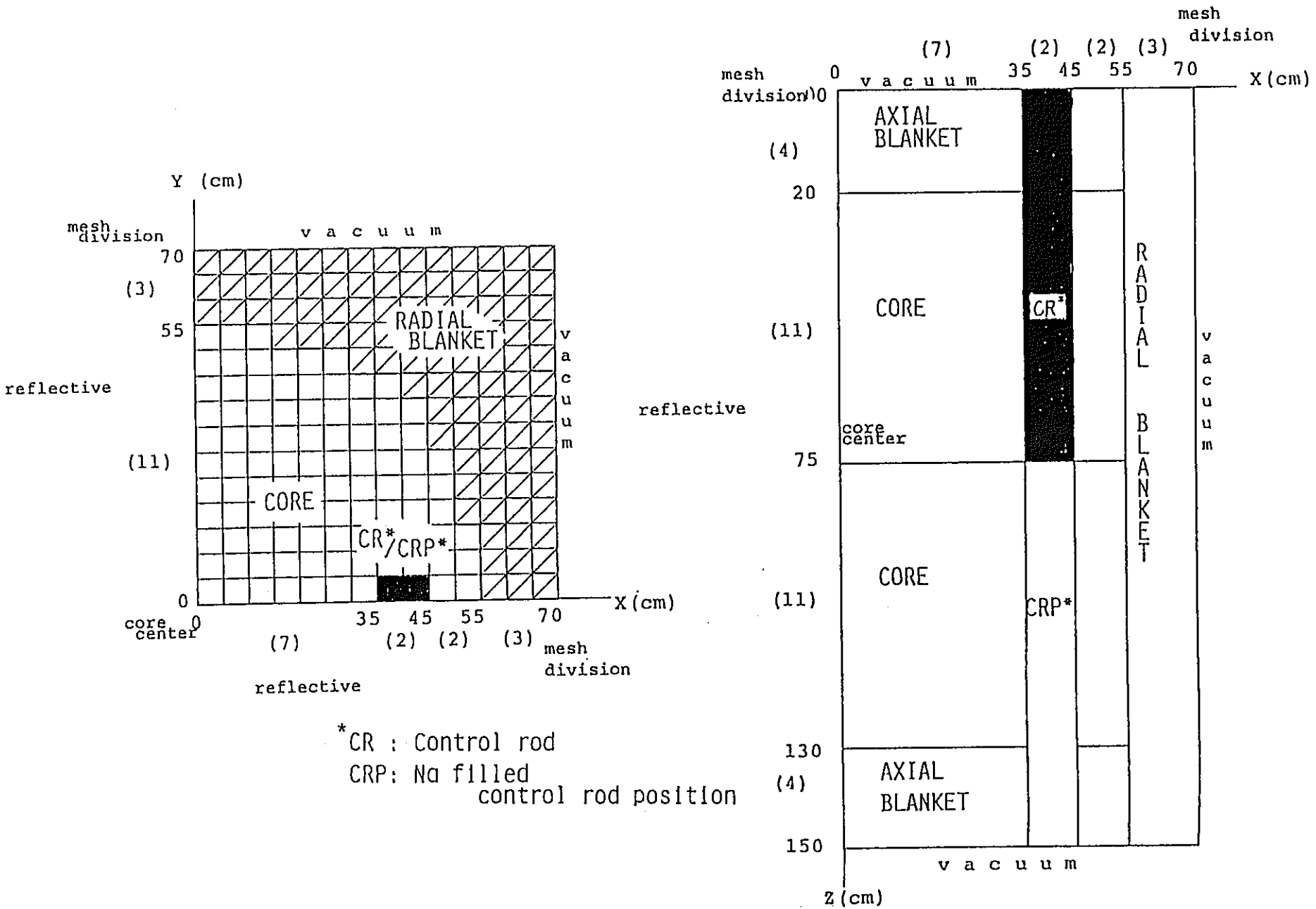
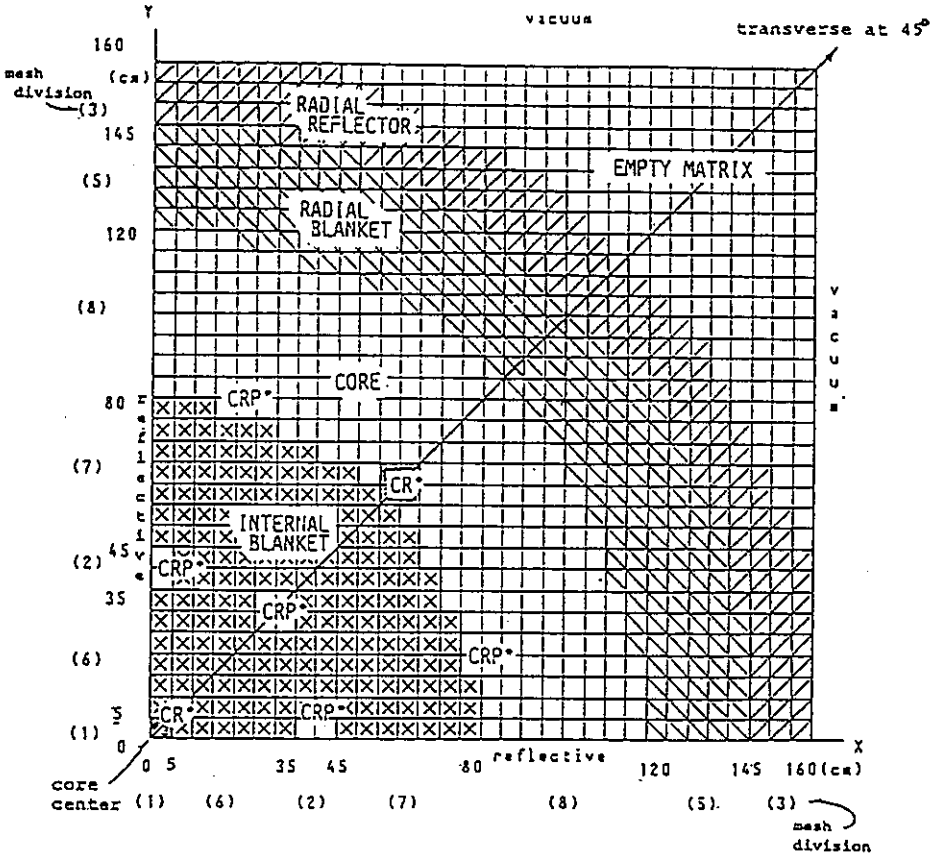


FIGURE 6 TEST PROBLEM 1 SMALL FBR



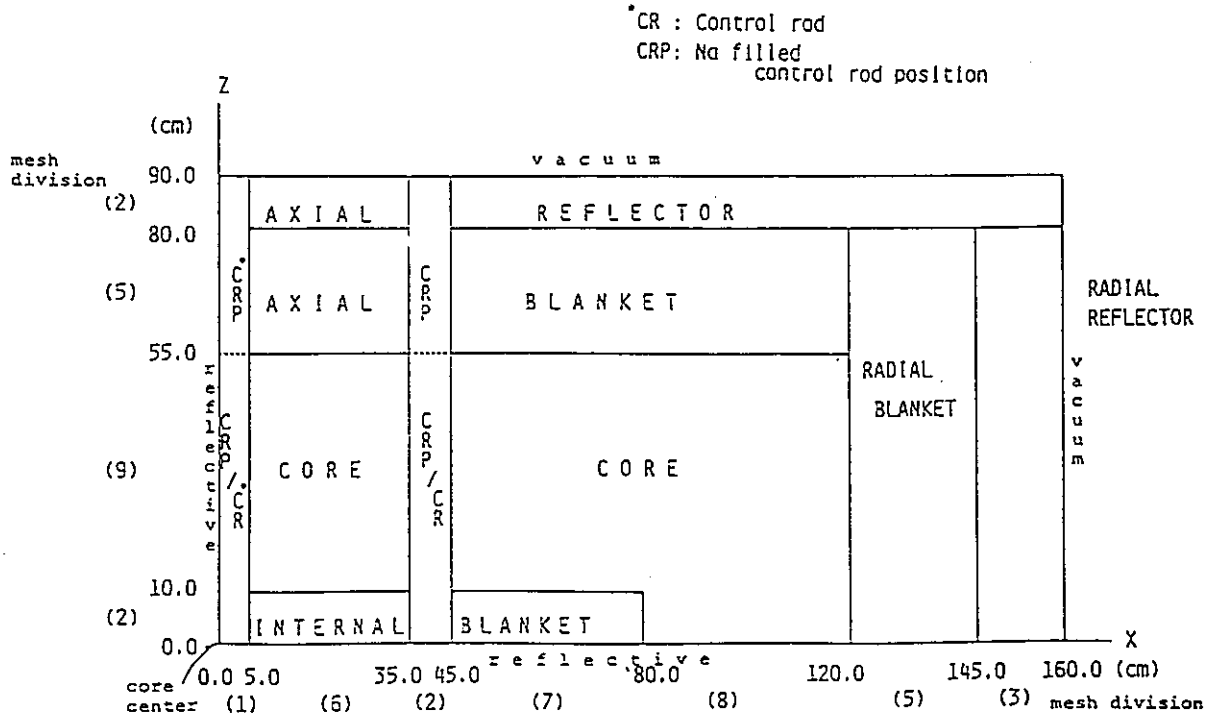
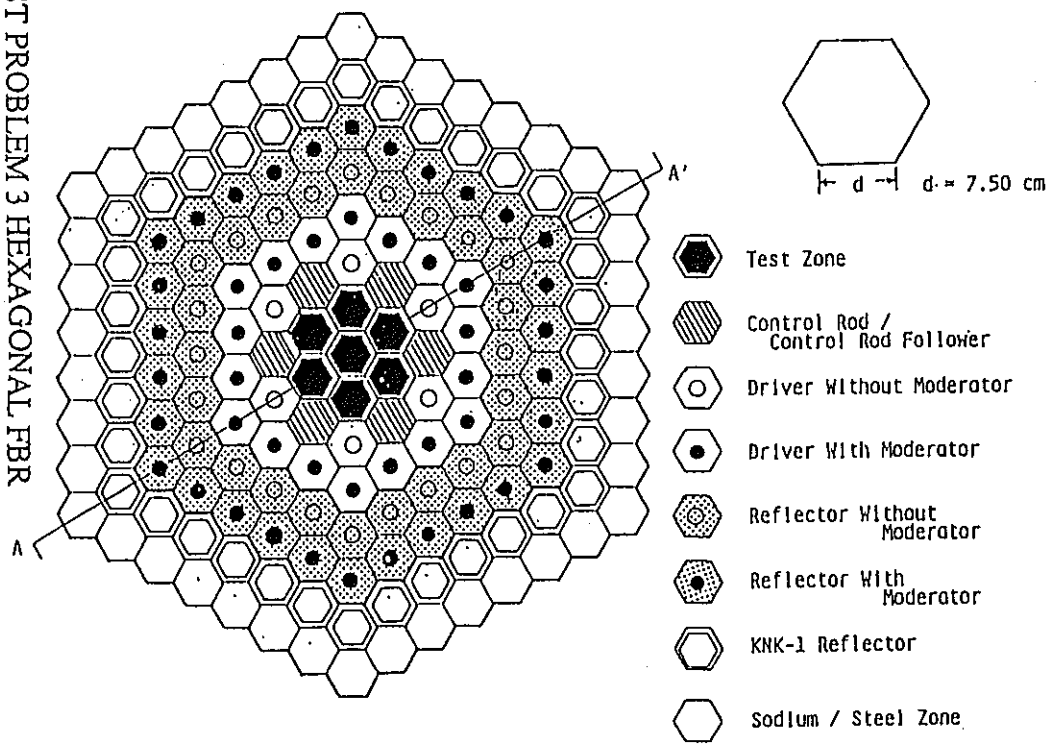


FIGURE 7 TEST PROBLEM 2 LARGE FBR



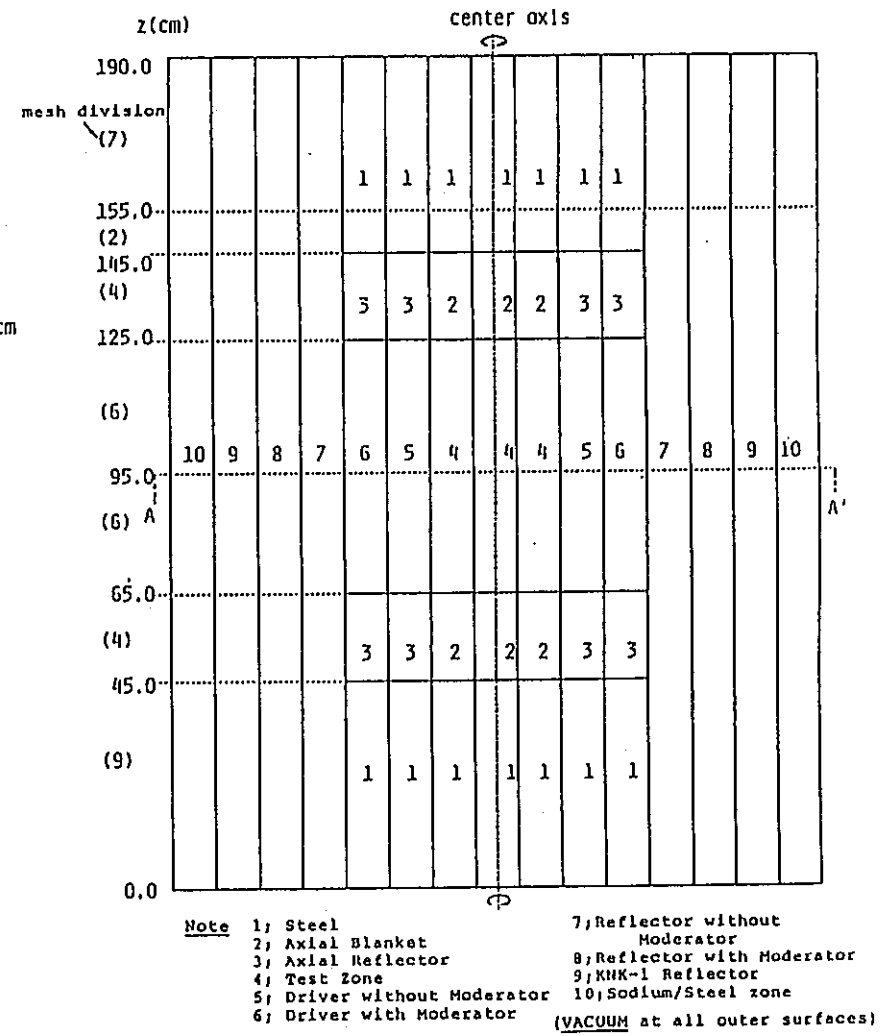


FIGURE 8 TEST PROBLEM 3 HEXAGONAL FBR

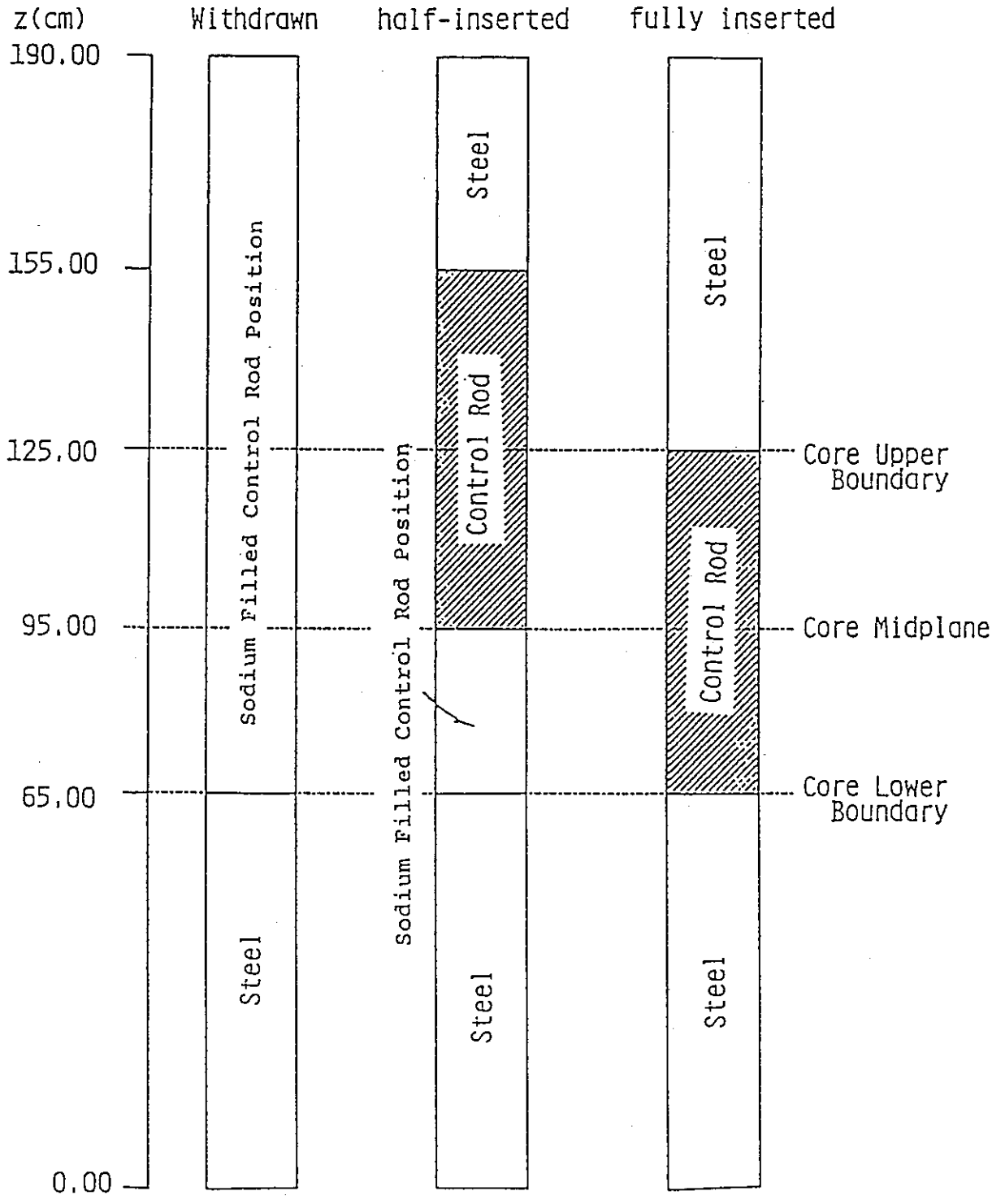


FIGURE 9 CONTROL ROD POSITIONS FOR PROBLEM 3

APPENDIX 1 DATA INPUT TO THE PNCD CODE

Input is in free format and code words will be given in bold type. Two means of reducing data length are available

n*a will input a n times

m(...) inputs whatever numeric data is in the brackets m times.

***TITLE**
As many title cards as desired may be submitted but all must begin with an asterisk

GC ng nds nup

where ng= number of groups
nds=number of downscatters
nup= number of upscatters and must be zero at present

MESHF n Reduce the submitted meshes by factor n

MESHQ Incorporate mesh correction term

XM $n_1 * x_1 \quad n_2 * x_2 \quad x_3 \quad . \quad . \quad .$
mesh in the x direction naturally single mesh values can be input the asterisk repeat is not obligatory
YM $y_1 \quad n_1 * y_2 \quad y_3 \quad . \quad . \quad .$
ZM $m_1 * z_1 \quad m_2 * z_2 \quad m_3 * z_3 \quad . \quad . \quad .$

similar mesh data for the y and z directions

BC $\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4 \quad \lambda_5 \quad \lambda_6$

Boundary conditions i.e. $\nabla\phi = -\lambda_i\phi$ in the order first y axis point then last followed by similar pairs for the x then z directions

ACCURACY $\alpha_1 \quad \alpha_2$

Convergence criteria, the first number allows convergence if the change between successive eigenvalues is less than it(default 0.00001) and the second tests maximum point fission source changes(default 0.0005).

I1

Signifies cross sections for material 1 will follow in the order transport for all groups, removal that is absorption plus scatter out and production. Next a full scatter band is give in the order 1 to 2, 1 to 3, 1 to 4..., 1 to nds, 2 to 3 and so on filled out with zeros so there are nds terms for each group giving ng*nds in total. Upscatter is not yet available so no values are included. Fission spectrum follows then capture and fission cross sections are optional for possible future editing capabilities.

I2 and so on.

The I numbers need not be consecutive or start at 1

C1 $n_1 * \mathbf{I}m_1 \quad \mathbf{I}m_2 \quad n_2 * \mathbf{I}m_3 \quad . \; . \; .$
C2 $n_4 * \mathbf{I}m_4 \quad n_5 * \mathbf{I}m_5 \quad n_6 * \mathbf{I}m_6 \quad . \; . \; .$
allocation of materials to points in the z direction, again the asterisk notation may help reduce data input. The **C**'s do not have to be consecutive and can begin at any number

CA $i_1 \quad i_2 \quad n_1 * i_3 \quad . \; .$

Allocation of **C**'s to the xy plane in the order x first. An irregular boundary may be specified by setting i's to zero. However the model must not be re-entrant

This completes the geometric specification. The data is straightforward and Appendix 2 gives an example.

SOLVE find the eigenvalue and fluxes

PROFILE $*l \quad m$ output fluxes along a given axis at a specified meshpoint in the perpendicular plane. Thus the example gives fluxes along the x axis at point l in the y mesh and m for z. Different plots can be obtained by putting the asterisk in other positions.

STOP ends the calculation

APPENDIX 2 EXAMPLE OF INPUT TO PNCD

```
*SMALL FBR
 MESHF 2 MESHQ
GC 4 2 0  XM 14*5  YM 14*5  ZM 30*5  BC 0  .5  0  .5  .5  .5
I1  1.14568E-1  2.05177E-1  3.29381E-1  3.89810E-1
    4.41354E-2  9.73474E-3  8.79433E-3  2.74496E-2
    2.06063E-2  6.10571E-3  6.91403E-3  2.60689E-2
    3.47967E-2  1.88282E-3  6.20863E-3  7.07208E-7  9.92975E-4  0  0  0
   0.583319    0.405450    0.011231    0.0
    7.45551E-3  3.52540E-3  7.80136E-3  2.74496E-2
I2  1.19648E-1  2.42195E-1  3.56476E-1  3.79433E-1
    5.05322E-2  1.15695E-2  8.06231E-3  1.58015E-2
    1.89496E-2  1.75265E-4  2.06978E-4  1.13451E-3
    4.04132E-2  2.68621E-3  9.57027E-3  1.99571E-7  1.27195E-3  0  0  0
   0.583319    0.405450    0.011231    0.0
    7.43283E-3  1.99906E-3  6.79036E-3  1.58015E-2
I3  1.16493E-1  2.20531E-1  3.44544E-1  3.88356E-1
    4.48883E-2  1.00853E-2  7.03838E-3  1.34694E-2
    1.31770E-2  1.26026E-4  1.52380E-4  7.87302E-4
    3.73170E-2  2.21707E-3  8.59855E-3  6.68299E-7  1.68530E-3  0  0  0
   0.583319    0.405450    0.011231    0.0
    5.35418E-3  1.48604E-3  5.35300E-3  1.34694E-2
I4  1.65613E-1  1.66866E-1  2.68633E-1  8.34911E-1
    4.99587E-2  6.04849E-3  3.62227E-3  4.36382E-3
  0.0       0.0       0.0       0.0
    4.84721E-2  8.46495E-4  5.64096E-3   6.57573E-7  2.41755E-3  0  0  0
  0.0       0.0       0.0       0.0
    6.39154E-4  4.06876E-4  1.20472E-3  4.36383E-3
I5  1.71748E-1  2.17826E-1  4.47761E-1  7.95199E-1
    4.83959E-2  6.76283E-3  4.71626E-3  5.70263E-3
  0.0       0.0       0.0       0.0
    4.62307E-2  1.13217E-3  6.27100E-3  1.03831E-6  2.77126E-3  0  0  0
  0.0       0.0       0.0       0.0
    1.13305E-1  4.90793E-4  1.94500E-3  5.70263E-3
I6  1.84333E-1  3.66121E-1  6.15527E-1  1.09486
    4.99599E-2  4.75382E-2  9.59362E-2  4.76591E-1
  0.0       0.0       0.0       0.0
    4.37775E-2  2.06054E-4  2.98432E-2  8.71100E-7  7.66209E-3  0  0  0
  0.0       0.0       0.0       0.0
    5.97638E-3  1.76941E-2  8.82741E-2  4.76591E-1
```

```
I7  6.58979E-2  1.09810E-1  1.86765E-1  2.09933E-1
    1.84572E-2  3.66790E-3  1.46178E-3  1.07518E-3
  0.0       0.0       0.0       0.0
    1.76894E-2  4.57012E-4  3.55466E-3  1.77599E-7  1.01280E-3  0  0  0
  0.0       0.0       0.0       0.0
    3.10744E-4  1.13062E-4  4.48988E-4  1.07518E-3
I8  1.36985E-2  1.69037E-2  3.12271E-2  6.29537E-2
    4.11854E-3  4.29715E-4  3.16628E-4  4.95515E-4
  0.0       0.0       0.0       0.0
    3.95552E-3  8.80428E-5  3.91394E-4  7.72254E-8  1.77293E-4  0  0  0
  0.0       0.0       0.0       0.0
    7.49800E-5  3.82435E-5  1.39335E-4  4.95515E-4
C1 4*3 22*1 4*3 C2 30*2 C3 15*6 15*7 C4 30*7
CA 7*1 2*3 2*1 3*2  2(11*1 3*2)  3(10*1 4*2)  2(9*1 5*2) 8*1 6*2
   6*1 8*2  3*1 11*2   3(14*2)
ALTER 7 1
SOLVE PROFILE * 1  16 PROFILE 8 1 *
STOP
```

APPENDIX 3 Cross Section Data for the XYZ Geometry Test

There are 4 energy groups and 2 down scatters with cross sections given in the order

1) transport
2) removal, that is absorption plus scatter out of the group
3) ν-fission
4) down scatter in the order i to i+1, i to i+2 filled to a full band
5) fission spectrum
6) capture

Core

1.14568E-1 2.05177E-1 3.29381E-1 3.89810E-1
4.41354E-2 9.73474E-3 8.79433E-3 2.74496E-2
2.06063E-2 6.10571E-3 6.91403E-3 2.60689E-2
3.47967E-2 1.88282E-3 6.20863E-3 7.07208E-7 9.92975E-4 0 0 0
0.583319 0.405450 0.011231 0.0
7.45551E-3 3.52540E-3 7.80136E-3 2.74496E-2

Radial Blanket

1.19648E-1 2.42195E-1 3.56476E-1 3.79433E-1
5.05322E-2 1.15695E-2 8.06231E-3 1.58015E-2
1.89496E-2 1.75265E-4 2.06978E-4 1.13451E-3
4.04132E-2 2.68621E-3 9.57027E-3 1.99571E-7 1.27195E-3 0 0 0
0.583319 0.405450 0.011231 0.0
7.43283E-3 1.99906E-3 6.79036E-3 1.58015E-2

Axial Blanket

1.16493E-1 2.20531E-1 3.44544E-1 3.88356E-1
4.48883E-2 1.00853E-2 7.03838E-3 1.34694E-2
1.31770E-2 1.26026E-4 1.52380E-4 7.87302E-4
3.73170E-2 2.21707E-3 8.59855E-3 6.68299E-7 1.68530E-3 0 0 0
0.583319 0.405450 0.011231 0.0
5.35418E-3 1.48604E-3 5.35300E-3 1.34694E-2

Axial Reflector

1.65613E-1 1.66866E-1 2.68633E-1 8.34911E-1
4.99587E-2 6.04849E-3 3.62227E-3 4.36382E-3
0.0 0.0 0.0 0.0
4.84721E-2 8.46495E-4 5.64096E-3 6.57573E-7 2.41755E-3 0 0 0
0.0 0.0 0.0 0.0

6.39154E-4 4.06876E-4 1.20472E-3 4.36383E-3

Radial Reflector

1.71748E-1 2.17826E-1 4.47761E-1 7.95199E-1
4.83959E-2 6.76283E-3 4.71626E-3 5.70263E-3
0.0 0.0 0.0 0.0
4.62307E-2 1.13217E-3 6.27100E-3 1.03831E-6 2.77126E-3 0 0 0
0.0 0.0 0.0 0.0
1.13305E-1 4.90793E-4 1.94500E-3 5.70263E-3

Control Rod

1.84333E-1 3.66121E-1 6.15527E-1 1.09486
4.99599E-2 4.75382E-2 9.59362E-2 4.76591E-1
0.0 0.0 0.0 0.0
4.37775E-2 2.06054E-4 2.98432E-2 8.71100E-7 7.66209E-3 0 0 0
0.0 0.0 0.0 0.0
5.97638E-3 1.76941E-2 8.82741E-2 4.76591E-1

Na filled Control Rod Position

6.58979E-2 1.09810E-1 1.86765E-1 2.09933E-1
1.84572E-2 3.66790E-3 1.46178E-3 1.07518E-3
0.0 0.0 0.0 0.0
1.76894E-2 4.57012E-4 3.55466E-3 1.77599E-7 1.01280E-3 0 0 0
0.0 0.0 0.0 0.0
3.10744E-4 1.13062E-4 4.48988E-4 1.07518E-3

Empty Matrix

1.36985E-2 1.69037E-2 3.12271E-2 6.29537E-2
4.11854E-3 4.29715E-4 3.16628E-4 4.95515E-4
0.0 0.0 0.0 0.0
3.95552E-3 8.80428E-5 3.91394E-4 7.72254E-8 1.77293E-4 0 0 0
0.0 0.0 0.0 0.0
7.49800E-5 3.82435E-5 1.39335E-4 4.95515E-4

APPENDIX 4 Cross Section Data for the Triangular-Z Case

There are 4 energy groups with 3 downscatters in the same cross section order as Appendix 3

Axial Blanket

1.40462E-1	2.25534E-1	3.27065E-1	3.41224E-1		
1.66570E-2	8.27417E-3	9.11702E-3	9.94260E-3		
2.96101E-3	6.56171E-5	1.14630E-4	4.93483E-4		
1.45483E-2	1.70276E-4	9.37083E-7	6.78885E-3	6.04793E-6	0
4.38782E-3	0	0	0	0	0
0.908564	0.087307	0.004129	0		
1.93752E-3	1.47927E-3	4.72919E-3	9.9426E-3		

Test Zone

1.24526E-1	2.01025E-1	2.86599E-1	3.68772E-1		
1.85260E-2	1.16553E-2	1.63924E-2	4.98120E-2		
1.79043E-2	1.59961E-2	2.40856E-2	7.33104E-2		
1.12738E-2	1.46192E-4	9.62178E-7	3.64847E-3	1.06888E-6	0
1.80479E-3	0	0	0	0	0
0.908564	0.087307	0.004129	0		
7.14117E-3	8.00576E-3	1.45876E-2	4.98120E-2		

Driver with Moderator

1.41428E-1	2.45394E-1	3.98255E-1	4.35990E-1		
2.70910E-2	3.33884E-2	4.61620E-2	6.51184E-2		
1.01663E-2	9.46359E-3	1.87325E-2	8.25335E-2		
2.09664E-2	1.39132E-3	6.10281E-5	2.67269E-2	1.08186E-3	0
3.29030E-2	0	0	0	0	0
0.908564	0.087307	0.004129	0		
4.67223E-3	5.57965E-3	1.32590E-2	6.51184E-2		

Driver without Moderator

1.40226E-1	2.28245E-1	3.25806E-1	4.18327E-1		
2.03389E-2	1.30318E-2	1.89212E-2	5.74211E-2		
1.59878E-2	1.64446E-2	2.71451E-2	8.45807E-2		
1.30790E-2	1.59938E-4	1.07166E-6	4.00117E-3	1.82716E-6	0
1.67341E-3	0	0	0	0	0
0.908564	0.087307	0.004129	0		
7.09892E-3	9.02877E-3	1.72478E-2	5.74211E-2		

Axial Reflector

1.32933E-1	1.78531E-1	2.83151E-1	4.62167E-1		
9.93754E-3	5.43570E-3	5.95728E-3	3.56939E-3		
0.0	0.0	0.0	0.0		
9.41231E-3	1.93791E-4	1.39307E-6	5.09881E-3	7.05075E-7	0
5.09601E-3	0	0	0	0	0
0.0	0.0	0.0	0.0		
3.30045E-4	3.36186E-4	8.61269E-4	3.56939E-3		

KNK-1 Reflector

1.51644E-1	1.42382E-1	1.65132E-1	8.04845E-1		
1.32174E-2	4.88035E-3	4.40958E-3	5.91325E-3		
0.0	0.0	0.0	0.0		
1.23901E-2	3.66930E-4	1.69036E-6	4.41927E-3	1.63280E-6	0
3.33075E-3	0	0	0	0	0
0.0	0.0	0.0	0.0		
4.58692E-4	4.59443E-4	1.07883E-3	5.91325E-3		

Reflector with Moderator

1.39164E-1	2.46993E-1	4.52425E-1	5.36256E-1		
3.32527E-2	6.21730E-2	7.93529E-2	2.41527E-2		
0.0	0.0	0.0	0.0		
2.96485E-2	3.06502E-3	1.41697E-4	5.91780E-2	2.69229E-3	0
7.81326E-2	0	0	0	0	0
0.0	0.0	0.0	0.0		
2.97516E-4	3.02674E-4	1.22034E-2	2.41527E-2		

Reflector without Moderator

1.59346E-1	2.16355E-1	3.48692E-1	6.24249E-1		
1.13773E-2	5.94461E-3	6.60689E-3	4.94333E-3		
0.0	0.0	0.0	0.0		
1.06607E-2	2.49956E-4	1.82565E-6	5.46711E-3	1.00157E-6	0
5.36879E-3	0	0	0	0	0
0.0	0.0	0.0	0.0		
4.64814E-4	4.76596E-4	1.23810E-3	4.94333E-3		

Steel

9.83638E-2 1.35140E-1 2.24749E-1 2.83117E-1
7.75880E-3 4.55944E-3 5.20211E-3 2.40965E-3
0.0 0.0 0.0 0.0
7.42377E-3 1.18163E-4 8.25890E-7 4.35250E-3 3.41675E-7 0
4.64594E-3 0 0 0 0 0
0.0 0.0 0.0 0.0
2.16042E-4 2.06601E-4 5.56175E-4 2.40965E-3

Sodium/Steel Zone

9.65097E-2 9.87095E-2 1.34200E-1 4.1267E-1
8.15474E-3 3.46017E-3 3.44411E-3 3.03759E-3
0.0 0.0 0.0 0.0
7.73409E-3 1.94719E-4 8.89615E-7 3.22568E-3 7.98494E-7 0
2.90481E-2 0 0 0 0 0
0.0 0.0 0.0 0.0
2.25039E-4 2.33696E-4 5.39303E-4 3.03759E-3

Control Rod

1 1.39085E-1 2.28152E-1 3.18806E-1 6.27366E-1
2.13632E-2 3.34526E-2 7.44539E-2 3.12550E-1
0.0 0.0 0.0 0.0
1.26066E-2 1.33314E-4 1.08839E-6 4.32219E-3 1.85491E-7 0
3.68781E-4 0 0 0 0 0
0.0 0.0 0.0 0.0
8.62218E-3 2.91302E-2 7.40851E-2 3.12550E-1

Sodium Filled Control Rod Position

7.27587E-2 1.00218E-1 1.60703E-1 1.51576E-1
6.39525E-3 4.09435E-3 4.68721E-3 1.20796E-3
0.0 0.0 0.0 0.0
6.23393E-3 7.02121E-5 4.16388E-7 4.01375E-3 1.26939E-7 0
4.49111E-3 0 0 0 0 0
0.0 0.0 0.0 0.0
9.06964E-5 8.04746E-5 1.96101E-4 1.20796E-3

APPENDIX 5 FORTRAN Listing of PNCD Code

```
      PROGRAM PNCD
C
C   MAIN ROUTINE SETS UP POINTERS FORTRAN 77 STYLE
C
C   BASIC COMMON CONTAINS PROBLEM PARAMETERS, NG - NUMBER OF GROUPS
C    NDSCAT- NUMBER OF DOWN SCATTERS,  NUPSCAT - UP SCATTER  CURRENTLY ZERO
C    IMAX,JMAX,KMAX NUMBER OF POINTS IN THE X,Z,Y DIRECTIONS RESPECTIVELY.
C   NT1 - NT8 WORKFILES NAMGM GEOMETRY 1 -XYZ  2 -TRIANGULAR-Z   3-HEXAGONAL-Z
C
      CHARACTER SECT(8)
      COMMON  /WORD/SECT
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA,MESHQ,NT8,NT9,NT10,ICB
      COMMON /STORE/ CH(6000000)
      DIMENSION ICH(1)
      EQUIVALENCE (CH,ICH)
      ITOP=6000000
      ACC1=.00001
      ACC2=.0005
      MESHF=1
      MESHC=0
      MESHQ=0
      NUPSCAT=0
      NAMGM=1
      NERR=0
      NT1=1
      NT2=2
      NT3=12
      NT4=4
      NT5=8
      NT6=9
      NT7=10
      NT8=11
      NPN=1
      CALL INITL
      CALL GODEC
C
C  READ DATA
C
      CALL READD
      LPN=(NPN*(NPN+1))/2
      IF(NPN.EQ.1) THEN
      DO 2 J=1,3
      ITOP=ITOP-1
    2 CH(ITOP)=1./3.
      LOCD=ITOP
      ENDIF
      IH=5
      IF(NAMGM.EQ.3) IH=6
      ICB=2
C
C   SET UP STORAGE POINRS IN CH
C
      KA=1
      KB=KA+IH*JMAX*ICB
      KC=KB+JMAX*ICB
      KD=KC+JMAX*ICB
```

```
      KE=KD+JMAX*ICB
      KF=KE
      IDB=2
      IF(NAMGM.EQ.3) KF=KE+JMAX*ICB
      KG=KF+JMAX*ICB
      KS=KB+IH*NPTS*ICB
      KQ=KS+NPTS*IDB
      KM=KQ+NPTS*NDSCAT*IDB
      IF(MESHQ.NE.0) KM=KQ+NPTS*MAX0(2,NDSCAT)*IDB
      KL=KM-JMAX*IDB
      KR=KM+JMAX*IDB
      KSM=KM+NPTS*IDB*LPN
      KSQ=KSM
      LIM=KSM+IMAX*JMAX*IDB
      IF(MESHC.NE.0) THEN
      KSQ=LIM+NPTS*IDB
      LIM=KSQ+3*NPTS*IDB
      ENDIF
      IF(MESHQ.NE.0) THEN
      KSQ=LIM
      LIM=KSQ+NPTS*IDB
      ENDIF
      DO 1 K=1,LIM
    1 CH(K)=0.
      LOCS=ITOP-NPTS
      IEXB=LOCS-NG*LPN
      ITOP=IEXB-JMAX*IDB
      WRITE(6,100) LIM,ITOP,NPTS
  100 FORMAT(1X,' CALCULATION STORAGE  ',I10,'  BOTTOM OF DATA  ',I10,
     1/,' NUMBER OF MESH POINTS ',I10)
C
C  SET UP DIFFERENCE COEFFICIENTS
C
      CALL FDIFC(CH(KA),CH(KB),CH(KC),CH(KD),CH(KE),CH(KF),CH(KG),
     1CH(KM),CH(KS),              CH(IPLB),CH(IPRB),CH(LOCCA),CH(LOCI),
     2CH(LOCS),CH(IPI),CH(IPK),CH(IPJ)          )
C
C  SOLVE FOR FLUXES AND KEFF
      CALL SOLVE(CH(KA),CH(KB),CH(KC),CH(KD),CH(KE),CH(KF),CH(KG),CH(KS)
     1,CH(KQ),CH(KL),CH(KM),CH(KR),CH(IPLB),CH(IPRB),CH(LOCCA),CH(LOCI),
     2CH(LOCS),CH(IPI),CH(IPK),CH(IPJ),CH(IEXB),CH(ITOP),CH(KSQ),
     3CH(KSM),CH(KB))
      CALL READD
      STOP
      END
```

```
      SUBROUTINE READD
      CHARACTER SECT(8)
      COMMON  /WORD/SECT
      COMMON  /ERR/NDERR
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1NIS,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA,MESHQ
      COMMON /STORE/CH(1)
1     CALL WORDEC(SECT,8,0)
2     CALL JNDEX(J,SECT,'GC*I*BC*C*CA*XM*YM*ZM*GEOMETRY*STOP*PN*MESHF*ME
     1SHC*ALTER*PROFILE*MESHQ&',8)
      IF(J.EQ.1) THEN
      NG=INDEC(0)
      NDSCAT=INDEC(0)
      NUPSCAT=INDEC(0)
      GO TO 1
      ELSEIF(J.EQ.2) THEN
      CALL DATXS
      GO TO 2
      ELSEIF(J.EQ.3) THEN
      CALL FARRAY(BC,LL)
      GO TO 1
      ELSEIF(J.EQ.4)THEN
      CALL DATC
      GO TO 2
      ELSEIF(J.EQ.5)THEN
      IPLB=ITOP-KMAX*MESHF
      IPRB=IPLB-KMAX*MESHF
      ITOP=IPRB
      CALL DATCA(CH(IPLB),CH(IPRB))
      GO TO 2
      ELSEIF(J.EQ.6) THEN
      CALL FARRAY(CH,IMAX)
      CH(IMAX+1)=CH(IMAX)
      ITOP=ITOP-(IMAX+1)*MESHF+1
      IPI=ITOP
      IF(MESHF.NE.1) CALL EXPF(CH,IMAX+1,MESHF)
      CALL EQVEC(CH,CH(IPI),(IMAX+1)*MESHF-1)
      GO TO 1
      ELSEIF(J.EQ.7) THEN
      CALL FARRAY(CH,KMAX)
      IF(MESHF.NE.1) CALL EXPF(CH,KMAX,MESHF)
      ITOP=ITOP-KMAX*MESHF
      IPK=ITOP
      CALL EQVEC(CH,CH(IPK),KMAX*MESHF)
      GO TO 1
      ELSEIF(J.EQ.8) THEN
      CALL FARRAY(CH,JMAX)
      IF(MESHF.NE.1) CALL EXPF(CH,JMAX,MESHF)
      ITOP=ITOP-JMAX*MESHF
      IPJ=ITOP
      CALL EQVEC(CH,CH(IPJ),JMAX*MESHF)
      GO TO 1
      ELSEIF(J.EQ.9)THEN
      CALL WORDEC(SECT,8,0)
      CALL JNDEX(NAMGM,SECT,'XYZ*TRIZ*HEXZ&',8)
      GO TO 1
      ELSEIF(J.EQ.10)THEN
      STOP
      ELSEIF(J.EQ.11) THEN
      NPN=INDEC(0)
```

```
      CALL MOMENT(CH,CH(10000))
      GO TO  1
      ELSEIF(J.EQ.12) THEN
      MESHF=INDEC(0)
      GO TO 1
      ELSEIF(J.EQ.13)THEN
      MESHC=1
      GO TO 1
      ELSEIF(J.EQ.14) THEN
      CALL ALTER(CH(LOCI))
      GO TO 1
      ELSEIF(J.EQ.15) THEN
      CALL PROFIL(CH,CH(IPLB),CH(IPRB))
      GO TO 1
      ELSEIF(J.EQ.16) THEN
      MESHQ=1
      GO TO 1
      ELSEIF(J.GT.16)THEN
      WRITE(6,100)
  100 FORMAT(1X,'DATA READ IN ')
      IMAX=IMA
      JMAX=JMAX*MESHF
      KMAX=KMAX*MESHF
      NERR=NERR+NDERR
      IF(NERR.EQ.0) RETURN
      WRITE(6,101) NERR,NDERR
  101 FORMAT(1X,'THERE ARE ',2I6,' ERRORS PROGRAM IS STOPPING')
      STOP
      ENDIF
      END
      SUBROUTINE EXPF(A,JM,MF)
      DIMENSION A(1)
       XF=MF
      JJ=JM
      JK=JM*MF
      DO 1 J=1,JM
      X=A(JJ)/XF
      JJ=JJ-1
      DO 1 K=1,MF
      A(JK)=X
    1 JK=JK-1
      RETURN
      END
      SUBROUTINE EXPI(M,JM,MF)
      DIMENSION M(1)
      JJ=JM+1
      JK=JM*MF
      DO 1 J=1,JM
      JJ=JJ-1
      DO 1 K=1,MF
      M(JK)=M(JJ)
    1 JK=JK-1
      RETURN
      END
      SUBROUTINE DATC
      CHARACTER SECT*8
      COMMON  /WORD/SECT
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
      COMMON /STORE/ CH(1)
      DIMENSION ICH(1)
      EQUIVALENCE (CH,ICH)
```

```
      DO 4 I=1,1000
4     ICH(I)=0
      MAXC=0
      GO TO 2
  1   CALL WORDEC(SECT,8,0)
  2   CALL JNDEX(J,SECT,'C&',8)
      IF(J.NE.1) GO TO 3
      IC=INDEC(O)
      MAXC=MAX0(IC,MAXC)
      CALL IARRAY(CH(1000),JJ)
      IF(MESHF.NE.1) CALL EXPI(CH(1000),JMAX,MESHF)
      IF(JJ.NE.JMAX) THEN
      WRITE(6,100) JMAX,JJ
100   FORMAT(1X,I6,' CHANNEL ZONES READ IN  ',I6,' SPECIFIED BY ZM')
      NERR=NERR+1
      ENDIF
      ITOP=ITOP-JMAX*MESHF
      CALL EQVEC(CH(1000),CH(ITOP),JMAX*MESHF)
      ICH(IC)=ITOP
      GO TO 1
  3   ITOP=ITOP-MAXC
      LCC=ITOP
      CALL EQVEC(CH(1),CH(ITOP),MAXC)
      RETURN
      END
      SUBROUTINE DATCA(IBL,IBR)
      CHARACTER SECT*8
      COMMON  /WORD/SECT
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA
      COMMON /STORE/ CH(1)
      DIMENSION IBL(1),IBR(1),ICH(1)
      EQUIVALENCE (CH,ICH)
      LOC=0
      MM=0
      NPTS=0
      GO TO 2
1     CALL WORDEC(SECT,8,0)
2     CALL JNDEX(J,SECT,'CA&',8)
      IF(J.EQ.1) THEN
      CALL IARRAY(CH(LOC+1),II)
      DO 5 MM=1,KMAX
      I1=LOC
      IBL(MM)=0
      IBR(MM)=0
      DO 4 I=1,IMAX
      I1=I1+1
      IF(IBL(MM).EQ.0.AND.CH(I1).NE.0)IBL(MM)=I
 4    IF(IBL(MM).NE.0.AND.IBR(MM).EQ.0.AND.CH(I1).EQ.0)IBR(MM)=I-1
      IF(IBR(MM).EQ.0) IBR(MM)=IMAX
      NPTS=NPTS+IBR(MM)-IBL(MM)+1
 5    LOC=LOC+IMAX
      NPL=IMAX*KMAX
      IMA=IMAX
      IF(II .NE.NPL) THEN
      WRITE(6,100) II,NPL
 100  FORMAT(1X,I6,' CHANNEL ASSIGNMENTS GIVEN,SHOULD BE ',I6)
      NERR=NERR+1
      ENDIF
      GO TO 1
      ELSE
```

```
      LOC1=1
      IF(MESHF.NE.1) THEN
      NPTS=0
      IO=0
      IF(NAMGM.EQ.2) IO=1
      IL1=MESHF
      IL2=MESHF
      IL3=MESHF
      IF(NAMGM.EQ.2) IL3=2*MESHF-1
      IF(IBL(1).NE.2*(IBL(1)/2).AND.IBL(1).NE.1) IO=-IO
      IMA=IMAX/2
      IF(IBL(1).EQ.1.AND.IMA.EQ.2*(IMA/2 )) IO=-IO
      IMM=1
      JM=IMAX*KMAX+1
      MP=0
      DO 11 MM=1,KMAX
      IF(IO.GT.0) IL1=IL3
      IF(IO.GT.0) IL2=1
      IF(IO.LT.0) IL1=1
      IF(IO.LT.0) IL2=IL3
      DO 10 M=1,MESHF
      IM=IMM
      MP=MP+1
      JM2=JM
      IF(IL3.NE.IL1) THEN
      ILL=(IL3-IL1)/2
      NZC=ICH(IM)
      IF(IBL(MM).EQ.1.AND.MM.NE.1.AND.IBL(MM).LT.IBL(MM-1))NZC=0
      IF(IBL(MM).EQ.1.AND.MM.NE.KMAX.AND.IBL(MM+1).GT.IBL(MM))NZC=0
      DO 6 K=1,ILL
      ICH(JM)=NZC
6     JM=JM+1
      ENDIF
      ILL=IL1
      ILR=IL2
      DO 8 I=1,IMAX
      DO 7 K=1,ILL
      ICH(JM)=ICH(IM)
7     JM=JM+1
      ILT=ILL
      ILL=ILR
      ILR=ILT
8     IM=IM+1
      IF(IL3.NE.ILR) THEN
      ILL=ILL/2
      NZC=ICH(IM-1)
      IF(IBR(MM).EQ.IMAX.AND.MM.NE.1.AND.IBR(MM).GT.IBR(MM-1))NZC=0
      IF(IBR(MM).EQ.IMAX.AND.MM.NE.KMAX.AND.IBR(MM+1).LT.IBR(MM))NZC=0
      DO 9 K=1,ILL
      ICH(JM)=NZC
    9 JM=JM+1
      ENDIF
      IL1=IL1-2*IO
   10 IL2=IL2+2*IO
      IMM=IM
   11 IO=-IO
      M=0
      IMA=(JM-IMAX*KMAX+1)/(KMAX*MESHF)
      IMB=IMA-1
      NPTS=0
      DO 13 MM=1,KMAX
      DO 13 MN=1,MESHF
      M=M+1
      IBL(M)=1
```

```
      IBR(M)=0
      LOC=(M-1)*IMA+IMAX*KMAX
      I1=LOC+1
      I2=LOC+IMA
      WRITE(6,999) (ICH(K),K=I1,I2)
999   FORMAT(1X,24I3)
      DO 12 I=1,IMB
      LOC=LOC+1
      IF(ICH(LOC).EQ.0.AND.ICH(LOC+1).NE.0) IBL(M)=I+1
12    IF(ICH(LOC).NE.0.AND.ICH(LOC+1).EQ.0) IBR(M)=I
      IF(IBR(M).EQ.0) IBR(M)=IMA
13    NPTS=NPTS+(IBR(M)-IBL(M)+1)
      LOC1=IMAX*KMAX+1
      LOC=IMA*KMAX*MESHF
      ENDIF
      ITOP=ITOP-LOC
      LOCCA=ITOP
      CALL EQVEC(CH(LOC1),CH(ITOP),LOC)
      NPTS=NPTS*JMAX*MESHF
      ENDIF
      RETURN
      END
      SUBROUTINE DATXS
      CHARACTER SECT*8
      COMMON  /WORD/SECT
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI
      COMMON /STORE/ CH(1)
      DIMENSION ICH(1)
      EQUIVALENCE  (CH,ICH)
      MAXI=0
      NXSM=0
      DO 1 J=1.1000
    1 ICH(J)=0
      GO TO 3
   2  CALL WORDEC(SECT,8,0)
   3  CALL JNDEX(J,SECT,'I&',8)
      IF(J.EQ.1) THEN
      II=INDEC(0)
      CALL FARRAY(CH(1001),NXS)
      NXSM=MAX0(NXS,NXSM)
      ITOP=ITOP-NXS
      CALL EQVEC(CH(1001),CH(ITOP),NXS)
      ICH(II)=ITOP
      MAXI=MAX0(II,MAXI)
      GO TO 2
      ELSE
      ITOP=ITOP-MAXI
      CALL EQVEC(CH(1),CH(ITOP),MAXI)
       LOCI=ITOP
      RETURN
      ENDIF
      END
      SUBROUTINE ALTER(IXS)
      CHARACTER SECT(8)
      COMMON  /WORD/SECT
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG
      COMMON /STORE/ CH(1)
      DIMENSION ICH(1),IXS(1)
```

```
      EQUIVALENCE (CH,ICH)
      NXS=(5+NDSCAT)*NG
      INUM=INDEC(0)
      VF=FLDEC(0)
      LOCX=IXS(INUM)
      DO 1 I=1,NXS
      CH(LOCX)=CH(LOCX)*VF
    1 LOCX=LOCX+1
      RETURN
      END
      SUBROUTINE MOMENT(A,IA)
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC
      COMMON /STORE/CH(1)
      DIMENSION C(6,15),IC(6,15),W(12),IW(12),Y(12),IY(12),IDV(3,3)
     1,A(1),IA(1)
      DATA IDV/1,2,2,1,2,2,1,1,3/
      LOC2=1
      LOC3=20001
      NMOM=(NPN*(NPN+1))/2
      LEND=3*NMOM
      LOCD=ITOP-LEND
      LOCT=LOCD
      DO 8 L=1,NPN,2
      NM=1
      DO 8 M=1,L
      DO 7 NN=1,NM
      LP=L-1
      MP=M-1
      WRITE(6,100) LP,MP,NN
  100 FORMAT(1X,'L=',I4,'  M=',I4,I4,'  MOMENT EQUATION')
      DO 1 I=1,15
      DO 1 J=1,6
      C(J,I)=0.
    1 IC(J,I)=0.
      CALL COEF(W,IW,L-1,M-1,NN)
      DO 5 J=1,6
      LL=L+2*((J-1)/3)-2
      MM=M+J-3*((J-1)/3)-3
      IF(LL.LT.0.OR.MM.LT.0.OR.MM.GT.LL) GO TO 5
      DO 4 K=1,2
      IF(K.EQ.2.AND.MM.EQ.0) GO TO 4
      XL=LL
      XL=1./(2.*XL+1)
      CALL COEF(Y,IY,LL,MM,K)
      JW=J+(K-1)*6
      LW=IW(JW)
      DO 2 JJ=1,6
      LLL=LL-1 +2*((JJ-1)/3)
      MMM=MM-2+JJ-3*((JJ-1)/3)
      IF(LLL.LT.0.OR.LLL.GT.NPN.OR.MMM.LT.0.OR.MMM.GT.LLL) GO TO 2
      LOC=((J-1)/3+(JJ-1)/3)*2+J+JJ-1
      LY1=IY(JJ)
      LY2=IY(JJ+6)
      LWY1=IDV(LW,LY1)
      LWY2=IDV(LW,LY2)+3
      C(LWY1,LOC)=W(JW)*Y(JJ)*XL+C(LWY1,LOC)
      C(LWY2,LOC)=W(JW)*Y(JJ+6)*XL+C(LWY2,LOC)
      IF(W(JW).NE.0..AND.Y(JJ).NE.0.)
     1IC(LWY1,LOC)=IW(JW)*IY(JJ)
      IF(W(JW).NE.0..AND.Y(JJ+6).NE.0.)
     1IC(LWY2,LOC)=IW(JW)*IY(JJ+6)
```

```
    2   CONTINUE
     3  CONTINUE
     4  CONTINUE
     5  CONTINUE
        DO 6 J=1,15
        LL=L+2*((J-1)/5)-3
        MM=M+J-5*((J-1)/5)-4
        DO 6 K=1,2
        I=3*(K-1)+1
        II=3*K
        IF(C(I,J).NE.0..OR.C(I+1,J).NE.0..OR.C(I+2,J).NE.0.) THEN
        WRITE(6,101) LL,MM,K,(IC(KK,J),C(KK,J),KK=I,II)
  101   FORMAT(4X,3I4,3(I4,E14.6))
        IF(LP.EQ.LL.AND.MP.EQ.MM.AND.NN.EQ.K) THEN
        CALL EQVEC(C(I,J),CH(LOCT),3)
        LOCT=LOCT+3
        ELSE
        CALL EQVEC(C(I,J),A(LOC2),3)
        CALL EQVEC(IC(I,J),IA(LOC2),3)
        LOC2=LOC2+3
        LOCF=(LL*(LL-1))/2+MAX0(2*MM-1,0)+K
C       IF(LL.EQ.0) LOCF=1
        IA(LOC3)=LOCF
        LOC3=LOC3+1
        ENDIF
        ENDIF
     6  CONTINUE
        IA(LOC3)=-1
     7  LOC3=LOC3+1
     8  NM=2
        LOCC=LOCD-LOC2
        CALL EQVEC(A,CH(LOCC),LOC2)
        LOCIC=LOCC-LOC2
        CALL EQVEC(IA,CH(LOCIC),LOC2)
        LENF=LOC3-20000
        LOCF=LOCIC-LENF
        CALL EQVEC(IA(20001),CH(LOCF),LENF)
        ITOP=LOCF
        RETURN
        END
        SUBROUTINE COEF(W,IW,L,M,I)
        DIMENSION W(12),IW(12),IS(2)
        DATA IS/1,-1/
        DO 1 J=1,6,3
        IW(J)=I
        IW(J+2)=I
        IW(J+1)=3
        IW(J+7)=3
        IW(J+8)=3-I
    1   IW(J+6)=3-I
        W(1)=1
        W(2)=2*(L-M)*(2-I)
        W(3)=-(L-M)*(L-M-1)*IS(I)
        W(4)=-1
        W(5)=2*(L+M+1)*(2-I)
        W(6)=(L+M+1)*(L+M+2)*IS(I)
        W(7)=-IS(I)
        W(8)=2*(L-M)*(I-1)
        W(9)=-(L-M)*(L-M-1)
        W(10)=IS(I)
        W(11)=2*(L+M+1)*(I-1)
        W(12)=(L+M+1)*(L+M+2)
        IF(M.EQ.1) THEN
        W(1)=2.
```

```
      W(4)=-2.
      W(7)=0.
      W(10)=0.
      ENDIF
      DO 2 J=1,12
2     W(J)=W(J)*0.5
      RETURN
      END
```

```
      SUBROUTINE FDIFC(A,B,C,D,E,F,G,FM,FS,IBL,IBR,ICA,IXS,LFS,
     1X,Y,Z  )
      REAL*8 A,B,C,D,E,F,G,FS
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA,MESHQ,NT8,NT9,NT10,ICB
      DIMENSION A(1),B(1),C(1),D(1),E(1),G(1),FM(1)            ,F(1),
     1IBL(1),IBR(1),ICA(1),IXS(1),FS(1)            ,LFS(1),X(1),Y(1)
     2,Z(1),IRC(3,4)
      DATA IRC/1,-1,-1,-1,1,-1,1,1,1,-1,-1,1/
      COMMON /STORE/CH(2)
      DIMENSION ICH(1)
      EQUIVALENCE (CH,ICH)
C
C   FORM DIFFERENCE COEFFICIENTS
C
      LXS=3*NG+NDSCAT*NG
      NPT=0
      AF=2.
      VF=1.
      DF=2.
      R3=1.
      IF(NAMGM.EQ.2) R3=SQRT(3.)
      IF(NAMGM.EQ.2) AF=R3*2.
      IF(NAMGM.EQ.2) VF=R3/4.
      IF(NAMGM.EQ.2) DF=AF
      IF(NAMGM.EQ.3) AF=2./SQRT(3.)
      IF(NAMGM.EQ.3) VF=SQRT(3.)*0.5
C
C XU , XUU AND XBB ARE USED TO SET G EQUAL ZERO IF TRIZ GEOMETRY IS
C BEING USED WHEN TRIANGLE POINTS IN Y DIRECTION
C
      XUU=1.
      XBB=1.
      REWIND NT2
      REWIND NT3
      DO 10 N=1,NG
      NDSCA=MIN0(N-1,NDSCAT)
      LOCDT=LOCD
      DO 10 LPL=1,NPN,2
      LPM=2*LPL-1
      XL=  LPM
      DO 10 LM=1,LPM
      MM=3
      IF(LM.NE.1) MM=LM-4*((LM-2)/4)-1
      BCB1=BC(1)
      BCB2=BC(2)
      IF(IRC(1,MM).LT.0.AND.BC(1).EQ.0.) BCB1=1E10
      BCB3=BC(3)
      IF(IRC(2,MM).LT.0.AND.BC(3).EQ.0.) BCB3=1E10
      BCB4=BC(4)
      BCB5=BC(5)
      BCB6=BC(6)
      IF(IRC(3,MM).LT.0.AND.BC(5).EQ.0.) BCB5=1E10
      II=IMAX/2+1
      DXX=CH(LOCDT)
      DYY=CH(LOCDT+1)
      DZZ=CH(LOCDT+2)
      LOCDT=LOCDT+3
      XU=1.
      IF(II.NE.2*(II/2)) XU=0.
```

```
      ILT=IBL(1)
      IRT=IBR(1)
      IL=ILT
      IR=IRT
      KK=1
      KS=1
C
C  IHX IHY FOR HEXAGONAL GEOMETRY TO ALLOW FOR PLAN OFFSET
C
      IHX=0
      IHY=0
      KF=1
      IF(NAMGM.EQ.3) IHX=1-KMAX/2+2*(KMAX/4)
      IF(NAMGM.EQ.3)IHY=1
      DO 5  MM=1,KMAX
      IF(NAMGM.EQ.3) IHX=1-IHX
      ILB=IL
      IL=ILT
      IRB=IR
      IR=IRT
      IF(MM.NE.KMAX) THEN
      IRT=IBR(MM+1)
      ILT=IBL(MM+1)
      ENDIF
      IF(BCB3.NE.BCB2.AND.IL.LT.ILT) BCB3=BCB2
      IF(BCB4.NE.BCB2.AND.IR.GT.IRT) BCB4=BCB2
      IF(NAMGM.EQ.2) THEN
      XU=1.-XU
      IF(ILT.LT.IL)XU=1
      IF(ILT.GT.IL)XU=0.
      XUU=XU
      XBB=1.-XU
      ENDIF
      LOCP=(MM-1)*IMAX+IL
      LOCPU=LOCP-IMAX+IHX-IHY
      NCHR=ICA(LOCP)
      LOCR=LCC+NCHR-1
      LOCR=ICH(LOCR)
      KG=KK-(IRB-IL+1+IHY-IHX)*JMAX*IH
      DO 5  II=IL,IR
      KF=KG+IH*JMAX
      LOCP=LOCP+1
      NCHR=ICA(LOCP)
      LOC=LOCR
      LOCR=NCHR+LCC-1
      LOCR=ICH(LOCR)
       NCHT=0
      NCHTH=0
      IF(MM.NE.KMAX) THEN
      LOCPT=LOCP+IMAX -1+IHX
      NCHT=ICA(LOCPT)
      LOCT=LCC+NCHT-1
      LOCT=ICH(LOCT)
      IF(NAMGM.EQ.3.AND.(II.NE.1.OR.IHX.EQ.1))THEN
      NCHTH=ICA(LOCPT-1)
      LOCTH=LCC+NCHTH-1
      LOCTH=ICH(LOCTH)
      ENDIF
      ENDIF
      NCHU=0
      NCHUH=0
      IF(MM.NE.1.AND.NAMGM.NE.3)NCHU=ICA(LOCPU)
      IF(MM.NE.1.AND.(IHX.EQ.1.OR.II.NE.1).AND.NAMGM.EQ.3)
     1NCHU=ICA(LOCPU)
```

```
      IF(MM.NE.1.AND.NAMGM.EQ.3) NCHUH=ICA(LOCPU+1)
      LOCPU=LOCPU+1
      AXY=Y(MM)*X(II)*VF
      AREAZ=Y(MM)*X(II)*VF*DZZ
      DO 4 J=1,JMAX
      AREAY=X(II)*Z(J)*AF*DYY
      AREAX=Y(MM)*Z(J)*AF*DXX
      VOL=AXY*Z(J)*XL
C
C   SET UP POINTER FOR FISSION SPECTRUM
C
      NZ=ICH(LOC)
      IF(N.EQ.1.AND.LPL.EQ.1) THEN
      NPT=NPT+1
      LFS(NPT)=IXS(NZ)+LXS-1
      ENDIF
C
C FIND MATERIAL NUMBERS IN AXIAL CHANNELS
C
      IF(J.NE.JMAX) NZ1=ICH(LOC+1)
      IF(II.NE.IR) NZ2=ICH(LOCR)
      IF(MM.NE.KMAX)NZ3=ICH(LOCT)
      IF(MM.NE.KMAX.AND.NAMGM.EQ.3) NZ3H=ICH(LOCTH)
      LOC=LOC+1
      LOCR=LOCR+1
      LOCT=LOCT+1
      LOCTH=LOCTH+1
C
C   LOCX NUMBERS ARE CROSS SECTION LOCATIONS
C
      LOCX=IXS(NZ)+N-1
      LOCX1=IXS(NZ1)+N-1
      LOCX2=IXS(NZ2)+N-1
      LOCX3=IXS(NZ3)+N-1
      LOCX3H=IXS(NZ3H)+N-1
      IF(J.NE.1) C(KK)=D(KK-1)
      IF (J.NE.JMAX) D(KK)=-AREAZ/(Z(J)*CH(LOCX)+Z(J+1)*CH(LOCX1))*2.
      IF (II.NE.IR)  B(KK)=-AREAX/(X(II)*CH(LOCX)+X(II+1)*CH(LOCX2))
      IF(NCHT.NE.0)
     1G(KK)=-AREAY/(Y(MM)*CH(LOCX)+Y(MM+1)*CH(LOCX3))*XUU
      IF(NCHTH.NE.0)
     1F(KK)=-AREAY/(Y(MM)*CH(LOCX)+Y(MM+1)*CH(LOCX3H))
      LOCNF=LOCX+2*NG
      IF(LPL.EQ.1) FS(KS)=CH(LOCNF)*VOL
      LOCA=LOCX
      IF(LPL.EQ.1) LOCA=LOCA+NG
      E(KK)=-B(KK)-C(KK)-D(KK)-G(KK)+VOL*CH(LOCA)
      IF(II.NE.IL)E(KK)=E(KK)-A(KK)
      IF(NCHU.NE.0) E(KK)=E(KK)-G(KG)
      IF(NCHUH.NE.0) E(KK)=E(KK)-F(KF)
      IF(NAMGM.EQ.3) E(KK)=E(KK)-F(KK)
      KG=KG+1
      KF=KF+1
      ZX=Z(JMAX)
      XMX=X(IR)
      IF(J.EQ.1) E(KK)=E(KK)+AREAZ*BCB5/(1.+1.5*CH(LOCX)*BCB5*Z(1))*3.
      IF(J.EQ.JMAX)E(KK)=E(KK)+AREAZ*BCB6/(1.+1.5*CH(LOCX)*BCB6*ZX)*3.
      IF(II.EQ.IL)E(KK)=E(KK)+AREAX*BCB3/(DF+3.*CH(LOCX)*BCB3*X(1))*3.
      IF(II.EQ.IR) E(KK)=E(KK)+AREAX*BCB4/(DF+3.*CH(LOCX)*BCB4*XMX)*3.
      IF(NCHU.EQ.0)
     1E(KK)=E(KK)+AREAY*BCB1/(DF+3.*CH(LOCX)*BCB1*Y(MM))*XBB*3.
      IF(NCHT.EQ.0)
     1E(KK)=E(KK)+AREAY*BCB2/(DF+3.*CH(LOCX)*BCB2*Y(MM))*XUU*3.
      IF(NCHUH.EQ.0.AND.NAMGM.EQ.3)
```

```
     1E(KK)=E(KK)+AREAY*BC(1)/(DF+3.*CH(LOCX)*BC(1)*Y(MM))*3.
      IF(NCHTH.EQ.0.AND.NAMGM.EQ.3)
     1E(KK)=E(KK)+AREAY*BC(2)/(DF+3.*CH(LOCX)*BC(2)*Y(MM))*3.
      IF(NDSCA.NE.0.AND.LPL.EQ.1) THEN
      LOCS=IXS(NZ)+3*NG+(N-2)*NDSCAT
       KSS=KS
       DO 1 NS=1,NDSCA
       FM(KSS)=CH(LOCS)*VOL
       LOCS=LOCS-NDSCAT+1
 1     KSS=KSS-NPTS
      ENDIF
      KS=KS+1
 4    KK=KK+1
      LOCR=LOCR-JMAX
      IF(NAMGM.EQ.2) THEN
       XBB=1.-XBB
       XUU=1.-XUU
       ENDIF
C
C  ADJUST C D E FOR FORWARD SUBSTITUTION BACKWARD DIFFERENCE
C
      KA=KK-JMAX
      KB=KK-1
      DO 2 J=2,JMAX
      D(KA)=D(KA)/E(KA)
      E(KA+1)=E(KA+1)-D(KA)*C(KA+1)
 2    KA=KA+1
      KA=KK-JMAX
      KK=KK+(IH-1)*JMAX
 5    KG=KG+(IH-1)*JMAX
      LEN1=IH*NPTS
      LEN2=NDSCAT*NPTS
      KS=1-(NDSCA-1)*NPTS
      LL=MAX0(1,NDSCA)*NPTS
      IF(LPL.EQ.1)CALL FILEW(NT3,FM(KS),LL)
      IF(LPL.EQ.1) THEN
      ENDIF
      CALL FILEW(NT3,B,IH*NPTS*ICB)
 10   IF(LPL.EQ.1) CALL FILEW(NT2,FS,NPTS*2)
C
C  FINITE DIFFERENCE COEFFICIENTS EVALUATED AND STORED
C
      RETURN
      END
```

```
      SUBROUTINE SOLVE(A,B,C,D,E,F,G,FS,Q,FL,FM,FR,IBL,IBR,ICA,IXS,LFS,
     1X,Y,Z,EXB,S,QS,SM,BB)
      REAL*8 A,B,C,D,E,G
      REAL*8 FL,FM,FR,Q,QS, FS,SF,SFL,EIGD,FMAX,FMIN
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA,MESHQ,NT8,NT9,NT10,ICB
      DIMENSION A(1),B(1),C(1),D(1),E(1),G(1),FL(1),FM(1),FR(1),F(1),
     1IBL(1),IBR(1),ICA(1),IXS(1),FS(1),Q(1),EXB(1),LFS(1),X(1),Y(1)
     2,Z(1),S(1),QS(1),SM(1),BB(1)
      COMMON /STORE/CH(2)
      DIMENSION ICH(1)
      EQUIVALENCE (CH,ICH)
C
C   CALCULATE OVER RELAXATION COEFFICIENTS
C
      REWIND NT3
      XNPT=NPTS
      SQRF=SQRT(XNPT)
      SQRR=1./SQRF
      NEX=0
      DO 20 N=1,NG
      DO 20 LPL=1,LPN
      DO 11 L=1,NPTS
   11 FM(L)=SQRR
      IF(LPL.EQ.1) CALL FILER(NT3,B,LEN)
      CALL FILER(NT3,B,LEN)
      SUMB=1.
      NITER=0
  3   NITER=NITER+1
      DO 6  K=1,NPTS
 6    FM(K)=FM(K)/SUMB
      KK=1
      KM=1
      IHX=0
      IHY=0
C
C  INCREMENTS FOR OFFSET IN HEXAGONAL GEOMETRY
C
      IF(NAMGM.EQ.3) IHX=1-KMAX/2+2*((KMAX)/4)
      IF(NAMGM.EQ.3) IHY=1
C
      DO 18 MM=1,KMAX
      ILL=IL
      ILR=IR
      IL=IBL(MM)
      IR=IBR(MM)
      IF(NAMGM.EQ.3) IHX=1-IHX
      LGAP=(ILR-IL+1-IHX+IHY)*JMAX
      LOC2=MM*IMAX+IL+IHX
      LOC1=LOC2-IMAX*2-IHY
      IF(MM.NE.KMAX) THEN
      ILU=IBL(MM+1)
      LGAP1=(IR-ILU+1+IHX)*JMAX
      ENDIF
      DO 39 I=IL,IR
      LC1=0
      LC2=0
      LC1H=0
      LC2H=0
      IF(MM.NE.1.AND.NAMGM.NE.3) LC1=ICA(LOC1)
```

```
      IF(MM.NE.1.AND.NAMGM.EQ.3.AND.(IHX.EQ.1.OR.I.NE.1))LC1=ICA(LOC1)
      IF(MM.NE.KMAX) LC2=ICA(LOC2)
      IF(MM.NE.1.AND.NAMGM.EQ.3) LC1H=ICA(LOC1+1)
      IF(MM.NE.KMAX.AND.NAMGM.EQ.3.AND.(I.NE.1.OR.IHX.EQ.1))
     1LC2H=ICA(LOC2-1)
      LOC1=LOC1+1
      LOC2=LOC2+1
      DO 7 J=1,JMAX
 7    FS(J)=0.
      IF( I.NE.IL)THEN
      KA=KK
      KT=KM
      DO 12 J=1,JMAX
      FS(J)=FS(J)-A(KA)*FL(KT)
      KT=KT+1
 12   KA=KA+1
      ENDIF
      IF(I.NE.IR) THEN
      KA=KK
      KT=KM
      DO 13 J=1,JMAX
      FS(J)=FS(J)-B(KA)*FR(KT)
      KT=KT+1
 13   KA=KA+1
      ENDIF
      IF(LC1.NE.0)THEN
      KT=KM-LGAP
      KG=KK-LGAP*IH
      DO 14 J=1,JMAX
      FS(J)=FS(J)-G(KG)*FM(KT)
      KT=KT+1
 14   KG=KG+1
      ENDIF
      IF(LC2.NE.0) THEN
      KT=KM+LGAP1
      KG=KK
      DO 15 J=1,JMAX
      FS(J )=FS(J)-G(KG)*FM(KT)
      KT=KT+1
 15   KG=KG+1
      ENDIF
C
C  HEXAGONAL GEOMETRY ONLY
C
      IF(LC1H.NE.0) THEN
      KT=KM-LGAP+JMAX
      KG=KK-LGAP*IH
      DO 47 J=1,JMAX
      FS(J)=FS(J)-F(KG)*FM(KT)
      KT=KT+1
 47   KG=KG+1
      ENDIF
      IF(LC2H.NE.0)THEN
      KT=KM+LGAP1-JMAX
      KG=KK
      DO 48 J=1,JMAX
      FS(J )=FS(J)-F(KG)*FM(KT)
      KT=KT+1
 48   KG=KG+1
      ENDIF
C
C END HEXAGONAL GEOMETRY ONLY
C
      FS(1)=FS(1)/E(KK)
```

```
      DO 16 J=2,JMAX
      KK=KK+1
   16 FS(J)=(FS(J)-C(KK)*FS(J-1))/E(KK)
      KM=KM+JMAX-1
      FM(KM)=FS(JMAX)
      JJ=JMAX
      DO 17 J=2,JMAX
      KM=KM-1
      KK=KK-1
      JJ=JJ-1
   17 FM(KM)=FS(JJ)-D(KK)*FM(KM+1)
      KA=KM+JMAX-1
      KM=KM+JMAX
   39 KK=KK+IH*JMAX
   18 CONTINUE
      SUMA=SUMB
      SUMB=0.
      DO 19 K=1,NPTS
   19 SUMB=SUMB+FM(K)*FM(K)
      SUMB=SQRT(SUMB)
      IF(ABS(SUMB-SUMA).GT..0005.AND.NIT.LT.50) GO TO 3
      NEX=NEX+1
      EXB(NEX)=2./(1.+SQRT(1.-SUMB))
   20 CONTINUE
      IF(LPN.EQ.1)
     1WRITE(6,120) (EXB(K),K=1,NG)
  120 FORMAT(1X,'OVER RELAXATION FACTORS  ',/,(5E14.6))
      IF(LPN.NE.1) THEN
      WRITE(6,149)
  149 FORMAT(1X,'   OVER RELAXATION FACTORS')
      K2=0
      DO 99 N=1,NG
      K1=K2+1
      K2=K2+LPN
   99 WRITE(6,150) (EXB(K), K=K1,K2)
  150 FORMAT(5E14.6)
      ENDIF
C
C   BEGIN CALCULATION
C
      NIT=0
      CON1=1.
      DEL=1.
      EIGD=1.
      MSW=0
      REWIND NT4
      DO 41 K=1,NPTS
   41 FM(K)=SQRR
      IF(LPN.NE.1) THEN
      K=NPTS
      DO 45 L=2,LPN
      DO 45 J=1,NPTS
      K=K+1
   45 FM(K)=0.
      ENDIF
      DO 51 N=1,NG
   51 CALL FILEW(NT4,FM,NPTS*2*LPN)
C
C   FORM FISSION SOURCE FOR OUTER ITERATION
C
   21 DO 53 K=1,NPTS
   53 Q(K)=FS(K)
      NIT=NIT+1
      DO 22 K=1,NPTS
```

```
   22  FS(K)=0.
C
C    IF OPTION MESHC SELECTED AND PROBLEM SUFFICIENTLY CONVERGED
C    CALCULATE SECOND ORDER MESH ERROR SOURCE
C
       IF(ABS(DEL).LE.ACC1*5..AND.MESHC.NE.0.AND.MSW.EQ.0) MSW=1
       IF(ABS(DEL).LE.ACC1*5..AND.MESHQ.NE.0.AND.MSW.EQ.0.AND.ACC2*2..GT.
      1ABS(FMAX-FMIN))MSW=2
       REWIND NT2
       REWIND NT4
       DO 24 N=1,NG
       CALL FILER(NT2,B,LEN)
       CALL FILER(NT4,FM,LEN)
       DO 23 K=1,NPTS
   23  FS(K)=B(K)*FM(K)+FS(K)
   24  CONTINUE
       FMAX=0.
       FMIN=100
       IF( NIT.EQ.1) GO TO 37
       SF=0.
       SFL=0.
       DO 25 K=1,NPTS
       IF(Q(K).EQ.0.) GO TO 25
       IF(FMAX.LT.FS(K)/Q(K)) KMA=K
       IF(FMIN.GT.FS(K)/Q(K)) KMI=K
       FMAX=DMAX1(FMAX,FS(K)/Q(K))
       FMIN=DMIN1(FMIN,FS(K)/Q(K))
       SFL=SFL+Q(K)*FS(K)
       SF=SF+FS(K)*FS(K)
   25  CONTINUE
C
C    ADJUST EIGENVALUE
C
       DEL=1.-SF/SFL
       EIGD=EIGD*SF/SFL
       EIG=EIGD
       WRITE(6,102) FMAX,EIG,FMIN
  102  FORMAT(1X,3F12.6)
C      IF(LPN.NE.1.AND.NIT.EQ.15) STOP
   37  IF(ABS(DEL).LT.ACC1.AND.ABS(FMAX-FMIN).LE.ACC2) THEN
       IF(MESHQ.NE.0.AND.MSW.EQ.0) GO TO 26
       IF(MSW.NE.0.AND.CON1.NE.0.) THEN
       CON1=0.
       GO TO 26
       ENDIF
       WRITE(6,101) EIG
  101  FORMAT(1X, 'EIGENVALUE IS ',F12.6)
       RETURN
       ELSEIF(NIT.GT.100) THEN
       WRITE(6,103)NIT,DEL,FMAX,FMIN
  103  FORMAT(1X, 'ITERATION LIMIT ',I4,' EXCEEDED WITHOUT CONVERGENCE',/
      1,' EIGENVALUE CHANGE ',F12.6,' SOURCE LIMITS  ',2F12.6)
       RETURN
       ENDIF
C
C    CALCULATE FLUXES IF NO CONVERGENCE
C
   26  DO 66 K=1,NPTS
   66  Q(K)=0.
       IF(MSW.EQ.2)
      1CALL QMESHA(A,B,C,G,Q ,FM,IBL,IBR,ICA,IXS,LFS,X,Y,Z,QS,SM,S,JMAX,
      1BB)
       REWIND NT3
       REWIND NT4
```

```
      REWIND NT5
      REWIND NT6
      NEXB=0
C
C   BEGIN LOOP OVER GROUPS FOR FLUX SOLUTION
C
      DO 40 N=1,NG
C     WRITE(6,329) N
C 329 FORMAT(1X,' GROUP ',I5)
C     NTF=NT4+N
C     REWIND NTF
      DO 52 K=1,NPTS
 52   Q(K)=0.
C
C   FORM FISSION SOURCE FOR GROUP N IN INNER ITERATION
C   ADD IN MESH ERROR SOURCE IF AVAILABLE
C
      IF(MSW.NE.0) CALL FILER(NT6,Q,LEN)
      L=0
      DO 27 K=1,NPTS
      LL=L
      L=LFS(K)
      IF(L.NE.LL) ISPEC=L+N
  27  Q(K)=FS(K)*CH(ISPEC)/EIGD+Q(K)
C     REWIND NNT
      CALL FILER(NT3,BB,LEN)
      NNT=NNT+1
C
C  ADD IN DOWN SCATTER
C
C251  FORMAT(1X ,' SOURCE ',/,(5E14.6))
      NDS=MIN0(NDSCAT,N-1)
      IF(NDS.GT.0) THEN
      KMS=(1-NDS)*NPTS
      KS=0
      DO 28 NN=1,NDS
      DO 28 K=1,NPTS
      KMS=KMS+1
      KS=KS+1
  28  Q(K)=Q(K)+FM(KMS)*BB(KS)
      NDT=MIN0(NDSCAT,N)-1
      KM=1-NDT*NPTS
      KS=KM+NPTS
      CALL EQVEC(FM(KS),FM(KM),NDT*NPTS*2)
      ENDIF
C
C        SET STARTING LOCATIONS OF MOMENT EQUATION COEFFICIENTS FOR PN
C
      LLU=LOCC
      LLV=LOCIC
      LLF=LOCF
      LLD=LOCD
      CALL FILER(NT4,FM,LEN)
C
C           LOOP OVER MOMENTS IF PN CALCULATION
C
      DO 65 LPL=1,LPN
      CALL FILER(NT3,B,LEN)
      NEXB=NEXB+1
      IF(LPL.GT.1.AND.MSW.NE.0) GO TO 65
      IF(LPN.NE.1) THEN
 61   IF(ICH(LLF).EQ.-1) GO TO 64
      IP=ICH(LLF)
      KM=(IP-1)*NPTS+1
```

```
      L=0
      LL=0
   62 IF(LL+2*L+1.GE.IP) GO TO 63
      LL=LL+2*L+1
      L=L+2
      GO TO 62
   63 M=IP-LL
C
C        ADD MOMENT IP CONTRIBUTION TO LPL EQUATION
C
      CALL DXYZ(B,C,G,FM(KM),Q,CH(LLV),CH(LLU),CH(LLD),IBL,IBR,SM,S,
     1JMAX,M)
      KM4=IMAX*JMAX
      LLF=LLF+1
      LLU=LLU+3
      LLV=LLV+3
      GO TO 61
   64 LLF=LLF+1
      ENDIF
      LLD=LLD+3
      NITER=0
      EXBG=EXB(NEXB)
      FSQ2=0
   29 NITER=NITER+1
      FSQ1=FSQ2
      KK=1
      KQ=0
      KM=(LPL-1)*NPTS
      IHX=0
      IHY=0
      IF(NAMGM.EQ.3) IHX=1-KMAX/2+2*(KMAX/4)
      IF(NAMGM.EQ.3) IHY=1
      DO 43 MM=1,KMAX
      ILL=IL
      ILR=IR
      IL=IBL(MM)
      IR=IBR(MM)
      IF(NAMGM.EQ.3) IHX=1-IHX
      LGAP=(ILR-IL+1-IHX+IHY)*JMAX
      IF(MM.NE.KMAX) THEN
      ILT=IBL(MM+1)
      IRT=IBR(MM+1)
      LGAP1=(IR-ILT+1+IHX)*JMAX
      ENDIF
      LOC2=MM*IMAX+IL+IHX
      LOC1=LOC2-2*IMAX-IHY
      DO 42 I=IL,IR
C     WRITE(6,241) (S(J),J=1,JMAX)
C 241 FORMAT(1X,'S',5E14.6)
      LC1=0
      LC2=0
      LC1H=0
      LC2H=0
      IF(MM.NE.1.AND.NAMGM.NE.3) LC1=ICA(LOC1)
      IF(MM.NE.1.AND.NAMGM.EQ.3.AND.(IHX.EQ.1.OR.I.NE.1))LC1=ICA(LOC1)
      IF(MM.NE.KMAX) LC2=ICA(LOC2)
      IF(MM.NE.1.AND.NAMGM.EQ.3) LC1H=ICA(LOC1+1)
      IF(MM.NE.KMAX.AND.NAMGM.EQ.3.AND.(I.NE.1.OR.IHX.EQ.1))
     1LC2H=ICA(LOC2-1)
      LOC1=LOC1+1
      LOC2=LOC2+1
      DO 38 J=1,JMAX
      KQ=KQ+1
   38 S(J)=Q(KQ)
```

```
      IF( I.NE.IL)THEN
      KA=KK
      KT=KM
      DO 30 J=1,JMAX
      KT=KT+1
      S(J)=S(J)-A(KA)*FL(KT)
   30 KA=KA+1
      ENDIF
      IF(I.NE.IR) THEN
      KA=KK
      KT=KM
      DO 31 J=1,JMAX
      KT=KT+1
      S(J)=S(J)-B(KA)*FR(KT)
   31 KA=KA+1
      ENDIF
      IF(LC1.NE.0)THEN
      KT=KM-LGAP
      KG=KK-LGAP*IH
      DO 32 J=1,JMAX
      KT=KT+1
      S(J)=S(J)-G(KG)*FM(KT)
   32 KG=KG+1
      ENDIF
      IF(LC2.NE.0)THEN
      KT=KM+LGAP1
      KG=KK
      DO 33 J=1,JMAX
      KT=KT+1
      S(J )=S(J)-G(KG)*FM(KT)
   33 KG=KG+1
      ENDIF
C
C  HEXAGONAL GEOMETRY ONLY
C
      IF(LC1H.NE.0) THEN
      KT=KM-LGAP+JMAX
      KG=KK-LGAP*IH     +JMAX*IH
      DO 49 J=1,JMAX
      S(J)=S(J)-F(KG)*FM(KT)
      KT=KT+1
   49 KG=KG+1
      ENDIF
      IF(LC2H.NE.0)THEN
      KT=KM+LGAP1-JMAX
      KG=KK
      DO 50 J=1,JMAX
      S(J )=S(J)-F(KG)*FM(KT)
      KT=KT+1
   50 KG=KG+1
      ENDIF
C
C   END HEXAGONAL GEOMETRY ONLY
C
      S(1)=S(1)/E(KK)
      DO 34 J=2,JMAX
      KK=KK+1
   34 S(J)=(S(J)-C(KK)*S(J-1))/E(KK)
      KA=KM+1
      KM=KM+JMAX
      FM(KM)=EXBG*(S(JMAX)-FM(KM))+FM(KM)
      FSQ2=FSQ2+FM(KM)*FM(KM)
      JJ=JMAX
      DO 35 J=2,JMAX
```

```
      KM=KM-1
      KK=KK-1
      JJ=JJ-1
      S(JJ)=S(JJ)-D(KK)*S(JJ+1)
   35 FM(KM)=FM(KM)+EXBG*(S(JJ)-FM(KM))
      KM=KM-1+JMAX
      KK=KK+IH*JMAX
  36  CONTINUE
  42  CONTINUE
  43  CONTINUE
      FSQ2=0.
      KK=(LPL-1)*NPTS
      DO 44 K=1,NPTS
      KK=KK+1
  44  FSQ2=FSQ2+FM(KK)*FM(KK)
C
C  CHECK INNER ITERATION CONVERGENCE
C
 233  FORMAT(1X,' FSQ1 ',2E14.6)
      IF(NITER.EQ.1) GO TO 29
      ABSF=ABS(1.-FSQ1/FSQ2)
C     IF(NIT.LE.5.AND.ABSF.LE.ACC2) GO TO 54
      IF(ABSF.LE.ACC1) GO TO 54
      IF(NITER.LE.100) GO TO 29
      WRITE(6,121) N
  121 FORMAT(1X,' LACK OF CONVERGENCE IN GROUP  ',I6)
  54  DO 46 K=1,NPTS
  46  Q(K)=0.
  65  CONTINUE
  40  CALL FILEW(NT5   ,FM,NPTS*2*LPN)
C
C END OF LOOP OVER GROUPS
C
      NTT=NT4
      NT4=NT5
      NT5=NTT
      GO TO 21
      END
```

```
      SUBROUTINE QMESHA(A,B,C,       G,FS,      FM,    IBL,IBR,ICA,IXS,LFS,
     1X,Y,Z,     QS,SMY,SX,JM,BB)
      REAL*8 A,B,C,G
      REAL*8 SX,FM,QS,SMY,SXD,SDY,SMZ,SMZ1,SMZ2,DX,DY,FS
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC,
     4EIG,IMA,MESHQ,NT8,NT9
      DIMENSION A(1),B(1),C(1),G(1),                    FM(1),
     1IBL(1),IBR(1),ICA(1),IXS(1),FS(1),                LFS(1),X(1),Y(1)
     2,Z(1),SMY(JM,1),SX(1),QS(1),BB(1)
      COMMON /STORE/CH(2)
      DIMENSION ICH(1)
      EQUIVALENCE (CH,ICH)
C
C     WORK OUT SECOND ORDER CORRECTION USING A SMALL AMOUNT OF EXTRA
C     MEMORY
C
C
C  DERIVE SECOND DERIVATIVES OF THE FLUX AND FORM LEAKAGE TERM
C
      REWIND NT3
      REWIND NT4
      REWIND NT5
      REWIND NT6
      REWIND NT7
      REWIND NT8
      R3=SQRT(3.)
      TD=2./3.
      TF=1./24.
      IF(NAMGM.EQ.2) TF=5./144.
      VF=1.
      IF(NAMGM.EQ.2) VF=SQRT(3.)*.25
      DO 5 N=1,NG
      CALL FILER(NT4,FM,LEN)
      CALL FILER(NT3,B,LEN)
      CALL FILER(NT3,B,LEN)
      KN=-NPTS
      DO 16 K=1,NPTS
      KN=KN+1
      FM(KN)=0.
   16 QS(K)=0.
      DO 13 I=1,IMAX
      DO 13 J=1,JMAX
   13 SMY(J,I)=0.
      KM=1
      IL=IBL(1)
      ILN=IBL(1)
      IRN=IBR(1)
      KK=1
      KN=KM-NPTS
      IU=0
      IF(BC(1).EQ.0.) THEN
      IIC=(IRN-IL)/2
      IF(IIC.NE.2*(IIC/2)) IU=1
      ENDIF
      DO 3 MM=1,KMAX
      ILL=IL
      IRL=IR
      IL=ILN
      IR=IRN
      IF(MM.NE.KMAX) THEN
```

```
      ILN=IBL(MM+1)
      IRN=IBR(MM+1)
      ENDIF
C
C           IU IS 1 IF FIRST TRIANGLE IN THE ROW POINTS DOWN
C
      IU=1-IU
      IF(ILL.GT.IL) IU=1
      IF(ILN.GT.IL) IU=0
      IUU=IU
      LOCPT=MM*IMAX+IL
      LOCP=LOCPT-IMAX
      LOCPU=LOCP-IMAX
      NCH=ICA(LOCP)
      LGAP=(IR-ILN+1)*JMAX
      LGAP1=(IRL-IL+1)*JMAX
      DO 15 J=1,JMAX
 15   SX(J)=0.
      IXB=3
      DO 2 I=IL,IR
      IUL=1-IUU
      NCH=ICA(LOCP)
      LOCA=LCC+NCH-1
      LOCA=ICH(LOCA)
      NCHU=0
      IF(MM.NE.1) NCHU=ICA(LOCPU)
      NCHT=0
      IF(MM.NE.KMAX)NCHT=ICA(LOCPT)
      IF(NCHT.NE.0) THEN
      LOCT=LCC+NCHT-1
      LOCT=ICH(LOCT)
      ENDIF
      IF(I.NE.IR) THEN
      NCHR=ICA(LOCP+1)
      LOCR=LCC+NCHR-1
      LOCR=ICH(LOCR)
      ENDIF
      IF(NAMGM.EQ.2) THEN
      LCP2=0
      LCM2=0
      IF(IUU.EQ.1) THEN
      IF(MM.NE.KMAX.AND.ICA(LOCPT+2).NE.0.AND.I+2.LE.IMAX)
     1LCP2=ICA(LOCPT+2)+LCC-1
      IF(MM.NE.KMAX.AND.ICA(LOCPT-2).NE.0.AND.I.GT.2)
     1 LCM2=ICA(LOCPT-2)+LCC-1
      KMF=KM+LGAP
      ELSE
      IF(MM.NE.1.AND.ICA(LOCPU+2).NE.0.AND.I+2.LE.IMAX)
     1 LCP2=ICA(LOCPU+2)+LCC-1
      IF(MM.NE.1.AND.ICA(LOCPU-2).NE.0.AND.I.GT.2)
     1 LCM2=ICA(LOCPU-2)+LCC-1
      KMF=KM-LGAP1
      ENDIF
      IF(LCP2.NE.0)LCP2=ICH(LCP2)
      IF(LCM2.NE.0)LCM2=ICH(LCM2)
      ENDIF
      LOCP=LOCP+1
      LOCPT=LOCPT+1
      LOCPU=LOCPU+1
      KM1=KM+JMAX
      KM2=KM+LGAP
      IF(NAMGM.NE.2) THEN
      IF(I.EQ.IL.OR.I.EQ.IR)
     1CALL EDGE(FM(KM),SX,X(I),IXS,LOCA,-BC(IXB),N)
```

```
      IXB=4
      IYB=1
      IF(NCHT.EQ.0) IYB=2
      IF( NCHU.EQ.0.OR.NCHT.EQ.0)
     1CALL EDGE(FM(KM),SMY(1,I),Y(MM),IXS,LOCA,-BC(IYB),N)
      ENDIF
      NZ=ICH(LOCA)
      LOXN=IXS(NZ)+N-1
C
C LOOP OVER Z DIRECTION
C
C  RECTANGULAR GEOMETRY
C
      IF(NAMGM.EQ.1) THEN
      DO 1 J=1,JMAX
      LOX=LOXN
      IF(J.EQ.1) SMZ1=-3.*BC(5)/(1.+1.5*Z(J)*BC(5)*CH(LOX))*FM(KM)
      IF(J.EQ.JMAX) SMZ2=-3.*FM(KM)*BC(6)/(1.+1.5*Z(J)*BC(6)*CH(LOX))
      LOCA=LOCA+1
      IF(J.NE.JMAX) THEN
      NZ=ICH(LOCA)
      LOXN=IXS(NZ)+N-1
      SMZ2=(FM(KM+1)-FM(KM))*2./(Z(J)*CH(LOX)+Z(J+1)*CH(LOXN))
      ENDIF
      SMZ=(SMZ2+SMZ1)*CH(LOX)*Z(J)
      FM(KN)=FM(KN)+SMZ
      IF(J.NE.JMAX) THEN
      SMZ1=-SMZ2
      QS(KM)=QS(KM)-SMZ*C(KK+1)
      QS(KM+1)=QS(KM+1)+C(KK+1)*SMZ
      ENDIF
      IF(J.GT.1) THEN
      QS(KM-1)=QS(KM-1)+SMZ*C(KK)
      QS(KM)=QS(KM)-SMZ*C(KK)
      ENDIF
      SXD=SX(J)
      IF(I.NE.IR) THEN
      NZ=ICH(LOCR)
      LOCXR=IXS(NZ)+N-1
      DX=(FM(KM1)-FM(KM))*2./(CH(LOX)*X(I)+CH(LOCXR)*X(I+1))
      SXD=(SX(J)+DX)*X(I)*CH(LOX)
      SX(J)=-DX
      QS(KM)=QS(KM)-B(KK)*SXD
      QS(KM1)=QS(KM1)+SXD*B(KK)
      LOCR=LOCR+1
      ENDIF
      IF(I.GT.IL)THEN
      QS(KM)=QS(KM)-A(KK)*SXD
      KM3=KM-JMAX
      QS(KM3)=QS(KM3)+A(KK)*SXD
      ENDIF
      FM(KN)=FM(KN)+SXD
      SDY=SMY(J,I)
      IF(NCHT.NE.0) THEN
      NZ=ICH(LOCT)
      LOCXT=IXS(NZ)+N-1
      DY=(FM(KM2)-FM(KM))*2./(CH(LOX)*Y(MM)+CH(LOCXT)*Y(MM+1))
      SDY=(SMY(J,I)+DY)*Y(MM)*CH(LOX)
      SMY(J,I)=-DY
      QS(KM)=QS(KM)-G(KK)*SDY
      QS(KM2)=QS(KM2)+G(KK)*SDY
      LOCT=LOCT+1
      ENDIF
      IF(NCHU.NE.0) THEN
```

```
      KM4=KM-LGAP1
      KKK=KK-LGAP1*IH
      QS(KM4)=QS(KM4)+G(KKK)*SDY
      QS(KM)=QS(KM)-SDY*G(KKK)
      ENDIF
      FM(KN)=FM(KN)+SDY
      IF(J.EQ.1) QS(KM)=QS(KM)+BC(5)/(1.+1.5*Z(1)*BC(5)*CH(LOX))
     1*(SMZ-(SDY+SXD)/3.)*X(I)*Y(MM)
      IF(J.EQ.JMAX) QS(KM)=QS(KM)+BC(6)*X(I)*Y(MM)*(SMZ-(SDY+SXD)/3.)/
     1(1.+1.5*Z(JMAX)*BC(6)*CH(LOX))
      IF(I.EQ.IL) QS(KM)=QS(KM)+BC(3)*Y(MM)*Z(J)*(SXD-(SMZ+SDY)/3.)/
     1(1.+1.5*X(I)*BC(3)*CH(LOX))
      IF(I.EQ.IR) QS(KM)=QS(KM)+BC(4)*Y(MM)*Z(J)*(SXD-(SMZ+SDY)/3.)/
     1(1.+1.5*X(I)*BC(4)*CH(LOX))
      IF(NCHU.EQ.0) QS(KM)=QS(KM)+BC(1)*X(I)*Z(J)*(SDY-(SMZ+SXD)/3.)/
     1(1.+1.5*Y(MM)*BC(1)*CH(LOX))
      IF(NCHT.EQ.0) QS(KM)=QS(KM)+BC(2)*X(I)*Z(J)*(SDY-(SMZ+SXD)/3.)/
     1(1.+1.5*Y(MM)*BC(2)*CH(LOX))
      KM1=KM1+1
      KM2=KM2+1
      KK=KK+1
      KN=KN+1
    1 KM=KM+1
C
C   TRIANGULAR GEOMETRY
C
      ELSEIF(NAMGM.EQ.2) THEN
      KKK=KK-LGAP1*IH*IUL
      KM2=KM+LGAP
      XUU=1+IUU
      XIU=1+IUL
      BC3=BC(3)
      BC4=BC(4)
      IF(IUU.EQ.1.AND.I.GT.IL+1)BC3=BC(2)
      IF(IUL.EQ.1.AND.I.GT.IL+1)BC3=BC(1)
      IF(IUU.EQ.1.AND.I.LT.IR-1)BC4=BC(2)
      IF(IUL.EQ.1.AND.I.LT.IR-1)BC4=BC(1)
      DO 17 J=1,JMAX
      LOX=LOXN
      IF(J.EQ.1)SMZ1=-3.*BC(5)*FM(KM)/(1.+1.5*Z(J)*BC(5)*CH(LOX))
      IF(J.EQ.JMAX)SMZ2=-3.*BC(6)*FM(KM)/(1.+1.5*Z(J)*BC(6)*CH(LOX))
      LOCA=LOCA+1
      IF(J.NE.JMAX) THEN
      NZ=ICH(LOCA)
      LOXN=IXS(NZ)+N-1
      SMZ2=(FM(KM+1)-FM(KM))*2./(Z(J)*CH(LOX)+Z(J+1)*CH(LOXN))
      ENDIF
      SMZ=(SMZ2+SMZ1)*Z(J)*CH(LOX)
      FM(KN)=FM(KN)+SMZ*1.2
      IF(J.NE.JMAX) THEN
      SMZ1=-SMZ2
      QS(KM)=QS(KM)-SMZ*C(KK+1)
      QS(KM+1)=QS(KM+1)+C(KK+1)*SMZ
      ENDIF
      IF(J.GT.1) THEN
      QS(KM-1)=QS(KM-1)+SMZ*C(KK)
      QS(KM)=QS(KM)-SMZ*C(KK)
      ENDIF
      IF(LCM2.NE.0) THEN
      NZ=ICH(LCM2)
      LOCXM=IXS(NZ)+N-1
      KM3=KMF-2*JMAX
      DXM=(FM(KM3)-FM(KM))/(CH(LOX)*X(I)+CH(LOCXM)*X(I-2))
      LCM2=LCM2+1
```

```
      ENDIF
      IF(LCM2.EQ.0)DXM=-3.*BC3*FM(KM)/(R3+3.0*BC3*CH(LOX)*X(I))
      IF(LCP2.NE.0) THEN
      NZ=ICH(LCP2)
      LOCXM=IXS(NZ)+N-1
      KM3=KMF+2*JMAX
      DXP=(FM(KM3)-FM(KM))/(CH(LOX)*X(I)+CH(LOCXM)*X(I+2))
      LCP2=LCP2+1
      ENDIF
      IF(LCP2.EQ.0)DXP=-3.*BC4*FM(KM)/(R3+3.0*BC4*CH(LOX)*X(I))
      IF(I.EQ.IL) SX(J)=-3.*BC(3)*FM(KM)/(R3+1.5*BC(3)*CH(LOX)*X(I))
      SU=X(I)*CH(LOX)*TD*(SX(J)+DXP)
      IF(I.GT.IL)THEN
      QS(KM)=QS(KM)-A(KK)*SU
      KM3=KM-JMAX
      QS(KM3)=QS(KM3)+A(KK)*SU
      ENDIF
      FM(KN)=FM(KN)+SU
      IF(I.EQ.IR) DX=-3.*BC(4)*FM(KM)/(R3+1.5*BC(4)*CH(LOX)*X(I))
      IF(I.NE.IR) THEN
      NZ=ICH(LOCR)
      LOCXR=IXS(NZ)+N-1
      DX=(FM(KM1)-FM(KM))*2./(CH(LOX)*X(I)+CH(LOCXR)*X(I+1))
      SX(J)=-DX
      LOCR=LOCR+1
      ENDIF
      SS=X(I)*CH(LOX)*(DXM+DX)*TD
      QS(KM)=QS(KM)-B(KK)*SS
      QS(KM1)=QS(KM1)+SS*B(KK)
      FM(KN)=FM(KN)+SS
      DY2=SMY(J,I)
      IF(NCHT.NE.0) THEN
      NZ=ICH(LOCT)
      LOCXT=IXS(NZ)+N-1
      DY1=(FM(KM2)-FM(KM))*XUU/(CH(LOX)*Y(MM)+CH(LOCXT)*Y(MM+1))
      SMY(J,I)=-DY1
      LOCT=LOCT+1
      ENDIF
      IF(NCHT.EQ.0)DY1=-3.*BC(2)*FM(KM)/(R3+1.5*XIU*BC(2)*CH(LOX)*Y(MM))
      IF(NCHU.EQ.0)DY2=-3.*BC(1)*FM(KM)/(R3+1.5*XUU*BC(1)*CH(LOX)*Y(MM))
      ST=(DY1+DY2)*TD*CH(LOX)*Y(MM)
      IF(NCHT.NE.0.AND.IUU.NE.0) THEN
      QS(KM)=QS(KM)-G(KKK)*ST
      QS(KMF)=QS(KMF)+G(KKK)*ST
      ENDIF
      IF( NCHU.NE.0.AND.IUU.EQ.0) THEN
      QS(KMF)=QS(KMF)+G(KKK)*ST
      QS(KM)=QS(KM)-ST*G(KKK)
      ENDIF
      FM(KN)=FM(KN)+ST
      IF(J.EQ.1) QS(KM)=QS(KM)+BC(5)/(1.+1.5*Z(1)*BC(5)*CH(LOX))
     1*(SMZ-(SS+ST+SU)*TF)*X(I)*Y(MM)*VF
      IF(J.EQ.JMAX)QS(KM)=QS(KM)+BC(6)/(1.+1.5*Z(JMAX)*BC(6)*CH(LOX))
     1*(SMZ-(SS+ST+SU)*TF)*X(I)*Y(MM)*VF
      IF(I.EQ.IL)QS(KM)=QS(KM)+BC(3)/(R3+1.5*X(I)*BC(3)*CH(LOX))
     1*(SU*4.-SMZ-(ST+SS)*2.)*Z(J)*Y(MM)/R3
      IF(I.EQ.IR)QS(KM)=QS(KM)+BC(4)/(R3+1.5*X(I)*BC(4)*CH(LOX))
     1*(SS*4.-SMZ-(ST+SU)*2.)*Z(J)*Y(MM)/R3
      IF(NCHU.EQ.0) QS(KM)=QS(KM)+1./(R3+1.5*Y(MM)*BC(1)*CH(LOX))
     1*(ST*4.-SMZ-(SS+SU)*2.)*Z(J)*X(I)*BC(1)/R3
      IF(NCHT.EQ.0) QS(KM)=QS(KM)+1./(R3+1.5*Y(MM)*BC(2)*CH(LOX))
     1*(ST*4.-SMZ-(SS+SU)*2.)*Z(J)*X(I)*BC(2)/R3
      KM1=KM1+1
      KM2=KM2+1
```

```
      KMF=KMF+1
      KKK=KKK+1
      KK=KK+1
      KN=KN+1
   17 KM=KM+1
      IUU=1-IUU
      ENDIF
C
C   END OF AXIAL LOOP
C
      KK=KK+(IH-1)*JMAX
    2 CONTINUE
    3 CONTINUE
      KN=-NPTS
      DO 4 K=1,NPTS
      QS(K)=QS(K)*0.125
      KN=KN+1
    4 FM(KN)=FM(KN)*TF
      CALL FILEW(NT7,QS,NPTS*2)
      KN=1-NPTS
    5 CALL FILEW(NT8,FM(KN),2*NPTS)
C
C    SECOND DERIVATIVE CALCULATION COMPLETED
C
      REWIND NT2
      REWIND NT3
      REWIND NT6
      REWIND NT7
      REWIND NT8
C
C   CALCULATE TOTAL FISSION SOURCE
C
      DO 6 N=1,NG
      CALL FILER(NT2,B,LEN)
      CALL FILER(NT8,FM,LEN)
      DO 6 K=1,NPTS
    6 FS(K)=FS(K)+B(K)*FM(K)
      REWIND NT8
      DO 12 N=1,NG
C
C  FORM MESH SOURCE IN QS BEGINNING WITH GROUP FISSION CONTRIBUTION
C ADDED TO EXISTING LEAKAGE TERMS
C
      CALL FILER(NT7,QS,LEN)
      L=0
      DO 7 K=1,NPTS
      LL=L
      L=LFS(K)
      IF(L.NE.LL) ISP=L+N
    7 QS(K)=FS(K)*CH(ISP)/EIG+QS(K)
C
C   SCATTER CONTRIBUTION
C
      CALL FILER(NT3,BB,LEN)
      NDSCA=MIN0(N-1,NDSCAT)
      IF(NDSCA.NE.0) THEN
      KK=0
      KS=(1-NDSCA)*NPTS
      DO 8 NN=1,NDSCA
      DO 8 K=1,NPTS
      KK=KK+1
      KS=KS+1
    8 QS(K)=QS(K)+BB(KK)*FM(KS)
      NDS=MIN0(NDSCAT,N)-1
```

```
      KK=1-NDS*NPTS
      KS=KK+NPTS
      CALL EQVEC(FM(KS),FM(KK),NDS*NPTS*2)
      ENDIF
      CALL FILER(NT8,FM,LEN)
      CALL FILER(NT3,B,LEN)
C
C   REMOVAL TERM
C
      KM=1
      DO 11 MM=1,KMAX
      IL=IBL(MM)
      IR=IBR(MM)
      LOCP =MM*IMAX+IL-IMAX
      DO 10 I=IL,IR
      AZ=VF*X(I)*Y(MM)
      NCH=ICA(LOCP)
      LOCA=LCC+NCH-1
      LOCA=ICH(LOCA)
      LOCP=LOCP+1
      DO 9 J=1,JMAX
      NZ=ICH(LOCA)
      LOCA=LOCA+1
      LTR=IXS(NZ)+NG+N-1
      VOL=AZ*Z(J)
      QS(KM)=QS(KM)-VOL*CH(LTR)*FM(KM)
    9 KM=KM+1
   10 CONTINUE
   11 CONTINUE
   12 CALL FILEW(NT6,QS,NPTS*2)
      DO 14 K=1,NPTS
   14 FS(K)=0.
      RETURN
      END
      SUBROUTINE EDGE(F,SM,X,IXS,LOC,B,N)
      REAL*8 SM,F
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH
      COMMON /STORE/CH(1)
      DIMENSION F(1),SM(1),IXS(1),ICH(1),X(1),B(1)
      EQUIVALENCE (CH(1),ICH(1))
      LOCA=LOC
      XM=X(1)
      BB=ABS(B(1))
      BD=B(1)
      DO 1 J=1,JMAX
      NZ=ICH(LOCA)
      LOCX=IXS(NZ)+N-1
      SM(J)=SM(J)+3.*BD*F(J)/(1.+1.5*XM*BB*CH(LOCX))
    1 LOCA=LOCA+1
      RETURN
      END
      SUBROUTINE DXYZ(B,C,G,FM,Q,ICD,DD,DDC,IBL,IBR,BM,S,JM,M)
      REAL*8 FM,Q,S,BM,DA,DB,SSS
      REAL*8 B,C,G
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF,MESHC
     4,EIG,IMA,MESHQ,NT8,NT9,NT10
       COMMON /STORE/CH(1)
      DIMENSION B(1),FM(1),ICD(1),DD(1),IBL(1),IBR(1),
```

```
     1BM(JM,1),S(1),DDC(3),IRM(3,4),Q(1),C(1),G(1),INZ(100),SR(100)
      DATA IRM/1,-1,-1,-1,1,-1,1,1,1,-1,-1,1/
C
C  EVALUATE OTHER MOMENT CONTRIBUTIONS FOR PN OPTION
C
      MM=3
      IF(M.NE.1) MM=M-4*((M-2)/4)-1
      ID1=ICD(1)
      D1=-DD(1)/DDC(1)
      D2=-DD(2)/DDC(2)
      Y1=-1.
      Y2=-1.
      Y3=-1.
      Y4=-1.
      Y5=-1.
      Y6=-1.
      IF(BC(5).EQ.0.)Y1=IRM(3,MM)
      IF(BC(1).EQ.0.)Y2=IRM(1,MM)
      IF(BC(3).EQ.0.)Y3=IRM(2,MM)
      IF(ID1.NE.1) GO TO 7
C
C    D2/DX2 TYPE TERMS
C
      D3=-DD(3)/DDC(3)
      KL=1
      KK=1
      KK1=KK+JMAX
      DO 6 MM=1,KMAX
      IL=IBL(MM)
      IR=IBR(MM)
      IF(MM.NE.KMAX) LGAP=(IR-IBL(MM+1)+1)*JMAX
      KK2=KK+LGAP
      IF(Y3.LT.0.) CALL PEDGE(B(KL),FM(KK),Q(KK),JMAX,D1)
      DO 5 I=IL,IR
      IF(MM.EQ.1.AND.Y2.LT.0.)
     1CALL PEDGE(G(KL),FM(KK),Q(KK),JMAX,D2)
      KL1=KL-LGAP*IH
      IF(MM.EQ.KMAX.AND.Y5.LT.0.)
     1CALL PEDGE(G(KL1),FM(KK),Q(KK),JMAX,D2)
      IF(Y1.LT.0.) Q(KK)=Q(KK)-2.*C(KL+1)*FM(KK)*D3
      DO 4 J=1,JMAX
      DZ=C(KL)*(FM(KK-1)-FM(KK))*D3
      DY=G(KL)*(FM(KK2)-FM(KK))*D2
      DX=B(KL)*(FM(KK1)-FM(KK))*D1
      Q(KK)=Q(KK)+DX+DY+DZ
      Q(KK-1)=Q(KK-1)-DZ
      Q(KK1)=Q(KK1)-DX
      Q(KK2)=Q(KK2)-DY
      KL=KL+1
      KK1=KK1+1
      KK2=KK2+1
  4   KK=KK+1
      IF(Y4.LT.0.) Q(KK-1)=Q(KK-1)-2.*C(KL-1)*FM(KK-1)*D3
    5 KL=KL+(IH-1)*JMAX
      KL1=KL-IH*JMAX*2
      KKL=KK-JMAX
      IF(Y6.LT.0.) CALL PEDGE(B(KL1),FM(KKL),Q(KKL),JMAX,D1)
    6 CONTINUE
      RETURN
C
C   D2/DXDY ETC        I.E. CROSS TERMS
C
    7 DO 2 J=1,JMAX
      SR(J)=1.
```

```
2     INZ(J)=1
      IF(BC(6).NE.0.) SR(JMAX)=-1.
      INZ(JMAX)=0
      K1=1
      K2=1
      IF(ID1.EQ.6) K2=4*JMAX+1
      IF(ID1.GE.3) D2=-DD(1)/DDC(3)
      IF(ID1.EQ.6) D1=-DD(1)/DDC(2)
      L1=1
      LOYM=IPK
      YM2=CH(LOYM)
      YM3=CH(LOYM)
      LGAP=0
      DO 9 MM=1,KMAX
      YY=1.
      IF(MM.EQ.KMAX)YY=-1.
      YM1=YM2
      YM2=YM3
      LOYM=LOYM+1
      IF(MM.NE.KMAX) YM3=CH(LOYM)
      LOXM=IPI
      XM2=CH(LOXM)
      XM3=CH(LOXM)
      IL=IBL(MM)
      IR=IBR(MM)
      LGAPL=LGAP
      LGAP=0
      IF(MM.NE.KMAX)LGAP=(IR-IBL(MM+1)+1)*JMAX
      INC1=JMAX
      IF(ID1.EQ.6) INC1=LGAP
      DM=D1*(XM3+XM2)
      IF(ID1.EQ.2)DM=DM/(YM1+2.*YM2+YM3)
      Y2T=1.
      IF(MM.EQ.1) Y2T=Y2
      IF(ID1.EQ.2) CALL QEDGE(Q(L1),FM(L1),B(K1),S,DM,JMAX,LGAPL,LGAP,
     1YM2,YM3,Y3,Y2T)
      ID=IL
      IF(ID1.EQ.3) CALL REDGE(Q(L1),FM(L1),B(K1),BM(1,ID),DM,CH(IPJ),
     1JMAX,INZ,Y3,SR,Y1)
      DO 9 I=IL,IR
      XX=1.
      IF(I.EQ.IR)XX=-1.
      INC1L=INC1
      IF(ID1.NE.6.AND.I.EQ.IR) INC1=0
      XM1=XM2
      XM2=XM3
      LOXM=LOXM+1
      IF(I.NE.IR) XM3=CH(LOXM)
      INCL=JMAX
      IF(I.EQ.IL)INCL=0
      Y3T=1.
      IF(I.EQ.IL) Y3T=Y3
C
C               D2/DXDY
C
      IF(ID1.EQ.2) THEN
      YM=1./(YM1+YM2*2.+YM3)
      XM=1./(XM1+2.*XM2+XM3)
      DM=D2*(YM2+YM3)
      DM=DM*XM
      IF(MM.EQ.1) CALL QEDGE(Q(L1),FM(L1),G(K1),BM(1,I),DM,JMAX,INCL,
     1INC1,XM2,XM3,Y2,Y3T)
      DO 1 J=1,JMAX
      L2=L1+INC1
```

```
      L4=L2+LGAP
      L3=L1+LGAP
      DA= D2*G(K1)*((FM(L2)*YM3+FM(L4)*YM2)*XX-S(J))*XM
      DB= D1*B(K1)*((FM(L3)*XM3+FM(L4)*XM2)*YY-BM(J,I))*YM
      BM(J,I)= FM(L1)*XM3+FM(L2)*XM2
      S(J) = FM(L1)*YM3+FM(L3)*YM2
      Q(L1)=Q(L1)+DA+DB
      Q(L3)=Q(L3)-DA
      Q(L2)=Q(L2)-DB
      L1=L1+1
    1 K1=K1+1
      ELSE
C
C             D2/DXDZ AND D2/DYDZ
C
      XX=1.
      DM=D1*(YM2+YM3)
      IF(MM.EQ.1.AND.ID1.EQ.6) CALL REDGE(Q(L1),FM(L1),G(K1),BM(1,I),
     1DM,CH(IPJ),JMAX,INZ,Y2,SR,Y1)
      LOZM=IPJ
      ZM2=CH(LOZM)
      ZM3=CH(LOZM+1)
      IF(ID1.EQ.3) THEN
      XY2= XM2
      XY3= XM3
      XYD=1./(XM1+2.*XM2+XM3)
      IF(I.EQ.IR) XX=-1.
      ELSE
      XY2=YM2
      XY3=YM3
      XYD=1./(YM1+2.*YM2+YM3)
      IF(MM.EQ.KMAX)XX=-1.
      ENDIF
      L2=L1+INC1
      SSS=(FM(L1)*XY3+FM(L2)*XY2*XX)*Y1
      L3=L1
      IF(MM.NE.1.AND.ID1.EQ.6) L3=L1-LGAPL
      IF(I.NE.ID.AND.ID1.EQ.3) L3=L1-INC1L
      IF(Y1.GT.0.)
     1Q(L1)=Q(L1)-D2*C(K1+1)*(FM(L2)-FM(L3))*(ZM2+ZM3)*XYD
      DO 8 J=1,JMAX
      L3=L1+INZ(J)
      L4=L3+INC1
      ZM1=ZM2
      ZM2=ZM3
      LOZM=LOZM+INZ(J)
      ZM3=CH(LOZM)
      ZD=1./(ZM1+ZM2*2.+ZM3)
      DA= D2*C(K1)*((FM(L2)*ZM1+FM(L2-1)*ZM2)*XX-BM(J,ID))*XYD
      DB= D1*B(K2)*((FM(L3)*XY3+FM(L4)*XY2*XX)*SR(J)-SSS)*ZD
      BM(J,ID)= FM(L1)*ZM1+FM(L1-1)*ZM2
      SSS = FM(L1)*XY3+FM(L2)*XY2*XX
      Q(L1)=Q(L1)-DA+DB
      Q(L1-1)=Q(L1-1)+DA
      Q(L2)=Q(L2)-DB
      L2=L2+1
      L1=L1+1
      K1=K1+1
    8 K2=K2+1
      IF(ID1.EQ.6) ID=ID+1
      ENDIF
      K1=K1+(IH-1)*JMAX
      K2=K2+(IH-1)*JMAX
    9 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE PEDGE(B,F,Q,JM,DY)
      REAL*8 F,Q
      DIMENSION B(1),F(1),Q(1)
C
C REFLECTIVE BOUNDARY CONTRIBUTION IF MOMENT NOT EVEN PARITY
C
      DO 1 J=1,JM
 1    Q(J)=Q(J)-DY*2.*F(J)*B(J)
      RETURN
      END
      SUBROUTINE QEDGE(Q,FM,B,S,DD,JM,INC1,INC2,YM2,YM3,Y1,Y2)
      REAL*8 Q,FM,S
      REAL*8 B
      DIMENSION Q(1),FM(1),B(1),S(1)
C
C           D2/DXDY  REFLECTIVE BOUNDARY TERMS
      DO 1 J=1,JM
      L1=J+INC2
  1   S(J)=(FM(J)*YM3+FM(L1)*YM2)*Y1
      IF(Y1.LT.0.) RETURN
      DO 2 J=1,JM
      L1=J+INC2
      L2=J-INC1
  2   Q(J)=Q(J)-DD*(FM(L1)-FM(L2)*Y2)*B(J)
      RETURN
      END
      SUBROUTINE REDGE(Q,FM,B,S,DD,ZM,JM,INZ,Y1,SR,YZ)
      REAL*8 Q,FM,S
      REAL*8 B
      DIMENSION Q(1),FM(1),B(1),S(1),ZM(1),INZ(1),SR(1)
C
C         D2/DXDZ AND D2/DYDZ   REFLECTIVE BOUNDARY TERMS
C
      ZM2=ZM(1)
      SS=FM(1)*YZ
      DO 1 J=1,JM
      ZM1=ZM2
      ZM2=ZM(J)
      S(J)=(FM(J)*ZM1+SS*ZM2)*Y1
  1   SS=FM(J)
      IF(Y1.LT.0.) RETURN
      ZM2=ZM(1)
      ZM3=ZM(1)
      SS=FM(1)*YZ
      DO 2 J=1,JM
      ZM1=ZM2
      ZM2=ZM3
      L1=J+INZ(J)
      ZM3=ZM(L1)
      ZZ=1./(ZM1+2*ZM2+ZM3)
      Q(J)=Q(J)-DD*(FM(L1)*SR(J)-SS)*B(J)*ZZ
  2   SS=FM(J)
      RETURN
      END
      SUBROUTINE FILER(N,A,L)
      DIMENSION A(1)
      LREC=500000
      READ(N) NT,L,LL,(A(K),K=1,LL)
      IF(NT.EQ.0) RETURN
      L2=LL
      DO 1 J=1,NT
      L1=L2+1
```

```
      L2=MIN0(L,L2+LREC)
    1 READ (N) NN,L,LL,(A(K),K=L1,L2)
      RETURN
      END
      SUBROUTINE FILEW(N,A,L)
      DIMENSION A(1)
      LREC=500000
      NT=L/LREC
      LL=L-NT*LREC
      L2=0
      IF(NT.GT.0) THEN
      DO 1 J=1,NT
      L1=L2+1
      L2=L2+LREC
    1 WRITE(N) NT,L,LREC,(A(K),K=L1,L2)
      ENDIF
      L1=L2+1
      LL=L-L2
      WRITE(N) NT,L,LL,(A(K),K=L1,L)
      RETURN
      END
      SUBROUTINE PROFIL(FM,IBL,IBR)
      CHARACTER SC,ITDEC,AI(3)
      REAL*8 FM
      COMMON  /BASIC/NG,NDSCAT,NUPSCAT,IMAX,KMAX,JMAX,IPI,IPK,IPJ,
     1LCC,NMS,NZS,IPIS,IPMS,IPZS,IPLB,IPRB,NPTS,ITOP,MAXC,NERR,
     2LOCCA,LOCI,NXSM,MAXI,NT1,NT2,NT3,NT4,NT5,ACC1,ACC2,BC(6),
     3NAMGM,NPN,LOCD,LOCC,LOCF,LPN,LOCIC,IH,NT6,NT7,MESHF
      COMMON /STORE/CH(1)
      DATA AI/'X','Y','Z'/
      DIMENSION IP(3),FM(1),IPO(2),IBL(1),IBR(1)
C     WRITE(6,101)
C 101 FORMAT(1X,' FLUX PROFILE OUTPUT')
      MM=0
      DO 1 M=1,3
      IP(M)=0
      IF(NEWDEC(0).EQ.0) THEN
      SC=ITDEC(1)
       IAX=M
      GO TO 1
      ENDIF
      IP(M)=INDEC(0)
      IP(M)=(IP(M)-1)*MESHF+1
      MM=MM+1
      IPO(MM)=IP(M)
    1 CONTINUE
      IF(NAMGM.EQ.2.AND.IP(1).NE.0) IP(1)=IP(1)-MESHF+1
C     WRITE (6,998) SC,(IP(M), M=1,3)
C998  FORMAT(1X,'IP ', A1,3I5)
      REWIND NT4
      DO 30 N=1,NG
      CALL FILER(NT4,FM,LR)
      K2=0
      DO 30 I=1,7
      K1=K2+1
      K2=K2+JMAX
 30   WRITE(6,31)I,(FM(K),K=K1,K2)
 31   FORMAT(1X,I3,/,(5E14.6))
      REWIND NT4
      IX=MAX0(1,IP(1))-1
      IY=MAX0(1,IP(2))-1
      IZ=MAX0(1,IP(3))-1
      LOCP=0
      IF(IY.GT.0) THEN
```

```
      DO 4 MM=1,IY
  4   LOCP=LOCP+(IBR(MM)-IBL(MM)+1)
      ENDIF
      INC=1
      IM=JMAX
      IF(IP(1).EQ.0) INC=JMAX
      IF(IP(1).EQ.0) IM=IMAX
      IF(IP(2).EQ.0) IM=KMAX
      DO 3 N=1,NG
      LOC=LOCP*JMAX+(IX-IBL(1)+1)*JMAX+IZ+1
      LOCH=NPTS
      LOCG=LOCH+1
      CALL FILER(NT4,FM,LEN)
C     WRITE(6,999)( FM(J),J=1,100)
CC999 FORMAT(5E14.6)
      DO 2 I=1,IM
      LOCH=LOCH+1
      FM(LOCH)= FM(LOC)
      IF(I.NE.IM.AND.IP(2).EQ.0)INC=(IBR(I)-IBL(I+1)+1)*JMAX
 2    LOC=LOC+INC
 3    WRITE(6,100)AI(IAX),IPO,N,(FM(L),L=LOCG,LOCH)
  100 FORMAT(1X,'FLUXES ALONG ',A1,'- AXIS AT ',2I4,' GROUP ',I3,
     1/,(5E14.6))
      RETURN
      END
```

```
      SUBROUTINE INITL
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      COMMON /ERR/NERR
      NERR=0
      LMAX=72
      NCOL=72
      NIN=5
      NOP=6
      RETURN
      END
      SUBROUTINE DECTP
      CHARACTER A(132)
      COMMON /DEC1/A
      COMMON /DEC2/NIN,NOP,LMAX
      COMMON /ERR/NERR
      READ (NIN,100,ERR=1)(A(J),J=1,LMAX)
100.  FORMAT(132A1)
      RETURN
 1    WRITE(NOP,102)
102   FORMAT(1X, 'NO DATA FOUND')
      STOP
      ENTRY WRITDP
      WRITE(NOP,101)(A(J),J=1,LMAX)
101   FORMAT(1X,132A1)
      RETURN
      END
      FUNCTION NEXCH(B,I,J,K)
      CHARACTER   A*40,B,C
      COMMON  /DEC1/C(132)
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      DATA A/'0123456789+-.ABCDEFGHIJKLMNOPQRSTUVWXYZ '/
1     NCOL=NCOL+1
      IF(NCOL.GT.LMAX) THEN
      CALL DECTP
      NCOL=1
      IPR=0
      ENDIF
      IF((NCOL.EQ.1.OR.IPR.EQ.0).AND.K.NE.0) THEN
      CALL WRITDP
      IPR=1
      ENDIF
      B=C(NCOL)
      DO 2 M=1,40
      IF(B.EQ.A(M:M)) GO TO 3
 2    CONTINUE
      M=41
  3   NEXCH=M
      IF(I.NE.0.AND.M.EQ.40) GO TO 1
      IF(J.EQ.0) NCOL=NCOL-1
      RETURN
      END
      SUBROUTINE GODEC
      CHARACTER A(132)
      CHARACTER AST*1
      COMMON /DEC1/A
      COMMON /DEC2/NIN,NOP,LMAX,NCOL
      COMMON /ERR/NERR
      DATA AST/'*'/
 1    READ (NIN,100)(A(J),J=1,LMAX)
 100  FORMAT(132A1)
 101  FORMAT(1X,132A1)
      IF(A(1).EQ.AST) WRITE(NOP ,101) (A(J),J=1,LMAX)
      IF(A(1).EQ.AST) GO TO 1
```

```
      XL=L-1
      Y=Y*Z
      D=D*X+XL*Y
 2    L=NEXCH(B,0,1,1)
      GO TO 1
 3    X=1.
      Z=0.1
      GO TO 2
 4    L=NEXCH(B,0,1,1)
      IF(L.EQ.11.OR.L.EQ.12) THEN
      IF(L.EQ.12)SE=-1.
      GO TO 4
      ENDIF
      IF(L.GT.10) GO TO 8
      XL=L-1
      F=10.*F+XL
      GO TO 4
 8    F=SE*F
      E=10.**F
 5    IF (IOP.EQ.1) DBLDEC=S*D*E
      IF (IOP.EQ.2)  FLDEC=S*D*E
      IF(N.GT.0.OR.L.EQ.40) GO TO 9
      WRITE(NOP,100) B
 100  FORMAT(1X,A1,' NUMBER SHOULD TERMINATE WITH BLANK IN MODE 0')
      NERR=NERR+1
 9    NCOL=NCOL-1
      RETURN
 6    WRITE(NOP,101) B
 101  FORMAT(1X,A1,' NUMBER DOES NOT BEGIN WITH ALLOWED SYMBOL')
      NERR =NERR+1
      NCOL=NCOL-1
      RETURN
      END
      SUBROUTINE JNDEX(J,WORD,S,N)
      CHARACTER AST,AMP,A,B,BL
      CHARACTER WORD(N),S(1)
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      DATA AST/'*'/,AMP/'&'/,BL/' '/
      L=0
      M=0
      K=1
      J=1
 1    L=L+1
      M=M+1
      A=WORD(M)
      B=S(L)
      IF(B.EQ.AST.OR.B.EQ.AMP) GO TO 2
      IF(M.EQ.N+1.OR.A.EQ.BL) GO TO 2
      IF(A.EQ.B) K=K+1
      GO TO 1
 2    IF(K.EQ.M.AND.(A.EQ.BL.OR.M.EQ.N+1).AND.
     1(B.EQ.AMP.OR.B.EQ.AST)) RETURN
 3    IF(B.EQ.AST) GO TO 4
      IF(B.EQ.AMP) GO TO 5
      L=L+1
      B=S(L)
      GO TO 3
 4    J=J+1
      M=0
      K=1
      GO TO 1
 5    J=J+1
      RETURN
      END
```

```
      NCOL=0
      RETURN
      END
      INTEGER FUNCTION NEWDEC(N)
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      M=NEXCH(B,1,0,0)
      L=-1
      IF(M.GT.40) L=0
      IF(M.LE.13) L=+1
      IF(N.EQ.1.AND.L.GE.11.AND.L.LE.13)L=0
      NEWDEC=L
      RETURN
      END
      INTEGER FUNCTION INDEC(N)
      COMMON /DEC2/NIN,NOP,LMAX,NCOL
      COMMON /ERR/NERR
      CHARACTER B
      M=0
      L=NEXCH(B,1,1,1)
      IS=1
      IF(L.GT.12) GO TO 3
      IF(L.LE.10) GO TO 1
      IF(L.EQ.12) IS=-1
      L=NEXCH(B,0,1,1)
    1 M=L+10*M-1
      L=NEXCH(B,0,1,1)
      IF(L.LE.10) GO TO 1
      IF(L.EQ.40) GO TO 2
      IF(L.GT.10.AND.N.EQ.1) GO TO 4
      WRITE(NOP,100) B
  100 FORMAT(1X,A1,'INTEGER MUST TERMINATE WITH BLANK IN MODE 0')
      NERR=NERR+1
    4 NCOL=NCOL-1
    2 INDEC=IS*M
      RETURN
    3 WRITE(NOP,101)B
  101 FORMAT(1X,A1,' INTEGER DOES NOT BEGIN WITH NUMBER OR SIGN')
      RETURN
      END
      DOUBLE PRECISION FUNCTION DBLDEC(N)
      CHARACTER B
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      COMMON /ERR/NERR
      IOP=1
      GO TO 7
      ENTRY FLDEC(N)
      IOP=2
    7 L=NEXCH(B,1,1,1)
      IF(L.GT.13) GO TO 5
      E=1.
      F=0.
      SE=1.
      S=1.
      D=0.
      X=10.
      Y=1.
      Z=1.
      IF(L.LE.10) GO TO 1
      IF(L.EQ.12) S=-1.
      IF(L.EQ.13) GO TO 3
      GO TO 2
    1 IF(L.EQ.18.OR.L.EQ.17) GO TO 4
      IF(L.EQ.13) GO TO 3
      IF(L.GT.10) GO TO 5
```

```
      SUBROUTINE WORDEC(SECT,N,L)
      CHARACTER SECT,BL,B
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      COMMON /ERR/NERR
      DIMENSION SECT(N)
      DATA BL/' '/
      J=0
      I=NEXCH(B,1.1,1)
      IF (I.LE.13) GO TO 6
  1   J=J+1
      IF(J.GT.N) GO TO 4
      SECT(J)=B
      I=NEXCH(B,0,1,1)
      IF(B.EQ.BL) GO TO 2
      IF((I.LE.13.OR.I.GT.40).AND.L.EQ.0) GO TO 7
      IF(I.GT.40) GO TO 2
      GO TO 1
  2   IF(J.EQ.N) RETURN
      J=J+1
      DO 3 M=J,N
  3   SECT(M)=BL
      RETURN
  4   WRITE (NOP,100)(SECT(J),J=1,N),B
  100 FORMAT(1X,'TOO MANY CHARACTERS WILL TRUNCATE',40A1)
      NERR=NERR+1
  5   L=NEXCH(B,0,1,1)
      IF(B.NE.BL) GO TO 5
      RETURN
  6   WRITE(NOP,101) B
 101  FORMAT(1X,A1,' WORD SHOULD BEGIN WITH A LETTER')
      NERR=NERR+1
      RETURN
C 7   WRITE(NOP,102) B
C102  FORMAT(1X,A1,' WORD SHOULD END WITH A BLANK IN MODE 0')
C     NERR=NERR+1
  7   NCOL=NCOL-1
      GO TO 2
      END
      CHARACTER FUNCTION ITDEC(N)
      CHARACTER B
      COMMON  /DEC2/NIN,NOP,LMAX,NCOL,IPR,IRD
      M=NEWDEC(0)
      K=N
      L=NEXCH(B,1,1,1)
      ITDEC=B
      IF(M.EQ.0) RETURN
      WRITE(NOP,100) B
 100  FORMAT(1X,A4,'  ITDEC HAS NOT PICKED UP A SPECIAL CHARACTER ')
      RETURN
      END
      SUBROUTINE FARRAY(X,N)
      CHARACTER RB,LB,A,ITDEC
      DIMENSION X(2),LOCL(20),NLEV(20)
      DATA RB/')'/,LB/'('/
      L=0
      LEV=0
  1   IF(NEWDEC(0).LT.0) GO TO 7
      Y=FLDEC(1)
      IY=1
      LDO=0
      IF(NEWDEC(0).NE.0) GO TO 2
  8   A=ITDEC(1)
      IY=Y
      IF(A.EQ.LB) GO TO 4
```

```
      IF(A.EQ.RB) GO TO 5
      Y=FLDEC(1)
  2   DO 3 I=1,IY
      L=L+1
  3   X(L)=Y
      LDO=1
      IF(NEWDEC(0).EQ.0) GO TO 8
      GO TO 1
  4   LEV=LEV+1
      LOCL(LEV)=L
      NLEV(LEV)=IY
      GO TO 1
  5   NT=NLEV(LEV)-1
      LL=LOCL(LEV)
      IF(LDO.EQ.0) THEN
      L=L+1
      X(L)=Y
      LDO=1
      ENDIF
      LEN=L-LL
      DO 6 I=1,NT
      CALL EQVEC(X(LL+1),X(L+1),LEN)
      LL=L
  6   L=L+LEN
      LEV=LEV-1
      IF(NEWDEC(0).EQ.0) GO TO 8
      GO TO 1
  7   WRITE(6,100) L
  100 FORMAT(1X,I10,' NUMBERS READ IN')
      N=L
C     CALL WORDEC(SECT,8,0)
      RETURN
      END
      SUBROUTINE IARRAY(M,N)
      CHARACTER RB,LB,A,ITDEC
      DIMENSION M(2),LOCL(20),NLEV(20)
      DATA RB/')'/,LB/'('/
      L=0
      LEV=0
 1    IF(NEWDEC(0).LT.0) GO TO 7
      K=INDEC(1)
      IDO=0
      IK=1
      IF(NEWDEC(0).NE.0) GO TO 2
  8   A=ITDEC(1)
      IK=K
      IF(A.EQ.LB) GO TO 4
      IF(A.EQ.RB) GO TO 5
      K=INDEC(1)
  2   DO 3 I=1,IK
      L=L+1
  3   M(L)=K
      IDO=1
      IF(NEWDEC(0).EQ.0) GO TO 8
      GO TO 1
  4   LEV=LEV+1
      LOCL(LEV)=L
      NLEV(LEV)=IK
      GO TO 1
  5   NT=NLEV(LEV)-1
      LL=LOCL(LEV)
      IF(IDO.EQ.0)THEN
      L=L+1
      M(L)=K
```

```
      IDO=1
      ENDIF
      LEN=L-LL
      DO 6 I=1,NT
      CALL EQVEC(M(LL+1),M(L+1),LEN)
      LL=L
    6 L=L+LEN
      LEV=LEV-1
      IF(NEWDEC(0).EQ.0) GO TO 8
      GO TO 1
    7 WRITE(6,100) L
  100 FORMAT(1X,I10,' NUMBERS READ IN')
      N=L
C     CALL WORDEC(SECT,8,0)
      RETURN
      END
      SUBROUTINE EQVEC(A,B,N)
      DIMENSION A(1),B(1)
      DO 1 J=1,N
    1 B(J)=A(J)
      RETURN
      END
```