AAEC/E634

AAEC/E634

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

AN IMPLICIT ITERATIVE SCHEME FOR SOLVING LARGE SYSTEMS OF LINEAR EQUATIONS

by

J.M. BARRY

J.P. POLLARD

DECEMBER 1986

ISBN 0 642 59842 8

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

AN IMPLICIT ITERATIVE SCHEME FOR SOLVING LARGE SYSTEMS OF LINEAR EQUATIONS

by

J.M. BARRY
J.P. POLLARD

ABSTRACT

An implicit iterative scheme for the solution of large systems of linear equations arising from neutron diffusion studies is presented. The method is applied to three-dimensional reactor studies and its performance is compared with alternative iterative approaches.

National Library of Australia card number and ISBN 0 642 59842 8

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

COMPUTER CALCULATIONS; CONVERGENCE; FOURIER ANALYSIS; ITERATIVE METHODS; MATRICES; MESH GENERATION; NEUTRON DIFFUSION EQUATION; NUMERICAL SOLUTION; REACTORS

CONTENTS

1. INTRODUCTION

The method of implicit non-stationary iteration (MINI) was introduced by Barry and Pollard [1977,1978] for solving large systems of linear equations arising from a finite difference discretisation of two- and three-dimensional elliptic equations. The steady-state neutron diffusion equation of interest in the present work is

$$-\nabla . D_g \nabla \phi_g + \sigma_{rg}\, \phi_g = \sum_{g'=1}^{NG} \sigma_{sg'-g}\, \phi_{g'} + \chi_g \sum_{g'=1}^{NG} \frac{\upsilon}{k} \sigma_{fg'}\, \phi_{g'} \quad , \tag{1.1}$$

Where D_g, σ_{rg}, $\sigma_{sg'\rightarrow g}$, $\nu\sigma_{fg}$, χ_g are given data for neutron energy groups $g = 1, 2, ...,NG$ (diffusion coefficient, macroscopic removal cross section, scattering transfer matrix with zero self scatter, fission emission and fission spectrum, respectively), whereas k and ϕ_g are the required neutron multiplication factor and neutron flux.

Basically, the overall strategy chosen for solving the three-dimensional neutron diffusion steady state problem in the code POW3D [Barry and Pollard 1986] calls for solution by layers. For example, a layer might be an x,y plane, through which an iteration process will typically pass several times before moving on to the next plane. The overall strategy consists of repeatedly solving layers (i) - (vi):

(i) 'outers' - calculate fission source to achieve criticality,

(ii) 'group rebalance' - rebalance the whole reactor by groups,

(iii) 'groups' - calculate the scatter source including upscatter,

(iv) 'region rebalance' - rebalance the whole reactor by volume regions,

(v) 'between plane inners' - solve for all planes, and

(vi) 'plane inners' - solve for a plane,

until convergence is achieved.

Except for the outer layer, which is accelerated by Chebyshev extrapolation, all other layers produce a set of N linear equations:

$$A\, x = c \quad , \tag{1.2}$$

where A is a sparse, real matrix with elements

$$a_{ij} \leq 0 \quad i \neq j, \; i=1, 2,...N \; ; \; j=1, 2,...,N \; , \tag{1.3}$$

$$a_{ii} \geq -\sum_{j \neq i} a_{ji} \qquad \text{(with inequality for at least one value of i)} \tag{1.4}$$

and/or

$$a_{ii} \geq -\sum_{j \neq i} a_{ij} \qquad \text{(with inequality for at least one value of i)} \tag{1.5}$$

$$\text{with } c_i \geq 0 \qquad \text{(not all zero).} \tag{1.6}$$

In addition, for all inner layers, A is irreducible, symmetric, positive definite and block tridiagonal. As a consequence of properties (1.3) - (1.6), the unknown elements are all positive,

$$x_i > 0 \quad i=1, 2,...,N \; ,$$

which is consistent with the physical requirements.

Discussions of how each matrix A for the different layers arises will be deferred. Certainly each layer gives rise to a different type of matrix. Nevertheless, we start with the simplest form of MINI, *i.e.* one which operates in a point mode.

2. POINT MINI

Perhaps the simplest effective iterative scheme for solving equation 1.2 for the inner layer is the Gauss-Seidel (GS) scheme:

$$x_i^{(n)} = [c_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(n)} - \sum_{j=i+1}^{N} a_{ij}\, x_j^{(n-i)}] / a_{ii},\ i=1, 2, \cdots, N \quad ,$$

where the index n is an iteration count (improperly formed sums are taken as zero). Because of the matrix properties (1.3-1.6), the GS approach converges for any trial solution $x^{(o)}$ [Faddeyeva 1959]. In addition, if all the elements of the trial solution are positive, so too are the elements of subsequent iterative solution. Although this positive preserving property is not essential (*e.g.* for successive over-relaxation [SOR] schemes), it was considered worthwhile preserving when MINI was devised.

Schemes based on successive over-relaxation are highly successful; however, they require the determination of an optimum extrapolation parameter given by Young [1954] as

$$\omega_{opt} = 2/[1 + \sqrt{1-\lambda_{GS}}] \quad .$$

The determination of the largest magnitude eigenvalue λ_{GS} of the *GS* iteration matrix can require considerable iterations before the optimal extrapolation factor can be applied. In practice, the overhead required to determine ω may amount to as many as 30 non-optimum iterations. For two-dimensional reactor source and criticality studies, a succession of 'layers' arises in which ω does not alter because changes affect only the source terms (the right hand side) of equation 1.1, hence the initial effort taken to determine ω need not be repeated.

During reactor kinetics calculations, however, the elements of A may change and recalculation of ω may be necessary. This led to us seek a method which required little memory of previous iterations. Although this was our motivation, we discovered a highly effective scheme for handling layers (iii) to (vi) of the static source or criticality problem 1.1. For simplicity, MINI is introduced in the point form in which it was first developed.

Essentially, the approach is to make the term not yet updated in the *GS* method,

$$\sum_{j=i+1}^{N} a_{ij}\, x_j^{(n-1)} \quad ,$$

implicitly dependent on the unknown term being sought, that is $x_i^{(n)}$. Perhaps the simplest way to do this is with the term

$$\sum_{j=i+1}^{N} a_{ij}\, x_j^{(n-1)}\, x_i^{(n)}/x_i^{(n-1)} \ ;$$

however, an erroneous solution is sometimes obtained. The implicit idea has been applied successfully to accelerating convergence of the alternating direction implicit method with implicit buckling corrections, ADI-B^2 [Hageman and Yasinsky 1969]. An alternative to the simple multiplying factor $x_i^{(n)}/x_i^{(n-1)}$ was sought.

A twofold approach is adopted. In addition to an implicit correction factor we returned to ideas similar to those of early hand relaxation, where an extrapolation parameter $\gamma_{ij}^{(n-1)}$ is associated with each non-zero matrix element a_{ij} (j > i) so that a better estimate of x_j than $x_j^{(n-1)}$ is obtained. We could replace $x_j^{(n-1)}$ in the GS method by the term

$$x_j^{(n-1)} + \gamma_{ij}^{(n-1)} (x_i^{(n)} - x_i^{(n-1)})\, x_j^{(n-1)}/x_i^{(n-1)} \ ;$$

however, in practice, the simplest combination

$$x_j^{(n-1)} + \gamma_{ij}^{(n-1)} (x_i^{(n)} - x_i^{(n-1)}) \tag{2.1}$$

is superior. Our MINI equations are then

$$\sum_{j=1}^{i} a_{ij}\, x_j^{(n)} + \sum_{j=i+1}^{N} a_{ij}\, [x_j^{(n-1)} + \gamma_{ij}^{(n-1)}\, (x_i^{(n)} - x_i^{(n-1)})] = c_i,\ i=1, 2, \ldots, N,$$

from which we obtain the expression

$$x_i^{(n)} = [c_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(n)} - \tag{2.2}$$

$$- \sum_{j=i+1}^{N} a_{ij}(x_j^{(n-1)} - \gamma_{ij}^{(n-1)}\, x_i^{(n-1)})]/(a_{ii} + \sum_{j=i+1}^{N} a_{ij}\, \gamma_{ij}^{(n-1)}),\quad i=1, 2, \ldots, N.$$

To ensure that intermediate solutions are always positive, the following stringent but not absolutely necessary condition is applied:

$$0 \leq \gamma_{ij}^{(n-1)} \leq min\ (1,\ x_j^{(n-1)}/x_i^{(n-1)}) \tag{2.3}$$

for the extrapolating parameters yet to be derived.

With the iteration increment

$$\delta_i^{(n)} = x_i^{(n)} - x_i^{(n-1)}\ , \tag{2.4}$$

and the i^{th} stage residual

$$R_i^{(n)} = c_i - \sum_{j=1}^{i-1} a_{ij}\ x_j^{(n)} - \sum_{j=i}^{N} a_{ij}\ x_j^{(n-1)}\ ,$$

the MINI iterature scheme may be expressed

$$\delta_i^{(n)} = R_i^{(n)} / (a_{ii} + \sum_{j=i+1}^{N} a_{ij}\ \gamma_{ij}^{(n-1)})\ ,\quad i=1, 2,\ldots, N.$$

In terms of iteration increments, the GS and SOR iterative schemes are

$$\delta_{i(GS)}^{(n)} = R_i^{(n)}/a_{ii}$$

and

$$\delta_{i(SOR)}^{(n)} = \omega\ \delta_{i(GS)}^{(n)}$$

respectively. On a point basis only, we can make the identification

$$\delta_i^{(n)} = \omega_i^{(n-1)}\ \delta_{i(GS)}^{(n)}\ ,$$

where

$$\omega_i^{(n-1)} = a_{ii}/(a_{ii} + \sum_{j=i+1}^{N} a_{ij}\ \gamma_{ij}^{(n-1)})$$

between MINI and GS.

Because of the restriction 2.3 applied to each $\gamma_{ij}^{(n-1)}$, the extrapolating parameters $\omega_i^{(n)}$ are bounded

$$1 \leq \omega_i^{(n)} \leq a_{ii}\ /(a_{ii} + \sum_{j=i+1}^{N} a_{ij}\ ).$$

The upper limit, unlike the SOR extrapolation parameter limit, can exceed 2. Consequently, MINI can take big leaps for some iterations (which it usually does during early stages).

2.1 Extrapolating Parameters

The extrapolating parameters γ_{ij} are established using three conditions:

(i) $\gamma_{ij}^{(n-1)} \sim \gamma_{ij}^{(n)}$ (the gammas are assumed to vary slowly between iterations),

(ii) $0 \leq \gamma_{ij}^{(n)} \leq \min\ (1,\ x_j^{(n)}/x_i^{(n)})$, and

(iii) $x_j^{(n)} = x_j^{(n-1)} + \gamma_{ij}^{(n)}\ \delta_i^{(n)}$

Condition (iii) is extended to accommodate condition (ii) thus:

$$x_j^{(n-1)} + \gamma_{ij}^{(n)}\ \delta_i^{(n)} = p_{ij} x_j^{(n)} + (1-p_{ij}^{(n)})\ x_j^{(n-1)}\ ,\quad p_{ij}^{(n)} \varepsilon [-1, 1],$$

where, if possible, we take $p_{ij}^{(n)} = 1$, otherwise we reduce it to meet our requirements. Either way

$$\gamma_{ij}^{(n)} = p_{ij}^{(n)}\ \delta_j^{(n)}/\delta_i^{(n)}\ .$$

For $p_{ij}^{(n)} = 1$ the idea is straightforward, otherwise we resort to a heuristic approach. Sometimes we may obtain a 'large' magnitude ratio of $\delta_j^{(n)}$ / $\delta_i^{(n)}$ which violates condition (ii). This could occur when the solution at the point i, unlike the rest of the solution, is almost correct - possibly the trial solution was nearly correct at this point! In this circumstance, it would seem difficult to predict an event at the point j from an event at point i. We prefer to use a small magnitude value of $\gamma_{ij}^{(n)}$, so instead an 'up-ended' ratio is taken which also maintains a degree of continuity as n changes for values near 1. Although the best choice of $\gamma_{ij}^{(n)}$ is not obvious for $\delta_j^{(n)} > \delta_i^{(n)}$, the authors have found the up-ending rule to be generally the most satisfactory. Collecting our ideas, the value of $p_{ij}^{(n)}$ is such that with

$$\bar{\gamma}_{ij}^{(n)} = \begin{cases} \left|\delta_j^{(n)}/\delta_i^{(n)}\right| & \text{if } \left|\delta_j^{(n)}/\delta_i^{(n)}\right| \leq 1, \\ \left|\delta_i^{(n)}/\delta_j^{(n)}\right| & \text{otherwise,} \end{cases}$$

then

$$\gamma_{ij}^{(n)} = \begin{cases} \bar{\gamma}_{ij}^{(n)} & \text{if } 0 \leq \bar{\gamma}_{ij}^{(n)} \leq x_j^{(n)}/x_i^{(n)}, \\ x_j^{(n)}/x_i^{(n)} & \text{otherwise,} \end{cases} \qquad (2.5)$$

except that $\gamma_{ij}^{(o)} = 0$ (a GS iteration) begins the process.

2.2 Computational Procedures

For a more adequate description of the computational aspects of MINI, a FORTRAN segment of its implementation is given, in which

(i) A(I,J) denotes the matrix elements a_{ij} of the supplied matrix of order N taken to be present in primary memory;

(ii) X(I) denotes on entry trial solution elements $x_i^{(o)}$ and on exit converged results $x_i^{(n)}$;

(iii) DELTA(I) denotes elements of a temporary vector of storage for elements $\delta_i^{(n)}$ which are zero on entry;

(iv) MAX denotes a limit to the number of iterations;

(v) ERR denotes a required error limit (say ERR $= 10^{-4}$) such that the solution is considered converged if $\delta_i^{(n)}/x_i^{(n-1)} \leq$ ERR, i = 1,2,...,N; and

(vi) TOOSML is used to guard against the possibility that the right hand side of equation 2.2 may become zero, for example when the inequality $\gamma_{ij}^{(n)} < x_j^{(n)}/x_i^{(n)}$ is violated because of machine roundoff errors. TOOSML is determined by the machine word length and has at times been necessary to make the iteration significant.

```
C    MINI  SOLUTION OF LINEAR EQUATIONS  AX = C
           DO 1 NIT=1,MAX
           ERROR = -1.
           DO 2 I = 1,N
           V = A(I,I)
           U = C(I) - V*X (I)
           IF(I.EQ.1)GO TO 4
           I1 = I - 1
           DO 3 J=1,I1
         3 U = U - A(I,J) * X (J)
         4 W1 = 0.
           W2 = 0.
           IF(I.EQ.N)GO TO 10
           I1 = I + 1
           DO 5 J= I1,N
           AIJ = A(I,J)
           IF(AIJ.EQ.0.)GO TO 5
           XI = ABS(DELTA(I))
           IF(XI.EQ.0.)GO TO 6
           XJ = ABS(DELTA(J))
           IF(XJ.GT.XI)GO TO 7
C    NORMAL
           GAMMA = XJ/XI
           GO TO 8
C    UPEND
         7 GAMMA=XI/XJ
         8  XI=X(I)
           XJ=X(J)
           IF(XJ.GT.XI)GO TO 9
C    DOWNHILL
           GIJD=XJ/XI
```

```
      IF(GAMMA.GT.GIJD)GAMMA = GIJD
    9 WA1 = XJ - GAMMA*XI
      IF (WA1.GT.O)GO TO 12
      WA1 = TOOSML*XJ
      GAMMA = GAMMA*(1-TOOSML)
   12 W2 = W2 - AIJ*WA1
    6 W1 = W1 + AIJ*GAMMA
    5 CONTINUE
      U = U + W2
      V = V + W1
   10 DELTA(I) = X(I)-U/V
      IF(ERROR.LT.ERR)ERROR=ABS(DELTA(I)/X(I))
      X(I) = X(I) + DELTA(I)
    2 CONTINUE
      IF(ERROR.LE.ERR) GO TO 11 1  CONTINUE
C   NOT CONVERGED IN MAX ITERATIONS
C   SET APPROPRIATE ERROR FLAGS
C   FINISH
   11 CONTINUE
```

3. GENERALISED MINI

MINI may be generalised as a multiblock technique and POW3D functions with MINI nested to three levels. The generalised MINI is defined for the linear system (1.2) of order N. A block solution process for non-overlapping ordered partitions $m = 1,2,\ldots, M$, is developed. (For example, $m = 1$ might denote an x-line with y,z fixed, $m = 2$ its immediate neighbouring line, etc.) An iteration process is adopted which passes successively from one partition to the next in order, $m = 1,2,\ldots,M$, and then is repeated until convergence to the specified accuracy is achieved. (The specified accuracy is initially large, but this is reduced with each overall pass through all the layers (i) to (vi) discussed in section 1.)

The following sets of indices are now introduced for a partial stage of iteration pass n:

J_m = the set of indices j for elements x_j to be updated together as a block,

J_m^- = the set of indices j for elements x_j that have been updated already,

$= J_1 \cup J_2 \cup \ldots \cup J_{m-1}$

and

J_m^+ = the set of indices j for elements x_j that have not yet been updated,

$= J_{m+1} \cup J_{m+2} \cup \ldots \cup J_M$.

The basic block Gauss-Seidel iteration process may be written as

$$J_m^- a_{ij} x_j^{(n)} + J_m a_{ij} x_j^{(n)} + J_m^+ a_{ij} x_j^{(n-1)} = c_i, \quad i \in J_m, m = 1, 2, \ldots, M, \tag{3.1}$$

where the J's also denote partial summation over index j.

The MINI approach for hastening convergence of the basic process given by equation 3.1 provides a better estimate of the last term on the left through equation 2.1. The MINI process is then given by

$$J_m a_{ij} x_j^{(n)} + (J_m^+ a_{ij} \gamma_{ij}^{(n-1)}) x_i^{(n)} = c_i - J_m^- a_{ij} x_j^{(n)} - J_m^+ a_{ij} (x_j^{(n-1)} - \gamma_{ij}^{(n-1)} x_i^{(n-1)}), \quad i \in J_m, \ m = 1, 2, \ldots, M. \tag{3.2}$$

The block of equations for the set J_m may be solved by a direct method or by a further iterative process, and even by recursive use of the MINI process.

Problems of interest all have positive solutions x_i and, in the iterative process, even intermediate solutions are kept positive by applying condition 2.3 to make the right hand side of equation 3.2 positive. For further assurance

of positive intermediate solutions for x_j^n, $j \varepsilon J_m$, on the left hand side of equation 3.2, the block matrix of coefficients with elements

$$a_{ik} + (J_m^+ a_{ij} \gamma_{ij}^{(n-1)}) \delta_{ik}, \quad i \varepsilon J_m, \quad K \varepsilon J_m,$$

where

$$\delta_{ik} = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise} \end{cases}$$

must not contain excessively large γ values. Let g_m be the upper γ limit for block m; provided that

$$\gamma_{ij}^{(n-1)} < g_m \quad , \tag{3.3}$$

the block matrix of coefficients will yield positive solutions. Collecting the restrictions 2.2 and 3.3, the overall restriction becomes

$$0 \leq \gamma_{ij}^{(n-1)} < min\ (g_m,\ x_j^{(n-1)}/x_i^{(n-1)}) \quad .$$

A preliminary calculation of the block γ-limits (g_m) could be carried out or g_m could be lowered empirically if solutions are negative; however, we use

$$g_m = 1 \tag{3.4}$$

throughout. Intermediate solutions are always positive for layers (iii) to (vi) of section 1 with the restriction 3.4. For some coarse mesh acceleration procedures, however, symmetry is lost and negative intermediate solutions are possible. Of course, restrictions to maintain positive solutions are only part of the overall restrictions; more importantly, it is necessary for the process to converge and condition 3.4 is then desirable.

4. CONVERGENCE

A proof of convergence for MINI is still elusive as the convergence behaviour tends to be non-monotonic. For point MINI it is possible to establish convergence for a more stringent restriction on $\gamma_{ij}^{(n)}$ than equation 2.3, namely $\gamma_{ij}^{(n)} < \frac{1}{2}$. Such a limit, however, makes MINI far less effective and hence unacceptable. Nevertheless, convergence has some interesting asymptotic aspects.

4.1 Asymptotic Analysis

From equation 1.2 and definition 2.4, the point iterative process may be expressed in matrix form as

$$Q^{(n-1)}\ \delta^{(n)} = c - A\ x^{(n-1)} \quad , \tag{4.1}$$

where

$$Q^{(n-1)} = \begin{pmatrix} a_{11} + \sum_{j=2}^{N} a_{1j} \gamma_{1j}^{(n-1)} & 0 & 0 \ldots 0 \\ a_{21} & a_{22} + \sum_{j=3}^{N} a_{2j} \gamma_{2j}^{(n-1)} & 0 \ldots 0 \\ \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & a_{N3} \ldots a_{NN} \end{pmatrix}$$

Now $Q^{(n-2)}\ \delta^{(n-1)} = c - A\ x^{(n-2)}$, which when subtracted from equation 4.1 gives

$$\delta^{(n)} = T^{(n-1)}\ \delta^{(n-1)} \quad , \tag{4.2}$$

where the MINI iteration matrix is

$$T^{(n-1)} = [Q^{(n-1)}]^{-1}\ [Q^{(n-2)} - A] \quad .$$

Essentially, $Q^{(n)}$ is an approximation for A when multiplied by a class of vectors; the 'closer' $Q^{(n)}$ is to A the 'smaller' is $T^{(n-1)}$. Although not strictly essential (as $T^{(n-1)}$ may alternate between 'big' and 'small' and yet the combined effect could be to reduce our increment vector $\delta^{(n)}$), we take as our convergence condition

$$\rho(T^{(n-1)}) < 1, \tag{4.3}$$

where ρ denotes the spectral radius.

We now consider two convergence possibilities.

(a) Let us assume that an asymptotic situation prevails in order to establish some plausible conditions for convergence. If

$$\gamma_{ij}^{(n-1)} = \gamma_{ij}^{(n)} = \gamma_{ij} ,$$

then $Q^{(n)}$ (=Q) and $T^{(n)}$ (=T) would be independent of n. Let λ be a largest magnitude eigenvalue of T and x an eigenvector such that

$$T\hat{x} = \lambda\hat{x}.$$

Consequently,

$$[(A - Q) + \lambda Q]x = 0 \quad . \tag{4.4}$$

Suppose a non-negative vector c exists such that the solution of

$$A\bar{x} = \bar{c}$$

consists of all positive 'maximum downhill' elements defined by

$$x_j/x_i < 1 \text{ for all } j > i \text{ with } a_{ij} \neq 0$$

and for any other c, $x_j/x_i < x_j/x_i$. Then we could make the acceptable choice

$$\gamma_{ij} = \overline{x_j/x_i} < min\ (1, x_j/x_i), \tag{4.5}$$

which gives

$$A\bar{x} = Q\bar{x}.$$

The solution vector, $\bar{x}$, is then also an eigenvector, $\hat{x}$, of equation 4.4 corresponding to $\lambda = 0$. For matrices of small order, we found that all other eigenvalues are also 0 for the choice of γ_{ij} given by equation 4.5 - we suggest that this is so for any order matrix of the type we are studying. Asymptotically, as the iteration proceeds, the γ_{ij} tend to the values given by equation 4.5 and convergence is assured because the required condition 4.3 is met.

A simple example

$$A = \begin{pmatrix} 1 & -a & 0 \\ -a & 1 & -b \\ 0 & -b & 1 \end{pmatrix}$$

with $\bar{c} = (1,0,0)^T$, where T denotes transpose, gives

$$\bar{\gamma}_{12} = a/(1-b^2) \text{ and } \bar{\gamma}_{23} = b.$$

(b) An asymptotic situation may not exist; for this case, numerical studies have demonstrated that γ_{ij} must not exceed 1 for convergence. (Even cases with γ_{ij} = random number in (0,1) converge, although slowly.)

4.2 Fourier Analysis

Further support for the higher γ limit of 1 is provided by a 'local mode' [Brandt 1977] Fourier analysis of point MINI. A point away from the boundary is selected and the error analysed at successive iterations in terms of its Fourier components at that point. Although it is difficult to generalise from the example [G. Doherty, Wollongong University, private communication], the 'local mode' approach provides an insight into the behaviour of iterative techniques.

Consider a general equation of the form

$$-a\ \frac{\partial^2 u}{\partial x^2} - b\ \frac{\partial^2 u}{\partial y^2} = F(x,y) \quad , \tag{4.6}$$

subject to appropriate Dirichlet or Neumann boundary conditions. This equation can be approximated at a mesh point with indices (α,β) situated away from the boundary by the following finite difference representation:

$$a(-U_{\alpha+1\beta} + 2U_{\alpha\beta} - U_{\alpha-1\beta}) + b(-U_{\alpha\beta+1} + 2U_{\alpha\beta} - U_{\alpha\beta-1}) = h^2 F_{\alpha\beta} \quad , \tag{4.7}$$

where an evenly spaced mesh of width h is assumed about (α,β).

Let u and $\bar{u}$ represents the (n) and (n + 1) iterative approximations respectively for U during a MINI iteration, where the solution procedure already has passed through the points $(\alpha,\beta - 1)$ and $(\alpha - 1,\beta)$. Applying MINI to the

form given in equation 4.7 (and, for convenience, dropping the iteration dependence of γ) gives

$$a(-u_{\alpha+1\beta} - \gamma_{\alpha+1\beta}(\overline{u}_{\alpha\beta} - u_{\alpha\beta}) + 2u_{\alpha\beta} - u_{\alpha-1\beta}) +$$
$$+ b(-u_{\alpha\beta+1} - \gamma_{\alpha\beta+1}(u_{\alpha\beta} - u_{\alpha\beta}) + 2u_{\alpha\beta} - u_{\alpha\beta-1}) = h^2 F_{\alpha\beta} \quad . \tag{4.8}$$

The finite difference approximation (4.7) may be altered by the inclusion of two zero terms $\gamma_{\alpha+1\beta}(U_{\alpha\beta} - U_{\alpha\beta})$ and $\gamma_{\alpha\beta+1}(U_{\alpha\beta} - U_{\alpha\beta})$, respectively, to

$$a(-U_{\alpha+1\beta} - \gamma_{\alpha+1\beta}(U_{\alpha\beta} - U_{\alpha\beta}) + 2U_{\alpha\beta} - U_{\alpha-1\beta}) +$$
$$+ b(-U_{\alpha\beta+1} - \gamma_{\alpha\beta+1}(U_{\alpha\beta} - U_{\alpha\beta}) + 2U_{\alpha\beta} - U_{\alpha\beta-1}) = h^2 F_{\alpha\beta} \quad . \tag{4.9}$$

Let the error between the solution U and the two subsequent iterates u and $\overline{u}$ be written as

$$v = U - u$$

and

$$\overline{v} = U - \overline{u}.$$

Subtracting equation 4.8 from 4.9 gives

$$a(-v_{\alpha+1\beta} - \gamma_{\alpha+1\beta}\overline{v}_{\alpha\beta} - v_{\alpha\beta}) + 2\overline{v}_{\alpha\beta} - \overline{v}_{\alpha-1\beta}) +$$
$$+ b(-v_{\alpha\beta+1} - \gamma_{\alpha\beta+1}(\overline{v}_{\alpha\beta} - v_{\alpha\beta}) + 2\overline{v}_{\alpha\beta} - \overline{v}_{\alpha\beta-1}) = 0 \tag{4.10}$$

as and expression of error about the point (α,β).

The gammas are assumed to be constant in any direction (that is positionally independent) for the following analysis. (In normal practice this is not generally the situation, so the analysis is indicative only.) Consequently, $\gamma_{\alpha+1\beta}$ and $\gamma_{\alpha\beta+1}$ are written as γ_x and γ_y, respectively, to indicate the independence. Now, in a Fourier study, the (θ_1, θ_2) components of the error vectors on subsequent iterations are

$$v_{\alpha\beta} = \sum_{\theta_1}\sum_{\theta_2} A_{\theta_1\theta_2} e^{i(\alpha\theta_1+\beta\theta_2)}$$

and

$$\overline{v}_{\alpha\beta} = \sum_{\theta_1}\sum_{\theta_2} \overline{A}_{\theta_1\theta_2} e^{i(\alpha\theta_1+\beta\theta_2)} \quad .$$

Substitution into the linear relation 4.10, and use of the linear independence of the Fourier vectors to separate coefficients, yields

$$\overline{A}_{\theta_1\theta_2}(-a\gamma_x + 2a - b\gamma_y + 2b - ae^{-i\theta_1} - be^{-i\theta_2}) +$$
$$+ A_{\theta_1\theta_2}(-ae^{i\theta_1} + a\gamma_x - be^{i\theta_2} + b\gamma) = 0$$

for all θ_1 and θ_2.

A smoothing coefficient, or damping factor, $\mu(\theta_1, \theta_2)$, is introduced to measure the damping effect on a particular Fourier term after one iteration. It is defined as the ratio of the absolute magnitude of the Fourier coefficients after an iteration to those before, *i.e.*

$$\mu(\theta_1, \theta_2) = \frac{|\overline{A}_{\theta_1\theta_2}|}{|A_{\theta_1\theta_2}|}.$$

For the point version of MINI

$$\mu_{MINI}(\theta_1, \theta_2) = \frac{|ae^{i\theta_1} + be^{i\theta_2} - a\gamma_x - b\gamma_y|}{|2a + 2b - a\gamma_x - b\gamma_y - ae^{-i\theta_1} - be^{-i\theta_2}|} \quad . \tag{4.11}$$

MINI is easily extended from a point to a line scheme [Barry *et al.* 1977], and in that case a Fourier approach for the line method appropriate to equation 4.6 with lines taken parallel to the x axis, gives

$$\mu_{LMINI}(\theta_1, \theta_2) = \frac{|be^{i\theta_2} - b\gamma_y|}{|2a + 2b - b\gamma_y - ae^{-i\theta_1} - ae^{-i\theta_1} - be^{i\theta_2}|} \tag{4.12}$$

For a three-dimensional form of equation 4.6, a block version of MINI based on (x,y) planes is appropriate. In this case, the damping factor (provided that the plane iterations have converged) is

$$\mu_{BMINI}(\theta_1, \theta_2, \theta_3) = \frac{\left| ce^{i\theta_3} - c\gamma_z \right|}{2a + 2b + 2c - c\gamma_z - ae^{-i\theta_1} - be^{-i\theta_2} - ae^{i\theta_1} - be^{i\theta_2} - ce^{-i\theta_3}\Big|} \tag{4.13}$$

Inspection of equations 4.11 to 4.13 reveals that, in general, for most a, b (and c), as θ increases $\mu(\theta)$ decreases and is less than 1. (Such strong smoothing is not indicated when θ is small; in fact as $\theta \to 0$, $\mu(\theta) \to 1$.) This analysis suggests that MINI (like successive over-relaxation schemes) will have little difficulty in removing the high frequency error components and, consequently, that course mesh rebalancing or a multigrid approach will assist convergence if applied at the appropriate time. For $\gamma = 1$, the damping factor for MINI for all a, b and θ is

$$\mu_{MINI}(\theta_1, \theta_2) = \frac{\left| ae^{i\theta_1} + be^{i\theta_2} - a - b \right|}{\left| ae^{-i\theta_1} + be^{-i\theta_2} - a - b\right|} = 1.$$

This indicates no reduction for any of the error component frequencies and supports the restriction placed on γ. Although strictly speaking it is possible for γ=1 to arise in the determination given by equation 2.5, its occurrence is unlikely and its persistence is certainly improbable.

The smoothing factor for the three forms 4.11 to 4.13 corresponding to the Laplacian operator (equation 4.6 with a=b=1) is plotted for representative frequency components in figure 1. The components chosen represent low ($\theta_1 = \pi/15$), middle ($\theta_5 = 5\pi/15$; $\theta_7 = 7\pi/15$) and high ($\theta_{14} = 14\pi/15$) frequency divisions of the spectrum. The point and line smoothing factors $\mu_{MINI}(\theta)$ and $\mu_{LMINI}(\theta)$ are shown for a 16 × 16 grid on a square region of side π, and the block form μ_{BMINI} is determined on a three-dimensional equivalent grid. Figure 1 demonstrates clearly, for all frequencies displayed, that MINI (in all forms) has less difficulty in removing high frequency error components and that the block MINI process for the third spatial dimension has better smoothing qualities than the simpler line or point form on a two-dimensional grid of comparable order.

4.3 Simple Numerical Example

A simple example of a non-symmetric type of matrix appropriate for consideration with a generalised form of MINI is

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

which will be used in blocks of two to solve $Ax = c$ with GS and MINI iterations. We find that $\bar{c} = (0, 1, 0, 0)^T$, (T denotes transpose), and $\gamma_{23} = 1$, which is the only γ required. It is interesting to calculate the eigenvalues of the iteration matrix (*cf.* equation 2.7) when

$$\gamma_{23} = \bar{\gamma}_{23} - \varepsilon \ .$$

They are $\{0,0,0,2\varepsilon/(1 + 2\varepsilon)\}$ hence the GS method ($\gamma_{23} = 0$, $\varepsilon = \bar{\gamma}_{23}$) has the eigenvalues 0,0,0,2/3 . We note also that any $0 < \gamma \leq 1$ will give faster convergence than the GS method.

Table 1 shows the results of a numerical experiment with the same matrix A for different iterative methods and with different constant vectors, c:

$$c_1 = (1, 0, 0, 0)^T, \ c_2 = (0, 1, 0, 0)^T,$$

$$c_3 = (0, 0, 1, 0)^T, \ c_4 = (0, 0, 0, 1)^T.$$

A trial solution of $x^{(0)} = (1,1,1,1)^T$ was used in all cases and the terminating condition was taken as

$$\delta_i^{(n)}/x_i^{(n-1)} \leq 10^{-4}, \quad i = 1, 2, \ldots, N,$$

which is also the one generally applied in the present work. A value of $g_1 = 3/2$ holds as a block γ-limit although, as usual, $g_1 = 1$ was used in the iterative process.

TABLE 1
NUMBER OF ITERATIONS TO CONVERGE
SIMPLE 4 × 4 PROBLEM

	c_1	c_2	c_3	c_4
Block GS	24	25	26	2*
Block MINI	4	3	3	2
Point GS	35	36	36	33
Point MINI†	17	17	19	17†

* Elements x_3 and x_4 of trial solution are correct.
† Almost identical successive over-relaxation (SOR).

The results from table 1 show that MINI satisfactorily solves the simple problem. The block MINI converges rapidly with the early achievement of an asymptotic γ. On the other hand, there is no $\overline{c}$ for the point MINI as different element ratios, $\overline{x_{i+1}}/\overline{x_i}$, require different $\overline{c}$'s to achieve their minimum value. Consequently, no totally asymptotic situation is ever achieved except, in part, when $\gamma_{34} = 1/2$ is obtained. Analysis of the point MINI process for the simple problem shows that the minimum largest magnitude eigenvalue of the iteration matrix $\gamma_L = \pm 1/\sqrt{3}$ ($= \pm 0.5774$) is obtained when $\gamma_{12} = 1/2$, $\gamma_{23} = 1$ and $\gamma_{34} = 1/2$. Table 2 shows the variation of the largest magnitude eigenvalue, λ_p, of the point MINI process (assumed to be instantaneously stationary) based on the γ's obtained at each stage of iteration $n = 1,2,...,17$ during solution of $Ax = c_1$. From Table 2 we note that the MINI process is persistently convergent and that λ_p approaches the theoretical limit λ_L to within a few per cent for some iterations. In general, when there is no asymptotic situation, or when it is not achieved during the early stages, little can be proved about convergence. Section 6 of this paper is devoted to numerical experiments for real reactor calculations, which support the conjecture that MINI converges provided that the γ values do not exceed unity.

TABLE 2
VARIATION OF LARGEST MAGNITUDE
ITERATION EIGENVALUE

n	λ_p	n	λ_p
	GS start	9	-0.5874
1	0.8257	10	-0.5984
2	0.7934	11	-0.6716
3	0.8257	12	-0.6097
4	0.6667	13	-0.5876
5	0.6489	14	-0.6426
6	-0.5805	15	-0.6398
7	-0.6256	16	-0.5879
8	-0.6750	17	-0.6149

5. REGION REBALANCE

In all the numerical experiments reported in section 6, the code POW3D employs region and energy group rebalance to hasten convergence. Region rebalance is a proved approach [Wachspress 1966; McCallien 1976] for hastening overall convergence of the inner layers. Region rebalance is applied to the individual group equations

$$-\nabla . D_g \nabla \phi_g + \sigma_r \phi_g = S_g , \quad g = 1, 2,...,G \tag{5.1}$$

derived from equation 1.1, where S_g denotes scattering and fission contribution based on previous estimates of the flux. Application takes place at every outer set of inner iterations (for the x,y,z block in three-dimensional studies and for the x,y subdivision in two-dimensional studies).

A coarse mesh embracing collections of the fine mesh points used for the inners is usually chosen to surround fuel regions, massive voids, *etc.*, and is such that the number of coarse mesh intervals in each direction is approximately the square root of the number of fine mesh intervals [Pollard 1975]. Although not mentioned in section 3, as this would have introduced further complexity in notation, equation 5.1 is integrated over 'boxes' surrounding the intersecting fine mesh lines. Here, within a 'coarse box', we write the flux as the previous iteration

flux times a multiplying function f_g^r for the coarse box,

$$\overline{\phi}_g = \phi_g f_g^r \quad ,$$

and integrate $\overline{\phi}_g$ substituted into 1.1 equation over the collection of coarse boxes. Spatial pyramid forms for f_g^r have been studied [Nakamura 1977]. However, in this work we take f_g^r as constant over each coarse box. The process is then simple, as effectively we add together sets of the spatially integrated form of equation (5.1). With this form of region rebalance, the resulting set of equations is of similar form to those obtained for the fine mesh except that they are non-symmetric and only column diagonally dominant. The equation may be solved by a direct method, as in the earlier code POW [Pollard 1974], but here a MINI approach is adopted. With the MINI approach, the same iterative routines can be used to solve both the coarse and fine mesh equations provided that symmetry is not assumed. The structure of the matrix generated by this form of rebalance, and the ability of MINI to solve the equations, suggests an immediate extension to multigrid techniques. For the results reported here, the number of coarse mesh points selected is approximately the square root of the number of fine mesh points.

The region rebalance could have included groups, so we could have had a coarse set of equations for a smaller problem, similar to the original $x' \times y' \times z' \times g'$ used in the SNAP-3D code [McCallien 1975]. Here we are content to accelerate group convergence locally and superimpose an overall group balance to cope with the bulk effects. The basic idea of a multiplying factor is retained, but only one is admitted per group (no spatial dependence is included) and we only substitute $\overline{\phi}_g$ into equation 1.1 which is integrated over the whole reactor. The resulting set of NG equations is solved by a direct method as NG is small. During the group pass on the first encounter of a group which includes upscatter, group rebalance is carried out only once per outer.

The use of group rebalance is important when GS is applied to solving the groups, although hardly so when MINI is used. The results in section 6 support this claim.

6. REAL REACTOR CALCULATIONS

Results for several real reactor models are reported. Calculations were done on an IBM3031 computer and unless otherwise stated, all routines were compiled with the IBM FORTRAN H compiler (OPT = 2). For comparison, the inner layers were solved iteratively with SLOR, MINI and incomplete Choleski conjugate gradient (ICCG - Meijerink and van der Vorst, [1977]) methods. Rebalance procedures in space and energy were operative; energy groups were solved by MINI (unless otherwise stated); and outer layers were solved by Chebyshev extrapolation.

6.1 Two-dimensional Studies

6.1.1 Steady-state TRIGA thermal reactor

A 2D (r,z) model of a TRIGA thermal reactor mockup containing a central hole for irradiation was considered by Froehlich [1969]. The hole was surrounded by a stainless steel tube with iron plugs at the top and bottom. The fuel was 20 per cent enriched uranium in zirconium hydride, canned in stainless steel and cooled by light water. The 'energy layer' was covered by five groups and a steady state half-reactor calculation was undertaken with a 32 $\times$ 30 mesh. (Froehlich used 42 radial mesh intervals, whereas we use a radial mesh with 8 intervals of 1.5 cm, 2 of 0.5 cm, 2 of 0.375 cm, 15 of 1.16 cm and 5 of 1.08 cm, corresponding to an overall radius of 36.55 cm.) In general, holes cause difficulties in diffusion theory calculations and this example was no exception. The problem was undertaken as being representative of the difficult calculations sometimes presented to a reactor neutronics code such as POW3D. Table 3 records the total number of inner layer iterations and the machine time. An outer layer error level of 10^{-3} (rather than 10^{-4}, as used in other examples) was taken for both methods, as convergence to a higher accuracy is difficult to assess with any certainty.

The main difficulty with the TRIGA problem is the flat solution flux across the void. Graphic inspection of the iteration process using 'movies' (with routines of Cawley and Trimble [1977]) revealed that the flat flux converges slowly to the required level (even though this is hastened with a region re-balance procedure in POW3D). MINI is somewhat quicker than SLOR. This, plus the difficulty of calculating extrapolation parameters for each energy group required by SLOR, gives MINI a slight advantage for this problem. The results suggest that MINI is satisfactory.

TABLE 3
ITERATIONS FOR THE 2D FIVE-GROUP TRIGA PROBLEM

Method Inner Layer	Method Energy Group	Energy Groups 1	2	3	4	5	Total	Time (min)
SLOR	GS	112	102	80	130	212	636	2.61
MINI	GS	85	58	61	68	75	347	2.08
ICCG	GS	76	34	36	58	93	297	1.83
SLOR	MINI	112	103	80	119	217	631	2.73
MINI	MINI	86	56	59	68	73	342	2.06
ICCG	MINI	76	34	36	58	121	325	1.97

6.1.2 Steady-state Moata 2D thermal reactor

Moata is a 100 kW Argonaut research reactor using highly enriched uranium fuel clad in aluminium plates, and cooled and moderated by light water with a graphite reflector. It is represented by a two-dimensional (x,y) geometry for a quarter reactor with 22×16 mesh intervals. Four energy groups cover the energy range, three of which involve upscatter. Details were given by by Pollard [1974] and the calculation (table 4) is typical of a 'yesterday' problem.

TABLE 4
ITERATIONS FOR 2D FOUR-GROUP MOATA

Method Inner Layer	Method Energy Group	Energy Groups 1	2	3	4	Total	Time (min)
SLOR	GS	144	91	78	118	431	0.78
MINI	GS	97	93	71	84	345	0.82
ICCG	GS	86	82	65	74	307	0.79
SLOR	MINI	172	100	86	124	482	0.94
MINI	MINI	100	93	69	82	344	0.88
ICCG	MINI	86	84	65	72	307	0.87

The results suggest MINI and SLOR have roughly the same machine time economy, although MINI takes significantly fewer iterations. The number of floating point operations for the three methods are compared on a point basis in table 5 for two-dimensional problems.

TABLE 5
NUMBER OF FLOATING POINT OPERATIONS PER POINT FOR 2D ALGORITHMS

Type	SLOR	MINI	ICCG
* /	10	12	20
+ -	6	7	17

Despite the additional arithmetic overheads per iteration, ICCG is marginally the better of the implementations considered here for two-dimensional problems. However, it has a large storage overhead. The version tested uses single precision arithmetic for all intermediate storage and calculations, to make storage requirements equivalent to the double precision SLOR and MINI implementations. For some very fine mesh calculations, single precision arithmetic is insufficient to achieve the necessary accuracy for convergence.

The results suggest that MINI is an appropriate method for solving the inner layers of two-dimensional problems, and that the time penalty for handling energy groups is not high for problems in which there is little upscatter.

6.1.3 Kinetics problem with significant upscatter.

In table 6, the advantage of using MINI to handle the energy groups is shown for a problem with significant neutron upscatter. The calculation is a two-group kinetics calculation reported by Hageman and Yasinsky [1969] and calculated by Pollard [1977] among others. The spatial mesh for the quarter (x,y) reactor model is 11 × 11. The time-dependence arises from the insertion of half a dollar of reactivity, and fission emission is included in the 'scattering matrix'.

The number of iterations reported are averaged for the 20 time steps recorded.

TABLE 6
AVERAGE NUMBER OF INNER ITERATIONS FOR A 2D KINETICS CALCULATION

Method Plane	Method Group	Energy Groups 1	2	Total	Time (min)
SLOR	GS	54	44	98	2.86
SLOR	MINI	27	19	46	1.54
MINI	GS	57	43	100	3.00
MINI	MINI	50	20	70	1.80
ICCG	GS	48	40	88	3.41
ICCG	MINI	31	18	49	1.73

6.2 Three-dimensional Studies

MINI was originally envisaged as a three-dimensional iterative technique for reactor calculations. It was first tested and proved in two dimensions to justify its extension to the three-dimensional form.

6.2.1 Steady-state 3-D Moata Reactor

Details of a three-dimensional Moata reactor model are given by Pollard [1974] and Barry and Pollard [1982]. A 23 × 17 × 17 spatial mesh scheme is used with four energy groups. Three of the four energy groups involve upscatter. In table 7, the number of inner iterations and machine times for all combinations of methods are recorded. The calculation is repeated with refined mesh in the y-direction (half mesh spacing, *i.e.* 33 solution points orthogonal to the line solution used for the SLOR and MINI plane drivers).

TABLE 7
NUMBER OF INNER ITERATIONS REQUIRED TO CONVERGE 4-GROUP 3-D MOATA PROBLEM

Method (x,y)	Group (z)	(x,y) Plane Iterations 1	2	3	4	Total (x,y) Plane Iterations	Total z Iterations	Time (min)
		Normal Mesh Spacing						
SLOR	SLOR	4075	5955	2834	4950	17814	443	23.95
SLOR	MINI	2981	3689	2328	3459	12457	279	20.36
MINI	SLOR	3481	6805	2616	4277	17179	484	28.96
MINI	MINI	2724	2943	1998	2465	10130	277	21.32
ICCG	SLOR	3059	4043	2733	3493	13328	532	39.07
ICCG	MINI	2053	2032	1471	1715	7271	275	27.76
ICCG							348	24.78
		1/2 Mesh Spacing in y-direction						
SLOR	SLOR	7189	14831	8334	23414	53768	561	100.25
SLOR	MINI	4098	5151	4221	6182	19652	275	49.53
MINI	SLOR	6375	7010	3568	6119	23072	614	69.16
MINI	MINI	3095	3167	2544	3012	11818	296	45.60
ICCG	SLOR	3654	4486	3093	5162	16395	642	86.36
ICCG	MINI	2439	2073	1661	1828	7951	279	52.17
ICCG							460	54.26

In this direction considerably more iterations are required with SLOR than with MINI. This is reflected to some extent in the reported use of central processor unit (CPU) time. The 3D form of ICCG also requires more z iterations than any other combination with MINI in the z direction. The block methods SLOR and MINI require more arithmetic in the plane than ICCG. It would seem that the choice of ICCG switches much of the computational effort from the plane to the z direction. Attempts to redress this unfortunate shift by using ICCG in the plane with a MINI z driver lowers the number of z passes significantly. The same benefit is not available with an ICCG-SLOR combination.

The SLOR and MINI plane iteration routines reported for the three-dimensional studies were coded in Assembler language for efficiency. When adjusted by a factor of 1.3 for a typical (x,y) plane of the Argonaut type, the times for MINI-MINI and ICCG are comparable.

For the half mesh spacing in the y direction, the SLOR-SLOR combination appears rather hopeless. This is due partly to the difficulties with which SLOR obtains estimates for ω. Correction to the CPU time results again show that MINI compares favourably with ICCG.

In a second Moata study, the number of energy groups is increased to eight, seven of which now involve upscatter. Consequently, GS and MINI are expected to differ significantly in their ability to handle the energy layer. Because energy group rebalance may assist convergence with so many thermal groups, results with and without it are reported as well.

The results for normal and half mesh spacing are presented in table 8. In some instances, convergence was not obtained within prescribed time limits and extrapolation is applied to estimate results. Single precision ICCG proved inadequate to complete the calculation to the accuracy required, so the convergence criteria were relaxed for all iterative schemes.

TABLE 8
RESULT OF THE 3-D 3-GROUP MOATA MODE

*Convergence never obtained in time available; extrapolation used to obtain these figures

Spatial Iterative Scheme (x,y)	Spatial Iterative Scheme (z)	Energy Iterative Scheme	Group Balance	x,y Plane Iterations x 100	z Plane Iterations	Time (min)
SLOR	SLOR	GS	OFF	*2200	5500	*280
SLOR	SLOR	MINI	OFF	598	1608	93.40
SLOR	SLOR	GS	ON	1647	4298	206.48
SLOR	SLOR	MINI	ON	402	1079	73.73
MINI	MINI	GS	OFF	*1550	*3800	*280
MINI	MINI	MINI	OFF	318	898	82.01
MINI	MINI	GS	ON	1077	2993	225.43
MINI	MINI	MINI	ON	281	803	76.87
ICCG		GS	OFF		*5410	*310
ICCG		MINI	OFF		*1364	108.25
ICCG		GS	ON		3635	231.05
ICCG		MINI	ON		946	87.62
ICCG	SLOR	MINI	ON	323	1463	125.53
ICCG	MINI	MINI	ON	183	805	92.72
			½ Mesh Spacing in y-direction			
SLOR	SLOR	MINI	ON	*3000	*8000	*680.0
MINI	MINI	MINI	ON	304	833	154.4
ICCG		MINI	ON		1048	167.9
ICCG		MINI	ON	192	812	171.6
ICCG	SLOR	MINI	ON	335	1484	233.4

The results indicate that energy group rebalance makes a considerable difference when the GS scheme is used for energy iterations, but its effect is lessened when MINI is used. MINI, however, is the most effective means of accelerating energy convergence and its performance is outstanding.

SLOR seems to become an inferior performer (on CPU time) as the reactor model becomes more refined, whereas ICCG and MINI-MINI are roughly comparable when allowances for language differences are made. It is observed that ICCG always requires more z passes than a MINI driven in the same direction.

6.2.2 Fast 3D reactor problem

Fast reactors involve little upscattering. Benchmark calculations made by Buckel et al. [1977] on a sodium-cooled fast breeder reactor (LMFBR) are recalculated here (table 9) in two (x,y,z) versions:

B1 of $20 \times 20 \times 19$ mesh points, and
B2 of $39 \times 39 \times 37$, a half spacing B1 mesh points.

TABLE 9
NUMBER OF INNER ITERATIONS TO CONVERGE LMFBR PROBLEM

*Convergence not achieved in time available;
extrapolation used to obtain the figures.

Method		Group 1	2	3	4	Total	Total z	Time
			(x,y) plane iterations			iterations	iterations	(min)
(x,y)	(z)			B1 - Coarse Mesh Spacing				
SLOR	SLOR	3516	3248	2165	2140	11069	226	17.82
SLOR	MINI	2867	2653	2150	2175	9845	194	18.30
MINI	SLOR	5751	3715	2155	1732	13353	333	24.92
MINI	MINI	2676	2466	1735	1744	8621	213	21.18
ICCG	SLOR	2486	2180	1489	1622	7777	267	24.03
ICCG	MINI	1770	1798	1174	1202	5944	215	24.03
ICCG							207	20.65
				B2 - Fine Mesh Spacing				
SLOR	SLOR	10141	37167	4475	5907	57690	744	279.4
SLOR	MINI	9433	7861	6328	6833	30455	366	204.3
MINI	SLOR	*9287	*33948	*4877	*4934	*52946	*687	*310.0
MINI	MINI	8439	7035	4273	4793	24540	306	189.8
ICCG	SLOR	*9160	*10970	*4960	*8112	33202	*634	*353.0
ICCG	MINI	5009	4464	2811	2969	15253	282	212.7
ICCG							454	228.1

For the simpler B1 geometry, there are few differences in recorded CPU time, but for the more involved B2 geometry SLOR again performs poorly. The CPU time for MINI-MINI and ICCG are somewhat comparable, but MINI takes fewer iterations to drive the z direction iteration process.

6.3 Other Factors determining the Choice of Iterative Method

It is not sufficient to measure the cost of an algorithm in terms of either raw CPU time alone or the number of iterations required to obtain a converged solution. Various aspects of computer architecture should be taken into account before a particular algorithm is selected for a given task on a machine set to a specific configuration. The choice is affected by the scale and nature of the multiprogramming environment, the sequential or parallel nature of the machine, the use of real or virtual memory, the way in which computer jobs are costed and the priority which which the return of results is required.

The I/O overheads in large three-dimensional reactor calculations can be very significant, so an attempt is made to assess the cost of their implementation. The I/O transfer mechanisms of the code POW3D are fully explained in Barry and Pollard [1986]. Suffice it to say that the amount of real memory available for a large problem is assumed to be sufficient to contain all the data and work space necessary for an (x,y) plane. This is realistic and all data transfers may be effected by direct I/O references or virtual memory. Either way, the costs are comparable. The I/O and arithmetic costs for a single z pass are given in table 10 for the three methods.

TABLE 10
NUMBER OF I/O TRANSFERS AND FLOATING POINT OPERATIONS PER z ITERATION FOR THE 3-DIMENSIONAL FORMS OF ITERATIVE SCHEMES AS IMPLEMENTED ON POW3D

N_x, N_y, N_z are the numbers of grid points in each direction while I is the number of iterations necessary to converge the 2-dimensional sub-systems.

Type of Operation	SLOR	MINI	ICCG
Matrix block transfers	N_z	N_z	$3\ N_z$
Vector block transfers	$3\ N_z$	$7\ N_z$	$20\ N_z$
*/ Floating point arithmetic	$(3 + 10I)N_xN_y$	$(9 + 12I)N_xN_y$	$37\ N_xN_y$
+ - Floating point arithmetic	$(5 + 6I)N_xN_y$	$(5 + 7I)N_xN_y$	$32N_xN_y$

It is difficult to compare the block forms SLOR and MINI directly with ICCG as the number of plane iterations (I) needs to be known. Provided MINI takes fewer than 2.3 plane iterations and SLOR fewer than 3.4, they appear to be better than ICCG at floating point multiplication and division operations. This is not unrealistic, as it is for the ordinary four-group reactor Moata, the average number of MINI iterations per plane is 2.15 and for the B2 fast reactor it is 2.16.

The number of I/O transfers strongly favours SLOR and MINI over ICCG. The matrix block transfers involve seven coefficients for each x,y grid point, whereas the vector block transfers involve a single number per grid point. The effect of the transfer on run time performance is shown in table 11 for the LMFBR study. Because exclusive use of the machine was never available, the real elapsed time has been discounted for the effects of a multi-user environment.

TABLE 11
TIMING CONSIDERATIONS FOR THE LMFBR STUDY

*Convergence was not achieved in the time available; extrapolation was used to obtain these figures

Method		Total CPU Time (min)	I/O CPU Time (min)	I/O Elapsed Time	Number I/O Calls X 1000	Total Plane Iterations	Total z Iterations
			B1 - Coarse Mesh Spacing				
SLOR	SLOR	17.8	1.8	16.8	36	11069	226
SLOR	MINI	18.3	2.3	22.4	44	9845	194
MIKNI	SLOR	24.9	2.7	26.5	56	13353	333
MINI	MINI	21.2	3.0	31.3	57	8621	213
ICCG	SLOR	24.0	2.1	11.2	44	7777	267
ICCG	MINI	24.0	1.7	12.7	57	5944	215
ICCG		20.7	5.0	62.5	102		207
			B2 - Fine Mesh Spacing				
SLOR	SLOR	279.4	9.9	73.6	153	57690	744
SLOR	MINI	204.3	12.0	144.7	155	30455	366
MINI	SLOR	310	12.7	152.0	194	52946	687
MINI	MINI	189.8	10.8	111.6	149	24.540	306
ICCG	*SLOR	353	8.9	98.9	442	33220	634
ICCG	MINI	212.7	10.5	118.3	57	14253	282
ICCG		228.1	25.3	296.2	401		454

7. CONCLUSIONS

The results suggest that MINI is a possible contender for solving spatial aspects of the neutron diffusion problem. The characteristics of each scheme are a little different and the most appropriate choice probably depends upon the problem at hand, the type of computer, and the priority with which a solution is required. ICCG is generally the fastest (in CPU time) for two-dimensional and small three-dimensional problems, although the savings are not great. Without an Assembler written version of ICCG (a more difficult task) the other schemes seem better. As the complexity of the problem increases, MINI-MINI and ICCG appear comparable in CPU time, but I/O penalties in data organisation favour MINI greatly. Although single precision ICCG is adequate for many problems, a double precision version would be necessary for large models with additional overheads in CPU time and storage requirements. The ability of MINI to iterate efficiently in the z direction makes it the preferred method for significant three-dimensional problems, and is the default option in POW3D.

As an energy driver, MINI is significantly more efficient than GS when there is considerable upscatter. Even without significant upscatters, a MINI groups driver involves relatively little additional expense and is worth retaining as the default option in the code for all thermal reactor problems. In addition, the ability of MINI to handle non-symmetric matrices of the form created in the coarse mesh rebalancing suggests the application of multigrid techniques based on MINI. Simple experiments on two-dimensional problems have been most encouraging.

8. REFERENCES

Barry, J.M., Jenkinson, J.H., Pollard, J.P. [1983] - A discrete Fourier analysis of coarse mesh rebalancing and some associated iterative methods. *J. Austral. Math. Soc.*, Ser. B, 25:190-216.

Barry, J.M., Pollard, J.P. [1977] - Method of implicit non-stationary iteration for solving neutron diffusion linear equations. *Ann. Nucl. Energy*, 4:485-493.

Barry, J.M., Pollard, J.P. [1978] - Application of the method of implicit non-stationary iteration (MINI) to 3D neutron diffusion problems. *Ann. Nucl Energy*, 6:121-131.

Barry, J.M., Pollard, J.P. [1982] - Solution of neutron diffusion equations by implicit non-stationary iteration. In *Numerical Solutions of Partial Differential Equations* (ed. J. Noye), North Holland, Amsterdam, pp.605-622.

Barry, J.M., Pollard, J.P. [1986] - AUS diffusion neutronics module POW3D - a mathematical description. AAEC/E612.

Brandt, A. [1977] - Multi level adaptive solutions to boundary-value problems. *Math. Comput.*, 31(138)333-390.

Buckel, G., Kufner, K., Stehle, B. [1977] - Benchmark calculations for a sodium-cooled breeder reactor by two- and three-dimensional diffusion methods. *Nucl. Sci. Eng.*, 64:75-89.

Cawley, R.J., Trimble G.D. [1977] - An interactive computing system for the AAEC Dataway network. AAEC/E245.

Faddeyva, V.N. [1959] - Computational Methods of Linear Algebra. Dover, New York.

Froehlich, R. [1969] - Computer independence of large reactor physics codes with reference to well behaved computer configurations. CONF-690401, 451-470.

Hageman, L.A., Yasinsky, J.B. [1969] - Comparison of alternating-direction time differencing methods with other implicit methods for the solution of neutron group-diffusion equations. *Nucl. Sci. Eng.*, 38:8-32.

McCallien, C.W.J. [1976] - Multidimensional neutron-diffusion calculations and coarse-mesh rebalancing. In *Computational Methods in Classical and Quantum Physics*, (ed) M.B. Hooper. Advance Publications, London, pp. 231-260.

Meijerink, J.A., van der Vorst, H.A. [1977] - An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comput.*, 31(137) 148-162.

Nakamura, S. [1977] - Computational Methods in Engineering and Science with Applications to Fluid Dynamics and Nuclear Systems. John Wiley, New York.

Pollard, J. P. [1974] - AUS module POW - a general purpose 0,1, and 2D multigroup neutron diffusion code including feedback-free kinetics. AAEC/E269.

Pollard, J.P. [1975] - Solution of neutron diffusion equations. In *Computational Methods in Mathematical Physics*, (eds) R.S. Anderssen, and R.O. Watts, University of Queensland Press, St Lucia, pp.138-157.

Pollard, J.P. [1977] - AUS diffusion model POW checkout - 1- and 2-dimensional kinetics calculations. AAEC/E387.

Wachspress, E.L. [1966] - Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics. Prentice-Hall., Englewood Cliffs, N.J.

Young, D.M. [1954] - Iterative methods for solving partial difference equations of the elliptic type. *Trans. Am. Math. Soc.*, 76: 92-111.

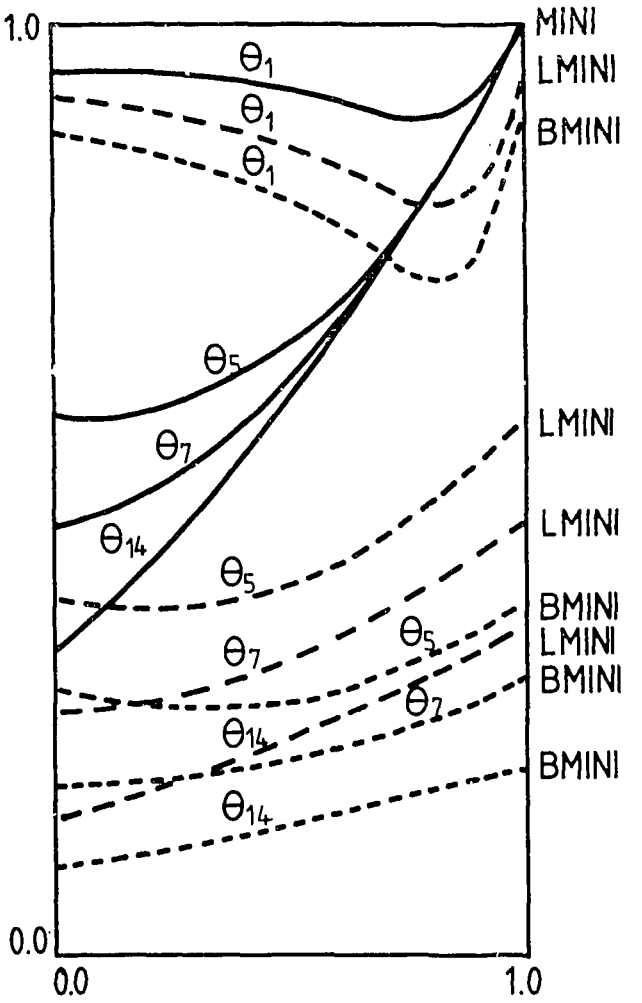

FIGURE 1. FOURIER ANALYSIS SMOOTHING FACTORS FOR THE TWO-DIMENSIONAL OPERATOR WITH MINI AND LMINI

The smoothing factor for the equivalent three-dimensional operator is included for BMINI. $\theta_1 = \frac{\pi}{15}$, $\theta_5 = \frac{5\pi}{15}$, $\theta_7 = \frac{7\pi}{15}$ and $\theta_{14} = \frac{14\pi}{15}$ components of error frequencies are displayed.