J. Nig. Ass. Math. Phys.
Vol. 4 (2000)

PARALLEL ITERATION METHODS FOR LINEAR SYSTEM OF EQUATIONS.

IKHILE, M. N. O.
DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BENIN, PMB 1154
BENIN CITY, NIGERIA.

ABSTRACT
In [13] we considered some multipoint iteration methods and their parallel cyclic implementation for large sparse linear systems. Interestingly, parallel computing methods offer a great promise of computing speed than can possibly be brought to bear on the numerical solution of many important practical problems of interest. In what follows, we continue our investigations of these methods.

1.0 INTRODUCTION

Parallel Cyclic Multipoint Methods.

It is of interest to solve the linear system of equations [1,3,4,7,9,15,17,18,19,20,21]

$$A\,x = b\,;\ \det(A) = 0\,. \qquad 1.1$$

$$A = [a_{ij}]_{i,j=1(1)n}\,;\ b = (b_1, b_2, \ldots, b_n)\,;\ x = (x_1, x_2, \ldots, x_n)^T \ ;\ n \geq 2$$

for the solution **x** by multipoint iteration methods with potential for parallelism. The parallel multipoint iteration methods we wish to consider is of the general form

$$x^{(2k+1)} = q_1 - \sum_{j=1}^{s} Q_{j,1}\ x^{(2k-j+1)}\ ;\ EGJ[P]$$

$$1.2a$$

$$x^{(2k+2)} = q_2 - \sum_{j=0}^{t} Q_{j,2}\ x^{(2k-j+2)}\ ;\ EGS[C]\ ;\ s \geq 1; t \geq 1$$

In particular, are the methods

$$x^{(2k+1)} = D^{-1}[b - \sum_{j=1}^{s} M_j\, x^{(2k-j+1)}];\ EGJ[P]$$

$$1.2b$$

$$x^{(2k+2)} = (D+L)^{-1}[b - \sum_{j=0}^{t} U_j\, x^{(2k-j+1)}];\ EGS[C]\ ;\ s \geq 1; t \geq 1$$

where necessarily **D** and **L** are diagonal and strictly lower triangular matrices for which det(**D**) ≠ 0, det (**D+L**) ≠ 0 respectively. However, the explicit nature of the matrices $M_j; U_j; j = 1(1)s$ are resolved in the context of a particular method. In the formalism of [13], the first is referred to as the predictor [p], while the second is the corrector [c]. The updating of the iterates is achieved in a cyclical manner that reveals their explicit potential for parallelisation, an advantage that is of remarkable interest in modern computing. In fact, each of the independent methods in (1.2) can be assigned to an independent processor for parallel execution of the problem in (1.1). The diagrams in fig (1.1) and fig (1.2) are of interest.

First iteration

	$x^{(0)}$	$x^{(-1)}$	$x^{(-2)}$	....	$x^{(1-s)}$	EGJ[P]
$x^{(1)}$	$x^{(1)}$	$x^{(0)}$	$x^{(-1)}$	....	$x^{(2-s)}$	EGS[C]

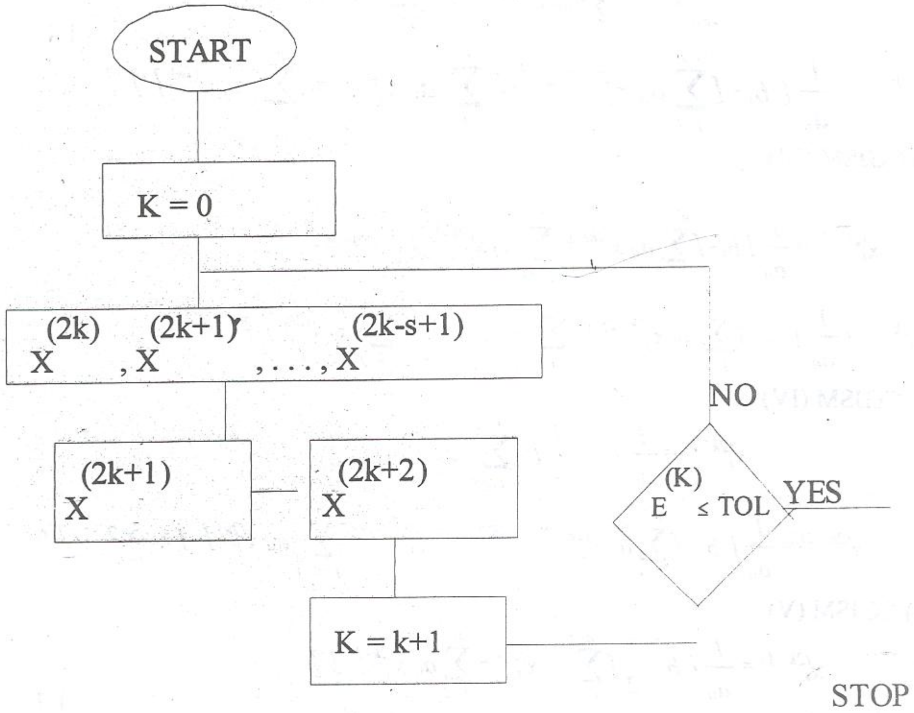
k-th iteration

	$x^{(2k+1)}$	$x^{(2k)}$	$x^{(2k-1)}$	....	$x^{(2k-s+1)}$	EGJ[P]
$x^{(2k+2)}$	$x^{(2k+2)}$	$x^{(2k+2)}$	$x^{(2k)}$	....	$x^{(2k-s+2)}$	EGS[C]

Fig (1.1): Cyclic Implementation

The sequential updating which requires only a single processor P is illustrated in the flow diagram in fig (1.2).

Fig.1.2 : A sequential implementation



with $E^{(k)} = || x^{(2k+2)} - x^{(2k+1)} ||$; TOL $\leq 10^{-p}$; p«2 .This flow diagram aligns into the sequencing

$$\rightarrow x^{(0)} \rightarrow x^{(1)} \rightarrow \ldots \rightarrow x^{(2k+1)} \rightarrow x^{(2k+2)} \rightarrow \ldots \quad : \qquad P$$

Typical examples include the following methods

(1) CGJSM (I)

$$x_i^{(2k+1)} = \frac{1}{a_{ii}} [b_i - \sum_{j=1}^{n} a_{ij} x_j^{(2k)}] \quad ; k = 0,1,2,\ldots \qquad 1.3$$

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k+2)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(2k+1)}]] ; \quad i = 1(1)n$$

(2) CGJSM (II)

$$x_i^{(2k-1)} = \frac{1}{a_{ii}} [b_i - \frac{1}{2} [\sum_{j=1}^{n} a_{ij} x_j^{(2k)} + \sum_{j=1}^{n} a_{ij} x_j^{(2k-1)}]]$$

1.4

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k-2)} + \frac{1}{2} [\sum_{j=i+1}^{n} a_{ij} x_j^{(2k-1)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(2k)}]]]$$

(3) CGJSM (III)

$$x_i^{(2k-1)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{m-1} a_{ij} x_j^{(2k)} + \sum_{j=m}^{n} a_{ij} x_j^{(2k-1)}]] \; ; m \geq 2$$

1.5

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k-2)} + \sum_{j=i+1}^{s} a_{ij} x_j^{(2k-1)} + \sum_{j=s+1}^{n} a_{ij} x_j^{(2k)}]]$$

(4) CGJSM (IV)

$$x_i^{(2k-1)} = \frac{1}{a_{ii}} [b_i - \frac{1}{2} [\sum_{j=1; j\neq i}^{n} a_{ij} x_j^{(2k)} + \sum_{j=1; j\neq i}^{n} a_{ij} x_j^{(2k-1)}]]$$

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k-2)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(2k-1)} + \sum_{j=s+1}^{n} a_{ij} x_j^{(2k)}]] ; s \geq 2$$ 1.6

(5) CGJSM (V)

$$x_i^{(2k-1)} = \frac{1}{a_{ii}} [b_i - \frac{1}{2} [\sum_{j=1}^{n} a_{ij} x_j^{(2k)} + \sum_{j=1}^{n} a_{ij} x_j^{(2k-1)}]]$$

1.7

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k-2)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(2k-1)}]]$$

(6) CGJSM (VI)

$$x_i^{(2k-1)} = \frac{1}{a_{ii}} [b_i - \sum_{j=1}^{n} a_{ij} x_j^{(2k)}]$$

1.8

$$x_i^{(2k-2)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(2k-2)} + \frac{1}{2} [\sum_{j=i+1}^{n} a_{ij} x_j^{(2k-1)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(2k)}]]]$$

(7) CGJSM (VII)

$$x_i^{(2k+1)} = \frac{1}{a_{ii}}[\, b_i - [\sum_{j=1;\, j\neq 1}^{m-1} a_{ij}\, x_j^{(2k)} + \sum_{j=m;\, j\neq 1}^{n} a_{ij}\, x_j^{(2k-1)}\,]\,]\,;\; m \geq 2$$

$$x_i^{(2k+2)} = \frac{1}{a_{ii}}[\, b_i - [\sum_{j=1}^{i-1} a_{ij}\, x_j^{(2k+2)} + \frac{1}{2}[\sum_{j=i+1}^{n} a_{ij}\, x_j^{(2k+1)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(2k)}\,]\,]\,] \qquad 1.9$$

(8) CGJSM (VIII)

$$x_i^{(2k+1)} = \frac{4}{5\, a_{ii}}[\, b_i - \sum_{j=1}^{n} a_{ij}\, x_j^{(2k)}\,] + \frac{1}{5} x_i^{(2k-1)}$$

$$x_i^{(2k+2)} = \frac{1}{a_{ii}}[\, b_i - [\sum_{j=1}^{i-1} a_{ij}\, x_j^{(2k+2)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(2k+1)}\,]\,] \qquad 1.10$$

The code name CGJSM (*) [*Cyclic Gauss-Jacobi-Seidel Method*] is for easy reference. Ready made starting iterate is

$$x^{(k)} = \begin{cases} (0,0,0,\ldots,0\,)^T\;; & k=0 \\ \left(\frac{b_1}{a_{11}}, \frac{b_2}{a_{22}}, \ldots, \frac{b_n}{a_{nn}}\right)^T; & k=1 \end{cases} \qquad 1.11$$

We illustrate the parallelism in the general case of the methods (1.2) as above. The arrows in the

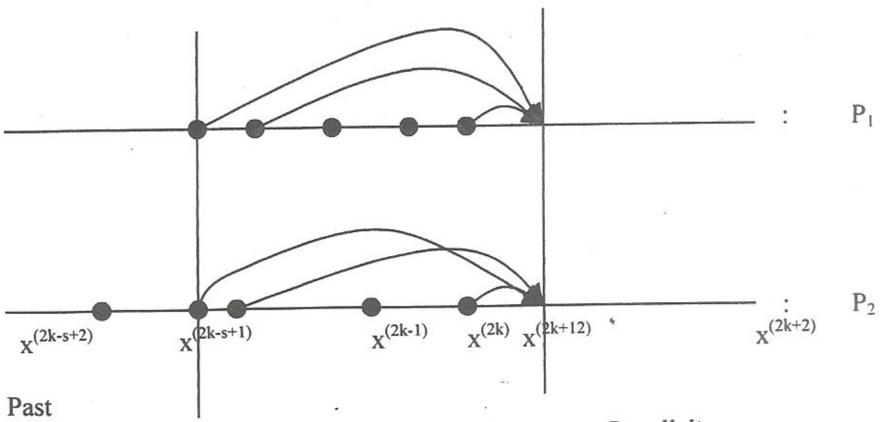


Fig. 1.3 : Illustrating Parallelism

Diagram (fig 1.3) shows that computation at points ahead of the front depends only on information behind the front. This is typical of a parallel computation. The sequencing of the process can be divided into two processes and each assigned to a different processor P_1

and P_2 respectively.

$$\rightarrow x^{(2k-s+1)} \rightarrow x^{(2k-s+2)} \rightarrow \ldots \rightarrow x^{(2k)} \rightarrow x^{(2k+1)} \rightarrow \ldots \qquad : P_1$$

$$\rightarrow x^{(2k-s+2)} \rightarrow \ldots\ldots\ldots \rightarrow x^{(2k+1)} \rightarrow x^{(2k+1)} \rightarrow \ldots \qquad : P_2$$

Fig 1.4: Parallel implementation

The actual realisation of this requires however , a parallel machine of multiprocessor capabilities. See [2,5,8,10,14]. The anticipated advantage is the computational speed-up which becomes relevant in a situation when response time is important, see fig (1.5).

2. CONVERGENCE OF THE CYCLIC MULTIPOINT METHODS .

The convergence of the methods is investigated by defining the block vector

$$V_k = (x^{(2k-s+1)}, x^{(2k-s+2)}, \ldots, x^{(2k+1)})^T \qquad 2.1$$

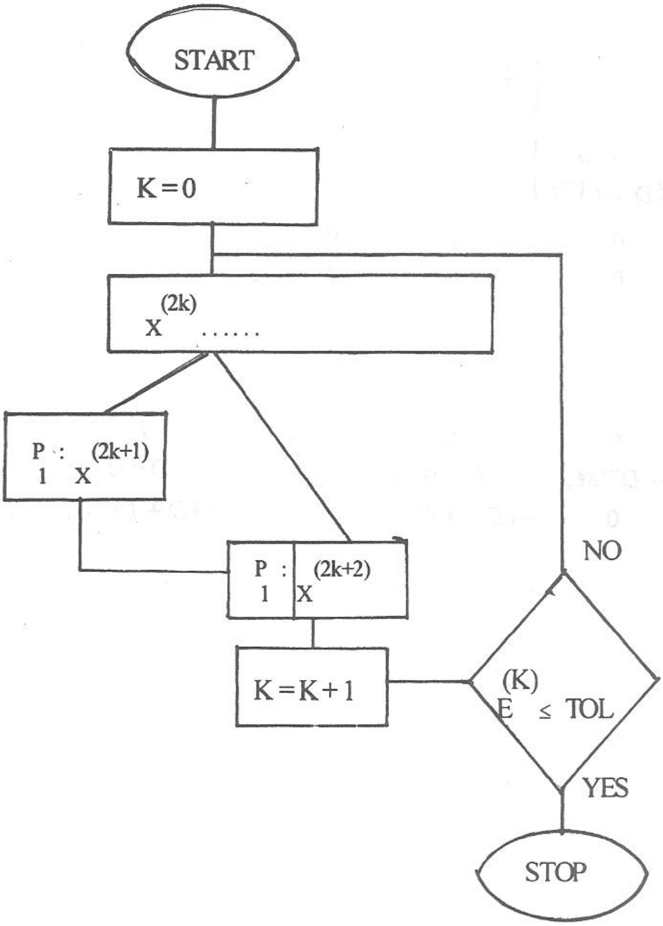
By this definition , the methods of (1.2) is transformed to the form

$$V_{k+1} = F + GV_k$$

$$F = \begin{pmatrix} 0 \\ \cdot \\ \ddots \\ \cdot \\ 0 \\ D^{-1}b \\ (D+L)^{-1}b \end{pmatrix};$$

$$G = \begin{pmatrix} 0 & I & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & I & 0 & \cdot & \cdot & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 0 & 0 & \cdot & \cdot & \cdot & I. & 0 \\ -D^{-1}M_s & -D^{-1}M_{s-1} & \cdot & \cdot & \cdot & -D^{-1}M_2 & -D^{-1}M_1 \\ 0 & -(D+L)^{-1}U_{s-1} & \cdot & \cdot & \cdot & -(D+L)^{-1}U_1 & -(D+L)^{-1}U_0 \end{pmatrix} \quad 2.2$$

Fig. 1.5: Parallel implementation



By induction

$$V_k = \left(I + \sum_{j=1}^{k-1} G^j \right) F + G^k V_0 = (I - G)^{-1} (I - G^k) F + G^k V_0 \ ; \ k \geq 2$$

Thus, if the spectral radius of the iteration matrix G of (2.2) is such that

$$\rho(G) < 1 \qquad 2.3$$

then the method of (1.2) is convergent .In particular, the method (1.4) is written as

$$x^{(2k+1)} = D^{-1} b - \frac{1}{2}[D^{-1} M\, x^{(2k-1)} + D^{-1} M\, x^{(2k)}]$$
$$x^{(2k+2)} = (D+L)^{-1} b - \frac{1}{2}[(D+L)^{-1} U\, x^{(2k)} + (D+L)^{-1} U\, x^{(2k+1)}] \qquad 2.3a$$

So that

$$\begin{pmatrix} x^{(2k)} \\ x^{(2k+1)} \\ x^{(2k+2)} \end{pmatrix} = \begin{pmatrix} 0 \\ D^{-1} b \\ (D+L)^{-1} b \end{pmatrix} + \begin{pmatrix} 0 & I & 0 \\ -\frac{1}{2} D^{-1} M & -\frac{1}{2} D^{-1} M & 0 \\ 0 & -\frac{1}{2}(D+L)^{-1} U & -\frac{1}{2}(D+L)^{-1} U \end{pmatrix} \begin{pmatrix} x^{(2k-1)} \\ x^{(2k)} \\ x^{(2k-1)} \end{pmatrix} \qquad 2.3b$$

with the definitions that

$$F = \begin{pmatrix} 0 & a_{12} & . & . & . & . & a_{1n} \\ a_{21} & 0 & a_{23} & & & & a_{12} \\ . & & & & & & \\ . & & & & & & \\ a_{n1} & . & . & . & . & a_{nn-1} & 0 \end{pmatrix};$$

$$U = \begin{pmatrix} 0 & a_{12} & a_{13} & . & . & . & a_{1n} \\ a_{21} & 0 & a_{23} & . & . & . & a_{2n} \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & a_{n-1,2} \\ 0 & . & . & . & . & . & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 0 & 0 & . & . & . & 0 \\ a_{21} & 0 & . & . & . & 0 \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ a_{n1} & . & . & . & a_{nn-1} & 0 \end{pmatrix} \qquad 2.3c$$

It will follow that the multipoint method (1.4) is convergent on the condition that

$$\rho\left(\begin{bmatrix} 0 & I & 0 \\ -\frac{1}{2}D^{-1}M & -\frac{1}{2}D^{-1}M & 0 \\ 0 & -\frac{1}{2}(D+L)^{-1}U & -\frac{1}{2}(D+L)^{-1}U \end{bmatrix}\right) < 1 \qquad 2.3d$$

However, we wish now to show that

Lemma 2.1

*Let the coefficient matrix A be **strictly diagonally dominant** (sdd), that is*

$$\sum_{l:\, j\neq i}^{n} \left(\left| \frac{a_{ij}}{a_{ii}} \right| \right) < 1;\ i = 1(1)n \qquad 2.4$$

Then, the parallel cyclic iteration methods expressed in (1.3) - (1.10) are convergent.

Proof.

The validity of this lemma is illustrated for the first method (1.3). The analysis extends analogously to the others. In this case

$$x_i^{(2k-1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j=1}}^{n} a_{ij}\, x_j^{(2k)} \right) \qquad 2.5$$

$$x_i^{(2k-2)} = \frac{1}{a_{ii}}\left(b_i - \left[\sum_{j=1}^{i-1} a_{ij}\, x_j^{(2k+2)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(2k+1)} \right] \right)$$

Define the error at the kth iteration as

$$e^{(k)} = \left(x - x_1^{(k)}, x - x_2^{(k)}, \ldots, x - x_n^{(x)}\right)^T ; k = 0, 1, 2, \ldots \qquad 2.6$$

$$= \left(e_1^{(k)}, e_2^{(k)}, \ldots, e_n^{(k)}\right)^T$$

where $e^{(0)}$ is the error in the starting iterate $x^{(0)}$

$$e_i^{(2k+1)} = \frac{1}{a_{ii}} \sum_{j=1}^{n} a_{ij}\, e_j^{(2k)}$$

2.7

$$e_i^{(2k+2)} = \frac{1}{a_{ii}} \left(\sum_{j=1}^{i-1} \bar{a}_{ij}\, e_j^{(2k+2)} + \sum_{j=i+1}^{n} a_{ij}\, e_j^{(2k+1)} \right)$$

Now,

$$\| e^{(2k+1)} \|_\infty \leq \lambda_1 \| e^{(2k)} \|_\infty$$

2.8

$$\| e^{(2k+2)} \|_\infty \leq \lambda_2 \| e^{(2k+2)} \|_\infty + \lambda_3 \| e^{(2k+1)} \|_\infty = \lambda_2 \| e^{(2k+2)} \|_\infty + \lambda_3 \lambda_1 \| e^{(2k)} \|_\infty$$

where

$$\lambda_1 = \max_{1 \leq i \leq n} \sum_{j=1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| ; \lambda_2 = \max_{1 \leq i \leq n} \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| ; \lambda_3 = \max_{1 \leq i \leq n} \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \qquad 2.9$$

Thus, equation (2.8) reduces nicely to

$$\| e^{(2k+2)} \| \leq \left(\frac{\lambda_3 \lambda_1}{1 - \lambda_2} \right)^{k+1} \| e^{(0)} \| \qquad 2.10$$

Therefore, convergence holds good for

$$\left| \frac{\lambda_3 \lambda_1}{1 - \lambda_2} \right| < 1 \qquad 2.11$$

This becomes readily true when the matrix **A** is *sdd*. Furthermore, (2.11) is equivalent to

$$\lambda_3 \lambda_1 + \lambda_2 < 1, \lambda_2 - \lambda_3 \lambda_1 < 1 \qquad 2.12$$

simultaneously. However, for the more general multipoint parallel cyclic methods expressed in (1.2) the error relation is given by

$$\| e^{(2k+2)} \| \leq W^{f(k)} \max_{0 \leq j \leq s-1} \left[\| e^{(j)} \| \right]; W < 1 \qquad 2.13$$

The explicit nature of the constant W and f(k) are resolved in the context of a particular method with the understanding that f(k) defines a linear function of the iteration index k and indeed f(k) = k+1 for the method (1.3) as seen from (2.10). In what is to follow, we consider some of these methods on two *sdd* matrices from [1]. We remark that the constituent methods are also iterative methods on their on right. Infact, the following can be proved for the particular method of (1.6a). The scheme (1.6a)

$$x_i^{(k-1)} = \frac{1}{a_{ii}} \left(b_i - \frac{1}{2} \left(\sum_{j=1;\, j \neq i}^{n} a_{ij}\, x_j^{(k)} + \sum_{j=1;\, j \neq i}^{n} a_{ij}\, x_j^{(k-1)} \right) \right); k = 0, 1, 2, ... \qquad 2.14$$

equivalently written in matrix notation as

$$x^{(k+1)} = D^{-1}b - \left[\frac{1}{2} D^{-1}M x^{(k)} + \frac{1}{2} D^{-1}M x^{(k-1)} \right] \qquad 2.15$$

is convergent if the spectral radius will be such that

$$\rho\left(\begin{bmatrix} \frac{1}{2} D^{-1}M & \frac{1}{2} D^{-1}M \\ I & 0 \end{bmatrix}\right) < 1 \qquad 2.16$$

irrespective of the arbitraries in the choice of the starting iterates $x^{(0)}$ and $x^{(-1)}$. This is equivalent to establishing that $\lim_{k\to\infty} V_k = (A^{-1}b \;\; A^{-1}b)^T$ by writing (2.14) as

$$V^*_{k+1} = \begin{pmatrix} D^{-1}b \\ 0 \end{pmatrix} + \begin{pmatrix} -\frac{1}{2} D^{-1}M & -\frac{1}{2} D^{-1}M \\ I & 0 \end{pmatrix} V^*_k \,;\; V^*_{k+1} = (x^{(k+1)}, x^{(k)})^T \qquad 2.17$$

and invoking the following, while applying limit processes.

Lemma 2.1

Let $\left[I - \sum_{j=1}^{s} B_j \right]$; $s \geq 2$ be a non-singular matrix where B_j ; $j = 1(1)s$ are abitrary. Then the inverse of

$$G = \begin{bmatrix} 1-B_1 & -B_2 & -B_3 & \dots & . & -B_s \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & . & . \\ . & . & . & . & . & \\ . & . & . & . & . & \\ . & . & . & . & . & \\ . & . & 0 & -1 & 1 & 0 \\ 0 & . & . & 0 & -1 & 1 \end{bmatrix} \qquad 2.18$$

is given explicitly by

$$G^{-1} = R\begin{bmatrix} 1 & \sum_{j=2}^{s} B_j & \sum_{j=3}^{s} B_j & \sum_{j=4}^{s} B_j & \cdots & B_{s-1}+B_s & B_s \\ 1 & 1-B_1 & \sum_{j=3}^{s} B_j & \sum_{j=4}^{s} B_j & \cdots & & \\ . & . & 1-\sum_{j=1}^{2} B_j & \sum_{j=4}^{3} B_j & & & \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ 1 & . & . & . & . & B_{s-1}+B_s & B_s \\ 1 & . & . & . & . & 1-\sum_{j=1}^{s-2} B_j & B_s \\ 1 & 1-B_1 & 1-\sum_{j=1}^{2} B_j & I-\sum_{j=1}^{3} B_j & \cdots & \cdots & 1-\sum_{j=1}^{s-1} B_j \end{bmatrix} \quad 2.19$$

$$R = \left(I - \sum_{j=1}^{s} B_j \right)^{-1} ; s \geq 2$$

A proof in part is available in [11,12], however a complete proof is possible by induction on s.

In fact, for an example from (1.2b) we write that

$\Lambda_{k-1} = F^* + G^* \Lambda_k$

with

$$G^* = \begin{pmatrix} -(D+L)^{-1}U_0 & -(D+L)^{-1}U_0 & -(D+L)^{-1}U_0 & . & . & . & -(D+L)^{-1}U_0 \\ I & 0 & & . & . & . & 0 \\ 0 & I & 0 & 0 & . & . & 0 \\ & & & & . & . & \\ & & & & & & \\ 0 & & & & & I & 0 \end{pmatrix}^{-1}$$

$$F^* = \begin{pmatrix} (D+L)^{-1}b \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}$$

If $\rho(G^*) < 1$, then by the above lemma

$$\lim_{k\to\infty} \Lambda_k = (1-G^*)^{-1}F^* =$$

$$R^* \begin{bmatrix} 1 & -\sum_{j=}^{t}(D+L)^{-1}U_j & -\sum_{j=2}^{t}(D+L)^{-1}U_j & \dots & -(D+L)^{-1}[U_{t-1}+U_t] & -(D+L)^{-1}U_t \\ 1 & 1+(D+L)^{-1}U_0 & -\sum_{j=0}^{t}(D+L)^{-1}U_j & \dots & -(D+L)^{-1}[U_{t-1}+U_t] & -(D+L)^{-1}U_t \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ 1 & 1+(D+L)^{-1}U_0 & 1+\sum_{j=0}^{1}(D+L)^{-1}U_j & \dots & & 1+\sum_{j=0}^{t-1}(D+L)^{-1}U_j \end{bmatrix} P.$$

$$R^* = \left(I + \sum_{j=0}^{t} (D+L)^{-1} U_j \right)^{-1} = \left[D + L + \sum_{j=0}^{t} U_j \right]^{-1} (D+L) = A^{-1}(D+L)$$

It becomes true that

$$\lim_{k \to \infty} \Lambda_k = (I - G^*)^{-1} F^* = A^{-1} b e^T ; e^T = (1,1,1,...,1)^T$$

and the convergence of (1.2b) is assured. Example of (1.2b(1)) is

$$x_i^{(k+1)} = \frac{1}{a_{ii}} [b_i - \frac{1}{2} [\sum_{j=1}^{n} a_{ij} x_j^{(k)} + \sum_{j=1}^{n} a_{ij} x_j^{(k-1)}]]$$

and that of (1.2b(ii)) is

$$x_i^{(k+1)} = \frac{1}{a_{ii}} [b_i - [\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} + \frac{1}{2} [\sum_{j=1}^{n} a_{ij} x_j^{(k)} + \sum_{j=1}^{n} a_{ij} x_j^{(k-1)}]]]$$

3. EFFECTS OF ROUNDOFF PROPAGATION

Roundoffs could be of influential effects [8,11,12,13,14,16] especially when the precision, p of the computing device is not too impressive. In particular, roundoffs propagates its effects in the case of the method (1.3) as follows :

$$x_i^{(2k+1)} = -\frac{1}{a_{ii}} \left[\sum_{j=1}^{n} a_{ij} x_j^{(2k)} (1+\lambda_j) \prod_{s=m(j)} (1+\eta_s^{(1)}) - b_i \right] (1+\rho_1^{(1)})(1+\rho_2^{(1)}).$$

$$x_i^{(2k+2)} = -\frac{1}{a_{ii}} [(\sum_{j=1}^{i-1} a_{ij} x_j^{(2k+2)} (1+\delta_j^{(1)}) \prod_{s=m(j)} (1+\eta_s^{(2)})$$

3.1

$$+ \sum_{j=i+1}^{n} a_{ij} x_j^{(2k+1)} (1+\delta_j^{(2)}) \prod_{s=m(j)''} (1+\eta_s^{(3)}))(1+\rho_1^{(2)}) - b_i](1+\rho_2^{(2)})(1+\rho_3^{(2)}).$$

where the meaning of the parameters are explained in a sense expressed in [10,11,12,13,16]:

λ_j: relative error in the multiplication of a_{ii} by $x_j^{(2k)}$.

$\prod_s (1+\eta_s^{(1)})$: relative error in the accumulated sum in (1.4a).

$\rho_2^{(1)}$: relative error in the addition of the b_i in (1.4a). $\rho_2^{(1)}$: relative error in the division of a_{ii}.

$\delta_j^{(1)}$: relative error in the multiplication of a_{ij} by $x_j^{(2k-2)}$ in (1.4b).

$\prod_s (1+\eta_s^{(2)})$: relative error in the first accumulated sum in (1.4b).

$\delta_j^{(2)}$: relative error in the multiplication $a_{ij} x_j^{(2k-2)}$ in (1.4b).

$\prod_s (1+\eta_s^{(3)})$: relative error in the accumulated sum in (1.4b).

$\rho_1^{(2)}$: relative error in the summation of the two sums of (1.4b).

$\rho_2^{(2)}$: relative error in the addition of b_i in (1.4b). $\rho_3^{(2)}$: relative error in the division of a_{ij} in (1.4b).

Expressing the result of (3.1) in matrix notations,

$$D_1^* x^{(2x+1)} = -[Q^* x^{(2k)} - b]$$
$$D_2^* x^{(2k+2)} = -[L^* x^{(2k+2)} + U^* x^{(2k+1)} - b] \qquad 3.2$$

with the entries of the matrices D_1^*, D_2^*, Q^*, U^*, and L^* defined as ,

$$d_{ii}^{*(1)} = \frac{a_{ii}}{(1+\rho_1^{(1)})(1+\rho_2^{(1)})} \; ; j = 1(1)n \quad ; d_{jj}^{*(2)} = \frac{a_{jj}}{(1+\rho_2^{(2)})(1+\rho_3^{(3)})} \; ; j = 1(1)n \qquad 3.3/3.4$$

$$q_{ii}^* = \begin{pmatrix} a_{ii}(1+\lambda_i) \prod_s (1+\eta_s^{(1)}) & ; \\ 0 & ; \quad i = j \end{pmatrix} \qquad 3.5$$

$$L_{ii}^* = \begin{pmatrix} a_{ii}(1+\delta_j^{(1)}) \quad (1+\rho_1^{(2)}) \prod_s (1+\eta_s^{(2)}); \; i > j, j = 1(1)i-1 & ; \quad i \neq j \\ 0 & ; \quad i < j \end{pmatrix} \qquad 3.6$$

$$U_{ii}^* = \begin{pmatrix} a_{ij}(1+\delta_j^{(2)})(1+\rho_1^{(2)}) \quad \prod_s (1+\eta_s^{(3)}) \; ; j = i+1(1)n, i < j & , \quad i \neq j \\ 0 & ; \quad i > j \end{pmatrix} \qquad 3.7$$

Similar analysis exist for the others . Now, it is clear from (3.2) that the roundoffs incurred in the computation process (1.4) induces some perturbation in the additive splitting of A. We are actually computing with the default splittings

$$A = \begin{Bmatrix} D + Q + \delta A \\ D + L + U + \delta A \end{Bmatrix}; \qquad Q = M \qquad 3.8$$

for the respective contituent method with δA , δ<u>A</u> as the error induced by roundoffs. However, if the precision p is reasonably high , that is $\{\lambda_i, \delta_j^{(m)}, \eta_s^{(u)}, \rho_v^{(m)}\} \leq 10^{-r}; m = 1,2; s = 1,2,...; u, v = 1,2,3; p >> 2$ and such that the coefficient matrix **A** is well-conditioned then the roundoff effects are sure to die off in the long run on the iteration index k when the convergence conditions holds as would be demanded. Now, by inspection of (2.3b) it is that the method is actually solving the block of linear equations

$$\begin{pmatrix} 0 & I & 0 \\ A & 0 & 0 \\ 0 & 0 & A \end{pmatrix} \begin{pmatrix} x \\ x \\ x \end{pmatrix} = \begin{pmatrix} x \\ b \\ b \end{pmatrix} \qquad 3.9$$

The first, which is a dummy system arises from the need to memorize $x^{(2k)}$. From the numerical results we provide, the convergence rate compare with that of GJM and GSM.

4 NUMERICAL EXPERIMENTS

In what follows we consider the methods on the following test problems :

Problem (I)

$$A = \begin{pmatrix} -4 & 1 & 0 & . & . & . & 0 \\ 1 & -4 & 1 & . & . & . & . \\ 0 & 1 & -4 & 1 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & 1 \\ 0 & . & . & . & . & 1 & -4 \end{pmatrix}; \quad b_i = \sum_{j=i}^{n} a_{ij}\,; i = 1,2,3,...$$

with $b = (b_1, b_2, b_3, ...)^T$ normalised as above to have the solution as $x = (1,1,1,...,1)^T$. The errors incurred propagates as

$$\|e^{(k)}\| \le \begin{cases} \left(\frac{1}{2}\right)^{k} \|e^{(0)}\| & GJM \quad ; \quad k = 1,2,3... \\ \left(\frac{1}{3}\right)^{k} \|e^{(0)}\| & GSM \quad ; \\ \left(\frac{1}{6}\right)^{\frac{k+1}{2}} \|e^{(0)}\| & CGJSM \quad ; \quad k \ge 3, \; k \; odd \end{cases}$$

in the respective methods. The next problems [1, 11,12] are

Problem(II) *Problem (III)*

$$A=\begin{pmatrix}4 & -2 & 0 & 0\\ -2 & 5 & -1 & 0\\ 0 & -1 & 4 & 2\\ 0 & 0 & 2 & 3\end{pmatrix}, b=\begin{pmatrix}0\\ 2\\ 3\\ -2\end{pmatrix}; \quad A=\begin{pmatrix}4 & -1 & 0 & -1 & 0 & 0\\ -1 & 4 & -1 & 0 & -1 & 0\\ 0 & -1 & 4 & 0 & 0 & -1\\ -1 & 0 & 0 & 4 & -1 & 0\\ 0 & -1 & 0 & -1 & 4 & -1\\ 0 & 0 & -1 & 0 & -1 & 4\end{pmatrix}, b=\begin{pmatrix}0\\ 5\\ 0\\ 6\\ -2\\ 6\end{pmatrix}$$

Table 4.1 : Problem II

Errors to problem (II) : $[E = \|x^{(2k+2)} - x^{(2k+1)}\|_4]$

			[12]	[12]	[11]	[11]	1.10a
k	GJM	GSM	EGJM(I)	EGJM(II)	EGSM(I)	EGSM(II)	DSM
10	1.45(-2)	4.61(-3)	9.33(-2)	6.90(-1)	3.77(-2)	3.98(-3)	4.17(-2)
11	9.71(-3)	1.97(-4)	5.75(-2)	1.08(-2)	2.38(-2)	2.30(-3)	3.29(-2)
12	6.22(-3)	8.43(-5)	4.31(-2)	4.41(-2)	1.87(-2)	1.95(-3)	2.55(-2)

CGJSM (I-VIII)

(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)
7.28(-4)	1.92(-1)	1.19(-3)	4.88(-2)	1.77(-4)	1.34(-4)	3.34(-3)	1.70(-3)
6.22(-5)	6.54(-2)	1.96(-3)	3.29(-2)	7.81(-5)	1.31(-4)	2.25(-3)	4.64(-3)
4.63(-6)	6.29(-2)	1.39(-3)	1.39(-3)	3.81(-5)	3.03(-5)	1.38(-3)	5.38(-3)

Table 4.2 : Errors to problem (III)

			[12]	[12]	[11]	[11]
k	GJM	GSM	EGJM(I)	EGJM(II)	EGSM(I)	EGSM(II)
10	1.79(-2)	1.05(-4)	1.14(-1)	2.90(-2)	1.97(-2)	2.37(-3)
11	1.08(-2)	3.87(-5)	9.75(-2)	1.59(-2)	7.81(-2)	1.21(-3)
12	6.52(-3)	5.84(-5)	9.05(-2)	1.19(-2)	7.61(-3)	6.54(-3)

CGJSM(I-VIII)

(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)
1.40(-5)	1.34(-1)	3.51(-2)	5.64(-3)	5.80(-4)	3.59(-3)	1.01(-4)	9.07(-4)
3.21(-6)	3.92(-1)	2.83(-4)	1.57(-4)	1.99(-4)	1.92(-3)	7.49(-3)	2.99(-4)
8.03(-7)	2.77(-1)	5.71(-5)	1.76(-4)	2.23(-4)	5.82(-5)	1.78(-2)	2.31(-4)

Consider that for all the test problems, CGJSM(I) is fastest in convergence. The reason for this will be accounted for in our conclusion.

5. CONCLUSION

Conclusively, the methods compares in convergence with the GJM and GSM, but in all cases of a *sdd* coefficient matrix, the GSM dominates in convergence efficiency save for the method CGJSM(I) which we wish to emphasis shortly, see table (4.1). However, the methods presented requires on implementation some extra-storage locations engendered by its multipoint nature which may not be a serious handicap considering the enormous storage capacity of emerging parallel computers. Futhermore; the algorithms possesses explicit potential for parallelisation since the constituent algorithms can be assigned to independent processors with a central control. The benefit of this is signifficant. The exploitation of this profound advantage requires in actual practice a computing device with inherently parallel architectural design, precisely a *supercomputer*. Furthermore, of paticular interest is the algorithm CGJSM(I) which is a cyclic GJM-GSM. This method is more accurate than the GJM and GSM. This may not be surprising because of

Lemma 5.1

If A is diagonally dominant then

(1) $\|(D+L)^{-1}U\|_\infty \le \|D^{-1}M\|_\infty \le 1; M = L+U, A = D+M$

(2) If there is convergence then the GSM is faster than that of GJM.

If A is *sdd* then the inequality in (1) is in the strict sense and the conclusion in (2) remains valid. In fact, the convergence speed of GSM in such a case is twice that of GJM. Now, from [13]

$$E^{(k+t)} \le \begin{cases} Q_1^k E^{(0)}; & GJM, Q_1 = \|D^{-1}M\|, t=0 \\ Q_2^k E^{(0)}; & GSM, Q_2 = \|(D+L)^{-1}U\|, t=0 \\ Q_2 Q^{\left(\frac{k-1}{2}\right)} E^{(0)}; & CGJSM(I), t=1, k \text{ odd} \end{cases}$$

and by the above Lemma, we note that $Q_2 Q_1 < Q_2 < Q_1 < 1$ for the numerical test problems considered and in general of a *sdd* A, so that the dominance in convergence speed of CGJSM (I) may not be surprising any longer. Indeed, a graphical comparision of CGJSM (1), GJM and GSM is seen in fig (5.1) when the coefficient matrix A is *sdd* or weakly diagonally dorminant, where $E^{(0)}$ is the errror in the starting iterate. Further than the above, we remark that the algorithm CGJSM (I) is also convergent when A is irreducible and weakly diagonally dorminant. To see this, let G be the eigenvalue of the iteration matrix of CGJSM (I) then

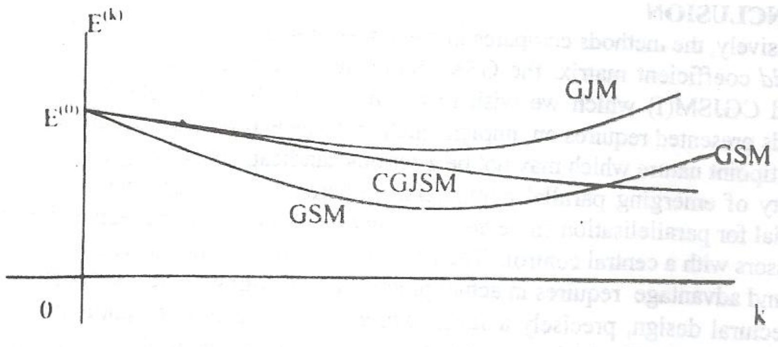


Fig. 5.1: A comparison of GJM, GSM and CGJSM(I)

$$\det\begin{bmatrix} D^{-1}M - \Gamma I & 0 \\ 0 & (D-L)^{-1}U - \Gamma I \end{bmatrix} \quad \det(D^{-1}M - \Gamma I)\det((D-L)^{-1}U - \Gamma I) = 0 \; ; \; M = L - U$$

Now, it is known that a weakly diagonally dorminant irreducible matrix is non-singular. Infact, by [21] it is that $\rho(D^{-1}M)<1$; $\rho\ (D+L)^{-1}U)<1$ which follows that the method converges on this condition of weak diagonal dorminancy and irreduciblity. The application of this is found on *problems (IV)-(VI)* from [19]. The numerical results are on Table (5.1) and see fig (5.1).

Problem (IV)

$$A = \begin{pmatrix} B_1 & D_1 & 0 & . & . & 0 \\ D_1 & B_2 & D_1 & 0 & . & 0 \\ 0 & D_1 & B_2 & D_1 & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & . & 0 & D_1 & B_3 & D_1 \\ 0 & . & . & 0 & D_1 & B_3 \end{pmatrix} \varepsilon R^{18x18}$$

$$B_1 = \begin{pmatrix} 2 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 2 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 2 \end{pmatrix}, B_2 = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}, B_3 = \begin{pmatrix} 3 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 3 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 3 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} -1 & 0 & 0 & \\ 0 & -1 & 0 & \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Problem (V)

$$A = \begin{pmatrix} B_1 & D_1 \\ B_2 & D_2 \end{pmatrix}, \varepsilon R^{18x18}; D_1 = \text{diag}(2,2,4,4,4,4,4,4,3),\ D_2 = \text{diag}(2,4,4,4,4,4,4,3,3)$$

$$B_1 = \begin{pmatrix} -\frac{1}{2} & -1 & 0 & . & . & . & . & . & 0 \\ -\frac{1}{2} & 0 & -1 & 0 & . & . & . & . & 0 \\ -1 & -1 & -1 & -1 & 0 & . & . & . & 0 \\ 0 & -1 & 0 & -1 & -1 & 0 & . & . & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 & . & 0 \\ 0 & . & 0 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & . & . & 0 & -1 & 0 & -1 & -1 & 0 \\ 0 & . & . & . & 0 & -1 & -1 & 0 & -1 \\ 0 & . & . & . & . & 0 & -1 & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix}; B_2 = \begin{pmatrix} -\frac{1}{2} & -\frac{1}{2} & -1 & 0 & . & . & . & . & 0 \\ -1 & 0 & -1 & -1 & 0 & . & . & . & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & . & . & 0 \\ 0 & -1 & 0 & -1 & -1 & 0 & . & . & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 & 0 & . & 0 \\ 0 & . & 0 & -1 & 0 & -1 & -1 & 0 & 0 \\ 0 & . & . & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & . & . & . & 0 & -1 & -1 & -1 & -1 \\ 0 & . & . & . & . & 0 & -1 & 0 & -\frac{1}{2} \end{pmatrix}$$

Problem (VI)

$$A = \begin{pmatrix} B_1 & D_1 & 0 & . & . & 0 \\ D_1^T & B_2 & E_1 & 0 & . & 0 \\ 0 & E_1^T & B_2 & E_1 & 0 & 0 \\ 0 & 0 & E_1^T & B_2 & E_1 & 0 \\ 0 & . & 0 & E_1^T & B_3 & E_2 \\ 0 & . & . & 0 & E_2^T & B_4 \end{pmatrix}, \varepsilon R^{18\_18}, B_1 = \begin{pmatrix} 2 & -1 & -\frac{1}{2} \\ -1 & 4 & 0 \\ -\frac{1}{2} & 0 & 2 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -\frac{1}{2} \end{pmatrix},$$

$$B_1 = \begin{pmatrix} 2 & -1 & -\frac{1}{2} \\ -1 & 4 & 0 \\ -\frac{1}{2} & 0 & 2 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -\frac{1}{2} \end{pmatrix}, B_2 = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

$$
B_1 = \begin{pmatrix}
-\frac{1}{2} & -1 & 0 & . & . & . & . & . & 0 \\
-\frac{1}{2} & 0 & -1 & 0 & . & . & . & . & 0 \\
-1 & -1 & -1 & -1 & 0 & . & . & . & 0 \\
0 & -1 & 0 & -1 & -1 & 0 & . & . & 0 \\
0 & 0 & -1 & -1 & 0 & -1 & 0 & . & 0 \\
0 & . & 0 & -1 & -1 & -1 & -1 & 0 & 0 \\
0 & . & . & 0 & -1 & 0 & -1 & -1 & 0 \\
0 & . & . & . & 0 & -1 & -1 & 0 & -1 \\
0 & . & . & . & . & 0 & -1 & -\frac{1}{2} & -\frac{1}{2}
\end{pmatrix}; B_2 = \begin{pmatrix}
-\frac{1}{2} & -\frac{1}{2} & -1 & 0 & . & . & . & . & 0 \\
-1 & 0 & -1 & -1 & 0 & . & . & . & 0 \\
0 & 0 & -1 & 0 & -1 & 0 & . & . & 0 \\
0 & -1 & 0 & -1 & -1 & 0 & . & . & 0 \\
0 & 0 & -1 & -1 & -1 & -1 & 0 & . & 0 \\
0 & . & 0 & -1 & 0 & -1 & -1 & 0 & 0 \\
0 & . & . & 0 & -1 & -1 & 0 & -1 & 0 \\
0 & . & . & . & 0 & -1 & -1 & -1 & -1 \\
0 & . & . & . & . & 0 & -1 & 0 & -\frac{1}{2}
\end{pmatrix}
$$

Problem (VI)

$$
A = \begin{pmatrix}
B_1 & D_1 & 0 & . & . & 0 \\
D_1^T & B_2 & E_1 & 0 & . & 0 \\
0 & E_1^T & B_2 & E_1 & 0 & 0 \\
0 & 0 & E_1^T & B_2 & E_1 & 0 \\
0 & . & 0 & E_1^T & B_3 & E_2 \\
0 & . & . & 0 & E_2^T & B_4
\end{pmatrix}, \varepsilon R^{18\_18}, B_1 = \begin{pmatrix}
2 & -1 & -\frac{1}{2} \\
-1 & 4 & 0 \\
-\frac{1}{2} & 0 & 2
\end{pmatrix}, D_1 = \begin{pmatrix}
0 & 0 & 0 \\
-1 & -1 & -1 \\
0 & -1 & -\frac{1}{2}
\end{pmatrix},
$$

$$
B_1 = \begin{pmatrix}
2 & -1 & -\frac{1}{2} \\
-1 & 4 & 0 \\
-\frac{1}{2} & 0 & 2
\end{pmatrix}, D_1 = \begin{pmatrix}
0 & 0 & 0 \\
-1 & -1 & -1 \\
0 & -1 & -\frac{1}{2}
\end{pmatrix}, B_2 = \begin{pmatrix}
4 & 0 & 0 \\
0 & 4 & 0 \\
0 & 0 & 2
\end{pmatrix}
$$

$$E_1=\begin{pmatrix}-1 & -1 & 0\\ 0 & -1 & -1\\ 0 & 0 & -1\end{pmatrix}, B_3=\begin{pmatrix}3 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 4\end{pmatrix}, E_2=\begin{pmatrix}-\frac{1}{2} & 0 & 0\\ -1 & -1 & 0\\ 0 & -1 & 0\end{pmatrix}, B_4=\begin{pmatrix}3 & 0 & -\frac{1}{2}\\ 0 & 4 & -1\\ -\frac{1}{2} & -1 & 3\end{pmatrix}$$

Problem (VII)

$$E_1=\begin{pmatrix}D_1 & F_1 & 0 & . & . & 0\\ F_1 & D_1 & F_2 & 0 & . & 0\\ 0 & D_1 & D_2 & F_3 & 0 & 0\\ 0 & 0 & F_3^T & D_3 & F_4 & 0\\ 0 & . & & F_4^T & D_5 & F_5\\ 0 & . & . & 0 & F_5^T & D_5\end{pmatrix}, D_1=\begin{pmatrix}10 & -2 & 0\\ -2 & 20 & -2\\ 0 & -2 & 10\end{pmatrix}, F_1=\begin{pmatrix}-4 & -1 & 0\\ -1 & -4 & -1\\ 0 & -1 & -4\end{pmatrix},$$

$$F_2=\begin{pmatrix}20 & -4 & 0\\ -4 & 20 & -4\\ 0 & -4 & 20\end{pmatrix}, \quad F_3^T=\begin{pmatrix}-4 & -1 & 0\\ -1 & -4 & -1\\ 0 & -1 & 4\\ 0 & 0 & -1\end{pmatrix}, F_4=\begin{pmatrix}-4 & -1 & 0 & 0\\ -1 & -4 & -1 & 0\\ 0 & -1 & -4 & -1\\ 0 & 0 & -1 & -4\end{pmatrix},$$

$$F_5=\begin{pmatrix}-1 & -4 & -1 & 0 & 0\\ 0 & -1 & -4 & -1 & -0.5\end{pmatrix}, D_5=\begin{pmatrix}20 & -4\\ -4 & 10\end{pmatrix},$$

$$D_3=\begin{pmatrix}20 & -4 & 0 & 0 & \\ -4 & 20 & -4 & 0 & \\ 0 & -4 & 20 & -4 & \\ 0 & 0 & 0 & -1 & 20\end{pmatrix}, D_4=\begin{pmatrix}21 & -4 & 0 & 0 & \\ -4 & 20 & -4 & 0 & \\ 0 & -4 & 19 & -4 & \\ 0 & 0 & 0 & -4 & 10\end{pmatrix},$$

The **b** are normalised as in Problem (I) to have the solution as **x**=$(1,1,1,...,1)^T$.

Table 5.1: Error upper bound

$$E^{(k)} \leq W^{f(k)}\, E^{(0)}\,;\, W = \rho(G) < 1$$

	(IV)	(V)	(VI)	(VII)
GSM	$(0.6194)^k E^{(0)}$	$(0.7273)^k E^{(0)}$	$(0.7012)^k E^{(0)}$	$(0.7884)^k E^{(0)}$
GJM	$(0.7983)^k E^{(0)}$	$(0.8187)^k E^{(0)}$	$(0.8288)^k E^{(0)}$	$(0.8646)^k E^{(0)}$
CGJSM(I)	$(0.4945)^{\frac{k+1}{2}} E^{(0)}$	$(0.5954)^{\frac{k+1}{2}} E^{(0)}$	$(0.5807)^{\frac{k-1}{2}} E^{(0)}$	$(0.6817)^{\frac{k-1}{2}} E^{(0)}$; *k odd*

From table (5.1), the error bounds of CGJS(I) decay faster than the GSM and GJM. Infact, this shows the faster convergence rate of CGJSM(I). In particular, for problem (VI) we shall have that

$$E^{(k)} \leq \begin{Bmatrix} (0.7012)^k E^{(0)} & GSM \\ (0.8282)^k E^{(0)} & GJM \\ (0.5807)^k E^{(0)} & CGJSM, \quad k \quad odd \end{Bmatrix} \qquad f(k) = \begin{Bmatrix} k & GSM, \quad GJM; \quad k = 1,2,3,.. \\ \dfrac{k+1}{2} & CGJSM, \quad k \quad odd \end{Bmatrix}$$

REFERENCES

[1] Burden R.L., Faires J.D. and Reynold A.C. ; (1981): Numerical Analysis. Prindle, Weber and Schmidt, Boston, USA.

[2] Chartier P.; Parallelism in the Numerical Solution of IVPs and DAEs. Ph.D. thesis University of Rennes, Rennes, France, (1993).

[3] David, R.H.; Experiments in Computational Matrix Algebra. The Random House/Birkhauser Mathematics Series, (1998).

[4] Eikemo, M.S., Espedal, M.S. and Fladmark, G.; On the Numerical Solution of a Three Dimensional Extrusion Model. *Computing and Visualisation in Science*, Vol. 1, pp. 1 - 14, Springer-Verlag, (1997).

[5] Evans, J.D.: Parallel Numerical Algorithms for Linear Systems. Parallel Processing Systems (ed D. Evans). Cambridge University Press, (1982).

[6] Evans, D.J. and Okeke, C.C. : The modified preconditioned Jacobi Methods for Iterative Solution of Linear Systems of Equations. *Intern. J. Of Computer Mathematics*, Vol. 44. Pp.53-69, (1992) .

[7] Gander, W.; and Hrebicek, J.; Solving Problems in Scientific Computing Using MAPLE and MATLAB. Springer-Verlag, (1998).

[8] Golub, G.H and Van Loan, C.F.; Matrix Computations. North Oxford Academic, Oxford, (1983).

[9] Hackbusch, W. and Sauter, S.A., Composite Finite Elements for Problems Containing Small Geometric Details. part II: Implementation and Numerical Results. *Computing and Visualisation in Science*, Vol.1, pp. 15-25, Springer-Verlag, (1997).

[10] Ikhile, M.N.O: Multipoint Methods for Large Sparse System of Linear Equations. Submitted to *International Journal of Computer and Mathematics with Applications*, U.S.A. (1999).

[11] Ikhile M.N.O; Multipoint Iteration Methods of Gauss-Seidel Type for Sparse System of Linear Equations. Submitted to *Benin Journal of Mathematical Sciences*, (1999).

[12] Ikhile M.N.O; Multipoint Iteration Methods for Sparse System of Linear Equations. Journal of the *Nigerian Association of Mathematical Physics*, Vol. 3 , 89-105 , (1999).

[13] Ikhile M.N.O; A New Parallel Cyclic Implementation of Iterative Methods for Large Sparse System of Linear Equations. Submitted to *International Journal of Computer Mathematics,* U.K. (1999).

[14] Modi, J.J.: Parallel Algorithms and Matrix Computation. Oxford Applied Mathematics and Computing Science Series. Clarendon Press, Oxford, (1988).

[15] Matheij, R.M.M.: Direct Solution of certain sparse Linear System. Computational Ordinary Differential Equations (ed. S.O. Fatunla), 1992.

[16] Ralston, A. and Rabinowitz, P.; A First Course in Numerical Analysis. Mcgraw-Hill Intern. Book Company, (1983).

[17] Reid, J.K.; Large Sparse Sets of Linear Equations. Proceedings of the Oxford

Conference of the Institute of Mathematics and its Applications, Academic Press (1971).

[18] Van Norton, R.; The Solution of Linear Equations by the Gauss-Seidel Method. Mathematical Methods for Digital Computers, (ed. Ralston, A. and Wilf, H.S.) John Wiley and Sons, Inc. New York, (1967).[19] Schwarz, H.R., Rutishauser, H. and Stiefel, E.; Numerical Analysis of Symmetric Matrices. Prentice-Hall, Inc. Englewood Cliffs, N.J., (1973).

[20] Sheldon, J.W.; Iterative Methods for the Solution of Elliptic Partial Differential Equations. Mathematical Methods for Digital Computers (ed. Ralston, A and Wilf, H.S.). John Wiley and Sons Inc., (1971).

[21] Young M.D. and Gregory T. R.; A Survey of Numerical Mathematics: Vol. I and II. Addison-Wesley Publishing Comp. Inc. (1972).