MODIFIED STRONGLY IMPLICIT METHOD

by

Avi Lin

Technical Report #271
April 1983
Abstract

A new iterative technique, the Modified Strongly Implicit Method, for solving convection-diffusion equations is described and demonstrated. This method is implicit and has two distinct features: (a) the source term of the algorithm is affected implicitly by all the boundaries, (b) the variables are solved implicitly along diagonals. It is shown that this technique is better and more efficient than the usual Strongly Implicit and ADI methods.
# TABLE OF CONTENTS

1. Introduction .......................................................... 6
2. Finite Difference Equations ........................................... 8
3. Matrix Representation of the Problem ................................. 13
4. The Factorization Technique ........................................... 15
5. The Strongly Implicit (SI) Method .................................... 21
6. The Modified Strongly Implicit (MSI) Methods ....................... 24
7. The Modified Gauss-Seidel (GS) Method ............................... 27
8. Examples ............................................................... 30
9. The MSI Method Features .............................................. 34
10. Conclusions ........................................................... 42
LIST OF TABLES AND FIGURES

Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Equations to be Used by SI, MSI1 and MSI2 in the Two Basic Sweeps</td>
<td>26</td>
</tr>
<tr>
<td>II</td>
<td>Convergence Results</td>
<td>32</td>
</tr>
</tbody>
</table>

Figure

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Region on Which the Elliptic Equation is Solved</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Matrix Presentation of the Algebraic Equations</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Schematic description of SI, MSI1 and MSI2 Methods</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Model Problem [6]</td>
<td>31</td>
</tr>
<tr>
<td>5</td>
<td>Convergence Curves for the Example Problem, eg. (41)</td>
<td>40</td>
</tr>
</tbody>
</table>
1. Introduction

This paper is concerned with the solution of the system of algebraic equations that result from the discretization of partial differential equations of the elliptic form. Typical two and three dimensional problems require the evaluation of several variables at a large number of grid points. Usually, this system of equations is not solved by direct methods; as the number of grid points is increased, direct methods become less efficient (in terms of CPU consumption and computer core requirement) and in many cases totally impractical. The matrix that describes the coefficients for the algebraic equations are sparse and regular, and therefore iterative solutions are easier to implement as well as being more economical. The Jacobi iterative method [1], the Gauss-Seidel iterative method [1], success-over-relaxation (SOR) [2], and alternating direction implicit method [3a, 3b], cyclic Chebyshev acceleration [4], and the strongly implicit method [5] are among some of the current iterative techniques typically considered. Methods that involve some optimal accelerating technique, e.g., gradient methods, are not counted here, as they can be added or adjusted to any of the above "basic" methods. Once an iterative method has been selected by showing that it has better convergence properties for certain problems than other similar methods, it is adopted as a reasonable method for solving the general class of problems, without knowing in advance if this technique is convergent for the appropriate system of algebraic equations.
Almost any iteration method can be presented as a time like
marching technique where every iteration is analogue to one
time step of marching. Although the basic finite difference
approximation for the partial differential equations is formulated
in a "stable manner", i.e. the diagonally dominance of the
algebraic system is recovered and sometimes subjected to certain
conditions, the chosen iteration method for solving this system
is not necessarily convergent. In this paper, two modifications
of the Strongly Implicit method [5] will be presented. The only
necessary and sufficient conditions for convergence of this
method is the development of a "stable finite difference"
formulation of the differential equation. A sample two-dimensional
equation, similar to that of Helliwell [6] is used to demonstrate
the methods and to make comparisons. Also, comparisons are
made to some accelerating methods [19] on some large scale model
problem.
2. **Finite Difference Equations**

Before choosing an iteration method for solving a particular elliptic or parabolic problem, a finite difference discretization for all the derivatives appearing in the differential equations must be selected. The iteration technique is then designed to solve the resultant algebraic system of equations. It is clear that the chosen difference expressions for the derivatives will be governed by the required accuracy and stability considerations for the particular solution technique. Therefore, each derivative must be considered separately (similar derivatives do not have to have similar difference forms). In most cases, it is desired, from the stability requirements, that the finite difference scheme be as implicit as possible. [7]. Before describing the special implicit forms to be discussed in this paper, let us define the problem to be solved.

The iterative numerical schemes in this study will be demonstrated on the following two-dimensional, linear, convection-diffusion partial differential equation [5]:

\[
\frac{\partial \phi}{\partial t} + a(x,y) \frac{\partial \phi}{\partial x} + b(x,y) \frac{\partial \phi}{\partial y} = A(x,y) \frac{\partial^2 \phi}{\partial x^2} + B(x,y) \frac{\partial^2 \phi}{\partial y^2} + S(x,y) \tag{1}
\]

where:
- \( t \) is a time-like coordinate (presents the iterations)
- \( x, y \) are the space coordinates
- \( A, B > 0 \) are the diffusion-like coefficients
- \( a, b \) are the convection-like coefficients
- \( S \) is a known source field.

and where the only solution of eq. (1) that is interested is the convergence solution, i.e., \( \partial \phi / \partial t = 0 \).
For solving Eq. (1) over the region $\Omega$ in the $(x,y)$ plane, a suitable boundary condition for $\phi$ on $\Omega$ boundaries and initial conditions at $t = 0$ throughout $\Omega$ must be specified. For simplicity, a unit square region $\Omega$ will be considered; a uniform grid is spread over this region as described in Fig. 1.

If $m$ is the number of points in the $x$ direction and $n$ is the number of points in the $y$ direction, the space coordinates are discretized by:

$$h_1 = \frac{1}{m-1}; \quad x_i = (i-1)h_1, \quad 1 \leq i \leq m$$

$$h_2 = \frac{1}{n-1}; \quad y_j = (j-1)h_2, \quad 1 \leq j \leq n$$

and the time is discretized by

$$t_l = l \cdot k, \quad l = 0, 1, 2, \ldots, \quad k = \text{time step (constant)}.$$

The following finite difference expressions might prescribe the $\phi$ derivatives appearing in Eq. (1):

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{l+1}_{i,j} - \phi^l_{i,j}}{k} + O(k) -- a \text{ first order forward difference for marching in time}$$

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi^{l+1}_{i+1,j} - 2\phi^{l+1}_{i,j} + \phi^{l+1}_{i-1,j}}{h_1^2} + O(h_1^2) -- a \text{ second order central difference}$$

and a similar formula for $\partial^2 \phi/\partial y^2$. For the first derivatives $\partial \phi/\partial x$ and $\partial \phi/\partial y$ we shall choose the KR method [8]:

...
FIG. 1. THE REGION ON WHICH THE ELLIPTIC EQUATION IS SOLVED.
\[
\frac{\partial \phi}{\partial x} = \frac{\alpha}{h_1} (\phi_{i+1,j}^{l+1} - \phi_{i,j}^{l+1}) + \frac{1-\alpha}{h_1} (\phi_{i,j}^{l+1} - \phi_{i-1,j}^{l+1}) \\
+ \frac{(-1)^\alpha}{2h_1} (\phi_{i+1,j}^{l} - 2\phi_{i,j}^{l} + \phi_{i-1,j}^{l})
\]

where \( \alpha = \begin{cases} 
1 & \text{for } a \leq 0 \\
0 & \text{for } a > 0
\end{cases} \)

A similar equation is written for \( \frac{\partial \phi}{\partial y} \). In [8], it is proved that substitution of Eqs. (3, 4, 5) into Eq. (1) gives a finite difference approximation which is unconditionally stable and consistent. At every point \((i,j)\) of the two dimensional domain, the following system of algebraic equations for \( \phi_{i,j}^{l+1} \) can be written:

\[
E_{i,j} \phi_{i+1,j}^{l+1} - W_{i,j} \phi_{i-1,j}^{l+1} + N_{i,j} \phi_{i,j+1}^{l+1} + S_{i,j} \phi_{i,j-1}^{l+1} \\
+ P_{i,j} \phi_{i,j-1}^{l+1} + D_{i,j} = 0
\]

where the coefficients of the matrices \( E, W, N, S, P \) and \( D \) are:

\[
E_{i,j} = \frac{1}{h_1} \left[ a_{i,j} \phi_{i,j}^{l} - \frac{A_{i,j}}{h_1} \right] ; \quad W_{i,j} = -\frac{1}{h_1} \left[ (1-\alpha) a_{i,j} \phi_{i,j}^{l} + \frac{A_{i,j}}{h_1} \right]
\]

\[
N_{i,j} = \frac{1}{h_2} \left[ \beta_{i,j} \phi_{i,j}^{l} - \frac{B_{i,j}}{h_2} \right] ; \quad S_{i,j} = -\frac{1}{h_2} \left[ (1-\beta) b_{i,j} \phi_{i,j}^{l} + \frac{B_{i,j}}{h_2} \right]
\]
\[ P_{i,j} = -(E_{i,j} + W_{i,j} + N_{i,j} + S_{i,j}) + \frac{1}{k} \]

\[ D_{i,j} = \frac{(-1)^\alpha}{2h_1} (\phi_{i+1,j}^\ell - 2\phi_i^\ell + \phi_{i-1,j}^\ell) + \frac{(-1)^\beta}{2h_2} (\phi_{i,j+1}^\ell - 2\phi_{i,j}^\ell + \phi_{i,j-1}^\ell) - \frac{\phi_{i,j}^\ell}{k} \]  

(7)

where \( \beta \) plays the same role for the y direction as \( \alpha \) does for the x direction in Eq. (5).
3. Matrix Representation of the Problem

For convenience in describing the development of the solution procedure, the algebraic equation (6) is written in the matrix form:

\[
[A] \bar{\phi} + M = 0
\]  

(8)

The expansion of \([A]\) in Eq. (8) is given in Fig. 2. If a rectangular field of \(M = (m)\) by \(N = (n)\) points is to be solved, the matrix \(A\) is a square array, \(K = M \times N\), composed of the coefficients appearing in Eq. (7) in sequenced equations. The vector \(\phi\) is composed of sequenced variables, and the vector \(M\) is the sequenced source terms. Each row in \(A\) consists of the coefficients of the unknown variables in one equation of the set. The subscripts \(i,j\) refer to the point \(i,j\) in the grid system used in setting up Eq. (6), rather than a location within \(A\). As was mentioned before, solving Eq. (8) directly is quite difficult, even with available direct techniques [9], since they are not presently useful for the general finite-difference form [Eq. (6)]. The only reasonable way to solve Eq. (8) is by iterative methods. The iterative techniques that will be discussed here fall under the general category of "Factorization Methods".

13
FIG. 2. MATRIX PRESENTATION OF THE ALGEBRAIC EQUATIONS.
4. The Factorization Technique

Generally, it can be shown that the origin of any implicit iterative method for solving Eq. (8), is in some factorization of $A$. Ideally, for solving Eq. (8), one might try to factor the matrix $A$ into two matrices $L$ and $U$ [1]:

$$A = L \cdot U$$  \hspace{1cm} (9)

where $L$ is a lower triangular matrix and $U$ is an upper triangular matrix. It is generally very difficult, and most of the times it is impossible to find an exact factorization as in Eq. (9). That is since the $L$ and $U$ of the exact $LU$ decomposition of $A$ are usually less sparse than $A$, it would require a considerably large amounts of CPU time and computer direct access memory (storage) to find such a factorization [12]. Therefore, the iterative factorization method is obtained by replacing the sparse matrix $A$ by a modified form $A + G$ such that the resulting matrix is sparse, and can be decomposed by the $L \cdot U$ decomposition of Eq. (9), where $L$ and $U$ are as sparse as $A$. The iterative method can then be formulated as

$$(A + G) \cdot \phi^{n+1} + M - G \cdot \phi^n = 0$$  \hspace{1cm} (10)

where the superscript $n$ denotes the iteration number. In some other methods it is preferred to present the iterative procedure as follows [13]:

$$(A + G) \cdot \phi^{n+1} = (A + G) \cdot \phi^n - \tau (A \cdot \phi^n + M)$$  \hspace{1cm} (11)
where $\tau$ is, generally, a sequence of parameters that will cause to a rapid convergence of the series $\phi^n$. It turns out [14, 15] that only if $A + \zeta$ is positive defined (and then the eigenvalues of $(A + \zeta)^{-1}[(1-\tau)A + \zeta]z^{-1}$ are positive), Chebysev minimax theory can be applied to find the optimal sequence $\{\tau_n\}$, that will minimize the error $(\phi^{n+1} - \phi^n)$ in the $L_2$-norm. Here, $\{\tau_n\}$ are depending on the eigenvalues of $(A + \zeta)^{-1}[(1-\tau)A + \zeta]z^{-1}$. Although several techniques for evaluating $\{\tau_n\}$ on this basis has been developed [16] this procedure is not applied in the present study since the considered system matrix $A$ is not symmetric due to the convection like terms in eq. (1). Based on the Chebysev polynomials, an algorithm has been developed for accelerating the convergence for nonsymmetric matrix $A$ [17], but this method is still two parametric techniques which depends on the $A$'s eigenvalues structure. In the present study only $\tau = 1$ will be considered, where all the other accelerating procedures (like Chebysev [17], conjugate gradient [18]) can be applied to the method that will be presented hereafter, for accelerating its rate of convergence. Since the rate of convergence of the iteration scheme described in eq. (10) depends on the particular choice of the matrix $\zeta$, the elements of $\zeta$ should be as small in magnitude as possible (comparing to those of $A$) to minimize the explicit perturbation and to maximize the rate of convergence [12] (for $\zeta = 0$ the solution is obtained within one iteration). There are an infinity of choices of the matrix $\zeta$ for which the LU factors will be sparse. In the present study we
shall be concerned with one family of possibilities in order to factor \( A \) as in Eq. (9); this family is obtained after adding a matrix \( G \), defined by:

\[
\begin{align*}
\left[ G\phi \right]_{i,j} &= Q_{i,j} \phi_{i+1,j-1} + R_{i,j} \phi_{i-1,j+1} \\
&+ \alpha [Q_{i,j} (\phi_{i,j} - \phi_{i+1,j} + \phi_{i,j-1}) \\
&+ R_{i,j} (\phi_{i,j} - \phi_{i-1,j} - \phi_{i,j+1})]
\end{align*}
\]  

(12)

where here the \((i,j)\) indices refer to the point \((i,j)\) in the grid system. This definition of \( G \) is analogous to the definition of "Strongly Implicit" given in [13]: \( A + G \) is said to be strongly implicit if the non-zero elements of \( A \) correspond to the non-zero elements of \( A + G \). Equation (12) is a special case of the family of \( G \) under this definition with the restriction that \((G\phi)_{i,j}\) will be defined within the numerical box \([i+1, j+1]\).

The strongly implicit method, as was first presented by H.L. Stone [5], and have two stages of development.

The first stage was by formulating the additional \( (G\phi)_{i,j} \) terms as in Eq. (12) with \( \alpha = 0 \). In order to get some flexibility and control over the rate of convergence, Stone introduced in the second stage, a parameter \( \alpha \), multiplying terms that are, as it is described in his paper, small and partially cancelled. The second and third lines of Eq. (12) describe the additional terms that were considered and the parameter \( \alpha \). Those terms were added to the left hand side of the basic elliptic difference equation to be solved, Eq. (6). This addition, which leads to the second
order factorization [13], is only first order accurate and since artificial viscosity is introduced into the system, the converged solution is for a slightly different problem. This point was indirectly mentioned also in [19] in number 10 of his list of references. In [13], second order strongly implicit is suggested. This procedure has also some free parameters, but will fail on a non equal spaced grid points. In the present approach the Stone's "a" terms are added to the G definition rather than to the original equation and the second order accuracy is retained. This formulation is a non-interesting special case of the parametrized strongly implicit technique [20],** that does not lead to any improvement in the rate of convergence.

From here on \( \alpha = 0 \) is assumed. There are three other ways to define the matrix \( \tilde{G} \) where the \( \phi \) 's in eq. (12) are on the opposite corners of the basic numerical box, but we shall concentrate, for simplicity, on the possibility defined by eq. (9), without losing generality. Not all pairs of matrices \( \tilde{Q} \) and \( \tilde{R} \) can lead to factorization of Eq. (9), because they must fulfill some specific relations between the matrices \( \tilde{E}, \tilde{W}, \tilde{N}, \tilde{S}, \) and \( \tilde{P} \) in Eq. (7). However, if one can find appropriate \( \tilde{Q} \) and \( \tilde{R} \), then \( \tilde{L} \) and \( \tilde{U} \) has only three non zero elements in each row regardless of the size of \( M \) and \( N \).

** This special case (with Ref. [20] notations) is:

\[
a(\tilde{Q}_{i,j} + \tilde{R}_{i,j}) = D[(1-\varepsilon_w)w_{i-1,j} + (1-\varepsilon_s)s_{i,j-1}]\cdot
\]
To show how to construct the matrices $Q$ and $R$, let us sequence $\phi_{i,j}$ in the $i$ direction. In this way, the matrix $A$ has three elements in each row that are located on the principle diagonal $(P_{i,j})$ and the two adjacent diagonals $(E_{i,j}, W_{i,j})$. The other two elements $N_{i,j}$ and $S_{i,j}$ are located on diagonals situated $M$ locations away, on each side of the principle diagonal, as shown in Fig. 2 for $\alpha = 0$. This figure also provides the definition of $L$ and $U$. It is shown that $L$ has nonzero elements on the diagonals corresponding to the $S$, $W$ and $P$ diagonals of $A$, and $U$ has nonzero elements on those diagonals corresponding to the $P$, $E$ and $N$ locations of $A$. By this arrangement, we have one degree of freedom to choose the principle diagonal of $U$ as 1:

$$\text{diag}(U) = 1$$

By multiplying $L \cdot U$ and setting this equal to $A + G$, the following relations may be obtained (see Fig. 2):

(a) $H_{i,j} = S_{i,j}$

(b) $K_{i,j} = W_{i,j}$

(c) $D_{i,j} = 1/(P_{i,j} + X_{i,j-1}S_{i,j} + Y_{i-1,j}W_{i,j})$

(d) $X_{i,j} = -D_{i,j}N_{i,j}$

(e) $Y_{i,j} = -D_{i,j}E_{i,j}$ (13)

where

(f) $O_{i,j} = -S_{i,j}Y_{i,j-1}$

(g) $R_{i,j} = -W_{i,j}X_{i-1,j}$
Equations (12) are the basis for the methods described in the present study. Before describing these procedures, let us briefly review the Strongly Implicit Method [for solving eq. (9)] which is also based on Eqs. (12).
5. The Strongly Implicit (SI) Method

The basic SI method as described by Stone [5] can be derived from the factorization algorithm and Eq. (13). The recursion formula of the method in this case is:

\[ \phi_{i,j} = X_{i,j} \phi_{i,j+1} + Y_{i,j} \phi_{i+1,j} + C_{i,j} \]  

(14)

where the matrices \( X \) and \( Y \) are defined in Eqs. (13d, 13e), and the matrix \( C \) is:

\[ C_{i,j} = -D_{i,j}[M_{i,j} + W_{i,j}C_{i-1,j} + S_{i,j}C_{i,j-1}] \]

\[- R_{i,j} \phi_{i-1,j+1} - Q_{i,j} \phi_{i+1,j-1} \]  

(15)

If eq. (1) has to be solved with Dirichlet boundary conditions, then the boundary conditions for \( X \), \( Y \) and \( C \) in Eq. (13) and Eq. (15) will be:

\[ X = Y = 0 \quad \text{along the boundaries} \]

\[ C = \phi \quad \text{along the boundaries} \]

The SI iteration loop is as follows: beginning at one of the four corners (say at the point \((1,1)\)) and going up to the opposite corner (point \((m,n)\)) the coefficient matrices \( X \) and \( Y \) (from Eq. (13d, 13e)) and \( C \) (from Eq. (15)) are evaluated with the help of the last solution of the \( \phi \) field. Going back from the last corner (point \((m,n)\)) to the first corner (point \((1,1)\)) and new \( \phi \) field is computed by Eq. (14).

* In his paper, eqs. (5), (6), (7) or eq. (11) with \( \alpha = 0 \).
This procedure has the distinct advantage of being implicit in both direction i and j as well as coupling all the boundary conditions. In [10] it was found for the coupled stream function-vorticity equations that this technique generally converged faster than many of more familiar, less implicit, iterative methods like ADI or line relaxation among the others. Boundary condition coupling was apparently quite strong. It is the present author's opinion (after solving many two dimensional linear elliptic equations) that the SI method converges more rapidly than the ADI method in some cases but not always. For a given equation (say the heat conduction equation) with certain boundary conditions, the rate of convergence of SI was slower than that of the ADI and for different boundary conditions the SI was faster. Since it is hard to include the boundary conditions in the theoretical analysis of the rate of convergence, it is difficult to tell before hand which method will converge faster for a given application. In [11] a similar conclusion was reached. The effect of boundary condition coupling was not considered. In the present one-variable analysis we shall present some modifications of this technique that significantly improve the rate of convergence of the SI method, much beyond the ADI technique, even for simple boundary conditions. No other extra steps for accelerating the rate of convergence will be discussed here. Important work has been done concerning the combination of the SI method with some accelerating methods. For example, in [12], [18], [19], [21], the conjugate gradient method was applied to accelerate the rate of convergence of the SI technique. The
special construction of the eigenvalues of the iteration matrix is the main reason to the success of this coupling. Other methods may be found in [13], [17]. In the present paper, we shall limit our study to the new technique itself, and by his desire, one can couple the present algorithm with any other accelerating technique.
6. The Modified Strongly Implicit (MSI) Methods

In this section we shall present the MSI methods in a very simple way. The MSI method is obtained by adding one step of calculations to the standard SI method. Equations (14) and (15) can be written as follows:

\[
(D_{i,j}, R_{i,j})\phi_{i-1, j+1} + \phi_{i, j} + (D_{i,j}, Q_{i,j})\phi_{i+1, j-1} \\
= -D_{i,j} \left[ M_{i,j} + W_{i,j} C_{i-1, j} + S_{i,j} C_{i, j-1} \right] \\
+ X_{i,j} \phi_{i, j+1} + Y_{i,j} \phi_{i+1, j} \tag{16}
\]

Like in the SI technique, the first sweep of the MSI iteration is accomplished by marching along the line that connects two opposite corners [say the line between point \((1,1)\) and the point \((m,n)\)]. This line, which will be named line \(I\), is intersected with \((m+n-2)\) diagonals (in the present example, diagonals of the form \(i+j = \text{constant}\), see Fig. 3). With the SI technique the coefficients \(C_{i,j}\) are calculated along such a diagonal by Eq. (15). With the MSI technique a preliminary step is taken before evaluating \(C_i\) using Eq. (16), \(\phi_{i,j}\) is solved implicitly (by a tri-diagonal inversion), treating the \(\psi\)'s on the next diagonal (in our example, \(\phi_{i+1, j}\) and \(\phi_{i, j+1}\)) as known and then evaluating the coefficients \(C_{i,j}\) along this diagonal from Eq. (15) like with the standard SI method. The solution for \(\phi\) is accomplished in the second sweep of this iteration in the same way like in the SI method by marching along the line \(I\), in the opposite direction [in the present example, the line that connects...
the point \((m,n)\) with the point \((1,1)\). For every point on every diagonal that intersects with this line, \(\phi_{i,j}\) is solved by Eq. (14). This is the first modification of the SI method and will be named MSI1. Applying this philosophy for the first part of MSI1 method, but modifying the second part, a different second sweep can be formulated: instead of solving \(\phi\) from Eq. (14) \(\phi_{i,j}\) is solved implicitly along the diagonal using Eq. (16), and \(C_{i,j}\) is updated by the following equation:

\[
C_{i,j} = \phi_{i,j} - X_{i,j}\phi_{i,j+1} - Y_{i,j}\phi_{i+1,j}
\]

which follows from Eq. (14). The MSI1 method with this modification will be named MSI2. The SI method and the two modifications MSI1 and MSI2 methods are described in Fig. 3, and summarized in Table I.

Intuitively, a "theoretical" improvement of the MSI methods on the SI method can be explained as follows: The MSI methods are more implicit than the SI method and therefore the effect of the two sides boundaries is felt immediately for all the implicit diagonal inversions. It can be said that the MSI technique is more implicit when compared with the SI technique; in the same manner that the ADI method is more implicit than the Gauss Seidel method; the ADI and the MSI algorithms can be derived by treating part of the source terms of the GS and SI techniques respectively, as unknown during the iteration. In the following section, we shall present a GS method as a special case of the SI technique. This method will be considered as a reference method in order to demonstrate the advantages of the MSI techniques.
<table>
<thead>
<tr>
<th>Sweep/Method</th>
<th>SI</th>
<th>MSI1</th>
<th>MSI2</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward sweep</td>
<td>eq. 15</td>
<td>(a) eq. 16 (b) eq. 15</td>
<td>(a) eq. 16 (b) eq. 15</td>
</tr>
<tr>
<td>backward sweep</td>
<td>eq. 14</td>
<td>eq. 14</td>
<td>(a) eq. 16 (b) eq. 17</td>
</tr>
</tbody>
</table>

Table 1: Equations to be Used by SI, MSI1 and MSI2 in the Two Basic Sweeps.
7. The Modified Gauss Siedel (MGS) Method

The Modified Gauss Siedel (MGS) method will serve here as a reference method to the SI and MSI methods for comparison of results and rate of convergence. The MGS method was chosen as this basic iteration procedure is like that of the SI method, but its formulation is much simpler. As with the SI method the MGS method also consists of two sweeps. For the example given in section 5 for the SI method, the two MGS sweeps will be formulated from Eq. (6) as follows:

the backward sweep:

\[
\phi^*_i, j = -\left(\frac{N_{i,j}}{P_{i,j}}\right)^{k-1} \phi_{i,j+1} - \left(\frac{E_{i,j}}{P_{i,j}}\right)^{k-1} \phi_{i+1,j} - \left(\frac{W_{i,j}}{P_{i,j}}\right)^{k-1} \phi_{i-1,j} - \left(\frac{D_{i,j}}{P_{i,j}}\right)^{k-1} \phi_{i,j-1}
\]

and the forward sweep:

\[
\phi_i, j = -\left(\frac{N_{i,j}}{P_{i,j}}\right) \phi^*_{i,j+1} - \left(\frac{E_{i,j}}{P_{i,j}}\right) \phi^*_{i+1,j} - \left(\frac{W_{i,j}}{P_{i,j}}\right) \phi^*_{i-1,j} - \left(\frac{D_{i,j}}{P_{i,j}}\right) \phi^*_{i,j-1}
\]

where \( k \) is the iteration number and * are the intermediate values.

This method is similar to the SI method formulated in Eqs. (14, 15). The two main differences between the SI method and MGS method are:

\[
\text{Technion - Computer Science Department - Technical Report CS0271 - 1983}
\]
(i) The number of basic calculations (additions, multiplications, ...) per iteration in the MGS method is less than in the SI method.

(ii) There is not any required computer core for the MGS method, while for the SI method there should be at least three MxN matrices.

The MGS method is described schematically in Fig. 3. It can be shown that the MGS method can be obtained from the general presentation stated by Eqs. (11) and (12) by assuming

\[ \text{diag}(U) = D_{i,j} \quad (20) \]

The MGS, MSI and SI methods present three different possibilities of the factorization family defined by Eq. (10). In the next section, we attempt to compare the features of these three methods by solving some sample problems.
Only points that are involved in the scheme are emphasized in this figure. In the first sweep, all the variables that are appearing to the right of the point in the figure are involved in the first equation of this sweep, and those that are appearing to the left of the point are involved in the second equation of this sweep. In the second sweep, it is opposite to the first one.

FIG. 3. SCHEMATIC DESCRIPTION OF SI, MSI1 AND MSI2 METHODS.
8. Example

The three aforementioned methods have been applied to the steady form of Eq. (1), with various functions for the four coefficients. The comparisons between the methods include the rate of convergence as a function of the work required. The field and the boundary conditions are described in Fig. 4. The initial guess is zero everywhere. The rate of convergence figures are summarized in Table II for various coefficients. For all the cases a $31 \times 31$ field was investigated. It can be seen that MSI2 is an improvement over MSII. MSG is the poorest and is twice as slow as the standard SI. The SI method is itself slower than any of the MSI methods. For a very stiff case $[A = 0.1, B = 0.1, a = 5 \sin (2\pi x), b = 5 \sin (2\pi y)]$ the SI and MGS are fairly close, while for a diffusion dominated case ($a, b$ small) the SI and MSI methods are close and the rate of convergence of MGS is about one third of all the others.

One possibility for measuring the efficiency of an iterative method is to define $\eta$ as:

$$\eta = \frac{\text{rate of convergence}}{\text{work required}} \quad (21)$$

where a reasonable definition for the amount of work required is

$$\text{work required} = \text{(storage requirement)} \times \text{(CPU in seconds)}$$
FIG. 4. MODEL PROBLEM (6)
TABLE II. CONVERGENCE RESULTS

<table>
<thead>
<tr>
<th></th>
<th>A=B=1</th>
<th>A=1, B=0.1</th>
<th>A=1.0</th>
<th>B=0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a=b=0$</td>
<td>$a=\sin (2\pi X)$</td>
<td>$a=0.1 \sin (2\pi X)$</td>
<td>$a=5\sin (2\pi X)$</td>
</tr>
<tr>
<td>SI</td>
<td>$10^{-1.7}$</td>
<td>$10^{-3.1}$</td>
<td>$10^{-6.2}$</td>
<td>$10^{-0.71}$</td>
</tr>
<tr>
<td>SI1</td>
<td>$10^{-2.3}$</td>
<td>$10^{-3.8}$</td>
<td>$10^{-6.8}$</td>
<td>$10^{-1.3}$</td>
</tr>
<tr>
<td>SI2</td>
<td>$10^{-3.1}$</td>
<td>$10^{-4.4}$</td>
<td>$10^{-7.7}$</td>
<td>$10^{-4.4}$</td>
</tr>
<tr>
<td>MSG</td>
<td>$10^{-0.8}$</td>
<td>$10^{-1.3}$</td>
<td>$10^{-2.8}$</td>
<td>$10^{-1.6}$</td>
</tr>
</tbody>
</table>

The entries are defined in eq. (1).
According to this the relative efficiency of the above numerical techniques for the case $a = 1, b = 1, A, B = O(1)$ are:

\[
\frac{\eta(SI)}{\eta(MGS)} \approx 1.3
\]

\[
\frac{\eta(MSI1)}{\eta(MGS)} \approx 2.2
\]

\[
\frac{\eta(MSI2)}{\eta(MGS)} \approx 2.6
\]

(22)

Following this example some features of the MSI methods will be presented in the next section.
9. The MSI Method Features

This section will include some mathematical analysis of the MSI algorithms. Let us define some well known notations:

- $L$ will be strictly lower matrix,
- $U$ will be strictly upper matrix,
- $D, \Lambda$ will be a diagonal matrices.

The matrix system $A$ consists of

$$A = L + D + U$$  \hspace{1cm} (23)

where, in the present example, from Fig. 2, $L$ consists of the $w_{i,j}$ and $s_{i,j}$ elements, $U$ consists of $E_{i,j}$ and $N_{i,j}$ elements and $D[(i-1)N+j] = \Phi_{i,j}$. From here on we shall leave out the matrix notation ($\hat{\Phi}$). In the SI algorithm the iteration matrix $(A + G)$ is factorized as

$$A + G = (L + \Delta)^{-1} (\Lambda + U)$$  \hspace{1cm} (24)

where

$$D = \Lambda + \text{diagonal} \ (L \Lambda^{-1} U)$$  \hspace{1cm} (25a)

and

$$G = \text{Off-diagonal} \ (L \Lambda^{-1} U).$$  \hspace{1cm} (25b)

The iteration as defined by eq. (10) is

$$(A + G) \Phi^{n+1} = G \Phi^n - M$$

or

$$\Phi^{n+1} = (A + G)^{-1} G \Phi^n - (A + G)^{-1} M$$  \hspace{1cm} (26)
If the iterative method is consistent and stationary [22] (which is almost always the case), then we can define

\[ e^n = \phi^n - \phi^{n-1} \]

and the error equation is

\[ e^n = B e^n \]  \hspace{1cm} (27)

where

\[ B = (A + G)^{-1} G \]  \hspace{1cm} (28)

The regular requirement from the series \( e^n \) to converge (to the trivial solution \( e = 0 \)) is that the spectral radius of \( B \), \( \lambda(B) \), will be less than 1:

\[ \lambda(B) < 1 \]  \hspace{1cm} (29)

It is difficult to analyze the spectral radius and the rate of convergence \( R \), defined by

\[ R = -\frac{\lambda(B)}{\lambda(B) - 1} \]  \hspace{1cm} (30)

basically, because \( B \) is, generally, a nonsymmetric matrix [13].

The MSII technique consists of the following two steps:

(i) \( (\Delta + L) C^n = G \phi^{n-1} - M \)  \hspace{1cm} (31a)

\[ \text{define} \ (\Delta + L)^{-1} = \Delta^{-1} + L \]  \hspace{1cm} (31b)

where \( L \) is another strictly lower matrix.

(ii) \( (I - \Delta^{-1} G + \Delta^{-1} U) \phi^n = L C \phi^{n-1} - (\Delta^{-1} + L) M \)  \hspace{1cm} (32)
By substituting from eq. (25)

\[ G = \Delta - D + L \Delta^{-1} U \]  

(33)

and combining the two steps, eqs. (31) and (32), the following iteration matrix will be obtained:

\[
B = D^{-1}L(\Delta + U)(A+G)^{-1}G = D^{-1}L(I + \Delta L)G
\]
or

\[
B = (D + U - L\Delta^{-1}U)^{-1}L(I + \Delta^{-1}U)
\]

(34)

depending when the matrix C is updated.

From this equation, the MSIL technique might be interpreted as being consist of two half steps \( j = 1 \) and \( j = 2 \), every one is of the form

\[
H_j \phi^{n+1} = N_j \phi^n + r_j
\]

(35a)

where

\[ A = H_j - N_j \]

(35b)

is a regular splitting, and the local iteration matrix is

\[ B_j = H_j^{-1}N_j \]

(35c)

In the first step we have,

\[
H_1 = A + G = (\Delta + L)\Delta^{-1}(\Delta + U)
\]

(36a)

\[
N_1 = G = \Delta - D + L\Delta^{-1}U
\]

(36b)

and in the second step,
\[ H_2 = \Delta + U - G = -D - (L^{\Delta^{-1}} + I) U \]  
(37a)

\[ N_2 = L(I + \Delta^{-1} U) \]  
(37b)

Since \( L \) is factorized \( B \), it can be said that \( n \) eigen values of the matrix \( B \) (out of \( n^2 \)) are zero (like \( L \), \( B \) is singular to the \( n \)th order). Qualitatively, it can be seen from the last equation that the diagonal terms of \( B(MSI1) \) are decreasing in a factor of \( \text{diag}(\Delta) / \text{diag}(D) \), (where \( \text{diag}(x) \) is a measurement for an element along the main diagonal of \( x \)). Since this factor is less than 1 (since \( G \) is small compared to \( A \) \([10, 18]\)), the eigenvalues of \( B(MSI1) \) will be spread in the region of the average of the eigenvalues of \( B(SI) \). It was proven for two non-zero convective cases of table II [which is related to eq. (1)], by using the symbolic computer language MACSYMA \([23]\), that for a field of \( 51 \times 51 \) grid points, most of the eigenvalues of \( B(SI) \) was around 0.96, where those of \( B(MSI1) \) was around 0.81. Thus, although \( H_2 \) in eq. (37a) is not a good approximation to \( A \), it helps to spread the eigenvalues of the iteration matrix \( B \) of the MSI1 technique in a range that is not so close to 1 as those of \( B(SI) \).

The MSI2 technique can be summarized in the following two steps:

(i) \[ H_1 = \Delta - G \]  
(38a)

\[ N_1 = -[L(I + U) + U] \]  
(38b)
(ii) $H_I = \Delta + U - G$ \hfill (39a)

$$N_I = L(I + \Delta^{-1} U)$$ \hfill (39b)

and

$$B(\text{MSI2}) = (\Delta + U - G)^{-1} L\Delta^{-1}(\Delta + U)(\Delta - G)^{-1}[L(\cdot + U) + U]$$

Qualitatively, the diagonal terms of $B(\text{MSI2})$ are reduced in a factor of $[\text{diag}(\Delta)/\text{diag}(D)]^2$ comparing to the $B(\text{SI})$; also $B(\text{MSI2})$ has $2(n-1)$ trivial (zero) eigenvalues. For some non-zero convection examples of table 1 it was concluded (by using the MACSYMA [23] symbolic language) that

$$\frac{\lambda[B(\text{MSI2})]}{\lambda[B(\text{MSI})]} \approx 0.95 \hfill (40)$$

No definite formula for the rate of convergence of the MSI techniques can be established [19], but all the signs have shown that it has better convergence properties than the pure SI method (or Stones' [5] method with $\alpha = 0$ as was mentioned above).

As a final example, the solution of a very high convection problem was found with the following iteration techniques:

- GS - the classical point Gauss Siedel explicit procedure.
- MGS - the alternating Gauss Siedel method as is defined in section 7.
- ADI - the alternating direction implicit method.
- LR - the implicit line relaxation method.
- ARL - the alternating line relaxation method (in [19] it is named ADI2).
ICCC - the incomplete Cholesky-conjugate gradient method as defined in [19].

SI - strongly implicit (with $\alpha = 0$).

MSI1 - (with $\alpha = 0$).

MSI2 - (with $\alpha = 0$).

Equation (1) was solved over the square defined in figure 4, with

\[
a = \pi \sin (\pi y) \cos (\pi x) \quad (41a)
\]
\[
b = \pi \epsilon - a \quad (41b)
\]
\[
A = B = \epsilon \quad (41c)
\]

and the exact solution is $\phi = [\sin \pi x \sin \pi y]/\epsilon$, with a significant boundary layer near regions near the walls.

Figure 5 presents the numerical rate of convergence for $\epsilon = 10^{-2}$ and $\epsilon = 10^{-5}$. The resulting matrix system is not symmetric and the diagonal dominance is very weak. A $51 \times 51$ equal spaced grid was chosen like in the example in figure 9 in [19], (but not the same PDE equation). To avoid influence of the programming equality and language, it is preferred to present the results in terms of iterations rather than in terms of CPU time consuming. The condition [19] of the present matrix was for $\epsilon = 10^{-2}$.

\[
\text{condition} = \frac{|\lambda_{\text{max}}|}{|\lambda_{\text{min}}|} \approx 983 \quad (42)
\]
FIG. 5: CONVERGENCE CURVES FOR THE EXAMPLE PROBLEM. Eq. (4.1)
In figure 5 the iterations are normalized with the number of the work units so that one normalized iteration is equal to one iteration of the GS method.

These results show, for this particular case, the main feature of the MSI techniques. They are more powerful than all other explicit or implicit methods, but still much more slower than the conjugate gradient accelerating techniques. The alternating Gauss Siedel method is not as powerful as seems to be in view of the strongly implicit algorithm.
10. **Conclusions**

A significant improvement on the rate of convergence of the iterative factorization technique of the form of Eq. (10) has been demonstrated. The Strongly Implicit method to this date was the only useful factorization of this type. It has been shown that two other schemes can be formulated out of this equation. One is the Modified Gauss Siedel method, which has the same type of iteration procedure as the Strongly Implicit method, and the second is the Modified Strongly Implicit method which is a combination of a line relaxation iteration technique (along diagonals of the rectangular mesh field) and the SI technique. The Modified Strongly Implicit methods are much more powerful than the strongly implicit method for the model problem; they are more efficient (22%) and 2-3 times faster. When compared with the regular ADI method it has been concluded from similar problems that the efficiency of the Modified Strongly Implicit method is about 26% higher. This result provides some information but certainly does not provide a conclusive result concerning the relative efficiency of ADI or strongly implicit methods. It is very clear that the MSI methods are slower than any accelerating methods like the conjugate gradient methods or the Chebyshev accelerating techniques. The present study can be extended by coupling it to any gradient method or any accelerating technique for higher rate of convergence, but it was not the aim of the present work. More study of the MSI technique should be done before a definite conclusion is made and before any
accelerating procedures are involved in these techniques. The main disadvantage of the MSI techniques is that the programming is difficult and quite complicated.
ACKNOWLEDGEMENT

The author would like to express his gratitude to Prof. S.G. Rubin of the University of Cincinnati for providing many helpful suggestions during the course of his study.

This research was supported by the Technion Research Foundation, Project No. 121-603, and partially by the Office of Naval Research under Contract No. N00014-79-C-0849, Project No. NR 061-258.
References


