Parameter Choices for ADI-like Methods on Parallel Computers

S. Malhotra* C. C. Douglas† M. H. Schultz‡

Abstract: Alternating direction iterative (ADI) methods do not usually work well on parallel computers due to having to do parallel rather than serial tridiagonal solves in all but one dimension. An ADI-like iteration is developed and analyzed which does not require parallel tridiagonal solves in any direction, has at least as good of a convergence rate as ADI, and has almost no communication when a small number of iterations are used. Acceleration parameters are investigated in order to reduce the number of of iterations to a small number. Numerical experiments on a network of workstations and a parallel computer are also included.

Key words: alternating direction implicit methods, ADI, partial differential equations, iterative methods, parallel computing

1 Introduction and Contributions to ADI by J. Douglas, Jr.

The alternating direction iterative (ADI) method is a smoother where information moves quickly in each of the dimensions of a boundary value problem. For problems on rectangular, regular meshes, the tridiagonal solves in each direction provide robustness in excess of line SOR.

On parallel computers, tridiagonals solves across computer memory boundaries slow classical ADI down to the point where ADI is not normally acceptable. Usually, a data transpose is performed in order to speed up the tridiagonal solves or a cyclic reduction-like step is used.

In this paper, a transpose free ADI-like method is analyzed and tested. The tridiagonal matrices to be inverted for elliptic problems are strongly diagonally dominant. Gauss-Seidel converges to the solution of the original system quickly. Hence only a very small number of Gauss-Seidel sweeps are required to obtain an accurate approximation.

The method runs fast all geometric directions, not just one. When only a small number of iterations are used, using a small overlap of subdomains allows only one communication of data between processors. This makes the ADI-like method a good candidate as a parallel solver.

*Morgan Stanley Dean Witter, 1585 Broadway, Third Floor, New York, NY 10036–8200, USA.

†Department of Mathematics, University of Kentucky, 715 Patterson Office Tower, Lexington, KY 40506–0027, USA.

‡Department of Computer Science, Yale University, P.O. Box 208285, New Haven, CT 06520-8285, USA.

In §2, classical ADI is briefly reviewed. In §3, some classical Gauss-Seidel theory is reviewed. In §4, the new ADG method is defined and analyzed. In §5, convergence of ADG is proven for a set of acceleration parameters associated with ADI. Numerical experiments are provided on a network of workstations and a parallel computer.

This paper is in an issue dedicated to the 70th birthday of Jim Douglas, Jr. He has been involved with ADI methods since its creation in the early 1950's at what is now Exxon Production Research in Houston, Texas. His interest in this method continues to this day.

The first paper that ever appeared with ADI in it was [12] rather than the commonly referenced [3]. Another early paper for the heat equation in two dimensions is [7]. The method was extended to mildly nonlinear problems in [8] and [10]. The three dimensional case was developed and analyzed in [9], which also includes analysis for the Schrödinger equation. The general ADI results for N space variables are found in [19].

The only paper ever written that shows convergence without resorting to requiring commutativity of operators was written by Pearcy [1]. While not optimal, a variable length sequence of parameters is shown to always exist to get convergence. Douglas and Pearcy wrote a paper [11] that included a functional analysis result based on a new concept, namely, the angle between two subspaces.

A general formulation for ADI methods for parabolic and hyperbolic problems can be found in [15], a paper with Gunn. Another paper with Gunn [14] provides high order correct difference schemes for multidimensional heat flow. Gunn also wrote a paper [22] that analyzed semi-implicit ADI methods using a multistage procedure, essentially a preconditioning method for semi-implicit methods.

A collection of papers were written applying ADI to finite element methods. An extensive treatment can be found in [17]. Mixed finite lement procedures are analyzed in [16], [20], and [21].

A number of Ph.D. dissertations have been written on ADI methods including ones by Brown [2], applying ADI to mixed methods, and Celia [23], applying ADI to collocation.

In more recent years, Douglas has been interested in applying ADI to the Helmhotz equation where the signs are the same in the operator (the symmetric indefinite version of the problem) and having complex Robin boundary conditions [18].

2 Classical ADI

In this section we review the classical ADI algorithm which will motivate the main results of this chapter. We shall limit our discussion to the solution of the inhomogeneous constant coefficient equation on a square domain. The continuous problem is defined by

$$-u_{xx}(x,y) - u_{yy}(x,y) + \sigma u(x,y) = S(x,y), \qquad (x,y) \in R, \tag{2.1}$$

on the unit square, $R : 0 < x, y < 1$ of the x-y plane, where $\sigma \geq 0$ and

$$u(x, y) = \gamma(x, y), \qquad (x, y) \in , ,$$

where $\gamma(x, y)$ is a prescribed function on the boundary , of R [26]. The problem is discretized by imposing a uniform $N \times N$ grid on the unit square with a grid spacing given by $h \equiv 1/(N + 1)$. The problem then reduces to the solution of the linear system

$$Au = k, \quad \text{where} \quad A = H + V + \Sigma. \tag{2.2}$$

The matrices H and V are the discretized versions of the differential operators in the horizontal and vertical directions respectively. The matrix Σ is a non-negative diagonal matrix. Each of the matrices H and V are each *completely reducible* [26]. In particular, they are reducible to the direct sum of irreducible Stieltjes matrices [26]. For the rest of the discussion we shall ignore the matrix Σ, which can be incorporated into the discussion by defining the modified matrices [13]

$$H' = H + \Sigma/2 \quad \text{and} \quad V' = V + \Sigma/2.$$

The ADI method proceeds by converting (2.2) to a pair of matrix equations

$$(H + \rho I)u = k - (V - \rho I)u \tag{2.3}$$
$$(V + \rho I)u = k - (H - \rho I)u \tag{2.4}$$

for some positive scalar ρ. The iterative method is then defined as

$$(H + \rho I)u^{m+\frac{1}{2}} = k - (V - \rho I)u^m, \tag{2.5}$$
$$(V + \rho I)u^{m+1} = k - (H - \rho I)u^{m+\frac{1}{2}}, \qquad m \geq 0, \tag{2.6}$$

where u^0 is an arbitrary initial vector approximation. The matrices $H + \rho I$ and $V + \rho I$ can be converted to tridiagonal matrices after suitable permutations of the rows and columns and the systems (2.5) and (2.6) are easily solved.

If the exact solution of the discretized problem (2.2) is denoted by u^* and the error e^m is defined by

$$e^m \equiv u^m - u^*,$$

it follows that

$$e^{m+1} = T_\rho e^m$$

where

$$T_\rho = (V + \rho I)^{-1}(H - \rho I)(H + \rho I)^{-1}(V - \rho I). \tag{2.7}$$

The above method converges to the solution for any positive ρ [26, Theorem 7.1]. In fact, for the model problem (2.2), the rate of convergence for the optimal choice of ρ is the same as for SOR with the optimal overrelaxation parameter.

3 Gauss Seidel Iteration

We investigate the effect of replacing the tridiagonal solve for one of the ADI half steps with Gauss-Seidel sweeps. In this section, we restate some of the properties of the Gauss-Seidel method as applied to the matrices obtained by the discretization of H and V in (2.2).

The Gauss-Seidel method is preferred to the Jacobi method since it converges twice as fast as the Jacobi method [26]. Furthermore, it does not require any auxiliary storage unlike the Jacobi method.

The eigenvectors of the operators H and V in (2.2) are given by

$$v_{j,m} = \sin(\pi j y)\sin(\pi m x) \qquad j, m = 1, \ldots, N.$$

For (2.2), using Gauss-Seidel with red-black ordering, we state the following result:

Theorem 3.1 *Red-black Gauss-Seidel applied to the solution of $(H + \rho I) = r$ leaves the two dimensional space spanned by the eigenvectors $v_{j,m}$ and $v_{j,m'}$ invariant, where*

$$m' = N + 1 - m.$$

Hence on V_m, the subspace spanned by $v_{j,m}$ and $v_{j,m'}$, red-black Gauss-Seidel has the matrix representation

$$M_\rho^1 = \begin{pmatrix} \delta(\frac{1+\delta}{2}) & \delta(\frac{1+\delta}{2}) \\ -\delta(\frac{1-\delta}{2}) & -\delta(\frac{1-\delta}{2}) \end{pmatrix} \tag{3.1}$$

with

$$\delta = \frac{2 - \lambda_m}{2 + \rho}. \tag{3.2}$$

k applications of the red-black point Gauss-Seidel method has a representation given by

$$M_\rho^k = \begin{pmatrix} \delta^{2k-1}\frac{(1+\delta)}{2} & \delta^{2k-1}\frac{(1+\delta)}{2} \\ -\delta^{2k-1}\frac{(1-\delta)}{2} & -\delta^{2k-1}\frac{(1-\delta)}{2} \end{pmatrix}. \tag{3.3}$$

Proof: Note that (3.1) is a restatement of a well known result [6] and that (3.3) is derived from repeated applications of the matrix in (3.1). ∎

Theorem 3.2 *In the subspace $V_m =$ span $\{v_{jm}, v_{jm'}\}$, let u^0 be an arbitrary initial iterate and u^* be exact solution to the problem*

$$(H + \rho I)u = r.$$

k applications of the Gauss-Seidel method generates the iterate u^k where

$$u^k = (I - M_\rho^k)u^* + M_\rho^k u^0. \tag{3.4}$$

As can be seen, the red-black Gauss-Seidel method reduces the error for eigenvectors with eigenvalues close to 2 (where δ is small) rapidly whereas the error along the extremal eigenvectors (with λ_m close to 0 or 4) dies down slowly. In particular, after k applications of the method, for the eigenspace defined by

$$V_{j,m} = \text{ span } \{v_{j,m}, v_{j,m'}\} \tag{3.5}$$

we have

$$\max |v^k_{j,m}, v^k_{j,m'}| \leq \delta^{2k-1}(1+\delta) \max |v^0_{j,m}, v^0_{j,m'}|.$$

The error in the eigenspace is uniformly bounded and decreases with the number of iterations.

4 The ADG(ρ,k) Method

In this section we analyze the properties of an ADI iteration with the tridiagonal solves for one half step replaced by k Gauss-Seidel sweeps. We denote this approximate method by ADG(ρ,k) where ρ is the ADI parameter used and k is the number of Gauss-Seidel iterations used to approximate the tridiagonal solve.

Section 5.2 extends the results of this section to the multiple ρ case for the model problem.

The convergence rate of any method which approximates ADI can be analyzed as a perturbation of the ADI method.

Lemma 4.0 *In the eigenspace $V_{j,m}$ defined in (3.5), let $u^*_{j,m}$ be the solution to the problem defined by (2.2), $u^0_{j,m}$ be an arbitrary initial iterate, $u^{ADI}_{j,m}$ be the iterate produced by a single iteration of ADI ((2.5) and (2.6)) and $u^{ADG}_{j,m}$ be the iterate produced by a single application of the ADG(ρ,k) method. If*

$$d_{j,m} \equiv u^0_{j,m} - u^*_{j,m},$$

$$\Delta_{j,m} \equiv u^{ADG}_{j,m} - u^{ADI}_{j,m}$$

and

$$D_\rho \equiv \begin{pmatrix} \lambda_m - \rho & 0 \\ 0 & \lambda_{m'} - \rho \end{pmatrix},$$

then in $V_{j,m}$ we have

$$\Delta_{j,m} = \frac{1}{\lambda_j + \rho} D_\rho M^k_\rho \begin{pmatrix} d_{j,m} \\ d_{j,m'} \end{pmatrix}. \tag{4.1}$$

with M^k_ρ defined in (3.3).

Lemma 4.0 can be proved by considering the effect of each half step of ADG(ρ,k) on the eigenvectors of the matrix A (see [24]). Note that (4.1) measures the difference in the iterates produced by the ADG(ρ,k) and ADI methods.

Theorem 4.1 *In the eigenspace $V_{j,m}$ defined in (3.5), let u^* be the solution to the problem defined by (2.2), u^0 be an arbitrary initial iterate and u^{ADG} be the iterate produced by a single application of the ADG(ρ,k) method. If*

$$e^0_{j,m} \equiv u^0_{j,m} - u^*_{j,m}$$

and

$$e_{j,m} \equiv u^{ADG}_{j,m} - u^*_{j,m}$$

we have

$$\|e_{j,m}\| \leq (\|T_\rho\| + \frac{\delta^{2k}}{2} \left| \frac{\lambda_m - \rho}{\lambda_j + \rho} \right|) \|e^0_{j,m}\|$$

with T_ρ defined in (2.7) and δ defined in (3.2).

5 Accelerated ADI with Multiple Parameters

For the model problem, the matrices H and V in (2.2) commute. In this special case, there is a well developed theory to accelerate the convergence of ADI methods by the choice of different ρs used cyclically. In this section we shall investigate the use of the ADG(ρ,k) approximation to speed the solution of (2.2) in this case.

For the model problem (2.2), we observe that the matrices H and V (and thus A) possess a common basis of orthogonal eigenvectors $v_{j,m}$, where

$$v_{j,m} = \sin(\pi j y) \sin(\pi m x) \qquad \textit{and} \qquad A v_{j,m} = \lambda_{j,m} v_{j,m}. \tag{5.1}$$

It is also readily verified that H and V commute: $HV = VH$. This condition can be met in practice for all equations with separable coefficients with an appropriate discretization of the problem [28]. Since H and V share a common set of eigenvectors, for any $v_{j,m}$, we have

$$H v_{j,m} = \lambda_m v_{j,m},$$
$$V v_{j,m} = \lambda_j v_{j,m},$$

with $\lambda_{j,m}$ in (5.1) given by

$$\lambda_{j,m} = \lambda_j + \lambda_m.$$

For any eigenvector $v_{j,m}$ we have

$$T_\rho v_{j,m} = \frac{(\lambda_j - \rho)}{(\lambda_j + \rho)} \frac{(\lambda_m - \rho)}{(\lambda_m + \rho)} v_{j,m}.$$

For any given ρ, we find that the error along all eigenvectors with eigenvalues close to ρ decreases very quickly (see Figure 5.1). For all eigenvectors with λ_j and λ_m sufficiently

different from ρ, the error decreases extremely slowly leading to a slow overall rate of convergence of the iteration (see Figure 5.1) . However, by varying the ρ's we can selectively reduce the error on different parts of the spectrum (see Figure 5.2) leading to a faster asymptotic rate of convergence for the entire method.

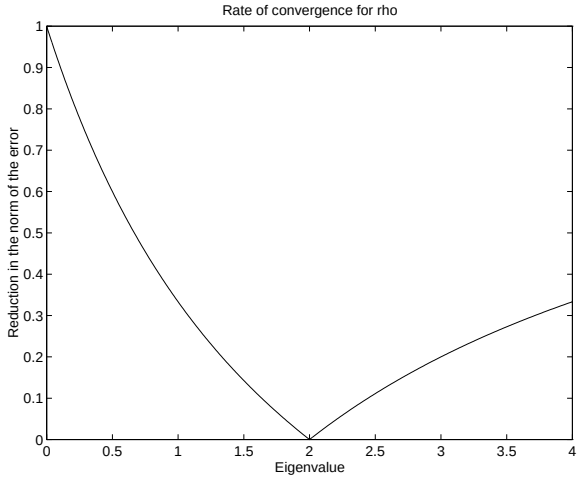

Figure 5.1: Rate of convergence for a single $\rho = 2$

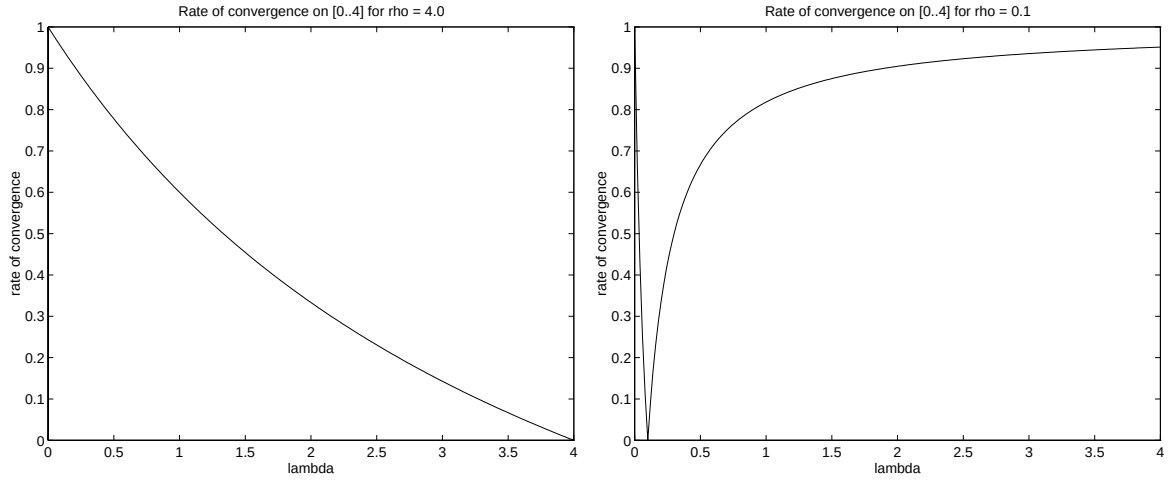

Figure 5.2: Convergence rates across the spectrum for different ρ's

One scheme for picking a series of ρ's [27] to accelerate the rate of convergence uses information about the extremal eigenvalues of the operators H and V. If

$$0 < \alpha \leq \lambda_m, \lambda_j \leq \beta,$$

we define

$$c = \left(\frac{\alpha}{\beta}\right),$$

$$n = \lceil \log_\delta(c) \rceil + 1,$$

and

$$\delta = (\sqrt{2} - 1)^2.$$

The ρ's are picked cyclically from the set

$$\rho_j = \beta c^{\frac{j-1}{n-1}}, \qquad j = 1, \ldots, n. \tag{5.2}$$

The rate of convergence of the above method is given by [27]

$$\|e_{i+n}\| \leq \delta \|e_i\|.$$

Since $n = O(|\log(h)|)$, the error goes down by a constant for every $\log(N)$ iterations.

5.1 The acceleration parameters

For the model problem (2.2), we can estimate the order of magnitude of the different ρ's selected in the course of the iteration. In this case,

$$\lambda_j = \lambda_m = 2 - 2\cos\left(\frac{\pi i}{N+1}\right), \qquad i = 1, \ldots, N.$$

All the eigenvalues for both operators are positive and lie in the interval $(0, 4)$. For a $N \times N$ grid we have

$$\alpha = 2 - 2\cos(\frac{\pi}{N+1}) \approx \frac{\pi^2}{N^2}$$

and

$$\beta = 2 + 2\cos(\frac{\pi}{N+1}) \approx 4.$$

Substituting in (5.2) gives

$$\rho_j \approx 4\delta^{j-1}, \qquad j = 1, \ldots, n.$$

The largest ρ is almost equal to 4 and subsequent ρ's decay to α in magnitude (see Figure 5.4). For most N of interest (typically a few hundred or thousand) a significant fraction of the parameters are *large* in magnitude (see Figure 5.3). We use an arbitrary cutoff point to separate large parameters from small ones, typically 0.1 for this problem. For the tridiagonal systems generated in (2.5) and (2.6) the magnitude of ρ determines the diagonal dominance of the system.

Since a significant fraction of the ρ's are large we expect that ADG(ρ,k) to converge for small k. In Section 5.2 we investigate the use of the ADG(ρ,k) method for large ρ.

5.2 Uniform Convergence Over the Entire Spectrum

In this section we investigate the use of the ADG(ρ,k) method for the model problem (2.2) with multiple ρ. The application of each ADG(ρ,k) step can be viewed as the application of the ADI operator which is perturbed by the operator $\mathcal{P}$,

$$e^{n+1} = ADG(\rho, k)e^n = (T_\rho + \mathcal{P}_\rho)e^n,$$

where T_ρ is defined in (2.7). $\mathcal{P}_\rho$ maps the error for the eigenvector $v_{j,m}$

$$\|v^n_{j,m}\| \leq \frac{\delta^{2k}}{2}\frac{\lambda_m - \rho}{\lambda_j + \rho}\|v^{n-1}_{j,m}\|. \tag{5.3}$$

Lemma 5.0 *After n iterations, the error in the modified iteration (using ADG(ρ,k) for the largest r parameters) is given by*

$$e^{i+n} = \left(\prod_{i=1}^{r}(T_{\rho_i} + \mathcal{P}_{\rho_i}) \prod_{i=r+1}^{n} T_{\rho_i}\right) e^i.$$

where T_{ρ_i} is defined in (2.7) and $\mathcal{P}_{\rho_i}$ is defined by (5.3).

Proof: The result can be obtained by replacing the T_ρ for the each of the largest r parameters with $(T_\rho + \mathcal{P}_\rho)$. The effect for all n steps is obtained by composing the operators for all ρ and the commutativity of H and V. ∎

We now analyze the spectral properties of the operator $\mathcal{P}$ for different ρ's occurring in the solution of the model problem. Since $\mathcal{P}_\rho$ measures the deviation of ADG(ρ,k) from the true ADI iterate we expect it to have a small norm for large ρ (where ADG(ρ,k) converges quickly). We shall perform the analysis for ADG(ρ, 3) to simplify the proof. In practice, the proof can be modified to any value of k for each iteration.

Lemma 5.0 *For $\rho = \beta \approx 4$ and $k = 3$, $\|\mathcal{P}_4\| \leq 6.5 \times 10^{-4}$.*

Proof: Since $0 < \lambda_m, \lambda_j < 4$, for $\rho = 4$ we have

$$|\delta| = \left|\frac{2 - \lambda_m}{2 + \rho}\right| \leq \frac{1}{3},$$

$$|\lambda_m - \rho| < 4$$

and

$$|\lambda_j + \rho| > \rho$$

which, using (5.3), gives

$$\|\mathcal{P}_4\| \leq |\frac{\delta^6}{2}\frac{\lambda_m - \rho}{\rho}| \leq \frac{1}{2 \cdot 3^6} = 6.5 \times 10^{-4}.$$ ∎

As can be seen from Lemma 5.0 the norm of the deviation for large ρ is quite small. We now compute the overall convergence rate of a scheme where ADI for the largest three ρ is replaced by ADG(ρ,3).

Theorem 5.1 *The rate of convergence for the model problem (2.2), using ADG(ρ, 3) for the largest three ρ's is given by*

$$\|e^{i+n}\| \leq 0.254\|e^i\|.$$

Proof: Using Lemma 5.0 the composite operator is given by

$$\mathcal{A} = \prod_{i=1}^{3}(T_{\rho_i} + \mathcal{P}_{\rho_i}) \prod_{i=4}^{n} T_{\rho_i}. \tag{5.4}$$

We shall analyze the contribution of each term in the expansion of (5.4). From [27] we have

$$\| \prod_{i=1}^{n} T_{\rho_i} \| \leq (\sqrt{2}-1)^2.$$

We analyze a single term to illustrate the analysis and then sum the contributions from all the other terms. One of the terms in the expansion of (5.4) is given by

$$\mathcal{T} = T_{\rho_1} T_{\rho_2} \mathcal{P}_{\rho_3} \prod_{i=4}^{n} T_{\rho_i}.$$

If

$$S_\rho = \{x \mid (\sqrt{2}-1)\rho \leq |x-\rho| \leq (\sqrt{2}+1)\rho\}$$

then

$$\frac{\|T_\rho v_{jm}\|}{\|v_{jm}\|} \leq \begin{cases} (\sqrt{2}-1)^2 & \text{if} \quad \lambda_j \in S_\rho \quad \text{and} \quad \lambda_m \in S_\rho, \\ (\sqrt{2}-1) & \text{if} \quad \lambda_j \in S_\rho, \lambda_m \notin S_\rho \quad \text{or} \quad \lambda_j \notin S_\rho, \lambda_m \in S_\rho, \\ 1 & \text{otherwise.} \end{cases}$$

Therefore we have

$$\|\mathcal{T}_{\rho_3}\| \leq \begin{cases} (\sqrt{2}-1)^2\|\mathcal{P}_{\rho_3}\| & \text{if} \quad \lambda_j \notin S_{\rho_3} \quad \text{and} \quad \lambda_m \notin S_{\rho_3}, \\ (\sqrt{2}-1)\|\mathcal{P}_{\rho_3}\| & \text{if} \quad \lambda_j \in S_{\rho_3}, \lambda_m \notin S_{\rho_3} \quad \text{or} \quad \lambda_j \notin S_{\rho_3}, \lambda_m \in S_{\rho_3}, \\ \|\mathcal{P}_{\rho_3}\| & \text{otherwise} \end{cases}$$

or

$$\|\mathcal{T}_{\rho_3}\| \leq \begin{cases} 0.045 & \text{if} \quad \lambda_j \notin S_{\rho_3} \quad \text{and} \quad \lambda_m \notin S_{\rho_3}, \\ 0.068 & \text{if} \quad \lambda_j \in S_{\rho_3}, \lambda_m \notin S_{\rho_3} \quad \text{or} \quad \lambda_m \in S_{\rho_3}, \lambda_j \notin S_{\rho_3}, \\ 0.072 & \text{otherwise.} \end{cases}$$

N	Iterations	Iterations
	Original	Modified
200	23	24
250	28	31
300	32	36
400	35	39
500	41	46

Table 5.1: Number of iterations for the modified and unmodified method

Similarly, estimating the norm of $\mathcal{T}_{\rho_1}$ and $\mathcal{T}_{\rho_2}$ we can show that

$$\|\mathcal{A}\| \leq ((\sqrt{2}-1)^2 + 0.072 + 0.010 + \ldots) = 0.254,$$

where the ignored terms sum up to less than 10^{-3}. ∎

Remark: The norm of the error in the composite method is reduced by a factor of 0.254 every n iterations. The error in the unmodified method, in contrast, goes down by a factor of 0.172 every n iterations.

5.3 Numerical Results

5.3.1 Serial Implementation. We implemented the above scheme on a SUN SPARC10 workstation to solve (2.2) on a square grid. The right hand side was chosen as a randomly generated vector with entries in the range $[0, 1]$. The method implemented can be represented as

$$ADG(\rho_1, 1) \circ ADG(\rho_2, 2) \circ ADG(\rho_3, 3) \circ \Pi_{j=4}^{n} ADI_{\rho_j}.$$

The iteration was terminated when the two norm of the residual was less than 10^{-4}. Table 5.1 contains the number of iterations to convergence for the method.

5.3.2 Parallel Implementation. Since ADG(ρ,k) involves replacing the original tridiagonal solves in the ADI method with a Gauss–Seidel iteration, we would like to be able to implement the Gauss–Seidel method (with red–black ordering) efficiently. The classical method for solving a tridiagonal equation on p processors involves a series of reduction operations on the variables and requires $(2\log(p) + 1)$ communication steps to complete [25]. Similarly, every red–black Gauss–Seidel iteration requires two communication steps to exchange the data at the boundaries of subdomain on the processor: one after the computation of the red points and the other after the computation

of the black points. By overlapping the subdomains on the processors by two points, the red and black half solves for the Gauss–Seidel method were aggregated.

We implemented the above scheme on a network of PowerPC machines connected by an ethernet. The MPI library [5] was used to communicate between the various processors. Table 5.2 contains the total time to reduce the norm of the residual to 10^{-4}.

N	Procs	Time (s)	Time (s)	Savings
		Original	Modified	%
400	2	9.31	7.23	22
	3	7.65	5.43	29
	4	6.68	5.13	23
	5	6.74	4.68	30
	6	6.76	4.57	32
	7	6.66	4.61	31
	8	7.78	5.18	33
500	2	13.58	11.01	19
	3	10.41	7.71	26
	4	13.37	10.41	22
	5	12.27	9.48	23
	6	11.20	10.06	10
	7	10.22	8.31	19
	8	9.52	7.77	18

Table 5.2: Parallel timing for the Poisson Equation, multiple ρs, PowerPC 560, ethernet

Acknowledgments: This work was supported in part by an ONR grant.

6 Conclusions

We have attempted to show that the ADG(ρ,k) method is an effective approximation to the ADI method. The lower computational cost of ADG(ρ,k) coupled with its easy parallelization properties make it an effective smoother for elliptic problems.

Due to the lower memory requirements of the ADG(ρ,k) method and the effective use of data in the cache this approach offers a significant speedup in the case of RISC processors. The method is trivially parallelized. Since the parallel overhead of the scheme can be further reduced by the aggregation of the Gauss-Seidel steps, the method is effective even on parallel machines with high latency.

The combination of the approximation with redundant computation allows us to complete an iteration with a single communication event between neighboring processors, as opposed to $O(\log p)$ communication events on p processors.

In the case of problems with variable coefficients, the ADG(ρ,k) approximation allows us to compute the solution without storing the factors of the matrix. This allows for much larger problems to be solved without resorting to the use of virtual memory.

The ideas of this chapter can also be applied to alternating direction line SOR methods with little or no modification. Alternating direction line SOR methods are often used as smoothers for multigrid methods since their smoothing properties are slightly better than the corresponding pointwise smoothers [4] (a smoothing rate of 0.2 for line SOR vs a smoothing rate of 0.25 for point SOR).

REFERENCES

[1] C. M. Pearcy. On convergence of alternating direction procedures. *Numerische Mathematik*, 4:172–176, 1962.

[2] D. C. Brown. *Alternating-direction iterative schemes for mixed methods for second order elliptic problems.* PhD thesis, University of Chicago, 1982.

[3] D. W. Peaceman and H. H. Rachford Jr. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3:28–41, 1955.

[4] I. S. Duff, R. G. Grimes, and J. G. Lewis. User's guide for the Harwell–Boeing sparse matrix collection (Release I). Technical Report TR/PA/92/86, CERFACS, Toulouse, 1992.

[5] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface.* The MIT Press, 1994.

[6] W. Hackbusch. On the multi–grid method applied to difference equations. *Computing*, 20:291–306, 1978.

[7] J. Douglas Jr. On the numerical integration of $u_{xx} + u_{yy} = u_t$ by implicit methods. *Journal of the Society for Industrial and Applied Mathematics*, 3:42–65, 1955.

[8] J. Douglas Jr. Alternating direction iteration for mildly nonlinear elliptic difference equations. *Numerische Mathematik*, 3:92–98, 1961.

[9] J. Douglas Jr. A correction to my paper "Alternating direction iteration for mildly nonlinear elliptic difference equations". *Numerische Mathematik*, 4:301–302, 1962.

[10] J. Douglas Jr. Alternating direction methods for three space variables. *Numerische Mathematik*, 4:41–63, 1962.

[11] J. Douglas Jr. and C. M. Pearcy. On convergence of alternating direction procedures in the presence of singular operators. *Numerische Mathematik*, 5:175–183, 1963.

[12] J. Douglas Jr. and D. W. Peaceman. Numerical solution of two–dimensional heat flow problems. *American Institute of Chemical Engineering Journal*, 1:505–512, 1955.

[13] J. Douglas Jr. and H. H. Rachford Jr. On the numerical solution of heat conduction problems in two or three space variables. *Transactions of the American Mathematics Society*, 82:421–439, 1956.

[14] J. Douglas Jr. and J. E. Gunn. Two high-order correct difference analogues for the equation of multi-dimensional heat flow. *Mathematics of Computation*, 17:71–80, 1963.

[15] J. Douglas Jr. and J. E. Gunn. A general formulation of alternating direction methods, I. Parabolic and hyperbolic problems. *Numerische Mathematik*, 6:428–453, 1964.

[16] J. Douglas Jr. and P. Pietra. A description of some alternating-direction iterative techniques for mixed finite element methods. In W. E. Fitzgibbon, editor, *Mathematical and Computational Methods in Seismic Exploration and Reservoir Modeling*, pages 37–53. SIAM, Philadelphia, 1986.

[17] J. Douglas Jr. and T. Dupont. Alternating-direction Galerkin methods on rectangles. In *Numerical Solution of Partial Differential Equations II*, pages 133–214. Academic Press, New York, 1971. B. Hubbard, ed.

[18] J. Douglas Jr., J. L. Hensley, and J. E. Roberts. An alternating-direction iteration method for Helmholz problems. In *Proceedings of the ISNA '92 (International Symposium on Numerical Analysis), Part I, Prague 1992*, volume 38, pages 289–300, 1993.

[19] J. Douglas Jr., R. B. Kellogg, and R. S. Varga. Alternating direction iteration methods for N space variables. *Mathematics of Computation*, 17:279–282, 1963.

[20] J. Douglas Jr., R. Durán, and P. Pietra. Formulation of alternating-direction iterative methods for mixed methods in three space. In E. L. Ortiz, editor, *Numerical Approximation of Partial Differential Equations*, pages 21–30. North-Holland, Amsterdam, 1987.

[21] J. Douglas Jr., R. Durán, and P. Pietra. *Alternating-Direction Iteration for Mixed Finite Element Methods*, volume 7 of *Computing Methods in Applied Science and Engineering*. North-Holland, 1990.

[22] J. E. Gunn. The solution of elliptic difference equations by semi-explicit iterative techniques. *Journal of SIAM, Series B, Numerical Analysis*, 2:24–45, 1965.

[23] M. A. Celia. *Collocation on deformed finite elements and alternating direction collocation methods*. PhD thesis, Princeton University, Princeton, 1983.

[24] S. Malhotra. *Topics in Mutigrid Methods*. PhD thesis, Yale University, 1996. Available electronically through MGNet, *http://www.mgnet.org/mgnet-papers.html*.

[25] R. W. Hockney and C. R. Jesshope. *Parallel Computers*. Adam Hilger Ltd., Bristol, U.K., 1981.

[26] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Inc, Englewood Ciffs, New Jersey, 1962.

[27] E. L. Wachspress. Optimum alternating–direction–implicit iteration parameters for a model problem. *Journal of the Society for Industrial and Applied Mathematics*, 10:339–350, 1962.

[28] E. L. Wachspress and G. J. Habetler. An alternating direction implicit iteration technique. *Journal of the Society for Industrial and Applied Mathematics*, 8:403–424, 1960.

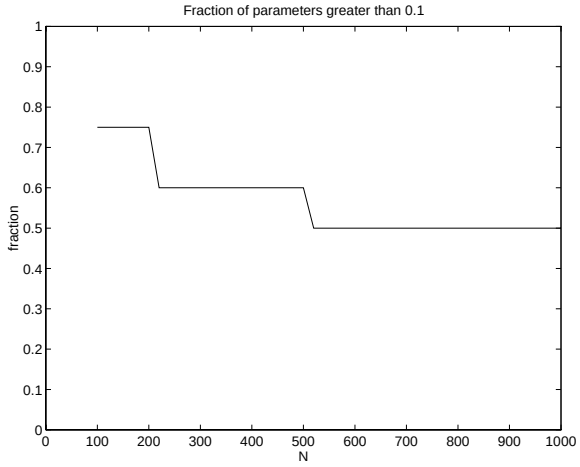

Figure 5.3: Fraction of *large* acceleration parameters

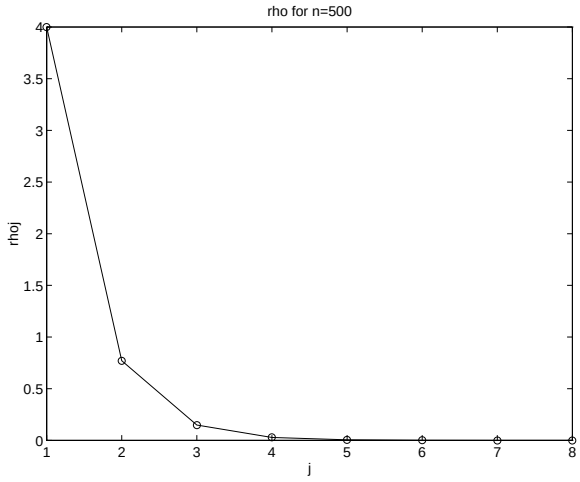

Figure 5.4: Computed ρs for the model problem $N = 500$.