RELAXED AND STABILIZED INCOMPLETE FACTORIZATIONS FOR NON-SELF-ADJOINT LINEAR SYSTEMS

HOWARD C. ELMAN

Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA.

Abstract.

Two classes of incomplete factorization preconditioners are considered for nonsymmetric linear systems arising from second order finite difference discretizations of non-self-adjoint elliptic partial differential equations. Analytic and experimental results show that relaxed incomplete factorization methods exhibit numerical instabilities of the type observed with other incomplete factorizations, and the effects of instability are characterized in terms of the relaxation parameter. Several stabilized incomplete factorizations are introduced that are designed to avoid numerically unstable computations. In experiments with two-dimensional problems with variable coefficients and on nonuniform meshes, the stabilized methods are shown to be much more robust than standard incomplete factorizations.

AMS(MOS) subject classification. Primary: 65F10, 65N20. Secondary: 15A06.

Key words: Linear systems, iterative methods, preconditioners, incomplete factorizations, non-self-adjoint, convection-diffusion.

1. Introduction.

Preconditioned conjugate gradient (PCG) methods using preconditioners based on sparse incomplete LU factorizations of the coefficient matrix are known to be effective for solving linear systems arising from discretizations of elliptic partial differential equations. For self-adjoint problems, the incomplete LU factorization (ILU) [22] and the modified incomplete LU factorization (MILU) [9], [15] have been subjected to theoretical analyses that in large part explain their behavior. In particular, suppose the differential operator has smooth coefficients, and either finite differences [9], [15] or piecewise linear finite elements [16] are used to discretize on a uniform $n \times n$ grid. Then the MILU preconditioner reduces the

The work presented in this paper was supported by the National Science Foundation under grants DMS-8607478, CCR-8818340, and ASC-8958544, and by the U.S. Army Research Office under grant DAAL-0389-K-0016.

Received January 1989. Revised June and August 1989.

condition number from $O(n^2)$ to $O(n)$, whereas the ILU preconditioner does not improve conditioning but does produce a system with compressed eigenvalues. Both preconditioners have also been used successfully for problems with non-smooth coefficients [1], [6], [8]. In general, the MILU preconditioner is more effective, and both methods are robust.

Preconditioned methods of this type using variants of the conjugate residual method applicable to nonsymmetric linear systems [10], [18], [24], have also been shown to be effective for discretized non-self-adjoint elliptic problems [2], [11], [12]. However, it is not the case that one or the other of these incomplete factorization methods is generally superior. Indeed, there are classes of problems, arising from finite differences, for which these methods are not robust. This phenomenon has been observed and analyzed for constant coefficient problems [13], where it is shown that the main difficulty comes from numerical instability of the triangular solves performed for the preconditioning. At least one of the ILU or MILU preconditioners is effective for most cases, but neither one can be successfully applied to arbitrary problems. Thus, for problems with variable coefficients, where the character of the problem may vary depending on the position in the underlying domain, standard preconditioned conjugate gradient methods are likely not to be robust.

In this paper, we examine variants of the ILU and MILU incomplete factorization preconditioners with the goal of improving the robustness of preconditioned iterative methods for discrete non-self-adjoint elliptic problems. We consider two classes of preconditioners:

1. *Relaxed incomplete factorizations.* For the MILU factorization, the errors in the off-diagonal entries of the preconditioner LU are accumulated and the factorization is modified in such a way that the sum across the rows of the error matrix is (approximately) zero. It has been observed that a *relaxed* incomplete factorization [1], [3], [4], in which only part of this modification is performed, leads to better performance of PCG on self-adjoint problems. (See also [21] for an alternative approach to relaxation.) We perform an analysis and empirical study of relaxed incomplete factorizations for solving non-self-adjoint problems.

2. *Stabilized incomplete factorizations.* The ILU and MILU factorizations, as well as the relaxed incomplete method, are defined strictly in terms of the algebraic structure of the coefficient matrix without reference to the values of its entries. However, the instabilities associated with preconditioners for constant coefficient non-self-adjoint difference operators can be simply characterized in terms of properties of the matrix entries [13]. Using this fact as a starting point, we introduce several incomplete factorizations whose definitions depend on both the algebraic structure and values of the coefficient matrix. These are adaptive in the sense that when they are applied to differential operators with variable coefficients, different values of the relaxation parameter may be used in different rows of the factors. In addition, following an idea of [26], the factors are forced to be diagonally dominant. By enhancing stability in these ways, we find that the resulting methods are considerable more robust.

An outline of the paper is as follows. In §2, we briefly describe the relaxed incomplete factorization of [3], [4]. In §3, we present a two-dimensional discrete constant coefficient model problem based on using centered finite differences to discretize all differential operators, and we perform a stability analysis of the relaxed incomplete LU preconditioning applied to this problem. In §4, we introduce several stabilized incomplete factorizations. In §5, we present the results of numerical experiments with the relaxed and stabilized incomplete factorization preconditioners, for two-dimensional constant coefficient and variable coefficient test problems on uniform meshes. In addition, we examine the preconditioners for solving constant coefficient problems discretized on nonuniform meshes on which boundary layers are resolved. In §6, we draw some conclusions.

2. The relaxed incomplete factorization.

Let A denote a matrix of order N, and let $\mathcal{N}\mathcal{Z}$be a set of indices contained in $\{(i,j) \mid 1 \le i,j \le N\}$ that includes all diagonal indices $\{(i,i)\}$. The relaxed incomplete LU factorization (RILU) of A is the product $Q = LU$ where the nonzero entries of L and U are defined by the following algorithm.

Algorithm: The relaxed incomplete LU factorization.

```
for $i = 1$ to $N$ $l_{ii} \leftarrow 0$
  for $j = 1$ to $N$
    $s_{ij} \leftarrow a_{ij} - \sum_{k=1}^{\min(i,j)-1} l_{ik}u_{kj}$
    if $(i,j) \in \mathcal{N}\mathcal{Z}$ then
      if $(i > j)$ then $l_{ij} \leftarrow s_{ij}$
      if $(i = j)$ then $l_{ii} \leftarrow l_{ii} + s_{ii}$
      if $(i < j)$ then $u_{ij} \leftarrow s_{ij}$
    else $l_{ii} \leftarrow l_{ii} + \omega s_{ij}$
    end
  end
  $u_{ii} \leftarrow 1$
  for $j = i + 1$ to $N$ $u_{ij} \leftarrow u_{ij}/l_{ii}$
  end
end
```

This is a straightforward generalization of the relaxed incomplete Cholesky factorization for symmetric matrices described in [3]. We do not consider whether the factorization is well-defined, cf. [6], [16]. (The stabilized methods of §4 are forced to be well-defined.) The relaxation parameter ω may be chosen from $(-\infty, 1]$; see [27] for previous results on negative relaxation parameters. This parameter determines how the rowsum modification from the MILU factorization is incorporated into the factorization. The choice $\omega = 1$ results in the MILU factorization, and the choice $\omega = 0$ gives the ILU factorization.[1] In all cases, the preconditioner Q satisfies $A = Q - R$ where $r_{ij} = 0$ for off-diagonal indices $(i,j) \in \mathcal{N}\mathcal{Z}$ and $r_{ii} = -\omega \sum_{j \neq 1} r_{ij}$. Every step of a preconditional iterative method requires a computation of the form $w \leftarrow Q^{-1}v = U^{-1}L^{-1}v$.

Our concern here is when A comes from a five-point finite difference operator on an $n \times n$ grid (see §3), and $\mathcal{N}\mathcal{Z}$ corresponds to the indices of nonzero entries of A,

$$\begin{array}{ll} (i,i) & 1 \le i \le n \\ (i,i-1),\ (i-1,i) & 1 \le i \le n, i \neq 1 \bmod n \\ (i,i-n),\ (i-n,i) & n+1 \le i \le n^2. \end{array}$$

(The generalization to meshes with different horizontal and vertical sizes is straightforward.) In this case, the factors have the form

$$Q = (D + L_A)(I + D^{-1}U_A), \tag{2.1}$$

where L_A and U_A are the strict lower-triangular and upper-triangular parts of A, respectively, and

$$\begin{aligned} d_i = a_{ii} &- \frac{l_{i,i-1}u_{i-1,i}}{d_{i-1}} - \frac{l_{i,i-n}u_{i-n,i}}{d_{i-n}} \\ &- \omega\left[\frac{l_{i,i-n}u_{i-n,i-(n-1)}}{d_{i-n}} + \frac{l_{i,i-1}u_{i-1,i+(n-1)}}{d_{i-1}}\right]. \end{aligned} \tag{2.2}$$

Here l_{ij} and u_{ij} refer to entries of L_A and U_A, and they are taken to be zero for indices $(i,j) \notin \mathcal{N}\mathcal{Z}$. The off-diagonal entries of R are

$$r_{i,i-(n-1)} = \frac{l_{i,i-n}u_{i-n,i-(n-1)}}{d_{i-n}}, \qquad r_{i,i+n-1} = \frac{l_{i,i-1}u_{i-1,i+(n-1)}}{d_{i-1}}. \tag{2.3}$$

3. Stability analysis for a constant coefficient model problem.

Consider the two-dimensional constant coefficient convection-diffusion equation

$$-\Delta u + 2P_1 u_x + 2P_2 u_y = f \tag{3.1}$$

on the unit square $\Omega = (0,1) \times (0,1)$ with Dirichlet boundary conditions $u = g$ on $\partial\Omega$. Let (3.1) be discretized by second order finite differences on a uniform $n \times n$ grid:

$$\Delta u \approx \frac{u_{s+1,t} - 2u_{st} + u_{s-1,t}}{h^2} + \frac{u_{s,t+1} - 2u_{st} + u_{s,t-1}}{h^2},$$

$$u_x \approx \frac{u_{s+1,t} - u_{s-1,t}}{2h}, \qquad u_y \approx \frac{u_{s,t+1} - u_{s,t-1}}{2h},$$

[1] The standard MILU factorization actually is applied to $A + cI$ where $c \ge 0$ typically depends on the mesh size of the discretization. We restrict our attention to $c = 0$ here.

where $h = 1/(n + 1)$. The result is a system of linear equations

$$Au = f \tag{3.2}$$

where A is of order $N = n^2$ and u and f are now vectors of size N. If the grid points are ordered using the rowwise natural ordering, then A is a block tridiagonal matrix of the form

$$tri[A_{t,t-1}, A_{tt}, A_{t,t+1}], \qquad 1 \leq t \leq n.$$

If the matrix and right hand side are scaled by h^2, then the blocks of A are given by

$$A_{t,t-1} = -(1 + p_2)I, \quad A_{tt} = tri[-(1 + p_1), 4, -(1 - p_1)], \quad A_{t,t+1} = -(1 - p_2)I,$$

where I is the identity matrix, $p_1 = P_1 h$, $p_2 = P_2 h$, $tri[a, b, c]$ denotes a tridiagonal matrix, and all blocks are of order n.

The rest of this section contains a recapitulation and generalization of the analysis of [13], where it was shown that for certain values of p_1 and p_2, numerical instability causes the ILU and MILU factorizations to be ineffective as preconditioners for solving (3.2). Although [13] only considered the cases $\omega = 0$ and $\omega = 1$, we now state the results in terms of the RILU factorization with $\omega \leq 1$. We will show how the analysis is generalized as necessary. The analysis consists of two main parts.

PROPOSITION 1. *Let $\hat{Q} = (\hat{D} + L_A)(I + \hat{D}^{-1}U_A)$ where $\hat{D} = \alpha I$,*

$$\alpha = 2 + [(1 - \omega)(2 + p_1^2 + p_2^2) + \omega(p_1 + p_2)^2]^{1/2}, \tag{3.3}$$

and $\omega \in [-1, 1]$. Then $\hat{Q}$ is the RILU factorization of $\hat{A}$, where $\hat{A} - A$ is a diagonal matrix of rank $4n - 2$ whose only nonzero entries are in rows corresponding to grid points next to $\partial\Omega$.

PROOF. We seek $\hat{A}$ such that $\hat{A} = \hat{Q} - \hat{R}$ and

$$\hat{q}_{ii} - \hat{a}_{ii} = \hat{r}_{ii} = -\omega \sum_{j \neq 1} \hat{r}_{ij}. \tag{3.4}$$

Let $\mathscr{I}$ denote the set of indices corresponding to interior grid points of Ω, i.e. points not adjacent to $\partial\Omega$. For all $i \in \mathscr{I}$, let $\hat{a}_{ii} = a_{ii} = 4$. Formally as functions of α, we have

$$\hat{q}_{ii} = \alpha + (1 - p_1^2)/\alpha + (1 - p_2^2)/\alpha, \qquad \sum_{j \neq i} r_{ij} = (2 - 2p_1 p_2)/\alpha,$$

for $i \in \mathscr{I}$. The value (3.3) is obtained by substituting these expressions into (3.4) and solving for the larger root of the resulting quadratic equation. Note that α is real for all p_1, p_2 when $\omega \in [-1, 1]$. Thus $\hat{Q}$, and therefore $\hat{r}_{ij}$ (for $j \neq i$) are determined. For $i \notin \mathscr{I}$, $\hat{a}_{ii}$ is chosen to satisfy

$$\hat{a}_{ii} = \hat{q}_{ii} + \omega \sum_{j \neq i} r_{ij}. \qquad ■$$

This result shows that the constant coefficient factorization $\hat{Q}$ is the RILU factorization of a matrix that is close to A. The actual RILU factorization of A has the form (2.1) where D is not constant, but we have observed that most entries of D are close to α. For the cases of $\omega = 0$ and 1, a heuristic proof of this observation and extensive empirical evidence are given in [13]. We have also observed empirically that the observation is valid for $\omega \neq 0, 1$. Although we know of no general proof of the assertion that the entries of D converge to α, we cite the following additional points to support it:

(i) When $|p_1| \leq 1$ and $|p_2| \leq 1$, the analysis of [27] shows that $\{d_{i+k(n+1)}\}_{k\geq 1}$ is convergent with limit α. The proof consists of showing by induction that $d_i \leq d_{i-1}$, $d_i \leq d_{i-n}$, and $\{d_{i+k(n+1)}\}_{k\geq 1}$ is monotonically decreasing and bounded below by 2 (for any $\omega \leq 1$).

(ii) If $|p_1|$, $|p_2|$ are both greater than one and p_1 and p_2 have the same sign, then a straightforward analysis shows that for $\omega \geq 0$,

$$4 \leq d_i \leq 4 + (p_1^2 + p_2^2 - 2 + \omega[(p_2+1)(p_1-1) + (p_1+1)(p_2-1)])/4.$$

Hence, $\{d_i\}$ contains a convergent subsequence. Similarly, if $|p_1|$, $|p_2|$ are both greater than one and p_1 and p_2 have opposite signs, then for $\omega \leq 0$, $\{d_i\}$ contains a convergent subsequence. As we show below, these choices of ω are of primary interest for problems with large $|p_1|$, $|p_2|$.

It is *not* possible to show that the sequence $\{d_i\}$ is bounded or even well-defined for arbitrary p_1 and p_2. In particular, for $p_1 = -14$, $p_2 = 0$, and $\omega = 1$, values at mesh points along the left vertical border of Ω satisfy $d_i = 4 - 16/d_{i-n}$, leading to the choice $d_{n+1} = 0$ and d_{jn+1} undefined for $j > 1$. Other examples with highly oscillatory values are easy to find, see [13]. Nevertheless, in all examples that we have encountered where the RILU factorization is well-defined, the sequence does tend to the limiting value α for indices corresponding to mesh points far from $\partial\Omega$. In the following, our analysis makes use of this observation. In §4, we will describe some techniques for modifying the RILU factorization that ensures that the factors are well-defined.

The second part of the analysis concerns the numerical properties of the triangular solves

$$v \leftarrow L^{-1}w, \qquad v \leftarrow U^{-1}w$$

corresponding to the constant coefficient factors $L = \hat{D} + L_A$, $U = I + \hat{D}^{-1}U_A$. Consider the lower triangular solve. Except for values v_j corresponding to points adjacent to the boundary of Ω, this computation entails the solution of the n'th order recurrence

$$\alpha v_j + \beta v_{j-1} + \gamma v_{j-n} = w_j \tag{3.5}$$

for the values $\{v_j\}_{j=1}^N$, where $\beta = -(1+p_1)$, $\gamma = -(1+p_2)$. But the numerical solution of this recurrence will be unstable if the corresponding characteristic polynomial

$$\gamma_n(z) = \alpha z^n + \beta z^{n-1} + \gamma \tag{3.6}$$

has any roots with modulus greater than one. Thus we have the following definition.

DEFINITION. *The lower triangular solve is stable if the roots of the polynomial* (3.6) *are bounded in modulus by one, and it is unstable otherwise.*

Similarly, the upper triangular solve essentially entails the solution to the recurrence

$$v_j + \delta v_{j+1} + \eta v_{j+n} = w_j,$$

where $\delta = -(1 - p_1)/\alpha$ and $\eta = -(1 - p_2)/\alpha$. The associated characteristic polynomial is

$$\mu(z) = z^n + \delta z^{n-1} + \eta,$$

and stability of the upper triangular solve depends on the maximal root. In [13], we established the following conditions for stability of these computations.[2]

THEOREM 1. *Necessary and sufficient conditions for stability of the lower triangular solve are*
1*l*. *for* $\beta \le 0$ *and* $\gamma \le 0$: $\alpha + \beta + \gamma \ge 0$;
2*l*. *for* $\beta \ge 0$ *and* $\gamma \ge 0$: $-\alpha + \beta + \gamma \le 0$;
3*l*. *for* $\beta \ge 0$ *and* $\gamma \le 0$: $\alpha - \beta + \gamma \ge 0$;
4*l*. *for* $\beta \le 0$ *and* $\gamma \ge 0$: $\alpha + \beta - \gamma \ge 0$.

Necessary and sufficient conditions for stability of the upper triangular solve are
1*u*. *for* $\delta \le 0$ *and* $\eta \le 0$: $1 + \delta + \eta \ge 0$;
2*u*. *for* $\delta \ge 0$ *and* $\eta \ge 0$: $-1 + \delta + \eta \le 0$;
3*u*. *for* $\delta \ge 0$ *and* $\eta \le 0$; $1 - \delta + \eta \ge 0$;
4*u*. *for* $\delta \le 0$ *and* $\eta \ge 0$: $1 + \delta - \eta \ge 0$.

These conditions are equivalent to requiring that both triangular factors be diagonally dominant in rows corresponding to mesh points in the interior of Ω.

Assessing the stability of the preconditioning solves involves invoking the appropriate cases of Theorem 1 for given values of p_1 and p_2. In the following, we refer to subsets of the (p_1, p_2)-plane as the NE ($p_1 \ge 0$, $p_2 \ge 0$), NW ($p_1 \le 0$, $p_2 \ge 0$), SW ($p_1 \le 0$, $p_2 \le 0$), and SE ($p_1 \ge 0$, $p_2 \le 0$) quadrants.

[2] The analysis of [13] actually only applies for odd n in cases 2*l* and 2*u*, and for even n in cases 3*l*, 4*l*, 3*u* and 4*u*; behavior in experiments is independent of parity.

COROLLARY 1. *In the NE quadrant, the triangular solves performed by the RILU preconditioning are stable if and only if one of the following conditions holds:*

$$\begin{array}{ll} p_1 p_2 \leq 1 & \textit{for } \omega < 1 \\ p_1, p_2 \textit{ arbitrary} & \textit{for } \omega = 1. \end{array}$$

In the NW quadrant, the triangular solves are stable if and only if one of the following conditions holds:

$$p_1 \geq -1, \qquad [(1 + \omega)|p_1| - 2]p_2 \leq 2|p_1| - (1 + \omega). \tag{3.7}$$

PROOF. There are four computations to analyze, consisting of the lower triangular solve and upper triangular solve in each of the two quadrants. We present the analysis for the lower triangles only; the arguments for the upper triangles are identical. See also [13].

Lower triangle, NE *quadrant* ($p_1 \geq 0, p_2 \geq 0$). Case 1l of Theorem 1 applies. The condition for stability is

$$2 + [(1 - \omega)(2 + p_1^2 + p_2^2) + \omega(p_1 + p_2)^2]^{1/2} \geq 2 + p_1 + p_2,$$

When $\omega < 1$, this inequality holds when $p_1 p_2 \leq 1$; when $\omega = 1$, it holds for all positive p_1 and p_2.

Lower triangle, NW *quadrant* ($p_1 \leq 0, p_2 \geq 0$). Here $\gamma \leq 0$ for all p_2 under consideration. If $p_1 \geq -1$, then $\beta \leq 0$ and case 1l applies. The stability condition is

$$[(1 - \omega)(2 + p_1^2 + p_2^2) + \omega(p_1 + p_2)^2]^{1/2} \geq p_2 - |p_1|,$$

which is immediate if $p_2 \leq |p_1|$. Otherwise, it follows by squaring both sides of the inequality and simplifying. If $p_1 < -1$, then case 3l applies and the stability condition is

$$[(1 - \omega)(2 + p_1^2 + p_2^2) + \omega(p_1 + p_2)^2]^{1/2} \geq |p_1| + p_2 - 2.$$

Again, the result is immediate if $|p_1| + p_2 - 2 \leq 0$. Otherwise, squaring both sides leads to the inequality (3.7). ■

REMARK 1. The stability regions of the NE and SW quadrants are symmetric with respect to the line $p_1 = -p_2$, and those of the NW and SE quadrants are symmetric with respect to $p_1 = p_2$. Figure 1, which is taken from [13], shows the stability regions for the ILU and MILU preconditioners. For $\omega \in (0, 1)$, the stability region of the RILU preconditioner is a proper subset of that of the ILU preconditioner.

REMARK 2. In the NE and SW quadrants, the restrictions do not depend on ω for $\omega < 1$, so that they are identical to those for the ILU preconditioner ($\omega = 0$). However, as we show in §5.1, the effects of instability in these quadrants become less pronounced as ω increases from 0 to 1. There is no stability restriction when $\omega = 1$.

REMARK 3. In the NW quadrant for $\omega > -1$, equality in (3.7) determines a hyperbola whose distance from the axes decreases as $\omega \to 1$, so that the restrictions are more severe as ω increases. The left side of Figure 1 shows this hyperbola for the case $\omega = 0$, and the right side shows the limiting asymptotes. Rewriting (3.7) in terms of ω gives

$$\omega \le \omega_{\max} \equiv \frac{2(|p_1| + |p_2|)}{1 + |p_1 p_2|} - 1. \tag{3.8}$$

(This expression also applies in the SE quadrant.) That is, for any p_1 and p_2 in the NW or SE quadrants, it is possible to find a value of ω for which the RILU preconditioner is stable.

4. Stabilized incomplete factorizations.

Corollary 1 shows that no fixed choice of ω for the RILU preconditioner produces a stable algorithm on the whole (p_1, p_2)-plane; for any ω, the RILU preconditioner suffers from instability for large enough $|p_1|$ and $|p_2|$. Moreover, the experiments in [13] indicate that for the particular instantiations corresponding to the ILU and MILU preconditioners, performance is poor when they suffer from instability, and it is good otherwise. Consequently, we expect RILU methods not to be robust when they are applied to variable coefficient problems that display different characters in different subregions of Ω (see §5.2). In this section, we consider strategies for stabilizing incomplete factorization methods for problems with large convection terms; these stabilized methods can also be applied to variable coefficient problems.

The stabilizing strategies are based on the results of Theorem 1 and Corollary 1. Consider the constant coefficient case. For any choice of p_1 or p_2, there is some value of ω for which the (constant diagonal) RILU preconditioner is stable. If p_1 and p_2 have the same sign, then the RILU preconditioner with $\omega = 1$ (MILU) is stable, and if p_1 and p_2 have opposite signs, then the RILU preconditioner is stable for

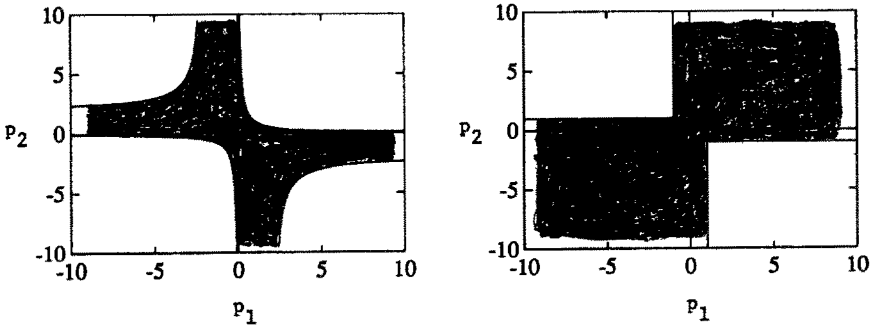

Fig. 1. Stability regions for the ILU (left) and MILU (right) preconditioners.

(algebraically) small enough ω. The idea for the stabilized incomplete factorization is to use an adaptive RILU factorization. If $|p_1|$ and $|p_2|$ are greater than one and have the same sign, then at the ith step of the factorization d_i is computed using the RILU strategy with $\omega = 1$; if $|p_1|$ and $|p_2|$ are greater than one and have opposite signs, then d_i is computed using the RILU strategy with $\omega = \omega_{\max}$. (It is possible to further tune this strategy using $\omega = \omega_{\max} - \delta$, $\delta \geq 0$; to avoid use of another parameter, we do not examine this idea.) Note that d_i is computed using (2.2); it is not replaced by its limiting value. See [28] for results using constant diagonal incomplete factorizations.

It remains to specify the modification strategy to use when one of $|p_1|$, $|p_2|$ is less than or equal to one and the other is greater than one. From (2.3), the off-diagonal error in the ith row of R are

$$(1 + p_2)(1 - p_1)/d_{i-n}, \qquad (1 + p_1)(1 - p_2)/d_{i-1}. \tag{4.1}$$

For large p_1 and p_2, these quantities are negative in the NE and SW quadrants and positive in the NW and SE quadrants. The modification techniques just defined are such that negative quantities are subtracted from the diagonal entry d_i, and positive quantities are added, so that the modification always increases the value on the diagonal. However, in the self-adjoint case ($p_1 = p_2 = 0$), the terms of (4.1) are positive, so that the diagonal d_i is decreased in value by the modification. Despite this, the MILU preconditioner is superior to the ILU preconditioner, and it is nearly the optimal version of RILU(ω). Hence, the sign of the error is not the sole criterion for deciding how to modify. We consider three strategies, all based on the MILU factorization, for performing the rowsum modification when just one of $|p_1|$, $|p_2|$ is less than or equal to one. The two terms of (4.1) are handled separately.

1. If either $|p_1| \leq 1$ or $|p_2| \leq 1$, then perform the modification.
2. If both $|p_1| \leq 1$ and $|p_2| \leq 1$, then perform the modification. If one quantity is greater than one and the other is less than or equal to one, perform the modification only if the error term is negative.
3. Perform the modification if and only if the error term is negative.

The first of these methods performs the modification most aggressively. It is equivalent to the MILU factorization for symmetric problems and for constant coefficient problems when (p_1, p_2) are such that the MILU preconditioner is stable. The third is the least aggressive. Indeed, it does the wrong thing when the first derivative terms are zero, but it has the advantage of being defined by purely algebraic criteria, so that it could be applied in other settings. The second is somewhere in between. There are some constant coefficient problems for which the second and third methods are not equivalent to any RILU factorization. For example, if one of p_1, p_2 is large and the other is small, then one error term is used for the modification but not the other. We denote by SILU$_1$, SILU$_2$, and SILU$_3$, respectively, the methods obtained by combining these strategies with the technique for large $|p_1|$, $|p_2|$ described above.

These methods can be adapted to variable coefficient problems as follows. Consider the problem

$$-(au_x)_x - (bu_y)_y + pu_x + qu_y + cu = f \tag{4.2}$$

on Ω, where a, b, c, p and q are smooth functions and a, b and c are positive on Ω. Second order finite differences for (4.2) are

$$-(au_x)_x \approx 1/h^2(-a_{s+\frac{1}{2},t}u_{s+1,t} + (a_{s+\frac{1}{2},t} + a_{s-\frac{1}{2},t})u_{st} - a_{s-\frac{1}{2},t}u_{s-1,t}),$$

$$-(bu_y)_y \approx 1/h^2(-b_{s,t+\frac{1}{2}}u_{s,t+1} + (b_{s,t+\frac{1}{2}} + b_{s,t-\frac{1}{2}})u_{st} - b_{s,t-\frac{1}{2}}u_{s-1,t}),$$

$$pu_x \approx p_{st}\frac{u_{s+1,t} - u_{s-1,t}}{2h}, \qquad qu_y \approx q_{st}\frac{u_{s,t+1} - u_{s,t-1}}{2h},$$

where for any function F and mesh indices (s, t), $F_{st} = F(sh, th)$. After scaling by h^2, the resulting computational molecule has the form below.

$$\begin{tikzcd}
 & -(A_N - P_N) \arrow[d, dash] & \\
-(A_W + P_W) \arrow[r, dash] & A_C \arrow[r, dash] \arrow[d, dash] & -(A_E - P_E) \\
 & -(A_S + P_S) &
\end{tikzcd}$$

Here

$$A_C = a_{s+\frac{1}{2},t} + a_{s-\frac{1}{2},t} + b_{s,t+\frac{1}{2}} + b_{s,t-\frac{1}{2}} + h^2c_{st},$$

$$A_N = b_{s,t+\frac{1}{2}}, \quad A_S = b_{s,t-\frac{1}{2}}, \quad A_E = a_{s+\frac{1}{2},t}, \quad A_W = a_{s-\frac{1}{2},t},$$

$$P_N = P_S = hq_{st}/2, \qquad P_E = P_W = hp_{st}/2.$$

The errors analogous to (4.1), expressed in terms of matrix indices (as in (2.3)) are proportional to

$$\begin{aligned}(1 + P_S/A_S)_{i,i-n}(1 - P_E/A_E)_{i-n,i-(n-1)},\\ (1 + P_W/A_W)_{i,i-1}(1 - P_N/A_N)_{i-1,i+(n-1)}.\end{aligned} \tag{4.3}$$

The ratios

$$P_S/A_S, \quad P_E/A_E, \quad P_W/A_W, \quad P_N/A_N \tag{4.4}$$

are analogues of the constant values p_1 and p_2, reflecting the relative contributions of first and second order terms in each entry. The actual errors of (4.3) are scaled by A_SA_E/d_{i-n} and A_WA_N/d_{i-1}, respectively. The stabilizing strategies for variable coefficient problems are to examine the signs and values of the four quantities of (4.4). If $|P_S/A_S|$ and $|P_E/A_E|$ are both greater than one and P_S and P_E have the same sign, then the first term of (4.3) (scaled by $A_S A_E/d_{i-n}$) is subtracted from the diagonal of the incomplete factorization. If $|P_S/A_S|$ and $|P_E/A_E|$ are both greater than one but P_S and P_E have opposite sign, then ω times the scaled first term of (4.3) is subtracted from the diagonal, where $\omega = \omega_{\max}$. Here $\omega_{\max}$ is determined from (3.8), using $|P_E/A_E|$ and $|P_S/A_S|$ in place of p_1 and p_2; its value typically varies from step to step during the factorization. The second error term of (4.3) is treated in an analogous way. If some of the ratios $|P/A|$ contributing to (4.3) are less than or equal to one and others are greater than one, then each of the three constant coefficient strategies are similarly generalized.

These methods do not guarantee that the resulting factors are diagonally dominant, which is shown in Theorem 1 to be a necessary condition for stability of the (constant diagonal) factors. Hence, following van der Vorst [26], we supplement these strategies with the requirement

$$d_i \geq \max\left\{\sum_j |l_{ij}|,\ \sum_j |u_{ij}|\right\}.$$

A defining expression for the diagonal analogous to (2.2), which also determines an implementation, is

$$d_i = \max\left\{\hat{d}_i, \sum_j |l_{ij}|,\ \sum_j |u_{ij}|\right\},$$

where

$$\hat{d}_i = a_{ii} - \frac{l_{i,i-1}u_{i-1,i}}{d_{i-1}} - \frac{l_{i,i-n}u_{i-n,i}}{d_{i-n}} - \left[\omega_{i1}\frac{l_{i,i-n}u_{i-n,i-(n-1)}}{d_{i-n}} + \omega_{i2}\frac{l_{i,i-1}u_{i-1,i+(n-1)}}{d_{i-1}}\right]. \tag{4.5}$$

This technique generalizes that of [26], which used only the ILU computation ($\omega = 0$) to construct $\hat{d}_i$. It is also used in the constant coefficient case, to ensure diagonal dominance before diagonal entries achieve their asymptotic values. Here $\{\omega_{i1}\}_{i=1}^N$ and $\{\omega_{i2}\}_{i=1}^N$ are vectors that contain the adaptively determined values of ω used for the modifications in the ith row. For example, for SILU$_1$, ω_{i1} is 1 if $(P_S/A_S)_{i,i-n}$ and $(P_E/A_E)_{i-n,i-(n-1)}$ have the same sign and both have absolute value greater than one, or if at least one of them has absolute value less than or equal to one; $\omega_{i1} = \omega_{\max}$ otherwise. The vectors can be defined at the time the matrix A is constructed (and they can be destroyed after the factors are computed). A consequence is that the SILU factorizations can be vectorized or computed in parallel in a wavefront along diagonal lines of the grid, exactly as for the ordinary RILU factorization; see [1] for details.

5. Numerical experiments.

In this section we present the results of numerical experiments that illuminate the stability analysis of §3 and test the RILU and SILU preconditioners on some sample problems with variable coefficients or nonuniform meshes. All experiments use the preconditioners with the Orthomin(1) iterative method [10]. This method tends to converge quickly for problems when it works, but it is less robust than methods such as GMRES(k) [24] or Orthomin(k) for large k. Thus, it is used to distinguish between strong and weak preconditioners. For all experiments, the preconditioning was applied on the right, i.e. the iteration was equivalent to solving

$$[AQ^{-1}][Qu] = f$$

using Orthomin(1). The stopping criterion for successful convergence was $\|r_i\|_2/\|r_0\|_2 \leq 10^{-6}$, where $r_i = f - Au_i$ is the residual for the ith iterate, and the Euclidean norm is used. Initial guesses will be specified below. Unless otherwise indicated, computations were performed on a VAX-8600 in double precision Fortran.

5.1. *Constant coefficient problems on uniform meshes.*

Tables 1–3 show iteration counts for convergence of Orthomin(1) with RILU preconditioning and several values of ω, for two classes of constant coefficient test problems. For both problem classes, the right hand side of (3.1) was chosen so that the solution is $u(x, y) = xe^{xy}\sin(\pi x)\sin(\pi y)$. The resulting problem satisfies homogeneous Dirichlet boundary conditions $u = g$ on $\partial\Omega$. The iteration counts are the averages (rounded to the nearest integer) over four initial guesses, of which three consist of random vectors with entries in $[-1, 1]$, and the fourth is the zero vector. A maximum of one hundred iterations was permitted. An asterisk (*) indicates that for at least one initial guess, the stopping criterion was not satisfied after one hundred iterations. (Hence, "100*" means that the stopping criterion was not satisfied for any of the inital guesses.) The choice $\omega = 0$ corresponds to the ILU preconditioner and $\omega = 1$ corresponds to the MILU preconditioner.

Table 1 contains data for the NE quadrant, with $P_1 = P_2 = P$ in (3.1), mesh size $h = 1/32$, and $p = p_1 = p_2 = Ph$. The results of §3 show that for $\omega < 1$, the RILU preconditioner is stable if and only if $p \leq 1$, and for $\omega = 1$, there is no stability restriction. The data indicates that instability is correlated with degradation of performance. This is particularly clear for $\omega = 0$ (see also [13]). Performance also degrades as ω decreases from 1 to 0, although good results are obtained for many values of p larger than 1, especially as $\omega \to 1$. We will return to this point later. The MILU preconditioner gives the best results for these problems, even in regions where all preconditioners are stable. Since performance declines with decreasing ω, we did not consider $\omega < 0$ for this quadrant.

Table 1. *Average iteration counts for convergence of Orthomin*(1) *with the RILU preconditioning, in the NE quadrant with* $P_1 = P_2 = P$ and $h = 1/32$.

		ω						
P	Ph	0 ILU	.2	.4	.6	.8	.9	1 MILU
10	.3125	27	25	23	21	17	14	12
20	.6250	16	15	13	11	8	7	7
30	.9375	6	5	5	4	4	4	4
40	1.2500	13	11	9	7	6	6	5
50	1.5625	100*	80*	14	11	9	7	7
60	1.8750	100*	100*	62*	14	11	9	9
100	3.1250	100*	100*	100*	81*	19	15	14
175	5.4688	100*	100*	100*	100*	29	22	20
200	6.2500	100*	100*	100*	100*	47*	24	22

Table 2. *Average iteration counts for convergence of Orthomin*(1) *with the RILU preconditioning, in the NW quadrant with* $-P_1 = P_2 = P$ *and* $h = 1/32$.

		ω						
P	Ph	0 ILU	.2	.4	.6	.8	.9	1 MILU
10	.3125	34	32	29	26	22	18	33
20	.6250	29	27	26	23	18	19	41
30	.9375	28	26	24	21	17	20	48
40	1.2500	27	25	23	21	22	39	100*
50	1.5625	28	25	25	25	38	100*	100*
60	1.8750	29	28	29	34	100*	100*	100*
80	2.5000	32	37	50	100*	100*	100*	100*
100	3.1250	40	62	100*	100*	100*	100*	100*
120	3.7500	59	100*	100*	100*	100*	100*	100*
140	4.3750	100*	100*	100*	100*	100*	100*	100*

Tables 2 and 3 contain data for the NW quadrant, with $-P_1 = P_2 = P, h = 1/32$, and $p = -p_1 = p_2 = Ph$. Table 2 shows results for $\omega \in [0, 1]$ and Table 3 shows results for $\omega \in [-1.2, -.2]$. For this class of problems, performance degrades as ω increases, and the results conform that negative values of ω can be used to improve performance. Table 4 shows $\omega_{\max}$ from (3.8), the maximal value of ω for which stability of the triangular solves is guaranteed. Comparison with Table 3 shows a clear correlation between performance and the stability bound for ω, for each p. (For example, for $p = 1.875$, $\omega_{\max} = .66$, and in Table 2, performance is good for $\omega \leq .6$ and poor for $\omega \geq .8$.) Note also that performance is fairly insensitive to values of ω less than 0.

Table 3. *Average iteration counts for convergence of Orthomin*(1) *with the RILU preconditioning and negative relaxation parameters, in the NW quadrant with* $-P_1 = P_2 = P$ *and* $h = 1/32$.

		ω					
P	Ph	−.2	−.4	−.6	−.8	−1.0	−1.2
10	.3125	35	36	38	39	40	42
20	.6250	30	31	32	33	34	35
30	.9375	29	30	31	32	32	33
40	1.2500	29	30	30	32	32	33
50	1.5625	29	29	30	31	32	34
60	1.8750	29	30	31	31	32	34
80	2.5000	32	32	32	32	32	33
100	3.1250	35	33	33	32	33	34
120	3.7500	41	35	34	33	33	34
140	4.3750	51	39	35	33	33	34

Table 4. *Maximal values of* ω *where stability is guaranteed, for* $-P_1 = P_2 = P$.

Ph	1.25	1.56	1.88	2.50	3.13	3.75	4.38
ω	.91	.82	.66	.34	.16	−.004	−.13

In Table 1, the performance of the RILU preconditioner in the NE quadrant seems to be better for $\omega \in (0, 1)$ than the analysis predicts. We now show that this is an artifact of the problem size. Consider the same type of problems used to generate Table 1 ($p_1 = p_2 = p$), but now let the parameter p be fixed and the mesh size vary. (Thus, the continuous problem (3.1) varies with the mesh size.) Table 5 shows the iteration counts to reach the stopping criterion, for a zero initial guess, where p is fixed at 1.2 ($P = p/h$). The results show that the effects of instability on performance become more pronounced, for p fixed, as the problem size grows. Note that the value $p = 1.2$ is close to the stability bound of $p = 1$; the results of Table 1 show that for larger values of p (and any ω) the effects of instability are evident for smaller problems than are required for this value.

That the results are better as $\omega \to 1$ is intuitively reasonable, since the factors are tending to the MILU factors (for which the computations are stable). Moreover, the dependence on problem size is consistent with the analysis of §3. For example, consider the difference equation (3.5). Suppose a computed solution $\{\hat{v}_j\}$ differs from the exact solution by some error $e_j = \hat{v}_j - v_j$, caused, say, by an error in the initial conditions. Then the errors $\{e_j\}$ satisfy the difference equation (3.5) with $w_j = 0$. If the characteristic polynomial (3.6) has n distinct roots $\{r_i\}$, then the error has the form

$$e_j = c_1 r_1^j + \dots + c_n r_n^j,$$

where $\{c_i\}$ depend on the initial errors (see e.g. [20]). Thus, if any characteristic root r_i has modulus greater than one, then the errors grow with increasing j. Hence, the effects of instability become more pronounced for larger n.

Table 5. *Number of iterations for convergence of Orthomin*(1) *with the RILU preconditioning, for* $p_1 = p_2 = 1.2$ *and varying* h.

	ω				
$1/h$	0	.2	.4	.6	.8
16	7	6	6	5	4
32	14	9	8	6	5
48	100*	13	12	8	6
64	100*	100*	19	9	6
80	100*	100*	100*	14	7
96	100*	100*	100*	17	8
128	100*	100*	100*	100*	9
144	100*	100*	100*	100*	10

Table 6. *Average number of iterations for convergence of Orthomin*(1) *with the ILU, MILU and SILU preconditioners, for* $P_2 = 0$ *and* $h = 1/32$.

p_1	ILU	MILU	SILU$_1$	SILU$_2$	SILU$_3$
.3125	33	17	17	17	33
.6250	23	18	18	18	23
1.2500	18	20	19	18	18
1.8750	16	19	19	17	17
2.5000	15	19	18	16	16
3.1250	15	18	17	15	15
−.3125	41	17	17	17	41
−.6250	29	19	19	19	29
−1.2500	20	21	21	21	21
−1.8750	16	22	20	18	18
−2.5000	15	100*	19	18	18
−3.1250	15	100*	18	18	18

In all of the examples considered so far, the parameters p_1 and p_2 are similar in character to one another, in the sense that both are either less than one in absolute value or both are greater than one in absolute value. Each of the SILU methods automatically reduces to a stable variant of RILU, and they are all robust for solving these problems; we omit detailed tables. Table 6 shows a few results where one of $|p_1|$, $|p_2|$ is less one and the other is greater than one. We consider the case $p_2 = 0$; experiments with $p_1 = 0$ gave similar results. Here, for $p_1 < 1$, SILU$_1$, SILU$_2$ and MILU are all equivalent, and SILU$_3$ is equivalent to ILU. However, for $p_1 > 1$, the latter stabilized methods are new and their performances are good.[3]

[3] Although §3 indicates that the MILU preconditioner is stable, its performance is bad for large negative p_1. On examining D for $p_1 = -2.5$, we found that most of its entries are close to the limit α of (3.3) as in Proposition 1, but a small number of them are negative. If the MILU factors are replaced by constant coefficient factors using α, then performance for $p_1 \leq -2.5$ is good. Hence, we believe that the poor results are due not to instability but to inaccuracy of the factorization. See the discussion following Proposition 1 above.

Finally, note that the optimal choice of the relaxation parameter can vary widely, depending on the choice of problem and stability properties. This contrasts with experimental evidence [1], [3] and Fourier analysis [7] for self-adjoint problems, where the optimal value is typically slightly less than one.

5.2. *Variable coefficient problems.*

In this section, we present some numerical experiments with three test problems with variable coefficients on Ω:
(V1) $-\Delta u + \frac{1}{2}\sigma(1 + x^2)u_x + 100u_y = 0$,
(V2) $-\Delta u + \sigma(1 - 2x)u_x + \sigma(1 - 2y)u_y = 0$,
(V3) $-(e^{-xy}u_x)_x - (e^{xy}u_y)_y + \sigma(x + y)u_x + r(x - y)u_y + 1/(1 + x + y)u = 0$.

For each problem $u = 0$ on $\partial\Omega$. The first two problems are taken from [5]. For the first problem, the coefficients of u_x and u_y are fixed in sign, and the magnitude of the contribution of the u_x derivative varies on Ω. For the other two problems, both the signs and magnitudes of the relative contributions of the terms of different order vary throughout Ω. We present results for the RILU preconditioner with selected values of ω, and for the stabilized preconditioners of §4. In all cases, the reported iteration counts are the averages (rounded to the nearest integer) over three initial guesses consisting of random vectors with entries in $[-1, 1]$.

Problem V1: Table 7 shows the average iteration counts for problem VI. When $\sigma > 0$, this problem is similar in character to the constant coefficient problem with coefficients in the NE quadrant, and when $\sigma < 0$, the problem is similar to the constant coefficient problem for the NW quadrant. The results show that the performance of the ILU, RILU and MILU preconditioned iterative methods is consistent with this observation. For all large positive σ considered, the ILU preconditioning is ineffective and the MILU preconditioning is effective, and for all large negative σ, the ILU preconditioning works and the MILU preconditioning fails. Thus, in both cases, the maximum ω for which the RILU preconditioning can be used is somewhere in $[0, 1]$; performance for $\omega = .5$ is consistent with this observation. The three stabilized preconditioners are robust and effective for both positive and negative σ. (In the table, multiple subscripts mean that the stabilized method with each subscript gave the same results.)

Problem V2: Here we only consider $\sigma > 0$. Figure 2 shows how in Ω the qualitative character of the coefficients corresponds to the four quadrants used to characterize the constant coefficient problem. As the figure indicates, all four quadrants are represented, so that we have no way of choosing a priori a fixed value of ω for the RILU preconditioning. Table 8 shows the performance of the preconditioners. The three stabilized preconditioners are robust on all problems, and they

Table 7. *Average iteration counts for convergence of Orthomin*(1) *with the ILU, RILU, MILU and SILU preconditionings, for Problem* $V1$ *with* $h = 1/32$.

σ	ILU ($\omega = 0$)	RILU ($\omega = .5$)	MILU ($\omega = 1$)	SILU_1	$\text{SILU}_{2,3}$
1	16	17	22	22	16
10	15	17	21	21	15
100	16	9	9	9	9
200	100*	45*	10	10	10
300	100*	22	12	12	12
400	100*	20	13	13	13
500	100*	19	14	14	14
600	100*	19	15	15	15
700	100*	19	15	15	15
800	100*	18	15	15	15
900	100*	17	15	15	15
1000	75*	16	14	15	14

σ	ILU ($\omega = 0$)	RILU ($\omega = .5$)	MILU ($\omega = 1$)	SILU_1	$\text{SILU}_{2,3}$
-1	16	17	22	22	16
-10	16	17	23	23	17
-100	22	19	100*	35	45
-200	24	25	100*	35	35
-300	22	30	100*	37	37
-400	21	85*	100*	34	34
-500	20	100*	100*	31	31
-600	19	100*	100*	28	28
-700	19	100*	100*	25	25
-800	18	100*	100*	23	23
-900	17	100*	100*	22	22
-1000	17	100*	100*	21	21

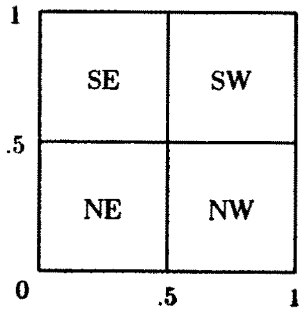

Fig. 2. Partitioning of Ω identifying the character of the coefficients for Problem V2 in terms of the four quadrants associated with constant coefficient problems.

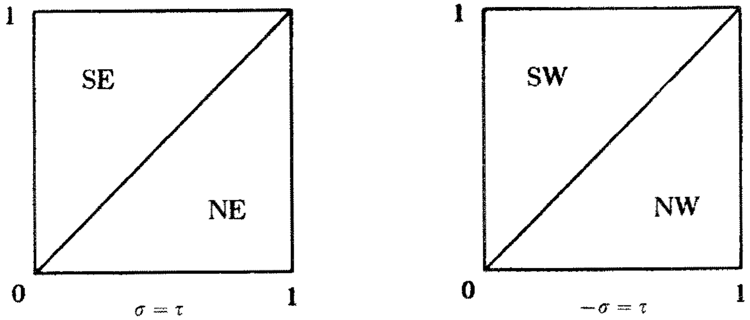

Fig. 3. Partitioning of Ω identifying the character of the coefficients for Problem V3 in terms of the four quadrants associated with constant coefficient problems.

are competitive with the best of the RILU methods when any of them work. The RILU methods are not robust for any of the values of ω considered.

PROBLEM V3: We consider the cases $\sigma = \tau > 0$ and $-\sigma = \tau > 0$. Figure 3 partitions Ω according to the qualitative character of the coefficients of these problems. When $\sigma = \tau$, the NE and SE quadrants are represented, and when $-\sigma = \tau$, the NW and SW quadrants are represented. Thus, there is no obvious fixed value of ω to use. Tables 9 and 10 show the experimental results. In Table 9, a hyphen indicates that a floating overflow occurred during some stage of the computation; in such cases, we did not attempt to test all three initial guesses. Once again, the SILU methods are able to solve all problems considered, whereas the RILU methods are not robust for any of the values of ω considered.

Table 8. *Average iteration counts for convergence of Orthomin*(1) *with the ILU, RILU, MILU and SILU preconditionings, for Problem V2 with $h = 1/32$ and $\sigma > 0$.*

σ	RILU ($\omega = -.5$)	ILU ($\omega = 0$)	RILU ($\omega = .5$)	MILU ($\omega = 1$)	SILU$_1$	SILU$_2$	SILU$_3$
1	23	21	18	19	19	19	21
10	20	19	16	15	15	15	19
100	15	13	11	17	17	15	18
200	22	14	16	100*	21	20	20
300	100*	18	27	100*	24	24	24
400	100*	24	83*	100*	28	29	29
500	100*	31	100*	100*	33	32	32
600	100*	58*	100*	100*	38	38	38
700	100*	68*	100*	100*	39	41	41
800	100*	93*	100*	100*	39	43	43
900	100*	100*	100*	100*	46	44	44
1000	100*	100*	100*	100*	54	46	46

5.3. *Constant coefficient problems on nonuniform meshes.*

Thus far, we have ignored the issue of whether the discrete solution is a good approximation to the continuous solution. If the coefficients of the first derivative terms are large and boundary layers are present in the solution, then with Dirichlet boundary conditions the discrete solution will be highly oscillatory and inaccurate. (See [23] (pp. 161ff), [25] and Figure 4.) However, the presence of such oscillations can be used to determine the existence and location of boundary layers (see [14]); more accurate solutions can then be computed using only local mesh refinement in regions containing the boundary layers [25]. To locate boundary layers, it may be necessary to solve coarse grid problems where the ILU, MILU and RILU precondi-

Table 9. *Average iteration counts for convergence of Orthomin*(1) *with the stabilized ILU preconditionings, for Problem V*3 *with* $\sigma = \tau > 0$ *and* $h = 1/32$.

σ	RILU ($\omega = -.5$)	ILU ($\omega = 0$)	RILU ($\omega = .5$)	MILU ($\omega = 1$)	SILU$_1$	SILU$_2$	SILU$_3$
1	24	22	19	20	20	20	22
10	29	26	22	20	20	20	26
100	23	21	21	30	27	24	22
200	55*	21	24	100*	26	24	24
300	100*	31	100*	100*	28	26	26
400	100*	100*	100*	100*	29	28	28
500	100*	–	100*	–	30	30	30
600	100*	–	100*	–	32	31	31
700	100*	–	100*	–	34	34	34
800	100*	–	100*	–	35	36	36
900	100*	–	100*	–	37	38	38
1000	100*	–	100*	–	39	40	40

Table 10. *Average iteration counts for convergence of Orthomin*(1) *with the stabilized ILU preconditionings, for Problem V*3 *with* $-\sigma = \tau > 0$ *and* $h = 1/32$.

σ	RILU ($\omega = -.5$)	ILU ($\omega = 0$)	RILU ($\omega = .5$)	MILU ($\omega = 1$)	SILU$_1$	SILU$_2$	SILU$_3$
−1	24	22	19	19	19	19	22
−10	26	24	19	19	19	19	24
−100	21	20	25	25	24	20	19
−200	100*	22	100*	100*	24	22	22
−300	100*	100*	100*	100*	27	27	27
−400	100*	100*	100*	100*	28	30	30
−500	100*	100*	100*	100*	29	32	32
−600	100*	100*	100*	100*	31	33	33
−700	100*	100*	100*	100*	33	34	34
−800	100*	100*	100*	100*	34	36	36
−900	100*	100*	100*	100*	36	37	37
−1000	100*	100*	100*	100*	39	39	39

tioners are unstable, as in §5.1. Moreover, when local refinement is used, large portions of the resulting coefficient matrices have the character of those occurring with coarse meshes (i.e. lack of diagonal dominance or M-matrix properties). In this section, we examine the performance of the relaxed and stabilized preconditioners for solving a constant coefficient problem where local refinement is used to achieve accuracy. These experiments were performed on a Sun 3/60 in double precision Fortran. (Cf. [17] and references therein for an alternative approach for achieving accuracy, based on defect correction.)

REMARK 1. The use of upwind difference schemes for the first derivative terms leads to smooth solutions [25], and no instability occurs in the iterative solvers [13]. However, the solutions are less accurate than those acquired using centered differences [25], and they are not of more use for locating boundary layers [14].

REMARK 2. Neumann boundary conditions can improve accuracy of the discrete solution when centered differences are used [19], [23] (p. 165). In our experience the resulting changes in the coefficient matrices have negligible effects on performance of sparse iterative solvers [11], [12].

Consider problem (3.1) where $P_1 = -P$, $P_2 = P$ and the right hand side and boundary conditions are determined by the solution

$$u(x, y) = \frac{e^{2P(1-x)} - 1}{e^{2P} - 1} + \frac{e^{2Py} - 1}{e^{2P} - 1}. \tag{5.1}$$

This function is a two-dimensional variant of a standard one-dimensional problem used in [23], [25]. It is nearly identically zero in Ω except for boundary layers of width $O(\varepsilon)$ near $x = 0$ and $y = 1$, where $\varepsilon = 1/(2P)$. We discretize (5.1) by piecing together two uniform meshes in both the x and y coordinates. For the x-coordinate, we divide the interval $0 < x < 1$ into two subintervals, $(0, 2\sqrt{\varepsilon}]$ and $(2\sqrt{\varepsilon}, 1)$. On the first interval, which contains a boundary layer parallel to the y-axis, we use a uniform mesh of maximal width $\tilde{h}$ satisfying $P\tilde{h} \leq .75$. In the second interval, where the solution is smooth, we use a uniform mesh of size $h = 1/32$. (The spacing in the second interval is uniform as one moves from right to left starting from $x = 1$; the spacing h^* between the rightmost point of the first interval and the leftmost point of the second interval may be different from h and $\tilde{h}$.) The vertical mesh in an analogous way, using the intervals $(0, 1 - 2\sqrt{\varepsilon})$ and $[1 - 2\sqrt{\varepsilon}, 1)$. Note that the intervals of length $O(\sqrt{\varepsilon})$ are conservative for this problem. For small ε, it is likely that good accuracy could be achieved using fewer fine grid points; the resulting matrices would be more like those arising from coarse grids.

Let $\{x_i\}_{i=0}^{n+1}$ and $\{y_j\}_{j=0}^{n+1}$ denote the mesh points in the nonuniform mesh, where n is the number of interior points in each component and $x_0 = y_0 = 0$, $x_{n+1} = y_{n+1} = 1$. Let $h_i^{(x)} = x_1 - x_{i-1}$ and $h_j^{(y)} = y_j - y_{j-1}$. The difference scheme used is given by

$$(5.2)\qquad \begin{aligned} -[u_{xx}]_{ij} &\approx -\frac{2}{h_{i-1}^{(x)}(h_{i-1}^{(x)}+h_i^{(x)})}u_{i-1,j} + \frac{2}{h_{i-1}^{(x)}h_i^{(x)}}u_{ij} \\ &\quad -\frac{2}{h_i^{(x)}(h_{i-1}^{(x)}+h_i^{(x)})}u_{i+1,j}, \\ [u_x]_{ij} &\approx \frac{1}{h_{i-1}^{(x)}+h_i^{(x)}}u_{i+1,j} - \frac{1}{h_{i-1}^{(x)}+h_i^{(x)}}u_{i-1,j}, \end{aligned}$$

with analogous definitions for u_{yy} and u_y. The truncation error of this scheme is of second order except at points at the interfaces between the different intervals, where it is $O(\tilde{h} - h^*)$ or $O(h - h^*)$.

Table 11 and Figure 4 show the effects of local mesh refinement on accuracy. For the values $P = 50$ and $P = 100$, the table reports the size N of the linear system and two error norms for the discrete solution. Here, the discrete l_2 and l_∞ error norms are given by

$$\left[\sum_{i,j=1}^{n} (u_{ij} - u(x_i, y_j))^2 h_i^{(x)} h_j^{(y)}\right]^{1/2}, \qquad \max_{1 \le i,j \le n} |u_{ij} - u(x_i, y_j)|,$$

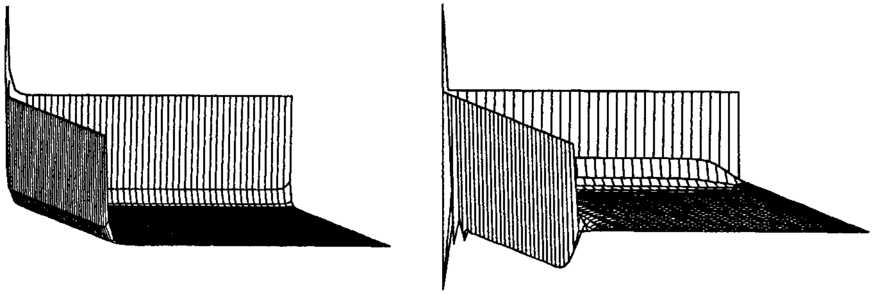

Fig. 4. Plots of the locally refined mesh solution (left) and coarse mesh solution (right), for $-P_1 = P_2 = P = 100$.

respectively. In addition to the locally refined mesh, the table shows results for both the coarse uniform mesh of width $h = 1/32$, and the fine uniform mesh of width $\tilde{h}$. (For $P = 50, \tilde{h} = 1/70$ and there were 25 coarse mesh points and 14 fine mesh points. For $P = 100, \tilde{h} = 7.443 \times 10^{-3} \approx 1/134$, and there were 27 coarse mesh points and 19 fine mesh points.) Figure 4 shows three-dimensional plots of the discrete solutions for the locally refined mesh (left) and the coarse mesh (right), for $P = 100$. The data clearly show that local mesh refinement achieves the same accuracy as uniform mesh refinement, at considerably less cost. The large oscillations present in the coarse mesh solution do not appear when local refinement is used.

Table 11. *Comparison of locally refined mesh, uniform coarse mesh and uniform fine mesh, for* (3.1) *and* $-P_1 = P_2 = P$.

P	Ph	N Coarse	N Local	N Fine	l_2 Error Coarse	l_2 Error Local	l_2 Error Fine	l_∞ Error Coarse	l_∞ Error Local	l_∞ Error Fine
50	1.5625	961	1521	4761	.065	.013	.013	.586	.132	.132
100	3.125	961	2116	17689	.142	.010	.010	1.301	.145	.146

We examined the relaxed and stabilized preconditioners for various values of P. Note that the matrix entries arising from the discretization (5.2) can vary dramatically in scale, depending on whether the entries come from coarse grid points. We present in detail the results of experiments in which the rows of the coefficient matrix were scaled so that all diagonal entries have the value one. (Cf. additional remarks below concerning the effects of scaling.) The resulting linear system is given by

$$D^{-1}Au = D^{-1}f, \tag{5.3}$$

where A is the matrix derived using (5.2) and D is the diagonal part of A. Preconditioning was applied on the right. Table 12 shows the average iteration counts for preconditioned Orthomin(1), over three initial guesses consisting of random vectors with entries in $[-1, 1]$, for solving (5.3) using the MILU, ILU and SILU preconditioners. Table 13 shows the results of using the RILU preconditioner with negative ω. Both tables also show the size N of the linear systems solved; local mesh refinement was used for $P > 30$.

Table 12. *Average iteration counts for convergence of Orthomin*(1) *with the ILU, MILU and SILU preconditioners, for* $-P_1 = P_2 = P$ *and nonuniform meshes.*

P	N	ILU	MILU	SILU$_1$	SILU$_2$	SILU$_3$
20	961	32	34	34	34	32
30	961	32	33	33	33	32
40	1296	32	100*	29	36	32
50	1521	34	100*	37	42	35
60	1681	36	100*	46	45	37
80	1849	39	100*	56	53	40
100	2116	47	100*	65	62	45
120	2304	56	100*	72	69	46
140	2601	90	100*	73	78	49

The results show that the stabilized methods are robust for solving these problems, and the RILU preconditioner is effective when $\omega \leq 0$. The MILU preconditioner (i.e. $\omega = 1$) is *not* effective for $Ph > 1$. In several experiments with $\omega \in (0, 1)$, we found the RILU preconditioner not to be robust. Note that this problem is like a variable coefficient problem with coefficients in the NW quadrant, so that the results are consistent with the analysis of §3 and experiments of §5.1.

Table 13. *Average iteration counts for convergence of Orthomin*(1) *with the RILU preconditioner* ($\omega < 0$), *for* $-P_1 = P_2 = P$ *and nonuniform meshes.*

		ω					
P	N	$-.2$	$-.4$	$-.6$	$-.8$	-1.0	-1.2
20	961	33	34	35	36	37	38
30	961	33	34	35	36	37	38
40	1296	34	35	36	37	38	39
50	1521	35	37	38	39	40	41
60	1681	37	39	40	42	43	43
80	1849	40	40	41	42	44	45
100	2116	45	44	44	45	46	47
120	2304	50	48	47	46	48	49
140	2601	58	52	50	49	50	52

We examined several other problems in conjunction with local mesh refinement; we briefly summarize our observations. First, replacing $e^{2P(1-x)}$ with e^{2Px} in (5.1) leads to a problem (3.1) with $P_1 = P_2 = P$, whose solution has boundary layers at $x = 1$ and $y = 1$. Here the coefficients are in the NE quadrant. We used Orthomin(1) with the ILU, MILU and SILU preconditioners, for $P = 50$ and $P = 100$. In this case, the MILU and SILU preconditioners were robust, and the ILU preconditioned method was unable to solve the problems. Again, this conforms with the results of § 3 and § 5.1. In addition, we performed the same set of experiments used to produce Tables 12 and 13, but without applying the diagonal scaling of (5.2). In these tests, the RILU preconditioner (for all values of ω used in Tables 12 and 13) failed to solve the problems for $P \geq 100$. The SILU$_1$ and SILU$_2$ methods each solved all but one problem (the failures were for $P = 60$ and $P = 80$, respectively). The SILU$_3$ method failed for $P \geq 80$. Thus in general, for problems on nonuniform meshes, the stabilized methods are able to solve a wide class of problems without estimation of parameters. The RILU methods are effective if the right relaxation parameter can be found, but they are sensitive to this choice of parameter.

6. Conclusions.

We have examined the relaxed incomplete LU factorization and developed some stabilized incomplete LU factorizations as preconditioners for solving discrete non-self-adjoint elliptic differential equations. We have found that the relaxed methods suffer from the same type of numerical instabilities exhibited by the more standard ILU and MILU preconditioners, although in some cases the use of a relaxation parameter diminishes the effects of instability. The stabilized methods were developed to prevent numerical instability from playing a role in computations with variable coefficient problems. The heuristics defining them were motivated by the characterization of instability for constant coefficient problems. In a series of numerical experiments with problems having variable coefficients, and with constant coefficient problems on variable meshes, we found that the stabilized methods were considerably more robust than the ILU, MILU or RILU methods. Except for the unscaled problems on nonuniform meshes (where SILU_3 was not robust), the performances of the three stabilized methods were very similar.

Acknowledgements.

The author acknowledges the referees for several useful comments, including the suggestion of taking a more careful look at negative ω, and for pointing out references [27] and [28]. In addition, Henk van der Vorst directed my attention to reference [26], and Gene Golub suggested the point of view taken in Proposition 1.

REFERENCES

[1] C. C. Ashcraft and R. G. Grimes, *On vectorizing incomplete factorization and SSOR preconditioners*, SIAM J. Sci. Stat. Comput. 9: 152–164, 1988.

[2] O. Axelsson and I. Gustafsson, *A modified upwind scheme for convective transport equations and the use of a conjugate gradient method for the solution of non-symmetric systems of equations*, J. Inst. Maths. Applics. 23: 321–337, 1979.

[3] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math. 48: 479–498, 1986.

[4] O. Axelsson and G. Lindskog, *On the rate of convergence of the preconditioned conjugate gradient method*, Numer. Math. 48: 499–523, 1986.

[5] E. F. F. Botta and A. E. P. Veldman, *On local relaxation methods and their application to convection-diffusion equations*, J. Comput. Phys. 48: 127–149, 1981.

[6] A. M. Bruaset, A. Tveito and R. Winther, *On the stability of relaxed incomplete LU factorizations*, Preprint, Institute of Informatics, University of Oslo, 1988.

[7] T. F. Chan, *Fourier Analysis of Blended Incomplete Factorization Preconditioners*, CAM Report 88-34, Department of Mathematics, UCLA, 1988.

[8] P. Concus, G. H. Golub and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Stat. Comput. 6: 220–252, 1985.

[9] T. Dupont, R. P. Kendall and H. H. Rachford Jr., *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal. 5: 559–573, 1968.

[10] S. C. Eisenstat, H. C. Elman and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal. 20: 345–357, 1983.

[11] H. C. Elman, *Preconditioned conjugate-gradient methods for nonsymmetric systems of linear equations*, in R. Vichnevetsky and R. S. Stepleman, Editors, *Advances in Computer Methods for Partial Differential Equations-IV*, IMACS, 1981, pp. 409–417.

[12] H. C. Elman, *Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations*, Ph.D. Thesis, Department of Computer Science, Yale University, 1982.

[13] H. C. Elman, *A stability analysis of incomplete LU factorizations*, Math. Comp. 47: 191–217, 1986.

[14] P. M. Gresho and R. L. Lee, *Don't suppress the wiggles – they're telling you something*, Computers and Fluids 9: 223–253, 1981.

[15] I. Gustafsson, *A class of first order factorizations*, BIT 18: 142–156, 1978.

[16] I. Gustafsson, *Stability and Rate of Convergence of Modified Incomplete Cholesky Factorization Methods*, Ph.D. Thesis, Department of Computer Sciences, Chalmers University of Technology and University of Göteborg, 1978.

[17] W. Hackbusch, *Multi-Grid Methods and Applications*, Sprnger-Verlag, Heidelberg, 1985.

[18] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, New York, 1981.

[19] G. W. Hedstrom and A. Osterheld, *The effect of cell Reynolds number on the computation of a boundary layer*, J. Comput. Phys. 37: 399–421, 1980.

[20] P. Henrici, *Elements of Numerical Analysis*, John Wiley & Sons, New York, 1964.

[21] T. A. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Math. Comp. 34: 473–497, 1980.

[22] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. 31: 148–162, 1977.

[23] P. J. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, 1982.

[24] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 7: 856–869, 1986.

[25] A. Segal, *Aspects of numerical methods for elliptic singular perturbation problems*, SIAM J. Sci. Stat. Comput. 3: 327–349, 1982.

[26] H. A. van der Vorst, *Iterative solution methods for certain sparse linear systems with a non-symmetric matrix arising from PDE-problems*, J. Comput. Phys. 44: 1–19, 1981.

[27] G. Wittum, *On the robustness of ILU-smoothing*, SIAM J. Sci. Stat. Comput. 10: 699–717, 1989.

[28] G. Wittum and F. Liebau, *On Truncated Incomplete Decompositions*, Report 509, SFB 123, Universitat Heidelberg, 1989.