Journal of Computational and Applied Mathematics 39 (1992) 259-275
North-Holland

RILU preconditioning; a computational study

A.M. Bruaset *
Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo 3, Norway

A. Tveito
Center for Industrial Research, P.O. Box 124, Blindern, N-0314 Oslo 3, Norway

Received 10 July 1990
Revised 28 March 1991

Abstract

Bruaset, A.M. and A. Tveito, RILU preconditioning; a computational study, Journal of Computational and Applied Mathematics 39 (1992) 259–275.

We present a computational study of the RILU preconditioning strategy applied to finite-difference discretizations of self-adjoint elliptic boundary value problems with highly discontinuous coefficients. The behaviour of the eigenvalues of the preconditioned system, the numerical stability of the factorization process and the efficiency of the preconditioning strategy are discussed.

Keywords: Incomplete factorizations, preconditioning, conjugate gradient method.

1. Introduction

In this paper we are concerned with the numerical solution of an elliptic partial differential equation of the form

$$-\nabla \cdot (K(x, y)\, \nabla u(x, y)) = f(x, y), \tag{1}$$

combined with suitable boundary conditions. Here K is a strictly positive and bounded function. A standard finite-difference approximation of this equation leads to a linear system of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{n,n}$ is a symmetric positive definite matrix and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$. Since $\boldsymbol{A}$ has only a few

* The work of this author was supported by the Norwegian Research Council for Science and the Humanities.

nonzero diagonals, this system is well suited for solution by an iterative method. For such methods a significant gain in efficiency is achieved when solving the equivalent system

$$M^{-1}Ax = M^{-1}b,$$

where the nonsingular matrix $M \in \mathbb{R}^{n,n}$ in a certain sense approximates A. In particular, we want the preconditioning matrix M to be sparse, symmetric and positive definite.

Over the past years incomplete LU factorizations have become very popular preconditioners for discretizations of elliptic partial differential equations. There are several reasons for this popularity. First of all they constitute efficient preconditioners with moderate storage requirements, cf. [1,8] for a review of the convergence properties of these methods. Secondly, the methods seem to be robust, cf. [6,7] for self-adjoint problems and [11,12] for non-self-adjoint problems. Thirdly, these preconditioners are easy to implement for arbitrary sparsity patterns, cf. [17]. We refer to [1] for a general introduction to the theory of this field. For numerical experiments we refer to a comprehensive study [9], in which different preconditioning strategies based on incomplete block factorizations are compared.

The purpose of this paper is to study the behaviour of the Preconditioned Conjugate Gradient (PCG) method as applied to linear systems of the form (2) arising from equation (1) in the case of highly discontinuous K-functions. In particular, we study the behaviour of the PCG method in conjunction with preconditioners based on the Relaxed Incomplete LU factorization (RILU) proposed in [2,3]. This preconditioning strategy is a combination of the Incomplete LU factorization (ILU) introduced in [18] and the Modified Incomplete LU factorization (MILU) suggested in [14–16], cf. also [10]. The ILU factorization is computed by performing a naive Gaussian elimination, except that the fill-in generated during the elimination process is left out. In the MILU factorization the fill-in terms are added to the main diagonal. The RILU factorization combines these two approaches by multiplying the fill-in by a relaxation parameter $\omega \in [0, 1]$. Thus RILU with $\omega = 0$ is identical with the ILU algorithm and RILU with $\omega = 1$ gives the MILU algorithm [1].

The theory developed for these factorizations is mainly concerned with the case of constant-coefficient elliptic operators, cf. [8]. In this case it is well known that the MILU factorization reduces the spectral condition number of the coefficient matrix from $\mathcal{O}(h^{-2})$ to $\mathcal{O}(h^{-1})$, thus for a given accuracy decreasing the number of iterations from $\mathcal{O}(n^{1/2})$ to $\mathcal{O}(n^{1/4})$. A similar reduction in the condition number is not present for the ILU factorization, but numerical experiments show a significant decrease in the number of iterations needed to achieve the specified accuracy also when using ILU.

In Section 2 we formulate the RILU factorization algorithm for a family of test problems. In Section 3 we present the results of numerical experiments with the method. We discuss the question of finding an optimal value of ω, and we present the spectral condition number κ of $M^{-1}A$ for several grid sizes. On the basis of the computed values of κ, we discuss the usefulness of the traditional convergence theorem for the PCG method. An alternative estimate introduced in [3] is also discussed.

[1] The MILU factorization as defined in [14–16] is applied to the perturbed coefficient matrix $A + cI$, where $c \geq 0$ usually depends on the grid size of the discretization. Using $\omega = 1$ in the RILU algorithm is identical to computing the MILU factorization of A itself, i.e., $c = 0$.

2. RILU factorization for the model problem

We consider the following elliptic boundary value problem:

$$\begin{aligned} &-\nabla \cdot (K(x, y)\, \nabla u(x, y)) = f(x, y), \quad (x, y) \in \Omega \setminus \partial\Omega, \\ &c_1 u(x, y) + c_2 \frac{\partial u}{\partial n}(x, y) = g(x, y), \qquad (x, y) \in \partial\Omega. \end{aligned} \tag{3}$$

Here $\Omega = [0, 1] \times [0, 1]$, $\partial\Omega$ denotes the boundary of Ω and n is the unit outward normal vector. Throughout this paper we require $K(x, y)$ to satisfy the inequality

$$0 < K_m \leqslant K(x, y) \leqslant K_M, \quad \forall (x, y) \in \Omega, \tag{4}$$

where K_m and K_M are finite constants.

We shall discretize (3) by a finite-difference method on a uniform $q \times q$ grid. Depending on the boundary conditions we choose either a *point-centered* or a *block-centered* grid, cf. [19].

To begin with we look at problems with Dirichlet conditions on $\partial\Omega$, i.e., $c_1 = 1$, $c_2 = 0$. Figure 1 shows the point-centered grid which is used for such problems. Since the solution u is known in all boundary nodes, contributions from these nodes will only influence the right-hand side of (2). Thus we need to solve only for the inner nodes (x_i, y_j), where $x_i = ih$ and $y_i = jh$ for $i, j = 1, 2, \ldots, q$, and where $h = 1/(q+1)$.

Denoting the finite-difference approximation to $u(x_i, y_j)$ by $u_{i,j}$ and letting $K_{i+1/2,j}$ represent $K(x_{i+1/2}, y_j)$, we use the following difference approximations:

$$\begin{aligned} \frac{\partial}{\partial x}\left(K\frac{\partial u}{\partial x}\right)_{i,j} &\approx \frac{1}{h^2}\left[K_{i+1/2,j}(u_{i+1,j} - u_{i,j}) - K_{i-1/2,j}(u_{i,j} - u_{i-1,j})\right], \\ \frac{\partial}{\partial y}\left(K\frac{\partial u}{\partial y}\right)_{i,j} &\approx \frac{1}{h^2}\left[K_{i,j+1/2}(u_{i,j+1} - u_{i,j}) - K_{i,j-1/2}(u_{i,j} - u_{i,j-1})\right]. \end{aligned} \tag{5}$$

These approximations give rise to a sparse linear system of equations of the form $Ax = b$ of order $n = q^2$, where the vector x contains the unknowns $u_{i,j}$, and where b contains contribu-

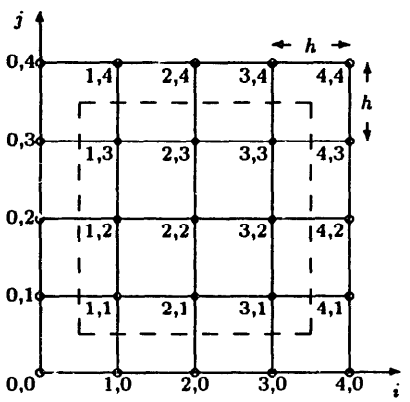

Fig. 1. The point-centered grid for $h = 1/(q+1) = \frac{1}{4}$. We want to compute the numerical solution to (3) for all nodes inside the dashed box. The remaining nodes (∘) lie on the boundary $\partial\Omega$ and are subject to Dirichlet conditions.

tions from the functions f and g in (3). The coefficient matrix is symmetric and has the form

$$A = \begin{pmatrix} \gamma_{1,1} & \beta_{1,1} & 0 & \cdots & 0 & \alpha_{1,1} & 0 & \cdots & 0 \\ \beta_{1,1} & \gamma_{2,1} & \beta_{2,1} & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \beta_{2,1} & \gamma_{3,1} & \beta_{3,1} & & & & \ddots & 0 \\ \vdots & \ddots & \beta_{3,1} & \ddots & \ddots & & & \ddots & \alpha_{q,q-1} \\ 0 & & & \ddots & & & & & 0 \\ \alpha_{1,1} & \ddots & & & & & & \ddots & \vdots \\ 0 & \ddots & & & & & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & \beta_{q-1,q} \\ 0 & \cdots & 0 & \alpha_{q,q-1} & 0 & \cdots & 0 & \beta_{q-1,q} & \gamma_{q,q} \end{pmatrix}. \tag{6}$$

After scaling the matrix and right-hand side by h^2, the matrix entries for the Dirichlet case are given by

$$\alpha_{ij} = \begin{cases} -K_{i,j+1/2}, & j \neq q, \\ 0, & j = q, \end{cases} \qquad \beta_{i,j} = \begin{cases} -K_{i+1/2,j}, & i \neq q, \\ 0, & i = q, \end{cases}$$
$$\gamma_{i,j} = K_{i-1/2,j} + K_{i+1/2,j} + K_{i,j-1/2} + K_{i,j+1/2}. \tag{7}$$

The indices i and j vary from 1 to q.

If $c_1 = 0$ and $c_2 = 1$ in (3), we have a Neumann problem. Just like in the Dirichlet case we can approximate the differential equation in (3) using the finite differences (5), but we have to take some extra precautions due to the new boundary conditions. For such problems it is common to use a block-centered grid as shown in Fig. 2. The nodes (x_i, y_j) in this grid have coordinates $x_i = (i - \frac{1}{2})h$ and $y_j = (j - \frac{1}{2})h$, $i, j = 1, 2, \ldots, q$, where $h = 1/q$. The Neumann

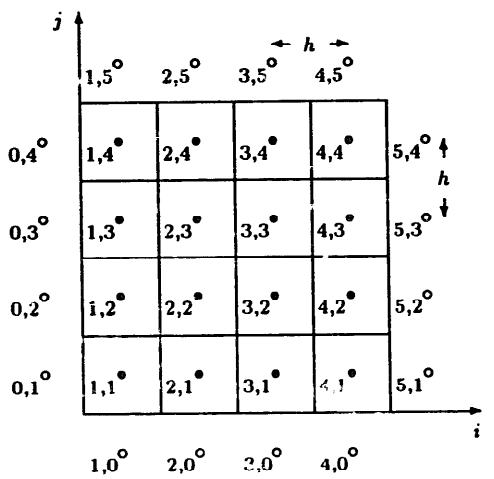

Fig. 2. The block-centered grid for $h = 1/q = \frac{1}{4}$. Using artificial nodes (∘) placed outside Ω we are able to implement the Neumann conditions by means of finite differences.

type boundary conditions can now be implemented as centered differences by introducing artificial nodes outside Ω. In this case the system matrix A of the form (6) has the following entries:

$$\alpha_{i,j} = \begin{cases} -K_{i,j+1/2}, & j \neq q, \\ 0, & j = q, \end{cases} \qquad \beta_{i,j} = \begin{cases} -K_{i+1/2,j} & i \neq q, \\ 0, & i = q, \end{cases}$$
$$\gamma_{i,j} = -(\alpha_{i,j-1} + \beta_{i-1,j} + \beta_{i,j} + \alpha_{i,j}), \tag{8}$$

for $i, j = 1, 2, \ldots, q$, where $\alpha_{i,0} = \beta_{0,j} = 0$.

One problem associated with the use of difference approximations like (5) is that we have to evaluate the coefficient function K in midpoints $(x_{i+1/2}, y_j)$ and $(x_i, y_{j+1/2})$. In many applications the function K is known only in the nodes (x_i, y_j), and the values $K_{i+1/2,j}$ and $K_{i,j+1/2}$ have to be estimated by some procedure. Another complicating factor is that K can have jump discontinuities. In such cases it is common to use harmonic mean values as K-values in the midpoints,

$$K_{i+1/2,j} = \frac{2K_{i,j}K_{i+1,j}}{K_{i,j} + K_{i+1,j}}, \qquad K_{i,j+1/2} = \frac{2K_{i,j}K_{i,j+1}}{K_{i,j} + K_{i,j+1}}, \tag{9}$$

where $K_{i,j} = K(x_i, y_j)$. See [4] for more details.

Now that we have a linear system of equations of the form $\boldsymbol{Ax} = \boldsymbol{b}$, we want to find the RILU factorization of A which can be used as a preconditioner in a combination with the PCG method. The general RILU algorithm is fully described in [2], but we will give a version which is adapted to problems where the coefficient matrix has the structure described above. Allowing fill-in from the elimination process only in the positions corresponding to the five nonzero diagonals of A, the following algorithm will compute the nonzero entries of the matrices L and U such that $A = LU - R$. The sparsity structure of A is maintained in both L and U.

Algorithm 1 (*RILU factorization of pentadiagonal matrices*). Given a matrix $A \in \mathbb{R}^{n,n}$ by (6). Let $\omega \in [0, 1]$. The RILU factorization is defined by the following algorithm.

$c_{1,1} := \gamma_{1,1}$
for $i := 1$ to q do
  $\rho_{i,1} := \gamma_{i,1}$
for $j := 1$ to $q - 1$ do
begin
  for $i := 1$ to $q - 1$ do
  begin
    $b_{i,j} := \beta_{i,j}/c_{ij}$
    $c_{i+1,j} := \rho_{i+1,j} - b_{i,j}(\beta_{i,j} + \omega\alpha_{i,j})$
    $a_{i,j} := \alpha_{i,j}/c_{i,j}$
    $\rho_{i,j+1} := \gamma_{i,j+1} - a_{i,j}(\alpha_{i,j} + \omega\beta_{i,j})$
  end
  $b_{q,j} := \beta_{q,j}/c_{q,j}$
  $c_{1,j+1} := \rho_{1,j+1} - b_{q,j}(\beta_{q,j} + \omega\alpha_{q,j})$
  $a_{q,j} := \alpha_{q,j}/c_{q,j}$

$a_{q,j} := \alpha_{q,j}/c_{q,j}$
$\rho_{q,j+1} := \gamma_{q,j+1} - a_{q,j}(\alpha_{q,j} + \omega\beta_{q,j})$
end
for $i := 1$ **to** $q-1$ **do**
begin
$b_{i,q} := \beta_{i,q}/c_{i,q}$
$c_{i+1,q} := \rho_{i+1,q} - b_{i,q}\beta_{i,q}$
end

For our model problem, the factors L and U computed by Algorithm 1 have the form

$$L = \begin{pmatrix} 1 & 0 & \cdots & & & & & \cdots & 0 \\ b_{1,1} & 1 & \ddots & & & & & & \vdots \\ 0 & b_{2,1} & 1 & & & & & & \\ \vdots & \ddots & b_{3,1} & \ddots & & & & & \\ 0 & & & \ddots & & & & & \\ a_{1,1} & \ddots & & & & & & & \\ 0 & \ddots & & & & & & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{q,q-1} & 0 & \cdots & 0 & b_{q-1,q} & 1 \end{pmatrix}, \tag{10}$$

$$U = \begin{pmatrix} c_{1,1} & \beta_{1,1} & 0 & \cdots & 0 & \alpha_{1,1} & 0 & \cdots & 0 \\ 0 & c_{2,1} & \beta_{2,1} & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & c_{3,1} & \beta_{3,1} & & & & \ddots & 0 \\ & & & \ddots & \ddots & & & \ddots & \alpha_{q,q-1} \\ & & & & & & & & 0 \\ & & & & & & & \ddots & \vdots \\ & & & & & & & \ddots & 0 \\ \vdots & & & & & & \ddots & \ddots & \beta_{q-1,q} \\ 0 & \cdots & & & & & \cdots & 0 & c_{q,q} \end{pmatrix}. \tag{11}$$

The subdiagonal entries of L not equal to zero are defined by

$$a_{i,j} = \frac{\alpha_{i,j}}{c_{i,j}} = -\frac{K_{i,j+1/2}}{c_{i,j}}, \qquad b_{i,j} = \frac{\beta_{i,j}}{c_{i,j}} = \begin{cases} -\dfrac{K_{i+1/2,j}}{c_{i,j}}, & i \neq q, \\ 0, & i = q, \end{cases} \tag{12}$$

for $i, j = 1, 2, \ldots, q$. The nonzero superdiagonal entries of U, $\alpha_{i,j}$ and $\beta_{i,j}$, are still given by (7) or (8) for Dirichlet and Neumann problems, respectively.

We are going to use $M = LU$ as a preconditioner for the system (2), and we are therefore concerned with the existence and stability of RILU factorizations. Analyzing Algorithm 1, we observe that the only critical points in the factorization process are when we calculate the fractions $\alpha_{ij}/c_{i,j}$ and $\beta_{i,j}/c_{i,j}$. It is known that under mild restrictions on A we obtain $c_{i,j} \neq 0$, cf. [1,2,16,18]. Although this property implies the existence of RILU factorizations in a mathematical sense, we have no guarantee that the algorithm is numerically stable. The computations may break down due to overflow or underflow caused by $c_{i,j}$ assuming a very small or very large value. Inspired by these observations we use the following definition of a stable factorization.

Definition 2. The RILU factorization of the model problem described by Algorithm 1 is called a *stable factorization* if there exist two constants c_m, c_M, $0 < c_m < c_M < \infty$, independent of the mesh size h such that

$$c_{i,j} \in [c_m, c_M], \quad i, j = 1, 2, \ldots, q.$$

In [7] it is shown that the RILU factorization for $\omega \leqslant 1 - \mathcal{O}(h)$ applied to the Dirichlet case of our model problem with a smooth K-function is stable according to Definition 2. This result is in [6] extended to the general case of positive, bounded K-functions and all $\omega \in [0, 1]$.

3. Numerical experiments with RILU

In this section we will investigate the performance of the RILU preconditioner combined with the PCG method for four different test problems of the form (3). This combination of RILU and PCG is for easy reference denoted by RILUCG.

We will not discuss details of the PCG method in this paper, for such information, see [13]. However, it should be mentioned that in all experiments we use $x^{(0)} = 0$ as start vector, and unless otherwise stated the iterations are halted when

$$\|r^{(k)}\|_2 \leqslant \epsilon \|r^{(0)}\|_2, \tag{13}$$

where $\epsilon = 10^{-4}$, and $r^{(k)} = b - Ax^{(k)}$ is the residual computed in iteration k.

For all four test problems we solve the corresponding system of equations for several grid sizes h and for a number of different relaxation parameters ω. In this way we estimate the optimal choice of ω with respect to minimization of the number of iterations needed by the PCG algorithm. This optimal value will vary for each problem and grid size. Whenever the smallest number of iterations is achieved for more than one value of ω, the optimal ω is chosen as the value giving the least relative residual error in terms of (13).

All solutions are computed on a RISC based DECstation 3100. If nothing else is said, all computations are done in double precision.

3.1. Problem 1: Discontinuities along grid lines

For a very simple test problem we apply the coefficient function

$$K(x, y) = \begin{cases} d, & (x, y) \in \Omega' = [\frac{1}{3}, \frac{2}{3}] \times [\frac{1}{3}, \frac{2}{3}], \\ 1, & (x, y) \in \Omega \setminus \Omega', \end{cases} \tag{14}$$

Table 1
Number of iterations for Problem 1

ω	$d=1$			$d=10^3$			$d=10^5$		
	$n=5476$	$n=10816$	$n=22201$	$n=5476$	$n=10816$	$n=22201$	$n=5476$	$n=10816$	$n=22201$
0	35	49	69	60	81	114	75	103	142
0.5	30	41	58	52	71	98	65	88	123
0.9	22	29	41	36	50	66	45	63	86
0.95	20	26	35	34	43	59	42	53	74
0.96	20	25	34	33	42	58	41	52	72
0.97	19	24	32	33	41	54	42	51	67
0.98	18	23	30	32	40	51	39 *	51	63
0.99	18	22	29	31	39	49	40	47	59
0.991	18 *	22	28	31 *	38	48	40	46	57
0.992	18	22 *	28	31	38	48	40	46 *	58
0.993	18	22	28	31	37 *	47	40	46	57
0.996	19	22	26 *	31	37	45 *	41	48	55 *
0.999	21	25	28	35	39	46	45	51	61
1	23	28	35	32	43	54	40	50	60

Estimated optimal values of ω are marked with the symbol *.

for a given constant d. Furthermore, the right-hand side of (3) is set to $f(x, y) \equiv 1$, and we let $c_1 = 1$, $c_2 = 0$ and $g(x, y) \equiv 0$ giving homogeneous Dirichlet conditions on the boundary. We will use three different values of d including $d = 1$ which reduces the problem to the standard Poisson equation.

For the particular problem given by (14) all discontinuities are along lines parallel to the x- or y-axis. This observation puts us in a position where we can easily construct a grid for which the discontinuities are located along grid lines. We achieve this effect by choosing $q = 3m - 1$ for $m \in \mathbb{N}$.

In Table 1 we list the number of iterations needed by RILUCG to find the solution with the accuracy requirements described above. The optimal parameter $\omega = \omega_{opt}$ marked with the symbol * is estimated for each combination of system size $n = q^2$ and d. We note that as n increases, ω_{opt} tends towards $\omega = 1$ for all values of d. This observation coincides with results reported in [2]. It suggests that $\omega_{opt} = 1 - \delta h$ for some positive constant δ, and for $d = 10^3$ and $\epsilon = 10^{-7}$ it is reported that $\delta \approx 1.8$. However, our experiments show clearly that the number of iterations increase only a small amount when using other ω-values reasonably close to the optimal choice. In fact, $\omega = 1$ is a rather good choice, which can also reduce the number of multiplications in the RILU algorithm.

Another noteworthy observation is that the number of iterations increases slowly as a function of n and d. It is quite remarkable that the number of iterations is only roughly doubled when the ratio K_M/K_m defined by (4) grows from 1 to 10^5.

When using the PCG method, we know that the relative error measured in "A-norm" is reduced by a factor of ϵ within

$$k = \left\lceil \tfrac{1}{2}\sqrt{\kappa(M^{-1}A)}\ \ln \frac{2}{\epsilon} \right\rceil \tag{15}$$

iterations, cf. [1]. Here M is the preconditioning matrix and $\kappa(M^{-1}A)$ is the spectral condition

number of $M^{-1}A$. This convergence estimate motivates a study of the condition number $\kappa(M^{-1}A)$ as a function of d and n. Using a Lanczos method implemented in the LASO package [2], we have computed the extreme eigenvalues of $M^{-1}A$ for $\omega = 0$, ω_{opt}, 1. All computations are done in double precision on a Sun 4/260. The results are shown in Table 2. The numbers in parentheses are the actual values for ω_{opt} used in each case.

The most striking effect shown in Table 2 is that the condition number is huge when d is large and $\omega < 1$. On the other hand, for $\omega = 1$ the condition number is reasonably small, and is slowly increasing as a function of the jump d. Assuming that $\kappa(M^{-1}A) = \mathcal{O}(h^{\alpha})$, we have used linear regression to estimate the growth rate α for $\omega = 0$, 1 and fixed values of d. These results are given in the bottom rows of each part of the table.

Restricting our attention to the largest value of n, we see a clear relation between the condition number and the values of d. For $\omega = 0$ and $\omega = \omega_{\text{opt}} = 0.996$ the condition numbers $\kappa(M^{-1}A)$ grow at approximately the same rate as d. For $\omega = 1$ the behaviour is quite different. Increasing d from 1 to 10^3 gives a modest rise in the condition number from 48 to 280. Going further and using $d = 10^5$, we get $\kappa(M^{-1}A) \approx 284$.

From the numbers in Table 2 and the convergence estimate (15) we would expect $\omega = 1$ to be the optimal value. However, we have already observed that this is not the case. Also, since the real ω_{opt} is rather close to one, the large difference in condition numbers for $\omega = \omega_{\text{opt}}$ and $\omega = 1$ is interesting. Using $n = 5476$ and $\omega = \omega_{\text{opt}} = 0.99$ the estimate (15) yields the maximum number of iterations $k = 4468$, which is an exaggerated figure compared to the actual number of iterations $k = 31$. This example shows that the traditional estimate is not very useful for problems involving variable coefficients with highly discontinuous behaviour. In general, if we examine the distribution of eigenvalues of $M^{-1}A$, we will find that some values in one or both ends of the spectrum are isolated from the rest. Under such circumstances PCG will converge faster than suggested by the traditional estimate.

In [3] three corrected estimates depending on the shape of the eigenvalue distribution are proposed. Let $S(M^{-1}A) = \{\lambda_i\}_{i \geq 1}$ denote the subset of disjoint eigenvalues of $M^{-1}A$ ordered in increasing order. Consider the following three nonuniform distributions of eigenvalues:

$$S(M^{-1}A) \in [a, b] \cup \left(\bigcup_{i=1}^{s} \lambda_i' \right), \qquad b < \lambda_s', \tag{16}$$

$$S(M^{-1}A) \in \left(\bigcup_{i=1}^{r} \lambda_i \right) \cup [a, b], \qquad \lambda_r < a, \tag{17}$$

$$S(M^{-1}A) \in \left(\bigcup_{i=1}^{r} \lambda_i \right) \cup [a, b] \cup \left(\bigcup_{i=1}^{s} \lambda_i' \right), \quad \lambda_r < a < b < \lambda_s', \tag{18}$$

where λ_i', $i = 1, 2, \ldots, s$, denotes the s largest eigenvalues in decreasing order. Axelsson and Lindskog [3] derive corrected convergence estimates based on Chebyshev polynomials for problems with an eigenvalue distribution of one of the forms (16)–(18). For such distributions the corrected estimates are much more accurate than (15). Examining the eigenvalue distribution for our model problem we find that the smallest eigenvalue is isolated from the rest, i.e., we have an eigenvalue distribution of the form (17). According to [3] we can then use the

[2] For details about this package we refer to the LASO2 Documentation by D.S. Scott (Comput. Sci. Dept., Univ. Texas, Austin). Both docmentation and source code are available from the NETLIB service.

Table 2
Condition numbers for Problem 1

n	$d=1$			$d=10^3$			$d=10^5$			
	$\omega=0$	$\omega=\omega_{opt}$	$\omega=1$	$\omega=0$	$\omega=\omega_{opt}$	$\omega=1$	$\omega=0$	$\omega=\omega_{opt}$		$\omega=1$
2500	93.975	18.299 (0.97)	15.359	33469.270	5161.199 (0.98)	70.846	3096827.5	71.570 (1)		71.570
3481	129.765	20.772 (0.98)	18.278	46207.575	5168.461 (0.99)	88.043	4587892.6	88.99[illegible] (1)		88.994
5476	202.292	22.672 (0.991)	23.197	72418.830	7750.665 (0.991)	117.971	6881838.9	814687.03	(0.99)	119.338
7921	290.936	32.380 (0.99)	28.168	105842.91	11796.119 (0.99)	149.058	10322758	134267.49	(0.992)	150.865
10816	395.830	39.023 (0.992)	33.180	142177.15	13589.523 (0.993)	180.986	13763667	1448678.8	(0.992)	183.243
22201	806.817	56.008 (0.996)	48.386	290695.54	31346.883 (0.996)	280.161	24774620	2137738.9	(0.996)	283.675
$\kappa(M^{-1}A)$	$\mathcal{O}(h^{-1.969})$		$\mathcal{O}(h^{-1.051})$	$\mathcal{O}(h^{-1.987})$		$\mathcal{O}(h^{-1.260})$	$\mathcal{O}(h^{-1.906})$			$\mathcal{O}(h^{-1.262})$

The figures given in parentheses are the actual values of ω_{opt} in each case.

Table 3
A comparison of estimated and actual number of iterations for Problem 1 with fixed $d = 10^5$ and fixed $\omega = 0.99$

n	λ_1	λ_2	$\max_i \lambda_i$	Traditional estimate	Corrected estimate ($r = 1$)	Actual number of iterations
2500	$2.35 \cdot 10^{-5}$	0.80572	8.66908	3008	38	26
3481	$1.69 \cdot 10^{-5}$	0.74684	8.74074	3562	40	27
5476	$1.08 \cdot 10^{-5}$	0.64884	8.79047	4468	44	31
7921	$7.5 \cdot 10^{-6}$	0.55781	8.80667	5366	48	34
10816	$5.5 \cdot 10^{-6}$	0.47769	8.81204	6268	52	37
22201	$2.7 \cdot 10^{-6}$	0.30420	8.81465	8948	68	46

corrected estimate

$$k = \left\lfloor \left(\ln \frac{2}{\epsilon} + \sum_{i=1}^{r} \ln \frac{b}{\lambda_i} \right) / \ln \sigma^{-1} \right\rfloor + r, \tag{19}$$

where $\sigma = (1 - \sqrt{a/b})/(1 + \sqrt{a/b})$, and in our case $r = 1$. Table 3 compares the estimates (15) and (19) to the actual number of iterations for Problem 1 with fixed $d = 10^5$ and fixed $\omega = 0.99$. For this purpose we halt the PCG iterations when

$$\| x - x^{(k)} \|_A \leqslant \epsilon \| x - x^{(0)} \|_A,$$

where [3] the "A-norm" is defined by $\| x \|_A = (x^T A x)^{1/2}$ and $\epsilon = 10^{-4}$. We see that the new convergence estimate gives reasonable results and is a dramatical improvement compared to the traditional estimate.

3.2. *Problem 2: Discontinuities along a circle*

In the previous problem we were able to construct a grid such that all discontinuities were located along grid lines. However, in general it can be quite difficult to construct such a grid. This is clearly seen if we choose the coefficient function

$$K(x, y) = \begin{cases} d, & (x, y) \in \Omega' = \left\{ (x, y) : \left(x - \tfrac{1}{2}\right)^2 + \left(y - \tfrac{1}{2}\right)^2 \leqslant \tfrac{1}{9} \right\}, \\ 1, & (x, y) \in \Omega \setminus \Omega', \end{cases} \tag{20}$$

where $d = 10^3, 10^5$. As in Problem 1 we use $f(x, y) \equiv 1$, $g(x, y) \equiv 0$, $c_1 = 1$ and $c_2 = 0$.

Table 4 shows the number of iterations used by RILUCG. For comparison with the figures in Table 1 we have used the same grid sizes n as we did for Problem 1. It is evident that the optimal value of ω does not differ significantly from the results reported in Table 1. However, in Table 4 the number of iterations for $\omega = \omega_{\text{opt}}$ has increased due to the fact that the discontinuities are no longer located along grid lines. Viewing the number of iterations as a function of d, we find that the increase is of the same magnitude as before.

The most outstanding difference between the two tables is that while $\omega = 1$ was a reasonable choice in Problem 1, this value is definitely the worst possible in Problem 2. In particular, we note that the increase in iterations as we approach $\omega = 1$ is very sudden. Note also that for

[3] To get an estimate of $x = A^{-1}b$ we solve the system requiring that $\| r^k \|_2 \leqslant 10^{-10} \| r^0 \|_2$.

Table 4
Number of iterations for Problem 2

ω	$d = 10^3$			$d = 10^5$		
	$n = 5476$	$n = 10816$	$n = 22201$	$n = 5476$	$n = 10816$	$n = 22201$
0	65	92	130	78	106	150
0.5	57	80	114	68	92	132
0.9	43	59	85	52	71	100
0.95	39	53	75	48	65	90
0.96	39	52	71	48 *	62	88
0.97	39	51	68	48	62	83
0.98	37 *	48	66	48	59	80
0.99	39	47 *	59	51	58 *	74
0.993	40	49	58	53	59	73 *
0.994	41	47	57 *	53	61	73
0.999	48	56	65	61	72	85
1	96	143	222	174	280	466

Estimated optimal values of ω are marked with the symbol *.

$d = 10^5$ the amount of computational work can be reduced by a factor of six by using RILU with a proper value of ω instead of MILU (i.e., $\omega = 1$). These observations suggest that the RILU preconditioner may be unstable with respect to perturbations in ω. In Section 3.5 we will investigate the stability of the RILUCG method more thoroughly.

3.3. Problem 3: Rectangular barriers

The general problem (3) can lead to a model applicable for two-dimensional incompressible flow of a single fluid in a heterogeneous medium. Let

$$K(x, y) = \begin{cases} 10^{-5}, & (x, y) \in \Omega' = \bigcup_{i=1}^{5} \Omega_i, \\ 1, & (x, y) \in \Omega \setminus \Omega', \end{cases} \tag{21}$$

where Ω_i, $i = 1, \ldots, 5$, are rectangular barriers with edges parallel to the x- and y-axis. These barriers, which in Fig. 3 are drawn with a dark pattern, represent zones of rock with lower permeability than the rest of the reservoir. In addition we apply the right-hand side $f(x, y) \equiv 0$ and the homogeneous Neumann condition given by $g(x, y) \equiv 0$, $c_1 = 0$ and $c_2 = 1$. The injection and production wells are implemented by specifying the value of the pressure $u(x, y)$ in two nodes:

$$\begin{aligned} u_{1,1} &= u\left(\tfrac{1}{2}h, \tfrac{1}{2}h\right) = 1, \\ u_{q,q} &= u\left(1 - \tfrac{1}{2}h, 1 - \tfrac{1}{2}h\right) = 0. \end{aligned} \tag{22}$$

These two extra conditions impose some modifications on the coefficient matrix A defined by (6) and (8). On the main diagonal we will have $\gamma_{1,1} = \gamma_{q,q} = 1$, while the off-diagonal entries $\alpha_{1,1}$, $\alpha_{q,q-1}$, $\beta_{1,1}$ and $\beta_{q-1,q}$ are replaced by zeros. In addition, the right-hand side $\boldsymbol{b}$ in (2) is modified accordingly.

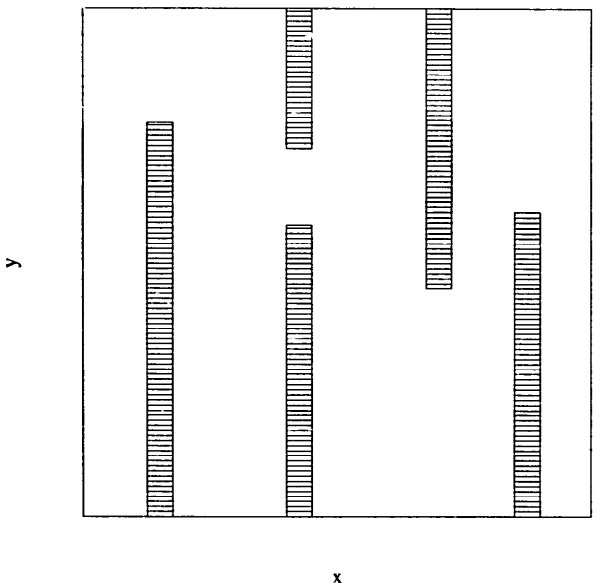

Fig. 3. The domain Ω for Problem 3. Inside the rectangular barriers Ω_i, $i = 1,\ldots, 5$, $K(x, y)$ is set to 10^{-5}, while $K(x, y) = 1$ elsewhere.

Models similar to the one described above are used in [5,19].

As discussed in Section 2 we solve the boundary problem on a block-centered grid. Since we have rectangular barriers as shown in Fig. 3, we could find an h sufficiently small to place all discontinuities along grid lines. However, this would be very difficult or even impossible to do for more realistic problems where the barriers can have quite irregular shapes. We therefore find it natural not to exploit the regularity of the current problem so that our results will be comparable to the results obtained for a more complex geometry.

From Table 5, which shows the number of iterations needed by RILUCG for Problem 3, we see that ω_{opt} is in this case very close or even equal to one. This observation conforms with the

Table 5
Number of iterations for Problem 3

ω	$n = 4900$	$n = 10000$	$n = 22500$
0	137	198	298
0.5	117	167	252
0.9	80	115	172
0.95	69	97	147
0.96	66	93	139
0.97	63	88	130
0.98	58	80	118
0.99	52	71	101
0.999	33	46	66
0.9999	28 *	34 *	45
1	30	36	45 *

Estimated optimal values of ω are marked with the symbol *.

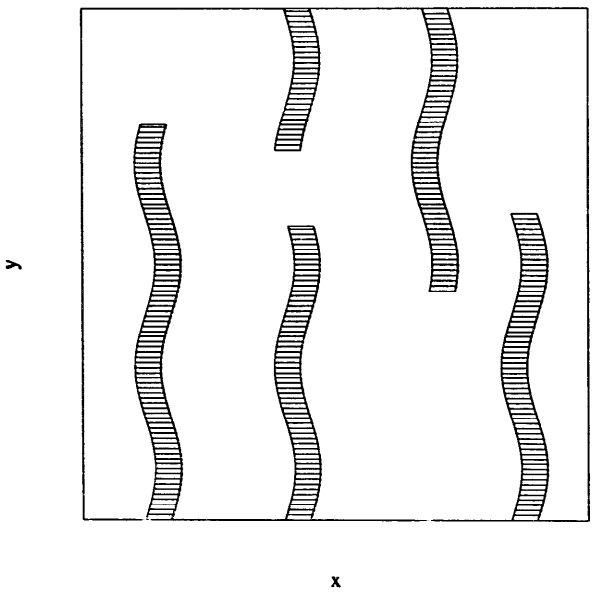

Fig. 4. The domain Ω for Problem 4. Inside the curved barriers Ω_i, $i=1,\ldots,5$, $K(x, y)$ is set to 10^{-5}, while $K(x, y)=1$ elsewhere.

tendency we saw in the case of Problem 1, even though the effect is stronger for the current problem. For practical purposes one would probably choose the relaxation parameter $\omega = 1$ independent of n. We also note that increasing n from 4900 to 22500 gives a moderate growth of approximately 61% in the number of iterations for $\omega = \omega_{opt}$.

3.4. Problem 4: Curved barriers

When discussing the modelling properties of the previous problem we mentioned that a real reservoir problem would not necessarily have rectangular barriers. In this section we will study a problem with curved barriers. We still define the coefficient function by (21), but now Ω_i, $i = 1,\ldots, 5$, are the curved barriers shown in Fig. 4. As before we use $f(x, y) \equiv g(x, y) \equiv 0$, $c_1 = 0$ and $c_2 = 1$. Injection and production are still implemented by the two conditions in (22).

Using the same grids and preconditioners as for the previous problem we find the iteration counts listed in Table 6. Optimal values for ω are still close to one, but not as close as in the case of rectangular barriers. We see that ω_{opt} approaches one as h tends to zero.

Comparing Problems 3 and 4 we see that the number of iterations has increased in the latter case. The increase in iterations as a function of n is also larger in this case, increasing n from 4900 to 22500 gives a growth of 81% in the number of iterations for $\omega = \omega_{opt}$ opposed to 61% for Problem 3. However, the most remarkable effect introduced by allowing curved barriers is that $\omega = 1$ is no longer a good choice. In fact, this parameter value is nearly as bad as the worst case $\omega = 0$. It would obviously be unwise to use the simple poblem with rectangular barriers to estimate an optimal ω for practical reservoir simulations where the coefficient $K(x, y)$ can have almost random behaviour.

Table 6
Number of iterations for Problem 4

ω	$n = 4900$	$n = 10000$	$n = 22500$
0	139	199	301
0.5	118	170	257
0.9	84	121	185
0.95	74	105	161
0.96	72	102	153
0.97	69	96	145
0.98	65	90	135
0.99	61	83	120
0.999	52 *	69	96
0.9995	53	66 *	96
0.9997	55	68	94 *
0.9999	62	75	103
1	133	182	346

Estimated optimal values of ω are marked with the symbol *.

3.5. Numerical instabilities

When discussing Problems 2 and 4, we mentioned that the number of iterations used by RILUCG increases very sudden as ω tends to one. This indicates that the RILU preconditioner is unstable with respect to perturbations in ω. In this section we will study this problem by comparing two implementations of RILUCG numerically. In the previous computations all floating-point operations are carried out in double-precision arithmetic. We will now use the same programs except that all calculations in the RILU and PCG algorithms are done in single precision. To assure that we solve identical problems in both cases, we still use double precision when building the coefficient matrix A. The convergence criterion is as described above, cf. (13).

As previously discussed, the interesting values of ω are close to one. We will therefore restrict ω to the interval [0.96, 1]. We also fix n to the largest value for each problem, i.e., $n = 22201$ for Problems 1, 2 and $n = 22500$ for Problems 3, 4. For the two first problems d is fixed at 10^5. The numbers of iterations used by the two implementations are shown as functions of ω in Fig. 5. Results obtained with single precision are marked with dashed lines, while the solid lines represent the double-precision computations.

When looking at the curves for Problems 1 and 2, we see that the single-precision solver needs a considerably larger number of iterations to reach convergence than its double-precision equivalent. This is in particular true for $\omega = 1$, e.g., for Problem 2 the difference is 849 iterations. We also observe that the iteration counts in the case of single precision oscillate for these problems. The double-precision solver gives results of a much more monotone behaviour. As far as monotonity is concerned, the two implementations give almost identical results for Problems 3 and 4. However, there is still a significant difference between the two solvers when ω reaches one. This difference is again largest when it comes to a problem where the discontinuities do not coincide with grid lines.

We would like to make sure whether the effects reported above are caused by the RILUCG method or are a result of a particular implementation of computer arithmetics. To check this,

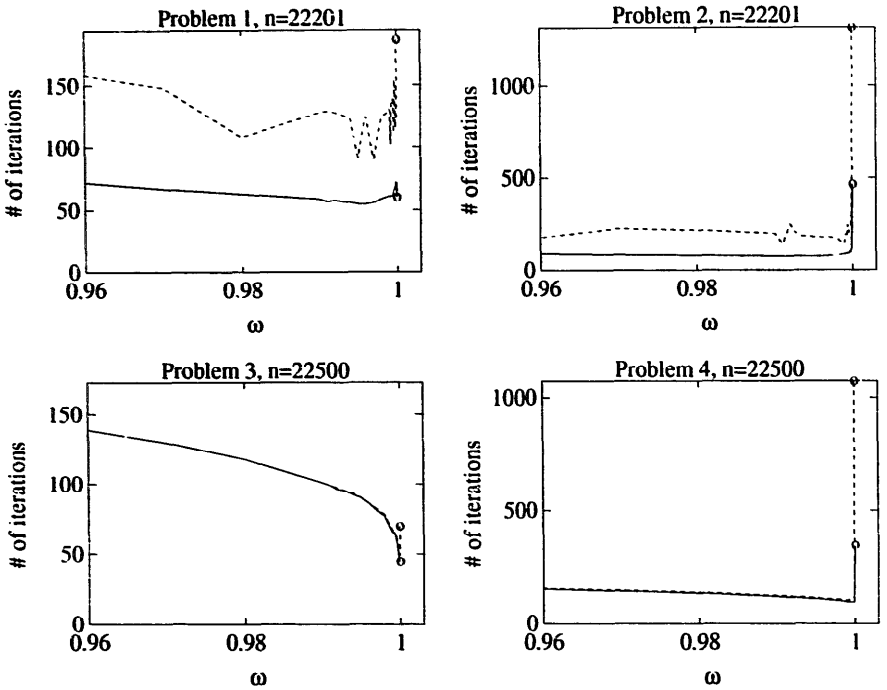

Fig. 5. A comparison between iteration counts obtained with single- and double-precision implementations of RILUCG. Dashed lines represent the single-precision computations, while results from the double-precision solver are marked with solid lines. The symbol ∘ indicates the number of iterations needed for $\omega = 1$.

we have performed all computations in this section on two different computers and compared the results. For every combination of test problem and floating-point representation the numbers of iterations used on a DECstation 3100 are identical to the iteration counts obtained with a Sun 4/260. This observation strongly indicates that the effects summarized in Fig. 5 are due to the chosen method and are independent of the hardware platform for which this method is implemented.

4. Conclusions

We have presented a computational study of the RILU preconditioning strategy applied to finite-difference discretizations of four test problems of the form (3), where the coefficient function $K(x, y)$ has a highly discontinuous behaviour. In all cases the RILU factorizations are stable in the sense of Definition 2, and can be used as preconditioners. We have experienced that the traditional convergence estimate for PCG is not of much use for our type of problems, and that corrected estimates adapted to the eigenvalue distribution of the preconditioned difference operator are preferable.

From a practical point of view it is interesting to know if we increase the efficiency of the preconditioner by introducing the parameter ω. We have observed that the proper choice of ω seems to reduce the number of iterations. In fact, for one test problem we have seen a reduction in the number of iterations by a factor of six. Thus we conclude that there are problems for which RILU with optimal ω is more efficient than the ILU ($\omega = 0$) and MILU ($\omega = 1$) preconditioners.

A disadvantage of RILU is that the optimal parameter value ω_{opt} has to be estimated. This is a highly nontrivial task due to the fact that RILUCG is numerically unstable with respect to perturbations in ω. We have shown numerically that when $\omega \simeq 1$, the number of iterations can vary significantly with small changes of ω. Another effect that should be noted is that the behaviour of RILUCG is highly dependent on whether we use single- or double-precision variables when implementing the method. The number of iterations is in general larger for single than for double precision, and the actual value of ω_{opt} can be quite different in the two implementations. This also indicates that an analytical estimate, if such could be derived, would be of limited practical use.

In all cases we have seen that ω_{opt} is close to one. However, one should not be tempted to uncritically use $\omega = 1$ as we have examples where this value is among the worst-possible choices.

References

[1] O. Axelsson and V.A. Barker, *Finite Element Solution of Boundary Value Problems. Theory and Computation* (Academic Press, London, 1984).

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

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

[4] K. Aziz and A. Settari, *Petroleum Reservoir Simulation* (Applied Science Publishers, New York, 1979).

[5] B.H. Begg, R.R. Carter and P. Dranfield, Assigning effective values to simulator gridblock parameters for heterogeneous reservoirs, *SPE Res. Engrg.* **4** (1989) 455–463.

[6] A.M. Bruaset and A. Tveito, A simplified MILU-like preconditioner for elliptic problems, Report 91689-1, Center for Industrial Research, Oslo, 1991.

[7] A.M. Bruaset, A. Tveito and R. Winther, On the stability of relaxed incomplete LU factorizations, *Math. Comp.* **54** (1990) 701–719.

[8] T.F. Chan and H.C. Elman, Fourier analysis of iterative methods for elliptic problems, *SIAM Rev.* **31** (1989) 20–49.

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

[10] 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** (1968) 559–573.

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

[12] H.C. Elman, Relaxed and stabilized incomplete factorizations for non-self-adjoint linear systems, *BIT* **29** (1989) 890–915.

[13] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, MD, 2nd ed., 1989).

[14] I. Gustafsson, A class of first order factorization methods, *BIT* **18** (1978) 142–156.

[15] I. Gustafsson, On modified incomplete Cholesky factorization methods for the solution of problems with mixed boundary conditions and problems with discontinuous material coefficients, *Internat. J. Numer. Methods Engrg.* **14** (1979) 1127–1140.

[16] I. Gustafsson, Stablility and rate of convergence of modified incomplete Cholesky factorization methods, Ph.D. Thesis, Dept. Comput. Sci., Chalmers Univ. of Technology and Univ. Göteborg, 1979.

[17] H.P. Langtangen, Conjugate gradient methods and ILU preconditioning of nonsymmetric matrix systems with arbitrary sparsity patterns, *Internat. J. Numer. Methods Fluids* **9** (1989) 213–233.

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

[19] D.W. Peaceman, *Fundamentals of Numerical Reservoir Simulation* (Elsevier, New York, 1977).