Journal of Computational Mathematics, Vol.22, No.6, 2004, 833–856.

COMBINATIVE PRECONDITIONERS OF MODIFIED INCOMPLETE CHOLESKY FACTORIZATION AND SHERMAN-MORRISON-WOODBURY UPDATE FOR SELF-ADJOINT ELLIPTIC DIRICHLET-PERIODIC BOUNDARY VALUE PROBLEMS *1)

Zhong-zhi Bai Gui-qing Li
(*LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*)

Lin-zhang Lu
(*Department of Mathematics, Xiamen University, Xiamen 361005, China*)

Abstract

For the system of linear equations arising from discretization of the second-order self-adjoint elliptic Dirichlet-periodic boundary value problems, by making use of the special structure of the coefficient matrix we present a class of combinative preconditioners which are technical combinations of modified incomplete Cholesky factorizations and Sherman-Morrison-Woodbury update. Theoretical analyses show that the condition numbers of the preconditioned matrices can be reduced to $\mathcal{O}(h^{-1})$, one order smaller than the condition number $\mathcal{O}(h^{-2})$ of the original matrix. Numerical implementations show that the resulting preconditioned conjugate gradient methods are feasible, robust and efficient for solving this class of linear systems.

Mathematics subject classification: 65F10, 65F50.
Key words: System of linear equations, Conjugate gradient method, Incomplete Cholesky factorization, Sherman-Morrison-Woodbury formula, Conditioning.

1. Introduction

Consider the two-dimensional second-order self-adjoint elliptic partial differential equation

$$-\nabla \cdot (a(\xi,\eta) \cdot \nabla u) + \theta(\xi,\eta) \cdot u = f(\xi,\eta) \tag{1.1}$$

in the unit square $\Omega = (0,1) \times (0,1)$ with the boundary conditions

$$\begin{cases} u(0,\eta) &=& g_0^{(1)}(\eta), & u(1,\eta) &=& g_1^{(1)}(\eta), \\ u(\xi,0) &=& g_0^{(2)}(\xi), & u(\xi,1) &=& g_1^{(2)}(\xi), \end{cases}$$

where $a(\xi,\eta)$ is a positive and piecewise differentiable function, $\theta(\xi,\eta)$ is a nonnegative bounded function, and $g_0^{(1)}(\eta)$, $g_1^{(1)}(\eta)$, $g_0^{(2)}(\xi)$, $g_1^{(2)}(\xi)$ and $f(\xi,\eta)$ are bounded functions. The case that $a(\xi,\eta) = 1$, $\theta(\xi,\eta) = 0$ and $g_0^{(1)}(\eta) = g_1^{(1)}(\eta) = g_0^{(2)}(\xi) = g_1^{(2)}(\xi) = 0$ has been extensively studied in literatures, e.g., [1, 12, 15, 16]. In this paper, we will study the case that

$$g_0^{(1)}(\eta) = g_1^{(1)}(\eta) \equiv g^{(1)}(\eta), \tag{1.2}$$

i.e., the boundary conditions are periodic on the ξ-direction and Dirichlet on the η-direction, respectively. Moreover, for simplicity but without loss of generality, we assume that $\theta(\xi,\eta) = 0$ and $g_0^{(2)}(\xi) = g_1^{(2)}(\xi) \equiv 0$ in the sequel.

* Received November 1, 2002; final revised April 20, 2004.
1) Subsidized by The Special Funds For Major State Basic Research Projects (No. G1999032803) and the National Natural Science Foundation (No. 10471146).

When the second-order self-adjoint elliptic Dirichlet-periodic boundary value problem (1.1)-(1.2) is discretized by the five-point central difference scheme with mesh size $h = \frac{1}{N+1}$, associated with the interior mesh point (ih, jh) we have the difference equation

$$s_{i,j}u_{i,j} - a_{i-\frac{1}{2},j}u_{i-1,j} - a_{i+\frac{1}{2},j}u_{i+1,j} - a_{i,j-\frac{1}{2}}u_{i,j-1} - a_{i,j+\frac{1}{2}}u_{i,j+1} = h^2 f_{i,j},$$

where

$$s_{i,j} = a_{i-\frac{1}{2},j} + a_{i+\frac{1}{2},j} + a_{i,j-\frac{1}{2}} + a_{i,j+\frac{1}{2}},$$

and for $j = 1, 2, \ldots, N$, we stipulate that $a_{(N+i)+\frac{1}{2},j} = a_{i-\frac{1}{2},j}$ in the light of the periodicity of the boundary condition (1.2). By arranging the unknowns $\{u_{i,j}\}_{1\le i\le N+1, 1\le j\le N}$ according to the natural ordering and letting $n = (N+1)N$, we obtain the system of linear equations:

$$\mathbf{A}x = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{n\times n} \text{ symmetric positive definite, and } \mathbf{b} \in \mathbb{R}^n, \tag{1.3}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & B_1 & & & \\ B_1 & \mathbf{A}_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{N-2} & \mathbf{A}_{N-1} & B_{N-1} \\ & & & B_{N-1} & \mathbf{A}_N \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} h^2 f_{1,1} \\ h^2 f_{1,2} \\ \vdots \\ h^2 f_{N+1,N-1} \\ h^2 f_{N+1,N} \end{pmatrix}, \tag{1.4}$$

and for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, N-1$,

$$\mathbf{A}_i = \begin{pmatrix} a_1^{(i)} & d_1^{(i)} & & & \sigma^{(i)} \\ d_1^{(i)} & a_2^{(i)} & d_2^{(i)} & & \\ & \ddots & \ddots & \ddots & \\ & & d_{N-1}^{(i)} & a_N^{(i)} & d_N^{(i)} \\ \sigma^{(i)} & & & d_N^{(i)} & a_{N+1}^{(i)} \end{pmatrix}, \quad B_j = \begin{pmatrix} b_1^{(j)} & & & & \\ & b_2^{(j)} & & & \\ & & \ddots & & \\ & & & b_N^{(j)} & \\ & & & & b_{N+1}^{(j)} \end{pmatrix}. \tag{1.5}$$

The sub-matrices $\mathbf{A}_i \in \mathbb{R}^{(N+1)\times(N+1)} (i = 1, 2, \ldots, N)$ are symmetric positive definite whose elements are defined by

$$a_j^{(i)} = s_{j,i}, \quad d_j^{(i)} = -a_{j+\frac{1}{2},i}, \quad \sigma^{(i)} = -a_{i-\frac{1}{2},i};$$

and the sub-matrices $B_i \in \mathbb{R}^{(N+1)\times(N+1)} (i = 1, 2, \ldots, N-1)$ are diagonal whose elements are defined by

$$b_j^{(i)} = -a_{j,i+\frac{1}{2}}.$$

Clearly, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is an irreducibly diagonally dominant Z-matrix. Therefore, it is an M-matrix. And so are the sub-matrices $\mathbf{A}_i$ $(i = 1, 2, \ldots, N)$. We refer the readers to [17, 18] for details.

The preconditioned conjugate gradient (PCG) method[11, 7, 10] is one of the most powerful methods for getting an accurate approximation to the solution $x^* \in \mathbb{R}^n$ of the system of linear equations (1.3). As a matter of fact, if a symmetric positive definite matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is employed as a preconditioner to the coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, then the corresponding PCG iteration converges to x^* within a relative error ε in at most $\frac{1}{2}\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})}\ln\frac{2}{\varepsilon} + 1$ number of iteration steps[2], where $\kappa(\mathbf{M}^{-1}\mathbf{A})$ represents the Euclidean condition number of the preconditioned matrix $\mathbf{M}^{-1}\mathbf{A}$. See also [9, 10, 4, 6]. Therefore, a good preconditioner is the key factor to considerably improve the convergence behaviour of the PCG iteration.

As we know, standard preconditioners to a symmetric positive definite matrix may be constructed by the *incomplete Cholesky* (IC) factorization[2, 10] and the *symmetric successive overrelaxation* (SSOR) iteration[17, 18, 1] techniques. See also [3, 5, 8, 15, 16]. However, these two classes of preconditioners are only applicable and efficient for a special class of symmetric positive definite matrix, e.g., a diagonally dominant or an irreducibly weakly diagonally dominant one[14, 12]. Moreover, the IC factorization may break down even for a symmetric positive definite matrix[13].

Considering the special structure of the system of linear equations (1.3)-(1.5), in this paper we present a class of combinative preconditioners to the coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ by technically combining *modified incomplete Cholesky* (MIC) factorizations[12] and *Sherman-Morrison-Woodbury* (SMW) update[9]. Theoretical analyses show that with these new preconditioners, the condition numbers of the preconditioned matrices can be reduced to $\mathcal{O}(h^{-1})$, one order smaller than the condition number $\mathcal{O}(h^{-2})$ of the original matrix $\mathbf{A}$. The feasibility, robustness and efficiency of the new preconditioners are further confirmed by numerical implementations of several examples of the second-order self-adjoint elliptic Dirichlet-periodic boundary value problem (1.1)-(1.2).

The organization of this paper is as follows: In Section 2 we define the combinative preconditioners, In Section 3 we establish several lemmas which are essential for discussing theoretical properties of the new preconditioners. The existence of the new preconditioners and the condition numbers of the preconditioned matrices are studied in Section 4. Finally, in Section 5, several numerical examples are implemented to show the feasibility, robustness and efficiency of the resulting preconditioned conjugate gradient iterations.

2. New Preconditioners

Let $e = (1, 0, \ldots, 0, 1)^T \in \mathbb{R}^{N+1}$. Then for $i = 1, 2, \ldots, N$ we have

$$\mathbf{A}_i = A_i + \sigma^{(i)} ee^T,$$

where

$$A_i = \begin{pmatrix} a_1^{(i)} - \sigma^{(i)} & d_1^{(i)} & & & \\ d_1^{(i)} & a_2^{(i)} & d_2^{(i)} & & \\ & \ddots & \ddots & \ddots & \\ & & d_{N-1}^{(i)} & a_N^{(i)} & d_N^{(i)} \\ & & & d_N^{(i)} & a_{N+1}^{(i)} - \sigma^{(i)} \end{pmatrix}. \tag{2.1}$$

If we introduce a matrix

$$A = \begin{pmatrix} A_1 & B_1 & & & \\ B_1 & A_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{N-2} & A_{N-1} & B_{N-1} \\ & & & B_{N-1} & A_N \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{2.2}$$

and N vectors

$$u_i = e_{(i-1)(N+1)+1} + e_{i(N+1)}, \qquad i = 1, 2, \ldots, N,$$

where

$$e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T \in \mathbb{R}^n,$$

then the coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ can be expressed as a rank-N update of the matrix A, i.e.,

$$\mathbf{A} = A + \sum_{i=1}^{N} \sigma^{(i)} u_i u_i^T. \tag{2.3}$$

Evidently, the matrix A is a symmetric positive definite M-matrix. From [14, 12] we know that it possesses MIC factorizations of the form $A = LL^T - R$, which is a regular splitting[17]. Here, $R = D + \widehat{R}$, $\widehat{R} = (\hat{r}_{i,j}) \in \mathbb{R}^{n\times n}$ is a negative semidefinite matrix (i.e., $(\widehat{R}x, x) \leq 0$ for any $x \in \mathbb{R}^n$) satisfying $\sum_j \hat{r}_{i,j} = 0$, $\forall i \in \{1, 2, \ldots, n\}$, and $D \in \mathbb{R}^{n\times n}$ is a positive diagonal matrix whose choice depends on the boundary conditions. Now, the structured expression (2.3) immediately leads to an approximate matrix

$$\mathbf{M} = M + \sum_{i=1}^{N} \sigma^{(i)} u_i u_i^T, \qquad \text{where } M = LL^T, \tag{2.4}$$

to the coefficient matrix $\mathbf{A}$. When the matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is symmetric positive definite and is employed to precondition the system of linear equations (1.3), we need to solve a generalized residual equation of the form $\mathbf{M}z = r$, for a given right-hand side $r \in \mathbb{R}^n$, at each of the PCG iteration steps, or equivalently, to compute the generalized residual vector $z = \mathbf{M}^{-1}r$. This can be efficiently realized by the well-known SMW formula (see Lemma 2.1). In this way, a class of combinative preconditioners based on the MIC factorizations and the SMW updates for the second-order self-adjoint elliptic Dirichlet-periodic boundary value problem (1.1)-(1.2) is well defined.

More precisely, in the following we will further describe the processes of both MIC factorizations of the matrix $A \in \mathbb{R}^{n\times n}$ and SMW inversions of the matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$.

2.1. The MIC factorizations

Consider the second-order self-adjoint elliptic Dirichlet-periodic boundary value problem(1.1) - (1.2). For simplicity, in the following we use difference stencils to show which grid points are involved, and coefficient notations for the matrices A, L, LL^T and $\widehat{R}$, regarded as operators (or corresponding matrices) applied to grid functions. In this notation, the matrix A is defined in Figure 2.1, where $m = N + 1$ is the band width of the matrix, and

$$\begin{aligned}
\alpha_{(\ell-1)m+j} &= \begin{cases} a_j^{(\ell)} - \sigma^{(\ell)}, & \text{for} \quad j = 1 \text{ or } m, \\ a_j^{(\ell)}, & \text{otherwise}, \end{cases} \qquad 1 \le \ell \le m-1, 1 \le j \le m, \\
\beta_{(\ell-1)m+j} &= -d_j^{(\ell)}, \quad 1 \le \ell, j \le m-1, \\
\gamma_{(\ell-1)m+j} &= -b_j^{(\ell)}, \quad 1 \le \ell \le m-2, 1 \le j \le m.
\end{aligned}$$

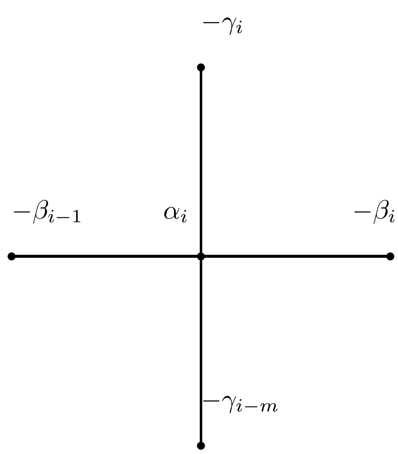

Figure 2.1. The five-point difference stencil of A

2.1.1. The MIC(0) formula

In this method, the matrix L has nonzero entries in positions where the lower part of the matrix A has nonzero entries. The involved matrices L, L^T, LL^T and $\widehat{R}$ are defined in Figures 2.2-2.5. This then results in the MIC(0) factorization $A = M - R$, where $M = LL^T$, $R = D + \widehat{R}$, $D = \psi h^2 \cdot \mathrm{diag}(A)$ ($\psi > 0$), with $\widehat{R} = (\hat{r}_{i,j})$ being negative semidefinite and $\sum_j \hat{r}_{i,j} = 0$, $\forall i$.

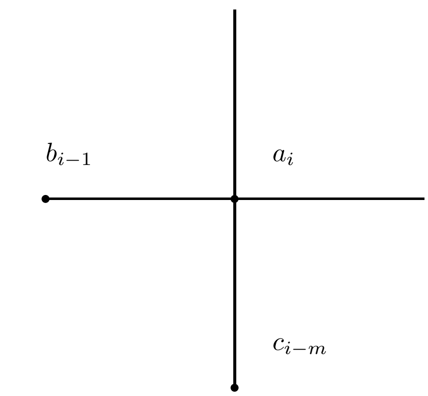

Figure 2.2. The difference stencil of L

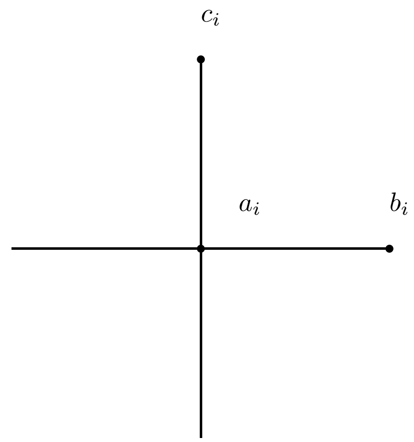

Figure 2.3. The difference stencil of L^T

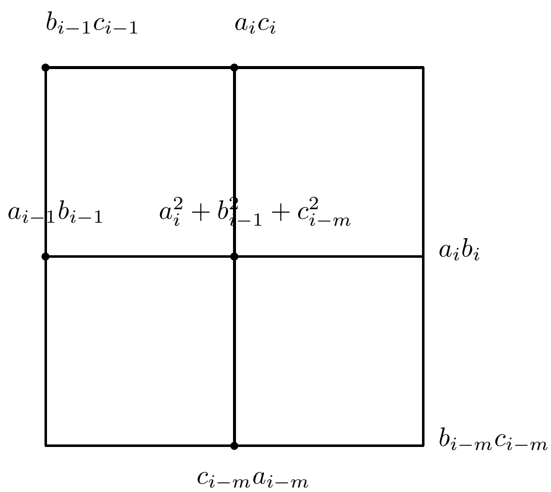

Figure 2.4. The difference stencil of LL^T

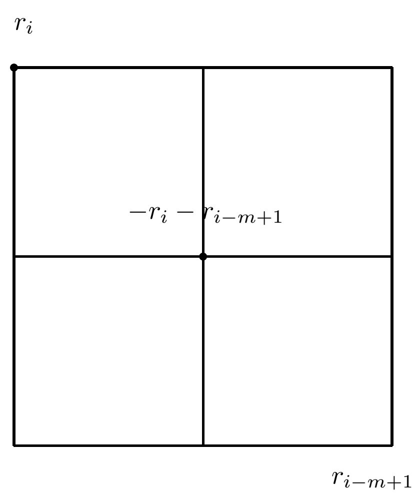

Figure 2.5. The difference stencil of $\widehat{R}$

According to [12], from Figures 2.1-2.5 we know that the entries of the matrices L and $\widehat{R}$ satisfy the following recursive formulas:

$$\left\{\begin{array}{lcl} a_i^2 & = & \alpha_i(1+\delta) - r_i - r_{i-m+1} - b_{i-1}^2 - c_{i-m}^2, \\ b_i & = & -\frac{\beta_i}{a_i}, \\ c_i & = & -\frac{\gamma_i}{a_i}, \\ r_i & = & b_{i-1}c_{i-1}, \\ \delta & = & \psi h^2, \quad \psi > 0, \end{array}\right. \tag{2.5}$$

where entries not defined should be replaced by zeros.

2.1.2. The MIC(1) formula

A natural step to get a more accurate factorization is to allow the matrix L to have nonzero entries in the positions where the matrix $\widehat{R}$, in the MIC(0), has nonzero entries. This leads to the MIC(1) factorization defined in Figures 2.6-2.8. In specific, we have $A = M - R$, where $M = LL^T$, $R = D + \widehat{R}$, $D = \psi h^2 \cdot \mathrm{diag}(A)$ $(\psi > 0)$, with $\widehat{R} = (\hat{r}_{i,j})$ being negative semidefinite and $\sum_j \hat{r}_{i,j} = 0$, $\forall i$.

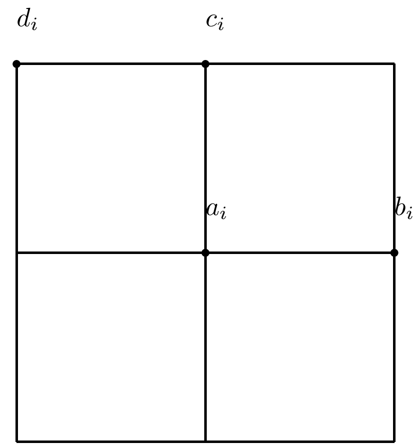

Figure 2.6. The difference stencil of L^T

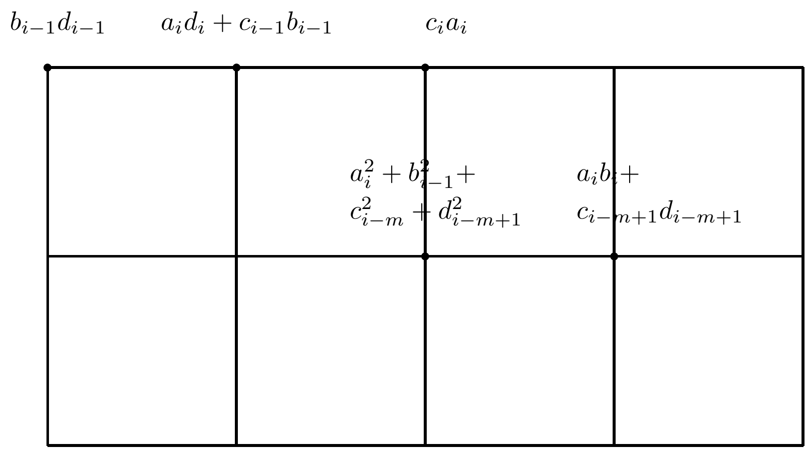

Figure 2.7. The difference stencil of LL^T

According to [12] again, from Figures 2.1 and 2.6-2.8 we know that the entries of the matrices L and $\widehat{R}$ satisfy the following recursive formulas:

$$\left\{\begin{array}{lcl} a_i^2 & = & \alpha_i(1+\delta) - r_i - r_{i-m+2} - b_{i-1}^2 - c_{i-m}^2 - d_{i-m+1}^2, \\ b_i & = & -\frac{\beta_i + c_{i-m+1}d_{i-m+1}}{a_i}, \\ c_i & = & -\frac{\gamma_i}{a_i}, \\ d_i & = & -\frac{b_{i-1}c_{i-1}}{a_i}, \\ r_i & = & b_{i-1}d_{i-1}, \\ \delta & = & \psi h^2, \quad \psi > 0, \end{array}\right. \tag{2.6}$$

where entries not defined should be replaced by zeros.

Continuing in this way we first come to the MIC(2) factorization, and then to the MIC(4) factorization, and so on.

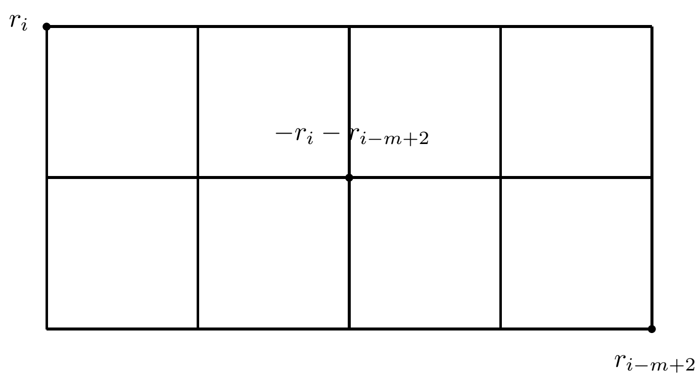

Figure 2.8. The difference stencil of $\widehat{R}$

2.1.3. The general MIC formula

For more general structured problems, the idea to obtain an MIC factorization is to let L have nonzero entries in the same positions as the matrix A, form the product LL^T to see where $\widehat{R}$ has nonzero entries, and extend L to have nonzero entries in these positions to get a more accurate factorization, and possibly continue in this manner for a few steps more.

2.2. The SMW inversions

We now turn to discuss how to efficiently invert the preconditioning matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ defined in (2.4) by making use of its structure. One basic tool is the Sherman-Morrison-Woodbury formula which expresses the inverse of a rank-k update of a matrix $A \in \mathbb{R}^{n\times n}$ into the inverse of the matrix A itself.

Lemma 2.1. (Sherman-Morrison-Woodbury formula (SMW-formula) [9]). *Let $A \in \mathbb{R}^{n\times n}$, and $U, V \in \mathbb{R}^{n\times k}$ be matrices such that both A and $(I + V^T A^{-1} U)$ are nonsingular. Then $A + UV^T$ is nonsingular and it holds that*

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}.$$

In particular, when $k = 1$, i.e., $U = u \in \mathbb{R}^n$ and $V = v \in \mathbb{R}^n$ are two vectors, and $1 + v^T A^{-1} u \neq 0$, the Sherman-Morrison-Woodbury formula reduces to the so-called *Sherman-Morrison* (SM) formula:

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}.$$

2.2.1. The SMW version

Let

$$\left\{ \begin{array}{lcl} U &=& (u_1, u_2, \ldots, u_N) \in \mathbb{R}^{n\times N}, \\ \Sigma &=& -\mathrm{diag}(\sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(N)}) \in \mathbb{R}^{N\times N}, \end{array} \right. \tag{2.7}$$

and $V = U\Sigma^{\frac{1}{2}}$, here we have applied the fact that $\sigma^{(i)} = -a_{i-\frac{1}{2},i} < 0$. Then according to (2.3) and (2.4) we can rewrite the matrices $\mathbf{A}$ and $\mathbf{M}$ as follows:

$$\left\{ \begin{array}{lcl} \mathbf{A} &=& A - U\Sigma U^T = A - VV^T, \\ \mathbf{M} &=& M - U\Sigma U^T = M - VV^T, \end{array} \right. \tag{2.8}$$

where $M = LL^T$.

For a known residual vector $r \in \mathbb{R}^n$, to compute the generalized residual vector $z = \mathbf{M}^{-1}r$ at each of the PCG iteration steps, by straightforwardly applying the SMW formula in Lemma 2.1 to the matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ we obtain

$$\begin{aligned} z &= (M - VV^T)^{-1}r \\ &= \left(M^{-1} + M^{-1}V(I - V^TM^{-1}V)^{-1}V^TM^{-1}\right)r \\ &= (I + ZW)y, \end{aligned}$$

where we have assumed that the matrix $I - V^TM^{-1}V \in \mathbb{R}^{N\times N}$ is nonsingular, and used the notations

$$y = M^{-1}r, \quad W = (I - V^TM^{-1}V)^{-1}V^T, \quad Z = M^{-1}V. \tag{2.9}$$

Denote $G = L^{-1}V \in \mathbb{R}^{n\times N}$, i.e., the matrix G satisfies the linear system $LG = V$. Then $I - V^TM^{-1}V$ is nonsingular if and only if $I - G^TG$ is nonsingular. Moreover, from (2.9) we equivalently have

$$y = M^{-1}r, \quad W = (I - G^TG)^{-1}V^T, \quad Z = L^{-T}G.$$

In addition, if we introduce the vector $s = L^Ty$, then computing $y = M^{-1}r$ is equivalent to solving the triangular sub-systems of linear equations

$$Ls = r \quad \text{and} \quad L^Ty = s.$$

In summary, we obtain the following SMW version for computing the generalized residual vector $z = \mathbf{M}^{-1}r$.

The SMW version:

1. *Solve s from $Ls = r$;*
2. *Solve y from $L^Ty = s$;*
3. *Solve G from $LG = V$, where $V = U\Sigma^{\frac{1}{2}}$;*
4. *Compute t by $t = V^Ty$ (or $t = G^Ts$);*
5. *Solve u from $(I - G^TG)u = t$;*
6. *Compute v by $v = Gu$;*
7. *Solve w from $L^Tw = v$;*
8. *Compute z by $z = y + w$.*

2.2.2. The recursive SM version

The inverse of the matrix $\mathbf{M}$ in (2.4) can also be computed recursively by applying the Sherman-Morrison formula a number of N steps. This results in another method for computing the generalized residual vector $z = \mathbf{M}^{-1}r$ at each of the PCG iteration steps, where $r \in \mathbb{R}^n$ is a known residual vector.

In fact, by letting $M_0 = M \equiv LL^T$, and for $i = 1, 2, \ldots, N$,

$$M_i = M_{i-1} + \sigma^{(i)}u_iu_i^T, \tag{2.10}$$

we have $\mathbf{M} = M_N$. When $\sigma^{(i)}u_i^TM_{i-1}^{-1}u_i + 1 \neq 0 (i = 1, 2, \ldots, N)$, after application of the SM formula we get

$$M_i^{-1} = M_{i-1}^{-1} - \frac{\sigma^{(i)}}{1 + \sigma^{(i)}u_i^TM_{i-1}^{-1}u_i} \cdot M_{i-1}^{-1}u_iu_i^TM_{i-1}^{-1}, \qquad i = 1, 2, \ldots, N. \tag{2.11}$$

For $i = 1, 2, \ldots, N$, define vectors

$$z_{i-1} = M_{i-1}^{-1} r, \quad p_{i-1} = M_{i-1}^{-1} u_i,$$

and scalars

$$\beta^{(i-1)} = p_{i-1}^T r = z_{i-1}^T u_i, \quad \delta^{(k-1,i)} = p_{k-1}^T u_i \ (k = 1, 2, \ldots, i), \quad \gamma^{(i-1)} = p_{i-1}^T u_{i+1}.$$

Then it follows from (2.11) that

$$z_i = z_{i-1} - \frac{\sigma^{(i)} \beta^{(i-1)}}{1 + \sigma^{(i)} \delta^{(i-1,i)}} \cdot p_{i-1}, \qquad i = 1, 2, \ldots, N.$$

In addition, by introducing vector $w_i = M_i^{-1} u_{i+2}$, we can obtain

$$\begin{aligned} w_i &= M_i^{-1} u_{i+2} \\ &= \left(M_{i-1}^{-1} - \tfrac{\sigma^{(i)}}{1+\sigma^{(i)}\delta^{(i-1,i)}} p_{i-1} p_{i-1}^T \right) u_{i+2} \\ &= \ldots \\ &= \left(M_0^{-1} - \textstyle\sum_{k=1}^{i} \tfrac{\sigma^{(k)}}{1+\sigma^{(k)}\delta^{(k-1,k)}} p_{k-1} p_{k-1}^T \right) u_{i+2} \end{aligned}$$

and

$$\begin{aligned} p_i &= M_i^{-1} u_{i+1} \\ &= \left(M_{i-1}^{-1} - \tfrac{\sigma^{(i)}}{1+\sigma^{(i)}\delta^{(i-1,i)}} \cdot p_{i-1} p_{i-1}^T \right) u_{i+1} \\ &= w_{i-1} - \tfrac{\sigma^{(i)}\gamma^{(i-1)}}{1+\sigma^{(i)}\delta^{(i-1,i)}} \cdot p_{i-1}. \end{aligned}$$

In summary, we obtain the following *recursive SM* (RSM) version for computing the generalized residual vector $z = \mathbf{M}^{-1} r$.

The RSM version:

1. *Initialization.*

   1.1 Solve p_0 *and* z_0 *from* $LL^T p_0 = u_1$ *and* $LL^T z_0 = r$*;*

   1.2 Compute $\beta^{(0)} = p_0^T r$, $\delta^{(0,1)} = p_0^T u_1$, *and* $\omega^{(0)} = \frac{\sigma^{(1)}}{1+\sigma^{(1)}\delta^{(0,1)}}$*;*

   1.3 Compute $z_1 = z_0 - \omega^{(0)} \beta^{(0)} p_0$.

2. *Recursion. For* $i = 2, 3, \ldots, N$*:*

   2.1 If $i = 2$ *then solve* w_0 *from* $LL^T w_0 = u_2$*, else*

   2.1.1 Solve q *from* $LL^T q = u_i$*;*

   2.1.2 Compute $w_{i-2} = q - \sum_{k=1}^{i-2} \omega^{(k-1)} \delta^{(k-1,i)} p_{k-1}$*;*

   2.2 Compute $\gamma^{(i-2)} = p_{i-2}^T u_i$*;*

   2.3 Compute $p_{i-1} = w_{i-2} - \omega^{(i-2)} \gamma^{(i-2)} p_{i-2}$*;*

   2.4 Compute $\beta^{(i-1)} = p_{i-1}^T r$*;*

   2.5 Compute $\delta^{(i-1,i)} = p_{i-1}^T u_i$*;*

   2.6 Compute $\omega^{(i-1)} = \frac{\sigma^{(i)}}{1+\sigma^{(i)}\delta^{(i-1,i)}}$*;*

   2.7 Compute $z_i = z_{i-1} - \omega^{(i-1)} \beta^{(i-1)} p_{i-1}$*;*

3. *Output.* $z = z_N$.

We remark that caution must be exercised in using this RSM version, however, because in general there is no guarantee of numerical stability through successive updating formulas (2.11) as the matrix changes. This phenomenon is confirmed by numerical results in Section 5.

3. Several Preparative Lemmas

To prove the positive definiteness and analyze the preconditioning properties of the new preconditioners, in this section we establish several necessary lemmas.

Lemma 3.1. *[14] Let $A \in \mathbb{R}^{n\times n}$ be a symmetric M-matrix, and $\mathcal{P} \subseteq \{(i,j) \mid i \neq j, 1 \leq i,j \leq n\}$ the off-diagonal indices with the property that $(i,j) \in \mathcal{P}$ implies $(j,i) \in \mathcal{P}$. Then there exist unique lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with $l_{i,j} = 0$ for $(i,j) \in \mathcal{P}$ and zero-diagonal matrix $R \in \mathbb{R}^{n\times n}$ with $r_{i,j} = 0$ for $(i,j) \notin \mathcal{P}$ such that $A = LL^T - R$. Moreover, this splitting is a regular splitting.*

This lemma describes the existence of the MIC(0) and the MIC(1) factorizations of the matrix $A \in \mathbb{R}^{n\times n}$ defined in (2.2). The following lemma presents a sufficient condition for examining that the condition number of the preconditioned matrix is of order $\mathcal{O}(h^{-1})$.

Lemma 3.2. *[12] Let $\widetilde{A}, \widetilde{M} \in \mathbb{R}^{n\times n}$ be two symmetric positive definite matrices, and $\widetilde{M} = \widetilde{A} + \widetilde{R} = \widetilde{A} + \widetilde{D} + \widehat{R}$. Assume that $\widehat{R}$ is negative semidefinite having zero row-sums and only local couplings, and $\widetilde{D}$ is positive diagonal with diagonal elements of size $\mathcal{O}(h^2)$. Then a sufficient condition to obtain $\kappa(\widetilde{M}^{-1}\widetilde{A}) = \mathcal{O}(h^{-1})$ is:*

$$0 \leq -(\widehat{R}x, x) \leq \frac{1}{1+\tau h}(\widetilde{A}x, x), \qquad \forall x \in \mathbb{R}^n,$$

where τ is a positive constant independent of the mesh size h.

Lemma 3.3. *[12] Let Θ, Ξ and Υ be reals, and ζ, χ be positive reals. Then*

$$\frac{(\Theta - \Xi)^2}{\zeta + \chi} \leq \frac{(\Theta - \Upsilon)^2}{\zeta} + \frac{(\Upsilon - \Xi)^2}{\chi}.$$

The matrix $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ defined in (2.2) is a "local" matrix so that the distance between two points in the mesh representing indices i and j is of order $\mathcal{O}(h)$ for $a_{i,j} \neq 0$, and the number of indices j such that $a_{i,j} \neq 0$ is of order $\mathcal{O}(1)$ for each i. Without loss of generality, we may assume that the elements $a_{i,j}$ are normalized to be of order $\mathcal{O}(1)$.

Because the matrices A, L and $\widehat{R}$ defined by (2.5) and (2.6) (see also Figures 2.1-2.8) are associated with the matrix $\mathbf{A}$ (see (1.4)) which comes from the five-point central difference discretization of the partial differential equation (1.1)-(1.2), it is reasonable for us to assume that

$$\alpha_i \geq \alpha, \quad 0 < \varsigma \leq \beta_i \leq \beta, \quad 0 < \varsigma \leq \gamma_i \leq \gamma, \qquad i = 1, 2, \ldots, n \tag{3.1}$$

and

$$2(\beta + \gamma) \leq \alpha. \tag{3.2}$$

This readily implies that the matrices A and $\mathbf{A}$ are both symmetric positive definite Z-matrices, and hence, they are also M-matrices.

Lemma 3.4. *Let r_i be defined by the MIC(0) formula (2.5). Then it holds that*

$$r_i \leq \frac{\alpha}{8} \cdot \frac{1}{1+\tau h},$$

where $\tau = 2\sqrt{\frac{2\psi}{\alpha}}$ is a positive constant independent of the mesh size h.

Proof. We first demonstrate the following universal bound for a_i:

$$a_i^2 \geq \frac{\alpha}{2}(1+\tau h), \qquad i = 1, 2, \ldots, n, \tag{3.3}$$

where τ is a positive constant independent of the mesh size h.

When $\psi = 0$, we can straightforwardly derive the estimates

$$a_i^2 \geq \frac{\alpha}{2}, \qquad i = 1, 2, \ldots, n,$$

from the recursive formula (2.5) and by induction on i. In fact, it is obvious that

$$a_1^2 = \alpha_1 \geq \alpha > \frac{\alpha}{2}.$$

In general, if we assume that

$$a_i^2 \geq \frac{\alpha}{2}, \quad i = 1, 2, \ldots, p-1,$$

then

$$\begin{array}{rcl} a_p^2 & = & \alpha_p - b_{p-1}(c_{p-1} + b_{p-1}) - c_{p-m}(b_{p-m} + c_{p-m}) \\ & \geq & \alpha - \frac{\beta(\beta+\gamma)}{a_{p-1}^2} - \frac{\gamma(\beta+\gamma)}{a_{p-m}^2} \\ & \geq & \alpha - \frac{(\beta+\gamma)^2}{\alpha/2} \\ & \geq & \alpha - \frac{\alpha}{2} \\ & = & \frac{\alpha}{2}. \end{array}$$

By induction, we obtain

$$a_i^2 \geq \frac{\alpha}{2}, \qquad i = 1, 2, \ldots, n.$$

In addition, we notice that for a sufficiently large N, the elements a_i approaches a constant a that satisfies

$$\varphi(a) \equiv a^2 + \frac{\alpha^2}{4a^2} - \alpha = 0.$$

Analogously, when $\psi > 0$, we can derive the estimates (3.3) by solving the one-variable quadratic equation

$$\varphi(a) - 4\psi h^2 = 0.$$

This immediately gives

$$a^2 - \frac{\alpha}{2} = 2\sqrt{\psi} h \cdot a.$$

After simple computations, we have

$$a = \sqrt{\psi} h + \sqrt{\psi h^2 + \frac{\alpha}{2}}$$

and

$$a^2 = 2\psi h^2 + \frac{\alpha}{2} + 2\sqrt{\psi} h \sqrt{\psi h^2 + \frac{\alpha}{2}} \geq \frac{\alpha}{2}(1+\tau h),$$

where $\tau = 2\sqrt{\frac{2\psi}{\alpha}}$. This shows the validity of (3.3).

It follows straightforwardly from

$$r_i = b_{i-1} c_{i-1}, \quad i = 1, 2, \ldots, n$$

that

$$r_i = \frac{\beta_{i-1}\gamma_{i-1}}{a_{i-1}^2} \leq \frac{2\beta\gamma}{\alpha(1+\tau h)} \leq \frac{(\beta+\gamma)^2}{2\alpha(1+\tau h)} \leq \frac{\alpha}{8} \cdot \frac{1}{1+\tau h}, \qquad i = 1, 2, \ldots, n.$$

Lemma 3.5. *Let r_i be defined by the MIC(1) formula (2.6). Then it holds that*

$$r_i \leq \frac{\beta\gamma}{\beta+4\gamma} \cdot \frac{1}{1+\tau h},$$

where $\tau = 2\sqrt{\frac{\psi}{\beta+4\gamma}}$ is a positive constant independent of the mesh size h.

Proof. For $\psi = 0$, we assert that

$$a_i^2 \geq \frac{\beta + 2\gamma + \sqrt{\beta(\beta+4\gamma)}}{2}, \quad i = 1, 2, \ldots, n \tag{3.4}$$

and

$$r_i \leq \frac{\beta\gamma}{\beta+4\gamma}, \quad i = 1, 2, \ldots, n. \tag{3.5}$$

In fact, from the MIC(1) formula (2.6) we see that when $\psi = 0$, a bound for a_i can be obtained by solving the one-variable nonlinear equation $\varphi(a) = 0$, where

$$\varphi(a) = \left(a - \frac{\gamma}{a}\right)^2 + \frac{\beta^2}{\left(a - \frac{\gamma}{a}\right)^2} - 2\beta. \tag{3.6}$$

Because

$$\begin{array}{rcl} (-b_i)a_i & = & \beta_i + c_{i-m+1}d_{i-m+1} \\ & = & \beta_i - \frac{c_{i-m+1}b_{i-m}c_{i-m}}{a_{i-m+1}} \\ & = & \beta_i - \frac{\gamma_{i-m+1}\gamma_{i-m}b_{i-m}}{a_{i-m+1}^2 a_{i-m}} \\ & \leq & \beta + \frac{\gamma^2(-b_{i-m})}{a_{i-m+1}^2 a_{i-m}}, \end{array}$$

we have

$$(-b) \cdot a \leq \beta + \frac{\gamma^2(-b)}{a^3},$$

where $-b$ is an upper bound of $-b_i$. By solving the equation

$$-ba + \frac{\gamma^2 b}{a^3} - \beta = 0,$$

we obtain

$$-b \leq \frac{\beta a^3}{a^4 - \gamma^2}$$

and

$$r_i = b_{i-1}d_{i-1} = -\frac{b_{i-1}b_{i-2}c_{i-2}}{a_{i-1}} = \frac{b_{i-1}b_{i-2}\gamma_{i-2}}{a_{i-2}a_{i-1}} \leq \frac{\gamma b^2}{a^2}.$$

Therefore,

$$a^2 \geq \alpha_i - \frac{\beta^2 a^6}{(a^4-\gamma^2)^2} - \frac{\gamma^2}{a^2} - \frac{\gamma^2\beta^2 a^2}{(a^4-\gamma^2)^2} - \frac{2\gamma\beta^2 a^4}{(a^4-\gamma^2)^2}.$$

As

$$\frac{\beta^2 a^2}{(\gamma - a^2)^2} = \frac{\beta^2 a^6}{(a^4-\gamma^2)^2} + \frac{\gamma^2\beta^2 a^2}{(a^4-\gamma^2)^2} + \frac{2\gamma\beta^2 a^4}{(a^4-\gamma^2)^2},$$

it follows that

$$a^2 + \frac{\gamma^2}{a^2} + \frac{\beta^2 a^2}{(\gamma - a^2)^2} \geq \alpha_i.$$

Hence,

$$\left(a-\frac{\gamma}{a}\right)^2+\frac{\beta^2}{\left(a-\frac{\gamma}{a}\right)^2}\geq \alpha_i-2\gamma,$$

which directly results in the estimate

$$\varphi(a)\equiv\left(a-\frac{\gamma}{a}-\frac{\beta}{a-\frac{\gamma}{a}}\right)^2\geq \alpha_i-2(\beta+\gamma)\geq 0.$$

This shows the validity of (3.6).

From (3.6) we immediately get

$$\left\{\begin{array}{lcl} a &=& \frac{\sqrt{\beta}+\sqrt{\beta+4\gamma}}{2},\\ a^2 &=& \frac{\beta+2\gamma+\sqrt{\beta(\beta+4\gamma)}}{2},\\ a^4 &=& \frac{\beta(\beta+4\gamma)+(\beta+2\gamma)\sqrt{\beta(\beta+4\gamma)}}{2}+\gamma^2, \end{array}\right. \tag{3.7}$$

and therefore, the estimate (3.4).

By (3.7) we can obtain

$$\begin{array}{lcl} |\frac{b}{a}| &\leq& \frac{\beta a^2}{a^4-\gamma^2}\\ &=& \frac{\beta(\beta+2\gamma+\sqrt{\beta(\beta+4\gamma)})}{\beta(\beta+4\gamma)+(\beta+2\gamma)\sqrt{\beta(\beta+4\gamma)}}\\ &=& \sqrt{\frac{\beta}{\beta+4\gamma}}. \end{array}$$

Therefore,

$$r_i\leq\gamma\left(\frac{b}{a}\right)^2\leq\frac{\beta\gamma}{\beta+4\gamma},\quad i=1,2,\ldots,n.$$

This demonstrates the validity of (3.5).

For $\psi>0$, we assert the following universal bounds for a_i and r_i:

$$\left\{\begin{array}{lcl} a_i^2 &\geq& \frac{\beta+2\gamma+\sqrt{\beta(\beta+4\gamma)}}{2}\cdot(1+\tau h),\\ r_i &\leq& \frac{\beta\gamma}{\beta+4\gamma}\cdot\frac{1}{1+\tau h}, \end{array}\right. \tag{3.8}$$

where $\tau=2\sqrt{\frac{\psi}{\beta+4\gamma}}$ is a positive constant independent of the mesh size h.

In fact, analogously to the case that $\psi=0$, we can obtain the bound for a_i by solving the one-variable nonlinear equation $\varphi(a)=4\psi h^2$, resulting in

$$a=\frac{\sqrt{\psi}h+\sqrt{\psi h^2+\beta}+\sqrt{2\psi h^2+\beta+2\sqrt{\psi}h\sqrt{\psi h^2+\beta}+4\gamma}}{2}.$$

It immediately follows that

$$\begin{array}{lcl} 2a^2 &=& 2\psi h^2+\beta+2\sqrt{\psi}h\sqrt{\psi h^2+\beta}+2\gamma\\ && +(\sqrt{\psi}h+\sqrt{\psi h^2+\beta})\sqrt{2\psi h^2+\beta+2\sqrt{\psi}h\sqrt{\psi h^2+\beta}+4\gamma}. \end{array} \tag{3.9}$$

Define a nonlinear function

$$f(t)=\left(t+\sqrt{t^2+\beta}\right)\sqrt{2t^2+\beta+2t\sqrt{t^2+\beta}+4\gamma}.$$

Then we have

$$\begin{array}{lcl} f^2(t) &=& 8t^4+8(\beta+\gamma)t^2+\beta(\beta+4\gamma)+4t(2t^2+\beta+2\gamma)\sqrt{t^2+\beta}\\ &\geq& 4(\beta+\gamma)t^2+4(\beta+2\gamma)\sqrt{\beta}t+\beta(\beta+4\gamma). \end{array}$$

Since

$$(\beta+4\gamma)f^2(t) \geq 4(\beta+2\gamma)^2t^2 + 4(\beta+2\gamma)(\beta+4\gamma)\sqrt{\beta}t + \beta(\beta+4\gamma)^2,$$

we get

$$(\beta+4\gamma)f^2(t) \geq [2(\beta+2\gamma)t + \sqrt{\beta}(\beta+4\gamma)]^2.$$

Therefore,

$$\begin{array}{rcl} f(t) & \geq & \frac{2(\beta+2\gamma)t+\sqrt{\beta}(\beta+4\gamma)}{\sqrt{\beta+4\gamma}} \\ & \geq & \left(\sqrt{\beta+4\gamma} + \frac{\beta}{\sqrt{\beta+4\gamma}}\right)t + \sqrt{\beta(\beta+4\gamma)}. \end{array}$$

Now, letting $t = \sqrt{\psi}h$, we obtain

$$\begin{array}{l} (\sqrt{\psi}h + \sqrt{\psi h^2+\beta})\sqrt{2\psi h^2 + \beta + 2\sqrt{\psi}h\sqrt{\psi h^2+\beta} + 4\gamma} \\ \quad \geq \left(\sqrt{\beta+4\gamma} + \frac{\beta}{\sqrt{\beta+4\gamma}}\right)\sqrt{\psi}h + \sqrt{\beta(\beta+4\gamma)}. \end{array} \tag{3.10}$$

It follows from (3.9) and (3.10) that

$$\begin{array}{rcl} 2a^2 & \geq & \beta + 2\gamma + 2\sqrt{\psi}h\cdot\sqrt{\beta} + \left(\sqrt{\beta+4\gamma} + \frac{\beta}{\sqrt{\beta+4\gamma}}\right)\sqrt{\psi}h + \sqrt{\beta(\beta+4\gamma)} \\ & = & (\beta+2\gamma+\sqrt{\beta(\beta+4\gamma)})(1+\tau h), \end{array}$$

where $\tau = 2\sqrt{\frac{\psi}{\beta+4\gamma}}$. This demonstrates the validity of the lower bounds for a_i in (3.8).

Noticing that

$$\frac{\beta+2\gamma+\sqrt{\beta(\beta+4\gamma)}}{2} = \left(\frac{\sqrt{\beta}+\sqrt{\beta+4\gamma}}{2}\right)^2,$$

by making use of (3.5) we obtain

$$r_i \leq \frac{\beta\gamma}{\beta+4\gamma}\cdot\frac{1}{1+\tau h}, \qquad i = 1,2,\ldots,n,$$

where $\tau = 2\sqrt{\frac{\psi}{\beta+4\gamma}}$. This demonstrates the validity of the upper bounds for r_i in (3.8).

4. Conditioning

In this section, we will first demonstrate the well-definiteness of the preconditioner $\mathbf{M}$, and then estimate the condition number of the preconditioned matrix $\mathbf{M}^{-1}\mathbf{A}$. To this end, we further assume that the bounds in (3.1)-(3.2) satisfy

$$5\beta\gamma \leq (\beta+4\gamma)\varsigma, \quad \alpha \leq 4\varsigma. \tag{4.1}$$

Evidently, for the model problem that $a(\xi,\eta) = 1$ and $\theta(\xi,\eta) = 0$ in (1.1), the assumptions (3.1), (3.2) and (4.1) are automatically satisfied since when we have $\alpha = 4$ and $\beta = \gamma = \varsigma = 1$.

Theorem 4.1. *Let $A \in \mathbb{R}^{n\times n}$ be the matrix defined by (2.2) (see also Figure 2.1), and $M = LL^T$ be its MIC factorization such that $A = M - R$, where $R = D + \widehat{R}$, $D = \psi h^2\cdot\mathrm{diag}(A)$ and $\widehat{R}$ is a negative semidefinite matrix of zero row-sums.*

(i) If L and $\widehat{R}$ are defined by the MIC(0) formula (2.5) (see also Figures 2.2-2.5), then

$$\begin{array}{rcl} 0 \leq -(\widehat{R}x,x) & \leq & \frac{\alpha}{4\varsigma}\cdot\frac{1}{1+\tau h}\cdot\min\{(Ax,x),(\mathbf{A}x,x)\} \\ & \leq & \frac{1}{1+\tau h}\cdot\min\{(Ax,x),(\mathbf{A}x,x)\}, \end{array}$$

where $\tau = 2\sqrt{\frac{2\psi}{\alpha}}$ is a positive constant independent of the mesh size h. Moreover, by Lemma 3.2, it holds that $\kappa(M^{-1}A) = \mathcal{O}(h^{-1})$;

(ii) If L and $\widehat{R}$ are defined by the MIC(1) formula (2.6) (see also Figures 2.6-2.8), then

$$\begin{array}{rcl} 0 \leq -(\widehat{R}x, x) & \leq & \frac{5\beta\gamma}{(\beta+4\gamma)\varsigma} \cdot \frac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\} \\ & \leq & \frac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\}, \end{array}$$

where $\tau = 2\sqrt{\frac{\psi}{\beta+4\gamma}}$ is a positive constant independent of the mesh size h. Moreover, by Lemma 3.2, it holds that $\kappa(M^{-1}A) = \mathcal{O}(h^{-1})$.

Proof. Let $A = (a_{i,j})$ and $x = (x_1, x_2, \ldots, x_n)^T$. Then by an elementary summation by parts we obtain

$$(Ax, x) = -\sum_i \sum_{j>i} a_{i,j}(x_i - x_j)^2 + \sum_i \sum_j a_{i,j} x_i^2. \tag{4.2}$$

In particular, by considering the structures of the matrices A (see Figure 2.1 or (2.2)) and $\mathbf{A}$ (see (1.4)) we see that (4.2) leads to

$$\begin{array}{rcl} (Ax, x) & \geq & \sum_i [\beta_i (x_i - x_{i+1})^2 + \gamma_i (x_i - x_{i+m})^2] \\ & \geq & \varsigma \sum_i [(x_i - x_{i+1})^2 + (x_i - x_{i+m})^2] \end{array} \tag{4.3}$$

and

$$\begin{array}{rcl} (\mathbf{A}x, x) & = & -\sum_i \sum_{j>i} a_{i,j}(x_i - x_j)^2 + \sum_i \sum_j a_{i,j} x_i^2 \\ & & -\sum_{i=1}^{m-1} a_{(i-1)m+1,im}(x_{(i-1)m+1} + x_{im})^2 \\ & \geq & -\sum_i \sum_{j>i} a_{i,j}(x_i - x_j)^2 + \sum_i \sum_j a_{i,j} x_i^2 \\ & & -2\sum_{i=1}^{m-1} a_{(i-1)m+1,im}(x_{(i-1)m+1}^2 + x_{im}^2) \\ & \geq & \sum_i [\beta_i (x_i - x_{i+1})^2 + \gamma_i (x_i - x_{i+m})^2] \\ & \geq & \varsigma \sum_i [(x_i - x_{i+1})^2 + (x_i - x_{i+m})^2]. \end{array} \tag{4.4}$$

We first demonstrate (i). From (2.5) (see also Figure 2.3) we have

$$-(\widehat{R}x, x) = \sum_{r_i \neq 0} r_i (x_i - x_{i+m-1})^2.$$

By applying Lemma 3.4 we get

$$-(\widehat{R}x, x) \leq \frac{\alpha}{8} \cdot \frac{1}{1+\tau h} \cdot \sum_{r_i \neq 0} (x_i - x_{i+m-1})^2.$$

It then follows from Lemma 3.3 with $\zeta = \chi = 1$ that

$$\begin{array}{rcl} -(\widehat{R}x, x) & \leq & \frac{\alpha}{4(1+\tau h)} \sum_{r_i \neq 0} [(x_i - x_{i-1})^2 + (x_{i-1} - x_{i+m-1})^2] \\ & = & \frac{\alpha}{4(1+\tau h)} \sum_{r_{i+1} \neq 0} [(x_{i+1} - x_i)^2 + (x_i - x_{i+m})^2]. \end{array} \tag{4.5}$$

Because $r_{i+1} = \frac{\beta_i \gamma_i}{a_i^2}$, we know that $r_{i+1} \neq 0$ if and only if $\beta_i \gamma_i \neq 0$. Therefore, by combining (4.3)-(4.5) we obtain

$$-(\widehat{R}x, x) \leq \frac{\alpha}{4\varsigma} \cdot \frac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\} \leq \frac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\}.$$

According to Lemma 3.2, the conclusion (i) is true.

We now turn to prove (ii). From (2.6) (see also Figure 2.8) we have

$$-(\widehat{R}x, x) = \sum_{r_i \neq 0} r_i (x_i - x_{i+m-2})^2.$$

By applying Lemma 3.5, and Lemma 3.3 twice (first with $\zeta = 2$, $\chi = 3$ and then with $\zeta = 1$ and $\chi = 2$), we obtain

$$
\begin{aligned}
-(\widehat{R}x, x) &\le \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \sum_{r_i \neq 0} \tfrac{1}{5}(x_i - x_{i+m-2})^2 \\
&\le \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \left(\sum_{r_{i+1} \neq 0} \left[\tfrac{1}{2}(x_{i+1} - x_i)^2 + \tfrac{1}{3}(x_i - x_{i+m-1})^2 \right] \right) \\
&\le \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \left(\tfrac{1}{2} \sum_{r_{i+1} \neq 0} (x_{i+1} - x_i)^2 \right. \\
&\quad \left. + \tfrac{1}{3} \left[\sum_{r_{i-m+1} \neq 0} \tfrac{1}{2}(x_{i-m} - x_{i-1})^2 + \sum_{r_{i+1} \neq 0} \tfrac{1}{2}(x_i - x_{i+m-1})^2 \right] \right) \\
&\le \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \left(\tfrac{1}{2} \sum_{r_{i+1} \neq 0} (x_{i+1} - x_i)^2 \right. \\
&\quad + \sum_{r_{i-m+1} \neq 0} \tfrac{1}{2} \left[\tfrac{1}{2}(x_i - x_{i-1})^2 + (x_i - x_{i-m})^2 \right] \\
&\quad \left. + \sum_{r_{i+1} \neq 0} \tfrac{1}{2} \left[(x_i - x_{i+m})^2 + \tfrac{1}{2}(x_{i+m} - x_{i+m-1})^2 \right] \right) \\
&= \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \left(\tfrac{1}{2} \sum_{r_{i+1} \neq 0} (x_{i+1} - x_i)^2 \right. \\
&\quad + \sum_{r_{i-m+1} \neq 0} \tfrac{1}{2} \left[(x_i - x_{i-1})^2 + (x_i - x_{i-m})^2 \right] \\
&\quad \left. + \sum_{r_{i+1} \neq 0} \tfrac{1}{2}(x_i - x_{i+m})^2 \right) \\
&\le \tfrac{5\beta\gamma}{(\beta+4\gamma)(1+\tau h)} \left(\sum_{r_{i+1} \neq 0} \left[\tfrac{1}{2}(x_{i+1} - x_i)^2 + \tfrac{1}{2}(x_i - x_{i+m})^2 \right] \right. \\
&\quad \left. + \sum_{r_{i-m+1} \neq 0} \left[\tfrac{1}{2}(x_i - x_{i-1})^2 + \tfrac{1}{2}(x_i - x_{i-m})^2 \right] \right).
\end{aligned}
\tag{4.6}
$$

From (4.3) and (4.4) we have

$$
\begin{aligned}
\min\{(Ax, x), (\mathbf{A}x, x)\} &\ge \sum_i \beta_{i-1}(x_i - x_{i-1})^2 + \sum_i \gamma_{i-m}(x_{i-m} - x_i)^2 \\
&\ge \varsigma \sum_i (x_i - x_{i-1})^2 + \varsigma \sum_i (x_{i-m} - x_i)^2,
\end{aligned}
\tag{4.7}
$$

where elements not defined should be replaced by zeros.

In addition, from the formulas (2.6) it is clear that $r_{i+1} = r_{i-m+1} = 0$ for such an i that $\beta_i = 0$, $\beta_{i-1} = 0$, $\gamma_i = 0$ and $\gamma_{i-m} = 0$.

By comparing (4.6) and (4.7) we obtain

$$
\begin{aligned}
-(\widehat{R}x, x) &\le \tfrac{5\beta\gamma}{\beta+4\gamma} \cdot \tfrac{1}{1+\tau h} \cdot \left[\tfrac{1}{2\varsigma} \min\{(Ax, x), (\mathbf{A}x, x)\} + \tfrac{1}{2\varsigma} \min\{(Ax, x), (\mathbf{A}x, x)\} \right] \\
&= \tfrac{5\beta\gamma}{(\beta+4\gamma)\varsigma} \cdot \tfrac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\} \\
&\le \tfrac{1}{1+\tau h} \cdot \min\{(Ax, x), (\mathbf{A}x, x)\}.
\end{aligned}
$$

According to Lemma 3.2 again, the conclusion (ii) is also true.

The following theorem describes the well-definiteness of the preconditioner $\mathbf{M}$.

Theorem 4.2. *Let $A \in \mathbb{R}^{n\times n}$ be the matrix defined in (2.2), $A = M - R$, with $M = LL^T$ and $R = D + \widehat{R}$, be the MIC(0) or the MIC(1) factorizations defined by (2.5) or (2.6), respectively, $\mathbf{A} \in \mathbb{R}^{n\times n}$ be the matrix defined in (1.4), $\mathbf{M} \in \mathbb{R}^{n\times n}$ be the preconditioner to $\mathbf{A}$ defined in (2.4), $U \in \mathbb{R}^{n\times N}$ and $\Sigma \in \mathbb{R}^{N\times N}$ be the matrices defined in (2.7) with $V = U\Sigma^{\frac{1}{2}}$, and $\{M_i\}_{i=1}^N$ be the matrix sequence defined in (2.10). Then for both MIC(0) and MIC(1), it holds that*

(i) *M is symmetric positive definite;*

(ii) *$\mathbf{M}$ is symmetric positive definite;*

(iii) *$I - V^T M^{-1} V$ is nonsingular;*

(iv) *$\sigma^{(i)} u_i^T M_{i-1}^{-1} u_i + 1 \neq 0$, $i = 1, 2, \ldots, N$.*

Proof. (i) is obviously true from Lemma 3.1.

From (2.3) and (2.4) (see also (2.8)), as well as the definitions of both MIC(0) and MIC(1) factorizations, we have

$$\mathbf{M} = M + (\mathbf{A} - A) = \mathbf{A} + R = \mathbf{A} + D + \widehat{R}. \tag{4.8}$$

According to Theorem 4.1 we get

$$0 \le -(\widehat{R}x, x) \le \frac{1}{1+\tau h}(\mathbf{A}x, x) < (\mathbf{A}x, x), \qquad \forall x \in \mathbb{R}^n \setminus \{0\}.$$

Hence, $\mathbf{A} + \widehat{R}$ is a symmetric positive definite matrix. Noticing that $D = \psi h^2 \cdot \mathrm{diag}(A)$ is positive diagonal, we therefore know that the matrix $\mathbf{M}$ is symmetric positive definite. This demonstrates the correctness of (ii).

To verify (iii), we recall from (2.8) that

$$\mathbf{M} = M - VV^T = LL^T - VV^T,$$

and thereby,

$$L^{-1}\mathbf{M}L^{-T} = I - (L^{-1}V)(L^{-1}V)^T.$$

Because $L^{-1}\mathbf{M}L^{-T}$ is nonsingular, we know that 1 is not an eigenvalue of the matrix $(L^{-1}V)$ $(L^{-1}V)^T$. It then follows that 1 is also not an eigenvalue of the matrix

$$(L^{-1}V)^T(L^{-1}V) = V^T M^{-1} V.$$

This immediately implies that the matrix $I - V^T M^{-1} V$ is nonsingular. Therefore, (iii) is also valid.

We now turn to (iv). Because M and $\mathbf{M}$ are both symmetric positive definite and

$$M \equiv M_0 \succeq M_1 \succeq \ldots \succeq M_{N-1} \succeq M_N \equiv \mathbf{M},$$

by (2.4) and (2.10) we can inductively verify the validity of (iv) in a similar fashion to (iii). Here, the ordering "$\succeq$" is defined according to the symmetric positive semidefiniteness, i.e., for two matrices $B, C \in \mathbb{R}^{n\times n}$, $B \succeq C$ if $B - C$ is symmetric positive semidefinite.

For the condition number of the preconditioned matrix $\mathbf{M}^{-1}\mathbf{A}$, we have the following estimate.

Theorem 4.3. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be the matrix defined in (1.4), and $\mathbf{M} \in \mathbb{R}^{n\times n}$ be the preconditioner defined in (2.4). Then for both MIC(0) and MIC(1), it holds that $\kappa(\mathbf{M}^{-1}\mathbf{A}) = \mathcal{O}(h^{-1})$.*

Proof. From (4.8) and Theorem 4.2 (ii) we see that

$$\mathbf{M} = \mathbf{A} + D + \widehat{R},$$

where $\mathbf{A}$ and $\mathbf{M}$ are symmetric positive definite matrices, $D = \psi h^2 \cdot \mathrm{diag}(A)$, and $\widehat{R}$ is the negative semidefinite matrix of zero row-sums. In addition, by Theorem 4.1 we know that

$$0 \le -(\widehat{R}x, x) \le \frac{1}{1+\tau h} \cdot (\mathbf{A}x, x), \qquad \forall x \in \mathbb{R}^n$$

holds for both MIC(0) and MIC(1) factorizations, where τ is a positive constant independent of the mesh size h. It then straightforwardly follows from Lemma 3.2 that $\kappa(\mathbf{M}^{-1}\mathbf{A}) = \mathcal{O}(h^{-1})$.

5. Numerical Examples

We use several numerical examples from different choices of the coefficient functions $a(\xi, \eta)$ and $\theta(\xi, \eta)$ in the two-dimensional second-order self-adjoint elliptic partial differential equation (1.1)-(1.2) to show feasibility, robustness and effectiveness of the new preconditioners.

In actual computations, all runs are started from the zero vector and terminated once the current residuals $r^{(k)} = \mathbf{b} - \mathbf{A}x^{(k)}$ satisfy $\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} \leq \varepsilon = 10^{-12}$, where $x^{(k)}$ is the current iteration. The reduction factor of an iteration is denoted by $\rho = \log\left(\frac{\|x^{(k)}-x^*\|_2}{\|x^{(0)}-x^*\|_2}\right)$, with x^* the exact solution of the system of linear equations (1.3).

The new PCG methods, RSMICCG(0), SMWICCG(0), RSMICCG(1) and SMWICCG(1), are compared with the known PCG methods, ICCG(0) and ICCG(1), as well as the CG method itself, respectively, for aspects of number of total iteration steps (denoted by "IT") and elapsed CPU time (denoted by "CPU"). In some tables, we use the symbol "–" to denote that the MIC factorization involved breaks down.

Example 5.1. The coefficient functions are

$$a(\xi,\eta) = \begin{cases} 1000, & 0 < \xi < 0.5, \quad 0 < \eta < 1, \\ 1, & 0.5 \leq \xi < 1, \quad 0 < \eta < 1, \end{cases} \qquad \text{and} \quad \theta(\xi,\eta) = 10.$$

For different discretization stepsizes, numerical results are listed in Table 5.1 and depicted in Figures 5.1a-5.1d.

Table 5.1. Iteration numbers and CPUs for Example 5.1

h^{-1}		16	32	40	48	64	80	100	128
RSMICCG(0)	IT	19	34	41	48	63	79	97	124
	CPU	0.01	0.07	0.14	0.27	0.70	1.46	3.13	6.96
SMWICCG(0)	IT	13	21	25	29	36	45	55	68
	CPU	0.01	0.04	0.07	0.15	0.36	0.76	1.65	3.61
ICCG(0)	IT	20	38	47	57	77	96	122	154
	CPU	0.01	0.07	0.15	0.25	0.69	1.40	3.08	6.41
RSMICCG(1)	IT	18	30	36	42	54	65	80	101
	CPU	0.01	0.07	0.14	0.26	0.64	1.40	2.83	6.46
SMWICCG(1)	IT	17	28	33	39	51	63	77	97
	CPU	0.00	0.05	0.10	0.20	0.52	1.14	2.33	5.22
ICCG(1)	IT	–	–	–	–	–	–	–	–
	CPU	–	–	–	–	–	–	–	–
CG	IT	43	151	200	268	414	576	819	1178
	CPU	0.01	0.14	0.29	0.54	1.63	3.97	9.64	26.55

Evidently, the iterations with preconditioners considerably outperform the CG iteration in iteration numbers and CPU times. For the preconditioned iterations based on MIC(0) factorization, we see that the SMWICCG(0) is the fastest one. According to h, the iteration numbers of the RSMICCG(0) are correspondingly smaller than those of the ICCG(0), and the CPU times of both methods are roughly the same. For the preconditioned iterations based on MIC(1) factorization, we see that the SMWICCG(1) is the fastest one. However, ICCG(1) fails to deliver an approximate solution to x^* due to break-down of the MIC(1) factorization. In addition, RSMICCG(1) is faster than RSMICCG(0) in all cases, and SMWICCG(1) is slower than SMWICCG(0) in most cases.

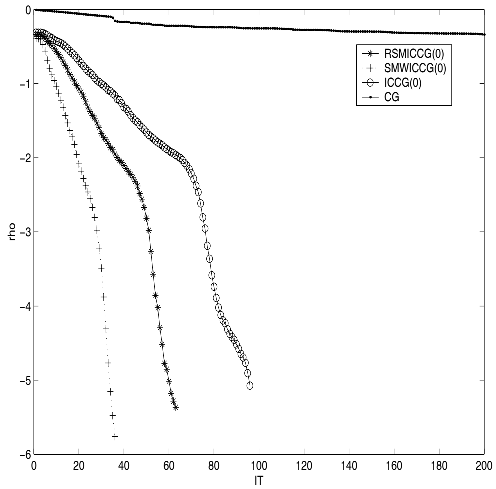

Figure 5.1a: Curves of ρ versus IT for Example 5.1 when $h^{-1} = 64$. The preconditioner uses MIC(0).

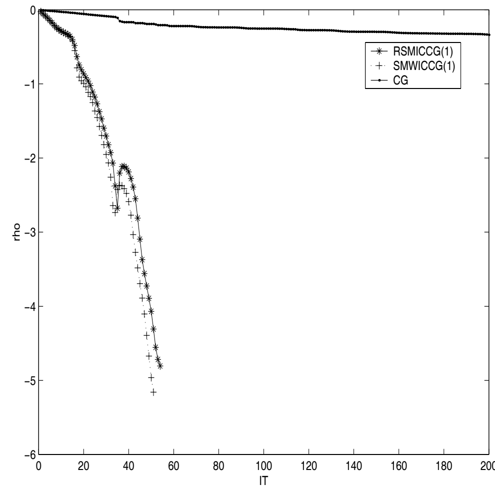

Figure 5.1b: Curves of ρ versus IT for Example 5.1 when $h^{-1} = 64$. The preconditioner uses MIC(1).

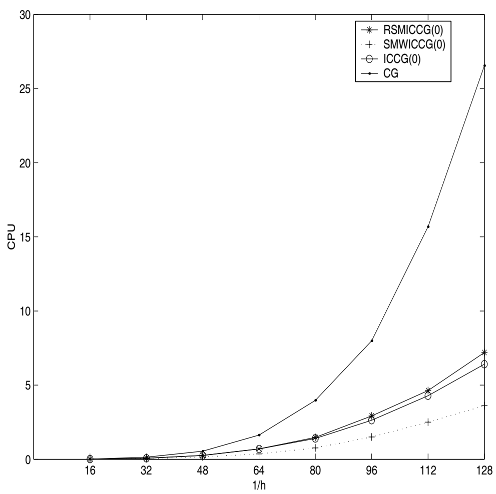

Figure 5.1c: Curves of CPU versus h^{-1} for Example 5.1. The preconditioner uses MIC(0).

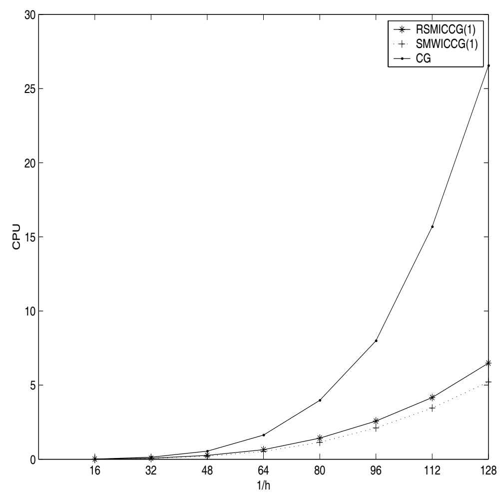

Figure 5.1d: Curves of CPU versus h^{-1} for Example 5.1. The preconditioner uses MIC(1).

Example 5.2. The coefficient functions are

$$a(\xi, \eta) = 1, \quad \text{and} \quad \theta(\xi, \eta) = 0.$$

For different discretization stepsizes, numerical results are listed in Table 5.2 and depicted in Figures 5.2a-5.2d.

Table 5.2. Iteration numbers and CPUs for Example 5.2

h^{-1}		16	32	40	48	64	80	100	128
RSMICCG(0)	IT	22	39	47	55	72	88	110	127
	CPU	0.01	0.08	0.16	0.30	0.78	1.59	3.48	7.20
SMWICCG(0)	IT	16	25	29	33	43	52	62	78
	CPU	0.01	0.05	0.08	0.16	0.42	0.88	1.85	4.10
ICCG(0)	IT	29	58	72	86	114	141	177	219
	CPU	0.01	0.11	0.21	0.38	1.02	2.02	4.32	9.52
RSMICCG(1)	IT	18	29	35	40	49	57	63	77
	CPU	0.01	0.06	0.13	0.26	0.59	1.27	2.42	5.30
SMWICCG(1)	IT	16	26	32	36	44	53	64	73
	CPU	0.01	0.04	0.09	0.18	0.44	0.95	1.95	3.95
ICCG(1)	IT	15	25	30	33	42	50	56	69
	CPU	0.01	0.05	0.09	0.15	0.40	0.78	1.43	3.24

For the preconditioned iterations based on MIC(0) factorization, we see that the SMWICCG(0) is the fastest one, then the RSMICCG(0), and the ICCG(0) is the slowest one, in both iteration numbers and CPU times. For the preconditioned iterations based on MIC(1) factorization, we see that the ICCG(1) is the fastest one, then the SMWICCG(1), and the RSMICCG(1) is the slowest one, in both iteration numbers and CPU times. However, the numerical behaviour of the SMWICCG(1) is comparable to that of the ICCG(1). In addition, the iteration numbers and CPU times of the SMWICCG(1) are comparable to those of the SMWICCG(0), and the other iterations with preconditioners based on MIC(1) factorization outperform those based on MIC(0) factorization, correspondingly.

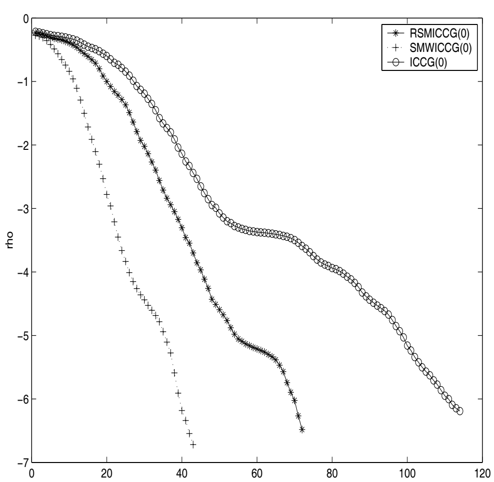

Figure 5.2a: Curves of ρ versus IT for Example 5.2 when $h^{-1} = 64$. The preconditioner uses MIC(0).

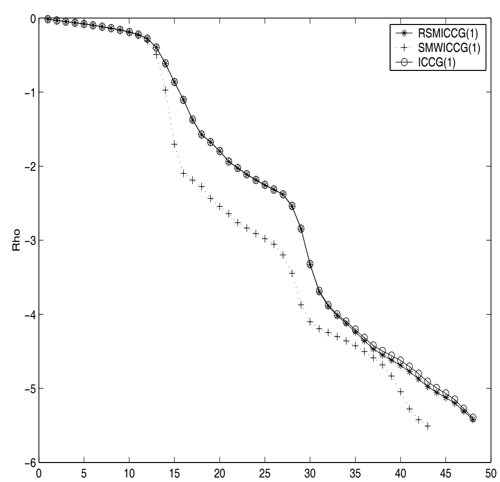

Figure 5.2b: Curves of ρ versus IT for Example 5.2 when $h^{-1} = 64$. The preconditioner uses MIC(1).

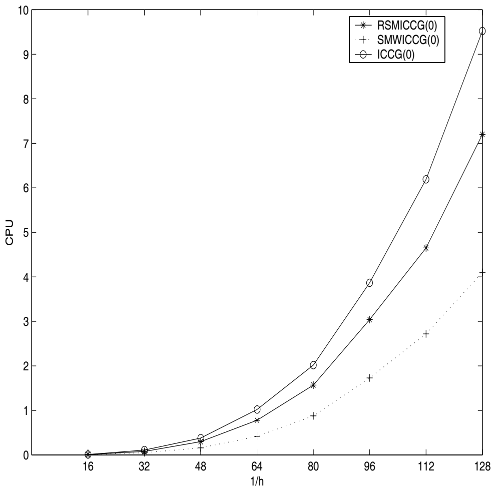

Figure 5.2c: Curves of CPU versus h^{-1} for Example 5.2. The preconditioner uses MIC(0).

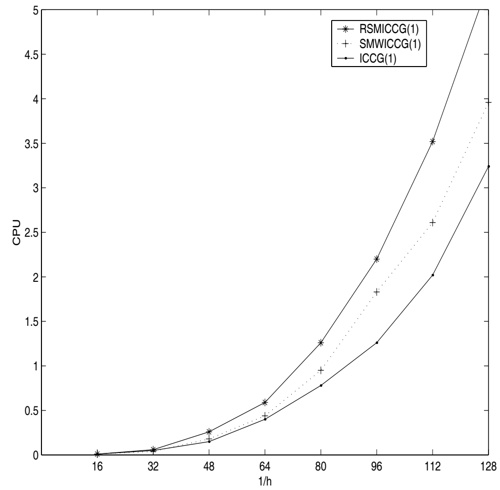

Figure 5.2d: Curves of CPU versus h^{-1} for Example 5.2. The preconditioner uses MIC(1).

Example 5.3. The coefficient functions are

$$a(\xi,\eta) = \begin{cases} 10000, & 0 < \xi < 0.5, \quad 0 < \eta < 1, \\ 0.1, & 0.5 \leq \xi < 1, \quad 0 < \eta < 1, \end{cases} \quad \text{and} \quad \theta(\xi,\eta) = 10.$$

For different discretization stepsizes, numerical results are listed in Table 5.3 and depicted in Figures 5.3a-5.3b.

Table 5.3. Iteration numbers and CPUs for Example 5.3

h^{-1}		16	32	40	48	64	80	100	128
RSMICCG(0)	IT	17	31	37	43	58	71	89	111
	CPU	0.01	0.07	0.14	0.26	0.66	1.38	3.09	6.72
SMWICCG(0)	IT	13	20	23	26	33	40	48	61
	CPU	0.01	0.03	0.07	0.13	0.33	0.70	1.30	3.24
ICCG(0)	IT	18	35	44	52	70	87	110	142
	CPU	0.01	0.06	0.13	0.23	0.62	1.29	2.68	6.21
RSMICCG(1)	IT	19	–	–	–	–	–	–	–
	CPU	0.01	–	–	–	–	–	–	–
SMWICCG(1)	IT	18	–	–	–	–	–	–	–
	CPU	0.00	–	–	–	–	–	–	–
ICCG(1)	IT	–	–	–	–	–	–	–	–
	CPU	–	–	–	–	–	–	–	–
CG	IT	67	224	345	471	818	1210	1944	3136
	CPU	0.01	0.20	0.49	0.97	3.15	7.94	22.46	67.22

Obviously, ICCG(1), and most cases of RSMICCG(1) and SMWICCG(1) fail to deliver an approximate solution to x^* due to break-down of the involved MIC(1) factorization. However, the iterations with preconditioners based on MIC(0) factorization succeed to produce an approximate solution in all cases, and they also outperform the CG in both iteration numbers and CPU times. For the preconditioned iterations based on MIC(0) factorization, we see that the SMWICCG(0) is the fastest one. The iteration numbers of the RSMICCG(0) is smaller than those of the ICCG(0), but the CPU times of the RSMICCG(0) are somewhat larger than those of the ICCG(0), correspondingly. Roughly speaking, the numerical behaviour of the RSMICCG(0) is comparable to that of ICCG(0).

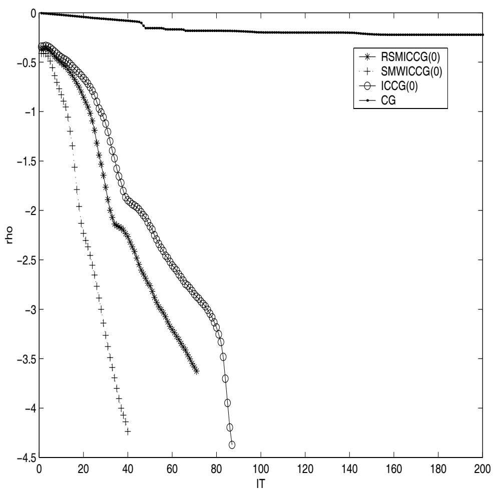

Figure 5.3a: Curves of ρ versus IT for Example 5.3 when $h^{-1} = 80$. The preconditioner uses MIC(0).

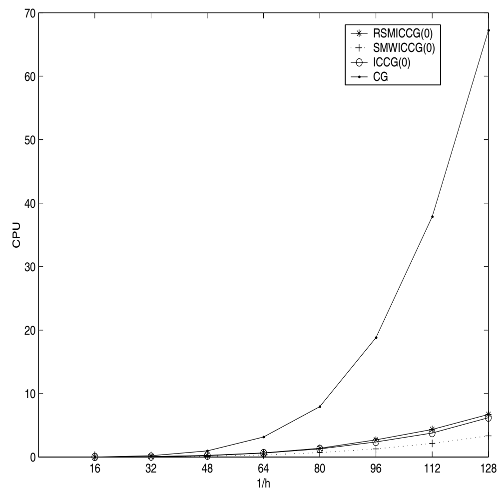

Figure 5.3b: Curves of CPU versus h^{-1} for Example 5.3. The preconditioner uses MIC(0).

Example 5.4. The coefficient functions are

$$a(\xi,\eta) = e^{\frac{1}{(\xi-0.5)^2+(\eta-0.5)^2+10}}, \quad \text{and} \quad \theta(\xi,\eta) = 1.$$

For different discretization stepsizes, numerical results are listed in Table 5.4 and depicted in Figures 5.4a-5.4c.

Table 5.4. Iteration numbers and CPUs for Example 5.4

h^{-1}		16	32	40	48	64	80	100	128
RSMICCG(0)	IT	22	38	46	55	71	88	111	143
	CPU	0.02	0.07	0.16	0.30	0.78	1.60	3.56	7.95
SMWICCG(0)	IT	15	24	29	33	43	51	62	79
	CPU	0.01	0.04	0.08	0.16	0.41	0.84	1.80	4.11
ICCG(0)	IT	29	56	69	83	110	138	174	223
	CPU	0.02	0.10	0.20	0.37	1.00	2.00	4.20	9.24
RSMICCG(1)	IT	18	30	37	43	54	65	79	100
	CPU	0.01	0.07	0.14	0.26	0.63	1.41	2.86	6.28
SMWICCG(1)	IT	16	27	32	37	45	55	69	88
	CPU	0.01	0.05	0.09	0.18	0.44	0.96	2.04	4.62
ICCG(1)	IT	16	26	31	36	46	55	69	85
	CPU	0.00	0.05	0.10	0.16	0.42	0.90	1.78	3.81
CG	IT	36	69	86	104	137	171	213	274
	CPU	0.01	0.06	0.12	0.21	0.53	1.15	2.58	5.77

Analogously, we observe that the iterations with preconditioners outperform the CG iteration in iteration numbers and CPU times. For the preconditioned iterations based on MIC(0) factorization, we see that the SMWICCG(0) is the fastest one, then the RSMICCG(0), and the ICCG(0) is the slowest one. For the preconditioned iterations based on MIC(1) factorization, we see that the SMWICCG(1) is the fastest one. The iteration numbers and CPU times of SMWICCG(1) are comparable to those of the ICCG(1). In addition, the RSMICCG(1) outperforms the RSMICCG(0). SMWICCG(1) is somewhat slower, but is almost comparable to SMWICCG(0).

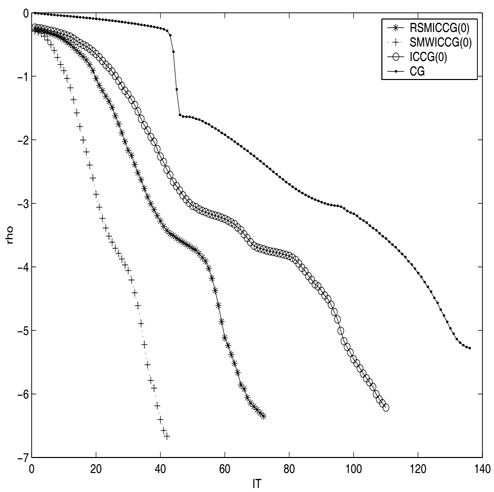

Figure 5.4a: Curves of ρ versus IT for Example 5.4 when $h^{-1} = 64$. The preconditioner uses MIC(0).

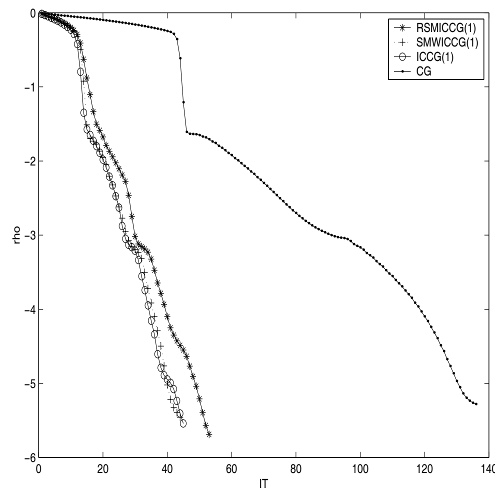

Figure 5.4b: Curves of ρ versus IT for Example 5.4 when $h^{-1} = 64$. The preconditioner uses MIC(1).

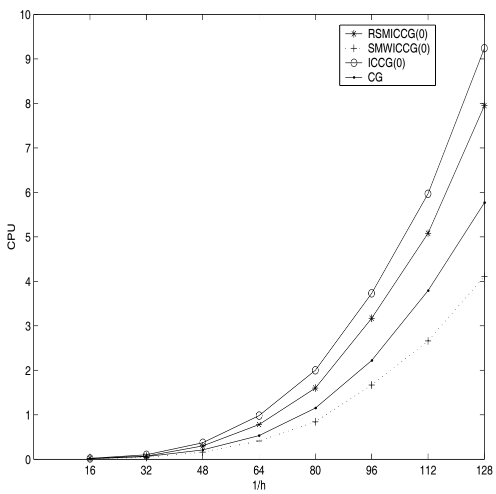

Figure 5.4c: Curves of CPU versus h^{-1} for Example 5.4. The preconditioner uses MIC(0).

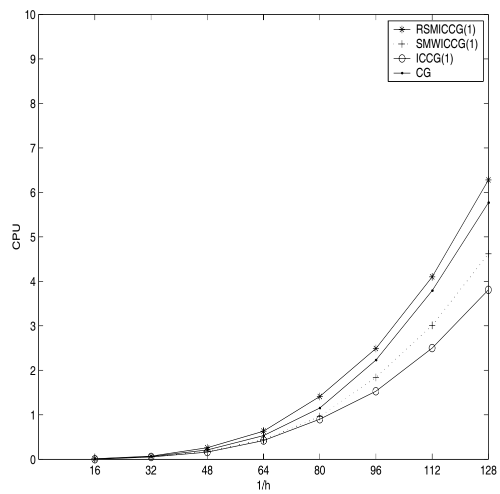

Figure 5.4d: Curves of CPU versus h^{-1} for Example 5.4. The preconditioner uses MIC(1).

References

[1] O. Axelsson, A generalized SSOR method, *BIT,* **12** (1972), 443-467.

[2] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.

[3] Z.Z. Bai, A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations, *Advances in Computational Mathematics,* **10** (1999), 169-186.

[4] Z.Z. Bai, Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems, *Applied Mathematics and Computation,* **109**:2-3 (2000), 273-285.

[5] Z.Z. Bai, Modified block SSOR preconditioners for symmetric positive definite linear systems, *Annals of Operations Research,* **103** (2001), 263-282.

[6] Z.Z. Bai and S.L. Zhang, A regularized conjugate gradient method for symmetric positive definite system of linear equations, *Journal of Computational Mathematics,* **20**:4 (2002), 437-448.

[7] J.W. Daniel, The conjugate gradient method for linear and nonlinear operator equations, *SIAM Journal on Numerical Analysis,* **4** (1967), 10-26.

[8] T. Dupont, R. Kendall and H.H. Rachford, Jr., An approximate factorization procedure for solving selfadjoint elliptic difference equations, *SIAM Journal on Numerical Analysis,* **5** (1968), 559-573.

[9] G.H. Golub and C.F. Van Loan, Matrix Computations, 3rd Edition, *The Johns Hopkins University Press,* Baltimore and London, 1996.

[10] A. Greenbaum, Iterative Methods for Solving Linear Systems, *SIAM,* Philadelphia, PA, 1997.

[11] M.R. Hestenes and E.L. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau Standards, Section B,* **49** (1952), 409-436.

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

[13] T.A. Manteuffel, An incomplete factorization technique for positive definite linear systems, *Mathematics of Computations,* **34** (1980), 473-497.

[14] 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, *Mathematics of Computations,* **31** (1977), 148-162.

[15] P. Saylor, Second order strongly implicit symmetric factorization methods for the solution of elliptic difference equations, *SIAM Journal on Numerical Analysis,* **11** (1974), 894-908.

[16] H.L. Stone, Iterative solution of implicit approximations of multidimensional partial differential equations, *SIAM Journal on Numerical Analysis,* **5** (1968), 530-558.

[17] R.S. Varga, Matrix Iterative Analysis, *Prentice Hall,* Englewood Cliffs, N.J., 1962.

[18] D.M. Young, Iterative Solution of Large Linear Systems, *Academic Press,* New York, 1971.