Circulant block-factorization preconditioners for elliptic problems

Ivan D. Lirkov, Svetozar D. Margenov and Panayot S. Vassilevski

Abstract

New circulant block-factorization preconditioners are introduced and studied. The general approach is first formulated for the case of block tridiagonal sparse matrices. Then estimates of the relative condition number for a model Dirichlet boundary value problem are derived. In the case of y-periodic problems the circulant block-factorization preconditioner is shown to give an optimal convergence rate. Finally, using a proper imbedding of the original Dirichlet boundary value problem to a y-periodic one a preconditioner of optimal convergence rate for the general case is obtained. The total computational cost of the preconditioner is $O(N \log N)$ (based on FFT), where N is the number of unknowns. That is, the algorithm is nearly optimal. Various numerical tests that demonstrate the features of the circulant block-factorization preconditioners are presented.

1 Introduction

In this paper we are concerned with the numerical solution of second order elliptic boundary value problems. Using finite differences or finite elements, such problems reduce to linear systems of the form $Au = b$, where A is a sparse matrix. We consider here symmetric and positive definite problems. The background of this study is the preconditioned conjugate gradient method. Let M be a preconditioner for the original matrix A. The construction of efficient preconditioners is motivated by the achievement of the following two goals: (a) to minimize the relative condition number $\kappa(M^{-1}A)$ and (b) to allow for efficient solution of the preconditioned system of equations $Mv = w$ for given vectors w. The simultaneous guarantee of the above criteria is the main objective of the studies in the field of constructing efficient preconditioning methods.

One of the most popular classical preconditioning technique is based on various *incomplete LU factorizations* of the matrix, see e.g., [2], [12]. The central idea of these methods is to approximate the exact factors in the Choleski (LU) factorization of the given sparse matrix such that the resulting approximate (lower and upper) triangular factors L and U have an in advance specified sparsity structure. For some of the pointwise incomplete factorization preconditioners it is proved (e.g., Gustafsson [14]) that $\kappa(M^{-1}A) = O(\sqrt{N})$, where N is the size of the discrete problem. The pointwise incomplete factorization algorithms are known (from practical experience) as very robust with respect to problem coefficients but a possible disadvantage is their inherently sequential nature. For parallel computation it is natural to look at block versions of the incomplete LU factorization methods. The basic schemes of incomplete block-factorization methods were proposed in Concus, Golub and Meurant [10], Axelsson [1], Axelsson and Polman [4], Axelsson and Eijkhout [5], see also Meurant [19], [20] and Chan and Vassilevski [9]. Earlier results are found in Kettler [16], Axelsson, Brinkkemper and Il'in [3]. The resulting convergence rate of the incomplete factorization preconditioners is not optimal (for a standard choice of the blocks in the matrix), i.e., it deteriorates with $h \to 0$, where h is the discretization parameter. This can be considered as a disadvantage of the block-ILU methods, although they provide highly parallel algorithms.

Another class of preconditioners based on a diagonal by diagonal averaging of the entries of a given matrix A to form a *circulant* approximation C was recently proposed in [7] (see also [21] and [15]). This leads to an improper in general approximation of the original Dirichlet boundary conditions with periodic ones. The use of the circulant approximations is motivated by their fast inversion based on the FFT. For the model problem, it is shown that the block circulant preconditioner can be constructed such that $\kappa(C^{-1}A) = O(\sqrt{N})$ which is asymptotically the same as for certain (modified) ILU type preconditioners. The circulant preconditioners are highly parallelizable, see e.g. [17] and [18], but they are substantially sensitive with respect to possible high variation of the coefficients of the given elliptic operator. In this respect they do not provide obvious advantages over the more classical incomplete block-factorization preconditioners.

The purpose of this paper is to relax the sensitivity of the circulant approximations with respect to possible high variation of the problem coefficients. Namely, we propose averaging of the coefficients of the given differential operator only along one of the coordinate directions (say, "y"). Thus if we have moderately varying coefficients in the y-direction the resulting preconditioners that we propose will give reasonable relative condition numbers.

The preconditioning technique we propose incorporates the circulant approximations into the framework of the LU block-factorization. The computational efficiency and parallelization of the resulting algorithm is as high as of the block circulant one ([7], [18]). The straightforward application of our new circulant block-factorization preconditioner to the Dirichlet boundary value problem leads to a relative condition number $\kappa(M^{-1}A) = O(\sqrt{N})$, i.e, of the same order as mentioned above. It is also true that if the solution of the elliptic problem is y-periodic, then the related preconditioner has an optimal convergence rate, i.e., $\kappa(M^{-1}A) = O(1)$. Based on this observation we use a proper imbedding of the original Dirichlet boundary value problem to a y-periodic one (possible for rectangular domains) and this allows us to obtain a preconditioner of optimal convergence rate for the Dirichlet boundary conditions case.

The remainder of this paper is organized as follows. In section §2 we describe the general ideas of the circulant block-factorization method. A model analysis of the relative condition number for two circulant algorithms based on averaging of the coefficients is presented in §3. In §4 we show numerical tests illustrating the convergence rate of the proposed algorithm for the considered Dirichlet boundary value problem. The optimal convergence rate of our preconditioner for y-periodic problems and how to imbed the original problem to y-periodic one is discussed in §5. Next, in §6 we show some additional numerical tests illustrating the optimal convergence rate of the circulant block-factorization preconditioner for the imbedded y-periodic problem. Some conclusions are drawn in §7.

2 Circulant block-factorization

We consider the following model 2D elliptic problem,

$$\begin{aligned} -(a(x,y)u_x)_x - (b(x,y)u_y)_y &= f(x,y), \qquad \forall (x,y) \in \Omega, \\ 0 < c_{\min} \le a(x,y),\, b(x,y) &\le c_{\max}, \\ u(x,y) = 0, \qquad \forall (x,y) \in \Gamma &= \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega = [0,1] \times [0,1]$ is covered by a uniform square mesh ω_h, with a size $h = 1/(n+1)$ for a given integer $n \ge 1$. Problem (1) is approximated by the standard 5-point finite difference stencil (the finite element method for linear triangular elements leads to a similar result). This discretization leads to a system of linear algebraic equations

$$Au = f. \tag{2}$$

If the grid points are ordered along, e.g., the y-grid lines, the matrix A admits a block tridiagonal structure (with blocks formed by the unknowns within a given grid-line). A can be written in the following form

$$A = tridiag(-A_{i,i-1}, A_{i,i}, -A_{i,i+1}) \qquad i = 1, 2, \ldots, n,$$

where

$$\begin{aligned} A_{i,i} = tridiag(-a_{j,j-1}, a_{j,j}, -a_{j,j+1}), &\quad j = (i-1)n+1, \ldots, in, \quad i = 1, 2, \ldots, n, \\ A_{i,i+1} = diag(a_{j,j+n}), &\quad j = (i-1)n+1, \ldots, in, \quad i = 1, \ldots, n-1, \\ A_{i,i-1} = diag(a_{j,j-n}), &\quad j = (i-1)n+1, \ldots, in, \quad i = 2, \ldots, n. \end{aligned}$$

The coefficients $a_{i,j}$ are positive and $a_{j,j} \ge a_{j,j-1} + a_{j,j+1} + a_{j,j+n} + a_{j,j-n}$, i.e., the matrix A satisfies the maximum principle.

Using the standard LU factorization procedure, we can first split $A = D - L - U$ into its block-diagonal and (negative) strictly block-triangular parts respectively. Then the *exact* block-factorization can be written in the form,

$$A = (X - L)(I - X^{-1}U),$$

where the blocks of $X = diag(X_1, X_2, \ldots, X_n)$ are to be determined. We have

$$A = X - L - U + LX^{-1}U.$$

Therefore

$$X = D - LX^{-1}U,$$

which rewritten component-wise gives

$$X_1 = A_{1,1}, \text{ and } X_i = A_{i,i} - A_{i,i-1}X_{i-1}^{-1}A_{i-1,i}, \qquad i = 2, \ldots, n \tag{3}$$

It is well-known that the above factorization exists if A is, for example, positive definite. This factorization can be used to solve system (2). This requires solution of linear systems involving the blocks X_i. Note that $\{X_i\}$ are in general full matrices and the resulting (direct) Gaussian elimination algorithm can become too expensive. The common idea of the block-ILU factorization methods is to approximate X_i (or X_i^{-1}) by sparse (band) matrices. The idea we explore in the present paper instead, is to first modify the original matrix A in such a way that the resulting matrices from the exact factorization of the thus modified matrix (in place of X_i) are now circulant.

We recall that a circulant matrix C has the form $(c_{k,j}) = \left(c_{(j-k)\,\mathrm{mod} N}\right)$, where N is the size of C. We will denote for any given coefficients $(c_0, c_1, \ldots, c_{n-1})$ by $C = (c_0, c_1, \ldots, c_{n-1})$ the circulant matrix

$$\begin{bmatrix} c_0 & c_1 & c_2 & \ldots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ldots & c_{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ c_1 & c_2 & \ldots & c_{n-1} & c_0 \end{bmatrix}.$$

Any circulant matrix can be factored as

$$C = F\Lambda F^*, \tag{4}$$

where Λ is a diagonal matrix containing the eigenvalues of C, and F is the Fourier matrix

$$F = \frac{1}{\sqrt{n}} \left\{ e^{2\pi \frac{jk}{n} i} \right\}_{0 \le j,k \le n-1}.$$

Here i stands for the imaginary unit.

We define now the general form of the circulant block-factorization preconditioning matrix C for the matrix A by

$$C = tridiag(-C_{i,i-1}, C_{i,i}, -C_{i,i+1}) \qquad i = 1, 2, \ldots, n,$$

where $C_{i,j} = Circulant(A_{i,j})$ is some given circulant approximation of the corresponding block $A_{i,j}$ (to be specified later). In what follows we use the exact block LU factorization for the preconditioner C. Note that the recursion (3), performed for C, is closed in the class of circulant matrices. That is, the corresponding blocks X_i are circulant and therefore the solution of the preconditioned system involving the matrix C can be performed efficiently based on the FFT using the representation (4) for the blocks $\{X_i\}$.

A detailed study of the properties of two averaging strategies for deriving circulant approximations of the blocks $A_{i,j}$ is given in the next section.

3 Model analysis of the relative condition number of two circulant block-LU factorization preconditioners

Two different approaches for circulant averaging are applied and studied in this section in the context of constructing circulant factorization preconditioners. We denote the resulting preconditioned algorithms by CBF1 and CBF2 respectively. A model analysis of the relative condition number $\kappa(C^{-1}A)$ is performed for the special case of the Laplace operator, i.e., when $a(x, y) = b(x, y) = 1$. The matrices in the case of this model problem are equipped with a subscript 0, i.e., we denote by A_0 and C_0 the matrices constructed for the Laplace operator.

3.1 Circulant block factorization preconditioning I

Algorithm CBF1

To determine the block matrix C we introduce the following mean-values

$$\begin{aligned}
\bar{a}_{i,i\pm1} &= \frac{1}{n}\sum_{j=(i-1)n+1}^{in} a_{j,j\pm n} \\
\bar{a}_{i,i,-1} &= \frac{1}{n}\sum_{j=(i-1)n+2}^{in} a_{j,j-1} \\
\bar{a}_{i,i,1} &= \frac{1}{n}\sum_{j=(i-1)n+1}^{in-1} a_{j,j+1} \\
\bar{a}_{i,i,0} &= \bar{a}_{i,i-1} + \bar{a}_{i,i,-1} + \bar{a}_{i,i,1} + \bar{a}_{i,i,1} + \frac{p}{n^2},
\end{aligned}$$

where p is a positive constant independent of n. Now we define the circulant matrices $C_{i,i\pm1}$ and $C_{i,i}$ by the relations

$$C_{i,i\pm1} = (\bar{a}_{i,i\pm1}, 0, \ldots, 0) = \bar{a}_{i,i\pm1} I, \qquad \text{where } I \text{ is the identity matrix,}$$
$$C_{i,i} = (\bar{a}_{i,i,0}, -\bar{a}_{i,i,1}, 0, \ldots, 0, -\bar{a}_{i,i,-1})$$

Finally the preconditioning matrix can be written in its block tridiagonal form as follows:

$$C = tridiag(-C_{i,i-1}, C_{i,i}, -C_{i,i+1})$$

The way of averaging used above is the same as in [7]. We get directly from the definition of C that it is symmetric (like A). Furthermore, the matrix C, satisfies the *maximum principle*, where the added diagonal term $\frac{p}{n^2}$ guarantees the strict diagonal dominance. These properties are summarized in the next lemma.

Lemma 1 *The preconditioning circulant matrix C defined by the CBF1 is symmetric and positive definite.*

Model analysis of the CBF1 algorithm

Consider $A_0 = tridiag(-I, B, -I)$ where $B = tridiag(-1, 4, -1)$ and the corresponding circulant preconditioner computed by Algorithm CBF1, $C_0 = tridiag(-I, D, -I)$ with $D = (2 + 2\beta + \frac{p}{n^2}, -\beta, 0, \dots, 0, -\beta), \beta = \frac{n-1}{n}$.

Consider also $G = tridiag(-1, 2, -1)$ and $H = (2\beta + \frac{p}{n^2}, -\beta, 0, \dots, 0, -\beta)$. Then we have

$$\begin{aligned} A_0 &= I \otimes G + G \otimes I, \\ C_0 &= I \otimes H + G \otimes I. \end{aligned}$$

We next estimate the condition number $\kappa(C_0^{-1} A_0)$. Let e_j be the j-th unit vector and y be an arbitrary n-vector. Then

$$\begin{aligned} H &= \beta G + \frac{p}{n^2} I - \beta (e_1 e_n^T + e_n e_1^T), \\ H &= \beta G + \frac{p}{n^2} I - \beta \left[(e_1 + e_n)(e_1 + e_n)^T - e_1 e_1^T - e_n e_n^T \right]. \end{aligned}$$

Since the matrices $(e_1 + e_n)(e_1 + e_n)^T$ and $G - e_1 e_1^T - e_n e_n^T$ are positive semi-definite, we have

$$y^T H y \leq 2\beta y^T G y + \frac{p}{n^2} y^T y.$$

We compute the eigenvalues $\lambda_k(G) = 4 \sin^2 \frac{k\pi}{2(n+1)}$, $k = 1, 2, \dots, n$. Hence $\lambda_k(G) \geq \frac{4}{(n+1)^2}$ and we get $y^T G y \geq \frac{4}{(n+1)^2} y^T y$. We obtain

$$\begin{aligned} y^T H y &\leq \left(2\beta + \frac{(n+1)^2 p}{4n^2} \right) y^T G y, \\ \lambda_{\min}(H^{-1} G) &\geq \frac{2}{p+4}. \end{aligned}$$

To estimate $\lambda_{\max}(H^{-1} G)$ we rewrite H in the form

$$H = \beta G + \frac{p}{n^2} I - \frac{\beta}{2} \left[(e_1 + e_n)(e_1 + e_n)^T - (e_1 - e_n)(e_1 - e_n)^T \right].$$

Using now the fact that the matrix $(e_1 - e_n)(e_1 - e_n)^T$ is positive semi-definite, we get

$$y^T H y \geq \beta y^T G y - \frac{\beta}{2} y^T (e_1 + e_n)(e_1 + e_n)^T y. \tag{5}$$

In order to estimate the last term we use the following estimate from [7],

$$y^T(e_1 + e_n)(e_1 + e_n)^T y \le n \left(\frac{4}{p} + \frac{1}{\pi} \right) y^T H y.$$

Substituting this inequality in (5), we obtain

$$\begin{aligned} y^T G y &\le \frac{5p+2}{p} n y^T H y, \\ \lambda_{\max}(H^{-1}G) &\le \frac{5p+2}{p} n. \end{aligned}$$

Therefore

$$\kappa(H^{-1}G) \le \frac{(5p^2 + 22p + 8)}{2p} n.$$

We summarize the above results in the following theorem.

Theorem 1 *The preconditioning matrix C defined by the CBF1 algorithm satisfies the following relative condition number estimate,*

$$\kappa(C_0^{-1} A_0) \le \frac{(5p^2 + 22p + 8)}{2p} n. \tag{6}$$

Remark 1 *The optimal value of the parameter $p = \sqrt{1.6}$ is easy to compute. Then the estimate*

$$\kappa(C_0^{-1} A_0) \le (2\sqrt{10} + 11) n$$

holds.

3.2 Circulant block factorization preconditioning II

Algorithm CBF2

The second approach of defining block circulant approximations can be interpreted as simultaneous averaging of the matrix coefficients and changing of the Dirichlet boundary conditions to periodic ones. We introduce now the following mean-values,

$$\begin{aligned} \bar{a}_{i,i\pm1} &= \frac{1}{n} \sum_{j=(i-1)n+1}^{in} a_{j,j\pm n} \\ \bar{a}_{i,i,-1} &= \frac{1}{n} \sum_{j=(i-1)n+2}^{in} a_{j,j-1} + \frac{d_i}{n} \\ \bar{a}_{i,i,1} &= \frac{1}{n} \sum_{j=(i-1)n+1}^{in-1} a_{j,j+1} + \frac{d_i}{n} \\ \bar{a}_{i,i,0} &= \frac{1}{n} \sum_{j=(i-1)n+1}^{in} a_{j,j}, \end{aligned}$$

where

$$d_i = \min(d_i^{(1)}, d_i^{(n)}),$$

and where

$$\begin{aligned}
d_1^{(1)} &= \frac{a_{1,1} - a_{1,2} - a_{1,n+1}}{2}, \\
d_i^{(1)} &= a_{(i-1)n+1,(i-1)n+1} - a_{(i-1)n+1,(i-2)n+1} - a_{(i-1)n+1,in+1} - a_{(i-1)n+1,(i-1)n+2}, \\
& \quad \forall i = 2, \ldots, n-1, \\
d_n^{(1)} &= \frac{a_{(n-1)n+1,(n-1)n+1} - a_{(n-1)n+1,(n-2)n+1} - a_{(n-1)n+1,(n-1)n+2}}{2},
\end{aligned}$$

and

$$\begin{aligned}
d_1^{(n)} &= \frac{a_{n,n} - a_{n,n-1} - a_{n,2n}}{2}, \\
d_i^{(n)} &= a_{in,in} - a_{in,(i-1)n} - a_{in,(i+1)n} - a_{in,in-1}, \qquad \forall i = 2, \ldots, n-1, \\
d_n^{(n)} &= \frac{a_{n^2,n^2} - a_{n^2,(n-1)n} - a_{n^2,n^2-1}}{2}.
\end{aligned}$$

The circulant blocks are defined explicitly by the formulas

$$\begin{aligned}
C_{i,i\pm1} &= (\bar{a}_{i,i\pm1}, 0, \ldots, 0) = \bar{a}_{i,i\pm1} I, \\
C_{i,i} &= (\bar{a}_{i,i,0}, -\bar{a}_{i,i,1}, 0, \ldots, 0, -\bar{a}_{i,i,-1}).
\end{aligned}$$

Then the block-tridiagonal preconditioning matrix C is defined by

$$C = tridiag(-C_{i,i-1}, C_{i,i}, -C_{i,i+1}).$$

Lemma 2 *The circulant preconditioning matrix C defined by the CBF2 algorithm is symmetric and positive definite.*

Proof: The symmetry follows directly from the construction of C, which is obtained by a block diagonal-by-diagonal averaging of a symmetric matrix. We have also from the definition, that C satisfies the *maximum principle* like the matrix A. Finally, we get the positive definiteness from the strict diagonal dominance of C in the rows corresponding to the mesh points adjacent to the vertical boundaries. ∎

Model analysis of the CBF2 algorithm

The corresponding circulant preconditioner is $C_0 = tridiag(-I, D, -I)$ with now $D = (4, -1, 0, \ldots, 0, -1)$.

Consider the matrices $G = tridiag(-1, 2, -1)$ and $H = (2, -1, 0, \ldots, 0, -1)$. Then

$$\begin{aligned}
A_0 &= I \otimes G + G \otimes I \\
C_0 &= I \otimes H + G \otimes I
\end{aligned}$$

We next estimate the condition number $\kappa(C_0^{-1} A_0)$. Let e_j be the j-th unit vector and y be an arbitrary n-vector. Then

$$\begin{aligned}
H &= G - (e_1 e_n^T + e_n e_1^T), \\
H &= G - \left[(e_1 + e_n)(e_1 + e_n)^T - e_1 e_1^T - e_n e_n^T\right].
\end{aligned}$$

Since the matrices $(e_1 + e_n)(e_1 + e_n)^T$ and $G - e_1 e_1^T - e_n e_n^T$ are positive semi-definite, we have

$$y^T H y \leq 2 y^T G y$$

which implies

$$y^T (H + \frac{1}{n^2} I) y \leq 2 y^T (G + \frac{1}{n^2} I) y$$

To estimate $\lambda_{\max}[G^{-1}H]$ we rewrite H as

$$H = G - \frac{1}{2}[(e_1 + e_n)(e_1 + e_n)^T - (e_1 - e_n)(e_1 - e_n)^T]$$

Using again the fact that $(e_1 - e_n)(e_1 - e_n)^T$ is positive semi-definite, we get

$$y^T H y \geq y^T G y - \frac{1}{2} y^T (e_1 + e_n)(e_1 + e_n)^T y \tag{7}$$

In order to estimate the last term, we borrow the following estimate from [7],

$$y^T (e_1 + e_n)(e_1 + e_n)^T y \leq n \left(4 + \frac{1}{\pi}\right) y^T (H + \frac{1}{n^2} I) y$$

Substituting this inequality in (7), we obtain

$$y^T (G + \frac{1}{n^2} I) y \leq 3 n y^T (H + \frac{1}{n^2} I) y$$

We compute the eigenvalues of G, $\lambda_k(G) = 4 \sin^2 \frac{k\pi}{2(n+1)}$, $k = 1, 2, \ldots, n$. Hence $\lambda(G) \geq \frac{1}{n^2}$ and therefore $G - \frac{1}{n^2} I$ is symmetric and positive semi-definite. We rewrite the matrices A_0 and C_0 now as

$$\begin{aligned} A_0 &= I \otimes (G + \frac{1}{n^2} I) + (G - \frac{1}{n^2} I) \otimes I \\ C_0 &= I \otimes (H + \frac{1}{n^2} I) + (G - \frac{1}{n^2} I) \otimes I \end{aligned}$$

Then for an arbitrary n^2-vector x it follows that

$$\frac{1}{2} \leq \frac{x^T A_0 x}{x^T C_0 x} \leq 3n.$$

Therefore the following theorem holds.

Theorem 2 *The preconditioning matrix C defined by the CBF2 algorithm gives a relative condition number estimated by*

$$\kappa(C_0^{-1} A_0) \leq 6n. \tag{8}$$

We point out that the above model analysis shows a better relative condition number estimate for the CBF2 algorithm in comparison with the CBF1 one.

In general the estimates obtained in this section are similar to the related ones for the block circulant preconditioners as introduced in [7]. The differencc in our construction is that the averaging procedure is only along the y-coordinate direction of the lines of ω_h. This fact assumes some particular advantages for problems with moderately varying coefficients in the y-direction. Furthermore we show in the next sections, that based on the CBF2 algorithm we will be able to construct an optimal order preconditioning algorithm for the considered elliptic boundary value problem.

4 Numerical tests I

In this section, we compare the convergence rate of our methods to the *block circulant* (BlockCr) and *pointwise circulant* (PointCr) preconditioners as introduced in [7], and the *modified incomplete factorization LU* (MILU) preconditioner (see e.g. in [2]). Our preconditioners are denoted by CBF1 and CBF2 respectively, following the notations introduced in §3. The test problem we used is taken from [7], and is

$$\frac{\partial}{\partial x}\left[(1+\epsilon e^{x+y})\frac{\partial u}{\partial x}\right]+\frac{\partial}{\partial y}\left[(1+\frac{\epsilon}{2}\sin(2\pi(x+y)))\frac{\partial u}{\partial y}\right]=f(x,y) \tag{9}$$

on $\Omega = [0,1]\times[0,1]$, where ϵ is a parameter. The computations are done with double precision on a SUN SPARCstation 2. The iteration stopping criterion is $\|r^{N_{it}}\|/\|r^0\| < 10^{-6}$, where r^j stands for the residual at the jth iteration step of the preconditioned conjugate gradient method.

Tables 1a÷1d show the number of iterations as a measure of the convergence rate of the compared five different preconditioners. All of these preconditioners are characterized by the estimate $\kappa(C^{-1}A) = O(n)$, or the expected number of the iterations have to grow as $N_{it} = O(\sqrt{n})$. The presented data demonstrate a slightly more quick growth of N_{it} with n, but in general the obtained results are in a good agreement with the theory.

The circulant preconditioners are competitive with the incomplete factorization MILU preconditioner for relatively small values of ϵ. This reflects the diagonal by diagonal (block) averaging of the coefficients, used in the circulant approximations. Such a fact was already observed in [7].

The comparison between the circulant preconditioners introduced in [7] (BlockCr and PointCr) and the circulant block factorization ones shows similar features. The CBF2 algorithm demonstrates some (predicted) advantages in comparison with the CBF1 algorithm for relatively large values of ϵ only.

Finally we point out that in contrast to BlockCr and PointCr, the last three algorithms have a preferred ordering direction of the mesh points. This is a well known fact that MILU converge faster for problems with a y-dominated anisotropy (i.e., when $\frac{b(x,y)}{a(x,y)} > 1$), being important for various practical applications. The proposed circulant block factorization algorithm converges faster for problems with a x-dominated anisotropy (when blocks $A_{i,i}$ are closer to diagonal matrix and the given circulant approximation improves).

5 Preconditioning of y-periodic problems and periodic imbedding of Dirichlet boundary value problems

Consider the problem (1), with modified boundary conditions:

$$-(a(x,y)u_x)_x - (b(x,y)u_y)_y = f(x,y) \qquad \forall (x,y) \in \Omega \tag{10}$$

where $\Omega = [0,1] \times [0,1]$, $u(0,y) = u(1,y) = 0$ $\forall y \in [0,1]$, and $u(x,0) = u(x,1)$, and $\frac{\partial u}{\partial y}(x,0) = \frac{\partial u}{\partial y}(x,1)$ $\forall x \in [0,1]$. The second set of boundary conditions are periodic and the related elliptic problem is called *y-periodic* one.

We assume that a uniform mesh ω_h, with stepsizes $h_x = 1/(n+1)$ and $h_y = 1/n$ is used. The 5-point finite difference stencil (like to the linear triangular finite elements) reduces the differential problem to a system of linear algebraic equations

$$Au = f.$$

Under the assumption of ordering the meshpoints along the y-grid lines, the matrix A admits the form,

$$A = tridiag(-A_{i,i-1}, A_{i,i}, -A_{i,i+1}) \qquad i = 1, 2, \ldots, n,$$

where

$$A_{i,i} = \begin{pmatrix} a^i_{1,1} & -a^i_{1,2} & 0 & \cdots & \cdots & \cdots & -a^i_{1,n} \\ -a^i_{2,1} & a^i_{2,2} & -a^i_{2,3} & \cdots & \cdots & \cdots & 0 \\ 0 & -a^i_{3,2} & a^i_{3,3} & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & \cdots & a^i_{n-2,n-2} & -a^i_{n-2,n-1} & 0 \\ 0 & \cdots & \cdots & \cdots & -a^i_{n-1,n-2} & a^i_{n-1,n-1} & -a^i_{n-1,n} \\ -a^i_{n,1} & \cdots & \cdots & \cdots & 0 & -a^i_{n,n-1} & a^i_{n,n} \end{pmatrix}, i = 1, 2, \ldots, n$$

and $A_{i,i-1}, A_{i-1,i}, i = 2, 3, \ldots, n$ are diagonal matrices.

One can see that the matrices $A_{i,i}$ have the same sparsity structure as the circulant approximations $C_{i,i}$, introduced in §3. Moreover, for the model problem ($a(x,y) = b(x,y) = 1$), all blocks of A are circulant. Therefore the block LU factorization recursion (3) remains closed during the elimination process into the class of circulant matrices.

In other words, the block LU factorization (3) defines a fast solver for the y-periodic discrete Poisson problems. By a standard "freezing coefficients" analysis of variable coefficients problem (cf., e.g., Bank [6], or the classical papers by D'yakonov [11] and Gunn [13]) one easily gets the following theorem.

Theorem 3 *Let the matrix A correspond to the finite difference approximation of the y-periodic problem (10). Then the circulant block-factorization algorithm CBF2 defines an optimal order preconditioner C, i.e., $\kappa(C^{-1}A) = O(1)$, uniformly in the mesh parameter $h \to 0$.*

Periodic imbedding of Dirichlet boundary value problems to y-periodic ones

We consider now the following imbedding of the original Dirichlet elliptic boundary value problem (1).

$$\mathcal{L}u = \mathcal{F}, \qquad \forall(x,y) \in [0,1] \times [-1,1] \tag{11}$$

where $\mathcal{L}(x,y)$ is an extension of the differential operator $L(x,y)$ related to the equation (1), and $\mathcal{F}(x,y)$ is a corresponding extension of the right side function $f(x,y)$, defined below. We have

$$\mathcal{L}(x,y) = \begin{cases} L(x,y) & \text{for} \quad 0 \le y \le 1 \\ L(x,-y) & \text{for} \quad -1 \le y \le 0 \end{cases},$$

This means, that the coefficients $a(x,y)$ and $b(x,y)$ of the operator (1) are extended outside Ω as shown above (i.e., using even-extension). The solution u and the right-hand side f are extended in odd-fashion. For example, we have

$$\mathcal{F}(x,y) = \begin{cases} f(x,y) & \text{for} \quad 0 \le y \le 1 \\ -f(x,-y) & \text{for} \quad -1 \le y \le 0 \end{cases},$$

and boundary conditions

$$u(0,y) = u(1,y) = 0 \qquad \forall y \in [-1,1],$$
$$u(x,1) = u(x,-1) \quad \forall x \in [0,1], \quad \text{and} \quad \frac{\partial u}{\partial y}(x,1) = \frac{\partial u}{\partial y}(x,-1) \quad \forall x \in [0,1].$$

The problem (11) is y-periodic with a solution coinciding to the solution of the problem (1) in $\Omega = [0,1] \times [0,1]$.

We call the problem (11) a y-periodic imbedding of the original Dirichlet boundary value problem (1). As a result, applying the algorithm CBF2 to the problem (11), we obtain spectrally equivalent preconditioner that requires $O(N \log N)$ operations per iteration for solving the corresponding discrete problem (1).

6 Numerical tests II

The numerical tests presented in this section illustrate the optimal convergence rate of the CBF2 algorithm for y-periodic problems, and of the y-periodic imbedding approach. Like for the reported results in §4, the computations are done with double precision on a SUN SPARCstation 2. The stopping criterion of the preconditioned conjugate gradient method is $\|r^{N_{it}}\|/\|r^0\| < 10^{-6}$, where r^j is the residual at iteration step j.

The function $f(x,y) = 4\pi^2 x(x-1)\left[\sin(2\pi y) - \frac{\epsilon}{2}\cos(2\pi(x+2y))\right] - [2 + \epsilon(2x+1)e^{x+y}]$ corresponds to the exact solution $u(x,y) = x(x-1)\sin 2\pi y$. The boundary conditions $u(0,y) = u(1,y) = 0$, $u(x,0) = u(x,1)$ and $\frac{\partial u}{\partial y}(x,0) = \frac{\partial u}{\partial y}(x,1)$ are also assumed. The stepsizes of ω_h are respectively $h_x = \frac{1}{n+1}$ and $h_y = \frac{1}{n}$.

In Table 2a the numbers of iterations of the CBF2 preconditioned conjugate gradient iterative solution related to the y-periodic problem (9) (see §4) are given. The results are the same for $n = 8, 16, \ldots, 256$, since the preconditioner has an optimal condition number.

Finally, in Table 2b the number of iterations of the CBF2 preconditioned conjugate gradient iterative solution related to the y-periodic imbedding of problem (9) are reported. The imbedding procedure is as described in §5. The boundary conditions are respectively modified to $u(0,y) = u(1,y) = 0$ $u(x,-1) = u(x,1)$ and $\frac{\partial u}{\partial y}(x,-1) = \frac{\partial u}{\partial y}(x,1)$. The stepsizes in ω_h are $h_x = \frac{1}{n+1}$ and $h_y = \frac{2}{n}$. The results are also independent of $n = 8, 16, \ldots, 256$ because of the optimal condition number of the preconditioner.

Both sets of results (in Tables 2a–2b) demonstrate an attractive efficiency of the circulant block-factorization preconditioning algorithm CBF2 for solving periodic and Dirichlet elliptic boundary value problems.

Acknowledgment

This report was partially supported by the National Science Foundation grant INT–92–20287 and also in part by the Bulgarian Ministry of Education and Science under grant MM/75-91.

References

[1] O. Axelsson, A survey of vectorizable preconditioning methods for large scale finite element matrices, *Colloquium Topics in Applied Numerical Analysis, (J.G. Verwer, ed.)*, Syllabus 4, Center of Mathematics and Informatics (CMI), Amsterdam (1983), 21–47.

[2] O. Axelsson and V.A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computations*, Academic Press, Orlando, F1.(1983).

[3] O. Axelsson, S. Brinkkemper, and V.P. Il'in, On some versions of incomplete block-matrix factorization methods, *Lin.Alg.Appl.*, **38** (1984), 3–15.

[4] O. Axelsson, B. Polman, On approximate factorization methods for block-matrices suitable for vector and parallel processors, *Lin. Alg. Appl.*, **77** (1986), 3–26.

[5] O. Axelsson and V.L. Eijkhout, Robust vectorizable preconditioners for three-dimensional elliptic difference equations with anisotropy, *Algorithms and Applications on Vector and Parallel Computers*, (H.J.J. te Riele , Th.J. Dekker, and H. van der Vorst, eds.), North Holland, Amsterdam, 1987.

[6] R.E. Bank, Marching algorithms for elliptic boundary value problems. II: The variable coefficient case, *SIAM J.Numer.Anal*, **14** (1977), 950–970.

[7] R.H.Chan and T.F.Chan, Circulant preconditioners for elliptic problems, *J. Numerical Lin.Alg.Appl.*,**1** (Mar. 1992), 77-101.

[8] T.F Chan and T.P. Mathew, The interface probing technique in domain decomposition, *SIAM J.Matrix Anal.Appl.* **13** (1992), 212–238.

[9] T.F.Chan and P.S.Vassilevski, A framework for block-ILU factorizations using block-size reduction, *UCLA CAM Report*, **92–46**, (1992).

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

[11] E.G. D'yakonov, On an iterative method for the solution of finite difference equations, *Dokl.Acad.Nauk SSSR*, **138** (1961), 522–525.

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

[13] J.E. Gunn, The solution of difference equations by semi-explicit iterative techniques, *SIAM J.Num.Anal.*, **2** (1965), 24–45.

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

[15] S. Holmgren and K. Otto, Iterative solution methods for block-tridiagonal systems of equations, *SIAM J.Matr.Anal.Appl.*, **13** (1992), 863–886.

[16] R. Kettler, Analysis and computations of relaxed schemes in robust multigrid and preconditioned conjugate gradient methods, *Multigrid Methods*, Proceedings, (W. Hackbusch and U. Trottenberg, eds.), Lect. Notes in Mathematics, **960**, Springer-Verlag (1982).

[17] C. Van Loan, *Computational frameworks for the fast Fourier transform*, SIAM, Philadelphia (1992).

[18] S.D. Margenov and I.T. Lirkov, Preconditioned conjugate gradient iterative algorithms for transputer based systems, *in Parallel and distributed processing, K.Boyanov editor*, Sofia (1993), 406–415.

[19] G. Meurant, The block-preconditioned conjugate gradient method on vector computers, *BIT*,**24** (1984), 623–633.

[20] G. Meurant, A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, *SIAM J.Matrix Anal.Appl.*,**13** (1992), 707–728.

[21] G. Strang, A proposal for Toeplitz matrix calculations, *Stud. Appl.Math.*, **74** (1986), 171–176.

Table 1a: Number of iterations for different preconditioners, $\epsilon = 0$.

n	N	BlockCr	PointCr	MILU	CBF1	CBF2
8	64	11	12	9	10	10
16	256	13	16	13	13	13
32	1024	17	20	19	17	17
64	4096	22	25	27	21	21
128	16384	28	33	40	28	28

Table 1b: Number of iterations for different preconditioners, $\epsilon = 0.01$.

n	N	BlockCr	PointCr	MILU	CBF1	CBF2
8	64	12	12	9	10	10
16	256	15	16	13	13	13
32	1024	20	20	19	17	17
64	4096	25	26	27	22	21
128	16384	33	34	40	28	28

Table 1c: Number of iterations for different preconditioners, $\epsilon = 0.1$.

n	N	BlockCr	PointCr	MILU	CBF1	CBF2
8	64	12	12	9	11	10
16	256	16	16	13	13	12
32	1024	20	20	19	18	16
64	4096	25	27	27	22	19
128	16384	35	36	39	29	25

Table 1d: Number of iterations for different preconditioners, $\epsilon = 1$.

n	N	BlockCr	PointCr	MILU	CBF1	CBF2
8	64	13	14	9	13	9
16	256	18	19	13	17	11
32	1024	25	27	18	23	15
64	4096	35	35	26	30	20
128	16384	50	51	38	41	26

Table 2a: Number of iterations for CBF2 preconditioner; y-periodic test problem.

ϵ	0.00	0.01	0.10	1.00
N_{it}	1	3	5	9

Table 2b: Number of iterations for CBF2 preconditioner; y-periodically imbedded problem.

ϵ	0.00	0.01	0.10	1.00
N_{it}	1	3	5	9