ABSTRACT. This paper deals with the iterative solution of a large sparse matrix when this matrix is in block-partitioned form arising from discrete elliptic Partial Differential Equations (PDEs). We introduce a block red-black coloring technique to reorder the unknowns, which is based on the standard red-black coloring. We propose some preconditioning techniques which combine the block incomplete LU factorization and the Schur complement technique, where the reduced system is preconditioned using different preconditioners. Some theoretical results to control the residual norm are presented.

1 Introduction

In this paper, we address the problem of developing preconditioners for the solution of large linear systems

\begin{equation}
Au = b,
\end{equation}

where \( A \) is an \( n \times n \) real sparse matrix. The above system is assumed to have the following block form

\begin{equation}
\begin{pmatrix}
B & F \\
E & C
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix} =
\begin{pmatrix}
f \\
y
\end{pmatrix}.
\end{equation}

This block structure arises from the standard and block Red-Black colorings described in Sections 2 and 3, or from the domain decomposition method which is based on decomposing the physical domain \( \Omega \) into a number of subdomains \( \Omega_i \), \( 1 \leq i \leq k \), for each of which an independent incomplete factorization can be computed and applied in parallel \([7–9, 25, 31]\). The main idea is to obtain more parallelism at the subdomain level rather than at the grid point level. Another approach to increase

AMS subject classification: 65F10, 65N06, 65N22.

Keywords: Sparse linear systems, ordering methods, block incomplete LU factorization, Schur complement, Krylov subspace.

Copyright © Applied Mathematics Institute, University of Alberta.
the degree of parallelism is to use several hyperplane wavefronts to sweep through the grid. For example van der Vorst [37] considered starting wavefronts from each of the four corners in a 2D rectangular grid. Earlier, Meurant [26] and van der Vorst [38] used a similar idea in which the grid is divided into equal parts (e.g., halves or quadrants) and each part is ordered in its own natural ordering.

A broad class of preconditioners is based on incomplete \( LU \) factorizations which were originally developed for M-matrices arising from the discretization of simple elliptic partial differential equations. For more details on this class of preconditioners, the interested readers can be refer to [1, 2, 10, 11, 20, 21, 23, 24, 27–31, 34–36, 39].

The matrix \( A \) can be written as

\[
A = \begin{pmatrix} B & \varepsilon \\ E & S \end{pmatrix} \begin{pmatrix} I & B^{-1} \varepsilon \\ \varepsilon & I \end{pmatrix},
\]

where \( S = C - EB^{-1} \varepsilon \) is the Schur complement matrix.

The matrix \( S \) does not need to be formed explicitly in order to solve the reduced system by an iterative method. For example, if a Krylov subspace method without preconditioning is used, then the only operations that are required with the matrix \( S \) are matrix-vector operations \( w = S v \). Such operations can be performed as follows

(a) Compute \( v' = F v \).
(b) Solve \( B z = v' \).
(c) Compute \( w = C v - E z \).

The above procedure involves only matrix-vector multiplications and one linear system solution (b) with \( B \). Note that the linear system with \( B \) must be solved exactly, by using a direct solution technique or by an iterative technique with a high level of accuracy.

It is well known that the ordering of the unknowns can have a significant effect on the convergence of a preconditioned iterative method and on its implementation on a parallel computer [13, 14], especially in the application of the \( ILU \) preconditioners. Indeed, if we consider a graph containing a central node and whose only edges join this node to the \((n-1)\) remaining ones. And consider now the two extreme orderings, the first giving the number 1 to the central node and the second giving it the last number \( n \). Let \( A_1 \) and \( A_2 \) be the adjacency matrices to these
two orderings respectively, which can be written as

\[
A_1 = \begin{pmatrix}
\times & \times & \times & \ldots & \times \\
\times & \times & \times & \ldots & \times \\
\times & \times & \times & \ldots & \times \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\times & \times & \times & \ldots & \times \\
\end{pmatrix}
\]

\[
A_2 = \begin{pmatrix}
\times & \times & \times & \ldots & \times \\
\times & \times & \times & \ldots & \times \\
\times & \times & \times & \ldots & \times \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\times & \times & \times & \ldots & \times \\
\end{pmatrix}
\]

Let \(A_1 = L_1U_1\) and \(A_2 = L_2U_2\) be the factorizations according to Gaussian elimination process, \(A_1\) and \(A_2\) have the same number of non-zero entries. In the first ordering the structure \((L_1 + U_1)\) has all its coefficients different from zero, whereas with the second ordering \((L_2 + U_2)\) it has the same structure as \(A_2\). Furthermore, the factors governed by the factorization \(ILU(0)\) of \(A_2\) are exactly \(L_2\) and \(U_2\)! This paper is organized as follows: Section 2 gives a brief overview of a standard red-black coloring technique. Section 3 describes an extension of the previous section named block red-black coloring. Section 4 provides some preconditioners arising from block \(ILU\) factorization and some theoretical results to control the residual norm. Section 5 gives some numerical experiments and a few concluding remarks.

2 Standard red-black coloring  The problem addressed by multicoloring is to determine a coloring of the nodes of the adjacency graph of a matrix such that any two adjacent nodes have different colors. For simple two-dimensional finite difference grid (a five-point operator), we can easily separate the grid points into two sets, red and black, so that the nodes in one set are adjacent only to nodes in the other set, as shown in Figure 1 for a \(12 \times 8\) grid (with white nodes correspond to red nodes). Assume that the unknowns are labeled by listing the red unknowns first
together, followed by the black ones, we obtain a system of the form

$$\begin{pmatrix} D_1 & E \\ E & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2.1}$$

where $D_1$ and $D_2$ are diagonal matrices.

One way to exploit the red-black ordering is to use the standard SSOR or $ILU(0)$ preconditioners for solving (2.1). The preconditioning operations are highly parallel. For example, the linear system that arises from the forward solving in SSOR will have the form

$$\begin{pmatrix} D_1 & E \\ E & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{2.2}$$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{red_black_grid.png}
\caption{A standard red-black coloring of $12 \times 8$ grid.}
\end{figure}

This system can be solved by performing the following sequence of operations:

Solve $D_1x = f$.
Compute $\tilde{g} = g - Ex$.
Solve $D_2y = \tilde{g}$.

This consists of two diagonal scalings (operations 1 and 3) and a sparse matrix-vector product (operation 2). The situation is identical with the $ILU(0)$ preconditioning. A simple look at the structure of the $ILU$ factors reveals that many more elements are dropped with the red-black ordering than with the natural ordering. The result is that the number of
iterations to achieve convergence can be much higher with the red-black ordering than with the natural ordering [30].

The second method that has been used in connection with the red-black ordering solves the reduced system which involves only the black unknowns. Eliminating the red unknowns from (2.1) results in the reduced system:

\[(2.3) \quad (D_2 - ED_1^{-1}F)y = g - ED_1^{-1}f.\]

Note that this new system is again a sparse linear system with about half as many unknowns. In addition, it has been observed that for easy problems, the reduced system can often be solved efficiently with only diagonal preconditioning. The computation of the reduced system is a highly parallel and inexpensive process. We can also use a multi-level ordering of the grid points. For more details on this technique and on the order of the condition number see the references of Brand and Heinmann [4] and [12] where they proposed a repeated red-black (RRB) ordering.

**3 Block red-black coloring** We now consider an extension of the red-black coloring which consists of transforming the system (1.1) into the following block form

\[(3.1) \quad \begin{pmatrix} B & F \\ E & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}.\]

Here \(B\) and \(C\) are block diagonal matrices, where each block is of size 2.

A block of size 2 will be found by coupling a node \(j\) with its neighbor \((j+1)\) with the same color (Red) followed by two nodes that are different color (Black). Assume again that the unknowns are labeled by listing the red unknowns first together, followed by the black ones. This block two-by-two colors is illustrated in Figure 2 for a 12×8 grid (with white nodes correspond to red nodes).
This leads to a $2 \times 2$ submatrix of the form
\begin{equation}
\begin{pmatrix}
b_{j,j} & b_{j,j+1} \\
b_{j+1,j} & b_{j+1,j+1}
\end{pmatrix}, \quad \begin{pmatrix}
c_{j,j} & c_{j,j+1} \\
c_{j+1,j} & c_{j+1,j+1}
\end{pmatrix}.
\end{equation}

This recursive block version technique \textit{Repeated Block Red-Black ordering (RBRB)} can be applied as in \cite{4, 12}. This technique starts with all red nodes indicated by $\circ$, followed by half of the remaining black nodes corresponding to alternating horizontal grid line indicated by $\star$. It can be easily checked that the remaining black nodes indicated by $\bullet$ form a regular grid again with a double mesh spacing as the original one. The process can then be repeated recursively until a sufficiently coarse level is reached on which the usual natural ordering can be applied. Figure 3 shows an example of \textit{RBRB} ordering.

Figure 3. A \textit{RBRB} ordering, stopping after 1 coarse level (indicated by $\bullet$).
4 Preconditioning techniques and preconditioned iterations

Some results to control the preconditioned residual norm are given in this section. First a preconditioner to solve the linear system of the block form is developed:

\[
\begin{pmatrix}
B & F \\
E & C \\
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
\end{pmatrix} = 
\begin{pmatrix}
f \\
g \\
\end{pmatrix},
\]

where $B$ and $C$ are square matrices. Furthermore, we assume that $B$ is nonsingular. The block lower-upper (LU) triangular factorization of $A$ can be written as

\[
\begin{pmatrix}
B & F \\
E & C \\
\end{pmatrix} = 
\begin{pmatrix}
B & S \\
E & I \\
\end{pmatrix}
\begin{pmatrix}
I & B^{-1}F \\
I & I \\
\end{pmatrix},
\]

where $S$ is the Schur complement matrix

\[
S = C - EB^{-1}F.
\]

To compute the solution $(x, y)^T$ of the linear system (4.1), we therefore need to perform the following steps:

(a) Compute $g' = g - EB^{-1}f$.
(b) Solve $Sy = g'$.
(c) Solve $Bx = (f - Fy)$.

If the cost for computing the exact inverse of $B$ is prohibitive, then we first approximate $B^{-1}$ by a sparse matrix $Y$ using sparse approximate inverse techniques; see [3, 18, 19], to approach the Schur complement $S$ with a sparse matrix. The matrix $\tilde{S} = C - EYF$ is sparse. And solving (b) with it is more economical since $\tilde{S}$ is sparser than $S$. Note that, we can also use an approximate factorization of $\tilde{S}$:

\[
\tilde{S} = \tilde{S} + R_{\tilde{S}} = L_{\tilde{S}}U_{\tilde{S}} + R_{\tilde{S}}
\]

which define a preconditioner for the Schur complement matrix, $\tilde{S} = L_{\tilde{S}}U_{\tilde{S}}$, and here, $R_{\tilde{S}}$ is the error matrix.

4.1 Left preconditioning ILU In the following we suppose that the inverse of $B$ is computed exactly. Then the preconditioner can be written as:

\[
M = \begin{pmatrix} B & \tilde{S} \\ E & I \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ I & I \end{pmatrix} = L_MU_M,
\]
with \( S \) being some approximate to the Schur complement \( S \). For example, in the form of an approximate \( LU \) factorization of \( S \). When \( C \) is a block diagonal matrix and is not singular, we can choose \( \tilde{S} = C \), since the only inverse that occurs is that of a block diagonal matrix with small blocks, or \( \tilde{S} = I \), i.e., no preconditioning of \( S \).

We can easily show that the preconditioned system has the particular form

\[
(4.6) \quad M^{-1} A = \begin{pmatrix} I & B^{-1} F \\ \tilde{S}^{-1} S & I \end{pmatrix} \begin{pmatrix} I & B^{-1} F \\ \tilde{S}^{-1} S & I \end{pmatrix}^{-1} = U_M^{-1} DU_M.
\]

Then the quality of the preconditioner \( M \) depends on that of \( \tilde{S} \), and the concentration of the spectrum of \( M^{-1} A \) depends only on that of \( \tilde{S}^{-1} S \), since

\[
\sigma(M^{-1} A) = \{1\} \cup \sigma(\tilde{S}^{-1} S).
\]

Consider now a GMRES iteration to solve the preconditioned system

\[
(4.7) \quad M^{-1} Au = M^{-1} b.
\]

Given an initial guess \( u_0 = (x_0, y_0)^T \), we denote the preconditioned initial residual by

\[
r_0 = M^{-1}(b - Au_0) = \begin{pmatrix} z_0 \\ s_0 \end{pmatrix}.
\]

It is useful to recall that the GMRES algorithm finds a vector \( u \) in the affine subspace \( u_0 + K_m \), where \( K_m \) is a Krylov subspace:

\[
K_m = K_m(M^{-1} A, r_0) = \text{Span} \{ r_0, (M^{-1} A)r_0, \ldots, (M^{-1} A)^{m-1} r_0 \}.
\]

The GMRES minimizes \( \| M^{-1}(b - Au) \|_2 \) for an arbitrary \( u \) in \( u_0 + K_m \)

\[
u = u_0 + w, \quad w \in K_m.
\]

Then, the preconditioned residual takes the form:

\[
r = M^{-1}(b - Au) = M^{-1}(b - Au_0 - Aw) = r_0 - M^{-1} Aw.
\]

\( \sigma(X) \) denotes the spectrum of a matrix \( X \).
Theorem 4.1. Assume that the reduced system

\[ \tilde{S}^{-1}S_y = \tilde{S}^{-1}(g - EB^{-1}f) \]

is solved with GMRES starting with an arbitrary initial guess \( y_0 \) and let \( s_m = \tilde{S}^{-1}(g' - S_y) \) be the preconditioned residual obtained at the \( m \)-th step. Then the preconditioned residual vector \( r_m \) obtained at the \( m \)-th step of GMRES for solving the block system (4.1) preconditioned by \( M \) of (4.6) and with an initial guess \( u_0 = (x_0, y_0)^T \) satisfies \( Bx_0 = f - Fy_0 \), satisfies the inequality

\[ \|r_m\|_2 \leq (1 + \|B^{-1}F\|_2)\|s_m\|_2. \]

In particular, if \( s_m = 0 \), then \( r_m = 0 \).

Proof. Let \( s_m = p_m(\tilde{S}^{-1}S)s_0 \) be the preconditioned residual obtained at the \( m \)-th step of the GMRES algorithm starting with an arbitrary initial guess \( y_0 \) to solve the preconditioned reduced system

\[ (4.8) \quad \tilde{S}^{-1}S_y = \tilde{S}^{-1}g', \]

where \( g' = g - EB^{-1}f \) and \( p_m \) is the \( m \)-th residual polynomial, which minimizes \( \|p(\tilde{S}^{-1}S)s_0\|_2 \) among all polynomials \( p \) of degree \( m \) satisfying the constraint \( p(0) = 1 \).

Let

\[ p_m(t) = \sum_{i=0}^{m} \alpha_i t^i. \]

The preconditioned residual \( r_m \) obtained at the \( m \)-th step of GMRES algorithm for solving the preconditioned unreduced system minimizes \( \|p(M^{-1}A)r_0\|_2 \) over all polynomials \( p \) of degree \( m \) which are consistent, i.e., \( p(0) = 1 \), and satisfies

\[ \|r_m\|_2 \leq \|p_m(M^{-1}A)r_0\|_2 \]

\[ \leq \left\| U_M^{-1} \begin{pmatrix} p_m(I) \\ p_m(\tilde{S}^{-1}S) \end{pmatrix} U_M \begin{pmatrix} z_0 \\ s_0 \end{pmatrix} \right\|_2 \]

\[ \leq \|U_M^{-1} \|_2 \left\| \begin{pmatrix} p_m(I) \\ p_m(\tilde{S}^{-1}S) \end{pmatrix} \begin{pmatrix} \delta \\ s_0 \end{pmatrix} \right\|_2, \]

where

\[ (4.9) \quad \delta = z_0 + B^{-1}Fs_0. \]
Due to the particular structure of $B$, that is a block diagonal matrix, and if we suppose that its blocks have a small size then, we have to solve the system by a direct solution technique. On these conditions we have to solve
\begin{equation}
Bx_0 = f - Fy_0
\end{equation}
extactly and we have $\delta = 0$. Then
\begin{equation}
|r_m|_2 \leq \|U_M^{-1}\|_2 \|p_m(S^{-1}S)s_0\|_2,
\end{equation}
that is,
\begin{equation}
|r_m|_2 \leq \|U_M^{-1}\|_2 \|s_m\|_2 \leq (1 + \|B^{-1}F\|_2)\|s_m\|_2.
\end{equation}

**Remark 4.1.** In general, systems involving $B$ may be solved in many ways, depending on their difficulty and what we know about $B$. If $B$ is known to be well conditioned, then triangular solves with incomplete $LU$ factors may be sufficient. For more difficult $B$ matrices, the incomplete factors may be used as a preconditioner for an inner iterative process for $B$. Suppose now that an iterative method is used to solve (4.10). Let $\bar{x}_0$ the exact solution and $x_0^0$ an initial guess, then at $k^{th}$ step, the error is
\begin{equation}
e_k^k = \bar{x}_0 - x_k^0.
\end{equation}
We have
\begin{equation}
Mr_0 = b - Au_0 = \begin{pmatrix} f - Bx_0^k - Fy_0 \\ g - Ex_0^k - Cy_0 \end{pmatrix} = \begin{pmatrix} Bc_0^k \\ \tilde{g} \end{pmatrix}.
\end{equation}
Also, we get
\begin{equation}
Mr_0 = \begin{pmatrix} Bz_0 + Fs_0 \\ Ez_0 + EB^{-1}Fs_0 + \tilde{S}s_0 \end{pmatrix}.
\end{equation}
From the first component we have
\begin{equation}
B(z_0 + B^{-1}Fs_0) = Bc_0^k,
\end{equation}
since $B$ is not singular then
\begin{equation}
\delta = e_0^k,
\end{equation}
and we have
\begin{equation}
|r_m|_2 \leq \|U_M^{-1}\|_2 \left(||p_m(1)||^2 ||e_0^k||_2^2 + \|s_m\||_2^2\right)^{1/2}.
\end{equation}
4.2 Split preconditioning Let \( A = L_M U_M + R_A \), where \( R_A \) is the error matrix and
\[
L_M = \begin{pmatrix} B & S \\ E & \tilde{S} \end{pmatrix}, \quad U_M = \begin{pmatrix} I & B^{-1}F \\ I & I \end{pmatrix},
\]
Then
\[
(4.14) \quad L_M^{-1}AU_M^{-1} = \begin{pmatrix} I \\ \tilde{S}^{-1}S \end{pmatrix},
\]
and the split preconditioning is obtained by solving:
\[
(4.15) \quad \begin{cases} 
L_M^{-1}AU_M^{-1}v = L_M^{-1}b \\
U_Mu = v.
\end{cases}
\]
The Krylov subspace associated with this system is:
\[
K_m(L_M^{-1}AU_M^{-1}, r_0) = \text{Span} \{ r_0, (L_M^{-1}AU_M^{-1})r_0, \ldots, (L_M^{-1}AU_M^{-1})^{m-1}r_0 \},
\]
where
\[
r_0 = L_M^{-1}b - (L_M^{-1}AU_M^{-1})U_Mu_0 = \begin{pmatrix} z_0 \\ s_0 \end{pmatrix},
\]
where \( u_0 = (x_0, y_0)^T \) is an initial guess to solve (4.1), and \( v_0 = UAu_0 \).

**Theorem 4.2.** Assume that the reduced system (4.8) is solved with GMRES starting with an arbitrary initial guess \( y_0 \) and let \( s_m = \tilde{S}^{-1}(g' - \tilde{S}y_m) \) be the preconditioned residual obtained at the \( m \)th step. Then the preconditioned residual vector \( r_{m+1} \) obtained at the \((m+1)\)th step of GMRES for solving the block system (4.15) with an arbitrary initial guess \( u_0 = (x_0, y_0)^T \) satisfies the inequality
\[
(4.16) \quad \|r_{m+1}\|_2 \leq \|I - \tilde{S}^{-1}S\|_2\|s_m\|_2.
\]
In particular, if \( s_m = 0 \), then \( r_{m+1} = 0 \).

**Proof.** Let \( p_m \) be the \( m \)th residual polynomial, which minimizes \( \|p(\tilde{S}^{-1}S)s_0\|_2 \) among all polynomials \( p \) of degree \( m \) satisfying the constraint \( p(0) = 1 \), then the residual vector of the \( m \)th GMRES approximation associated with the preconditioned reduced system (4.8) is the form \( s_m = p_m(\tilde{S}^{-1}S)s_0 \).

We define the polynomial
\[
q_{m+1}(t) = (1 - t)p_m(t).
\]
The rest is similar to the proof of Theorem 4.1. \( \square \)
Definition 4.1. A matrix $B \in \mathbb{R}^{n \times n}$ is said to be a $Z$-matrix if all of its off-diagonal entries (if any) are nonpositive: $b_{i,j} \leq 0$, for all $i \neq j$.

Remark 4.2. Assume that $B$ is a block diagonal matrix, where each block is of size 2. Furthermore, we suppose that $B$ is a $Z$-matrix and strictly diagonally dominant. Then each block $\begin{pmatrix} b_{j,j} & b_{j,j+1} \\ b_{j+1,j} & b_{j+1,j+1} \end{pmatrix}$ is nonsingular and its inverse can be written

\begin{equation}
\begin{pmatrix} b_{j,j} & b_{j,j+1} \\ b_{j+1,j} & b_{j+1,j+1} \end{pmatrix}^{-1} = \frac{1}{\det_j} \begin{pmatrix} b_{j+1,j+1} & -b_{j,j+1} \\ -b_{j+1,j} & b_{j,j} \end{pmatrix} = \begin{pmatrix} \beta_{j,j} & \beta_{j,j+1} \\ \beta_{j+1,j} & \beta_{j+1,j+1} \end{pmatrix},
\end{equation}

where $\det_j = b_{j,j}b_{j+1,j+1} - b_{j+1,j}b_{j,j+1}$. Since $B$ is strictly diagonally dominant then there exists $\varepsilon_j > 0$ such that $\det_j > \varepsilon_j$, then one easily has

\begin{equation}
0 \leq \beta_{i,j} < \frac{\gamma}{\varepsilon_B} \quad \text{for each} \quad 1 \leq i, j \leq m,
\end{equation}

where $\varepsilon_B = \min \{\varepsilon_j\}$ and $\gamma = \max_{1 \leq i, j \leq n} \{|a_{i,j}|\}$. In other words,

$$
\|B^{-1}\|_2 \leq \|B^{-1}\|_F = \left( \sum_{j=1}^{m} \sum_{i=1}^{m} \beta_{i,j}^2 \right)^{\frac{1}{2}} \leq \sqrt{2m} \frac{\gamma}{\varepsilon_B}.
$$

By Theorem 4.1, we deduce

\begin{equation}
\|r_m\|_2 \leq \left( 1 + \sqrt{2m} \frac{\gamma}{\varepsilon_B} \|F\|_2 \right) \|s_m\|_2.
\end{equation}

4.3 Schur complement preconditioning Assume that $C$ has the same order $m$ and the same structure (block diagonal matrix of size 2) and satisfies the same properties of $B$. Then, we can choose $S = C$ as a preconditioner for the Schur complement matrix $S$. In this case and
from Theorem 4.2, one easily has

\[ \| r_{m+1} \|_2 \leq \| I - C^{-1} S \|_2 \| s_m \|_2 \]
\[ \leq \| C^{-1} E B^{-1} F \|_2 \| s_m \|_2 \]
\[ \leq \| C^{-1} \|_2 \| B^{-1} \|_2 \| E \|_2 \| F \|_2 \| s_m \|_2 \]
\[ \leq 2m \frac{\gamma}{\varepsilon_C \varepsilon_B} \| E \|_2 \| F \|_2 \| s_m \|_2. \]

Then, we get

\[ \| r_{m+1} \|_2 \leq n \left( \frac{\gamma}{\varepsilon} \right)^2 \| E \|_2 \| F \|_2 \| s_m \|_2, \]

where \( \varepsilon = \min \{ \varepsilon_B, \varepsilon_C \} \).

Now take \( \hat{S} = \hat{L}_S \hat{U}_S + \hat{R}_S \) to be an approximate factorization of \( S \).

The other version in which the Schur complement matrix \( S \) is allowed to be preconditioned from the left and the right by \( L_S \) and \( U_S \), respectively.

Consider the reduced system

\[ \begin{cases}
L_S^{-1} S U_S^{-1} w = L_S^{-1} g' \\
U_S y = w,
\end{cases} \tag{4.21} \]

where \( g' = g - E B^{-1} f \). The system (4.21) is solved by GMRES with an arbitrary initial guess \( y_0 \) \( (U_S y_0 = w_0) \). Let

\[ s_m = L_S^{-1} g' - (L_S^{-1} S U_S^{-1}) U_S y_m. \tag{4.22} \]

Then the residual vector obtained at the \((m+1)^{th}\) step of GMRES for solving the system (4.2) with left preconditioning \( L_A \) and right preconditioning \( U_A \) and with an initial guess \( u_0 = (x_0, y_0)^T \) satisfies the inequality

\[ \| r_{m+1} \|_2 \leq \| I - L_S^{-1} S U_S^{-1} \|_2 \| s_m \|_2. \tag{4.23} \]

The proof of this result starts with the equality

\[ \left( \begin{array}{c}
B \\
E \\
L_S
\end{array} \right)^{-1} A \left( \begin{array}{c}
I \\
B^{-1} F \\
U_S
\end{array} \right)^{-1} = \left( \begin{array}{c}
I \\
L_S^{-1} S U_S^{-1}
\end{array} \right). \tag{4.24} \]

The rest is similar to that of the previous result.
5 Numerical experiments and conclusion  We consider the five point upwind finite difference scheme of the Laplace and convection-diffusion equations [40]. The advantage of this scheme is that it can produce stable discretizations, while with the standard central finite difference approximation and for small mesh sizes, spurious oscillations appear in the discrete solution of the convection-diffusion equation [27]. We use the standard and block red-black colorings described in sections 2 and 3 with uniform meshsize $h$ in both $x$ and $y$ directions. In these cases, we compute the Schur complement matrix $S$ explicitly.

In our tests, we present the number of preconditioned GMRES(20) iterations [33] with four different grid sizes. In Tables 4 and 7, "solv." shows the CPU time in seconds for the solution phase; "prec." shows the CPU time in seconds taken in constructing the preconditioners; "spar." shows the sparsity ratio which is the ratio between the number of nonzero elements of the preconditioner in its block $ILU$ factorization and that of the original matrix. The numerical experiments were conducted on a Pentium III CPU at 550 MHZ with 64Mb RAM.

The right hand side was generated by assuming that the exact solution is a vector of all ones and the initial guess was a vector of some random numbers. The computations were stopped when the 2-norm of the residual was reduced by a factor of $10^{-8}$ or when 200 iterations were reached, indicated by (-).

We denote $M_S$ as the preconditioner defined by (4.5), where the solution of the reduced system

$$(5.1) \quad Sy = g - EB^{-1}f,$$

is performed using backward and forward solution steps by applying ILUT $(10^{-4}, 10)$ factorization [29]. And by $M_I$ and $M_C$ the preconditioners where the reduced system (5.1) is unpreconditioned and preconditioned by $C^{-1}$, respectively. The first set of test problems is a Laplace equation with Dirichlet boundary conditions.

$$(5.2) \quad -\Delta u = \text{rhs},$$

which is defined on the unit square.

The second set of test problem is a convection-diffusion equation with Dirichlet boundary conditions,

$$(5.3) \quad -\nu\Delta u + \nabla \cdot \nabla u = \text{rhs},$$
which is defined on the unit square. \( \mathbf{v} \) is a convective flow (velocity vector) and the viscosity parameter \( \nu \) governs the ratio between convection and diffusion and define the inverse of Reynolds number \( Re = 1/\nu \).

Looking at the results for the Laplace problem (Table 1), it turns out that the application of the preconditioner \( M_S \) offers a small number of iterations to achieve the convergence than \( M_I \) and \( M_C \), for both standard and block red-black orderings. Furthermore, \( M_I \) appears to be better with the standard red-black colorings than with the block version, that is not the case for the preconditioners \( M_C \) and \( M_S \). One can easily see that the number of iterations increases strongly with \( h^{-1} \) for both preconditioners \( M_I \) and \( M_C \) and weakly for \( M_S \).

Tables 2 and 3, 5 and 6 show the number of iterations for the convection-diffusion problem in two cases discretized by using standard and block colorings, preconditioned by three preconditioners, Case 2 was proposed by Ruge and Stuben [28] to test their algebraic multigrid method, this case has some semi-recirculation effect because the first of its convection coefficients vanishes at \( y = 0.5 \), while the first case has neither stagnation point nor semi-recirculation effect. We observe that for the preconditioner \( M_I \), the number is less with the standard ordering than with the block version, except when \((10^{-6} \leq \nu \leq 10^{-2}) \) and for \( h^{-1} = 33 \) and 65 in the first case, and when \((\nu = 10^{-2} \text{ for all mesh-sizes}) \) in the second case. For the preconditioner \( M_C \) and for the second case, the block version requires less number of iterations than the standard version, but for the first case we cannot deduce which ordering is better. For the preconditioner \( M_S \), we observe a moderate deterioration for \( 10^{-3} \leq \nu \leq 10^{-1} \) for both cases and both orderings. We show that with the standard coloring we obtain a number of iterations less than with the block version for both cases, except for \( h^{-1} = 65 \) and \( h^{-1} = 105 \) when \( \nu = 1 \) in the first case and \( 10^{-2} \leq \nu \leq 1 \) in the second one.

<table>
<thead>
<tr>
<th>( h^{-1} )</th>
<th>( M_I )</th>
<th>( M_C )</th>
<th>( M_S )</th>
<th>( M_I )</th>
<th>( M_C )</th>
<th>( M_S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>23</td>
<td>22</td>
<td>5</td>
<td>37</td>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>33</td>
<td>69</td>
<td>69</td>
<td>8</td>
<td>77</td>
<td>60</td>
<td>9</td>
</tr>
<tr>
<td>65</td>
<td>140</td>
<td>148</td>
<td>14</td>
<td>197</td>
<td>152</td>
<td>12</td>
</tr>
<tr>
<td>105</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>16</td>
</tr>
</tbody>
</table>

TABLE 1: Iterations to convergence for the Laplace problem by different preconditioners.
Case 1: $\mathbf{v} = \begin{pmatrix} \cos x \\ \sin y \end{pmatrix}$.

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>17</th>
<th>33</th>
<th>65</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>29</td>
<td>71</td>
<td>148</td>
<td>147</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>32</td>
<td>77</td>
<td>157</td>
<td>153</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>29</td>
<td>118</td>
<td>177</td>
<td>8</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>29</td>
<td>122</td>
<td>169</td>
<td>6</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>29</td>
<td>122</td>
<td>187</td>
<td>4</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>29</td>
<td>122</td>
<td>172</td>
<td>3</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>29</td>
<td>122</td>
<td>178</td>
<td>3</td>
</tr>
</tbody>
</table>

**TABLE 2:** Iterations to convergence for the convection-diffusion problem (Case 1) by different preconditioners using Standard version.

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>17</th>
<th>33</th>
<th>65</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>77</td>
<td>196</td>
<td>151</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>38</td>
<td>88</td>
<td>147</td>
<td>157</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>39</td>
<td>114</td>
<td>176</td>
<td>168</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>39</td>
<td>120</td>
<td>198</td>
<td>9</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>38</td>
<td>122</td>
<td>197</td>
<td>6</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>38</td>
<td>118</td>
<td>197</td>
<td>5</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>38</td>
<td>120</td>
<td>197</td>
<td>4</td>
</tr>
</tbody>
</table>

**TABLE 3:** Iterations to convergence for the convection-diffusion problem (Case 1) by different preconditioners using Block version.

From Tables 4 and 7, it is seen that with the standard red-black coloring $M_S$ requires less CPU time for the processing phase and less space memory than with the block version. We also see that the CPU time for the solution phase depends on the number of iterations required to achieve the convergence.
6 Conclusion

We have presented a technique to reorder the nodes of a graph adjacency to a sparse matrix, this technique is based on point red-black coloring, according to this ordering we have also presented a block ILU preconditioner with different approximations of the Schur complement matrix for solving sparse matrices. We have shown both from a theoretical and a practical point of view that the efficiency of the global preconditioner $M$ for the block original system depends strongly on how well $S$ approximates the Schur complement matrix $S$. From the

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>17</th>
<th>33</th>
<th>65</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>29</td>
<td>30</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>10^{-1}</td>
<td>41</td>
<td>36</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>10^{-2}</td>
<td>60</td>
<td>53</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>77</td>
<td>62</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>82</td>
<td>61</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>10^{-5}</td>
<td>83</td>
<td>60</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>83</td>
<td>60</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
TABLE 6: Iterations to convergence for the convection-diffusion problem (Case 2) preconditioned by different preconditioners using Block version.

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>17</th>
<th>33</th>
<th>65</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$</td>
<td>$M_I$</td>
<td>$M_C$</td>
<td>$M_S$</td>
<td>$M_I$</td>
</tr>
<tr>
<td>1</td>
<td>36 27 6 76 60 9</td>
<td>194 153 12 - - -</td>
<td>16 - - -</td>
<td></td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>41 30 6 98 60 9</td>
<td>163 158 13 - - -</td>
<td>16 - - -</td>
<td></td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>55 32 6 136 92 9</td>
<td>- - 12 - - -</td>
<td>18 - - -</td>
<td></td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>96 31 5 182 115 7</td>
<td>- 178 10 - - -</td>
<td>13 - - -</td>
<td></td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>99 19 5 - 116 6 - 8 - - -</td>
<td>10 - - -</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>98 21 5 - - 120 5 - 7 - - -</td>
<td>10 - - -</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>116 37 12 - - 120 14 - 7 - - -</td>
<td>9 - - -</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 7: Performance of the preconditioner $M_S$ for solving the convection-diffusion problem (Case 2).

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>Standard red-black coloring</th>
<th>Block red-black coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$</td>
<td>solu. prec. spar.</td>
<td>solu. prec. spar.</td>
</tr>
<tr>
<td>1</td>
<td>0.24 0.21 2.90</td>
<td>1.10 0.38 2.90</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>0.26 0.21 2.90</td>
<td>1.15 0.38 2.90</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>0.22 0.21 2.89</td>
<td>1.12 0.38 2.89</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>0.12 0.13 2.79</td>
<td>0.60 0.34 2.85</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>0.11 0.10 2.56</td>
<td>0.44 0.33 2.63</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>0.11 0.08 2.14</td>
<td>0.38 0.27 2.24</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>0.10 0.08 1.80</td>
<td>0.34 0.22 1.91</td>
</tr>
</tbody>
</table>

numerical experiments that we have conducted in this paper (Laplace and convection-diffusion equations), we have learned that the preconditioner $M_S$ is much better than $M_I$ and $M_C$ where the convergence is not reached for small $\nu$ and large problems (with small meshsize $h$) and leads to a considerably faster convergence. As can be seen by comparing Tables, we can solve more problems with the standard red-black ordering and $M_S$ preconditioner than with the block version, but the use of large blocks may be more efficient for some parallel architecture.

Acknowledgment The authors would like to thank the anonymous referee for their accurate remarks in improving the quality of this paper and for directing them to several of the references.
REFERENCES


Ecole Normale superieure, Bensouda B.P. 5206 Fes, Morocco
E-mail address: guessous_najib@hotmail.com

Departement de Mathematiques et Informatique Faculte des Sciences
Dhar-Mahraz B.P. 1796 Atlas Fes, Morocco
E-mail address: dsouhar@hotmail.com