Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

A multilevel Schur complement preconditioner with ILU factorization for complex symmetric matrices

Rainer Schlundt

submitted: November 26, 2018

Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: rainer.schlundt@wias-berlin.de

No. 2556
Berlin 2018

2010 *Mathematics Subject Classification.* 65F08, 65F15, 65N22, 65Y05.

Key words and phrases. Complex symmetric sparse linear system, Schur complement, multilevel preconditioner, incomplete LU factorization, Bunch-Kaufman pivoting, domain decomposition, low rank approximation.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

A multilevel Schur complement preconditioner with ILU factorization for complex symmetric matrices

Rainer Schlundt

Abstract

This paper describes a multilevel preconditioning technique for solving complex symmetric sparse linear systems. The coefficient matrix is first decoupled by domain decomposition and then an approximate inverse of the original matrix is computed level by level. This approximate inverse is based on low rank approximations of the local Schur complements. For this, a symmetric singular value decomposition of a complex symmetric matix is used. The block-diagonal matrices are decomposed by an incomplete LDL^T factorization with the Bunch-Kaufman pivoting method. Using the example of Maxwell's equations the generality of the approach is demonstrated.

1 Introduction

We consider iterative methods for solving large sparse systems

$$Ax = b, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$, $A = A^T$, $A \neq A^H$, $b \in \mathbb{C}^n$, and $x \in \mathbb{C}^n$. Krylov subspace methods combined with a preconditioner solve the above system (1). For example, left preconditioning consists of modifying the original system into the system $M^{-1}Ax = M^{-1}b$. The preconditioner M is an approximation to A. The solve of the preconditioned system is relatively inexpensive.

The domain decomposition (DD) approach decouples the original matrix A. We do not form the global Schur complement system and do not solve it exactly. Let A be partitioned in 2×2 block form as

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix}, \tag{2}$$

where $B \in \mathbb{C}^{m\times m}$, $C \in \mathbb{C}^{s\times s}$, $E \in \mathbb{C}^{m\times s}$, and $n = m + s$. We will receive the following basic block factorization of (2)

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix} = \begin{pmatrix} I & 0 \\ E^T B^{-1} & I \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & B^{-1}E \\ 0 & I \end{pmatrix}, \tag{3}$$

where $S \in \mathbb{C}^{s\times s}$, $S = C - E^T B^{-1} E$, is the Schur complement. Using

$$A^{-1} = \begin{pmatrix} I & -B^{-1}E \\ 0 & I \end{pmatrix} \begin{pmatrix} B^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -E^T B^{-1} & I \end{pmatrix}, \tag{4}$$

the original system (1) can be easily solved if S^{-1} is available. The goal is to approximate S^{-1} such that $S^{-1} \approx C^{-1} + LRA = \tilde{S}^{-1}$, where LRA stands for low rank approximation matrix. The preconditioner M then has the following form

$$M = \begin{pmatrix} I & 0 \\ E^T B^{-1} & I \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & \tilde{S} \end{pmatrix} \begin{pmatrix} I & B^{-1}E \\ 0 & I \end{pmatrix}. \tag{5}$$

DOI 10.20347/WIAS.PREPRINT.2556 Berlin 2018

We can write

$$S = C - E^T B^{-1} E = C^{1/2}\big(I - C^{-1/2} E^T B^{-1} E C^{-1/2}\big) C^{1/2} = C^{1/2}\big(I - G\big) C^{1/2} \tag{6}$$

and

$$S^{-1} = C^{-1/2}\big(I - G\big)^{-1} C^{-1/2} = C^{-1} + C^{-1/2} G \big(I - G\big)^{-1} C^{-1/2} \,. \tag{7}$$

The symmetric matrix $G \in \mathbb{C}^{s \times s}$ has a symmetric singular value decomposition (SSVD)

$$G = C^{-1/2} E^T B^{-1} E C^{-1/2} = W \Sigma W^T \,, \tag{8}$$

where W is a unitary matrix and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_s)$ with nonnegative σ_i (cf. [1]). Then LRA is an approximation of $C^{-1/2} G \big(I - G\big)^{-1} C^{-1/2}$. The remaining sections are organized as follows. Section 2 gives an overview of the domain decomposition framework and the multilevel preconditioning technique proposed in [10]. The incomplete LDL^T factorization with the Bunch-Kaufman pivoting is discribed in Section 3. Numerical experiments of a model problem are presented in Section 4. Implementation details for a real symmetric matrix A are described in [7, 8].

2 Domain decomposition and multilevel preconditioning

An interesting class of domain decomposition methods is the hierarchical interface decomposition (HID) ordering (cf. [2]). An HID ordering can be obtained from a standard graph partitioning (cf. METIS [4]). The reordered matrix has the following multilevel recursive form :

$$A_j = P_j C_{j-1} P_j^T = \begin{pmatrix} B_j & E_j \\ E_j^T & C_j \end{pmatrix} \quad \text{and} \quad C_0 \equiv A \quad \text{for} \quad j = 1, \ldots, lev \,. \tag{9}$$

P_j is a permutation matrix and lev the number of levels. Each block B_j in A_j has a block-diagonal structure resulting from this HID ordering. Analogous to (2), let A_j be partitioned at level j in block form as

$$A_j = \begin{pmatrix} B_j & E_j \\ E_j^T & C_j \end{pmatrix} = \begin{pmatrix} B_{j_1} & & & E_{j_1} \\ & \ddots & & \vdots \\ & & B_{j_p} & E_{j_p} \\ E_{j_1}^T & \ldots & E_{j_p}^T & C_j \end{pmatrix} \,, \tag{10}$$

where $B_j \in \mathbb{C}^{m_j \times m_j}$ is a block-diagonal matrix, $B_j = \operatorname{diag}(B_{j_1}, \ldots, B_{j_p})$, $C_j \in \mathbb{C}^{s_j \times s_j}$, $E_j \in \mathbb{C}^{m_j \times s_j}$, $E_j^T = (E_{j_1}^T, \ldots, E_{j_p}^T)$, $B_{j_i} \in \mathbb{C}^{m_{j_i} \times m_{j_i}}$, $E_{j_i} \in \mathbb{C}^{m_{j_i} \times s_j}$, $1 \le i \le p$, $n_j = m_j + s_j$, and $m_j = m_{j_1} + \cdots + m_{j_p}$. Analogous to (3), at each level j, the factorization of A_j is determined by

$$A_j = \begin{pmatrix} B_j & E_j \\ E_j^T & C_j \end{pmatrix} = \begin{pmatrix} I & 0 \\ E_j^T B_j^{-1} & I \end{pmatrix} \begin{pmatrix} B_j & 0 \\ 0 & S_j \end{pmatrix} \begin{pmatrix} I & B_j^{-1} E_j \\ 0 & I \end{pmatrix} \,, \tag{11}$$

where $S_j = C_j - E_j^T B_j^{-1} E_j$ is the Schur complement at level j. Thus

$$A_j^{-1} = \begin{pmatrix} I & -B_j^{-1} E_j \\ 0 & I \end{pmatrix} \begin{pmatrix} B_j^{-1} & 0 \\ 0 & S_j^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -E_j^T B_j^{-1} & I \end{pmatrix} \tag{12}$$

is the inverse of A_j. Analogous to (7), S_j^{-1} can be approximated by C_j^{-1} plus an approximation of $C_j^{-1/2} G_j \big(I - G_j\big)^{-1} C_j^{-1/2}$. The preconditioner M_j then has the following form

$$M_j = \begin{pmatrix} I & 0 \\ E_j^T B_j^{-1} & I \end{pmatrix} \begin{pmatrix} B_j & 0 \\ 0 & \tilde{S}_j \end{pmatrix} \begin{pmatrix} I & B_j^{-1} E_j \\ 0 & I \end{pmatrix} \tag{13}$$

and

$$C_j^{-1} = P_{j+1}^T M_{j+1}^{-1} P_{j+1}\,. \tag{14}$$

At each level j, the symmetric matrix $G_j \in \mathbb{C}^{s_j \times s_j}$ has a singular value decomposition (SVD)

$$G_j = C_j^{-1/2} E_j^T B_j^{-1} E_j C_j^{-1/2} = U_j \Sigma_j V_j^H\,, \tag{15}$$

where U_j and V_j are unitary matrices and $\Sigma_j = \mathrm{diag}(\sigma_{j_1}, \ldots, \sigma_{j_{s_j}})$ the singular values with real nonnegative σ_{j_i}. For the matrix G_j there exists a unitary matrix W_j such that

$$G_j = C_j^{-1/2} E_j^T B_j^{-1} E_j C_j^{-1/2} = W_j \Sigma_j W_j^T \tag{16}$$

is an SSVD. An SSVD of a symmetric matrix can be determined from its SVD. Therefore we have to modify the singular vectors corresponding to nonzero singular values (cf. [1]). The matrix $G_j\bigl(I - G_j\bigr)^{-1}$ results in

$$G_j\bigl(I - G_j\bigr)^{-1} = \quad W_j\bigl(I - \Sigma_j W_j^T W_j\bigr)^{-1}\Sigma_j W_j^T \tag{17}$$

and consequently

$$C_j^{-1/2} G_j\bigl(I - G_j\bigr)^{-1} C_j^{-1/2} = \quad Z_j\bigl(I - \Sigma_j Z_j^T C_j Z_j\bigr)^{-1}\Sigma_j Z_j^T\,, \quad Z_j = C_j^{-1/2} W_j. \tag{18}$$

Thus, the computation of a low rank approximation to $S_j^{-1} - C_j^{-1}$ (cf. (7), (18)) can be obtained by the following SSVD problem

$$C_j^{-1} E_j^T B_j^{-1} E_j C_j^{-1} = Z_j \Sigma_j Z_j^T. \tag{19}$$

Finally, the preconditioned system

$$M^{-1} A x = M^{-1} b \quad \text{with} \quad M^{-1} = C_0^{-1} = P_1^T M_1^{-1} P_1 \tag{20}$$

is to be solved.

3 Incomplete $\mathrm{LDL}^{\mathrm{T}}$ with Bunch-Kaufman pivoting

The matrices $B_{j_i}, 1 \le j \le lev, 1 \le i \le p$, (cf. (10)) are factored using a form of incomplete Cholesky factorization, so we have $B_{j_i} \approx L_{j_i} D_{j_i} L_{j_i}^T$. The core of the decomposition is a Crout variant of incomplete LU (ILU), introduced for symmetric matrices by [5], which itself extends works by [6] and [3]. The Crout-based decomposition is an attractive way for computing an incomplete LDL^T factorization for symmetric matrices, because it naturally preserves structural symmetry. This is especially true when applying dropping rules for the incomplete factorization. It is natural to store L_{j_i} by columns and to have the lower triangular part of B_{j_i} stored similary.
Let the matrix A ($A \leftarrow B_{j_i}$) be the sum of the matrices $\hat{L}$ and $\hat{D}$, that is, $A = \hat{L} + \hat{D} + \hat{L}^T$. $\hat{L}$ is the strict lower part of A und $\hat{D}$ the diagonal. Only the lower triangular part of A, ($\hat{L} + \hat{D}$), is in the Compressed Sparse Column (CSC) format stored. This is synonymous with the storage of ($\hat{D} + \hat{L}^T$) in CSR (Compressed Sparse Row) format. Algorithm 1 shows the Crout version of incomplete Cholesky factorization for a symmetrix matrix A, $A \approx LDL^T$, using a delayed update strategy for the factors. The k-th column update procedure is described in Algorithm 2. In the Bunch-Kaufman pivoting method, to find the next pivot only requires searching up to two columns in the reduced matrix. The two columns must be updated before proceeding with the search in the algorithm. Algorithm 3 descibes the Bunch-Kaufman pivoting strategy. In Algorithm 1 the $s \times s$ pivot is typically 1×1 or 2×2. The following dropping rule is used. Only the largest nonzero entries in every column are kept. The pre-specified maximum number of fill-ins per column is a multiple of the average number of nonzero elements per column in the original matrix. A bi-index data structure was used to address implementation difficulties in sparse matrix operations (cf. [3, 5, 6]).

Algorithm 1 Crout version of incomplete LDL^T factorization

Input: Symmetric matrix A, matrix size n
Output: Matrices P, L , and D, such that $PAP^T \approx LDL^T$

procedure ILDLC
  $k = 1$
  while $k \le n$ **do**
    Find a $s \times s$ pivot in $A_{k:n,k:n}$, $s \in \{1, 2\}$ ▷ call BKPIVOT
    Apply dropping rules to $w_{k+s:n,1:s}$
    $L_{k+s:n,k:k+s-1} = w_{k+s:n,1:s} D^{-1}_{k:k+s-1,k:k+s-1}$ ▷ $w_{1:k+s-1,1:s} = 0$
    $L_{k:k+s-1,k:k+s-1} = I$
    for $i = k + s, \ldots, n$ **do**
      if $\{L_{i,k} \neq 0 \vee L_{i,k+1} \neq 0\}$ **then**
        $A_{i,i} = A_{i,i} - L_{i,k:k+s-1} D_{k:k+s-1,k:k+s-1} L^T_{i,k:k+s-1}$
      end if
    end for
    $k = k + s$
  end while
end procedure

Algorithm 2 k-th column update procedure

Input: Column vector w, partial factors L and D, matrix size n, current column index k
Output: Updated column w

procedure UPDATE
  $i = 1$
  while $i < k$ **do**
    $s \leftarrow$ size of the diagonal block with $D_{i,i}$ as it top left corner
    if $\{L_{k,i} \neq 0 \vee L_{k,i+1} \neq 0\}$ **then**
      $w_{k+1:n} = w_{k+1:n} - L_{k+1:n,i:i+s-1} D_{i:i+s-1,i:i+s-1} L^T_{k,i:i+s-1}$ ▷ $w_{1:k} = 0$
    end if
    $i = i + s$
  end while
end procedure

Algorithm 3 Bunch-Kaufman pivoting method at step k

Input: Symmetric matrix A at step k, partial factors L and D, matrix size n, current column index k
Output: Symmetric updated matrix A, $s \times s$ pivot, column vectors $w_{1:s}$

```
procedure BKPIVOT
    α = (1 + √17)/8
    Load and update A_{k+1:n,k}:   w_{1:k,1} = 0, w_{k+1:n,1} = A_{k+1:n,k}          ▷ call UPDATE
    Let λ = ‖w_{k+1:n,1}‖_∞ = max_{k+1≤j≤n} |w_{j,1}| and w_{l,1} = λ               ▷ l smallest integer
    if |A_{k,k}| ≥ αλ then
        Use D_{k,k} = A_{k,k} as a 1 × 1 pivot
        s = 1
    else
        Load and update A_{k+1:n,l}:   w_{1:k,2} = 0, w_{k+1:n,2} = (A_{l,k:l−1}, A^T_{l+1:n,l})^T   ▷ call UPDATE
        Let σ = max_{k+1≤j≤n} |w_{j,2}|
        if |A_{k,k}|σ ≥ αλ^2 then
            Use D_{k,k} = A_{k,k} as a 1 × 1 pivot
            s = 1
        else if |A_{l,l}| ≥ ασ then
            Use D_{k,k} = A_{l,l} as a 1 × 1 pivot
            w_{k+1:n,1} = w_{k+1:n,2}
            s = 1                          ▷ interchange the k-th and the l-th rows and columns
        else
            Use ( D_{k,k}    D_{k,k+1}   ) = ( A_{k,k}  λ       ) as a 2 × 2 pivot
                ( D_{k+1,k}  D_{k+1,k+1} )   ( λ        A_{l,l} )
            s = 2                          ▷ interchange the (k + 1)-th and the l-th rows and columns
        end if
    end if
end procedure
```

4 Numerical experiments

Using the example of Maxwell's equations we demonstrate the generality of the approach. We obtain in vector notation the following equations in integral form:

$$\oint_P \vec{E} \cdot \mathrm{d}\vec{l} = -\frac{\partial}{\partial t} \iint_A \vec{B} \cdot \mathrm{d}\vec{A} \qquad \oint_P \vec{H} \cdot \mathrm{d}\vec{l} = \frac{\partial}{\partial t} \iint_A \vec{D} \cdot \mathrm{d}\vec{A} + \iint_A \vec{J} \cdot \mathrm{d}\vec{A}$$
$$\oiint_S \vec{B} \cdot \mathrm{d}\vec{S} = 0 \qquad \oiint_S \vec{D} \cdot \mathrm{d}\vec{S} = \iiint_V q \,\mathrm{d}V \,. \tag{21}$$

The constitutive relations belonging to them are

$$\vec{D} = \varepsilon \vec{E} \,, \qquad \vec{B} = \mu \vec{H} \,, \qquad \vec{J} = \kappa \vec{E} \,. \tag{22}$$

Here, A is a surface with boundary curve P, V is a volume bounded by a surface S, and q is the volume charge density. An orthogonal dual mesh is used to discretize the Maxwell's equations using the Finite Integration Technique (FIT, [13, 14, 9]). The electric and magnetic voltages and fluxes over elemtary objects are defined as state variables in the following way:

$$\begin{aligned}
e_i &= \int_{L_i} \vec{E} \cdot \mathrm{d}\vec{l} & h_j &= \int_{\tilde{L}_j} \vec{H} \cdot \mathrm{d}\vec{l} & i &= 1, \ldots, n_e \\
d_i &= \iint_{\tilde{A}_i} \vec{D} \cdot \vec{n} \,\mathrm{d}\vec{A} & b_j &= \iint_{A_j} \vec{B} \cdot \vec{n} \,\mathrm{d}\vec{A} & j &= 1, \ldots, n_f \\
j_i &= \iint_{\tilde{A}_i} \vec{J} \cdot \vec{n} \,\mathrm{d}\vec{A} & q_k &= \iiint_{\tilde{V}_k} q \,\mathrm{d}V & k &= 1, \ldots, n_p \,.
\end{aligned}$$

where $\vec{n}$ is the outward-pointing normal of the faces A_j and $\tilde{A}_i$, respectively. If all field quantities vary sinusoidally with time, the coefficient matrices of the corresponding linear systems of equation are complex, symmetric, and indefinite. Using Krylov subspace methods, (20) can be solved iteratively (cf. [11, 12]).

We consider different dimensions of the coefficient matrices of the corresponding systems of linear equations, in fact $n = 16\,632$, $n = 40\,824$, $n = 472\,416$, and $n = 4\,020\,192$. At each level j, $j = 1, 2, \ldots$, the matrix A is partitioned into p, $p \in \{0, 2, 3, 5, 10, 15\}$, non-overlapping subsets $B_{j_i}, i = 1, \ldots, p$. At each p, $p \in \{2, 3, 5, 10, 15\}$, we compute the k, $k \in \{5, 10, 15, 20\}$, largest singular values and the corresponding singular vectors to obtain a low rank approximation. For $p = 0$ the solution is computed by [11, 12]. The same applies on the one hand for the computation of the solution with the coefficient matrices B_{j_i} and on the other hand we comput an incomplete LDL^T factorization of B_{j_i} (cf. Section 3). The testing platform consists of Intel Xeon W3520 processors with 2.67 GHz.

The following notation is used throughout the section:

- icf: "T" indicates that the matrices B_{j_i} were formed by an incomplete LDL^T factorization, on the other hand "F"
- its: number of iterations of preconditioned solver to reduce the initial residual by a factor of 10^{-8}

- s-t: wall clock time for the iteration phase of the solver in seconds
- p-t: s-t plus wall clock time to build the preconditioner in seconds
- "-" indicates that the preconditioner is not created

From Tables 1-4, we find the number of iterations and the wall clock times for the different dimensions. In Table 5, we find the corresponding dimensions of the matrices B_{j_i} and C_j for $j = 1, 2, \ldots$ and $i = 1, \ldots, p$ of the dimension $n = 4\,020\,192$. For the other dimensions, i.e., $n = 16\,632$, $n = 40\,824$, and $n = 472\,416$, the corresponding informations can be found in Tables 4-6 of [10]. The informations from the Tables 1-4 are shown graphically in the Figures 1-4. The red line indicates the values for $p = 0$.

The following notation is used for the figures:

- (a): number of iterations (its, icf = "F")
- (b): wall clock time for the iteration phase (s-t, icf = "F")
- (c): the proportion of wall clock time s-t (coloured) in the total time p-t (icf = "F")
- (d): number of iterations (its, icf = "T")
- (e): wall clock time for the iteration phase (s-t, icf = "T")
- (f): the proportion of wall clock time s-t (coloured) in the total time p-t (icf = "T")

With the exception of $n = 4\,020\,192$, it can be seen that the lowest iteration numbers in the iteration process have been achieved for itc = "F", $p = 2$, and $k \in \{5, 10, 15, 20\}$. For small dimensions, also $p = 3$ is useful. This process is very time consuming. In general, the iteration numbers for icf = "T" are greater than those for icf = "F" and smaller than those for $p = 0$. They are comparable for $n = 16\,632$. For high-dimensional problems the computation of the incomplete LDL^T factorization of the matrices B_{j_i} is also time consuming. Likewise, the computation with B_{j_i} for greater $k \in \{5, 10, 15, 20\}$ becomes more time consuming. Experimental results indicate that this preconditioner based on Schur complement approach is robust in the iteration phase.

5 Conclusions

This paper presents a preconditioning method based on a Schur complement approach with low rank approximations for solving complex symmetric sparse linear systems. It tries to approximate the inverse of the Schur complement by exploiting low rank approximations. For this, a hierarchical graph decomposition reorders the matrix into a multilevel block form. On the negative side, building this preconditioner can be time consuming. A solve with the matrix B_j amounts to p local and independent solves with the matrices B_{j_i}, $i = 1, \ldots, p$. These can be carried out by a preconditioned Krylov subspace iteration and by an incomplete LDL^T factorization with Bunch-Kaufman pivoting, respectively. A big part of the computations to build a preconditioner based on Schur complement approach is attractive for massively parallel machines. This also applies to the application of the preconditioner in the iteration process.

Table 1: The number of iterations and the wall clock times for $n = 16\,632$.

number of subsets		k largest singular values							
		5		10		15		20	
p	icf	F	T	F	T	F	T	F	T
0		its = 157 p-t = 0.238							
2	its	128	123	124	121	126	121	131	136
	s-t	23.584	0.569	22.653	0.612	23.037	0.709	24.028	0.988
	p-t	44.441	1.743	37.777	1.619	36.599	1.679	42.278	2.426
3	its	142	126	147	126	142	127	143	131
	s-t	22.424	0.616	23.032	0.748	22.273	0.917	22.859	1.152
	p-t	45.940	1.583	37.425	1.692	36.136	1.806	41.737	2.431
5	its	146	142	143	141	143	141	143	140
	s-t	19.490	0.735	19.209	0.920	19.483	1.160	19.904	1.435
	p-t	23.290	1.134	27.267	1.539	33.953	1.981	35.059	2.605
10	its	182	177	187	176	187	182	180	177
	s-t	20.314	1.282	21.171	1.776	21.393	2.575	21.286	3.195
	p-t	29.963	2.215	33.603	2.954	34.703	4.156	39.078	5.523
15	its	211	208	208	204	217	215	214	215
	s-t	21.885	1.701	21.806	2.545	23.964	3.946	24.767	5.157
	p-t	34.714	3.053	34.976	4.241	41.654	6.940	48.224	9.660

Table 2: The number of iterations and the wall clock times for $n = 40\,824$.

number of subsets		k largest singular values							
		5		10		15		20	
p	icf	F	T	F	T	F	T	F	T
0		its = 510 p-t = 1.609							
2	its	172	401	168	410	173	493	167	509
	s-t	200.5	4.892	196.0	5.479	203.3	7.647	194.4	8.883
	p-t	238.5	9.180	252.8	9.972	271.8	12.627	238.7	14.216
3	its	182	389	173	385	176	387	181	379
	s-t	157.6	5.111	151.2	5.777	152.1	6.831	158.2	7.900
	p-t	193.9	7.960	191.5	8.760	112.5	10.127	229.5	11.392
5	its	294	398	283	380	285	371	275	371
	s-t	144.2	5.661	139.3	6.823	142.6	8.334	138.6	10.426
	p-t	166.0	7.608	168.0	8.981	181.6	10.957	189.7	13.704
10	its	347	430	341	417	330	392	298	381
	s-t	133.5	6.752	132.5	8.906	132.9	11.078	120.9	14.027
	p-t	153.9	8.254	165.1	11.085	174.2	13.985	172.2	18.149
15	its	382	465	366	431	362	431	368	416
	s-t	127.6	8.015	124.6	10.498	126.2	14.383	131.2	18.195
	p-t	151.0	9.505	161.6	13.157	167.4	18.256	184.8	23.587

Table 3: The number of iterations and the wall clock times for $n = 472\,416$.

number of subsets		k largest singular values							
		5		10		15		20	
p	icf	F	T	F	T	F	T	F	T
0		its = 918 p-t = 32.714							
2	its	344	943	310	909	262	945	270	921
	s-t	6930.6	157.3	6237.5	156.2	5297.6	163.8	5451.5	166.0
	p-t	8203.2	1054.6	7699.3	1038.6	7011.9	1063.8	7559.3	1086.7
3	its	537	814	532	817	550	958	530	936
	s-t	6420.6	122.6	6416.1	141.9	6709.4	179.1	6473.3	191.7
	p-t	7572.4	635.5	7527.8	652.0	7809.3	679.4	7783.3	694.6
5	its	596	851	614	1095	616	1035	617	1062
	s-t	4495.3	129.9	4620.4	204.0	4705.8	217.0	4666.6	252.2
	p-t	5278.7	368.8	5401.6	451.6	5539.9	457.9	5550.6	495.4
10	its	716	870	758	891	719	873	763	874
	s-t	3418.5	138.8	3718.3	193.5	3595.5	225.7	3888.7	260.8
	p-t	3898.7	241.3	4463.7	318.1	4224.1	343.6	4865.9	383.8
15	its	827	947	820	996	831	979	991	987
	s-t	3228.0	162.3	3231.7	234.4	3394.0	285.0	4084.3	353.9
	p-t	3755.0	255.6	4043.0	329.5	4246.6	387.0	5028.2	461.5

Table 4: The number of iterations and the wall clock times for $n = 4\,020\,192$.

number of subsets		k largest singular values							
		5		10		15		20	
p	icf	F	T	F	T	F	T	F	T
0		its = 14 659 p-t = 5 295							
2	its	-	9 611	-	9 560	-	9 805	-	9 847
	s-t	-	15 159	-	15 602	-	15 626	-	16 033
	p-t	-	79 548	-	80 733	-	80 279	-	80 785
3	its	-	6 943	-	6 624	-	6 794	-	7 004
	s-t	-	10 434	-	10 075	-	10 784	-	11 507
	p-t	-	52 502	-	52 472	-	53 262	-	53 961
5	its	-	8 993	-	9 250	-	9 059	-	9 491
	s-t	-	13 813	-	13 654	-	15 355	-	16 574
	p-t	-	38 719	-	38 349	-	40 175	-	41 328
10	its	-	10 705	-	11 115	-	10 668	-	10 865
	s-t	-	16 029	-	18 810	-	19 803	-	22 051
	p-t	-	27 911	-	30 667	-	31 716	-	33 899
15	its	-	13 164	-	13 388	-	11 992	-	12 218
	s-t	-	20 679	-	23 568	-	21 998	-	26 554
	p-t	-	27 926	-	30 894	-	29 273	-	33 868

Table 5: The dimensions of the matrices B_{j_i} and C_j for $n = 4\,020\,192$.

p	level j	$\dim(B_{j_i})$					$\dim(C_j)$
2	1	1 995 946	1 995 891				28 355
	2	13 976	13 978				401
	3	193	194				14
3	1	1 326 396	1 328 109	1 314 339			51 348
	2	17 021	16 946	16 847			534
	3	172	174	175			13
5	1	783 419	788 212	793 646	778 816	778 951	97 1301
	2	19 355	19 309	19 301	19 153	19 231	691
	3	138	134	134	131	135	19
10	1	390 394	388 218	388 765	382 473	375 916	186 772
		380 469	381 039	386 203	385 166	374 777	
	2	18 041	18 204	18 119	17 744	18 003	5426
		18 494	18 253	18 255	18 061	18 172	
	3	537	542	537	534	535	56
		539	536	543	532	535	
15	1	251 999	242 253	250 967	253 231	253 313	230 150
		256 837	245 764	254 114	268 013	251 458	
		248 194	252 735	251 524	254 254	255 386	
	2	14 788	14 598	14 770	14 767	15 079	7249
		15 223	15 147	14 897	14 640	14 571	
		15 062	14 915	14 939	14 890	14 615	
	3	482	478	479	473	477	74
		485	480	477	477	471	
		469	480	480	482	485	

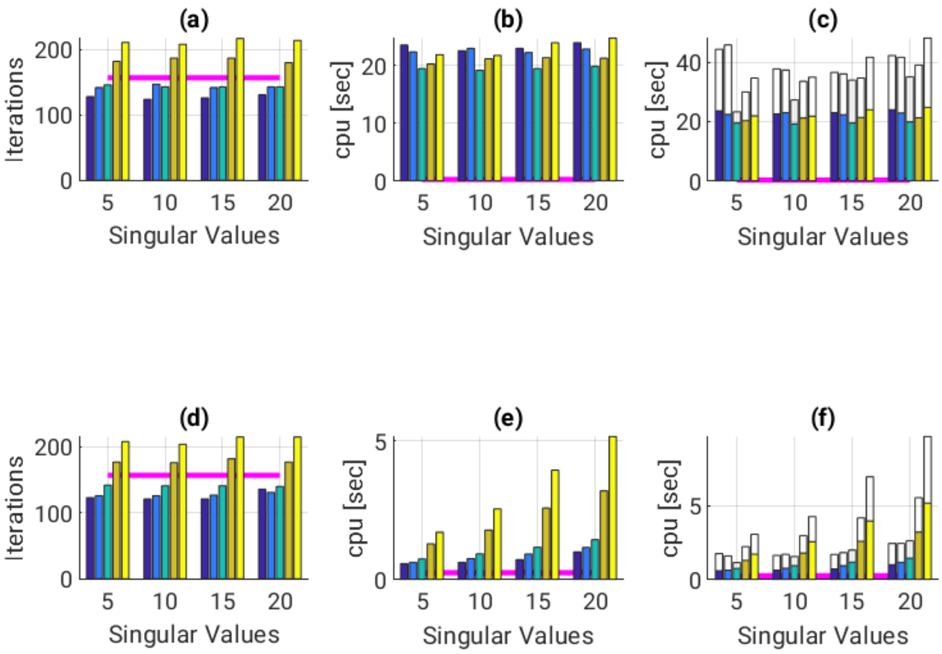

Figure 1: The number of iterations and wall clock times for $n = 16\,632$

DOI 10.20347/WIAS.PREPRINT.2556

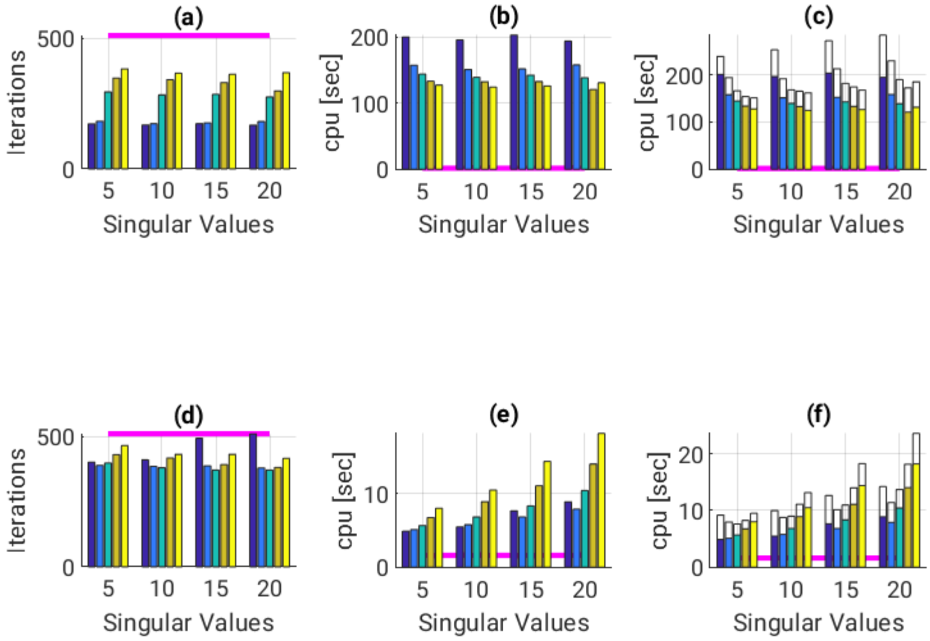

Figure 2: The number of iterations and wall clock times for $n = 40\,824$

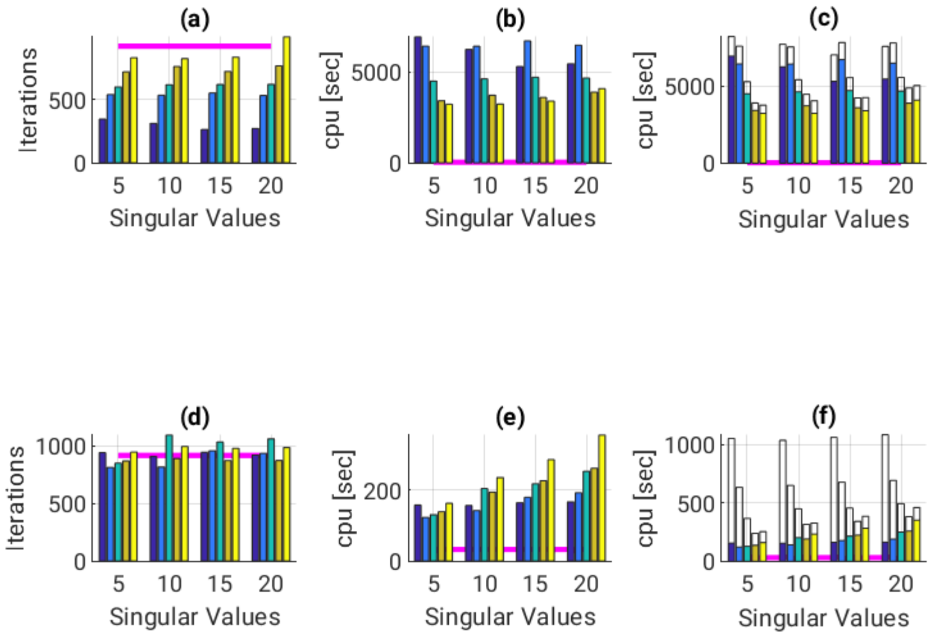

Figure 3: The number of iterations and wall clock times for $n = 472\,416$

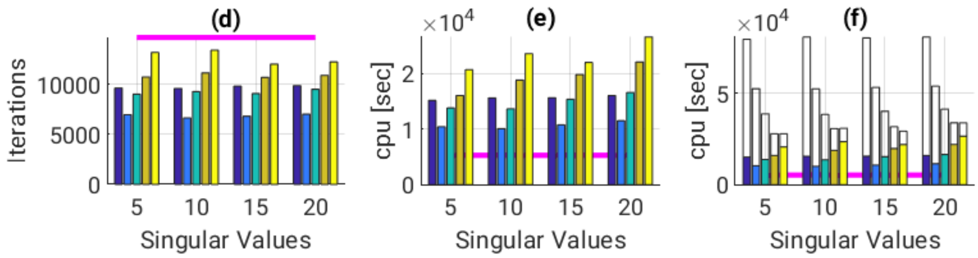

Figure 4: The number of iterations and wall clock times for $n = 4\,020\,192$

References

[1] Angelika Bunse-Gerstner and William Gragg. Singular value decomposition of complex symmetric matrices. *Journal of Computational and Applied Mathematics*, 21:41–54, 1988.

[2] Pascal Henon and Yousef Saad. A parallel multistage ILU factorization based on a hierrarchical graph decomposition. *SIAM J. Sci. Comput.*, 28(6):2266–2293, 2006.

[3] Mark T. Jones and Paul E. Plassmann. An improved incomplete Cholesky factorization. *ACM Trans. Math. Software*, 21(1):5–17, 1995.

[4] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[5] Na Li and Yousef Saad. Crout versions of ILU factorization with pivoting for sparse symmetric matrices. *Electron. Trans. Numer. Anal.*, 20:75–85, 2005.

[6] Na Li, Yousef Saad, and Edmond Chow. Crout versions of ILU for general sparse matrices. *SIAM J. Sci. Comput.*, 25(2):716–728, 2003.

[7] Ruipeng Li and Yousef Saad. Low-rank correction methods for algebraic domain decomposition preconditioners. Technical Report ys-2014-5, Dept. Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 2014.

[8] Ruipeng Li, Yuanzhe Xi, and Yousef Saad. Schur complement based domain decomposition preconditioners with low-rank corrections. Technical Report ys-2014-3, Dept. Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 2014.

[9] Rainer Schlundt. Regular triangulation and power diagrams for Maxwell's equations. WIAS Preprint No. 2017, Weierstraß-Institut für Angewandte Analysis und Stochastik, 2014.

[10] Rainer Schlundt. A multilevel Schur complement preconditioner for complex symmetric matrices. WIAS Preprint No. 2452, Weierstraß-Institut für Angewandte Analysis und Stochastik, 2017.

[11] Rainer Schlundt, Franz-Josef Schückle, and Wolfgang Heinrich. Shifted linear systems in electromagnetics. Part I: Systems with identical right-hand sides. WIAS Preprint No. 1420, Weierstraß-Institut für Angewandte Analysis und Stochastik, 2009.

[12] Rainer Schlundt, Franz-Josef Schückle, and Wolfgang Heinrich. Shifted linear systems in electromagnetics. Part II: Systems with multiple right-hand sides. WIAS Preprint No. 1646, Weierstraß-Institut für Angewandte Analysis und Stochastik, 2011.

[13] Thomas Weiland. A discretization method for the solution of Maxwell's equations for six-component fields. *Electronics and Communication (AEÜ)*, 31:116–120, 1977.

[14] Thomas Weiland. On the unique numerical solution of Maxwellian eigenvalue problems in three dimensions. *Particle Accelerators (PAC)*, 17:277–242, 1985.