The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-–9 June 2022, Oslo, Norway

# A HYBRID FACTORIZATION ALGORITHM FOR SPARSE MATRIX WITH MIXED PRECISION ARITHMETIC

**ATSUSHI SUZUKI**[1]

[1] RIKEN Center for Computational Science
Minatojima-minami-machi, Chuo, Kube Hyogo 650-0047, Japan e-mail:
Atsushi.Suzuki.aj@a.riken.jp

**Key words:** Sparse direct solver, mixed precision arithmetic, block Krylov subspace method

**Abstract.** A new hybrid algorithm for $LDU$-factorization for large sparse matrix combining iterative solver, which can keep the same accuracy as the classical factorization, is proposed. The last Schur complement will be generated by iterative solver for multiple right-hand sides using block GCR method with the factorization in lower precision as a preconditioner, which achieves mixed precision arithmetic, and then the Schur complement will be factorized in higher precision. In this algorithm, essential procedure is decomposition of the matrix into a union of moderate and hard parts, which is realized by $LDU$-factorization in lower precision with symmetric pivoting and threshold postponing technique.

## 1 INTRODUCTION

There are two kinds of demand in finding a solution of linear system with large sparse matrix in numerical simulation by using mixed precision arithmetic. One is for solving the system with very high condition number in numerical simulation of complex physical model and/or with large variety of physical coefficients. In this case, a monolithic direct factorization solver using "quadruple" precision could be only feasible tool. However arithmetic complexity by "double-double" data structure, which is a faster implementation of "quadruple" arithmetic using modern hardware named as fused multiply-add unit, is 25 times higher than "double". Hence, it is necessary to introduce mixed precision arithmetic with "double" to reduce computational complexity. The other is to solve rather moderate problem on forthcoming CPU with more single precision units than double.

Usages of mixed precision arithmetic in numerical linear algebra are hot research topics and a survey paper [1] covers the recent developments. The main tool to improve or recover the accuracy of the solution obtained by lower precision either direct solver or iterative solver is the iterative refinement, which generates new right-hand side from the residual to improve the solution. However convergence of the refinement process depends on the condition number and it is still not easy to improve the solution for matrix with high condition number. For some singular matrix whose condition number on the image is moderate, accurate factorization is mandatory especially to perform rank-revealing. Our aim is to construct factorization itself, which fits to usage of mixed precision arithmetic, not to improve the accuracy of the solution.

In section 2, classical $LDU$-factorization with symmetric pivoting with threshold postponing is viewed and a way to decomposition of the matrix into a union of moderate and hard parts

is proposed. Section 3 describes a novel method to generate Schur complement matrix of the hard part by solving linear system of the moderate part with multiple right-hand sides. Section 4 verifies efficiency of the proposed algorithm in accuracy and computing time by numerical examples.

## 2 FACTORIZATION WITH SYMMETRIC PIVOTING FOR LARGE SPARSE MATRICES

Let us assume that the matrix $K \in \mathbb{R}^{L\times L}$ is scaled so that diagonal entries take one of $-1$, 0 and 1, which could be realized by a scaling with only diagonal entries $[Q]_{ii} = 1/\sqrt{[\overline{K}]_{ii}}$ when $[\overline{K}]_{ii} \neq 0$, otherwise $[Q]_{ii} = 1$. Here the original matrix $\overline{K}$ is scaled as $Q\overline{K}Q = K$.

### 2.1 factorization with symmetric pivoting and recursive generation of Schur complement

The $LDU$-factorization of the matrix $K$ consists of recursive generation of the Schur complement,

$$\Pi^T \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \Pi = \Pi^T \begin{bmatrix} K_{11} & 0 \\ K_{21} & S_{22} \end{bmatrix} \begin{bmatrix} I_1 & K_{11}^{-1}K_{12} \\ 0 & I_2 \end{bmatrix} \Pi\,, \tag{1}$$

$$S_{22} = K_{22} - K_{21}K_{11}^{-1}K_{12}\,. \tag{2}$$

Here pivoting strategy is a symmetric one that is expressed by using the permutation $\Pi$ with $\Pi^T = \Pi^{-1}$ and $\Pi$ is decomposed into a union of diagonal blocks as $\Pi = \mathrm{diag}\{\Pi_1\,, \Pi_2\}$. Indices $\Lambda = \{1, \cdots, L\}$ is decomposed into a direct sum of $\overline{\Lambda}_1 \oplus \overline{\Lambda}_2$, where $\Pi_i$ is a one to one operation on $\overline{\Lambda}_i$. $\overline{\Lambda}_1$ denotes indices of already factorized part of the matrix and further factorization will be performed for $S_{22}$ on $\Lambda_2$. Each entry of $\Pi$ is selected during the factorization. Let us suppose that $k\times k$ sub-matrix $K_{11}$ is already factorized and $\{\Pi(1), \cdots, \Pi(k)\}$ are obtained and in the rest of matrix $(L-k)\times(L-k)$ sub-matrix $K_{22}$ needs to be factorized. We find the maximum absolute value in the diagonal entries of $K_{22}$, label $k_0$ as the index of such entry, define $\Pi(k+1) = k_0$, and set $d = [K]_{k_0\,k_0}$. The rows and columns of $K$ with indices $k$ and $k_0$ are exchanged each other and the result is stored as $\widetilde{K}$. Schur complement matrix $S'_{22}$, whose size is $(L-k-1)\times(L-k-1)$, is calculated by rank-1 update with weight $1/d$, $S'_{22} = \widetilde{K}'_{22} - [\widetilde{K}_{22}]_{\downarrow 1} d^{-1} [\widetilde{K}_{22}]_{1\,\rightarrow}$. This is the essential operation of $LDU$-factorization.

In practical computation of large sparse matrix, $\Lambda_1$ will be created by following the elimination tree of the nested-dissection ordering with threshold postponing with user defined parameter $\tau$.

### 2.2 nested-dissection ordering and threshold postponing

Let us introduce the nested-dissection ordering [2] and suppose that $\Lambda$ is decomposed into $2^m - 1$ sub-indices with $m$-level bi-section tree, $\Lambda = \bigoplus_{1\le\ell\le 2^m-1} \Lambda_\ell$, where $k$-th level contains $2^{k-1}$ sub-index sets $\{\Lambda_\ell\}$. For stability of the factorization, a given threshold parameter $\tau > 0$ is introduced to perform postponing of factorization. During the $LDU$-factorization of the sub-matrix with index $\Lambda_j$, if the ratio in absolute value of successive diagonal entries becomes smaller than $\tau$, i.e., $|[K]_{i+1\,i+1}/[K]_{ii}| < \tau$, then the lower block of the matrix is not factorized. The index will be decomposed as $\Lambda_j = \{j_1, \cdots, j_i\} \oplus \{j_{i+1}, \cdots, j_M\} = \widetilde{\Lambda_j} \oplus \widehat{\Lambda_j}$, where $\widetilde{\Lambda_j}$ is set of the

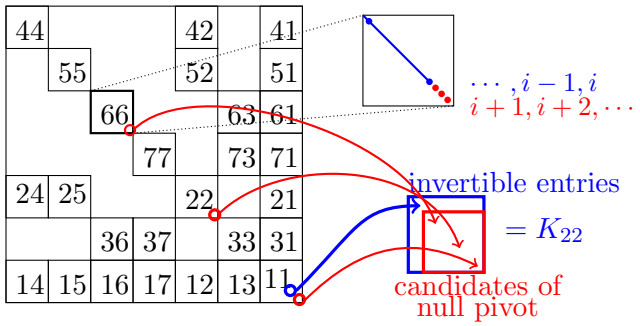


**Figure 1**: symmetric pivoting with threshold postponing following nested-dissection ordering

indices for the factorized part. For $m$-level bisection tree of the nested-dissection ordering, an $LDU$-factorizable part with $1 \times 1$ pivot is collected as $\bigoplus_{1 \le \ell \le 2^m - 1} \widetilde{\Lambda_\ell}$ and $\Lambda_0 = \bigoplus_{1 \le \ell \le 2^m - 1} \widehat{\Lambda_\ell}$ for the postponed entries. At the end of all threshold factorization following the elimination tree, we will again apply the $LDU$-factorization to the last Schur complement with indices $\Lambda_0$ and will obtain $\Lambda_0 = \widetilde{\Lambda_0} \oplus \widehat{\Lambda_0}$. Here we will enlarge $\widehat{\Lambda_0}$ with $n$ entries by moving the last entries of $\widetilde{\Lambda_0}$ to ensure $S_{22}$ has an image space, which contributes to comparison between zero eigenvalues and nonzero ones. Usually we take $n = 4$ entries [3]. By this process, the very last Schur complement matrix $K_{22}$ in (1) collecting postponed pivots $\Lambda_2 := \bigoplus_{0 \le \ell \le 2^m - 1} \widehat{\Lambda_\ell}$ will have large condition number or singular for the case that the original matrix is not invertible, and may contain $2 \times 2$ pivoting entries when $K_{11}$ is not definite. On the contrary, $K_{11}$ in (1) with index $\Lambda_1 := \bigoplus_{0 \le \ell \le 2^m - 1} \widehat{\Lambda_\ell}$ has moderate condition number that can be factorized with appropriate permutation. Figure 1 shows schematic explanation of the threshold postponing with 3-level bisection tree in nested-dissection ordering.

## 3 HYBRID FACTORIZATION ALGORITHM

A new algorithm is constructed by replacing the solution of the linear system with multiple right-hand sides (RHSs), $K_{11}X_{12} = K_{12}$ of the first block in (1) by an iterative solver. Factorization for $K \in \mathbb{R}^{L \times L}$ with user-defined threshold $\tau$ is performed in a hybrid way as follows.

**Algorithm 1** *LDU-factorization with internal iterative solver*
1. factorize $K$ with threshold postponing in lower precision and extract moderate part $K_{11}$ with finding indices $\Lambda_1$ and permutation $\Pi_1$ with $N = \#\Lambda_1$ .
2. decompose matrix $K$ into $2 \times 2$ blocks as $\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$ with $K_{11} \in \mathbb{R}^{N \times N}$, $K_{12} \in \mathbb{R}^{N \times M}$, $K_{21} \in \mathbb{R}^{M \times N}$, and $K_{22} \in \mathbb{R}^{M \times M}$ .
3. find solution $X_{12}$ satisfying $K_{11}X_{12} = K_{12}$ by an iterative solver using the $LDU$-factorization of $K_{11}$ with permutation $\Pi_1$ in lower precision as preconditioner .
4. construct the Schur complement $S_{22} := K_{22} - K_{21}X_{12}$ in higher precision .
5. factorize $S_{22}$ in higher precision with finding a symmetric pivoting expressed by the permutation $\Pi_2$ that may contain $2 \times 2$ entries .

We can utilize this solution in lower precision as a preconditioner for the iterative solver in higher precision.

If the condition number of $K_{11}$ is in the range of the maximum floating digits of the lower precision, the solution $X_{12}$ of $K_{11}X_{12} = K_{12}$ will be obtained very accurately with the residual closed to the machine epsilon of the higher precision. Therefore, even for the case that the original matrix is singular, the Schur complement $K_{22}$ is well constructed without large perturbation during the operation of $K_{11}^{-1}$ and kernel detection for rank-reveling [3] works well as the original full factorization algorithm. Solution procedure with forward and backward substitutions are performed block-wisely for the linear system after applied permutations $\Pi = \mathrm{diag}[\Pi_1, \Pi_2]$. The following algorithm describes a procedure to find $[x_1^T, x_2^T]^T$ satisfying

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} . \tag{3}$$

**Algorithm 2** *forward/backward substitutions for hybrid factorization*
1. solve $K_{11}\, y_1 = b_1$ by the same iterative solver in Algorithm 1 in higher precision .
2. compute $y_2 = b_2 - K_{21} y_1$ .
3. solve $S_{22}\, x_2 = y_2$ by forward and backward substitutions in higher precision .
4. update $x_1 = y_1 - X_{12} x_2$ with $X_{12}$ is computed in Algorithm 1 .

Solution of the moderate part of the liner system $K_{11}y_1 = b_1$ is obtained by the iterative solver within given accuracy to specify the convergence. Thanks to preconditioner by the *LDU*-factorization in lower precision for $K_{11}$ with moderate condition number, we can expect enough accuracy of the solution by the preconditioned iterative solver closed to one by the direct solver.

### 3.1 Iterative solver for solution of linear system with multiple RHSs

There are two kinds of solver for the system $K_{11}X_{12} = K_{12}$. For simplicity, we first describe algorithms for linear system with single RHS, $A\, x = b$ where $A \in \mathbb{R}^{N\times N}$ is invertible and it stands for $K_{11}$ and $b \in \mathbb{R}^N$ will be one of the column vector of $K_{12}$. Let us denote $\widehat{Q}\widehat{x} = \widehat{b}$ as the linear system in lower precision with solution $\widehat{x}$ for the RHS, $\widehat{b}$, which is converted from the given data $b$ in higher precision by the floating point casting operation.

### 3.2 Iterative refinement

The iterative refinement is a classical method to improve the accuracy of the linear system. For mixed precision arithmetic, solution of the linear system is found in lower precision but calculation of the residual is performed in higher precision. Therefore we can expect iterative renfinement will converge with higher accuracy.

**Algorithm 3** *iterative refinement to improve solution in lower precision*
1. find $\widehat{e_0}$ satisfying $\widehat{Q}\widehat{e_0} = \widehat{b}$ in lower precision .
2. convert $x_0 \leftarrow \widehat{e_0}$ from lower precision to higher precision .
3. compute residual $r_0 = b - A\, x_0$ of solution .
4. loop $n = 0, 1, 2, \cdots$
   4a. truncate $\widehat{r_n} \leftarrow r_n$, higher precision data to lower preicision .
   4b. find $\widehat{e_n}$ satisfying $\widehat{Q}\widehat{e_n} = \widehat{r_n}$ in lower precision .
   4c. update solution $x_{n+1} = x_n + \widehat{e_n}$ by adding lower precision data .
   4d. compute residual $r_{n+1} = b - A\, x_{n+1}$ of solution .

Here in the procedure for updating $n+1$-th solution, addition of lower precision data can be performed directly without preparation of a working vector $e_n$ in higher precision converting form lower precision $\widehat{e_n}$. To make clear of the role of the preconditioner $Q$ in higher precision, let $Q\,e_n = r_n$ denote the operation to find $\widehat{e_n}$ satisfying $\widehat{Q}\widehat{e_n} = \widehat{r_n}$ in lower preicision for given data $r_n$ that is converted to $\widehat{r_n}$ and up-converting $\widehat{e_n}$ in lower to $e_n$. Calculation of the residual $r_1$ in the first step is viewed as following using the assumption that $A$ is invertible,

$$
\begin{aligned}
r_1 &= b - A\,x_1 = b - A(x_0 + e_1) = r_0 - Ae_1 = r_0 - AQ^{-1}r_0 = (I - AQ^{-1})r_0\,,\\
x_1 &= x_0 + A^{-1}(I - (I - AQ^{-1}))r_0 = x_0 + A^{-1}AQ^{-1}r_0 = x_0 + Q^{-1}r_0\,.
\end{aligned}
$$

By the same argument, residual and solution at $n$-th step are obtained as

$$
\begin{aligned}
r_n &= (I - AQ^{-1})^n r_0\\
x_n &= x_0 + A^{-1}(I - (I - A\,Q^{-1})^n)r_0\\
&= x_0 + \binom{n-1}{0}Q^{-1}r_0 - \binom{n-1}{1}(Q^{-1}A)Q^{-1}r_0 + \cdots + (-1)^{(n-1)}\binom{n-1}{n-1}(Q^{-1}A)Q^{n-1}r_0
\end{aligned}
$$

using the binomial expansion. We conclude that iterative refinement will find solution $x_0$ in a Krylov subspace with $Q^{-1}A$ and $Q^{-1}r_0$

$$
x_n \in x_0 + \mathrm{span}[Q^{-1}r_0, (Q^{-1}A)Q^{-1}r_0, (Q^{-1}A)^2Q^{-1}r_0, \cdots, (Q^{-1}A)^{n-1}Q^{-1}r_0]\,.
$$

### 3.3 preconditioned GCR method for single RHS

The iterative refinement procedure to improve accuracy of the solution obtained by lower precision arithmtic can be viewed as an iterative process to find solution in the Krylov subspace of preconditioned matrix with fixed coefficient for linear combination. The coefficient by the binomial expansion is not optimal and we can use the standard proceudre of Krylov subspace solver family. The most easiest method in implementation is Generalized Conjugate Residual (GCR) method [4] and it is also closed to the iterative refinement procedure with further approximation. A preconditioned GCR method for the linear system $A\,x = b$ with $A \in \mathbb{R}^{N\times N}$ and $b \in \mathbb{R}^N$ by using solution $Q\,y = f$ with $Q \in \mathbb{R}^{N\times N}$ and $f \in \mathbb{R}^N$ in lower precision as a right preconditioner

$$
AQ^{-1}Q\,x = AQ^{-1}\widetilde{x} = b
$$

is given as Algorithm 4. When $A$ has moderate condition number, the solution process by $LDU$-factorization of $A$ in lower precision that is expressed as $Q^{-1}$ is well performed and $AQ^{-1}$ is very closed the identity matrix, $AQ^{-1} \simeq I_N$. In practice the following right preconditioned GCR converges in few iterations, which is rather natural consequence by selection of moderate part of the matrix $K_{11}$ using threshold postponing in lower precision. An example of convergence history will be shown in Section 3.5.

**Algorithm 4** *preconditioned GCR method*
find $x_0$ satisfying $Q\,x_0 = b$
$r_0 = b - A\,x_0$
$w_0 = Q^{-1}r_0$

$p_0 = w_0$
loop $n = 0, 1, 2, \cdots$
$\alpha_n := \dfrac{(r_n, Ap_n)}{(A\,p_n, A\,p_n)} = \dfrac{(r_n, AQ^{-1}\widetilde{p_n})}{(AQ^{-1}\widetilde{p_n}, AQ^{-1}\widetilde{p_n})}$
$x_{n+1} := x_n + \alpha_n p_n$
$r_{n+1} := r_n - \alpha_n A\,p_n$
find $w_{n+1}$ satisfying $Q\,w_{n+1} = r_{n+1}$
for $0 \le m \le n$
$\beta_{m\,n} := -\dfrac{(A\,w_{n+1}, A\,p_n)}{(A\,p_m, A\,p_m)} = -\dfrac{(AQ^{-1}r_{n+1}, AQ^{-1}\widetilde{p_n})}{(AQ^{-1}\widetilde{p_m}, AQ^{-1}\widetilde{p_m})}$
$p_{n+1} := w_{n+1} + \sum_{m=0}^{n} \beta_{m\,n} p_m$

In practical computation, to avoid two times multiplication of $A$ to $p_n$ and to $w_{n+1}$, $A\,p_n$ is stored as $q_n$ and is updated in the same manner as $p_{n+1}$ using a new vector $z_{n+1} := A\,w_{n+1}$, which results in $q_{n+1} := z_{n+1} + \sum_{m=0}^{n} \beta_{m\,n} q_m$. Here $A\,w_{n+1}$ is performed by the SpMV (Sparse Matrix-Vector multiplication) opertaion.

We can see the residual at $n$-step belongs to $n+1$-dimensional Krylov subspace, $r_n \in K_{n+1}(r_0, AQ^{-1})$ and approximate solution $\widetilde{x}_n$ of the right preconditioned system $AQ^{-1}\widetilde{x} = b$ is found as $\widetilde{x}_n \in \widetilde{x}_0 + K_n(r_0, AQ^{-1})$ with the initial approximation $Q\,x_0 = \widetilde{x}_0$. By defintion of $\widetilde{x}_n = Q\,x_n$, approximation of the linear system is written as $x_n \in x_0 + K_n(Q^{-1}r_0, Q^{-1}A)$.

Orthogonality on residual, for all $y \in K_n(r_0, AQ^{-1})$, it holds that $(r_{n+1}, AQ^{-1}y) = 0$ and search vectors $(AQ^{-1}\widetilde{p_m}, AQ^{-1}\widetilde{p_n}) = 0$ for $m \neq n$. Both are verified by induction. Since we can assume $AQ^{-1} \simeq I_N$, we get

$$\alpha_n = \frac{(r_n, AQ^{-1}\widetilde{p_n})}{(AQ^{-1}\widetilde{p_n}, AQ^{-1}\widetilde{p_n})} \simeq \frac{(r_n, \widetilde{p_n})}{(\widetilde{p_n}, \widetilde{p_n})} \quad \text{and } \beta_{m\,n} \simeq -\frac{(r_{n+1}, \widetilde{p_n})}{(\widetilde{p_m}, \widetilde{p_m})}\,.$$

If we could approximate $\alpha_n \simeq 1$ and $\beta_{m\,n} \simeq 0$ with $0 \le m \le n$ up to $n$-th step, we will have

$$x_{n+1} \simeq x_n + p_n, \quad r_{n+1} \simeq r_n - A\,p_n, \quad \text{and} \quad p_n \simeq Q^{-1}r_{n+1},$$

which leads to the same procedure of the iteartive refinement.

We can expect that the preconditioned GCR method converges faster than the the iterative refinement thanks to better combination of coefficients for Krylov subspace basis to achieve the Galerkin orthogonality.

### 3.4 preconditioned block GCR method for multiple RHSs

In practice, the linear system consists of multiple RHSs, $K_{12}$ and then it will be more efficient to use block GCR method for multiple RHS, because dimension of Krylov subspace in block version is much larger than one for single RHS. Comparison of convergence of single and multiple RHSs versions will be illustrated in Section 3.5.

A preconditioned block GCR method for the linear system for multiple RHSs, $A\,[x^{(1)}, \cdots, x^{(M)}] = [b^{(1)}, \cdots, b^{(M)}]$ by using solution $Q\,y = f$ in lower precision as a right preconditioner is obtained by introducing multiplication of matrix $A$ to $[w_{n+1}^{(1)}, \cdots, w_{n+1}^{(M)}]$, which is called SpMM (Sparse

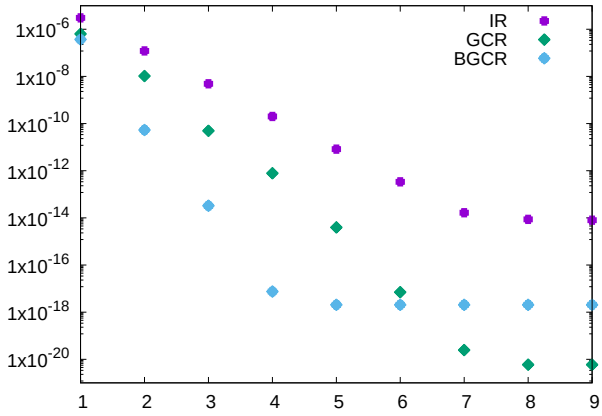


**Figure 2**: convergence history of IR/GCR/BGCR

Matrix-Matrix multiplication) operation. Determination of magnitude of the search vector $\alpha_n$ in updating procedure is replaced by operation by $M \times M$ matrices as

$$[\mathcal{M}_n]_{\ell,k} := (q_n^{(k)}, q_n^{(\ell)}) \quad \text{and} \quad [\mathcal{A}_n]_{\ell,k} := (r_n^{(k)}, q_n^{(\ell)}) \quad \text{for} \quad 1 \le \ell, k \le M$$
$$[x_{n+1}^{(1)}, \cdots, x_{n+1}^{(M)}] := [x_n^{(1)}, \cdots, x_n^{(M)}] + [p_n{}^{(1)}, \cdots, p_n{}^{(M)}]\mathcal{M}_n^{-1}\mathcal{A}_n$$

During the iteration, $[q_{n+1}^{(1)}, \cdots, q_{n+1}^{(M)}]$ are updating with keeping identity to multiplication of $A$ to $[p_{n+1}^{(1)}, \cdots, p_{n+1}^{(M)}]$. Both sets of vectors are calculated from linear combination expressed by $\mathcal{M}_n^{-1}\mathcal{B}_{m\,n}$ with $[\mathcal{B}_{m\,n}]_{\ell,k} := -(z_{n+1}^{(k)}, q_n^{(\ell)})$. We call this iteration as **Algorithm 5**. When all matrices $\mathcal{M}_n$ at $n$-th iteration are invertible, the approximate solution $x_n^{(k)}$ with $1 \le k \le M$ is found as $x_n^{(k)} \in x_0^{(k)} + K_n(Q^{-1}[r_0^{(1)}, \cdots, r_0^{(M)}], Q^{-1}A)$ and $(r_n^{(k)}, AQ^{-1}y) = 0$ for any vector $y \in K_n([r_0^{(1)}, \cdots, r_0^{(M)}], AQ^{-1})$.

### 3.5 Convergence history of preconditioned GCR method

Figure 2 shows convergence history to find a solution of $X_{12}$ satisfying $K_{11}X_{12} = K_{12}$ with matrix in Section 4.2. Sixteen entries are postponed during factorization of $K$ by single precision arithmetic and $K_{12}$ consists of 16 column vectors with $N = 374,520$. Each iteration drawn by colors, e.g., purple for iterative refinement (IR): Algorithm 3, green for preconditioned GCR: Algorithm 4 , and light blue for preconditioned block GCR (BGCR): Algorithm 5, shows convergence of the first column vector of sixteen RHSs. We can see convergence of IR is slower than other GCR solvers and block GCR converges after 5 iterations to the machine epsilon of the double precision.

## 4 NUMERICAL EXAMPLES

In this section, feasibility of hybrid factorization and solution Algorithms 1 and 2 with inner iterative solver by Algorithm 5 using preconditioner in lower precision solution, will be demonstrated by three sparse matrices.

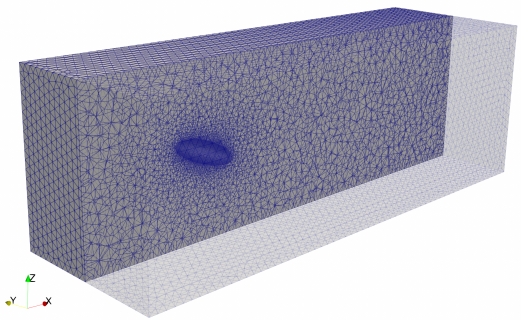

**Figure 3**: finite element mesh decomposition of the box domain excluding an ellipsoid

### 4.1 symmetric matrix from matrix market

The matrix `phb1HYS` from the matrix market database is symmetric and has $36,414$ total number of unknowns and $4,344,765$ originated from protein data bank and was used in evaluation of SpMV performance in [5]. By setting pivot threshold $\tau = 0.05$, 17 entries are excluded by factorization in single precision. By this decomposition, Algorithm 1 is performed with $N = \#\Lambda_1 = 36,397$ and $M = \#\Lambda_2 = 17$. The maximum and minimum eigenvalues are calculated by the power method as $\lambda_{\max}(K_{11}) = 1.0002$, $\lambda_{\min}(K) = 3.8109 \times 10^{-4}$ and then $\kappa(K_{11}) = 2.6448 \times 10^3$. Since the eigenvalue distribution of the whole matrix has significant jumps in the smallest, $\lambda_{\min}(K) = 9.9876 \times 10^{-10}$, $\lambda_{\max}(K) = 34.854$ and then $\kappa(K) = 3.4862 \times 10^{11}$. The single precision arithmetic can factorize the moderate part $K_{11}$ but accuracy of the solution of the whole system with $K$ is very poor. Since the matrix `phb1HYS` is symmetric, $LDL^T$-factorization is performed instead of $LDU$-factorization, but block GCR : Algorithm 5 is used for the system with $K_{11}$ and 17 RHSs.

### 4.2 finite element matrix from incompressible flow problem

Let $\Omega$ be the flow region consisting of a box domain excluding an ellipsoid. The size of the box is $6 \times 2 \times 2$ and diameters of the ellipsoid is $(0.75, 0.5, 0.25)$ with 20 degree slanted. Finite element mesh decomposition $\overline{\Omega} = \bigcup_e e$ with tetrahedra $\{e\}$, whose diameter is denoted as $h_e = |e|$, is depicted in Figure 3. Here nonuniform mesh subdivision is used and smallest mesh size $h = 0.01$ on the surface of the ellipsoid and largest mesh size $h = 0.1$ on the inlet boundary.

We consider the Stokes problem to find the velocity $u$ and the pressure $p$, $-\nabla \cdot D(u) + \nabla p = 0$ and $\nabla \cdot u = 0$ with full homogeneous Neumann data on the all boundary surfaces. Here $D(u)$ denotes the strain rate tensor $D(u) = (\nabla u + (\nabla u)^T)/2$. This boundary condition $2D(u)n - n\,p = 0$ with outer normal $n$ is not physical one, but a domain decomposition method with an artificial boundary condition like FETI method [6] leads to a floating sub-problem with such kind of full Neumann boundaries. Finite element matrix is obtained by discretization of the weak formulation with P1/P1 elements and a stabilization parameter $\delta$ that is set as 0.01 in this example,

$$\overline{K} = \begin{bmatrix} A & B^T \\ -B & \delta D \end{bmatrix}. \tag{4}$$

Each block is defined using finite element basis functions $\{\varphi_i\}$ for the velocity unknown and

$\{\psi_i\}$ for the pressure unknown,

$$[A]_{i\,j} = \int_\Omega D(\varphi_j) : D(\varphi_i), \quad [B]_{i\,j} = -\int_\Omega \nabla\cdot\varphi_j\psi_i, \quad [D]_{i\,j} = \sum_e \int_e h_e^2 \nabla\psi_j\cdot\nabla\psi_i$$

The stiffness matrix $\overline{K}$ with total numbers of unknowns $374,536$ and nonzeros $21,146,848$ is generated from tetrahedral mesh decomposition in Figure 3 by FreeFEM software package [7]. We call this matrix as `stokes`. For constant vectors $a$ and $b$ whose total degrees of freedom is 6, we have $D(a\times x+b)=0$ and $\nabla\cdot(a\times x+b)$, where $a\times x+b$ called rigid body modes. Then it is clear that the stiffness matrix $\overline{K}$ defined in (4) has six dimensional kernel. The maximum and minimum eigenvalues are calculated by the power method as $\lambda_{\max}(K)=2.5003$, $\lambda_{\min}(K|_{\mathrm{Im}(A)})=8.0038\times10^{-8}$, and $\lambda_{\min}(K)=1.1437\times10^{-19}$. Here the operator $K|_{\mathrm{Im}(A)}$ is one-to-one in $\mathrm{Im}(A)$, i.e., the orthogonal complement of $\mathrm{Ker}(A)$, which is obtained numerically by $LDU$-factorization procedure. The condition number of $K$ is $\kappa(K|_{\mathrm{Im}(K)})=3.1240\times10^{7}$ when it is restricted on $\mathrm{Im}(A)$ and $\kappa(K)=2.1863\times10^{19}$ for all unknowns.

Applying single precision arithmetic with pivot threshold $\tau=0.75$, 16 entries are postponed and the Schur complement $S_{22}$, whose size is $16\times16$, has six dimensional kernel. By this decomposition, $\lambda_{\max}(K_{11})=1.0000$, $\lambda_{\min}(K_{11})=1.6677\times10^{-8}$, and $\kappa(K_{11})=5.9962\times10^{7}$.

### 4.3 finite element matrix from semi-conductor problem

The semi-conductor problem is mathematically modeled by the drift-diffusion equations with the electrostatic potential $\varphi$, the electron density $n$, and the hole density $p$. By introducing Slotboom variables for the electron density $\eta=e^{-\varphi}n$ and for the hole density $\xi=e^{\varphi}p$, the drift and diffusion terms are combined into a single term and the following nonlinear system [8] is obtained as

$$-\nabla\cdot(\lambda^2\nabla\varphi) = e^{-\varphi}\xi - e^{\varphi}\eta + C(x), \quad -\nabla(e^{\varphi}\nabla\eta)=0, \quad \nabla(e^{-\varphi}\nabla\xi)=0, \tag{5}$$

where $\lambda$ denotes Debye length, and $C(x)$ is given function to represent doping density of $N$-rich or $P$-rich material. Here we consider two dimensional problem with a box domain $0.3\times0.2$. The $N$-region, $\Omega_N$ with doping density $n_d$ consists of $x<0.1$ and $x>0.2$ and in the middle is the $P$-region $\Omega_P$ with density $n_a$, which is called N-P-N device. $C(x)=n_d=10^{20}$ for $x<0.1$ or $x>0.2$ and $C(x)=-n_a=-6\times10^{17}$ for $0.1<x<0.2$. Dirichlet boundary conditions $\varphi_D$, $n_D$, and $p_D$ are given on $x=0$ and 0.3 and Neumann boundary conditions are given on other sides.

The electrostatic potential $\varphi_*$ by setting $\xi\equiv1$ and $\eta\equiv1$ and then satisfying $-\nabla(\lambda^2\nabla\varphi_*) = e^{-\varphi_*}-e^{\varphi_*}+C(x)$ and $\varphi_*=\sinh^{-1}(n_d/(2n_i))$ with $n_i=1.08\times10^{10}$ on $x=0$ and $x=0.3$ is called thermal equilibrium. The left of Figure 4 shows distribution of $\varphi_*$ in the N-P-N device with $\Omega=(0,0.3)\times(0,0.2)$. A Newton iteration to obtain the thermal equilibrium is rather straightforward and is a part of the Gummel map [8], which is a kind of fixed point method in total. To obtain a solution of the nonlinear system (5), we will apply a Newton iteration starting from the thermal equilibrium $(\varphi,\eta,\xi)=(\varphi_*,1,1)$. By introducing expression on hole current density $J_p=-e^{-\varphi_*}\nabla\xi$, a mixed formulation of an elliptic equation with coefficient $e^{-\varphi_*}$ in the first step of the Newton step is obtained as

$$\int_\Omega e^{\varphi_*}J_p\cdot v - \int_\Omega \nabla\cdot v\,\xi - \int_\Omega \nabla\cdot J_p\,q = \int_\Omega f\cdot v - \int_{\Gamma_D}\xi_D v\cdot\nu, \tag{6}$$

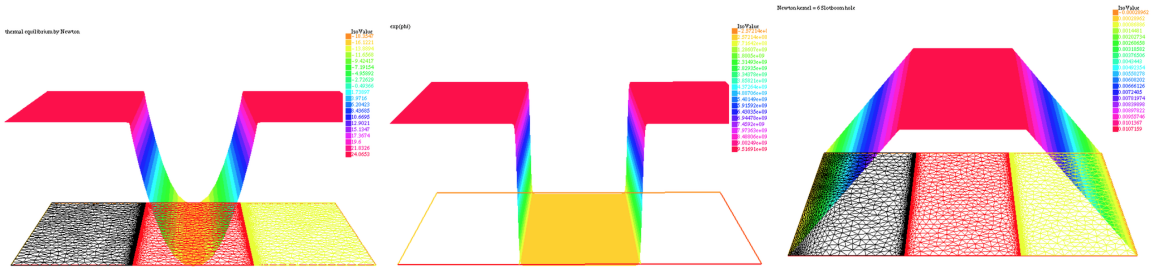

**Figure 4**: distribution of functions for N-P-N semiconductor device, left : electrostatic potential $\varphi_*$, middle : $e^{\varphi_*}$ in matrix coefficient, right : pseudo kernel of the hole $\xi^0$

where external force $f$ represents nonlinear coupling between electrostatic potential unknown $\varphi$ and hole unknowns $(J_p, \xi)$ and $\Gamma_D$ is a part of the Dirichlet boundary with $\xi_D = e^{\varphi_D} p_D$ and $\nu$ denotes the outer normal to boundary $\Gamma_D$.

Finite element matrix is obtained by discretization of a weak formulation of the weak formulation with RT0/P1 elements, where RT0 is the Raviart-Thomas finite element in the lowest order for vectorial unknown function $H(\mathrm{div}\,;\Omega) = \{v \in L^2(\Omega)\,;\, \int_\Omega \nabla \cdot v < +\infty\}$ [9],

$$\overline{K} = \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix}, \tag{7}$$

Here mass and constraint matrices are defined using finite element basis functions $\{\varphi_i\}$ for $J_p$ and $\{\psi_i\}$ for $\xi$,

$$[A]_{ij} = \int_\Omega e^{\varphi_*} \varphi_j \cdot \varphi_i, \quad [B]_{ij} = -\int_\Omega \nabla \cdot \varphi_j \psi_i \,.$$

We call this matrix as `dd-hole`. The electrostatic potential $\varphi_*$ takes negative value in the $P$-region and ratio of $e^{\varphi_*}$ between $N$-region and $P$-region becomes below $10^{-16}$. By approximating $e^{\varphi_*} \simeq 0$ in $P$-region, the first term of (6) by the domain integration is replaced by $\int_{\Omega_N} e^{\varphi_*} J_p \cdot v$ and for arbitrary constant $c$, $(J_p^0, \xi^0)$ that satisfies

$$J_p^0 = 0,\ \xi^0 = c \text{ in } \Omega_P,\ -\nabla\cdot(e^{-\varphi_*}\nabla\xi^0) = 0 \text{ in } \Omega_N \text{ with } \xi^0 = c \text{ on } \partial\Omega_N \cap \partial\Omega_P \text{ and } J_p^0 = -e^{\varphi_*}\nabla\xi^0$$

will be the solution of the modified weak formulation. Shifted solution $(J_p, \xi)$ of (6) by $(J_p^0, \xi^0)$ still almost satisfies the same weak formulation with difference as $\int_{\Omega_P} e^{\varphi_*} J_p \cdot v$, which is the residual of the approximation of $e^{\varphi_*}$ by zero. This property confirms the stiffness matrix $\widetilde{K}$ is singular with one dimensional kernel, when all coefficients are stored in double precision. The middle and the right of Figure 4 show the exponential weight $e^{\varphi_*}$ with the thermal equilibrium and the kernel function $\xi^0$. The maximum and minimum eigenvalues calculated by quadruple precision for given matrix in double precision, $\lambda_{\max}(K) = 6.13488 \times 10^{10}$, $\lambda_{\min}(K|_{\mathrm{Im}(A)}) = 3.5704 \times 10^{-12}$. The condition number of $K$ on $\mathrm{Im}(A)$ is $\kappa(K|_{\mathrm{Im}(K)}) = 1.7193 \times 10^{22}$. Applying double precision arithmetic with pivot threshold $\tau = 0.01$, 13 entries are postponed and the Schur complement $S_{22}$, whose size is $13 \times 13$, has one dimensional kernel. By this decomposition, $\lambda_{\max}(K_{11}) = 2.0080 \times 10^2$, $\lambda_{\min}(K_{11}) = 3.6200 \times 10^{-12}$, and $\kappa(K_{11}) = 5.5472 \times 10^{13}$.

### 4.4 performance comparison of mixed and pure precision arithmetic

Here we summarize performance of the proposed algorithm in accuracy and computational speed. We used one core of Apple M1 Max CPU running at 3.23 GHz, which has capability to perform two single precision arithmetic instead of one double precision in the same cycle. For matrices `phb1HYS` in Section 4.1 and `stokes` in Section 4.2, double precision arithmetic is used for higher accuracy and single and double arithmetic are used as mixed precision which is labeled as mixed(double+single) in the Table 1 . For matrix `dd-hole` in Section 4.3, quadruple precision arithmetic that is realized as double-double in QD library [10] is used for higher accuracy and double and double-double arithmetic are used as mixed precision which is labeled as mixed(quadruple+double). The error and the residual of the linear system is calculated from the RHS that is set to satisfy the solution is $[x]_i \equiv i (\mathrm{mod}\ 11)$. Since the later two matrices are singular, detected kernel dimension is also shown.

**Table 1**: error, residual and elapsed time for factorization by pure and mixed precision arithmetic

| `phb1HYS` | $n = 36,414$, $nnz = 4,344,765$ | | |
|---|---|---|---|
| | double | mixed(double+single) | single |
| error | $1.2650 \times 10^{-6}$ | $1.6647 \times 10^{-6}$ | $3.1178 \times 10^{-2}$ |
| residual | $7.6201 \times 10^{-16}$ | $9.6442 \times 10^{-16}$ | $4.4906 \times 10^{-7}$ |
| time in second | 0.5328 | 0.4604 | 0.4053 |
| `stokes` | $n = 374,536$, $nnz = 21,146,848$ | | |
| | double | mixed(double+single) | single |
| error | $4.3646 \times 10^{-13}$ | $2.0301 \times 10^{-12}$ | $1.7046 \times 10^{-2}$ |
| residual | $1.2730 \times 10^{-15}$ | $5.1881 \times 10^{-15}$ | $7.0494 \times 10^{-7}$ |
| dim. of kernel | 6 | 6 | 0 |
| time in second | 33.390 | 22.983 | 15.228 |
| `hole` | $n = 40,323$, $nnz = 401,243$ | | |
| | quadruple | mixed(quadruple+double) | double |
| error | $5.3652 \times 10^{-20}$ | $1.3061 \times 10^{-21}$ | $8.9382 \times 10^{-6}$ |
| residual | $5.3850 \times 10^{-32}$ | $1.5514 \times 10^{-32}$ | $4.7860 \times 10^{-16}$ |
| dim. of kernel | 1 | 1 | 1 |
| time in second | 16.599 | 2.6459 | 0.4064 |

## 5 CONCLUSIONS

We have constructed a new hybrid algorithm for $LDU$-factorization for large sparse matrix introducing iterative solver for generation of Schur complement matrix in higher precision, where the matrix is decomposed into a union of moderate and hard parts. Numerical tests confirm the solution by the proposed algorithm by mixed precision arithmetic can keep accuracy as higher precision arithmetic.

When quadruple precision arithmetic are realized by using double-double data structure and are performed on the hardware equipped with fused multiply-add unit, ratio of arithmetic complexity of double-double to double is 25 to 1. Therefore for the linear system that has huge

condition number more than the range of the maximum floating digits of the double precision, mixed precision arithmetic with quadruple and double attains substantial speed-up, which was verified by a matrix from the semi-conductor problem.

Since recent CPU has ratio of arithmetic complexity of double to float is 2 to 1, some speed-up is obtained, but solution phase by iteration procedure for recovering double precision accuracy for the Schur complement matrix masks the efficiency. It is necessary to implement our hybrid factorization algorithm on the system with more single floating point arithmetic units than double and to evaluate the performance.

For the solution phase of the large system with multiple RHSs to generate Schur complement, in the preconditioned part by forward/backward substitution in lower precision already well utilizes the BLAS level 3 routine, e.g., `TRSM`, but it is necessary to optimize SpMM operation in double and quadruple precision, because such kind of sparse linear algebra library is not provided yet.

## REFERENCES


[1] Higham, N. J. and Mary, T., Mixed precision algorithms in numerical linear algebra. *Acta Numerica* (2022) 347–414.

[2] George, A. Numerical experiments using dissection methods to solve n by n grid problems. *SIAM J. Numer. Analy.* (1977) **14**:161-–179.

[3] Suzuki, A. and Roux, F.-X., A dissection solver with kernel detection for symmetric finite element matrices on shared memory computers. *Int. J. Numer. Meth. Engng.* (2014) **100**:136–164.

[4] Saad, Y., *Iterative methods for sparse linear systems (2nd ed.)*. SIAM, (2003)

[5] Williams, S., Oliker, L., Vuduc, R., Shalf, J., Yelick, K., and Demmel, J., Optimization of sparse matrix-vector multiplication on emerging multicore platforms, *Parallel Computing* (2009) **35**:178–194.

[6] Farhat, C, Roux F.-X., Implicit parallel processing in structural mechanics. *Computational Mechanics Advances* (1994) **2**:1–124.

[7] Hecht, F., C++ tools to construct our user-level language., *ESIAM: M2AN*, (2002) **36**:809–836.

[8] Brezzi, F., Marini, D. D., Micheteletti, S., Pietra, P., Sacco, R. and Wang, S., Discretization of Semiconductor Device Problems(I), *Handbook of numerical analysis*, Ciarlet, P. G. ed, (2005) **13**:317–441.

[9] Boffi, D., Brezzi, F. and Fortin, M. *Mixed finite element methods and applications*, Springer, 2010.

[10] Bailey, D. H., Li. X.S., Hida, Y. QD: A double-double/quad-double package, Computer software, doi:10.11578/dc.20210416.14 , (2003).