

# A fill-in method for IC( $p$ ) preconditioning in large-scale ill-conditioned linear systems

Koki Masui and Fumihiko Ino

Graduate School of Information Science and Technology, The University of Osaka  
Osaka 565-0871, Japan, masui@ist.osaka-u.ac.jp

This paper proposes an Incomplete Cholesky preconditioner with fill-in that improves the convergence of iterative methods and reduces memory usage for ill-conditioned symmetric linear equations. We developed an IC preconditioning where the fill-in level is a decimal and controls the number of fill-ins. As a result, our proposed method succeeded in finely controlling the number of fill-ins, reducing the memory usage for the preconditioning matrix by 22% and the computation time by 13% compared to the conventional methods.

*Index Terms*—Convergence of numerical methods, electromagnetic fields, iterative methods, large-scale systems.

## I. INTRODUCTION

IN solving the complex symmetric linear equations derived from high-frequency electromagnetic field analysis [1], it is known that the convergences of iterative methods such as the COCG [2] and COCR [3] methods are slow. Moreover, the convergence becomes worse as the problem becomes larger and more complex. For these problems, the preconditioning method such as Incomplete Cholesky preconditioning with fill-in (IC( $p$ )) is used to improve the convergence of iterative methods [4], where  $p$  is the level of fill-in, which is a non-negative integer. As the value of  $p$  increases, the preconditioning matrix approaches the coefficient matrix and improves the convergence.

However, increasing the value of  $p$  can cause a huge number of non-zero elements in the preconditioning matrix, which results in higher computational costs per iteration and significantly more memory usage.

Some dropping methods are proposed to mitigate these issues [5], [6]. However, handling tuning issues such as setting parameter thresholds, which require expert knowledge, is necessary. Besides, a method IC(0.5) is proposed to control the number of fill-ins by the position of non-zero elements in the coefficient matrix [7]. Although IC(0.5) performs effectively for convergence and calculation time in many cases, it does not improve the convergence much compared to IC(1) for ill-conditioned and larger problems. Therefore, we propose a method IC(1.5) that improves convergence compared to IC(1) and reduces the amount of memory compared to IC(2).

## II. FILL-IN FOR IC PRECONDITIONING

Equations (1) and (2) show how to determine the position of non-zero elements in the preconditioning matrix for IC( $p$ ). Here,  $lev_{ij}$  is the level of  $L_{ij}$ , that is the element in row  $i$  and column  $j$  in the preconditioning matrix  $L$ . First, the initial fill-in levels of all positions are determined by (1). Next, each value of  $lev_{ij}$  is updated by (2).  $L_{ij}$  is non-zero if  $lev_{ij} \leq p$ . After computing the all the values of  $lev_{ij}$ , the IC( $p$ ) preconditioning matrix is calculated by the Algorithm 1. Here,  $A$  is a coefficient

---

### Algorithm 1 IC decomposition with fill-in

---

```

1:  $L_{00} := 1.0$ 
2:  $D_{00} := \alpha A_{00}$ 
3: for  $i = 1, 2, 3, \dots, n-1$  do
4:   for  $j = 0, 1, 2, \dots, i$  do
5:     if  $lev_{ij} \leq p$  then
6:       if  $i = j$  then
7:          $L_{ij} := 1.0$ 
8:       else
9:          $L_{ij} := (A_{ij} - \sum_{k=1}^{j-1} L_{ik} D_{kk} L_{jk}) / D_{jj}$ 
10:      end if
11:    else
12:       $L_{ij} := 0.0$ 
13:    end if
14:  end for
15:   $D_{ii} := \alpha A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2 D_{kk}$ 
16: end for

```

---

matrix,  $L$  is a lower triangular matrix,  $D$  is a diagonal matrix, and  $\alpha$  is the acceleration factor.

$$lev_{ij} = \begin{cases} 0, & \text{if } A_{ij} \neq 0 \\ \infty, & \text{otherwise,} \end{cases} \quad (1)$$

$$lev_{ij} = \min\{lev_{ij}, \min_{0 \leq k < j} \{lev_{ik} + lev_{jk} + 1\}\}. \quad (2)$$

## III. PROPOSED METHOD

Our proposed method, IC(1.5) decides non-zero element positions from the elements with  $lev_{ij} \leq 1$  in the preconditioning matrix. Equation (3) shows how to determine the non-zero element positions in IC(1.5).

First, elements with  $lev_{ij} \leq 1$  are determined by (1) and (2).

Next, all non-zero elements in the preconditioning matrix are determined by (3).

$$lev_{ij} = \min\{lev_{ij}, 1 + 1/(\sum_{k=0}^j I(lev_{ik} \leq 1 \wedge lev_{jk} \leq 1))\}. \quad (3)$$

where  $I$  represents the indicator function. Here,  $1/(\sum_{k=0}^j I(lev_{ik} \leq 1 \wedge lev_{jk} \leq 1))$  is considered as  $\infty$  if  $\sum_{k=0}^j I(lev_{ik} \leq 1 \wedge lev_{jk} \leq 1) = 0$ . Equation (3) reduces the value of  $lev_{ij}$  that contains the results of multiple references for IC( $p$ ) preconditioning among the elements with  $lev_{ij} = 2$ . Hence, this method improves the convergence compared to IC(1) while reducing the number of non-zero elements compared to IC(2).

Finally, the IC(1.5) preconditioning matrix is computed by the Algorithm 1 with  $p = 1.5$ .

#### IV. NUMERICAL EXPERIMENTS

All experiments were performed using an Intel Xeon Silver 4314 (2.4GHz) processor and GCC 11.4.0 compiler with “-O3 -mfma” optimization flags. Equation (4) shows a wave equation having an electric field  $\mathbf{E}$  derived from Maxwell’s equation.

$$\begin{aligned} \int_{\Omega} \text{rot} \mathbf{E}_h \cdot \mu^{-1} \text{rot} \mathbf{E}_h^* d\Omega - \int_{\Omega} (\omega^2 \epsilon' - j\omega\sigma) \mathbf{E}_h \cdot \mathbf{E}_h^* d\Omega \\ = j\omega \int_{\Omega} \mathbf{J}_h \cdot \mathbf{E}_h^* d\Omega. \end{aligned} \quad (4)$$

where  $\mu$  (H/m) is the permeability,  $\epsilon_0$  (F/m) is the permittivity of a vacuum,  $\epsilon' = 80.0$  is the relative permittivity,  $\omega = 2\pi f$  (rad/s) is a single angular frequency,  $j$  is the imaginary unit,  $\mathbf{J}_h$  (A/m<sup>2</sup>) is the current density,  $\sigma = 0.52$  (S/m) is the conductivity, and  $f = 300$  (MHz) is the frequency.

Three coefficient matrices, each with 160,013, 979,464, and 1,646,787 degrees of freedom (DoF), were obtained from TEAM Workshop Problem 29 with a phantom model [8]. For each complex symmetric matrix  $A$ , we solved the equation  $Ax = b$  by using the COCG method with IC( $p$ ) preconditioning. The initial solution and the convergence tolerance based on the residual norm were set to  $x = 0$  and  $10^{-9}$ , respectively. We evaluated the number of non-zero elements in the preconditioning matrix and the performance of the COCG method. In addition, we investigated the convergence behavior of the COCG method with fill-in levels set to 0, 0.5, 1, 1.5, and 2. In all cases, we set the acceleration factor  $\alpha$  to 1.05.

Table I shows the number of non-zero elements in the preconditioning matrix with different fill-in levels. The proposed IC(1.5) succeeded in finely controlling the number of fill-in regardless of the matrix values, reducing the memory usage for the preconditioning matrix by 22% compared to IC(2).

Table II shows that the number of iterations decreased as the fill-in level increased. Moreover, the proposed method reduced the computation time by 13% compared to conventional methods.

Figure 1 shows the convergence history for different fill-in levels. The number of iterations to convergence progressively reduced as the fill-in level increased. In particular, the proposed IC(1.5) showed a similar convergence behavior compared to IC(2). As a result, IC(1.5) achieved 50% less iterations compared to IC(1) at less cost than IC(2).

TABLE I  
NUMBER OF NON-ZERO ELEMENTS IN PRECONDITIONING MATRIX

| DoF       | IC(0)      | IC(0.5)    | IC(1)      | IC(1.5)    | IC(2)       |
|-----------|------------|------------|------------|------------|-------------|
| 160,013   | 1,343,678  | 1,763,199  | 3,516,246  | 7,768,424  | 9,827,883   |
| 979,464   | 8,387,104  | 11,080,163 | 22,268,604 | 50,206,392 | 63,799,785  |
| 1,646,787 | 14,148,990 | 18,707,111 | 37,688,670 | 85,325,247 | 108,516,995 |

TABLE II  
NUMBER OF ITERATION AND COMPUTATION TIME FOR CONVERGENCE

| DoF       |          | IC(0)  | IC(0.5) | IC(1)      | IC(1.5)      | IC(2)  |
|-----------|----------|--------|---------|------------|--------------|--------|
| 160,013   | # iter.  | 14,318 | 12,397  | 4,229      | 2,585        | 2,421  |
|           | time (s) | 381    | 380     | <b>210</b> | 214          | 244    |
| 979,464   | # iter.  | 49,488 | 42,616  | 13,994     | 6,808        | 6,093  |
|           | time (s) | 7,462  | 7,205   | 3,391      | <b>3,026</b> | 3,221  |
| 1,646,787 | # iter.  | 64,392 | 55,426  | 28,182     | 11,977       | 11,095 |
|           | time (s) | 15,685 | 15,040  | 11,109     | <b>8,654</b> | 9,908  |

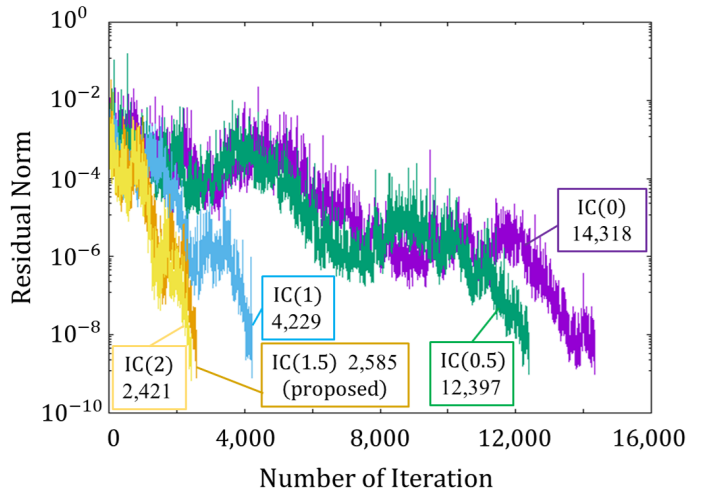


Fig. 1. Convergence history for different methods at matrix size 160,013.

#### ACKNOWLEDGMENT

This work was supported by in part by JSPS KAKENHI Grant-in-Aid for Early-Career Scientists 24K20780.

#### REFERENCES

- [1] A. Takei, S. Yoshimura, and H. Kanayama, “Large-Scale Parallel Finite Element Analyses of High Frequency Electromagnetic Field in Commuter Trains,” *Comput. Modeling Engrg. Sci.*, 31, pp.13-24, 2008.
- [2] H.A. van der Vorst and J.B.M. Melissen, “A Petrov-Galerkin type method for solving  $Ax=b$  and where  $A$  is symmetric complex,” *IEEE Trans. Magn.*, 26, pp.706-708, 1990.
- [3] T. Sogabe and S.L. Zhang, “A COCR method for solving complex symmetric linear systems,” *J. Comput. Appl. Math.*, Vol.199-2, pp.297-303, 2007.
- [4] Y. Saad, “Iterative Methods for Sparse Linear Systems,” 2nd edn. *SIAM Society for Industrial & Applied Mathematics*, Philadelphia, 2003.
- [5] Y. Saad, “A Dual Threshold Incomplete LU Factorization,” *Numerical Linear Algebra with Applications*, Vol.1, pp.387-402, 1994.
- [6] M. Bollhoefer, “A Robust ILU with Pivoting based on Monitoring the Growth of the Inverse Factors,” *Linear Algebra and Its Applications*, Vol.338, pp.201-218, 2001.
- [7] K. Masui and F. Ino, “IC( $p$ ) preconditioning for large symmetric linear equations,” In *Proc. International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia)*, Nagoya, 2024.
- [8] Y. Kanai, “Description of TEAM Workshop Problem 29: Whole body cavity resonator,” *TEAM Workshop*, Tucson, 1998.