IC($\boldsymbol{p}$) preconditioning for large symmetric linear equations

Koki MASUI (Osaka University) Fumihiko INO (Osaka University)

Backgrounds and objectives

- Demand for speed-up of the iterative method for solving complex symmetric linear equations derived from edge finite element analysis of high-frequency electromagnetic fields
  - ✓ poor convergence due to ill-conditioned coefficient matrix
- IC(p) can improve the convergence (p: fill-in level)
  - ✓ the number of non-zero elements in the preconditioning matrix increases by a factor of several
  - ✓ this method can **exhaust memory in huge problems**

- we present a method that is able to control the level of fill-in

- p is not integer
- we propose a method, called **IC(0.5)**

IC($\boldsymbol{p}$) preconditioning w/ fill-in

IC(0): non-zero element positions of preconditioning matrix and coefficient matrix are the same

IC(p): generates the preconditioning matrix with additional non-zero element positions depending on the value of p.

The additional non-zero element positions are determined by

$$Level_{i,j} = \min\{Level_{i,j}, \min_{1 \le k \le \min(i,j)} \{Level_{i,k} + Level_{k,j} + 1\}\}$$

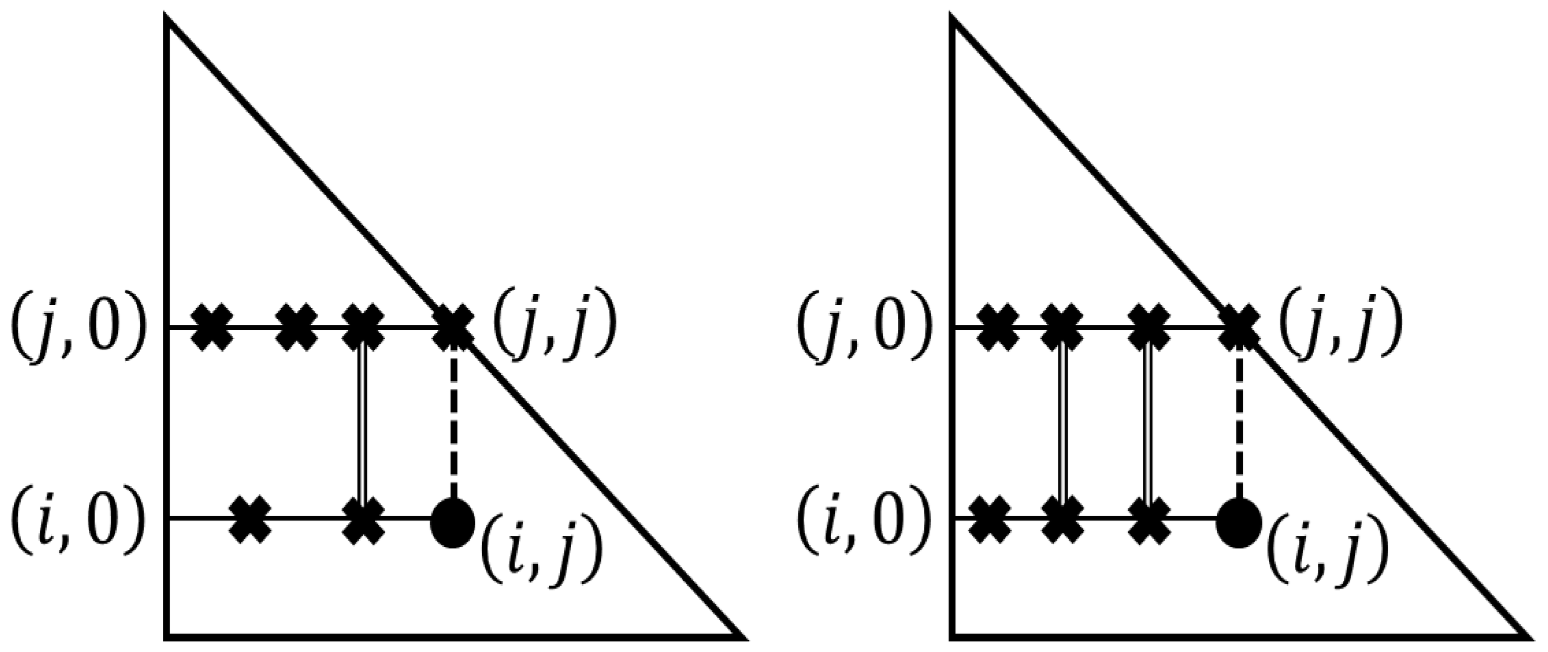
$L_{0,0} := 1.0$
$D_0 := A_{0,0} \times \alpha$
for $i = 1,2,3,\dots,n-1$
  for $j = 0,1,2,\dots,i$
    if $(i = j)$ **then**
      $L_{i,j} := 1.0$
    else
      if $(Level_{i,j} = 0)$ **then**
        $L_{i,j} := (A_{i,j} - \sum_{k=1}^{j-1} L_{i,k} D_k L_{j,k}) / D_j$
      else if $(Level_{i,j} \le p)$ **then**
        $L_{i,j} := -\sum_{k=1}^{j-1} L_{i,k} D_k L_{j,k} / D_j$
      end if
    end if
  end for
  $D_i := A_{i,i} \times \alpha - \sum_{k=1}^{i-1} L_{i,k}^2 D_k$
end for

Algorithm of IC($\boldsymbol{p}$) preconditioning w/ fill-in

IC(0.5) preconditioning (proposed method)

IC(1):the element in (i, j) becomes additional non-zero if there are any non-zero elements in rows i and j that have the same column(Fig. 1(a))

IC(0.5): the element in (i, j) becomes additional non-zero **if two or more non-zero elements** in rows i and j have the same column(Fig. 1(b))



(a) Fill-in with IC(1) **(b)Fill-in with IC(0.5)**

Figure 1: How to determine the additional non-zero element positions.

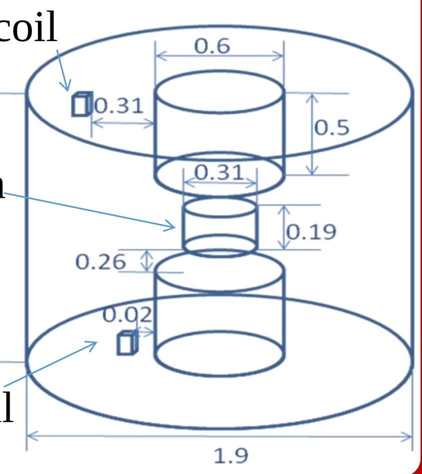
Numerical model

- Edge finite element equation for vector wave equation

$$\int_\Omega \mathrm{rot}\mathbf{E}_h \cdot \mu^{-1} \mathrm{rot}\mathbf{E}_h^* d\Omega - \int_\Omega (\omega^2 \varepsilon' - j\omega\sigma) \mathbf{E}_h \cdot \mathbf{E}_h^* d\Omega = j\omega \int_\Omega \mathbf{J}_h \cdot \mathbf{E}_h^* d\Omega$$

where $\mathbf{E}_h$ (V/m) is FE approximation of electric field, $\mathbf{E}_h^*$ (V/m) test function satisfying $\mathbf{E}_h^* \times n = 0$ on $\partial\Omega$, $\mathbf{J}_h$ (A/m²) FE approximation of current density, μ (H/m) permeability, ε' (F/m) real part of complex permittivity, σ (S/m) electric conductivity, $\omega = 2\pi f$ (rad/s) single angular frequency, f (Hz) frequency, n outward normal vector on the boundary, and j imaginary unit.

- TEAM Workshop Problem 29
  - # of rows, columns: 160,013, 439,176, 979,464
  - # of nonzeros: 2,123,849, 7,036,670, 15,794,744
  - $\varepsilon' = 80.0$ (F/m)
  - $\sigma = 0.52$ (S/m)
  - $f = 1,\ 300$ (MHz)



Numerical experiments

- Computer
  - CPU (clock frequency): Intel Core i9-9900 (3.6 GHz)
  - Compiler (flag): GCC-11.4.0 (-O3 -mfma)
- Solver
  - COCG method with shifted IC(p) preconditioner(p = 0, 0.5, 1)
  - Acceleration Factor: 1.1
  - Convergence criteria: $\varepsilon = 10^{-9}$

① The number of nonzero elements in the preconditioner

DoF\preconditioner	IC(0)	IC(0.5) (proposed method)	IC(1)
160,013	1,343,678	**1,763,199**	3,516,246
439,176	3,737,923	**4,926,496**	9,853,213
979,464	8,387,104	**11,080,163**	22,268,604

Our proposed method succeeded in controlling the non-zero elements in preconditioning matrix

② Iteration counts and calculation time

1 MHz

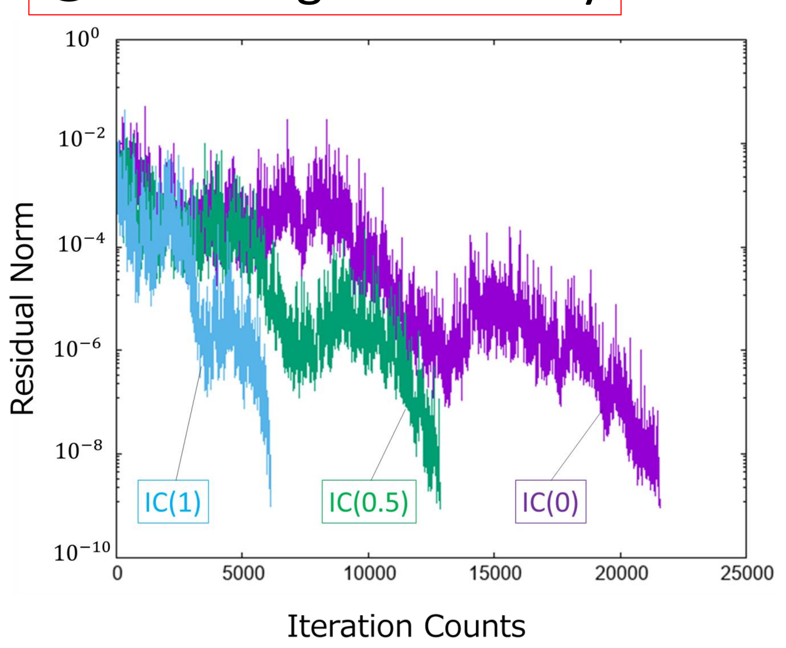
DoF\preconditioner	IC(0)		IC(0.5)		IC(1)	
	Iter.	Time [s]	Iter.	Time [s]	Iter.	Time [s]
160,013	9,491	202	8,026	197	5,264	199
439,176	7,055	406	5,211	347	4,482	463
979,464	8,199	1,106	6,519	1,015	5,312	1,249

300 MHz

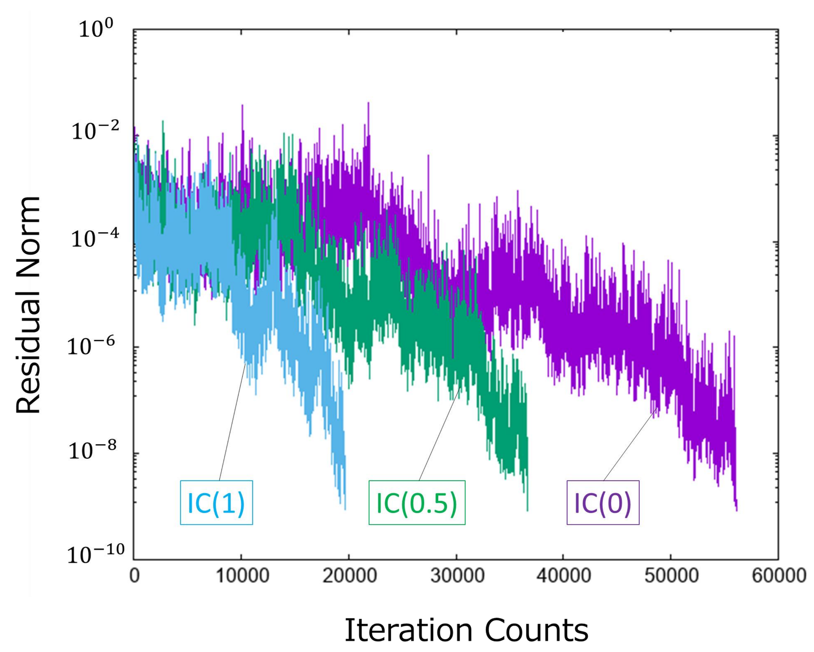
DoF\preconditioner	IC(0)		IC(0.5)		IC(1)	
	Iter.	Time [s]	Iter.	Time [s]	Iter.	Time [s]
160,013	21,562	455	12,842	311	6,119	228
439,176	33,220	1,934	19,510	1,309	10,625	1,095
979,464	56,708	7,492	36,667	5,655	19,662	4,651

- Succeeded in reducing the calculation time compared to IC(0) in all cases.
- In the case of 1 MHz, proposed method succeeded in reducing the calculation time compared to both of IC(0) and IC(1), up to 15%.

③ Convergence history



160,013 (1 MHz)



979,464 (1 MHz)

Our proposed method behaves between IC(0) and IC(1)

Concluding remarks and future works

- ✓ Proposed IC(p) preconditioning method that can control the level of p
- ✓ Succeeded in reducing the calculation time compared to IC(0).
  - In some cases, proposed method succeeded in reducing compared to **both of IC(0) and IC(1)**

◆Future work

- ✓ Investigate the effectiveness of the method with other problems
- ✓ Consider better fill-in methods.