

大規模電磁場問題に対する効率的な倍々精度演算を用いた 反復法の実装について

梶井晃基^{1,2} 荻野正雄³ 片桐孝洋⁴

概要: 高周波電磁場問題などで得られる複素対称線形方程式は収束性が悪いことが知られている。反復法における前処理としては IC(0)(不完全コレスキー分解)前処理が広く使われている。一方で、フィルインを考慮した IC(1)等はあまり使われておらず、その有効性は自明ではない。特に、大規模電磁場解析問題については研究例が少なく、章中規模に有効であった手法が大規模問題でも有効であるかどうかは不明である。そこで今回は自由度 100 万以上の電磁場解析問題に対してフィルインを施した場合の反復法の収束性の変化を調査し、その性能評価を行った結果を報告する。

キーワード: 反復法, 前処理, 大規模電磁場解析, フィルイン, IC(不完全コレスキー分解)

1. 緒言

辺要素有限要素法による時間調和渦電流解析[1]や高周波電磁場解析[2]などで得られる複素対称線形方程式の求解においては、反復法の収束性が悪いことが知られている。また、解析する対象が大規模・複雑になるにつれ収束性はさらに悪化する。対称線形方程式に対する反復法としては、Krylov 部分空間法の 1 つである共役勾配(Conjugate Gradient: CG)法やその複素数版である共役直交共役勾配(Conjugate Orthogonal Conjugate Gradient: COCG)法[3]が広く用いられてきた。近年は、共役残差(Conjugate Residual: CR)法やその複素数版である共役直交共役残差(Conjugate Orthogonal Conjugate Residual: COCR)法[4]や最小残差(MINimal RESidual: MINRES)法アルゴリズムに基づいた MINRES-like_CS 法[5]など新たな反復法の開発も進んでいる。

このような反復法において収束性を向上させるために前処理を施すことが一般的であり、対称線形方程式に対する前処理としては不完全コレスキー分解(IC, Incomplete Cholesky)前処理[6]がよく知られている。特に辺要素有限要素法による電磁場解析分野では、加速係数付き IC 前処理[7]が広く用いられ、加速係数の値は反復回数に大きく影響する重要なパラメータであり、問題ごとに最適な値を決定する必要がある。加速係数は前処理行列の対角成分に通常 1.0 よりも少し大きな値を掛け反復法の収束性を加速させるための係数であり、その値の自動決定に関する研究も行われている[8][9]が、大規模解析でも実用的な手法は確立されていない。また、Krylov 部分空間法の収束性は丸め誤差

の影響を受けるため、多倍長精度演算によってその影響を小さくすることで収束性改善が期待できることが経験的に知られている。特に複素対称向けの反復法では、多倍長精度演算を用いることで IC 前処理の加速係数や SSOR(Symmetric Successive Over-Relaxation)前処理の緩和係数など前処理のパラメータが収束性に与える影響を小さくできることが報告されている[10]。しかし、多倍長精度演算は計算コストが高いことから、全て倍精度演算で行った場合に比べて総計算時間を削減できるケースは限定される。

IC 前処理の多くはフィルインを行わない IC(0)が用いられ、非対称行列の問題も含めるとフィルインを許容する閾値を決めて不完全分解を行う ILUT 分解[11]や Inverse-based ドロッピング分解手法[12]などはいくつか研究があるが、電磁場解析向けの研究例は少ない。

これまで著者らはこれらの倍精度数を 2 つ用いた倍々精度演算を用いる事で複素対称線形方程式問題の収束性改善に成功し、倍精度と倍々精度の混合精度演算や IC(1)などのフィルインを用いることで従来の倍精度演算システムと比べて高速化に成功した[13][14]。一方で、その対象となる問題は主に自由度 13 万程度の中規模の行列であった。そのため、これらの手法が 100 万自由度以上の大規模問題に対しても有効であるかどうかは自明ではない。

そこで本研究では主に高周波電磁場解析を対象とし、得られる複素対称線形方程式に対して数値実験を行うことで、大規模電磁場解析向けの最適な手法について調査する。

1 名古屋大学 大学院情報学研究科
Nagoya University Graduate School of Informatics
2 日本学術振興会特別研究員-DC2
JSPS Research Fellowship for Young Scientists(DC2)
3 大同大学 情報学部
Daido University, Faculty of Informatics

4 名古屋大学 情報基盤センター
Nagoya University, Information Technology Center

2. 多倍長精度演算

2.1 倍々精度演算

倍々精度演算としては QD library[15]が広く利用されている。しかし、著者らの研究により、C++で書かれた QD library よりも、同等の機能を C 言語で構造体を用いて実装した方が高速となることが明らかになっている[10]。そこで本研究では構造体を用いた倍々精度数を実装し、その実装方法について述べる。

まず、実数の倍々精度数を実装するうえで、倍々精度数の上位ビットと下位ビットのための倍精度変数 2 つを含む構造体 `dd_real` を図 1 にように実装した。さらにこれを用いて倍々精度複素数の構造体 `dd_comp` を図 2 のように実装した。四則演算の実装例については文献[10]に示しているが、これらの変数を用いて倍々精度複素数演算を実装した。

```
struct dd_real{
    double hi; //上位ビット
    double lo; //下位ビット
};
```

図 1. 倍々精度実数の実装

```
struct comp{
    struct dd_real re; //実部
    struct dd_real im; //虚部
};
```

図 2. 倍々精度複素数の実装

2.2 混合精度演算

本研究は有限要素法による電磁場解析を対象としているが、一般的に有限要素解析では係数行列や右辺ベクトルの評価を行い、得られた線形方程式を反復法などで解く手順となる。今回は反復法による求解部分のみを多倍長精度演算の性能評価対象とするため、方程式の組み立て部分は倍精度演算で行われるものとする。つまり、線形ソルバの入力となる係数行列と右辺ベクトルは倍精度であり、例えば計算途中のベクトル変数を倍々精度数で実装した場合、Krylov 部分空間法の係数行列ベクトル積 $\mathbf{q} = \mathbf{A}\mathbf{p}$ では行列 \mathbf{A} は倍精度、ベクトル \mathbf{q} と \mathbf{p} は倍々精度となる。よって、倍精度と倍々精度の混合精度計算が必要となる。ここで、QD library は実数における倍精度と倍々精度などの混合精度演算は実装されており、C++から使う場合は演算子がオーバ

ーロードされている。しかし、`complex` クラステンプレートを利用する複素数では演算子のオーバーロード実装が十分ではなく、例えば倍精度複素数と倍々精度複素数の混合精度演算を行う場合は倍精度複素数を一旦倍々精度複素数の変数に格納して下位ビットをそれぞれ 0 にしたのち演算を行うなど効率的なコーディングが難しい問題がある。そのため、倍精度複素数と倍々精度複素数の混合精度演算のための実装をする必要がある。そこで、今回は倍精度複素数の構造体 `double_comp` を図 3 のように実装し、これを用いて複素数の混合精度演算を実現した。

```
struct double_comp{
    double re; //実部
    double im; //虚部
};
```

図 3. 倍精度複素数の実装

2.3 混合精度演算反復法

これまで著者らは中規模問題に対し、内積計算を高精度化した混合精度反復法により高速化に成功した[13]。図 4 に $\mathbf{A}\mathbf{x} = \mathbf{b}$ を解くための混合精度 COCG 法のアルゴリズムを示す。ただし、添え字の D と DD はそれぞれ倍精度数と倍々精度ということを表し、 ϵ は収束判定を、 (\mathbf{x}, \mathbf{y}) は内積 $\sum \bar{x}_i y_i$ を表している。内積計算において例えば α を計算するにあたり \mathbf{r} と \mathbf{z} の乗算は倍精度で行い、その足し合わせを倍々精度で行う。そして \mathbf{p} と \mathbf{q} の内積計算も同様に行った後、それらの除算も倍々精度で行うことで α を計算する。その後 α の下位ビットを捨てることで (0 にすることで) 倍精度として計算する。

```
1 calculate preconditioning matrix  $M_D^{-1}$ 
2 initial guess  $\mathbf{x}_D^0$ 
3  $\mathbf{r}_D^0 = \mathbf{b}_D - A_D \mathbf{x}_D^0$ 
4  $\mathbf{z}_D^0 = M_D^{-1} \mathbf{r}_D^0$ 
5  $\mathbf{p}_D^0 = \mathbf{z}_D^0$ 
6 for  $n = 0, 1, 2 \dots$  until  $\|\mathbf{r}_D^n\| / \|\mathbf{r}_D^0\| \leq \epsilon$ 
7  $\mathbf{q}_D^n = A_D \mathbf{p}_D^n$ 
8  $\alpha_{DD} = (\bar{\mathbf{r}}_D^n, \mathbf{z}_D^n)_{DD} / (\bar{\mathbf{p}}_D^n, \mathbf{q}_D^n)_{DD}$ 
9  $\mathbf{x}_D^{n+1} = \mathbf{x}_D^n + \alpha_{DD} \mathbf{p}_D^n$ 
10  $\mathbf{r}_D^{n+1} = \mathbf{r}_D^n - \alpha_{DD} \mathbf{q}_D^n$ 
11  $\mathbf{z}_D^{n+1} = M_D^{-1} \mathbf{r}_D^{n+1}$ 
12  $\beta_{DD} = (\bar{\mathbf{r}}_D^{n+1}, \mathbf{z}_D^{n+1})_{DD} / (\bar{\mathbf{r}}_D^n, \mathbf{z}_D^n)_{DD}$ 
13  $\mathbf{p}_D^{n+1} = \mathbf{z}_D^{n+1} + \beta_{DD} \mathbf{p}_D^n$ 
14 end for
```

図 4. 混合精度 COCG 法

3. 数値実験

3.1 高周波電磁場問題

高周波電磁界のモデルとして、変位電流を含む Maxwell 方程式から導かれる電場 E [V/m] を未知数とする波動方程式を考える[2].

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

ただし、 μ [H/m] は透磁率、 ϵ_0 [F/m] は真空の誘電率、 ϵ' は比誘電率、 ω [rad/s] は単一角周波数、 i は虚数単位、 J [A/m²] は電流密度である。式(1)に全周囲で電場 E の接線方向がゼロとなる境界条件を与え、辺要素有限要素法を適用する。

テスト問題として、TEAM Workshop Problem 29 として知られる医療用リエントラント型空洞共振器の高周波電磁界解析を考える。

図 4 に形状を示すが、中心部には円盤状の誘電体ファントムを設置している。ただし、比誘電率 $\epsilon' = 80.0$ 、導電率 $\sigma = 0.52$ [S/m] とする。また、単一角周波数は $\omega = 2\pi f$ とし、周波数 f を 300 [MHz] とした時の計算を行う。

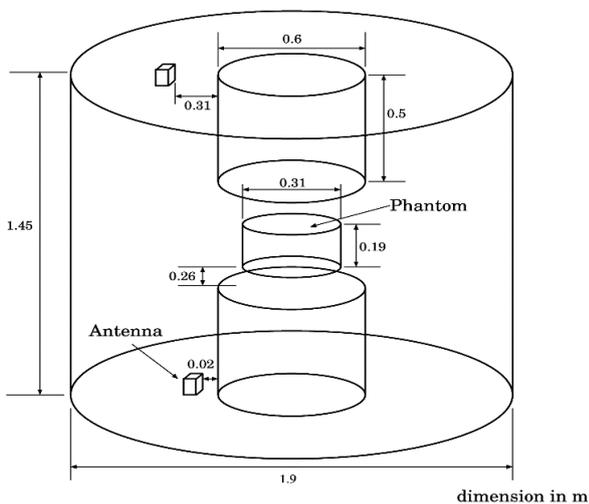


図 5 TEAM Workshop Problem 29 によるテストモデル[16]

3.2 性能評価

数値実験に用いる計算機は Intel Core i7-7700 3.60GHz の CPU と 64GB メモリで構成されており、コンパイラは gcc 7.3.0、最適化のオプションは -O3 -mfma とした。さらに並列化のためにコンパイル時に -fopenmp の引数を追加し、スレッド数は 8 とした。

また今回使用する行列のサイズとしては表 1 の行列を用いた。

表 1. 使用した行列のサイズと非ゼロ要素数

Size	非ゼロ要素数
439,176	7,036,670
979,464	15,794,744
1,646,787	26,651,193

また、今回は前処理として IC(0) および IC(1) を用い、反復法には COCG 法を用いる。また、計算精度としては従来の倍精度手法(double_Complex)と構造体を用いた倍精度手法(double_comp)、また内積計算のみを高精度化した混合精度手法(mixed)の 3 つを用いる。

まず、表 2 に、それぞれの行列に対して IC(0) と IC(1) を用いた時の前処理行列 L の非ゼロ要素数の変化を示す。ただし、 L は IC 分解を用いて $A \equiv LDL^T$ とした時の下三角行列である。 (L^T) は L の転置行列を表す

表 2 より、フィルインを用いる事で前処理行列の非ゼロ要素数が 2~3 倍程度に増えることが分かった。

表 2. フィルインによる前処理行列の非ゼロ要素数の変化

Size	非ゼロ要素数	
	IC(0)	IC(1)
439,176	3,737,923	9,853,214
979,464	8,387,104	22,268,605
1,646,787	14,148,990	37,688,671

次に同様に解くべき方程式を $A\mathbf{x} = \mathbf{b}$ とし、COCG 法で解くことを考える。ただし、 A は複素対称行列であり、初期解 $\mathbf{x}^0 = \mathbf{0}$ とする。収束判定は相対残差ノルム $\|\mathbf{r}^n\|_2 / \|\mathbf{r}^0\|_2$ が 10^{-9} 以下となったときとした。これらについても同様に、IC 前処理のフィルインレベル及び計算精度が計算性能に与える影響を評価する。IC 前処理の加速係数は全てのケースにおいて 1.05, 1.1, 1.15 の 3 パターンを用いる。表 3 ~ 5 にそれぞれ行列サイズが 439,176 と 979,464 と 1,646,787 の時の結果を示す。

表 3~5 より、フィルインを用いる事で、反復回数を約 1/4 から 1/2 に削減することに成功した。一方で、最適な加速係数は IC(0) では 1.1 または 1.15 であったのに対し、IC(1) ではすべての場合において 1.05 であった。計算時間においては構造体の倍精度複素数(double_comp)を用いる事で double_Complex に比べて全てのケースにおいて高速化に成功している。また、混合精度手法を用いる事で、おおよ

その場合で高速化に成功しており、全てのケースにおいてフィルイン付きの混合精度手法の結果が最速となった。さらに、前処理に IC(0)を用いて計算精度に double_Complex を用いていた従来の手法と比較すると、IC(1)を用いて混合精度演算を用いた提案手法は反復回数を約 74%，計算時間を約 65%短縮することに成功し、大規模問題においても本手法が有効であるということが明らかとなった。

さらに、今回の混合精度手法において、高精度化したのは内積計算のみであるため、図 4 の変数を用いると、p と q および r と z の内積計算のための変数と α と β のみを倍々

精度変数にすればよい。これらの変数は全てスカラであるため、メモリ使用量は全て倍精度演算とした時と比べてほぼ同じであり、反復法全体でおおよそ $20nnz + 20nnz_L + 104N$ バイトで実現することができる。ただし、nnz は係数行列の非ゼロ要素数、nnz_L は前処理行列の非ゼロ要素数であり、N は行列の自由度である。今回用いた自由度 164 万の問題においてメモリ使用量はおおよそ 1.6GByte であるため、これらの問題は一般の計算機においても十分実装可能だということが明らかになった。

表 3. 行列サイズ 439,176 の行列に対する性能評価

	Acceleration Factor	IC(0)		IC(1)	
		Iteration Counts	Time [sec]	Iteration Counts	Time [sec]
double_Complex	1.05	25,766	1,706	7,131	920
	1.1	22,299	1,484	9,071	1134
	1.15	22,800	1,512	11,284	1385
double_comp	1.05	25,011	997	7,143	552
	1.1	22,493	889	9,080	670
	1.15	22,656	892	11,267	805
mixed	1.05	24,179	979	6,851	536
	1.1	21,614	882	8,784	657
	1.15	22,025	901	10,979	794

表 4. 行列サイズ 979,464 の行列に対する性能評価

	Acceleration Factor	IC(0)		IC(1)	
		Iteration Counts	Time [sec]	Iteration Counts	Time [sec]
double_Complex	1.05	49,449	7,312	14,060	4,091
	1.1	41,531	6,147	18,468	5,213
	1.15	40,234	5,940	22,801	6,335
double_comp	1.05	49,418	4,423	14,122	2,476
	1.1	41,513	3,720	18,326	3,053
	1.15	40,522	3,627	21,733	3,528
mixed	1.05	48,249	4,440	13,682	2,445
	1.1	39,998	3,681	17,150	2,932
	1.15	38,905	3,586	21,339	3,532

表 5. 行列サイズ 1,646,787 の行列に対する性能評価

	Acceleration Factor	IC(0)		IC(1)	
		Iteration Counts	Time [sec]	Iteration Counts	Time [sec]
double _Complex	1.05	64,370	16,858	28,170	13,918
	1.1	55,384	14,544	30,703	15,018
	1.15	51,245	13,167	34,727	16,829
double_comp	1.05	64,886	9,882	28,340	8,075
	1.1	55,405	8,489	30,820	8,781
	1.15	51,401	7,991	34,961	9,721
mixed	1.05	63,601	10,445	26,883	7,971
	1.1	54,285	8,540	30,166	8,680
	1.15	50,880	8,040	34,115	9,686

4. 結言

今回の研究では、実際のシミュレーションシステムへの組み込みへ向けて、大規模問題に対して数値計算を行った。その結果、以下の知見が得られた。

- OpenMP による並列化により、計算時間の短縮に成功
- 大規模問題においても IC(1)は有効
- 従来のフィルイン無し倍精度手法(IC(0)前処理 double _Complex)と比べて最速で計算時間を約 1/3 程度まで短縮することに成功
- 計算精度や問題規模によって最適な加速係数が異なる
- 提案手法は高速化に成功している上、メモリ消費量の増大を抑え、一般の計算機でも十分実装が可能であることが明らかとなった

今後は実際の解析システムの高速化に向けてさらに大規模な問題にも適用していくとともに、アプリケーションシステムに組み込んで性能評価を行うことで、本システムの有効性をさらに検討し、高効率な大規模解析システムの開発に貢献していく。

謝辞 本研究の一部は JSPS 科研費 JP19J12838 の支援を受けたものである。

参考文献

[1] 杉本振一郎, 金山寛, 浅川修二, 吉村忍. 階層型領域分割法を用いた 4,400 万複素自由度の時間調和渦電流解析. 日本計算工学会論文集, 2007, Paper No.20070027.

[2] 武居周, 吉村忍, 金山寛. 階層型領域分割法による高周波電磁場の大規模解析. 電気学会論文誌 A, 2008, 128-A, 9, p. 591-597.

[3] van der Vorst, H.A. and Melissen, J.B.M.. A Petrov-Galerkin type method for solving $Ax=b$ and where A is symmetric complex. IEEE Trans. Magn., 1990, 26, p. 706-708.

[4] Sogabe, T. and Zhang, S.L.. A COCR method for solving complex symmetric linear systems. J. Comput, 2007, Appl. Math., 199-2, p. 297-303.

[5] 荻野正雄, 武居周, 野津裕史, 杉本振一郎, 吉村忍. 高周波電磁界シミュレーションにおける複素対称行列向けの反復解法の性能評価. 日本計算工学会論文集, 2014, Paper No. 20140017.

[6] Meijerink J. A. and van der Vorst H. A.. An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix. Mathematics of Computation, 1977, Vol. 31, No. 137, p. 148-162.

[7] Fujiwara, K., Nakata, T. and Fusayasu, H.. Acceleration of Convergence Characteristic of the ICCG Method. IEEE Trans. Magn., 1993, Vol. 29, No. 2, p. 1958-1961.

[8] Takada A., Noguchi, S. and Igarashi, H.. A New Acceleration Factor Decision Method for ICCG Method Based on Condition Number. IEEE Trans. Magn., 2012, Vol. 48, No. 2, p. 519-522.

[9] Kitao, J., Takahashi, Y., Fujiwara, K., Mifune, T., and Iwashita T.. Automatic Determination of Acceleration Factor Based on Residual and Functional in Shifted ICCG Method for 3-D Electromagnetic Field Analyses. IEEE Trans. Magn., 2013, Vol. 49, p. 1741-1744.

[10] 梶井晃基, 荻野正雄. 複素対称線形方程式における多倍長精度共役直交共役勾配法の性能評価. 日本計算工学会論文集, 2018, Paper No, 20180007.

[11] Saad, Y. A Dual Threshold Incomplete LU Factorization. Numerical Linear Algebra with Applications, 1994, Vol. 1, p. 387-4-2.

[12] Bollhoefer, M.. A Robust ILU with Pivoting based on Monitoring the Growth of the Inverse Factors. Linear Algebra and Its Applications, 2001, Vol. 338, p. 201-218.

[13] Koki Masui, Masao Ogino, "Research on the Convergence of Iterative Method Using Mixed Precision Calculation Solving Complex Symmetric Linear Equation", IEEE Trans. Magn., 2020, Vol. 56, No. 1, p. 1-4.

- [14] 榎井晃基, 曾我部知広, 片桐孝洋, 荻野正雄, “大規模電磁界解析向けの線形方程式のための IC 分解前処理のフィルインレベルについて”, 情報処理学会 第 172 回 HPC 研究会, 沖縄産業支援センター, 2019 年 12 月.
- [15] Y. Hida, X. S. Li, D. H. Bailey, “Library for double-double and quad-double arithmetic”, NERSC Division, Lawrence Berkeley National Laboratory, 2007.
- [16] Y. Kanai, “Description of TEAM Workshop Problem 29”, Whole body cavity resonator, TEAM Workshop in Tucson, 1998.