連立一次方程式の成り立ちと悪条件問題に対する解法

桝井　晃基

シミュレーションはさまざまな場面で有用な手段であるが，その種類によっては処理に数か月以上かかる問題もある．数値シミュレーションの多くは，その過程で連立一次方程式を解く必要がある．方程式の自由度や解きやすさは問題の種類によって大きく変わり，多くの解法が存在する．本稿では，連立一次方程式の成り立ちと，特に大規模かつ悪条件問題に対する解法について述べ，数値実験例を示す．

キーワード：反復法，連立一次方程式，電磁場解析

1.　はじめに

現代社会において，数値シミュレーションは，産業や科学技術の発展に不可欠な要素となっている．たとえば，自動車や航空機の設計，通信技術の進展，さらには医療機器の開発など，多岐にわたる分野で活用されている [1, 2]．数値シミュレーションは，実際の試作品を製作することなく，短時間かつコンピュータ上で性能を評価できるため，開発コストや時間の大幅な削減が可能となる．しかし，シミュレーションは現実の物理現象を数値的に模擬したものであり，必ずしも実際の結果と完全に一致するとは限らない．そのため，理論値と実測値の間に差異が生じる可能性がある．

近年，より高精度なシミュレーションを実現するために，より大規模な問題の解析が行われている [3]．しかしながら，問題の規模が大きくなるにつれ，計算時間が増加し，シミュレーションの利便性が損なわれるという課題が生じている．したがって，高精度なシミュレーションを効率的に行うためには，その高速化が不可欠である．図 1 にシミュレーションの流れを示す．このように，数学的モデル化を行い，数値計算を経た後，可視化により各点に加わる力を視覚的に確認できる．シミュレーションの処理時間のうち，連立一次方程式の求解が大半を占めることが多いため，求解の高速化に対する需要が高まっている．本稿では，シミュレーションの基本的な流れを概説し，連立一次方程式の成り立ちと数値解法，およびその効率的な手法について述べる．

ますい　こうき
大阪大学大学院情報科学研究科
〒 565–0871 大阪府吹田市山田丘 1–5
masui@ist.osaka-u.ac.jp

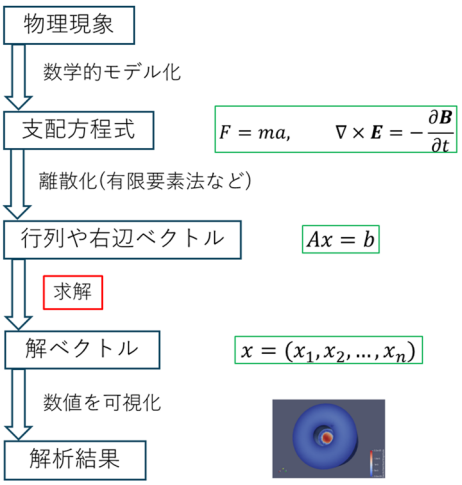

図 1　シミュレーションの流れ

2.　連立一次方程式の成り立ち

連立一次方程式はある支配方程式から有限要素法 (FEM) などの離散化により得られることが多い．支配方程式には，物体に働く力と加速度の関係を表した運動方程式や，電磁場に関する法則を表すマクスウェル方程式など分野によってさまざまな方程式がある．これらの方程式を用いて物体にかかる力などを求めることができるが，現実にある物体は連続体かつ複雑な形状をしていることが多く，単純な形状の物体を除き，直接解析することは困難である．そこで有限要素法などでは，物体を細かいメッシュに分割し，そのメッシュの集合体により元の物体を近似して表現する．メッシュの頂点を節点と呼ぶが，各メッシュの節点にかかる力を一つ一つ計算することで解析領域全体にどのように力がかかっているかを求めることができる．

分割するメッシュの形状は手法によってさまざまで

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

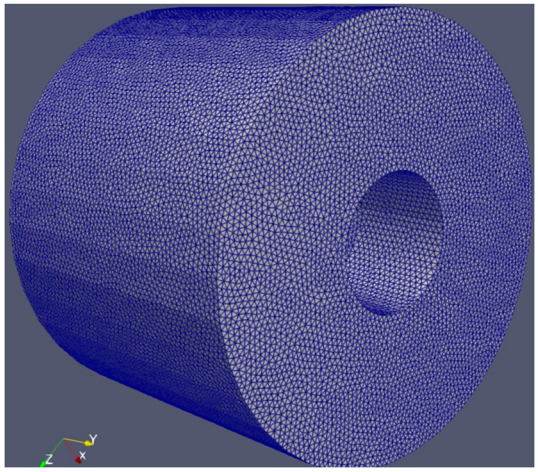

図 2　四面体のメッシュに分割されたモデル

あるが，2 次元平面では三角形や四角形に，3 次元では図 2 のように四面体，もしくは六面体に分割されることが多い．図 2 はがん治療などで用いられる機器のモデルの一部であり，100 万個以上のメッシュに分割されている [4]．一般的にメッシュは細かく分割したほうが元の形状に近くなるため解析の精度が良くなるが，その分計算すべき節点の数が増えるため，計算コストは増加する．実際はこのトレードオフを考慮しながらメッシュの分割方法を考える必要がある．そして各節点と影響を及ぼしている節点との関係式を立て，その足し合わせが全体の行列となる．

例として，図 3 のように複数のバネがつながっている状態において，ある力 F で引っ張った場合，それぞれのバネがどの程度伸びるか知りたいとする．ここで，k_1, k_2 はそれぞれのバネ定数である．u_1 から u_3 がそれぞれの変位であり，これらの差により，それぞれバネがどのくらい伸びたのかを調べることができる．

これを図 4 のように，まず一つずつのバネに分解する．左側のバネに着目すると，フックの法則により，式 (1), (2) が導かれる．この 2 式は行列とベクトルの形に変形することができ，式 (3) のようになる．このときのバネ定数による行列は要素剛性マトリクスと呼ばれる．

$$f_{11} = k_1(u_1 - u_2) \tag{1}$$

$$f_{21} = k_1(-u_1 + u_2) \tag{2}$$

$$\begin{pmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} \tag{3}$$

同様に，右側のバネに着目し，f_{22} と f_{32} について

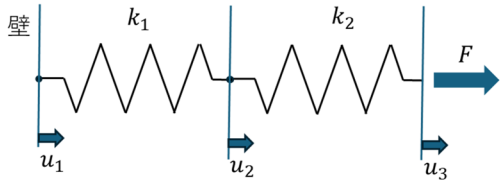

図 3　複数つながったバネ

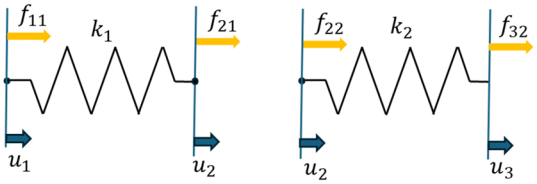

図 4　要素ごとに分割されたバネ

も方程式をたてることで式 (4) が導かれる．

$$\begin{pmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_{22} \\ f_{32} \end{pmatrix} \tag{4}$$

式 (3), (4) を重ね合わせ，$f_{21} + f_{22} = 0$ となるから，この 2 式は式 (5) のようにまとめることができる．

$$\begin{pmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_{11} \\ 0 \\ F \end{pmatrix} \tag{5}$$

ここで，u_1 は壁と接しており動かないことから，境界条件である $u_1 = 0$ を与えることで，最終的に解くべき方程式は式 (6) になる．

$$\begin{pmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 \\ F \end{pmatrix} \tag{6}$$

このように個々の要素剛性マトリクスを重ね合わせることで最終的な係数行列を生成する．最終的にこの方程式を解くことで，解（今回はすべてのバネの伸び）を調べることができる．単純な構造の場合は，係数行列は図 5 のように対角付近に非ゼロ要素が集まり，それ以外はゼロとなる疎行列となることが多い．一般的には対角付近に非ゼロ要素が集中しているほうが方程式が解きやすくなるため，行列のバンド幅が小さくなるように節点番号を入れ替えるといった手法もある [5]．

3.　連立一次方程式の求解

A を n 次正方行列，x, b を n 次元ベクトルとして，解くべき連立一次方程式を $Ax = b$ とする．これを解くための手法は大きく分けて直接法と反復法の 2 種類に分類される．

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

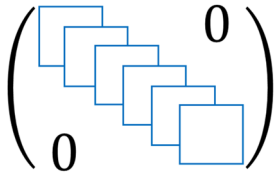

図 5　要素剛性マトリクスの集合（係数行列）

直接法は，ガウスの消去法や LU 分解に代表される手法であるが，A の逆行列 A^{-1} を求め，$x = A^{-1}b$ を計算することで，解を求める手法である．行列の要素のうち大部分がゼロのものを疎行列，逆にゼロ以外の要素が多く入っているものを密行列と呼び，直接法は A が疎行列か密行列かにかかわらず有限回の計算回数 $(O(n^3))$ で解を求めることができる．しかし，行列の自由度 (n) が大きくなると計算時間が増加し，メモリ使用量 $(O(n^2))$ も膨大になるため，実用時間内に解を求めることが難しいといった問題がある．

反復法は初期解 x_0 からスタートし，残差ノルム $||r|| = ||b - Ax||$ がある一定の閾値以下になったときの x を解とする手法である．この閾値は求められる精度によって異なる．反復法は，係数行列 A が疎行列の場合，計算すべき箇所が行列の非ゼロ要素数分だけで済むため，大規模問題においてよく用いられる．また，行列ベクトル積がメインの計算となるため，並列計算に向いている．一方，収束までの反復回数は問題の種類，すなわち係数行列によって大きく異なり，条件数が大きい（悪条件な）問題では収束しない場合がある．反復法には，共役勾配法（conjugate gradient method，以降 CG 法と記す） [6] や Jacobi 法などといったソルバの種類や，対角スケーリングや不完全コレスキー前処理（incomplete cholesky preprocessing；以降 IC 前処理と記す）などの前処理手法のほかに，演算精度などのパラメータがあり，問題によってどのような手法を用いるかを適切に選択する必要がある．

本稿では，演算精度と前処理手法について，それらの概要と電磁場解析問題に対する数値実験例を示す．今回扱う電磁場解析問題は，係数行列が複素数かつ収束性が悪いことが知られている．

3.1　高精度演算

一般的に数値計算において，浮動小数点数を表現する場合は IEEE 754 準拠の倍精度浮動小数点数 (double) が使われる．おおよその問題に対しては，この程度の精度で十分であり，深層学習などの分野では単精度浮動小数点数 (single) やそれ以下の精度の数値が用いられることもある．しかし，科学技術計算分野などにおいて大規模計算が必要となる場合に，丸めによる誤差が蓄積していくことで計算結果の精度が大きく低下するといった問題がある．特に，反復法として広く用いられる CG 法 [6] や共役残差法（conjugate residual method，以降 CR 法と記す） [7]，またその複素数版である共役直交共役勾配法（conjugate orthogonal conjugate gradient method，以降 COCG 法と記す） [8] や共役直交共役残差法（conjugate orthogonal conjugate residual method，以降 COCR 法と記す） [9] などの Krylov 部分空間法は丸め誤差の影響を強く受けることが知られている．これらの反復法は理論的には行列の自由度 n 回以下の反復で収束することが証明されているが，数値実験においてはそれ以上の反復を要する，あるいは収束しない場合も存在する．

これに対して，高精度演算を用いることで収束性を改善し，安定的に解を得ることができる．ここでいう高精度演算は倍精度演算よりも多倍長な精度，たとえば倍々精度（擬似 4 倍精度）や 8 倍精度などのことを指す．高精度演算を実現したライブラリはさまざまな種類のものが知られている [10]．しかし，それらの多くはソフトウェア実装であるため，ハードウェアで実装されている倍精度演算と比べると，必要なメモリ量や演算量が増加し，一つの演算にかかるコストが 10 倍以上に増加することもある．これらは時間がかかっても安定的に解を求めるときなどに利用されることが多い．

一方で，計算速度も求める場合は倍精度数を二つ用いた倍々精度数がよく用いられる．これは 4 倍精度に近い精度をもち，倍精度演算ベースであるため，ハードウェアによる高性能な処理が期待できる．図 6 に倍精度，4 倍精度，倍々精度の構造を示す．IEEE 754 準拠の倍精度数は符号部 1 ビット，指数部 11 ビット，仮数部 52 ビットで表され，10 進数でおよそ 16 桁の精度をもつ．一方，4 倍精度数は符号部 1 ビット，指数部 15 ビット，仮数部 112 ビットで表されるため，10 進数でおよそ 34 桁の精度をもつ．

倍々精度数では，図 6 のように二つの倍精度数を用いてそれぞれ上位ビットと下位ビットのように割り当てるため，仮数部が合計で 104 ビットとなり，10 進数でおよそ 32 桁の精度となる．4 倍精度と比べて少し精度は劣るが，（実装の方法にもよるが）倍精度と比べて 2〜3 倍程度の計算時間に抑えることができる．

複素数演算においては，倍精度演算はたとえば，C 言語では double _Complex などの型を使うことができる．一方，倍々精度演算は標準では実装されていないが，QD library などのライブラリに実装されてお

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

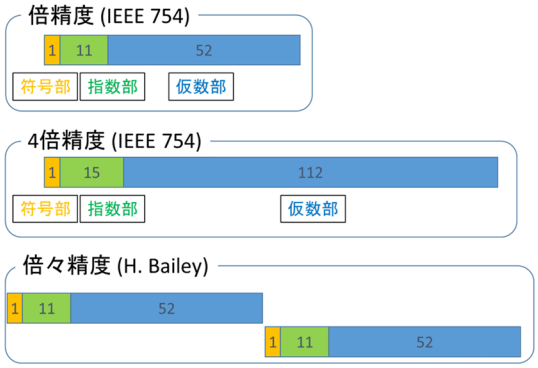

図 6　浮動小数点数

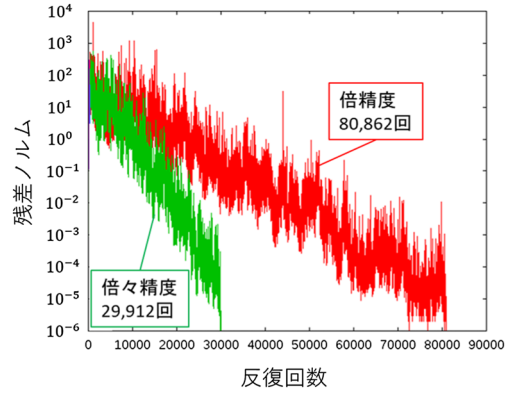

図 7　高精度演算適用による反復回数削減

り，C 言語などで使うことができる [10].

反復法へ高精度演算を適用し，COCG 法を計算したときの数値実験例を図 7 に示す．図の横軸は反復回数，縦軸は残差ノルムであり，残差ノルムが 10^{-6} 以下となったときに収束としている．係数行列は，行列のデータベースである SuiteSparse Matrix Collection [11] より行列 qc2534 を使用している．この行列は電磁場解析で現れる問題であり，複素対称行列である．また，この行列は自由度 2,534 の行列であるため，COCG 法は理論的には 2,534 回以内の反復で計算が終了する．

しかし，図 7 からわかるように倍精度演算を用いた場合，丸め誤差の影響により反復回数が 80,862 回にまで増加している．これに対し，高精度演算を用いることで丸め誤差の影響を減らし，倍精度に比べ反復回数を大きく削減した．このことから高精度演算の有用性が示されている．

ただし，1 反復あたりの計算時間は倍々精度演算のほうがかかるため，総計算時間は（このケースにおいて最適化があまりなされていないこともあるが，）倍々精度のほうがわずかに早い程度となっている．そのため，すべての変数を倍々精度とするのではなく，精度が必要な所にのみ倍々精度を適用した混合精度演算による反復法が計算時間の観点で有効であり，内積計算のみを倍々精度にした反復法により高速化に成功した研究例もある [12]．混合精度演算による研究例はさまざまな種類のものがあり，低精度演算を組み合わせた高速化の例もある [13].

3.2　前処理手法

反復回数を削減する手法として前処理 (Preconditioning) も非常に有用である．反復法において実用上は前処理をかけることが一般的である．前処理つき反復法では，前処理行列を M として，$Ax = b$ の代わりに $M^{-1}Ax = M^{-1}b$ などのように変形することで，係数行列の条件数を下げることができる．このとき，$M = A$ であれば解くべき方程式の係数行列は単位行列となるため，方程式がすぐ解けることになる．ただし，M^{-1}，すなわち A^{-1} を計算することになるため，直接法のときと同様に大規模な密行列の計算をしなければならない．対称行列に対する前処理としては，対角スケーリング，SSOR 前処理，IC 前処理が広く用いられる．この中で IC 前処理は収束性をより改善するため，悪条件問題に対して有用である．

IC 分解は，対称行列 A を下三角行列 L とその転置行列 L^T を用いて，$LL^T \approx A$ と近似する手法である．IC 前処理はこの LL^T を前処理行列とする手法である．IC 分解では，フィルインという要素がある．これは，疎行列においてゼロであった要素が分解の過程で非ゼロになることを意味する．フィルインをどの程度許容するかによって，反復法の収束性や 1 反復あたりの計算コストが大きく変化する．フィルインを許容する IC 分解では，フィルインレベルを p とし，IC(p) のように表す [14]．図 8 が係数行列の非ゼロ要素分布を表す場合，IC 分解成分 L における前処理行列の非ゼロ要素分布は図 9 のようになる．ただし，図の黒丸は行列の非ゼロ要素を表し，赤丸がフィルインによって追加された非ゼロ要素を表す．

$p = 0$，すなわちフィルインを許容しない IC(0) 前処理は図 9（左）のように，L と A の下三角部分における非ゼロ要素位置は一致する．一方で，フィルインを許容する IC(1) 前処理は図 9（右）のようになり，係数行列ではゼロであった要素位置に前処理行列には非ゼロの値が入ることになる．問題の種類にもよるが，フィルインのレベルを 1 上げるごとに前処理行列 L の非ゼロ要素数が 2〜3 倍に増加する．

数値実験例として，高周波電磁場解析問題に対する性

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

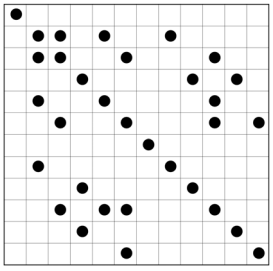

図 8　係数行列 A の非ゼロ要素分布

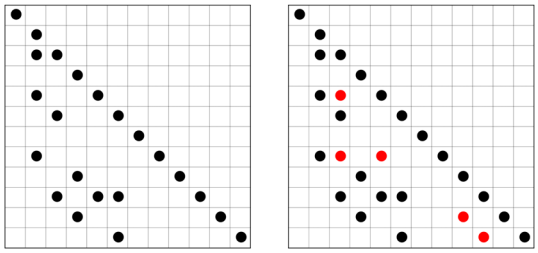

図 9　前処理を IC(0)（左）と IC(1)（右）としたときの IC 分解成分 L の非ゼロ要素分布

能を評価する．モデル本体の形状は図 2 に示したものであるが，電磁波の影響が及ぶ周りの空気領域も解析領域となる．これらを含めてメッシュ分割したものから係数行列を作成しているため，行列の自由度は 1,007,028 と大規模な問題となっている．IC(0) と IC(1) をそれぞれ前処理とした COCG 法の収束履歴を図 10 に示す．横軸が反復回数，縦軸が残差ノルムを示している．ここでは収束判定は 10^{-9} とし，すべての計算を倍精度演算としている．図 10 より，フィルインを許容する IC(1) 前処理により反復回数を大きく削減している．ただし，高精度演算のときと同様に，この例では IC(1) 前処理は IC(0) と比べて 1 反復あたりの計算時間が 2 倍近くかかるため，総計算時間は 3 割程度の削減となっている．

計算時間の観点で最適なフィルインレベルは問題によって異なり，フィルインを許容しない IC(0) が最適なこともあれば，さらにフィルインレベルを上げた IC(2) や IC(3) が最適となるケースもある [15]．たとえば行列の条件数を調べることで最適なフィルインレベルはある程度わかるが，大規模問題においてはそもそも条件数を求めるのが難しい（または方程式を解くのと同程度の計算コストがかかる）こともあり，数値実験をしてみないとわからない場合もある．また，前処理行列の非ゼロ要素の中で，収束性に影響の小さいものを棄却，つまりゼロとすることで，フィルインの数を調

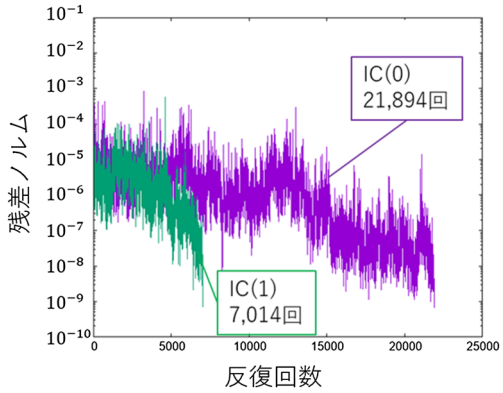

図 10　IC(1) 前処理による反復回数削減

整し，1 反復あたりの計算時間を短縮するドロッピングという手法もある [16]．

4.　おわりに

本稿では，連立一次方程式の成り立ちと，その解法について述べ，数値実験にて高精度演算と前処理手法により反復回数を削減する例を示した．数値実験の例はこれまでに私が研究してきた内容で，見ていただきたい部分でもあるが，それに加え，中学校や高等学校で学習する連立一次方程式がシミュレーションとどのように関連しているかについて，少しでも関心をもっていただけると幸いである．

計算の高速化という観点においては，最新の CPU や GPU を活用した並列化が不可欠であるが，一方で，アルゴリズムの開発や並列化に向けた反復法についての研究は今も盛んであり，世界に目を向けると隔年で Copper Mountain Conference On Iterative Methods [17] という国際会議が開催されている．さらに前処理手法に焦点をあてた Preconditioning [18] という国際会議もあり，こちらも隔年で開催されている．興味がある方はウェブサイトなどを覗いてみてはいかがだろうか．

謝辞　本研究は JSPS 科研費 24K20780 の助成を受けたものである．また，本記事の執筆機会を提供してくださった東京都市大学の相原研輔氏と，編集委員を担当してくださった九州大学の伊豆永洋一氏に謝意を表したい．

参考文献

[1] 武居周，室谷浩平，吉村忍，金山寛，“数値人体モデルを用いたマイクロ波帯域の有限要素電磁界解析，”日本シミュレーション学会論文誌，**4**, pp. 81–95, 2012.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

[2] A. Takei, S. Yoshimura and H. Kanayama, “Large-scale parallel finite element analyses of high frequency electromagnetic field in commuter trains,” *Engineering & Sciences*, **31**, pp. 13–24, 2008.
[3] S. Sugimoto, A. Takei and M. Ogino, “High-frequency electromagnetic field - heat conduction coupled analysis of numerical human body model with 270 million,” In *Proceedings of The Computational Mechanics Conference*, OS-1704, 2023.
[4] Y. Kanai, “Description of TEAM workshop problem 29: Whole body cavity resonator,” *TEAM Workshop in Tucson*, 1998.
[5] W. Liu and A. Sherman, “Comparative analysis of the cuthill–mcKee and the reverse cuthill–mcKee ordering algorithms for sparse matrices,” *SIAM Journal on Numerical Analysis*, **12**, pp. 198–213, 1976.
[6] M. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, NBS Washington, 1952.
[7] D. Luenberger, “The conjugate residual method for constrained minimization problems,” *SIAM Journal on Numerical Analysis*, **7**, pp. 390–398, 1970.
[8] H. Vorst and J. Melissen, “A petrov-galerkin type method for solving Ax=b and where A is symmetric complex,” *IEEE Transactions on Magnetics*, **26**, pp. 706–708, 1990.
[9] T. Sogabe and S. Zhang, “A COCR method for solving complex symmetric linear systems,” *Journal of Computational and Applied Mathematics*, **199**, pp. 297–303, 2007.
[10] D. Bailey, High-Precision Software Directory, https://www.davidhbailey.com/dhbsoftware/ （2025 年 2 月 20 日閲覧）
[11] T. Davis and Y. Hu, “The university of Florida sparse matrix collection,” *Association for Computing Machinery*, **38**, pp. 1–25, 2011.
[12] K. Masui and M. Ogino, “Research on the convergence of iterative method using mixed precision calculation solving complex symmetric linear equation,” *IEEE Transactions on Magnetics*, **56**, 7503604, 2020.
[13] Y. Zhao, T. Fukaya and T. Iwashita, “Numerical behavior of mixed precision iterative refinement using the BiCGSTAB method,” *Journal of Information Processing*, **31**, pp. 860–874, 2023.
[14] Y. Saad, *Iterative Methods for Sparse Linear Systems,* 2nd Editoin, SIAM Society for Industrial & Applied Mathematics, 2003.
[15] 桝井晃基，伊野文彦，“大規模電磁場問題に対する反復法におけるフィルインと加速係数の性能評価，”情報処理学会研究報告，Vol.2021-HPC-178, No. 15, 2021.
[16] M. Bollhoefer, “A robust ILU with pivoting based on monitoring the growth of the inverse factors,” *Linear Algebra and Its Applications*, **338**, pp. 201–218, 2001.
[17] Copper Mountain Conference On Iterative and Multigrid Methods, https://grandmaster.colorado.edu/copper/ （2025 年 2 月 20 日閲覧）
[18] International Conference On Preconditioning Techniques For Scientific and Industrial Applications, https://www.math.emory.edu/~yxi26/Precond24/index.html （2025 年 2 月 20 日閲覧）

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.