

複素対称問題向きへの E-SSOR 前処理の拡張

小玉 捷平¹ 藤野 清次² 岩里 洸介³

概要：反復法を用いて複素連立一次方程式 $Cz = b$ の数値解を効率的に求めることを考える．外部散乱波問題のような Helmholtz 方程式の問題では係数行列 C が複素数の行列要素を持つ．一般に，複素数の四則演算に要する計算量は実数のそれに比べて非常に多い．一方，係数行列が実対称行列のとき，E-SSOR 前処理は他の不完全分解型の前処理よりも計算量を低く抑えられる．本論文では，前処理行列 M の固有値をシフトさせることにより，E-SSOR 前処理を複素対称問題向けに拡張したシフト付き E-SSOR 前処理を提案する．数値実験を通じて他の前処理と収束性を比較し，提案法が優れていることを明らかにする．

キーワード：Eisenstat-SSOR，前処理，複素対称問題

Extension of E-SSOR preconditioning suited to complex symmetric problem

Abstract: We consider efficient numerical solutions of complex linear systems as the Helmholtz equations in the external scattering problems. In case of complex problems, the total amount of floating-point operations increases fairly compared with that of real problems. Accordingly it is crucial to choose optimally the preconditioning technique for the above complex problem. Therefore, we extend the so-called Eisenstat-SSOR preconditioning to be robust and efficient for the Helmholtz equations. Through numerical experiments, we verify the efficiency of the modified complex E-SSOR preconditioning.

Keywords: Eisenstat-SSOR, Preconditioning, Complex symmetric problem

1. はじめに

解くべき複素連立一次方程式を

$$Cz = b \quad (1)$$

とする．ただし， $C \in \mathbb{C}^{N \times N}$ を複素対称行列， $z \in \mathbb{C}^N$ を複素解ベクトル， $b \in \mathbb{C}^N$ を複素右辺ベクトルとする．この方程式に対する数値解法として反復法がよく用いられる．外部散乱波問題のような Helmholtz 方程式の問題では係数行列 C は複素数の行列要素を持つ．複素数の四則演算に要する計算量は実数のそれに比べて非常に多いので，係数行列が実数対称行列のときより収束までの時間が長い．した

がって，反復法の収束性の改善と計算量の削減は重要な課題である．反復法は前処理によって大幅に反復回数を削減出来る [11]．しかし，前処理は反復 1 回当たりの計算量が増加するため，場合によっては収束までの時間がそれ程減らないことがある．E-SSOR 前処理は，他の不完全分解型の前処理よりも計算量が少ないため，実対称行列である問題に対してかなりの改善効果があり，複素数の場合も有望であると思われる．

一方，係数行列が複素対称行列のとき，シフト付き IC 分解が非常に有用である．シフト付き IC 分解では，シフト量 α と呼ばれるパラメータを導入し，前処理行列を $A + \alpha I = U^T U - R$ と置いた．この処置により，元の行列の固有値分布を実軸付近から遠ざけ，元の IC 分解の収束性が改善されることが明らかにされた．

本論文では，前処理行列 M にシフト量を付加して，E-SSOR 前処理を複素対称問題向けに拡張した 3 種類のシフト付き E-SSOR 前処理を提案する．数値実験を通して，提案法の収束性が優れていることを明らかにする．

¹ 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, School of Engineering, Kyushu University
² 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University
³ 九州大学大学院システム情報学府情報学専攻
Graduate School of Information Science and Electrical Engineering, Kyushu University

本論文の構成は以下の通りである．第 2 節で，従来の複素対称行列用前処理について概説する．第 3 節で，3 種類のシフト付き E-SSOR 前処理について詳しく記述する．第 4 節で，数値実験を通して，3 種類のシフト付き E-SSOR 前処理の性能を評価し，改善効果を検証する．最後に，第 5 節でまとめを行う．

2. 従来の複素対称行列用前処理

2.1 E-SSOR 前処理

複素連立一次方程式

$$Cz = b \quad (2)$$

において，行列 C を

$$C = L + D + U \quad (3)$$

と分離する．ただし， $C \in \mathbb{C}^{N \times N}$ を複素対称行列， $z \in \mathbb{C}^N$ を複素解ベクトル， $b \in \mathbb{C}^N$ を複素右辺ベクトルとする．また， L, D, U は，行列 C の狭義複素下三角行列，複素対角行列，狭義複素上三角行列を各々意味し，行列の上付き添字 T は行列の転置を意味する． C は複素対称行列であるため，狭義複素上三角行列 U は L^T である．SSOR 前処理では，緩和係数と呼ぶ実数 ω を対角項に掛け，複素前処理行列 M を次式で与える．

$$M = (L + D/\omega)(D/\omega)^{-1}(L^T + D/\omega). \quad (4)$$

このとき，(2) 式は以下のように変換される．

$$\begin{aligned} (D/\omega)(L + D/\omega)^{-1}C(L^T + D/\omega)^{-1}(L^T + D/\omega)z \\ = (D/\omega)(L + D/\omega)^{-1}b \end{aligned} \quad (5)$$

前処理後の複素係数行列 $\tilde{C}$ ，複素解ベクトル $\tilde{z}$ ，複素残差ベクトル $\tilde{r}$ は各々

$$\tilde{C} = (D/\omega)(L + D/\omega)^{-1}C(L^T + D/\omega)^{-1}, \quad (6)$$

$$\tilde{z} = (L^T + D/\omega)z, \quad (7)$$

$$\tilde{r} = (D/\omega)(L + D/\omega)^{-1}r \quad (8)$$

と表される．このとき，前処理後の複素係数行列 $\tilde{C}$ を

$$\begin{aligned} \tilde{C} &= (D/\omega)(L + D/\omega)^{-1}C(L^T + D/\omega)^{-1} \\ &= (D/\omega)((L^T + D/\omega)^{-1} + (L + D/\omega)^{-1} \\ &\quad \times (I + (1 - 2/\omega)D(L^T + D/\omega)^{-1})) \end{aligned} \quad (9)$$

と式変形すると，前処理後の複素係数行列 $\tilde{C}$ と複素ベクトル v の積 $\tilde{C}v$ を次の 4 ステップで計算することで計算量を削減できる [1][3]．

$$1. \quad y = (L^T + D/\omega)^{-1}v \quad (10)$$

$$2. \quad u = v + (1 - 2/\omega)Dy \quad (11)$$

$$3. \quad w = (L + D/\omega)^{-1}u \quad (12)$$

$$4. \quad \tilde{C}v = (D/\omega)(y + w) \quad (13)$$

この前処理を従来型 E-SSOR 前処理と呼ぶ．

2.2 CSIC(Complex Shift IC) 分解とシフト処理

ここでは，閾値 (tolerance) による不完全コレスキー分解 (IC(tol) 分解) の改良である複素シフト付き IC 分解 (Complex Shifted IC 分解：以後，CSIC 分解と略す) について述べる．

一般に，係数行列が実対称行列のとき，IC 分解は上三角行列の対角要素の計算過程で平方根の中の値が負になり計算が中断することが時々起こる [8] [9]，複素対称行列の場合このような中断は起こり難い．しかしながら，行列の固有値が左半平面に存在する場合，不完全な分解のため，前処理後の行列の最小固有値が原点に接近し，複素反復法の収束性を悪化させる場合が起こる．図 1 に，前処理の前後の行列の固有値分布の変化の模式図を示す．閾値による IC 分解では，前処理後の行列の固有値分布は点 (1, 0) に接近する [2]，固有値の中には原点に急接近する場合があり，反復法の収束性が悪化する．

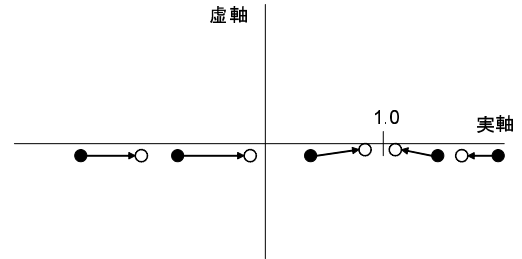


図 1 前処理の前 (黒丸) 後 (白丸) の行列の固有値分布の変化の模式図．

そこで，CSIC 分解と呼ばれる，前処理後の行列の最小固有値が原点に接近することを防止する IC 分解が提案された [6] [7]．この方法では分解前に係数行列の対角項に一定量の複素数を加えることで，あらかじめ係数行列の固有値分布を実軸付近から遠ざけ，IC 分解を適用する．図 2 は予め係数行列の固有値分布を実軸上付近から遠ざけることを表した模式図である．この場合のシフト量は負の値である．このようなシフト処理により，固有値の原点への接近を防止できると思われる．

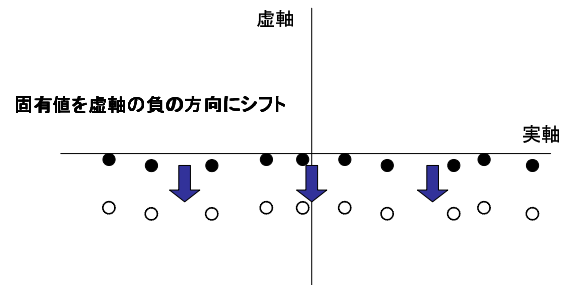


図 2 負の方向にシフトしたときの係数行列の固有値分布の変化の模式図．

複素行列に対する IC(tol) 分解と CSIC 分解を行列表現

で表すと以下になる．ただし， R は分解の不完全さを形式的に表す行列， I は単位行列， i は虚数単位， α は CSIC 分解でのシフト量を意味する実数パラメータとする．

$$C = LL^T - R, \quad (14)$$

$$C + \alpha i I = LL^T - R. \quad (15)$$

式 (14) と式 (15) における下三角行列 $L = [l_{ij}]$ の具体的な要素の計算は，各行 $i = 1, 2, \dots, n$ に対して，以下に示す手順で行う．ただし，要素 a_{ij} は係数行列 C の要素を，* 印付きの a_{ij}^* は分解過程において作業用として使う配列の要素を各々表し， tol は棄却判定に用いる正の定数とする．

[下三角行列 $L (= [l_{ij}])$ の計算手順]

$$a_{ii}^* = \begin{cases} a_{ii} & (\text{IC 分解での対角項の設定}) \\ a_{ii} + \alpha i & (\text{CSIC 分解での対角項の設定}) \end{cases}$$

$$a_{ij}^* = a_{ij} - \sum_{k=1}^{i-1} l_{ki} l_{kj},$$

$$l_{ii} = \sqrt{a_{ii}^* - \sum_{k=1}^{i-1} l_{ki}^2},$$

$$l_{ij} = \begin{cases} a_{ij}^* / l_{ii} & |a_{ij}^*| \geq \text{tol} \text{ のとき} \\ 0 & |a_{ij}^*| < \text{tol} \text{ のとき} \end{cases} \quad (j = i + 1, \dots, n).$$

CSIC 分解のアルゴリズムを以下に示す．要素 $\overline{a_{ii}}$ ， $\overline{a_{jj}}$ は行列 C の第 i ， j 行の対角要素に各々対応し，分解過程で更新される．一方， a_{ij}^* は最終的に下三角行列 L の非対角要素 l_{ij} になる要素を表す． α はシフト量を表す実数パラメータ， tol は要素 l_{ij} の棄却判定用閾値を意味する．

for $i = 1, \dots, n$

$$\overline{a_{ii}} = a_{ii} + \alpha i$$

end for

for $i = 1, \dots, n$

for $j = i + 1, \dots, n$

$$a_{ij}^* = a_{ij}$$

end for

for $k = 1, \dots, i - 1$

for $j = i + 1, \dots, n$

$$a_{ij}^* = a_{ij}^* - l_{ki} l_{kj} \quad (\text{非対角項の計算})$$

end for

end for

for $j = i + 1, \dots, n$

if $|a_{ij}^*| \leq \text{tol}$ then

$$a_{ij}^* = 0 \quad (\text{閾値より小さい要素を棄却})$$

end if

end for

$$l_{ii} = \sqrt{\overline{a_{ii}}} \quad (\text{対角項を求める})$$

for $j = i + 1, \dots, n$

$$l_{ij} = a_{ij}^* / l_{ii} \quad (\text{非対角項を求める})$$

$$\overline{a_{jj}} = \overline{a_{jj}} - l_{ij}^2 \quad (\text{対角項の計算})$$

end for

end for

3. 3種類のシフト付き E-SSOR 前処理

3.1 対角シフト付き E-SSOR 前処理_v1

複素前処理行列 M_{DS1} を以下のように定義する．

$$M_{\text{DS1}} = (L + D i / \omega)(D / \omega)^{-1}(L^T + D i / \omega). \quad (16)$$

式中の i は虚数単位を意味し， $i^2 = -1$ である．このとき，(2) 式は以下のように変換される．

$$\begin{aligned} & (D / \omega)(L + D i / \omega)^{-1} C (L^T + D i / \omega)^{-1} (L^T + D i / \omega) z \\ &= (D / \omega)(L + D i / \omega)^{-1} b \end{aligned} \quad (17)$$

前処理後の複素係数行列 $\tilde{C}$ ，複素解ベクトル $\tilde{z}$ ，複素残差ベクトル $\tilde{r}$ は各々

$$\tilde{C} = (D / \omega)(L + D i / \omega)^{-1} C (L^T + D i / \omega)^{-1}, \quad (18)$$

$$\tilde{z} = (L^T + D i / \omega) z, \quad (19)$$

$$\tilde{r} = (D / \omega)(L + D i / \omega)^{-1} r \quad (20)$$

と表される．このとき，前処理後の複素係数行列 $\tilde{C}$ を

$$\begin{aligned} \tilde{C} &= (D / \omega)(L + D i / \omega)^{-1} C (L^T + D i / \omega)^{-1} \\ &= (D / \omega)((L^T + D i / \omega)^{-1} + (L + D i / \omega)^{-1} \\ &\quad \times (I + (1 - 2 i / \omega) D (L^T + D i / \omega)^{-1})) \end{aligned} \quad (21)$$

と式の変形をすると，複素係数行列 $\tilde{C}$ と複素ベクトル v の積 $\tilde{C}v$ を次の 4 ステップで計算することで計算量を削減できる．

$$\left. \begin{aligned} 1. \quad & y = (L^T + D i / \omega)^{-1} v \\ 2. \quad & u = v + (1 - 2 i / \omega) D y \\ 3. \quad & w = (L + D i / \omega)^{-1} u \\ 4. \quad & \tilde{C}v = (D / \omega)(y + w) \end{aligned} \right\} \quad (22)$$

3.2 対角シフト付き E-SSOR 前処理_v2

複素前処理行列 M_{DS2} を以下のように定義する．

$$M_{\text{DS2}} = \{L + (D + \alpha i I) / \omega\} (D / \omega)^{-1} \{L^T + (D + \alpha i I) / \omega\}. \quad (23)$$

α はシフト量を表す実数パラメータである．このとき，(2) 式は以下のように変換される．

$$\begin{aligned} & (D / \omega) \{L + (D + \alpha i I) / \omega\}^{-1} C \{L^T + (D + \alpha i I) / \omega\}^{-1} \\ & \{L^T + (D + \alpha i I) / \omega\} z \\ &= (D / \omega) \{L + (D + \alpha i I) / \omega\}^{-1} b \end{aligned} \quad (24)$$

前処理後の複素係数行列 $\tilde{C}$ ，複素解ベクトル $\tilde{z}$ ，複素残差ベクトル $\tilde{r}$ は各々

$$\tilde{C} = (D / \omega) \{L + (D + \alpha i I) / \omega\}^{-1} C \{L^T + (D + \alpha i I) / \omega\}^{-1}, \quad (25)$$

$$\tilde{z} = \{L^T + (D + \alpha i I) / \omega\} z, \quad (26)$$

$$\tilde{r} = (D / \omega) \{L + (D + \alpha i I) / \omega\}^{-1} r \quad (27)$$

と表される．このとき，前処理後の複素係数行列 $\tilde{C}$ を

$$\begin{aligned}\tilde{C} &= (D/\omega)\{L + (D + \alpha iI)/\omega\}^{-1}C\{L^T + (D + \alpha iI)/\omega\}^{-1} \\ &= (D/\omega)(\{L^T + (D + \alpha iI)/\omega\}^{-1} + \{L + (D + \alpha iI)/\omega\}^{-1} \\ &\quad \times (I - 2D/\omega - 2\alpha i/\omega + D))\end{aligned}\quad (28)$$

と式の変形をすると，複素係数行列 $\tilde{C}$ と複素ベクトル v の積 $\tilde{C}v$ を次の 4 ステップで計算することで計算量を削減できる [1][3]．

$$\left. \begin{aligned} 1. \quad & \mathbf{y} = \{L^T + (D + \alpha iI)/\omega\}^{-1}\mathbf{v} \\ 2. \quad & \mathbf{u} = \mathbf{v} + (D - 2\alpha iI/\omega - 2D/\omega)\mathbf{y} \\ 3. \quad & \mathbf{w} = \{L + (D + \alpha iI)/\omega\}^{-1}\mathbf{u} \\ 4. \quad & \tilde{C}\mathbf{v} = (D/\omega)(\mathbf{y} + \mathbf{w}) \end{aligned} \right\} \quad (29)$$

3.3 全体シフト付き E-SSOR 前処理

複素前処理行列を以下のように定義する．

$$M_{CS} = (Li + Di/\omega)(D/\omega)^{-1}(L^T i + Di/\omega). \quad (30)$$

このとき，前処理後の複素係数行列 $\tilde{C}$ ，複素解ベクトル $\tilde{z}$ ，複素残差ベクトル $\tilde{r}$ を各々

$$\tilde{C} = (D/\omega)(Li + Di/\omega)^{-1}C(L^T i + Di/\omega)^{-1}, \quad (31)$$

$$\tilde{z} = (L^T i + Di/\omega)z, \quad (32)$$

$$\tilde{r} = (D/\omega)(Li + Di/\omega)^{-1}r \quad (33)$$

と定義すると，(2) 式は以下のように変換される．

$$\tilde{C}\tilde{z} = (D/\omega)(Li + Di/\omega)^{-1}b \quad (34)$$

このとき，前処理後の複素係数行列 $\tilde{C}$ を

$$\begin{aligned}\tilde{C} &= (D/\omega)(Li + Di/\omega)^{-1}C(L^T i + Di/\omega)^{-1} \\ &= (D/\omega)(-i(L^T i + Di/\omega)^{-1} + (Li + Di/\omega)^{-1} \\ &\quad \times (-iI + (1 - 2/\omega)D(L^T i + Di/\omega)^{-1}))\end{aligned}\quad (35)$$

と式変形すると，前処理後の複素係数行列 $\tilde{C}$ と複素ベクトル v の積 $\tilde{C}v$ を次の 4 ステップで計算することで計算量を削減できる．

$$\left. \begin{aligned} 1. \quad & \mathbf{y} = -i(L^T i + Di/\omega)^{-1}\mathbf{v} \\ 2. \quad & \mathbf{u} = -i\mathbf{v} + (1 - 2/\omega)Di\mathbf{y} \\ 3. \quad & \mathbf{w} = (Li + Di/\omega)^{-1}\mathbf{u} \\ 4. \quad & \tilde{C}\mathbf{v} = (D/\omega)(\mathbf{y} + \mathbf{w}) \end{aligned} \right\} \quad (36)$$

4. 数値実験

4.1 計算機環境と計算条件

計算機環境と計算条件を以下に示す．計算機は CX400 (CPU:CPU:Intex Xeon E5-2680, クロック周波数:2.7GHz, OS: Red Hat Linux Enterprise 6.1) を使用した．プログラムは Fortran90, コンパイラは Fujitsu Fortran Driver version 1.2.0. を使用した．最適化オプションは “-Kfast”

表 1 テスト行列の特徴

Table 1 Characteristics of test matrix.

行列	次元数	非零要素数	平均非零要素数
dtm_scatter_2pi	123,000	2,293,464	18.65
dtm_scatter_55	123,000	2,293,464	18.65

を用いた．計算はすべて倍精度浮動小数点演算で行い，時間計測には時間計測関数 “getrusage” を用いた．収束判定条件は相対残差の 2 ノルム: $\|r_{k+1}\|_2/\|r_0\|_2 \leq 10^{-9}$ ，初期近似解 x_0 はすべて 0，最大反復回数は 100,000 回とした．解法は COCG 法を用いた．前処理はスケーリングなし，対角化スケーリング，絶対対角化スケーリングの 3 種類と従来版 E-SSOR，3 種類のシフト付き E-SSOR の 4 種類の前処理を組み合わせた．また，IC(0) 分解付き COCG 法での実験を行い，E-SSOR 前処理との収束性を比較した．緩和係数 ω は -1.8, -1.5, -1.2, -1.0, -0.5, 0.5, 1.0, 1.2, 1.5, 1.8 の 10 通り変化させた．残差ベクトル $\tilde{r}$ の変換による余分な計算を除くために，反復 5 回ごとに収束判定は 1 度だけ行った．

4.2 テスト行列の特徴

表 1 にテスト行列の特徴を示す．これらの行列は物体回りの散乱波の様子を模擬する問題から得られ，Helmholtz 方程式に対する複素対称行列として記述される．行列名の最後尾の数字は周波数を表し，その値が 2π と 55 であることを意味する．

4.3 実験結果

表 2 に 5 種類の前処理付き COCG 法の収束性を示す．表中では，対角シフト付き E-SSOR 前処理_v1，同_v2 を対角行列 D をシフトした前処理として “DS_v1, 同_v2” と表記する．また，全体シフト付き E-SSOR 前処理を複素係数行列 C 全体をシフトした前処理として “CS” と表記する．表 2 の結果から以下のことが分かる．

- (1) 波数の大きさに関わらず，行列 dtm_scatter_2pi と行列 dtm_scatter_55 の両方に対して，全体シフト付き E-SSOR 前処理が最も速く収束した．
- (2) 特に，行列 dtm_scatter_2pi に対して，IC(0) 分解が前処理なしのときの 0.93 倍の反復時間で収束したのに対して，全体シフト付き E-SSOR 前処理は 0.69 倍の反復時間で速く収束し，他の前処理よりも優れた収束性を示した．
- (3) 行列のスケーリングは，絶対対角化スケーリングよりも通常の対角化スケーリングを適用したケースの方が少ない反復回数で収束した．

表 2 5 種類の前処理付き COCG 法の収束性

Table 2 The convergence rate of COCG method with five kinds of preconditionings.

(a) matrix : dtn_scatter_2pi							
前処理	対角化	ω シフト 量	反復 回数	反復 時間 [sec.]	比	平均 時間 [msec.]	比
なし	なし	- -	2,623	19.773	1.00	7.538	1.00
	対角	- -	2,439	18.485	0.93	7.579	1.01
	絶対対角	- -	2,475	19.784	1.00	7.994	1.06
IC(0)		- -	1,035	18.360	0.93	17.739	2.35
従来型	なし	1.0 -	1,050	14.325	0.72	13.643	1.81
	対角	1.2 -	1,045	13.988	0.71	13.386	1.78
	絶対対角	1.0 -	1,050	14.043	0.71	13.374	1.77
DS_v1		-0.5 -	2,995	40.716	2.06	13.595	1.80
DS_v2	なし	1.0 -	1,050	14.238	0.72	13.560	1.80
	対角	1.2 -	1,045	14.353	0.73	13.735	1.82
	絶対対角	1.0 -	1,050	14.297	0.72	13.616	1.81
CS	なし	1.2 0.0075	1,045	13.955	0.71	13.354	1.77
	対角	1.2 0.075	1,020	13.651	0.69	13.383	1.78
	絶対対角	1.2 0.01	1,025	13.788	0.70	13.452	1.78
(b) matrix : dtn_scatter_55							
前処理	対角化	ω シフト 量	反復 回数	反復 時間 [sec.]	比	平均 時間 [msec.]	比
なし	なし	- -	4,735	35.682	1.00	7.536	1.00
	対角	- -	4,765	35.983	1.01	7.552	1.00
	絶対対角	- -	4,698	37.703	1.06	8.025	1.06
IC(0)		- -	2,077	36.854	1.03	17.744	2.35
従来型	なし	1.0 -	2,145	28.592	0.80	13.330	1.77
	対角	1.2 -	2,035	27.092	0.76	13.313	1.77
	絶対対角	1.2 -	2,085	27.921	0.78	13.391	1.78
DS_v1		-0.5 -	4,855	67.478	1.89	13.899	1.84
DS_v2	なし	1.2 -	2,130	28.770	0.81	13.507	1.79
	対角	1.2 -	2,035	27.688	0.78	13.606	1.81
	絶対対角	1.0 -	2,115	28.738	0.81	13.588	1.80
CS	なし	1.2 0.01	2,095	27.848	0.78	13.293	1.76
	対角	1.2 -0.25	1,980	26.443	0.74	13.355	1.77
	絶対対角	1.2 -0.25	1,980	26.424	0.74	13.345	1.77

5. まとめ

波動現象の波数の大きさに関わらず，全体シフト付き E-SSOR 前処理が最速で収束し，収束性が優れていることがわかった．

参考文献

- [1] Chan, T. F., van der Vorst, H. A.: Approximate and Incomplete Factorizations, in D.E. Keyes, A. Samed and V. Venkatakrishnan (eds.), Parallel Numerical Algorithms, 1997.
- [2] Dongarra, J., Duff, I., Sorensen, D., van der Vorst, H. A.: Numerical Linear Algebra for High-Performance Computers, SIAM, Philadelphia, 1998.
- [3] Eisenstat, S. C.: Efficient implementation of a class of preconditioned conjugate gradient methods, SIAM J. Sci. Stat. Comput., Vol.2, No.1, pp.1-4, 1981.
- [4] Fujino, S., Fujiwara, M., Yoshida, M.: A proposal of preconditioned BiCGSafe method with safe convergence, Proc. of The 17th IMACS World Congress on Scientific Computation, Appl. Math. Simul., CD-ROM, Paris, France, 2005.
- [5] 伊東千晶, 岩里洸介, 村上啓一, 藤野清次: Cache-Cache(カシュカシ) Elements 並列化手法の提案, 日本計算工学会論文集, Vol.2014, No.20140013, 2014.
- [6] Magolu, M. M. M.: Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix, Commun. Numer. Meth. Engng, Vol. 16, pp.801-817, 2000.
- [7] Magolu, M. M. M., Beauwens, R., Warzee, G.: Incomplete factorization-based preconditionings for solving the Helmholtz equation, Int. J. Numer. Meth. Engng., Vol. 50, pp.1077-1101, 2001.
- [8] 柿原正伸, 藤野清次: 構造解析で現れる線形方程式に対する対角緩和つき準口バースト ICCG 法の収束性評価, 日本計算工学会講演集, 2004.
- [9] 柿原正伸, 藤野清次: 緩和係数 ω を自動決定する対角緩和準口バースト ICCG 法の収束性, 情報処理学会論文誌, Vol.46, NO.SIG 4(ACS 9), pp.45-55, 2005.
- [10] 柿原正伸: 対称行列に対するクリロフ部分空間法の不完全分解前処理, 九州大学大学院 システム情報科学府情報学専攻 修士論文, 2005.
- [11] Y. Saad: Iterative Methods for Sparse Linear Systems, 2nd Ed, SIAM, Philadelphia, 2003.
- [12] University of Florida Sparse Matrix Collection: <http://www.cise.ufl.edu/research/sparse/matrices/index.html>