

大規模行列系に対する頑強な2階ILU分解前処理

山口 遼[†] 野寺 隆^{††}

行列の前処理とは、反復回数を減少させるなど反復法における収束を改善する方法として有効な手法である。特に、ILU分解前処理は係数行列が大規模で疎な非対称行列な場合によく用いられる方法である。本稿では、このILU分解前処理を発展させた2階ILU分解(ILU2分解)前処理とその頑強性について考察する。数値実験において反復回数や収束時間を比較し、この行列の前処理の有効性を考察する。

Preconditioning of robust 2nd order ILU decomposition for large sparse linear systems

RYOICHI YAMAGUCHI[†] and TAKASHI NODERA^{††}

A preconditioning is an effective technique for the convergence of an iterative method, which is reduced iterations. In particular, the preconditioner, ILU decomposition, is using for a large, sparse, and nonsymmetric coefficient matrix. In this paper, this preconditioner expanded into 2nd order ILU decomposition. For numerical experiments, we show the effectiveness of its preconditioner with respect to iterations and computation time.

1. 序 論

物理現象を定式化すると、偏微分方程式の境界値問題に帰着する場合が多い。このような問題は、有限差分法や有限要素法などを用いて離散近似を行うことにより、次の連立1次方程式が得られる。

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad x, b \in \mathbb{R}^n \quad (1)$$

ただし、 A は正則な係数行列、 b は右辺ベクトル、 x は解ベクトルである。この式(1)における係数行列 A が離散化した際に大規模で疎となる場合を考える。そのため、比較的単純な演算を繰り返しながら、ある初期値から真の解へと逐次的に近似解を求める反復法が解法として一般に有効である。

反復法の中で代表的な共役勾配法(Conjugate Gradient method, CG法ともいう)¹⁾は係数行列 A が対称正定値行列な場合において有効な算法である。また、係数行列 A が非対称な場合においては一般最小残差法(Generalized Minimum Residual method, GMRES法)²⁾が算法として有効である。ただし、GMRES法は反復回数とともに計算量が増大してしまうため、係数行列が大型な場合はリスタートを適用したGMRES

(m)法がよく用いられている。

反復法の収束を改善する方法として行列の前処理を利用する場合がある。これは前処理行列^{5),6)}を用いて、式(1)を変換することで、元の方程式の性質を改善することにより、収束までに必要な反復回数を減少させ、計算時間を短縮させる方法である。その前処理行列として代表的なものが、不完全行列分解の1つで係数行列が対称である場合に用いられる不完全コレスキー分解(Incomplete Cholesky decomposition, IC分解)である。また、係数行列が非対称である場合にはILU分解(Incomplete LU decomposition)^{6),7)}が用いられる。さらに、行列の前処理には不完全行列分解のほかに、行列分離や近似逆行列を用いた前処理の技法などがある。

本稿では、2章において行列の前処理としてよく用いられるILU分解とドロッピング(dropping)の仕方に使われるレベルについて述べる。次に、3章でそのILU分解を発展させた2階ILU分解(ILU2分解)について詳しく述べる。そして、4章では頑強性について述べる。さらに、5章では行列の前処理を施したGMRES(m)法を用いての数値実験とその結果について述べ、6章で結論と今後の課題について述べる。

2. 行列の前処理

式(1)の連立1次方程式における行列の前処理として左前処理と右前処理がある。左前処理とは正則行列

[†] 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
^{††} 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

```

01: for i := 1 to n do
02:   for j := 1 to i - 1 do
03:      $l_{i,j} = \frac{1}{u_{j,j}} (a_{i,j} - \sum_{k=1}^{j-1} (l_{i,k} u_{k,j}))$ 
04:   endfor
05:    $l_{i,i} = 1$ 
06:   for j := i to n do
07:      $u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} (l_{i,k} u_{k,j})$ 
08:   endfor
09: endfor

```

図1 LU分解の算法

M を用いて次のように変換することである。

$$M^{-1}Ax = M^{-1}b \quad (2)$$

また、右前処理とは同じく正則な行列 M を用いて次のように変換することである。

$$\begin{cases} AM^{-1}y = b \\ M^{-1}y = x \end{cases} \quad (3)$$

このような行列 M を前処理行列という。この式 (2) と式 (3) の行列 M を係数行列 A に近似できるならば、式 (1) の性質を改善する可能性がある。また、この前処理行列 M を決定するときに重要なこととして、行列 M の逆行列とベクトルとの積が容易に計算できることなどが挙げられる。

2.1 LU 分解

最初に、LU 分解について述べる。LU 分解とは行列 A を

$$A = \bar{L}\bar{U} \quad (4)$$

と下三角行列 \bar{L} と上三角行列 \bar{U} の積に分解することである。LU 分解において行列 \bar{L} または行列 \bar{U} の対角要素の値をすべて 1 と設定することで一意に決めることができる。これは Gauss の消去法を用いてもできるが、直接分解も可能である。行列 \bar{L} の対角成分をすべて 1 としたとき、すなわち行列 \bar{L} が下単位三角行列の LU 分解の直接分解を図 1 に示す。

2.2 ILU 分解

ILU 分解とは以下のように不完全に LU 分解したものである。

$$A = LU + R \quad (5)$$

ただし、行列 R は行列 A と行列積 LU の差を表す誤差行列である。

LU 分解の式 (4) において行列 A が疎の場合でも行列 \bar{L} と行列 \bar{U} は一般に密行列になる。すなわち、行列 A のゼロ成分の添え字と同じ添え字の行列 \bar{L} 、 \bar{U} の成分が非ゼロとなることがある。これをフィルイン (fill-in) という。つまり、ILU 分解とはこのフィル

ンした成分をある程度無視して、LU 分解を行うことである。

この不完全分解の仕方として行列 A の非ゼロ要素の添え字集合を次のように置く。

$$G_A = \{(i, j); a_{i,j} \neq 0\} \quad (6)$$

そして、次のようになる添え字集合 G を決める。

$$G \supseteq G_A \quad (7)$$

そこで、行列 A を ILU 分解する際に、一般に行列 L と行列 U の要素の中で集合 G に属する添え字を持つもののみを計算し、他のものをゼロとすると、式 (5) における行列 L 、 U が求められ、前処理行列 M が次のように決定できる。

$$M = LU \quad (8)$$

この式 (8) の M の逆行列とベクトルとの積は三角行列の逆行列とベクトルとの積を計算すればよいので容易である。そのため、ILU 分解は前処理行列としてよく用いられている。

2.3 レベル

行列を不完全分解するとき、フィルインを考慮しなからの添え字集合 G の決定の仕方というドロップピングに関する問題が存在する。この問題に対して、一般にはレベルという概念が広く用いられている。

レベルとは初期設定として行列 A の (i, j) 成分が非ゼロのとき、すなわち添え字集合 G_A に属するものをレベル 0 とし、行列 A の (i, j) 成分がゼロのときはレベル ∞ とする。次に、レベル ∞ のところのレベルを再設定するため、 (i, j) 成分のレベルを次のように一般に定義する。

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

この定義に従ってレベル ∞ のところのレベルを 1 から順に決定することができる。

ILU 分解の ILU(0) や ILU(1) は、このレベルの概念に基づいて構成されていて、ILU(0) はレベル 0 の添え字の集合を集合 G と決定することでドロップピングし、ILU(1) はレベル 0 とレベル 1 の添え字の集合を集合 G と決定することでドロップピングしている。

3. 2 階 ILU 分解 (ILU2 分解)

3.1 ILU2 分解

近年、I. E. Kaporin^{8),9)} は係数行列が対称なときに用いる IC 分解を拡張した 2 階 IC 分解 (IC2 分解) を提案した。この IC2 分解を非対称な場合に拡張したのが ILU2 分解である。

ILU2 分解は係数行列 A を次のように分解することである。

$$A = LU + L_0U + LU_0 - S \quad (9)$$

ただし、行列 L と行列 U はそれぞれ対角要素の値すべてが 1 の下三角行列と上三角行列であり、行列 L_0 , U_0 は狭義下三角行列、狭義上三角行列である。また、行列 S は 2 階における誤差行列である。さらに、式 (9) を変形すると次のようになる。

$$A + L_0 U_0 + S = (L + L_0)(U + U_0) \quad (10)$$

このような分解を行うと、通常の ILU 分解を利用する前処理行列よりも高い近似精度の前処理行列で処理されるので、頑健性の点においても優れていると考えられる。

次に、式 (9) を成分表示する。ただし、 S は簡素化のため 0 とする。また、行列 L は下単位三角行列より対角成分 $l_{i,i} = 1$ であり、行列 L_0 , U_0 は狭義三角行列より $l_{i,i}^{(0)} = u_{i,i}^{(0)} = 0$ である。 $i > j$ のとき、次のようになる。

$$\begin{aligned} a_{i,j} &= \sum_{k=1}^j (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \\ &= (l_{i,j} + l_{i,j}^{(0)}) u_{j,j} \\ &\quad + \sum_{k=1}^{j-1} (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \end{aligned}$$

このときの $L + L_0$ の成分は次のように表せる。

$$\begin{aligned} l_{i,j} + l_{i,j}^{(0)} \\ = \frac{1}{u_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \right) \end{aligned}$$

さらに、 $i \leq j$ のときは次のようになる。

$$\begin{aligned} a_{i,j} &= \sum_{k=1}^i (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \\ &= (u_{i,j} + u_{i,j}^{(0)}) \\ &\quad + \sum_{k=1}^{i-1} (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \end{aligned}$$

このときの $U + U_0$ の成分も次のように同様に記述できる。

$$\begin{aligned} u_{i,j} + u_{i,j}^{(0)} \\ = a_{i,j} - \sum_{k=1}^{i-1} (l_{i,k} u_{k,j} + l_{i,k}^{(0)} u_{k,j} + l_{i,k} u_{k,j}^{(0)}) \end{aligned}$$

以上から行列 L と行列 L_0 、行列 U と行列 U_0 は一意に定まらない。そこで、行列 L と行列 L_0 、行列 U と行列 U_0 に構造的直交性を持たせることで、それらの値を決定する。構造的直交性を持たせるためには、行列 L , L_0 , U , U_0 の添え字集合 G_L , G_{L_0} , G_U , G_{U_0}

に関して次のようにする。

$$G_L \neq G_{L_0}, \quad G_U \neq G_{U_0} \quad (11)$$

すなわち、任意の i, j に対して行列 L と行列 L_0 、行列 U と行列 U_0 の成分が

$$l_{i,j} \times l_{i,j}^{(0)} = 0, \quad u_{i,j} \times u_{i,j}^{(0)} = 0 \quad (12)$$

と成り立つように決定することである。これにより、行列 U と行列 U_0 、行列 L と行列 L_0 がそれぞれ一意に定まる。以上のように得られた ILU2 分解を前処理行列として利用する。

4. 頑強性

IC 分解において M. A. Ajiz ら¹⁰⁾ が、IC2 分解においては I. E. Kaporin⁸⁾ がそれぞれの分解に対して頑強性を持たせることを行っている。IC 分解においては、ある閾値を設定して分解における基準値がその閾値以下となる場合に、その要素は棄却するが、同時に対角項に対して修正を行う方法である。また、IC2 分解に関しては分解した成分に 0 以上 1 以下の値を掛けることや対角項の修正などを行っている。

今回、前章で述べた ILU2 分解前処理に頑強性を持たせることを試みる。閾値を設定して ILU2 分解により計算された成分が閾値以下となる場合にこの成分を棄却する。また、分解により計算された行列 U の非対角成分を 0 以上 1 以下の値を掛けるなどの非対角項に対しての補正を行うこととする。

5. 数値実験

5.1 数値実験の環境と条件

数値実験は以下の動作環境のもと実装させた。

- 計算機 : Dell PowerEdge 1750
- OS : Red Hat Linux 9
- CPU : 3.00 GHz × 1 Intel(R) XEON(R)
- メモリ : 4.0 MB

また、数値実験で用いた条件は以下である。

- 収束判定条件 : $\|\mathbf{r}_m\| / \|\mathbf{r}_0\| < 1.0 \times 10^{-12}$
- 初期値ベクトル : $\mathbf{x}_0 = (0, 0, \dots, 0)^T$
- 最大反復回数 : 10000
- プログラム言語 : C 言語
- 計算精度 : 倍精度

ただし、ベクトル \mathbf{r}_m は第 m 回目の残差ベクトルを表す。計算時間に関しては clock() 関数で求めた値を秒単位で表した。

また、使用した算法は GMRES(m) 法である。GMRES(m) 法は 1986 年に Y. Saad ら²⁾ によって提案された方法である GMRES 法のリスタート型であり、係数行列が非対称で正則な場合に有効な解法である。また、Krylov 部分空間法という反復法の解法の 1 つでもある。GMRES(m) 法の算法は図 2 に示す。

```

01: choose:  $x_0$ 
02:  $r_0 := b - Ax_0$ ;
03:  $\beta := \|r_0\|_2$ ;  $v_1 := r_0/\beta$ ;
04: start:
05: for  $i := 1$  to  $m$  do
06:    $\hat{v} := Av_i$ ;
07:   for  $j := 1$  to  $i$  do
08:      $h_{j,i} := (\hat{v}, v_j)$ ;
09:      $\hat{v} := \hat{v} - h_{j,i}v_j$ ;
10:   enddo
11:    $h_{i+1,i} := \|\hat{v}\|_2$ ;
12:    $v_{i+1} := \hat{v}/h_{i+1,i}$ ;
13:   compute  $y_m = \min_y \|\beta e_1 - H_m^* y\|_2$ ;
14: enddo
15:  $x_m := x_0 + V_m y_m$ ;
16: if  $\|b - Ax_m\|_2 \leq \epsilon$  then
17:   stop iteration
18: endif
19:  $x_0 := x_m$ ;  $r_0 := b - Ax_0$ ;
20:  $\beta := \|r_0\|_2$ ;  $v_1 := r_0/\beta$ ;
21: goto start

```

図2 GMRES(m)法の算法

5.2 数値例1の問題

正方領域 $\Omega = [0, 1] \times [0, 1]$ における2次元楕円型偏微分方程式の境界値問題を考える³⁾.

$$-u_{xx} - u_{yy} + D(u_x + u_y) = f(x, y)$$

$$u(x, y)|_{\partial\Omega} = 1.0$$

この方程式のメッシュ幅を $1/129$ として、5点中心差分近似を用いて離散化し、真の解が $u(x, y) = 1.0$ となるように右边を設定し、数値実験を行った。そこで得られた連立1次方程式の次元は16384となり、係数行列 A は次のようなブロック3重対角行列となる。

$$A = \begin{pmatrix} T & D_1 & & 0 \\ D_2 & \ddots & \ddots & \\ & \ddots & \ddots & D_1 \\ 0 & & D_2 & T \end{pmatrix} \quad (13)$$

ただし、ブロック行列 T は3重対角行列であり、ブロック行列 D_1, D_2 は対角行列である。また、行列 T, D_1, D_2 はこの場合において次元は128となる。以上から、このブロック3重対角行列 A は図3のような5重対角行列ともなる。

この数値実験では、比較のため前処理なしのGMRES(m)法と前処理付きのGMRES(m)法を用いる。ただし、前処理行列として用いるのはILU分解とILU2分解、そして簡単な頑強性を保持させたRILU2分解である。ILU分解に関してはILU(0), ILU(1)の

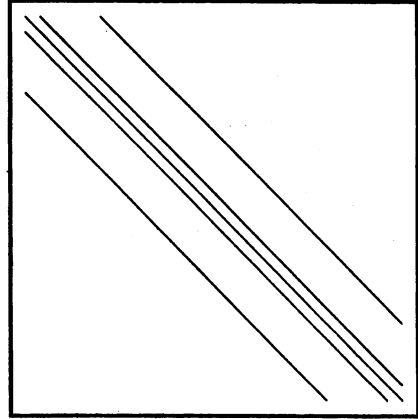


図3 行列Aの構造

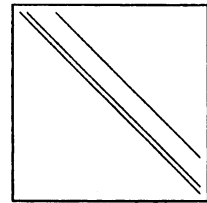
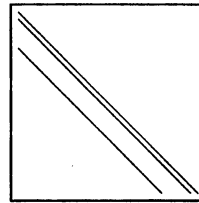


図4 ILU(0):行列Lの構造 図5 ILU(0):行列Uの構造

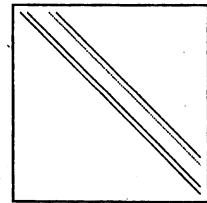
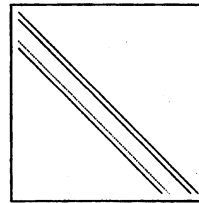


図6 ILU(1):行列Lの構造 図7 ILU(1):行列Uの構造

2つを使う。ILU(0)は行列 A が図3の場合に行列 L と行列 U の構造が図4, 図5の形を取る。ILU(1)は同様の行列 A の場合、行列 L と行列 U の構造が図4, 図5の形に点線部分を加えた図6, 図7の形を取る。また、ILU2分解の場合は行列 L, U の取り方はILU(0)と同じにし、行列 L_0, U_0 は2種類の取り方(便宜的にILU2-1とILU2-2と名付ける)を行う。行列 L_0, U_0 は図6, 図7の点線部分、すなわちレベル1の部分を取るのであるが、ILU2-1は点線部分の要素を1つ飛ばしに取っていき、ILU2-2は点線部分の要素すべてを取るようにする。RILU2分解はILU2-2に関して閾値の設定や非対角項の補正を行ったものである。

また、 $Dh = 2^0, 2^{-1}, 2^{-2}, 2^{-3}$ の4通りで変化さ

せ、リスタート周期 $m = 10, 20, 30$ の 3 通りで行った。さらに、反復回数に関しては、GMRES(m) 法のなかの Arnoldi 過程の 1 反復につき 1 回と数えた。

5.3 数値例 1 の結果

表 1 において反復回数と計算時間をそれぞれ Dh とリスタート周期 m を変化させた場合について示した。ILU 分解は ILU(0), ILU(1) を, ILU2 分解は ILU2-1, ILU2-2 を, そして RILU2 分解をそれぞれ行列の前処理として用いた GMRES(m) 法と前処理を施さない GMRES(m) 法との比較で表した。結果, ILU(1) を用いた前処理より ILU2-1 を用いた前処理のほうが反復回数や計算時間がすべてにおいて余計かった。しかし, ILU2-2 が ILU(1) より反復回数や計算時間が少なくなるところがあった。また, RILU2 に関してはほとんど ILU2-2 より反復回数が少なくなった。しかし, ILU(1) より反復回数が少なくなるものがほとんどなかった。

5.4 数値例 2 の問題

数値例 1 と同様に正方領域 $\Omega = [0, 1] \times [0, 1]$ における 2 次元楕円型偏微分方程式の境界値問題を考える³⁾。

$$-u_{xx} - u_{yy} + D[(y-2)u_x + (x-1/3)(x-2/3)u_y] = f(x, y)$$

$$u(x, y)|_{\partial\Omega} = 1.0$$

この方程式のメッシュ幅 h を $1/129$ として, 5 点中心差分近似を用いて離散化し, 真の解が $u(x, y) = 1.0$ となるように右辺を設定し, 数値実験を行った。そこで得られた連立 1 次方程式の次元は 16384 となり, 係数行列 A は数値例 1 と同様に式 (13) のブロック三重対角行列となり, そのブロック行列は次元は 128 となる。前処理に関しては行列 A を 5 重対角行列とみなし, 数値例 1 と同様に ILU(0), ILU(1), ILU2-1, ILU2-2, RILU2 を用いて比較する。

また, 同様に $Dh = 2^0, 2^{-1}, 2^{-2}, 2^{-3}$ の 4 通りで変化させ, リスタート周期 $m = 10, 20, 30$ の 3 通りで行った。さらに, 反復回数に関しても同様に, GMRES(m) 法のなかの Arnoldi 過程の 1 反復につき 1 回と数えた。

5.5 数値例 2 の結果

表 2 においては数値例 1 と同様に ILU 分解は ILU(0), ILU(1) を用い, ILU2 分解は ILU2-1, ILU2-2, そして RILU2 を用いた前処理を施した GMRES(m) 法と前処理を施さない GMRES(m) 法との比較を反復回数と計算時間で表した。結果, ILU(1) を用いた前処理より ILU2-1 を用いた前処理のほうが反復回数や計算時間がすべてにおいて余計にかかった。しかし, $Dh = 2^{-2}$ で $m = 10$ のとき, ILU2-2 が ILU2-1 より反復回数と計算時間がともに余計なかった。また, $Dh = 2^{-2}$ で $m = 20$ のとき, ILU2-2 が ILU(1) より反復回数や計算時間が少なくなった。RILU2 に関しては, ILU2-2 より反復回数が同じかそ

れ以下となった。しかし, 計算時間が ILU2-2 より上回る場合があった。また, 同じく $Dh = 2^{-2}$ で $m = 10$ のとき, RILU2 が反復回数と計算時間がともに最小だった。

6. 結論と今後の課題

本稿では, 非対称な係数行列を持つ連立 1 次方程式に対する 2 階 ILU 分解による行列の前処理の有効性を示すため数値実験を行った。今回の数値実験において, ドロップングの際, ILU(1) と ILU2-2, RILU2 はほとんど同じ成分を使用したため, 結果から見るに差異が明確にあらわれなかった。

今後の課題として, 今回用いた数値例は 2 次元問題のみであるので 3 次元問題や物理現象に現れる一般疎行列の問題を用いるなどして, 数値例を充実させていくことが挙げられる。また, 悪条件問題に対しての ILU2 分解前処理に最適なドロップングや収束の安定化を検討していきたいと思う。さらに, 今回用いた RILU2 分解前処理は簡単なものだったので, より頑強にするように様々な工夫をしていきたいと思う。

参考文献

- 1) Hestenes, M. R. and Stiefel, E.: Method of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stand.*, Vol. 49, pp. 409-436, (1952).
- 2) Saad, Y. and Schultz, M. H.: GMRES: A generalized minimal residual algorithms for solving nonsymmetric linear equations, *SIAM J. Sci. Stat. Comput.*, Vol. 7, pp. 856-869, (1986).
- 3) Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix Anal. Appl.*, Vol. 13, pp. 926-943, (1992).
- 4) Simoncini, V.: On the convergence of restarted Krylov subspace methods, *SIAM J. Matrix Anal. Appl.*, Vol. 22, pp. 430-452, (2000).
- 5) Barrett, R. et al.: Templates for the solution of linear systems: building blocks for iterative methods, *SIAM*, (1994).
- 6) Bruaset, A. M.: A survey of preconditioned iterative methods, *Longman Science & Technical*, (1995).
- 7) Saad, Y.: Iterative methods for sparse linear systems, *PWS Publishing Company*, Boston, (1996).
- 8) Kaporin, I. E.: High quality preconditioning of a general symmetric positive definite matrix based on its $U^T U + U^T R + R^T U$ -decomposition, *Numer. Linear Algebra Appl.*, Vol. 5, pp. 483-509, (1998).
- 9) Kaporin, I. E.: Using the modified 2nd order

表 1 数値例 1 の結果 : 反復回数と計算時間 ($h = 1/129$)

| Dh Method | 2^0 | | 2^{-1} | | 2^{-2} | | 2^{-3} | |
|------------------|------------|-----------|------------|-----------|------------|-----------|------------|-----------|
| | Iterations | Time(Sec) | Iterations | Time(Sec) | Iterations | Time(Sec) | Iterations | Time(Sec) |
| GMRES(10) | 488 | 0.50 | 441 | 0.45 | 556 | 0.57 | 1222 | 1.26 |
| ILU(0)-GMRES(10) | 147 | 0.36 | 221 | 0.47 | 230 | 0.49 | 206 | 0.44 |
| ILU(1)-GMRES(10) | 61 | 0.13 | 122 | 0.25 | 151 | 0.31 | 154 | 0.33 |
| ILU2-1-GMRES(10) | 89 | 0.19 | 168 | 0.35 | 168 | 0.35 | 182 | 0.39 |
| ILU2-2-GMRES(10) | 69 | 0.14 | 150 | 0.31 | 134 | 0.28 | 164 | 0.34 |
| RILU2-GMRES(10) | 63 | 0.15 | 143 | 0.33 | 140 | 0.32 | 164 | 0.37 |
| GMRES(20) | 659 | 1.04 | 621 | 0.97 | 516 | 0.80 | 632 | 0.98 |
| ILU(0)-GMRES(20) | 179 | 0.53 | 271 | 0.70 | 287 | 0.74 | 336 | 0.88 |
| ILU(1)-GMRES(20) | 35 | 0.09 | 125 | 0.32 | 206 | 0.53 | 187 | 0.47 |
| ILU2-1-GMRES(20) | 61 | 0.16 | 225 | 0.57 | 232 | 0.61 | 198 | 0.50 |
| ILU2-2-GMRES(20) | 36 | 0.09 | 124 | 0.31 | 180 | 0.46 | 181 | 0.47 |
| RILU2-GMRES(20) | 35 | 0.10 | 131 | 0.36 | 172 | 0.44 | 197 | 0.54 |
| GMRES(30) | 780 | 1.61 | 808 | 1.65 | 682 | 1.39 | 624 | 1.30 |
| ILU(0)-GMRES(30) | 181 | 0.64 | 299 | 0.93 | 360 | 1.11 | 366 | 1.13 |
| ILU(1)-GMRES(30) | 29 | 0.09 | 83 | 0.25 | 121 | 0.38 | 146 | 0.45 |
| ILU2-1-GMRES(30) | 46 | 0.14 | 120 | 0.39 | 241 | 0.75 | 232 | 0.71 |
| ILU2-2-GMRES(30) | 30 | 0.09 | 87 | 0.27 | 145 | 0.44 | 168 | 0.51 |
| RILU2-GMRES(30) | 29 | 0.11 | 86 | 0.28 | 128 | 0.41 | 154 | 0.51 |

表 2 数値例 2 の結果 : 反復回数と計算時間 ($h = 1/129$)

| Dh Method | 2^0 | | 2^{-1} | | 2^{-2} | | 2^{-3} | |
|------------------|------------|-----------|------------|-----------|------------|-----------|------------|-----------|
| | Iterations | Time(Sec) | Iterations | Time(Sec) | Iterations | Time(Sec) | Iterations | Time(Sec) |
| GMRES(10) | 560 | 0.59 | 504 | 0.52 | 550 | 0.56 | 969 | 0.98 |
| ILU(0)-GMRES(10) | 177 | 0.43 | 200 | 0.42 | 214 | 0.45 | 210 | 0.44 |
| ILU(1)-GMRES(10) | 142 | 0.30 | 141 | 0.30 | 156 | 0.32 | 148 | 0.31 |
| ILU2-1-GMRES(10) | 145 | 0.30 | 168 | 0.35 | 166 | 0.35 | 163 | 0.34 |
| ILU2-2-GMRES(10) | 132 | 0.28 | 147 | 0.30 | 175 | 0.36 | 145 | 0.30 |
| RILU2-GMRES(10) | 142 | 0.32 | 148 | 0.33 | 174 | 0.40 | 155 | 0.35 |
| GMRES(20) | 666 | 1.03 | 621 | 0.97 | 591 | 0.91 | 715 | 1.11 |
| ILU(0)-GMRES(20) | 279 | 0.83 | 280 | 0.74 | 311 | 0.80 | 303 | 0.80 |
| ILU(1)-GMRES(20) | 137 | 0.35 | 199 | 0.51 | 246 | 0.62 | 180 | 0.46 |
| ILU2-1-GMRES(20) | 184 | 0.47 | 251 | 0.64 | 248 | 0.66 | 259 | 0.66 |
| ILU2-2-GMRES(20) | 152 | 0.39 | 256 | 0.65 | 212 | 0.55 | 177 | 0.45 |
| RILU2-GMRES(20) | 142 | 0.39 | 219 | 0.60 | 199 | 0.51 | 177 | 0.51 |
| GMRES(30) | 754 | 1.56 | 741 | 1.52 | 650 | 1.44 | 688 | 1.44 |
| ILU(0)-GMRES(30) | 373 | 1.27 | 436 | 1.35 | 382 | 1.18 | 356 | 1.11 |
| ILU(1)-GMRES(30) | 85 | 0.26 | 147 | 0.45 | 175 | 0.54 | 206 | 0.63 |
| ILU2-1-GMRES(30) | 166 | 0.50 | 240 | 0.73 | 297 | 0.91 | 235 | 0.72 |
| ILU2-2-GMRES(30) | 88 | 0.28 | 166 | 0.50 | 203 | 0.62 | 229 | 0.70 |
| RILU2-GMRES(30) | 87 | 0.29 | 148 | 0.48 | 181 | 0.59 | 213 | 0.68 |

incomplete Cholesky decomposition as the conjugate gradient preconditioning, *Numer. Linear Algebra Appl.*, Vol. 9, pp. 401-408, (2002).

- 10) Ajiz, M. A. and Jennings A.: A robust incomplete Choleski-conjugate gradient algorithm, *International Journal for Numerical Methods in Engineering*, Vol. 20, pp. 949-966, (1984).