

シュールコンプリメントに対する前処理の一考察

森屋 健太郎[†] 野寺 隆^{††}

大規模で疎な連立 1 次方程式は、独立集合を元に係数行列の非零要素を並べかえて、より次元数の小さい連立 1 次方程式に変換することができる。このような次元数の小さい連立 1 次方程式は、シュールコンプリメントと呼ばれている。しかし、シュールコンプリメントに変換するボトルネックを埋め合わせるためには、通常は前処理を適用することが不可欠である。本稿では、シュールコンプリメントに対する近似逆行列を計算するのに Newton 法を用いる。さらに、初期値には対角行列を選択し、Newton 法の漸化式の行列どうしの乗算を回避する計算法を提案する。数値実験の結果から、本稿で提案した計算法で近似逆行列を求めると、Newton 法の反復が 1 回ないしは 2 回でも近似逆行列が前処理としての性能を十分発揮することを示す。

A Note on Preconditioner for the Schur-Compliments

KENTARO MORIYA [†] and TAKASHI NODERA^{††}

The large and sparse linear systems of equations can be transformed into the other smaller linear systems, by re-ordering nonzero elements, which based on the independent set. This smaller linear systems are called "Schur-Compliments". However, in order to make up for the overhead to transform into Schur-Compliments, the preconditioner is required. In this paper, we use Newton method for computing the approximate inverse of Schur-Compliments. Moreover we also propose the scheme which chooses the diagonal matrix as the initial values and avoids the multiplication of two matrices in the recurrence of Newton method. The numerical results show that the proposed scheme works fairly well as the preconditioner even if the number of iteration steps is one or two.

1. はじめに

大型で疎な連立 1 次方程式

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

を考える。独立集合を元に、係数行列 A の行と列を並び換えることによって、方程式 (1) を次元数のより小さい係数行列を持つ別の連立 1 次方程式へ変換することができる。この変換後の次元数の小さい連立 1 次方程式は、通常シュールコンプリメントと呼ばれている。しかし、シュールコンプリメントに変換するためには、独立集合を元に係数行列 A の行と列を並び換えるオーバーヘッドがかかることになる。従って、方程式の変換に要する計算時間のオーバーヘッドを埋め合わせるために、シュールコンプリメントには、通常は前処理が適用される。本稿では、シュールコンプリメントに対する前処理として近似逆行列を用い、その計算を Newton 法で行うことを考える。Newton 法により近似逆行列を計算する場合、行列どうしの乗算が必要

なので、Newton 法の反復の途中で近似逆行列にフィロイン (元々 0 であった要素が 0 でなくなること) が生じることになる。特に、近似逆行列の初期値に密行列を選ぶとその傾向は顕著である。そこで、本稿では近似逆行列を求めるときの計算量とメモリを削減するために、対角行列を Newton 法の初期値として選択する。さらに、Newton 法の漸化式で必要となる行列どうしの乗算を回避するような計算法を提案する。このような初期値の選択と行列どうしの乗算を回避する計算により、Newton 法の反復を 1 回から 2 回で止めたとしても、近似逆行列が前処理としての性能を十分に発揮することを示す。

2 章では、係数行列 A の独立集合を求めて、方程式 (1) をシュールコンプリメントへ変換する方法を簡単に述べる。3 章では、Newton 法による近似逆行列の計算法を述べ、Newton 法の初期値の選択法と行列どうしの乗算を行わない計算法を提案する。4 章で数値実験の結果を報告し、そこから得られた結論を 5 章で述べる。

2. シュールコンプリメントへの変換

本章では、独立集合を基に係数行列 A の行と列の

[†] 青山学院大学理工学部
Faculty of Science and Technology, Aoyama Gakuin University
^{††} 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

並び換えを行い、シュールコンプリメントを生成する方法について述べる。

2.1 独立集合に基づく置換

本節では、Geedyなアルゴリズムに基づいて、独立集合を求める方法を述べる。方程式(1)の解 x について以下の“条件1”が成立するとき、 i 番目の要素 x_i と j 番目の要素 x_j は互いに独立であるという。

[条件1] $\forall a_{ij} \in A$ について、 $a_{ij} = 0$ かつ $a_{ji} = 0$ である。

つまり、係数行列 A の対角成分について互に対称な要素 a_{ij} と a_{ji} がいずれも0なら、 x の i 番目と j 番目の要素は互いに独立となる。 x の互いに独立である要素の集合のことを独立集合という。

シュールコンプリメントを生成するためには、この独立集合に属する解 x の要素をすべて x の上部へ集めればよい。それにともなって、行列 A と右辺ベクトル b の要素の並び換えが必要となる。今、 x の独立集合が

$$G = \{x_{s_1}, x_{s_2}, \dots, x_{s_k}\}$$

で、集合 G に属する要素数が k 個だった場合を考える。この場合に、係数行列 A の s_1, s_2, \dots, s_k 行を上部に、 s_1, s_2, \dots, s_k 列を左部に寄せ、右辺ベクトル b の s_1, s_2, \dots, s_k 番目の要素を上部に寄せればよい。このような並び換えを行うことで、方程式(1)は

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2)$$

となる。 x_1 の要素が独立集合 G に属し、 x_2 の要素は G に属さないことになる。従って、

$$x_1 = [x_{s_1}, x_{s_2}, \dots, x_{s_k}]^T, (x_{s_i} \in G)$$

かつ

$$x_2 = [x_{s_{k+1}}, x_{s_{k+2}}, \dots, x_{s_n}]^T, (x_{s_i} \notin G)$$

である。また、ブロック行列 A_1 は

$$A_1 = \text{diag}\{a_{s_1 s_1}, a_{s_2 s_2}, \dots, a_{s_k s_k}\} \quad (3)$$

である $k \times k$ の対角行列となる。なお、独立集合 G は図1で示す算法に基づいて実装することができる。

2.2 シュールコンプリメントの構成

前節で述べたように、行列 A_1 は式(3)のような対角行列となる。この性質を利用して、方程式(2)から次元数が $\tilde{n} = n - k$ のより小さな連立1次方程式を得ることができる。方程式(2)の上式を x_1 について解くと

$$x_1 = A_1^{-1} b_1 - A_1^{-1} A_2 x_2 \quad (4)$$

である。 A_1 が対角行列なので、 A_1^{-1} は容易に計算が可能である。従って、式(4)を方程式(2)の下式 x_1 に代入すると \tilde{n} 次元の連立1次方程式

$$C x_2 = \tilde{b} \quad (5)$$

を導くことができる。ただし、

```

set G = 0
set marki = 0  i = 0, 1, ..., n
for i = 1 to n do
  if marki = 0 then
    G = G ∪ xi
    for j = i + 1 to n do
      if aij ≠ 0 then
        markj = 1
      endif
      if aji ≠ 0 then
        markj = 1
      endif
    enddo
  endif
enddo

```

図1 独立集合 G を得る方法

Fig. 1 How to obtain the independent set G

$$C = A_4 - A_3 A_1^{-1} A_2$$

$$\tilde{b} = b_2 - A_3 A_1^{-1} b_1$$

である。方程式(5)は、通常シュールコンプリメントと呼ばれる。方程式(5)を解けば、得られた解 x_2 を式(4)に代入して、残りの解 x_1 を求めることができる。従って、より次元数の小さい方程式(5)を解けば、実質、方程式(1)の解を求められることになる。しかし、シュールコンプリメントに変換するには独立集合を求めてから方程式(1)を方程式(2)のように並び換えねばならぬので、これらの処理にかかる計算コストが割高になる。

3. 近似逆行列の計算

本章では、方程式(5)に、Newton法を用いた近似逆行列を適用することを考える。さらに、近似逆行列の計算コストと記憶領域を抑えるための初期値の選択法と行列どうしの乗算を回避する計算法を提案する。

3.1 Newton法

代数方程式

$$f(x) = \frac{1}{x} - c = 0 \quad (6)$$

を考える。方程式(6)に対するNewton法は

$$x_{l+1} = x_l - \frac{f(x_l)}{f'(x_l)} = (2 - x_l c) x_l \quad (7)$$

となる。ここで、シュールコンプリメント(5)の係数行列 C の近似逆行列を N とおく。行列 N と C をそれぞれ方程式(6)の x と c に対応させると、前処理行列 N を求めるためのNewton法の漸化式は

$$N_{l+1} = (2I - N_l C) N_l \quad (8)$$

となる^{1),4)}。ただし、 N_l はNewton法の l 回目の反復における行列 N である。Newton法の収束性は、初期値 N_0 によって左右されるので、初期値を適切に選択することは非常に重要である。

表 1 N_1 を得るのにかかる演算量

Table 1 The arithmetic cost for obtaining N_1

演算項目	$K_1 = CN_0$	$K_2 = N_0K_1$	$N_1 = 2N_0 - K_2$	計
加算	0	0	\bar{n}^2	\bar{n}^2
乗算	\bar{n}^2	\bar{n}^2	\bar{n}^2	$3\bar{n}^2$
計	\bar{n}^2	\bar{n}^2	$2\bar{n}^2$	$4\bar{n}^2$

表 2 N_1v を得るのにかかる演算量

Table 2 The arithmetic cost for obtaining N_1v

$l = 1$ の場合

演算項目	$u_1 = N_0v$	$u_2 = N_0Cu_1$	$N_1v = \sum_{j=1}^2 u_j$	計
加算	0	\bar{n}^2	\bar{n}	$\bar{n}^2 + \bar{n}$
乗算	\bar{n}	$\bar{n}^2 + \bar{n}$	0	$\bar{n}^2 + 2\bar{n}$
計	\bar{n}	$2\bar{n}^2 + \bar{n}$	\bar{n}	$2\bar{n}^2 + 3\bar{n}$

$l = 2$ の場合

演算項目	$u_1 = N_0v$	$u_2 = N_0Cu_1$	$u_3 = N_0Cu_2$	$u_4 = N_0Cu_3$	$N_2v = \sum_{j=1}^4 u_j$	計
加算	0	\bar{n}^2	\bar{n}^2	\bar{n}^2	$3\bar{n}$	$3\bar{n}^2 + 3\bar{n}$
乗算	\bar{n}	$\bar{n}^2 + \bar{n}$	$\bar{n}^2 + \bar{n}$	$\bar{n}^2 + \bar{n}$	0	$3\bar{n}^2 + 4\bar{n}$
計	\bar{n}	$2\bar{n}^2 + \bar{n}$	$2\bar{n}^2 + \bar{n}$	$2\bar{n}^2 + \bar{n}$	$3\bar{n}$	$6\bar{n}^2 + 7\bar{n}$

3.2 初期値の選択

Benら²⁾は、Newton法の漸化式(8)の初期値を $N_0 = \alpha_0 C^T$ (9)

と選択すれば、必ず N_l が C^{-1} に収束することを証明している。ただし、 α_0 はある正の値であり、もっとも簡単な選び方は

$$\alpha_0 = \frac{1}{\|C\|_1 \|C\|_\infty} \quad (10)$$

ととることである。ただし、

$$\|C\|_1 = \max_j \sum_{i=1}^{\bar{n}} |c_{ij}|, \quad \|C\|_\infty = \max_i \sum_{j=1}^{\bar{n}} |c_{ij}|$$

である。従って、理論的には、初期値を式(9)のように選択するのが最良の方法である。しかし、行列 C の非零要素が多くなるにつれて、より多くのメモリを必要とする。また、行列 C^T を得るのに行列 A_1, A_2, A_3, A_4 の乗算が必要となる。従って、式(9)により初期値 N_0 を選択することは、計算とメモリのコストの観点からは望ましくない。

一方、Groszら⁵⁾は、 A_1 がブロック対角行列を生成するように独立集合を求め、ブロック対角行列 A_1 の近似逆行列をNewton法で計算している。その際に、対角行列をNewton法の初期値に選択している。このことを元にして、本稿ではNewton法の初期値 N_0 に対角行列を選択することを提案する。具体的には

$$N_0 = \begin{cases} 1/c_{ii}, & i = j \\ 0, & i \neq j \end{cases} \quad (11)$$

と設定する。式(11)で初期値 N_0 を選択するほうが、式(9)で選択するよりもNewton法の収束性が良いことがしばしばある。今、

$$C = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

である例を考える。このとき、厳密な C の逆行列は

$$C^{-1} = \begin{bmatrix} -2 & 1 \\ 5/3 & -2/3 \end{bmatrix}$$

である。式(11)に基づいて N_0 を設定すると

$$N_0 = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/6 \end{bmatrix}$$

であり、これを式(8)に代入すると

$$N_1 = \frac{-1}{4} \begin{bmatrix} -2 & 1 \\ 5/3 & -2/3 \end{bmatrix} = \frac{-1}{4} C^{-1}$$

となる。従って、この例の場合実質Newton法の反復1回で厳密な逆行列が得られることになる。それに対して、初期値を式(9)で設定すると

$$N_1 = \left(\frac{1}{99}\right)^2 \begin{bmatrix} 230 & 629 \\ 387 & 738 \end{bmatrix} \neq \gamma C^{-1}, \quad \exists \gamma \in \mathbb{R}$$

であり、1回の反復では厳密な逆行列を得られないことは明らかである。このことから、初期値 N_0 を式(11)で設定してもNewton法で近似逆行列を得ることが可能であり、かつ式(9)で初期値を設定した場合よりも早く収束することが期待できる。また、式(11)で初期値を設定すると、行列ノルム $\|C\|_1$ と $\|C\|_\infty$ を計算する必要がないので、これらにかかる計算コストも削減できる。なお、式(11)で初期値を選択した場合、 N_0 はJacobi法によって計算された前処理行列と全く同じになる。

3.3 行列どうしの乗算をしない計算法

Newton法の漸化式(8)を用いると近似逆行列 N_l を計算することができる。しかし、計算の過程に行列どうしの乗算が含まれるので、非常に大きな計算コストがかかる。また、行列どうしの乗算により、行列 N_l の元々零であった位置が非零になるので、必要となるメモリも増大することになる。今、 C が $\bar{n} \times \bar{n}$ の正則行列であり、すべての要素が非零であると仮定する。

また、行列 N_0 が式 (9) のように、対角行列であると仮定する。表 1 で示すように、この場合の N_1 を求めるための演算量は乗算と加算を合わせて $4\tilde{n}^2$ 回かかることになる。さらに、Newton 法の反復を重ねるにつれて、 N_i にフィルインが発生し、 N_i の非零要素数は N_0 以上になる。従って、 N_i を計算するには、最低でも加算と乗算が合わせて $4l\tilde{n}^2$ 回だけ必要になる。この演算コストを削減するために、本稿では行列どうしの乗算をしないで済む計算法を提案する。

非定常な反復法では、前処理行列 N_i を直接必要としているのではなく、前処理行列と任意のベクトルの積 $N_i v$ が必要である。このことを利用して、本稿では行列 C 、 N_0 だけを利用して、直接 $N_l v$ の計算を行うことを提案する。 $l=1$ と $l=2$ の場合、 $N_l v$ を C 、 N_0 と v のみで表すと、それぞれ

$$\begin{aligned} N_1 v &= (2I - N_0 C) N_0 v \\ &= 2N_0 v - N_0 C N_0 v \end{aligned} \quad (12)$$

$$\begin{aligned} N_2 v &= (2I - N_1 C) N_1 v \\ &= \{2I - (2I - N_0 C) N_0 \times \\ &\quad C(2I - N_0 C) N_0\} v \\ &= 4N_0 v - 6N_0 C N_0 v \\ &\quad + 4(N_0 C)(N_0 C N_0 v) \\ &\quad - (N_0 C)(N_0 C)(N_0 C N_0) v \end{aligned} \quad (13)$$

となる。ところで、前節の例で示したように、 N_0 を式 (11) で設定すると、Newton 法の反復が 1 回だけでも性能の良い近似逆行列が得られる可能性があることが分かった。従って、前処理を適用するポトルネックを減少させるために、本稿では Newton 法の反復を最大 2 回に設定する。このように、Newton 法の反復が 1 回と 2 回の場合にかかる加算と乗算の演算量を表 2 に示す。本来、 $N_0 v$ にかかる加算の回数は \tilde{n} であるが、 N_0 が対角行列であることを考慮すると、加算は一切必要としない。表 2 から、 $N_1 v$ 、 $N_2 v$ に要する演算量は、それぞれ $2\tilde{n}^2 + 3\tilde{n}$ 、 $6\tilde{n}^2 + 7\tilde{n}$ である。それに対して、行列 N_1 、 N_2 を Newton 法の漸化式で求めると、演算量は最低でもそれぞれ $4\tilde{n}^2$ 、 $8\tilde{n}^2$ かかることになる。従って、方程式 (5) の次元数 \tilde{n} が大きいほど $N_l v$ を式 (12)、(13) に基づいて計算するほうが、Newton 法で近似逆行列 N_l 自身を計算するよりも計算量が少なくて済む。また、 $N_l v$ を計算する方法では、近似逆行列 N_l のフィルインを一切考慮しないで済むので、あらかじめフィルインする場所のメモリを設けなくて良いという利点もある。近似逆行列 N_l については、 $N_l v$ を計算する場合、 N_0 の要素さえ記憶すれば良いので、メモリは高々 \tilde{n} しか必要としない。それに対して、 N_l を直接計算する場合、メモリは \tilde{n}^2 だけ必要となる。

4. 数値実験

本稿で提案した計算法に基づき、シュールコンプリ

表 4 シュールコンプリメントにおける計算時間の内訳
Table 4 The classification of the computation time in schur-compliments

内訳	独立集合	行と列の入れ換え	N_0 の計算
計算時間	18.0	61.0	2.0

メントに近似逆行列を適用する数値実験を行う。実験に用いた計算機の仕様は以下のとおりである。

- CPU: Pentium4 2.66GHz
- MEM: 512MB
- OS: Linux RedHat 9

[数値例] 対角要素がすべて非零であり、それ以外の非零要素の位置をランダムに設定した行列 A を 2 種類作成する。これらの行列を係数とする連立 1 次方程式を解く数値実験を行った。これら 2 種類の連立 1 次方程式をそれぞれ、“タイプ 1”、“タイプ 2”と呼ぶことにする。ただし、係数行列 A の非零要素と右辺 b の要素は $[-1, 1]$ の範囲の乱数によって決定し、いずれの方程式の次元数も 65,536 である。また、“タイプ 1”と“タイプ 2”の係数行列は非零要素の位置が全く異なるが、右辺のベクトルは同じものを用いている。ここで、“タイプ 1”、“タイプ 2”の方程式について、係数行列 A の行番号と列番号のいずれも 1 から 5000 の範囲に属する非零要素の分布を図 2 に表示する。図 2 から、“タイプ 1”、“タイプ 2”のいずれの方程式の係数行列も非零要素が全く不規則に存在していることが分かる。また、“タイプ 1”、“タイプ 2”の方程式について、シュールコンプリメントを計算すると、それらの次元数はそれぞれ 29,457、29,554 となった。Newton 法で計算した近似逆行列 N_0 、 N_1 、 N_2 を GMRES(m) 法³⁾ に右側前処理として適用し、それぞれの性能を比較した。また、シュールコンプリメントを生成せず、元の連立 1 次方程式に直接近似逆行列を適用したものと比較を行った。ここで、GMRES(m) 法の初期近似解には 0 ベクトルを設定する。数値結果としては、それぞれの相対残差ノルムの収束判定条件を

$$\|r_i\|_2 / \|b\|_2 < 1.0 \times 10^{-12}$$

とし、GMRES(m) 法がこの判定条件を満たすまでに要する計算時間と反復回数を表 3 に示す。表記法としては、近似逆行列 N_l を適用した GMRES(m) 法のことを“ N_l +GMRES(m)”としている。“タイプ 1”の方程式に関しては、シュールコンプリメントを生成しない場合、前処理を適用しても相対残差ノルムが収束判定条件を満たすことは一切なかった。それに対して、シュールコンプリメントを生成した場合、前処理なしの GMRES(m) 法はすべて収束しなかったが、近似逆行列 N_1 ないしは N_2 を適用すると GMRES(10) 法以外ではすべて収束していることが確認できる。従って、Newton 法の反復回数が 1 回から 2 回でも性能の良い近似逆行列が得られる。ここで、“タイプ 1”の方程式について、計算時間と反復回数に対する相対残差

表 3 数値例の計算結果 (time: 計算時間 (秒), iter: 反復回数)

Table 3 The numerical results (time: computation time (sec), iter: iteration steps)

(a) “タイプ 1” の場合					(b) “タイプ 2” の場合				
算法	シュールコンプリメント				算法	シュールコンプリメント			
	生成する		生成しない			生成する		生成しない	
	time	iter	time	iter		time	iter	time	iter
GMRES(10)	GMRES(10)
GMRES(20)	GMRES(20)
GMRES(30)	GMRES(30)
GMRES(40)	GMRES(40)
N_0 +GMRES(10)	N_0 +GMRES(10)
N_0 +GMRES(20)	N_0 +GMRES(20)	90.0	199
N_0 +GMRES(30)	N_0 +GMRES(30)	87.0	87
N_0 +GMRES(40)	90.0	125	N_0 +GMRES(40)	90.0	127
N_1 +GMRES(10)	N_1 +GMRES(10)	85.0	92
N_1 +GMRES(20)	87.0	120	N_1 +GMRES(20)	84.0	58
N_1 +GMRES(30)	83.0	49	N_1 +GMRES(30)	85.0	77
N_1 +GMRES(40)	84.0	59	N_1 +GMRES(40)	88.0	99	139.0	1239
N_2 +GMRES(10)	N_2 +GMRES(10)	81.0	16
N_2 +GMRES(20)	82.0	28	N_2 +GMRES(20)	82.0	26
N_2 +GMRES(30)	82.0	39	N_2 +GMRES(30)	83.0	37	64.0	591
N_2 +GMRES(40)	83.0	49	N_2 +GMRES(40)	83.0	45	28.0	221

(...): 5 分間で残差ノルムが収束しなかった場合

N_0 : Jacobi 法によって求めた前処理行列に相当する

(...): 5 分間で残差ノルムが収束しなかった場合

N_0 : Jacobi 法によって求めた前処理行列に相当する

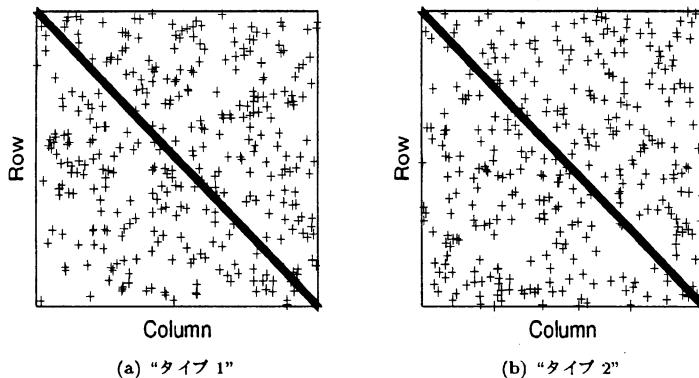
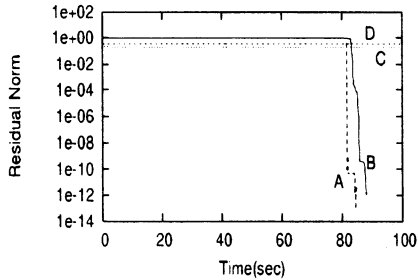


図 2 係数行列 A の非零要素の分布
Fig. 2 The distribution of the nonzero elements of the coefficient matrix A

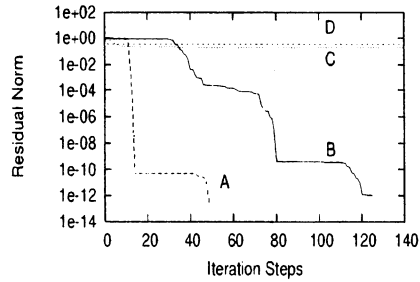
ノルムの収束の振舞いを 図 3 に示す。図 3 におけるシュールコンプリメントの例では、近似逆行列を前処理とすれば GMRES(m) 法は収束するが、80 秒付近まで相対残差ノルムが全く減少していない。これは、反復を開始する前に独立集合 G を求めてシュールコンプリメントを生成するのに時間がかかるためである。そこで、シュールコンプリメントについて、反復を開始できるまでにかかる時間の内訳を表 4 に示す。表 4 で示すように、計算時間の内訳は 3 つに分けることができる。3 つの内訳の中でもっともオーバーヘッドの高いものは、係数行列の行と列を入れ替える部分であった。

“タイプ 2” の方程式に関しては、シュールコン

プリメントを生成しない場合、近似逆行列 N_1 を適用した GMRES(m) 法で収束することがあった。特に、 N_2 +GMRES(30) 法、 N_1 +GMRES(40) 法、 N_2 +GMRES(40) 法では、シュールコンプリメントを生成した場合よりも計算時間は少なくて済んでいる。特に、この場合の GMRES(30) 法と GMRES(40) 法は、シュールコンプリメントを使用せずに直接近似逆行列 N_2 を組み込んだほうがシュールコンプリメントを使用した場合よりも早く収束している。シュールコンプリメントを使用すると、最初の 80 秒程度まで全く残差ノルムが減少しないのは、“タイプ 1” の方程式と同様、GMRES(m) 法の反復の前にオーバーヘッドがかかるためである。しかし、反復回数はシュールコン



(a) 残差ノルム vs. 計算時間



(b) 残差ノルム vs. 反復回数

図3 タイプ1: 計算時間と反復回数に対する残差ノルムの振舞い, A: N_2 +GMRES(40), B: N_0 +GMRES(40), C: N_2 +GMRES(40), D: N_0 +GMRES(40), A, B シュールコンプリメントあり, C, D シュールコンプリメントなし

Fig. 3 Type1: The behaviors of the residual norm as for the computation time and iteration steps, A: N_2 +GMRES(40), B: N_0 +GMRES(40), C: N_2 +GMRES(40), D: N_0 +GMRES(40), A, B with using Schur-Compliments, C, D without using Schur-Compliments

プリメントを使用したほうが、使用しない場合に比べて10%以下まで減少している。従って、Newton法によって計算された近似逆行列は、シュールコンプリメントに適用するほうが前処理としてより高い性能を発揮できる。また、GMRES(m)法のリスタート周期 m を変化させても安定して収束しているのは、シュールコンプリメントを使った場合である。シュールコンプリメントに対して前処理を適用すると、リスタート周期 m に関係なく残差ノルムは収束している。それに対して、シュールコンプリメントを使用せずに前処理を適用しても GMRES(10)法と GMRES(20)法では収束しなかった。従って、シュールコンプリメントを使用した方が性能が安定しているといえる。

なお、いずれの方程式についても、本稿で提案した計算法を用いずに Newton法で近似逆行列 N_i 自身を計算させてみた。しかし、いずれのケースでもメモリ不足が原因で、Newton法の反復を1回たりとも行うことができなかった。

5. まとめ

本稿では、シュールコンプリメントの近似逆行列を Newton法を用いて計算することを考えた。さらに、計算量とメモリのコストを減少させるために、対角行列を初期値に選択すること、行列どうしの乗算を一切用いない計算法を提案した。本稿の数値例の結果から、近似逆行列の初期値を対角行列に選ぶと、Newton法の反復が2回程度でも近似逆行列が十分に前処理としての機能を発揮することが分かった。また、行列どうしの演算を一切用いなくても、Newton法により近似逆行列を求められることが判明した。従って、Newton法で近似逆行列を計算するには、本稿で提案した方法で計算することが有効である。

今後の課題は、シュールコンプリメントの生成にかかるボトルネックをさらに減少させることと、本稿の数値例よりも大規模な問題で検証を行うことが挙げられる。

参考文献

- 1) G. Schulz.: Iterative Berechnung der Reziproken Matrix, *Z. Angew. Math. Mech.*, Vol. 13, pp. 57-59, (1933).
- 2) A. Ben-Israel, D. Cohen.: On Iterative Computation of Generalized Inverses and Associated projections, *SIAM J. Numer. Anal.*, Vol. 3, pp. 410-419, (1966).
- 3) Y. Saad, M. K. Schultz.: GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, Vol. 7, pp. 856-869, (1986).
- 4) V. Pan, R. Schreiber.: An Improved Newton Iteration for the Generalized Inverse of a Matrix with Applications, *SIAM J. Sci. Stat. Comp.*, Vol. 12, No. 5, pp. 1109-1130, (1991).
- 5) L. Grosz.: Preconditioning by incomplete block elimination, *Numer. Linear. Algebra. Appl.*, Vol. 7, pp. 527-541, (2000).