# 緩和係数 $\omega$ を自動決定する対角緩和準ロバスト ICCG 法の収束性

柿 原 正 伸† 藤 野 清 次††

Ajiz らにより考案されたロバスト不完全コレスキー分解は，分解中に対角項の修正を行うことにより分解の破綻が起きないように工夫した分解法として知られ，共役勾配（CG）法の前処理として有限要素法解析の分野でよく使用されている．しかし，CG 法の収束性から考えたとき改良する余地がまだ残されている．本論文では，CG 法の収束性のよりいっそうの向上を目指し，RIC 分解における対角項に対する修正量を緩和させることによって収束性の大幅な向上を実現し，かつ緩和係数 $\omega$ も自動決定する分解法を提案する．そして，数値実験によって新しい分解法つき CG 法の有効性を検証する．

# Convergence of Diagonally Relaxed Quasi RICCG Method with Automatic Decision of Relaxation Parameter $\omega$

Masanobu Kakihara† and Seiji Fujino††

A popular technique in FEM (Finite Element Method) analysis is the robust incomplete Cholesky decomposition developed by Ajiz et al. This technique is based on the idea of stabilization for diagonal entries, and no breakdown can occur during the incomplete decomposition. In this paper, we propose new preconditioning which decides automatically diagonal relaxation parameter $\omega$ for diagonal entries to enhance greatly convergence of the preconditioned CG iteration. Through numerical experiments for realistic problems, it will be made clear that the new approach insures convergence rates of PCG method.

## 1. はじめに

前処理つき共役勾配法（Preconditioned Conjugate Gradient method：以下，PCG 法と略す）は，偏微分方程式の有限要素法などによる離散化で得られる，大型で疎な対称正定値行列を係数行列とする連立一次方程式に対する数値解法としてよく使用される．特に，係数行列が $M$-行列のとき，不完全コレスキー分解つき CG 法が非常に有効であるとされる[12]．しかし，構造解析や固体力学の分野で得られる $M$-行列以外の行列の場合，不完全分解の途中で対角項の値が負になり，それ以降の分解が不可能になることが知られている．

一方，Ajiz らにより考案された不完全コレスキー分解は，分解過程で発生するフィルインの影響を考慮することにより，この問題を解決する優れた分解法である．すなわちこの方法では要素の棄却ごとに対角項の要素に修正を加え，これにより対角項の値が負にならないことを保証する[1),6)~8)]．分解が途中で破綻することがないので，この分解法はロバスト不完全コレスキー（Robust Incomplete Cholesky：以下，**RIC** と略す）分解とも呼ばれる．また，この分解法は Kaporin によりいろいろな数学的性質が明らかにされている[9)]．

本論文では，PCG 法の収束性のいっそうの向上を目指して，従来の RIC 分解における行列の対角項に対する修正量の大きさに着目した．すなわち，従来はその大きさが安全側に過大評価される傾向があり，その結果 CG 法の収束性の向上を阻害していた可能性がある点に注目した．そして，新しく緩和用のパラメータ $\omega$ を導入することによって対角項に対する修正量の大きさを緩和させ，CG 法の大幅な収束性の向上を実現する．

本論文の構成は以下のとおりである．2 章で PCG 法の概略を記述する．3 章では，従来の不完全コレスキー分解法について述べる．4 章では，RIC 分解のアルゴリズムおよび RIC 分解で現れる形式的な 2 つの行列の要素の値について記述する．5 章では，RIC 分解のロバスト性に対する考察を行う．また行列が非負

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 九州大学情報基盤センター
Computing and Communications Center, Kyushu University

定値行列になるための条件についても考察する．6 章では，対角緩和つき準ロバスト IC 分解，解析分野ごとの閾値の特徴および対角緩和係数と収束性の関係，対角緩和係数 $\omega$ の自動決定法について記述する．7 章では，数値実験結果を報告する．テスト行列には，疎行列データベースからの問題だけでなく，実際のコンクリート橋の設計で使われた応力解析の問題から生じた行列を取り上げる．最後に 8 章でまとめを行う．

## 2. PCG 法

係数行列として対称正定値行列 $A = [a_{ij}]$ を持つ次の連立一次方程式を PCG 法で解くことを考える．

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

ここで，$A$ は大きさ $n \times n$ の正方行列，$\boldsymbol{x}$，$\boldsymbol{b}$ は大きさ $n$ の解および右辺ベクトルとする．次に係数行列 $A$ を次のように行列と行列の積の形に近似分解する．

$$A \simeq U^T U \tag{2}$$

ここで，$U$ は上三角行列，上付き添字 $^T$ は転置を表す．この行列 $U$ および $U^T$ を用いて，方程式 (1) を

$$(U^{-T} A U^{-1})\ (U\boldsymbol{x}) = U^{-T}\boldsymbol{b} \tag{3}$$

と変換する．この変換によって，方程式 (1) は係数行列 $U^{-T}AU^{-1}$，解ベクトル $U\boldsymbol{x}$，右辺ベクトル $U^{-T}\boldsymbol{b}$ を持つ新たな方程式になる．ここで，方程式 (3) に対して CG 法[3] を適用し，さらにそのアルゴリズムに対して式変形を行うことにより PCG 法が得られる．PCG 法のアルゴリズムは以下のように表される．$\boldsymbol{x}_0$ は初期近似解，$\varepsilon$ は収束判定用の微小な値である．

**[PCG 法のアルゴリズム]**

$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0,\ \boldsymbol{p}_0 = (U^TU)^{-1}\boldsymbol{r}_0$
**for** $k = 0, 1, \cdots$
  $\alpha_k = \frac{(\boldsymbol{r}_k, (U^TU)^{-1}\boldsymbol{r}_k)}{(\boldsymbol{p}_k, A\boldsymbol{p}_k)}$
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
  $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$
  **if** $||\boldsymbol{r}_{k+1}||_2 \ / \ ||\boldsymbol{r}_0||_2 \le \varepsilon$ **stop**
  $\beta_k = \frac{(\boldsymbol{r}_{k+1}, (U^TU)^{-1}\boldsymbol{r}_{k+1})}{(\boldsymbol{r}_k, (U^TU)^{-1}\boldsymbol{r}_k)}$
  $\boldsymbol{p}_{k+1} = (U^TU)^{-1}\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$
**end for**

## 3. 不完全コレスキー分解

本章では，係数行列 $A$ に対するフィルインを考慮した不完全コレスキー分解[17] について述べる．この不完全コレスキー分解は以下の式ように表される．

$$A = U^T U - R - R^T. \tag{4}$$

ここで，行列 $U$ と $R$ は各々上三角行列とする．行列 $U$ は PCG 法のアルゴリズム中においてその転置との積 $U^TU$ の形で現れる．一方，行列 $R$ は，行列 $U$ のスパース（疎）性の保持のための形式的な行列を意味し，PCG 法のアルゴリズム中には現れない．上記の不完全コレスキー分解 (4) を行うとき，あらかじめ閾値として用いるパラメータ tol を設定し，分解の対象の要素の大きさが閾値 tol よりも大きいときはその要素の計算を実行し，反対に小さいときはその要素の値は零とおいて計算を進めることにする．すなわち，上三角行列 $U = [u_{ij}]$ の各行 $i = 1, \ldots, n$ に対して，以下の手順で計算を行う．

**[不完全コレスキー分解の手順]**

$$a^*_{ij} = a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}, \quad (j = i+1, \ldots, n) \tag{5}$$

$$u_{ii} = \sqrt{\overline{a_{ii}} - \sum_{k=1}^{i-1} u_{ki}^2}, \tag{6}$$

$$u_{ij} = \begin{cases} a^*_{ij} \ / \ u_{ii} & \frac{|a^*_{ij}|}{\sqrt{(\overline{a_{ii}})(\overline{a_{jj}})}} > \mathrm{tol} \text{ のとき} \\ 0 & \frac{|a^*_{ij}|}{\sqrt{(\overline{a_{ii}})(\overline{a_{jj}})}} \le \mathrm{tol} \text{ のとき} \end{cases} \tag{7}$$

$$(j = i+1, \ldots, n).$$

ここで，要素 $a^*_{ij}$ は分解過程においては作業用配列として使われ，分解が終了した後は行列 $U$ の非対角要素になる要素を表す．一方，対角要素 $\overline{a_{ii}}$ は，通常の不完全コレスキー分解では，(i) 対角要素 $\overline{a_{ii}} = a_{ii}$ と置かれる．しかし，場合によっては，式 (6) の右辺の平方根の中の式の値が負になり，分解が破綻することがある．

そこで，次章で扱う RIC 分解では，(ii) 対角要素 $\overline{a_{ii}} = a_{ii}$ と置き，式 (7) において要素 $u_{ij}$ が棄却されるごとに対角要素 $\overline{a_{ii}}$ に適当な修正を加えることで，式 (6) での分解の破綻を防ぐ．数値実験では RIC 分解と従来から用いられている，分解を行う前に対角要素の値を (iii)$\overline{a_{ii}} = \gamma a_{ii}$ $(\gamma > 1.0)$ と置くことで，分解の破綻を防ぐ IC 分解[13]（以後，加速係数つき IC 分解と呼ぶ）との比較を行った．

## 4. RIC 分解

前章で述べた不完全コレスキー分解の途中で分解の破綻が起きないように改良された RIC 分解[1),5)] について記述する．RIC 分解では分解の途中で破綻が起きないことが理論的に保障されていることから，**ロバストである**と呼ばれる[1),9)]．本章では，RIC 分解のアルゴリズムとロバスト性について記述する．

### 4.1 RIC 分解の行列表現

RIC 分解は，係数行列 $A$ を

$$A = U^T U - R - R^T - D \tag{8}$$

と分解する方法である．前章と同様に，行列 $U$ は分解後の上三角行列を表し，行列 $R$ は分解手順の説明用の形式的な行列を表す．一方，対角行列 $D$ は，行列 $U$ の要素が棄却されたとき対角項の符号が負にならないように対角項を修正する役割を果たすが，PCG 法のアルゴリズムには用いられない．

### 4.2 RIC 分解のアルゴリズム

RIC 分解のアルゴリズムを以下に示す．ただし，要素 $\overline{a_{ii}}$，$\overline{a_{jj}}$ は行列 $A$ の第 $i$，$j$ 行の対角要素に各々対応し，分解過程において次々と更新される．一方，$a^*_{ij}$ は最終的に上三角行列 $U$ の非対角要素 $u_{ij}$ になる要素を表す．また，tol は要素 $u_{ij}$ の棄却判定のための閾値とする．

**[RIC 分解のアルゴリズム]**

**for** $i = 1, \cdots, n$
　$\overline{a_{ii}} = a_{ii}$
**end for**
**for** $i = 1, \cdots, n$
　**for** $j = i+1, \cdots, n$
　　$a^*_{ij} = a_{ij}$
　**end for**
　**for** $k = 1, \cdots, i-1$
　　**for** $j = i+1, \cdots, n$
　　　$a^*_{ij} = a^*_{ij} - u_{ki} u_{kj}$（非対角項の計算）
　　**end for**
　**end for**
　―――――――――――――――――――-
　**for** $j = i+1, \cdots, n$
　　$\xi = |a^*_{ij}| / \sqrt{(\overline{a_{ii}})(\overline{a_{jj}})}$
　　**if** $\xi \le \mathrm{tol}$ **then**
　　　$a^*_{ij} = 0$（閾値より小さい要素を棄却）
　　　$\overline{a_{ii}} = (1+\xi)\overline{a_{ii}}$（対角項の修正）
　　　$\overline{a_{jj}} = (1+\xi)\overline{a_{jj}}$（対角項の修正）
　　**end if**
　**end for**
　―――――――――――――――――――-
　$u_{ii} = \sqrt{\overline{a_{ii}}}$（対角項を求める）
　**for** $j = i+1, \cdots, n$
　　$u_{ij} = a^*_{ij} / u_{ii}$（非対角項を求める）
　　$\overline{a_{jj}} = \overline{a_{jj}} - u_{ij}^2$（対角項の計算）
　**end for**
**end for**

### 4.3 形式的行列 $R$ と行列 $D$ の要素の値

行列 $R = [r_{ij}]$ および対角項を修正する行列 $D = [d_k]$ の要素の値は次のように表される．ただし $\xi = |a^*_{ij}| / \sqrt{(\overline{a_{ii}})(\overline{a_{jj}})}$ とする．各記号の意味は前 4.2 節と同じである．

(i) $\xi \le \mathrm{tol}$ のとき（閾値より小さい要素を棄却）

$$r_{ij} = a^*_{ij}, \tag{9}$$

$$d_k = \begin{cases} \sqrt{\frac{(\overline{a_{ii}})}{(\overline{a_{jj}})}}\, a^*_{ij} & (k = i \text{ のとき}), \\ \sqrt{\frac{(\overline{a_{jj}})}{(\overline{a_{ii}})}}\, a^*_{ij} & (k = j \text{ のとき}), \\ 0 & (k \ne i, j \text{ のとき}). \end{cases} \tag{10}$$

(ii) $\xi > \mathrm{tol}$ のとき

$$r_{ij} = 0, \quad d_k = 0. \tag{11}$$

## 5. RIC 分解のロバスト性に対する考察

ここでは，RIC 分解がロバスト性を持つための対角項の修正量の大きさについて考える．

### 5.1 ロバスト性保持と対角項の修正

はじめに，分解の計算途中で現れる要素 $a^*_{ij}$ の絶対値の大きさがあらかじめ設定した閾値よりも小さいとき，その要素を棄却する行列 $R$ を次のように表す．

$$R = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & 0 & a^*_{ij} & \vdots \\ \vdots & 0 & 0 & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{12}$$

要素 $a^*_{ij}$ が棄却されるとき，行列 $A$ の第 $i$ 行と第 $j$ 行の対角項に加えられる未確定の修正量を各々 $d_i$，$d_j$ とする．このとき修正対角行列 $D$ は次のように表される．

$$D = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & d_i & 0 & \vdots \\ \vdots & 0 & d_j & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{13}$$

ここで，$S = R + R^T + D$ で表される行列 $S$ を導入すると，行列 $S$ は次のように表される．

$$S = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & d_i & a^*_{ij} & \vdots \\ \vdots & a^*_{ij} & d_j & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{14}$$

行列 $A$ が正定値行列かつ行列 $S$ が非負定値行列のとき，行列 $(A+S)$ は正定値行列となり式 (8) で表される RIC 分解では破綻が起きない．そこで，行列 $S$ が非負定値行列になるように行列 $D$ の対角要素 $d_i$，

$d_j$ を決める方法について次に考える．

### 5.2 対角要素 $d_i$，$d_j$ の決定

RIC 分解の原論文[1),5)] では行列 $S$ が非負定値行列になる対角要素 $d_i$，$d_j$ の導出に関して詳細な説明がなされていないため，ここでは，RIC 分解の対角要素 $d_i$，$d_j$ を導出する方法について記述する．

行列 $S$ が非負定値行列になる条件について考えるうえで，簡単のために，行列 $S$ の第 $i$ 行と第 $j$ 行の非零要素を取り出した $2 \times 2$ の行列 $S'$ を考える．行列 $S'$ は対称行列であるので，固有値がすべて実数であることに注意すると，行列 $S'$ が非負定値行列となるための必要十分条件は，以下の 2 つの性質を同時に満足することである．ただし，$\mathrm{tr}(S')$ は行列 $S'$ の対角要素の総和を，$\det(S')$ は行列 $S'$ の行列式を，$\lambda_1$，$\lambda_2$ は行列 $S'$ の 2 つの固有値を各々表す．

［**性質 1**］$\lambda_1\lambda_2 = \det(S') = d_i d_j - {a^*_{ij}}^2 \geq 0$,

［**性質 2**］$\lambda_1 + \lambda_2 = \mathrm{tr}(S') = d_i + d_j \geq 0$.

したがって，行列 $S'$ を拡大した行列 $S$ についても同様に，上の 2 つ性質を満たすように $d_i$，$d_j$ を選べば非負定値行列となる．ここで,［**性質 1**］について，修正量 $d_i$，$d_j$ が $d_i d_j - {a^*_{ij}}^2 > 0$ を満たす場合，過剰な修正量から，前処理行列として用いる行列 $U^T U$ の係数行列 $A$ への近似の度合いが悪化することがある．そのため,［**性質 1**］の特別な場合：

［**性質 1a**］$d_i d_j - {a^*_{ij}}^2 = 0$

を考えることにする．4 章で述べた RIC 分解では要素 $a^*_{ij}$ が棄却されるとき，対角項の修正量：$d_i$，$d_j$ は［**性質 1a**］と［**性質 2**］を同時に満たし，かつ $d_i$，$d_j$ が分解中での係数行列の対角項 $\overline{a_{ii}}$，$\overline{a_{jj}}$ の大きさと比例するように定められる．すなわち，次の 3 つの関係式

$$d_i d_j - {a^*_{ij}}^2 = 0, \tag{15}$$

$$d_i + d_j \geq 0, \tag{16}$$

$$d_i : d_j = \overline{a_{ii}} : \overline{a_{jj}}, \tag{17}$$

が成り立つように定めると，結局 $d_i$，$d_j$ は $d_i = \sqrt{\frac{(\overline{a_{ii}})}{(\overline{a_{jj}})}}|a^*_{ij}|$，$d_j = \sqrt{\frac{(\overline{a_{jj}})}{(\overline{a_{ii}})}}|a^*_{ij}|$ となり，対角項に対する修正行列 $D$ は次の式で与えられる．

$$D = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \sqrt{\frac{(\overline{a_{ii}})}{(\overline{a_{jj}})}}|a^*_{ij}| & 0 & \vdots \\ \vdots & 0 & \sqrt{\frac{(\overline{a_{jj}})}{(\overline{a_{ii}})}}|a^*_{ij}| & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{18}$$

以上の議論より，前述の RIC 分解のアルゴリズム中の 2 つの線で挟んだ部分が得られる．

### 5.3 対角項の修正に関する関連研究

対角項の修正行列 $D$ に関する関連研究として，Dickinson ら[2)] や Hladik ら[4)] による提案法がある．そこでは修正量 $d_i$，$d_j$ の値が［**性質 1a**］と［**性質 2**］を満たし，かつ $d_i$，$d_j$ の大きさが同じになるように定められ，以下に示す比較的単純な修正行列 $D$ が用いられた．

$$D = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & |a^*_{ij}| & 0 & \vdots \\ \vdots & 0 & |a^*_{ij}| & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{19}$$

式 (18) と式 (19) はともにロバスト性を保証する修正行列であるが修正量が異なる．これらの異なった修正量を導入した数学的な根拠は原論文では特に記載されていない．本論文では 4 章で記述した RIC 分解の改良を行うため，対角項の修正量が分解中での係数行列の対角項の大きさに比例するように定めるという条件を用いた式 (18) で表される修正行列 $D$ を用いる．

## 6. 対角緩和つき準ロバスト IC 分解

ここでは，PCG 法の収束性をよりいっそう向上させた**対角緩和つき準ロバスト IC 分解**を新しく提案する．元の RIC 分解と異なり，この分解法にはロバスト性の理論的保証はない．しかし，収束のロバスト性はかなり保持しているので，この特長を**準ロバスト性**と呼ぶことにする．

### 6.1 対角緩和つき準ロバスト IC 分解のアルゴリズム

ここでは，修正行列 $D$ を可変パラメータ $\omega$（$0 < \omega \leq 1$）を用いて次の式 (20) で表す．

$$D = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \omega\overline{a_{ii}} & 0 & \vdots \\ \vdots & 0 & \omega\overline{a_{jj}} & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}. \tag{20}$$

この修正行列 $D$ は，パラメータ $\omega$ を付加したことにより，ロバストではなくなるが，修正行列 $D$ の第 $i$ 行と第 $j$ 行の要素の大きさは，分解中の係数行列 $A$ の対角項の $\overline{a_{ii}}$，$\overline{a_{jj}}$ の大きさに比例する．以下，このような対角項の修正量を緩和させる処理を**対角緩和**（diagonal relaxation），パラメータ $\omega$ を**対角緩和係数**と呼ぶ．さらに，対角緩和つき準ロバスト RIC 分解を前処理として用いる CG 法を**対角緩和つき準**

**RICCG 法**と呼ぶ．RIC 分解における対角緩和の処理は，4.2 節の RIC 分解のアルゴリズム中の 2 つの線で挟んだ部分を以下のように書き直せば得られる．

**for** $j = i+1, \cdots, n$
　$\xi = |a^*_{ij}| / \sqrt{(\overline{a_{ii}})(\overline{a_{jj}})}$
　**if** $\xi \le \mathrm{tol}$ **then**
　　$a^*_{ij} = 0,$
　　$\overline{a_{ii}} = (1+\omega)\overline{a_{ii}},$
　　$\overline{a_{jj}} = (1+\omega)\overline{a_{jj}}$
　**end if**
**end for**

### 6.2　解析分野ごとの閾値 tol の特徴

計算時間の観点で最適な閾値 tol は解析分野，離散化条件や計算機の特性などにより異なり，理論的に最適な閾値を自動決定することは難しい．ただ，次に示すように，解析分野によって，ある程度最適に近い閾値は経験的に与えることはできる．

**表 1** に構造解析と電磁界解析の行列に対して対角緩和つき準ロバスト RICCG 法の閾値 tol を 0.0001，0.0005，0.001，0.005，0.01，0.05，0.1 の 7 通り，対角緩和係数 $\omega$ の閾値に対する倍率 $\omega$ を 0.000005, 0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005，0.01 の 8 通り変化させて計算した場合，最も計算時間が短くなったときの閾値 tol と対角緩和係数 $\omega$ の組合せを示す．ここでは行列データベース[10),11),15),18)]や実際の電磁界解析問題で得られた行列を用いた．

表 1 に示した結果より，構造解析の分野の問題に対する閾値の大部分は 0.001〜0.005 の範囲にあり，一方，電磁界解析の分野の問題に対する閾値の大部分は 0.01〜0.05 の範囲にあることが分かる．このように，解析分野によっては，経験的に最適に近い閾値が得られた．

### 6.3　対角緩和係数 $\omega$ と収束性の関係

ここでは，対角緩和係数 $\omega$ と CG 法の収束性との関係について考える．**図 1** に行列 CT20STIF と行列 PWTK における対角緩和つき準 RICCG 法の対角緩和係数 $\omega$ と反復回数の関係を示す．ただし，分解中で用いる閾値 tol が 0.001 の結果である．この図より，次のことが分かる．対角緩和係数 $\omega$ の値が小さすぎると分解の破綻が起きる．一方，分解が成功するときは対角緩和係数 $\omega$ の値の増加に従って，対角緩和つき準 RICCG 法の反復回数は増加する傾向がある．分解が成功するとき，最適に近い対角緩和係数 $\omega$ は，2 本の線の各左端に位置するような小さな値のときに得られる可能性が高いと思われる．

**表 1**　構造解析と電磁界解析の行列に対する対角緩和つき準ロバスト RICCG 法の最適な閾値 tol と対角緩和係数 $\omega$ の組合せ
Table 1　Optimal combination of threshold parameters tol and relaxation parameters $\omega$ used in Diagonally Relaxed Quasi RICCG method for matrix in the field of structural and electromagnetic analysis.

| | 行列 | 次元数 | tol | $\omega$ |
|---|---|---|---|---|
| 構造解析 | CT20STIF | 52,329 | 0.01 | 0.0005 |
| | SMT | 25,710 | 0.01 | 0.0001 |
| | TUBE1-2 | 21,498 | 0.005 | 0.0001 |
| | SHIPSEC5 | 179,860 | 0.005 | 0.00005 |
| | ENGINE | 143,571 | 0.005 | 0.00005 |
| | PWTK | 217,918 | 0.001 | 0.00005 |
| | BEAM | 10,626 | 0.001 | 0.00001 |
| | CABLE | 59,002 | 0.001 | 0.00001 |
| | BRIDGE | 341,055 | 0.001 | 0.00001 |
| | SHIPSEC8 | 114,919 | 0.001 | 0.00001 |
| | SHIPSEC1 | 140,874 | 0.001 | 0.000005 |
| | BCSSTK35 | 30,237 | 0.0005 | 0.00001 |
| | S3DKT3M2 | 90,449 | 0.0005 | 0.000005 |
| 電磁界解析 | BRAKE | 769,496 | 0.05 | 0.01 |
| | THINHEAD | 174,395 | 0.05 | 0.005 |
| | CUSPHEAD | 1,179,789 | 0.05 | 0.005 |
| | CUBEMAG1 | 438,440 | 0.01 | 0.001 |
| | CUBEMAG2 | 1,490,460 | 0.01 | 0.001 |
| | CUBEMAG3 | 3,545,680 | 0.01 | 0.001 |
| | HEAD | 647,701 | 0.01 | 0.001 |
| | IRONCOIL | 990,000 | 0.01 | 0.0001 |
| | MOTOR | 10,899 | 0.01 | 0.0001 |

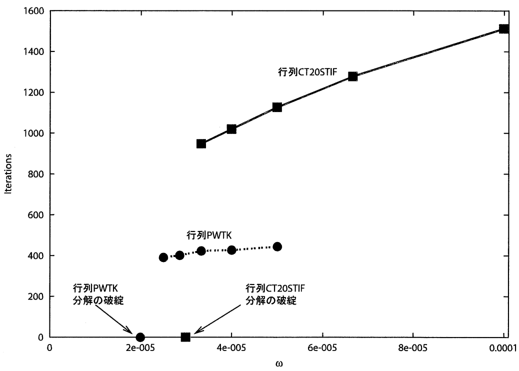


**図 1**　行列 CT20STIF と行列 PWTK における対角緩和つき準 RICCG 法の緩和係数 $\omega$ と反復回数の関係
Fig. 1　Relaxation parameter $\omega$ v.s. iterations in Diagonally Relaxed Quasi RICCG method for matrices CT20STIF and PWTK.

### 6.4　対角緩和係数 $\omega$ の自動決定

前述した対角緩和つき準 RICCG 法の対角緩和係数の特徴を利用して，対角緩和係数 $\omega$ を自動的に決定する方法を考える．具体的には **図 2** に対角緩和係数 $\omega$ の自動調節の手順を示す．ただし，対角緩和係数 $\omega$ は値の小さなる閾値よりもさらに微小な値（たとえば閾値の数十分の 1 のような値）となり取扱いや識別がしにくいため，対角緩和係数の表現に対しては閾値に

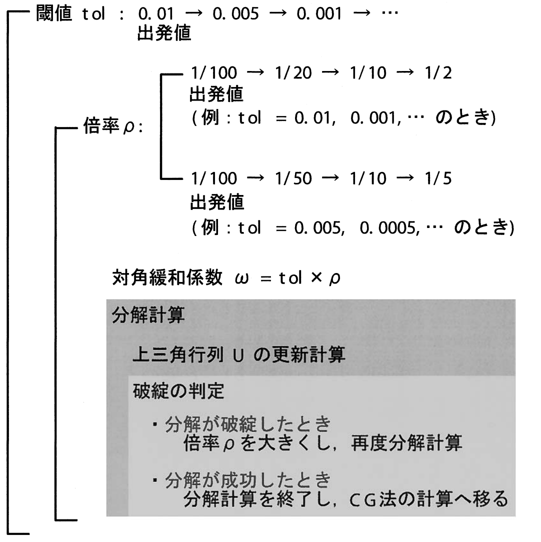


**図 2** 対角緩和つき準 RIC 分解における対角緩和係数 $\omega$ の自動決定の手順

Fig. 2 Procedure of automatic decision for parameter $\omega$ in Diagonally Relaxed Quasi RIC factorization.

対する倍率 $\rho$ を導入した．

後述の数値実験では，図 2 に示す手順に沿って閾値 tol と対角緩和係数 $\omega$ の値を定めた．たとえば，閾値 tol を $10^{-2}$ や $10^{-3}$ などと設定した場合は，倍率 $\rho$（$\omega = $ tol $\times \rho$ とする）は分解が成功するまで，$1/100 \rightarrow 1/20 \rightarrow 1/10 \rightarrow 1/2$ のように変化させる．また，閾値 tol を $5 \times 10^{-3}$ や $5 \times 10^{-4}$ などと設定した場合は，倍率 $\rho$ は $1/100 \rightarrow 1/50 \rightarrow 1/10 \rightarrow 1/5$ のように変化させる．2 つのケースに分けた理由は係数 $\omega$ の値が 5 や 10 などの扱いやすい数にするためである．

- 閾値 tol の値を，大きい値から小さい値にだんだんと変化させた理由は，閾値 tol が大きいほど分解に要する時間が短いため，仮に分解の破綻が起きた場合でも分解を再計算する手間がより少なくなるためである．
- 一方，倍率 $\rho$ の値を小さい値から大きい値にだんだんと変化させた理由は，この場合分解の破綻が早い段階で起こることが多くなるが，反復計算において，最短に近い計算時間となる可能性が高いためである
- 分解の破綻が起きたときの緩和係数の増加量を上で述べた手順よりも小さく（たとえば 2 倍ずつに）した場合，分解回数が多くなるが，分解が成功したときは反復計算時間が短くなる傾向となる．一方，緩和係数の増加量を大きく（たとえば 10 倍ずつに）したときは分解の破綻回数が少なくなるが，性能の良い前処理行列は作られず CG 法の収束性は悪くなる傾向となる．

## 7. 数値実験

### 7.1 計算機環境と計算条件

数値実験は，CPU：Intel Pentium 4（クロック周波数 3.2 GHz），主メモリ：2 Gigabytes 搭載の HP 社 workstation xw4100 を使用して行った．コンパイルは Intel Fortran Compiler version 7.1 を用い，最適化オプションは-O2（推奨される-O3 よりも速かった）を使用した．計算はすべて倍精度演算で行った．CG 法の収束判定条件は相対残差 $L_2$ ノルム：$||\boldsymbol{r_{k+1}}||_2/||\boldsymbol{r_0}||_2$ の値が $10^{-8}$ 以下のときとした．問題 1 の右辺項は厳密解がすべて 1 となるように定めた．一方，問題 2 の右辺項は解析条件の中の荷重条件から得られる値とした．初期近似値 $\boldsymbol{x}_0$ はすべて 0 とした．最大反復回数は行列の次元数と同じ値とした．また，行列はあらかじめ対角項をすべて 1 に正規化した．プログラムの実装について，データ構造は CCS 形式（Harwell-Boeing 形式）[11] を用い，上三角行列の非零要素数が計算前にあらかじめ分からないため，システムが利用できる最大メモリ量を調べ，その大きさに応じて一次元配列を大きく確保している．

### 7.2 テスト問題

以下の 2 種類の問題をテスト用の問題として取り上げた．行列は全部で 6 つである．

- 問題 1：複数の疎行列データベースから選んだ 3 つの行列．
- 問題 2：実際のコンクリート橋梁の応力解析で現れた 3 つの行列．

**問題 1**

3 つのテスト行列の主な特徴を**表 2** に示す．行列 CT20STIF と PWTK はフロリダ大学の疎行列データベース[18] から，行列 S3DKT3M2 は Matrix Market [11] から各々ダウンロードして数値実験で使用した．

**問題 2**

有限要素法による構造解析システム[19] を用いて作成した 3 つの行列をテストした．**表 3** に 3 つのテスト行列の主な特徴を示す[14),16)]．

- 行列 BEAM は橋梁間にトラックによる荷重を課したときの応力解析の問題で，離散化はシェル要素のみで行った．一般に，シェル要素による離散化行列の解析は難しく，収束までに多くの反復回数が必要になるとされる．
- 行列 CABLE は橋桁のケーブル固定部分の応力解析で生じた問題で，ソリッド要素だけで離散化

**表 4**　行列 CT20STIF，PWTK，S3DKT3M2 に対する前処理なしと従来の前処理つき CG 法の数値実験結果

Table 4　Numerical results of nonpreconditioned and traditional preconditioned CG methods for matrices CT20STIF，PWTK and S3DKT3M2.

| | CT20STIF | | | PWTK | | | S3DKT3M2 | | |
|---|---|---|---|---|---|---|---|---|---|
| 前処理＋ CG | 反復回数 | 合計時間 | メモリ量 | 反復回数 | 合計時間 | メモリ量 | 反復回数 | 合計時間 | メモリ量 |
| 前処理なし | 26,063 | 341 | 20.0 | 16,844 | 920 | 85.9 | 40,579 | 736 | 29.2 |
| IC | max | - | - | max | - | - | max | - | - |
| 加速係数つき IC | 3,936 | 103 | 20.4 | 4,680 | 491 | 87.6 | 14,918 | 526 | 29.9 |

**表 2**　問題 1 のテスト行列の主な特徴

Table 2　Description of tested matrices for problem 1.

| 行列 | 次元数 | 非零要素数 | 解析内容 |
|---|---|---|---|
| CT20STIF | 52,329 | 2,698,463 | エンジン部分の剛性行列 |
| PWTK | 217,918 | 11,634,424 | 加圧風洞に関する剛性行列 |
| S3DKT3M2 | 90,449 | 3,753,461 | シリンダーの FEM 解析 |

**表 3**　問題 2 のテスト行列の特徴

Table 3　Description of tested matrices for problem 2.

| | 行列 | | |
|---|---|---|---|
| 項目 | BEAM | CABLE | BRIDGE |
| 次元数 | 10,626 | 59,002 | 341,055 |
| 非零要素数 | 233,268 | 1,986,094 | 11,302,638 |
| 平均バンド幅 | 214 | 1,741 | 1,510 |
| 総ノード数 | 1,977 | 20,194 | 112,235 |
| 総要素数 | 2,832 | 16,084 | 91,802 |

を行った．ソリッド要素のみによる解析のとき，シェル要素の場合と比較して，行列の条件数が小さくなるが次元数は大きくなる傾向がある．

- 行列 BRIDGE は長さ約 100 m のコンクリート橋上に複数台のトラックを載せたときの応力解析の問題である．ただし，縦断面での対称性を利用して解析モデルは端の片側半分である．離散化にはシェル要素とソリッド要素の両方を使用した．

### 7.3　実験結果と考察

#### 7.3.1　問題 1 に対する実験結果

**表 4** に 3 つの行列 CT20STIF, PWTK, S3DKT3M2 に対する前処理なしと従来型の前処理つき CG 法の数値実験の結果を示す．表中の “IC” とは，フィルインを考慮しない IC 分解[12] を，“加速係数つき IC” とは，不完全コレスキー分解のときに係数行列 $A$ の対角項に加速係数を乗じたフィルインを考慮しない IC 分解を各々表す．また，“反復回数” は CG 法が収束するまでの反復回数，“合計時間” は前処理行列の作成に要した時間と CG 法の反復計算時間の合計時間とする．計算の単位はすべて秒，メモリ量の単位はすべて Megabytes とする．反復回数の欄の “max” は CG 法が最大反復回数で収束しなかったことを表す．表 4 の結果から，フィルインを考慮しない ICCG 法はすべての問題に対して収束せず，また前処理なしの CG 法は収束するが反復回数が多く，3 つの行列が解き難い問題であることが分かる．

**表 5**，**表 6**，**表 7** に行列 CT20STIF，PWTK，S3DKT3M2 に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の数値実験の結果を各々示す．ここでは，閾値 tol の値は 0.05 から 0.0001（行列 S3DKT3M2 のときは 0.00005）まで変化させた．また，**対角緩和つき準 RICCG 法**の対角緩和係数 $\omega$ の値については，6.2 節の対角緩和係数 $\omega$ の自動決定の手順に従い，閾値 tol に対する倍率 $\rho$ の値を変化させた．表中の倍率 $\rho$ の値は分解計算が終了したときの値である．

表中の項目について，“分解回数” は**対角緩和つき準 RICCG 法**において分解の計算が終了するまでに実行した分解の回数を表す．また，“Pre 時間” は前処理行列の計算時間すなわち分解計算に要した分解回数分の全時間を，“CG 時間” は CG 法の反復計算時間を，“合計時間” は前処理時間と CG 時間の合計時間を各々表す．さらに，“時間比 1” は各閾値に対する RICCG 法（下段）の合計時間を 1.0 としたときの**対角緩和つき準 RICCG 法**（上段）の合計時間の比を，同様に “時間比 2” は加速係数つき ICCG 法（表 4 および**表 8** の合計時間を 1.0 としたときの**対角緩和つき準 RICCG 法**または RICCG 法の合計時間の比を各々表す．表中の太字の数値は調べたすべての閾値 tol の値の中で 2 つの解法の各々計算時間が最も短いものを示す．

表 5～表 7 の結果から計算時間やメモリ量について，以下のことが分かる．

- 行列 PWTK，S3DKT3M2 に対して，**対角緩和つき準 RICCG 法**の最短の合計時間は加速係数つき ICCG 法のそれに比べてその比率（表中の “時間比 2” の欄）が各々0.20～0.52 および 0.10～0.44 の範囲に各々あり，改善の度合いが非常に大きい．
- 閾値が小さい場合（tol = 0.0001 のとき）を除いて，**対角緩和つき準 RICCG 法**の計算時間は RICCG 法よりも短い．

**表 5** 行列 CT20STIF に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の実験結果
Table 5 Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix CT20STIF.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/10 | 0.005 | 3 | 6,187 | 0.29 | 119 | 119 | 0.64 | 1.15 | 28.1 |
| | - | - | - | 10,077 | 0.19 | 185 | 185 | 1.0 | 1.79 | 26.8 |
| 0.01 | 1/20 | 0.0005 | 2 | 2,565 | 1.66 | 69.2 | **70.8** | 0.56 | **0.68** | 43.1 |
| | - | - | - | 4,985 | 0.75 | 124 | 125 | 1.0 | 1.21 | 39.6 |
| 0.005 | 1/10 | 0.0005 | 3 | 2,578 | 4.34 | 80.9 | 85.2 | 0.78 | 0.82 | 50.6 |
| | - | - | - | 3,633 | 1.35 | 107 | 109 | 1.0 | 1.05 | 48.0 |
| 0.001 | 1/20 | 0.00005 | 2 | 1,127 | 16.1 | 54.8 | 70.9 | 0.74 | 0.68 | 84.0 |
| | - | - | - | 1,984 | 5.56 | 90.2 | **95.8** | 1.0 | **0.93** | 77.5 |
| 0.0005 | 1/50 | 0.00001 | 2 | 651 | 39.4 | 38.6 | 78.1 | 0.78 | 0.75 | 107 |
| | - | - | - | 1,626 | 12.2 | 87.8 | 100 | 1.0 | 0.97 | 96.3 |
| 0.0001 | 1/100 | 0.000001 | 1 | 314 | 144 | 28.5 | 172 | 1.14 | 1.66 | 185 |
| | - | - | - | 852 | 81.8 | 69.1 | 150 | 1.0 | 1.45 | 162 |

**表 6** 行列 PWTK に対する対角緩和つき準 RICCG 法（上段）と RICCG 法（下段）の実験結果
Table 6 Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix PWTK.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/10 | 0.005 | 3 | 3,069 | 1.04 | 255 | 256 | 0.69 | 0.52 | 129 |
| | - | - | - | 4,650 | 0.96 | 370 | 371 | 1.0 | 0.75 | 125 |
| 0.01 | 1/20 | 0.0005 | 2 | 1,185 | 3.20 | 131 | 135 | 0.51 | 0.27 | 179 |
| | - | - | - | 2,509 | 2.39 | 257 | 260 | 1.0 | 0.52 | 167 |
| 0.005 | 1/10 | 0.0005 | 3 | 1,116 | 5.54 | 142 | 148 | 0.74 | 0.30 | 207 |
| | - | - | - | 1,649 | 3.90 | 195 | 199 | 1.0 | 0.40 | 197 |
| 0.001 | 1/20 | 0.00005 | 2 | 445 | 16.8 | 85.5 | **102** | 0.65 | **0.20** | 313 |
| | - | - | - | 812 | 12.3 | 142 | 155 | 1.0 | 0.31 | 293 |
| 0.0005 | 1/100 | 0.000005 | 1 | 336 | 34.3 | 78.7 | 113 | 0.93 | 0.23 | 395 |
| | - | - | - | 488 | 19.4 | 102 | **121** | 1.0 | **0.24** | 350 |
| 0.0001 | 1/100 | 0.000001 | 1 | 134 | 126 | 42.0 | 168 | 1.05 | 0.34 | 600 |
| | - | - | - | 301 | 71.6 | 87.5 | 159 | 1.0 | 0.32 | 532 |

**表 7** 行列 S3DKT3M2 に対する対角緩和つき準 RICCG 法（上段）と RICCG 法（下段）の実験結果
Table 7 Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix S3DKT3M2.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/10 | 0.005 | 3 | 8,560 | 0.48 | 232 | 233 | 0.47 | 0.44 | 44.0 |
| | - | - | - | 19,683 | 0.23 | 488 | 488 | 1.0 | 0.92 | 39.7 |
| 0.01 | 1/20 | 0.0005 | 2 | 3,410 | 1.04 | 124 | 125 | 0.51 | 0.23 | 62.2 |
| | - | - | - | 6,990 | 0.80 | 240 | 241 | 1.0 | 0.45 | 59.5 |
| 0.005 | 1/50 | 0.0001 | 2 | 2,357 | 2.72 | 101 | 103 | 0.49 | 0.19 | 74.4 |
| | - | - | - | 5,467 | 1.08 | 209 | 210 | 1.0 | 0.39 | 67.5 |
| 0.001 | 1/100 | 0.00001 | 1 | 726 | 5.11 | 52.3 | 57.4 | 0.36 | 0.10 | 124 |
| | - | - | - | 2,780 | 2.76 | 156 | 159 | 1.0 | 0.30 | 95.8 |
| 0.0005 | 1/100 | 0.000005 | 1 | 547 | 7.89 | 47.9 | **55.8** | 0.37 | **0.10** | 157 |
| | - | - | - | 2,160 | 4.23 | 146 | 150 | 1.0 | 0.28 | 116 |
| 0.0001 | 1/100 | 0.000001 | 1 | 320 | 31.8 | 42.9 | 74.8 | 0.53 | 0.14 | 264 |
| | - | - | - | 1,151 | 16.0 | 123 | **139** | 1.0 | **0.26** | 205 |
| 0.00005 | 1/100 | 0.0000005 | 1 | 246 | 64.9 | 38.6 | 103 | 0.70 | 0.19 | 324 |
| | - | - | - | 847 | 35.4 | 112 | 147 | 1.0 | 0.27 | 266 |

- 調べたすべての閾値に対する平均計算時間について，**対角緩和つき準 RICCG 法**は RICCG 法に比べて，行列 CT20STIF では約 0.77，行列 PWTK では約 0.76，行列 S3DKT3M2 では約 0.49 となった．
- **対角緩和つき準 RICCG 法**の使用メモリ量はす

**表 8**　行列 BEAM，CABLE，BRIDGE に対する前処理なしと従来の前処理つき CG 法の数値実験結果
Table 8　Numerical results of nonpreconditioned and traditional preconditioned CG methods for matrices BEAM，CABLE and BRIDGE.

| | BEAM | | | CABLE | | | BRIDGE | | |
|---|---|---|---|---|---|---|---|---|---|
| 前処理＋ CG | 反復回数 | 合計時間 | メモリ量 | 反復回数 | 合計時間 | メモリ量 | 反復回数 | 合計時間 | メモリ量 |
| 前処理なし | max | - | - | 7,022 | 137 | 27.8 | 17,915 | 1,962 | 158 |
| IC | max | - | - | max | - | - | max | - | - |
| 加速係数つき IC | 7,274 | 31.2 | 3.60 | 2,733 | 99.1 | 28.3 | 7,001 | 1,471 | 161 |

**表 9**　行列 BEAM に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の実験結果
Table 9　Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix BEAM.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/10 | 0.005 | 3 | 2,467 | 0.03 | 7.30 | 7.33 | 1.04 | 0.23 | 4.65 |
| | - | - | | 2,372 | 0.02 | 6.98 | 7.00 | 1.0 | 0.22 | 4.57 |
| 0.01 | 1/20 | 0.0005 | 2 | 812 | 0.09 | 2.98 | 3.08 | 0.61 | 0.09 | 6.11 |
| | - | - | | 1,370 | 0.07 | 4.95 | 5.02 | 1.0 | 0.16 | 5.91 |
| 0.005 | 1/100 | 0.00005 | 1 | 485 | 0.31 | 2.04 | 2.35 | 0.54 | 0.07 | 7.09 |
| | - | - | | 1,065 | 0.11 | 4.24 | 4.35 | 1.0 | 0.13 | 6.67 |
| 0.001 | 1/100 | 0.00001 | 1 | 210 | 0.45 | 1.30 | **1.75** | 0.46 | **0.05** | 10.4 |
| | - | - | | 607 | 0.33 | 3.43 | 3.76 | 1.0 | 0.12 | 9.46 |
| 0.0005 | 1/100 | 0.000005 | 1 | 174 | 0.78 | 1.30 | 2.08 | 0.55 | 0.06 | 12.6 |
| | - | - | | 480 | 0.54 | 3.19 | **3.73** | 1.0 | **0.11** | 11.2 |
| 0.0001 | 1/100 | 0.000001 | 1 | 119 | 2.76 | 1.33 | 4.09 | 0.89 | 0.13 | 20.5 |
| | - | - | | 276 | 1.83 | 2.75 | 4.58 | 1.0 | 0.14 | 18.0 |

**表 10**　行列 CABLE に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の実験結果
Table 10　Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix CABLE.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/50 | 0.001 | 2 | 2,117 | 0.34 | 53.7 | 54.1 | 0.77 | 0.54 | 38.2 |
| | - | - | - | 2,868 | 0.23 | 69.9 | 70.1 | 1.0 | 0.70 | 353 |
| 0.01 | 1/100 | 0.0001 | 1 | 591 | 1.42 | 21.0 | 22.4 | 0.47 | 0.22 | 57.2 |
| | - | - | - | 1,411 | 1.03 | 46.4 | 47.4 | 1.0 | 0.45 | 52.1 |
| 0.005 | 1/100 | 0.00005 | 1 | 314 | 2.54 | 13.5 | **16.1** | 0.37 | **0.16** | 69.9 |
| | - | - | - | 1,038 | 1.84 | 40.6 | 42.5 | 1.0 | 0.42 | 63.0 |
| 0.001 | 1/100 | 0.00001 | 1 | 158 | 10.6 | 10.6 | 21.2 | 0.54 | 0.21 | 118 |
| | - | - | - | 512 | 7.37 | 31.3 | **38.7** | 1.0 | **0.39** | 105 |
| 0.0005 | 1/100 | 0.000005 | 1 | 123 | 23.8 | 9.96 | 33.8 | 0.76 | 0.34 | 152 |
| | - | - | - | 390 | 15.8 | 28.6 | 44.4 | 1.0 | 0.44 | 134 |
| 0.0001 | 1/100 | 0.000001 | 1 | 73 | 131 | 9.39 | 140 | 1.18 | 1.41 | 273 |
| | - | - | - | 212 | 94.2 | 24.4 | 118 | 1.0 | 1.19 | 242 |

べての行列を通して，フィルインを考慮しない加速係数つき ICCG 法に比べて約 1.4〜11 倍多く必要となった．また，RICCG 法と比較すると 1.2 倍多くのメモリ量が必要となる．これは，対角項の値が緩和されることにより小さくなり，それにともなって計算される非対角項の値が大きくなるため，RIC 分解と対角緩和つき準 RIC 分解で同じ閾値を用いた場合，対角緩和つき準 RIC 分解で棄却される非対角項の数が少なくなるためである．

また，閾値，対角緩和係数と分解回数の関係について以下のことが分かる．

- 閾値が大きいときほど，分解回数が多くなる傾向がある．しかし閾値が大きいときの前処理時間は短いため，計算全体に対する影響の度合いは小さい．
- 閾値が大きいほど，分解が成功するときの倍率 $\rho$，すなわち対角項への修正量は大きくなる傾向がある．
- 同様に，閾値が大きいほど分解回数が多くなり，分解が成功するときの対角緩和係数 $\omega$ も大きく

**表 11** 行列 BRIDGE に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の実験結果
Table 11 Numerical results of Diagonally Relaxed Quasi RICCG and RICCG methods for matrix BRIDGE.

| tol | $\rho$ | $\omega$ | 分解回数 | 反復回数 | Pre 時間 | CG 時間 | 合計時間 | 時間比 1 | 時間比 2 | メモリ量 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 1/10 | 0.005 | 3 | 5,232 | 1.55 | 773 | 774 | 0.68 | 0.52 | 212 |
| | - | - | - | 8,002 | 1.32 | 1135 | 1137 | 1.0 | 0.77 | 204 |
| 0.01 | 1/20 | 0.0005 | 2 | 2,079 | 9.67 | 421 | 431 | 0.63 | 0.29 | 317 |
| | - | - | - | 3,573 | 5.35 | 679 | 684 | 1.0 | 0.46 | 298 |
| 0.005 | 1/100 | 0.00005 | 1 | 1,364 | 13.0 | 347 | 360 | 0.54 | 0.24 | 396 |
| | - | - | - | 2,846 | 9.50 | 649 | 659 | 1.0 | 0.44 | 360 |
| 0.001 | 1/100 | 0.00001 | 1 | 632 | 54.4 | 257 | **312** | 0.55 | **0.21** | 685 |
| | - | - | - | 1,476 | 37.0 | 529 | **566** | 1.0 | **0.38** | 604 |
| 0.0005 | 1/100 | 0.000005 | 1 | 469 | 129 | 227 | 356 | 0.62 | 0.24 | 890 |
| | - | - | - | 1,141 | 77.5 | 491 | 569 | 1.0 | 0.38 | 778 |
| 0.0001 | 1/100 | 0.000001 | 1 | 235 | 702 | 186 | 888 | 0.98 | 0.60 | 1642 |
| | - | - | - | 619 | 477 | 429 | 906 | 1.0 | 0.61 | 1420 |

なる．

### 7.3.2 問題 2 に対する実験結果

表 8 に 3 つの行列 BEAM，CABLE，BRIDGE に対する前処理なしと従来の前処理つき CG 法の数値実験の結果を示す．表 8 からフィルインを考慮しない ICCG 法は収束せず，特に，シェル要素で離散化したときの行列 BEAM は前処理なしの CG 法でも収束せず解き難い問題であることが分かる．それに対して，ソリッド要素で離散化したときの行列 CABLE は前処理なしの CG 法でも収束し，しかも反復回数も少なく比較的解きやすい問題であることが分かる．

**表 9**，**表 10**，**表 11** に行列 BEAM，CABLE，BRIDGE に対する**対角緩和つき準 RICCG 法**（上段）と RICCG 法（下段）の数値実験の結果を各々示す．表 9〜表 11 より，以下のことが分かる．

- 前処理なしの CG 法や加速係数つき ICCG 法では解き難い行列 BEAM に対して，RICCG 法と**対角緩和つき準 RICCG 法**は調べたすべての閾値において，加速係数つき ICCG 法に比べて合計時間が短い．特に，**対角緩和つき準 RICCG 法**が最も速い場合の合計時間は加速係数つき ICCG 法の合計時間に対する比率が 0.05（表 8 の “時間比 2” の欄）であり著しい改善効果が得られた．
- 加速係数つき ICCG 法でも比較的容易に解ける行列 CABLE に対して，**対角緩和つき準 RICCG 法**は閾値が 0.05 以外のすべての場合において，分解は最初の 1 回だけで済み，分解の再計算はする必要がなかった．また，加速係数つき ICCG 法の合計時間に対する比率も 0.16 で効率が良い．
- 行列 BRIDGE の場合，**対角緩和つき準 RICCG 法**は調べたすべての閾値において RICCG 法よりも合計時間が短くなり，加速係数つき ICCG 法の合計時間に対する比率も 0.21 で大きな効果が得られた．

## 8. ま と め

RICCG 法の収束性を大幅に向上させる**緩和係数 $\omega$ の自動決定を行う対角緩和つき準 RICCG 法**を提案した．この**対角緩和つき分解法**は，RIC 分解の閾値による棄却と対角項に対する修正量の緩和という 2 つの処理を組み合わせたものであり，その緩和係数 $\omega$ の自動決定の手順についても具体的に示した．そして，数値実験において提案した方法の有効性を実証した．

すなわち，問題 1 の疎行列データベースから選んだ 3 つの行列に対して，従来の加速係数つき ICCG 法に対する時間比で，**対角緩和つき準 RICCG 法**は最高で 10%（行列 S3DKT3M2 のとき）にまで短縮させた．また，元の RICCG 法に対する時間比でも，**対角緩和つき準 RICCG 法**は同様に 36%（行列 S3DKT3M2 のとき）に短縮させた．さらに，問題 2 の実際のコンクリート橋梁の応力解析で現れた行列に対しても，従来の加速係数つき ICCG 法に対する時間比で，**対角緩和つき準 RICCG 法**は同様にわずか 5%（行列 BEAM のとき）にまで短縮させた．同様に，元の RICCG 法に対する時間比でも，**対角緩和つき準 RICCG 法**は最高で 37%（行列 CABLE のとき）にまで短縮させた．このように，**緩和係数 $\omega$ の自動決定を行う対角緩和つき準 RICCG 法**は，実際問題の解決に有望な方法であると結論づけられる．

今後の課題は構造解析や電磁界解析以外の分野の問題に対して，**対角緩和つき準 RICCG 法**を適用し，最適な閾値の探索や対角緩和係数の自動決定の有効性を調べたい．

**謝辞** 論文を丁寧に読んでいただき，有益な助言を

いただいた匿名の査読者に深く感謝します．有限要素法に関する有用なご教示をいただいた（株）ホクトシステム原田義明氏に心より感謝します．多くの助言と協力を得た九州大学大学院の井上明彦氏，吉田正浩氏に感謝します．

## 参　考　文　献

1) Ajiz, M.A. and Jennings, A.: A robust incomplete Choleski-conjugate gradient algorithm, *Int. J. Numer. Methods Engrg.*, Vol.20, pp.949–966 (1984).

2) Dickinson, J. and Forsyth, P.: Preconditioned conjugate gradient methods for three dimensional linear elasticity, *Int. J. Numer. Methods Engrg.*, Vol.37, pp.2211–2234 (1994).

3) Hestenes, M. and Stiefel, E.: Method of Conjugate Gradient for Solving Linear Systems, *J. Res. Nat. Stand.*, Vol.49, pp.409–436 (1952).

4) Hladik, I., Reed, M.B. and Swoboda, G.: Robust preconditioners for linear elasticity FEM analyses, *Int.J. Numer.Methods Engrg.*, Vol.40, pp.2109–2127 (1997).

5) Jennings, A. and Malik, G.M.: Partial Elimination, *J. Inst. Maths. Applics.*, Vol.20, pp.307–316 (1977).

6) 柿原正伸，藤野清次：Ajiz-Jennings による不完全分解前処理の改良，情報処理学会 SWoPP 松江 2003 研究会報告，pp.25–30 (2003).

7) Kakihara, M. and Fujino, S.: An improvement of Ajiz-Jennings type of incomplete factorization preconditioning by means of post filtering, *INFORMATION*, Vol.7, pp.605–618 (2004).

8) 柿原正伸，藤野清次：収束の安定と高効率性を兼ね備えた対角緩和つき RICCG 法について，情報処理学会九州支部「火の国情報シンポジウム」予稿集 CD-ROM (2004.3).

9) Kaporin, I.E.: High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU+U^TR+R^TU$-decomposition, *Numer. Lin. Alg. Appl.*, Vol.5, pp.483–509 (1998).

10) Kouhia, R.: Sparse Matrices web page. http://www.hut.fi/~kouhia/sparse.html

11) Matrix Market web page. http://math.nist.gov/MatrixMarket/

12) 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, *Math. Comput.*, Vol.31, pp.148–162 (1977).

13) 三好俊郎，坂田信二，吉田有一郎，斉藤直人：計算力学と CAE シリーズ 14，スーパーコンピューティング，培風館 (2001).

14) 日本機械学会（編）：シェルの振動と座屈ハンドブック，技報堂出版 (2003).

15) PARASOL test data web page. http://www.parallab.uib.no/parasol/data.html

16) サボナディエル，J.C. ほか (著)，神谷紀生 (訳)：有限要素法を使った CAD，サイエンス社 (1991).

17) Tuff, A.D. and Jennings, A.: An iterative method for large systems of linear structural equations, *Int.J. Numer.Methods Engrg.*, Vol.7, pp.175–183 (1973).

18) University of Florida Sparse Matrix web page. http://www.cise.ufl.edu/research/sparse/matrices/

19) 有限要素法による構造解析システム，FEMLEEG ユーザガイド，ホクトシステム (2003).

(平成 16 年 7 月 2 日受付)
(平成 16 年 11 月 2 日採録)

**柿原　正伸**

1981 年生．2002 年 3 月九州大学工学部情報工学科卒業．九州大学大学院システム情報科学府修士課程在籍中．共役勾配法の不完全分解前処理に興味を持つ．

**藤野　清次**（正会員）

1950 年生．1974 年京都大学理学部卒業．1993 年博士（工学，東京大学）．2001 年九州大学情報基盤センター研究部教授．現在に至る．その間共役勾配法系統の反復法とその前処理の研究を行う．日本応用数理学会会員．