土木学会論文集 No. 437／I-17, pp. 173~182, 1991. 10

共役勾配法を基礎とする連立一次方程式解法の効率化に関する考察

吉田　裕*・中川昌弥**・田中知足***

本研究は，有限要素法で対象とすることとなる疎な大次元連立一次方程式の効率的な解法のアルゴリズムを構築することを目的として，CG法を基礎とする種々の連立一次方程式のアルゴリズムの収束性，要する計算時間など，その特性を比較検討したものである．

また，ICCG法における不完全コレスキー分解の過程で必要になる加速係数の最適な値を決める手だてを与えた．

Keywords：*conjugate gradient method, incomplete Choleski factorization, accelerating factor, finite element method, large sparse matrix*

1. はじめに

コンピュータの演算速度，容量など，その性能は3年程度の間に2~4倍の割合で指数関数的に進歩してきており，従来の理論的あるいは実験的手立てを補完する形で，コンピュータを利用した数値解析に基づいて現象を解明しようとする学問分野，すなわち計算力学が目覚ましく進歩し，工学諸分野にわたる汎用解析プログラムの開発，普及と相まって，産業界における日常の設計解析業務にも幅広くとり入れられ，広く計算工学と呼称されるに至っている．しかし，コンピュータの性能が向上すればする程解析対象モデルの精緻さに対する要求も高くなり，数万~数十万元といった大次元の非線形問題をスーパーコンピュータで何十時間もかけて解析することも稀なことではなくなってきており，解析法の効率化は現在でもなお重要な課題として残されている．

線形解析の場合はいうまでもなく，非線形解析の場合も結局は連立一次方程式を繰り返し解くことに帰着するのが普通であるから，解析の効率化の要となるのは連立一次方程式の解法である．有限要素法や差分法で対象とすることになる係数行列は通常零要素が大部分を占める疎な行列であり，内部記憶を有効に使うことが高速処理につながるから，できる限り少ない容量でいかに大次元の方程式をより速く解くことができるかが効率化の判断の鍵となる．

連立一次方程式の解法の算法のうち，共役勾配法（Conjugate Gradient Method＝CG法）[1)~7)]はマトリックスの乗算のみで解を求めることができるので，疎な大次元行列に対して，要する記憶容量の面で非常に有効であるが，係数行列の特性の違いによって収束の状況が大幅に異なり，安定した解析を展開するためには多くの問題点を抱えている．

本研究は，有限要素法で対象とすることになる疎な大次元連立一次方程式の効率的な解法のアルゴリズムを構築することを目的として，CG法を基礎とする種々の連立一次方程式の解法のアルゴリズムの収束性，要する計算時間，などその特性を比較検討した結果を報告するものである．

CG法は各反復計算ごとに残差ベクトルが直交するように解を修正しながら正しい解を探索していく解法で，理論的には自由度に相当する回数の反復計算を行なえば正しい解が得られることが保証されているものであるが，丸めの誤差による桁落ちの影響を大きく受けるので，方程式の係数のオーダーをそろえるためのスケーリングを施すだけでも収束性の大幅な改善につながる．CG法の適用に際しての収束に要する反復計算回数は問題によって大幅に異なるが，その収束性は係数マトリックスの条件数に依存するので[8)]，条件数によって適性をある程度予測することは可能である．しかし，固有値を求めるためにはそれなりの記憶容量と計算時間を必要とするので，その都度条件数を求めることはCG法を適用する意義に照らして実用上においては相容れるものではない．

ここでは，有限要素法による構造解析の対象となる一般的な問題の中で比較的CG法と相性の良いものと悪いものの代表例として，具体的な穴のあいた薄板の平面応力問題と平板曲げ問題とを設定し，対象の違いやスケーリングの仕方の違いが条件数にどのように関係し，そのことがCG法の収束性にどのように影響するかを確認する．

CG法の記憶容量の面の有効性を犠牲にすることなく収束性を改善する方法として，不完全なコレスキー分解

* 正会員　工博　東京工業大学教授　工学部土木工学科
（〒152　東京都目黒区大岡山2-12-1）
** 正会員　工修　JR東日本（研究当時東京工業大学大学院生）
*** 学生会員　東京工業大学大学院生

を介する共役勾配法（Incomplete Cholesky Conjugate Gradient Method＝ICCG法）がよく研究されている[9)~26)]．不完全なコレスキー分解の仕方にも種々の方法が採られるが，いずれの場合も手間をかけるだけのことはあり，収束性の大幅な改善につながるのが普通である．しかし，前進代入および後退代入計算の部分は計算結果に依存性が生ずるのでベクトル化することは難しく[27),28)]，ベクトル計算機を対象とする場合に，要する計算時間に対する実質的な効率化につながるかどうかは定かではない．

ここでは，不完全コレスキー分解を介することがどのように収束性の改善につながるかを検証し，同時に通常ICCG法の加速係数として理解されているパラメータの意義およびその振舞いについて新たな知見を与える．さらに，係数マトリックスの格納の仕方や計算過程を明確にした上で，ICCG法のアルゴリズムにおけるマトリックスの積や前進代入，後退代入，などの各計算部分に要する計算時間を比較検討し，特にベクトル計算機を念頭においてCG法の効率化の可能性を議論する．

2. 本研究で採用したアルゴリズムの概要

（1） CG法およびICCG法の理論

CG法は，解くべき連立一次方程式を$Ax=b$とし，係数マトリックスAが正定値対称行列の場合に，以下のような手順で示すことができる．

1) $r_0=b-Ax_0$, $p_0=r_0$
　ここに，x_0は任意の初期ベクトル
2) $\alpha_k=\dfrac{(p_k, r_k)}{(p_k, Ap_k)}$
3) $x_{k+1}=x_k+\alpha_k p_k$
4) $r_{k+1}=r_k-\alpha_k Ap_k$
5) $\beta_k=-\dfrac{(r_{k+1}, Ap_k)}{(p_k, Ap_k)}$
6) $p_{k+1}=r_{k+1}+\beta_k p_k$
7) 2）へ

……(1·a~f)

ICCG法は，与えられた係数マトリックスの不完全なコレスキー分解

$$A=\tilde{U}^T\tilde{D}\tilde{U}+R \quad \cdots\cdots(2)$$

の$\tilde{U}^T\tilde{D}\tilde{U}$を用いて，上記CG法のアルゴリズム中のステップ1），5），6）を以下のように書き換えて収束性を改善しようとするものであり，このことに伴って反復計算の度に前進代入，後退代入計算が必要になる．

1′) $r_0=b-Ax_0$, $p_0=(\tilde{U}^T\tilde{D}\tilde{U})^{-1}r_0$
5′) $\beta_k=-\dfrac{((\tilde{U}^T\tilde{D}\tilde{U})^{-1}r_{k+1}, Ap_k)}{(p_k, Ap_k)}$
6′) $p_{k+1}=(\tilde{U}^T\tilde{D}\tilde{U})^{-1}r_{k+1}+\beta_k p_k$

……(3·a~c)

（2） スケーリングの方法

ここでは，係数マトリックスの対称性を崩さないスケーリングの方法として，下記の2通りの方法を採用した．すなわち，係数マトリックスを，Aの対角要素からなる対角マトリックスをDとして下記のように置き換え，与えられた方程式$Ax=b$を$\tilde{A}\tilde{x}=\tilde{b}$に変換して解く方法がその1つである．また，節点の自由度を単位とする対角上に並ぶ部分マトリックスよりなる対角ブロックをDとして変換する方法が2番目の方法である．

$$\left.\begin{aligned}\tilde{A}&=D^{-1/2}AD^{-1/2}\\ \tilde{x}&=D^{1/2}x\\ \tilde{b}&=D^{-1/2}b\end{aligned}\right\} \quad \cdots\cdots(4·a~c)$$

（3） ICCG法における不完全コレスキー分解の方法

CG法では反復計算の都度更新されるベクトルと与えられた係数マトリックスとの積の計算が必要であり，係数マトリックスは元の値のまま確保しておかなければならない．ICCG法では，さらに不完全にコレスキー分解された三角マトリックスを記憶するための容量が必要になる．記憶容量が少なくてすむことがCG法の最大の利点であることを考えれば，不完全にコレスキー分解する際に書き換える係数マトリックスの要素を極力少なくすることが望ましい．通常用いられている方法は，非対角要素に関しては完全に元の値のままに保存して，対角要素だけを書き換えるものである．すなわち，書き換えられた対角マトリックスと元の係数マトリックスの上半分で構成される三角マトリックスを不完全コレスキー分解として用いる方法である．

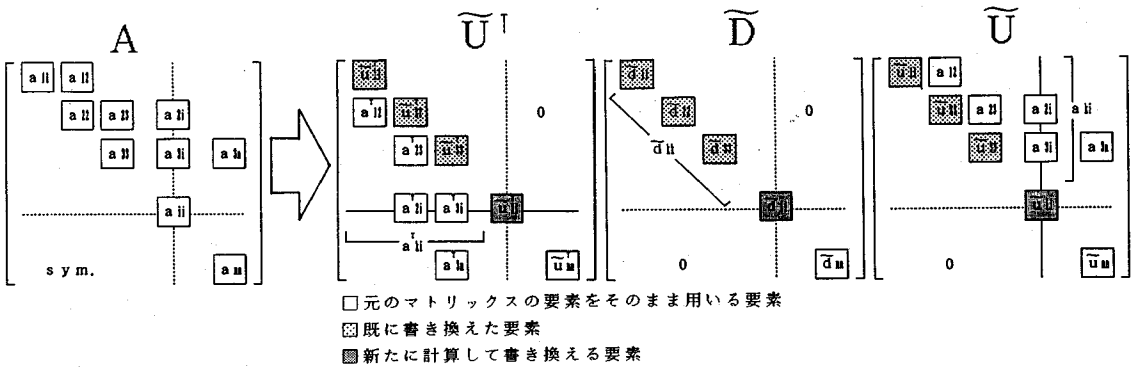
ここでは，対角要素でスケーリングを施した係数マトリックスに対して，対角要素のみを書き換える上述の方法と，節点の自由度に対応する対角上の部分マトリックスで構成される対角ブロックを下記の手順で書き換える方法，の2通りの方法を採用した．図—1に，不完全コレスキー分解の仕方を図式的に示した．

$$\left.\begin{aligned}\tilde{u}_{11}&=a_{11}\\ \tilde{u}_{ii}&=a_{ii}\\ \tilde{u}_{ii}&=(\text{const.})\times a_{ii}-\sum_{k=1}^{i-1}\tilde{u}_{ki}^T\tilde{d}_{kk}\tilde{u}_{ki}\\ \tilde{d}_{ii}&=\tilde{u}_{ii}^{-1}\end{aligned}\right\} \quad \cdots\cdots(5·a~d)$$

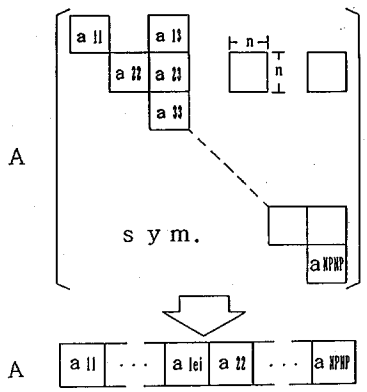
上記の手順において導入されるconst.は加速係数と呼ばれ，通常の場合は1.0~2.0程度の値が採られるが，その最適値は場合によって異なり，一般的には経験によって決められるものとされている．したがって，ICGG法を有効に使いこなすためには経験が必要であると認識されている一面がある[29)]．

（4） 係数マトリックスの格納方法

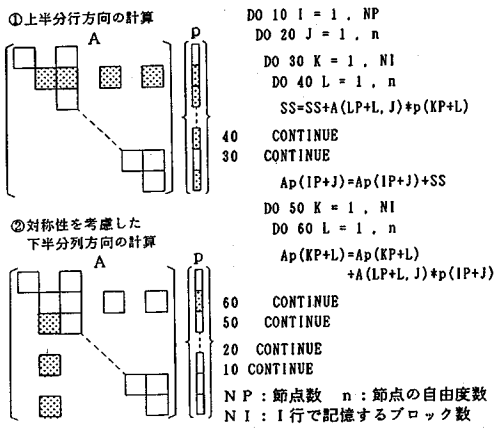
係数マトリックスは，対称性を考慮してその上半分だけを記憶する．その際，節点の自由度の大きさの部分マトリックスを単位ブロックとして，非零ブロックだけを図—2に示すように行方向に一次元的に記憶する．この



図—1 不完全コレスキー分解の仕方



図—2 係数マトリックスの記憶方法



図—3 積 Ap の計算手順の概要

ときに，各行ごとに記憶したブロックの数と，それぞれのブロックが対応する列番号を index として確保しておく必要がある．

(5) CG 法における積 Ap の計算方法

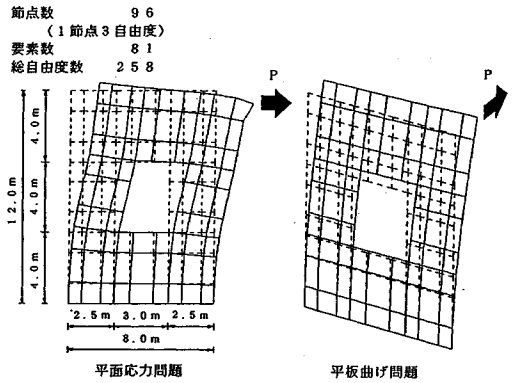
係数マトリックスはその上半分だけを記憶するので，CG 法の計算手順における係数マトリックス A と修正ベクトル p の積 Ap の計算は，係数マトリックスの上半分と下半分に対応する2つの計算部分に分けて行う必要がある．計算手順を図—3に示す．係数マトリックス上半分の第 i 行の要素と修正ベクトル p との積を図中30のループで計算する．一方，第 i 行の要素は同時に係数マトリックスの下半分の第 i 列に対応するから，これらに対応する積を図中50のループで求める．このとき最内側のループ長は各ブロックの大きさ，すなわち節点の自由度の数となる．

(6) ICCG 法における前進代入および後退代入計算

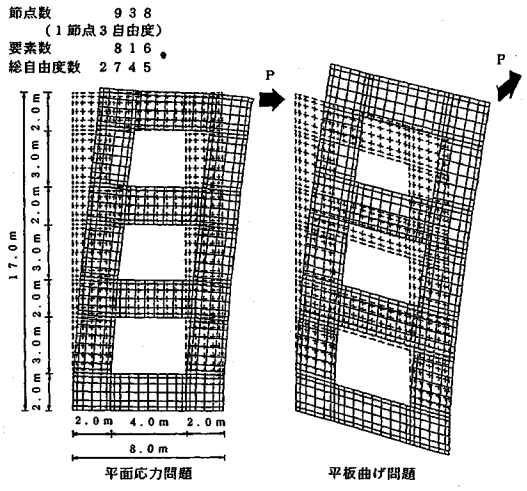
ICCG 法の計算手順のうち，式 (3·a~c) における $(\tilde{U}^T\tilde{D}\tilde{U})^{-1}r$ の計算は，実際には $(\tilde{U}^T\tilde{D}\tilde{U})z=r$ を対象として前進代入および後退代入計算によって z を求める過程で実行する．

3. 解析対象の設定

CG 法の収束性は，対象とする問題の種類によって大幅に異なる．ここではこのことを踏まえ，有限要素法による構造解析の対象となる一般的な問題の中で比較的 CG 法と相性の良いものと悪いものの代表例として平面応力問題と平板曲げ問題を取り上げ，より現実的な問題となるように穴を有する構造として解析対象を設定する．CG 法の収束性は係数マトリックスの条件数に依存するが，問題の違いやスケーリングの仕方の違いが固有値の分布や条件数にどのように影響し，そのことが CG 法の収束性とどのように関係するかを確認することを目的として，図—4に示すような解析対象 (1) を設定した．大次元の係数マトリックスの固有値を求めることはそう容易ではなく，例えばハウスホルダー法で係数マトリックスを三重対角化し，二分法で固有値を求める場合，三重対角化のためにマトリックスの右上半分または左下半分に相当する記憶容量と，自由度 n に対して $2n^3/3$ 回の乗算と $n-2$ 回の平方根の計算が必要となる[30]ために，比較的自由度の小さな問題を設定したものである．CG 法の収束性を議論するために図—5に示すような解析対象 (2) を設定した．



図—4 解析対象（1）



図—5 解析対象（2）

採用した要素は応力仮定のハイブリッド法に基づいて誘導した，面内回転変位を節点変位として有する1節点3自由度の平面応力要素と，1節点3自由度の平板曲げ要素である．節点変位の数が同じであるから，平面的に同じ要素分割を用いる場合には平面応力問題と平板曲げ問題の総自由度数は同じになる．

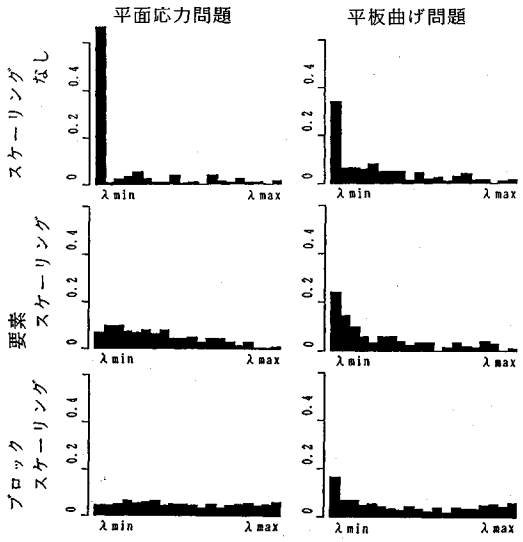
いずれの問題も諸元は，図中に示したとおりであり，解析対象（1）の総自由度数は258，解析対象（2）の総自由度数は2745である．

4. CG法の収束性の実際

方程式の係数のオーダーをそろえるためのスケーリングを施すだけでも固有値の分布は大きく変わり，収束性の大幅な改善につながることが知られている．ここでは，対象とする問題の種類の違いやスケーリングの方法の違いが，係数マトリックスの条件数や固有値の分布にどのように影響するかを確認し，そのことがCG法の収束性とどのように関係するかを具体的な計算結果を示して確認する．

表—1 各問題の条件数と収束演算回数の比較

		平面応力問題	平板曲げ問題
スケーリングなし	最大固有値	2.06014×10^{11}	9.53806×10^{11}
	最小固有値	3.63216×10^{4}	1.12968×10^{3}
	条件数	5.67194×10^{6}	8.44315×10^{8}
	収束回数	789	2188
要素スケーリング	最大固有値	2.90635	3.70397
	最小固有値	4.30193×10^{-4}	3.93337×10^{-6}
	条件数	6.75592×10^{3}	9.41678×10^{5}
	収束回数	108	383
ブロックスケーリング	最大固有値	2.03152	2.35775
	最小固有値	4.33487×10^{-4}	3.98841×10^{-6}
	条件数	4.68646×10^{3}	5.91150×10^{5}
	収束回数	95	296

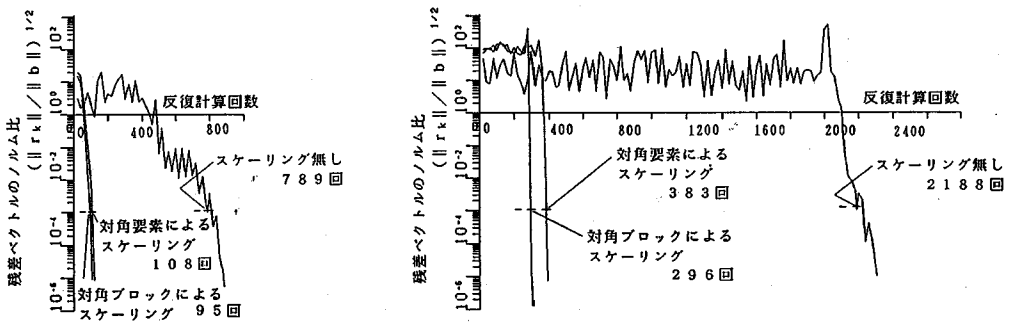


図—6 係数マトリックスの固有値の分布

解析対象は3.で示した解析対象（1）の平面応力問題および平板曲げ問題である．解析は，係数マトリックスをそのまま用いる場合，および2.(2)に示した，係数マトリックスに対角要素または対角ブロックによるスケーリングを施す場合の3通りの場合を比較して行った．

(1) スケーリングの仕方と固有値の分布および条件数

設定した問題のそれぞれに対して，スケーリングの有無および方法の違いによる固有値および固有値の分布の



（a） 平面応力問題　　（b） 平板曲げ問題

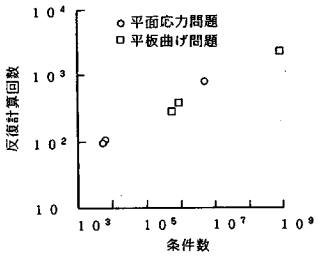
図—7 CG 法の収束状況

実際を表—1 および図—6 に示す．表—1 は固有値および条件数（最大固有値と最小固有値の比）などの値を具体的に表示したものであり，図—6 は最大固有値と最小固有値の間を 20 の領域に等分割して各領域に含まれる固有値の数を棒グラフで示して，固有値の分布特性を明らかにしたものである．スケーリングの仕方が同じ場合，平板曲げ問題の条件数は平面応力問題に比べて約 2 桁大きな値となっていることがわかる．また，スケーリングを施すことにより固有値の分布は平均化し，条件数が約 3 桁減少していることがわかる．対角ブロックでスケーリングを施す場合は対角要素による場合に比べて固有値の分布がより一様になり，条件数も小さくなるために収束性も若干改善されることがわかる．

（2） CG 法の収束特性

設定したそれぞれの条件のもとでの CG 法の収束状況の変化を図—7 に示す．収束に要する繰り返し演算回数は問題により大幅に異なり，平板曲げ問題では平面応力問題の約 3 倍の演算回数を要している．また，スケーリングを施すことにより収束演算回数は約 1/6～1/8 に改善されることがわかる．

係数マトリックスの条件数と CG 法の収束性の関係を，縦軸に収束に要する演算回数，横軸に条件数をそれぞれ対数でとってプロットしたものが図—8 である．通常言われているように，条件数と CG 法の収束性との間には相当な相関関係があることがわかる．



図—8 条件数と収束性の関係

5. ICCG 法における加速係数について

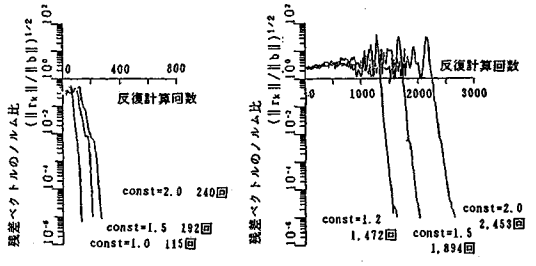
ICCG 法は，係数マトリックスの不完全コレスキー分解を介して CG 法の収束性を速めようとするものであるが，不完全コレスキー分解を行う際に通常加速係数と呼ばれている値を決めることが必要になる．この値のとり方によって収束に要する繰り返し計算回数は大きく影響を受けるが，この値の決め方に決定的な方法はなく，ICCG 法を採用する際の難しさはこの値の決め方にある，と言っても過言ではない．

ここでは，加速係数の採り方が，ICCG 法の収束性に及ぼす影響について具体的な数値計算例を基に種々検討した結果をまとめ，加速係数の役割およびその最適な値について考察する．

（1） 加速係数の値と収束性

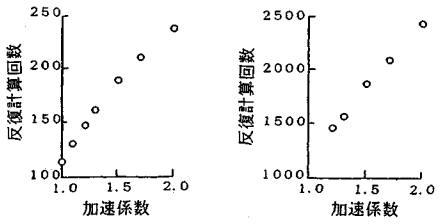
自由度の大きな解析対象（2）の平面応力問題および平板曲げ問題に対角ブロックを書き換える不完全コレスキー分解を用いた ICCG 法を適用し，加速係数の値を種々変化させて計算を行った．

加速係数の値によって収束状況がどのように変わるかを図—9 および図—10 に示す．平板曲げ問題で加速係数を 1.0 または 1.1 とした場合には 7 000 回に及ぶ繰り返し計算を続けても収束する様子がみられなかった．このような結果に至る要因を探るために演算過程を検討したところ，不完全なコレスキー分解を用いて対角ブロックを書き換えている過程で対角要素の値が負となることが分かった．平面応力問題では加速係数の値を 1.0 とした場合でも対角要素の値が負とはならず，加速係数の値が 1.0 の場合に最も収束性が早く，この値より大きくなればなる程収束性が悪くなっている．平板曲げ問題では対角要素が負となる加速係数の値の範囲があり，加速係数の値がこの領域にある場合には収束に至らず，不完全なコレスキー分解に基づく書換えの過程で対角要素が負とならない範囲で 1.0 に最も近い加速係数の値 1.2 の場合に最も収束性が良く，この値より大きな値をとればとる



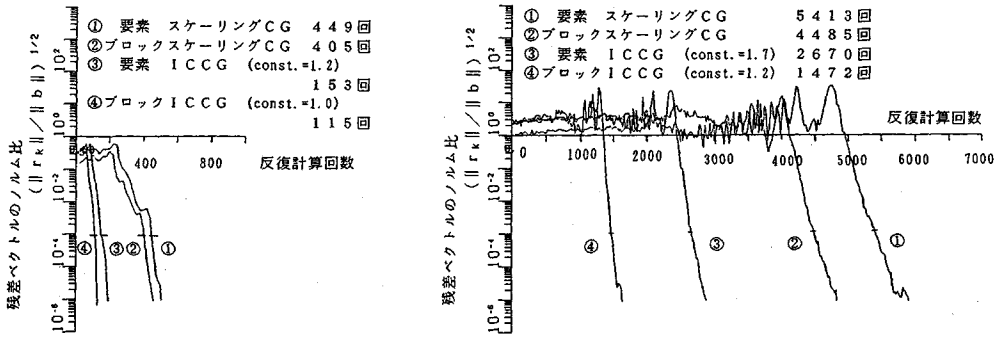
（a） 平面応力問題　　（b） 平板曲げ問題

図—9 加速係数による収束状況の違い



（a） 平面応力問題　　（b） 平板曲げ問題

図—10 加速係数と反復計算回数の関係



（a） 平面応力問題　　（b） 平板曲げ問題

図—11 前処理の方法が収束性に及ぼす影響

程収束性が悪化していることが分かる．

結論として，ICCG 法では不完全なコレスキー分解を用いているために，コレスキー分解の過程で係数マトリックスの対角上の要素の値が負になる場合が生ずるが，加速係数はこれが負にならないように対角上の要素の値を嵩上げする意味合いがあり，加速係数の値は対角要素が負にならない範囲で 1.0 に近い値をとるほど収束性が良いと言える．

（2） 最適な加速係数の求め方について

ICCG 法の収束性に最適な加速係数の値は，前述の通りであり，具体的にこの値を求める簡素な方法を編み出すことが望ましいが，ここでは最も単純な方法として，加速係数の値を 1.0 から 0.1 刻みなどで漸増させ，不完全コレスキー分解の書換えの過程で得られる対角要素の値の符号を判定しながら不完全コレスキー分解を繰り返し，対角要素が負とならない最も 1.0 に近い加速係数の値を求める方法をとった．

この方法はあまりにもまともな手立てであり，相当な計算時間がかかるように見えるが，実際にこの過程に要した計算時間は平板曲げ問題の加速係数 1.2 の場合で収束過程の 1 回の反復計算に相当する計算時間と同程度であった．仮に最適な加速係数が 2.0 の場合でもこの方法で不完全コレスキー分解に要する計算時間は反復収束計算の数回分程度であり，加速係数の違いが収束回数に及ぼす影響に比べるとほとんど問題にならない程度の計算時間であることがわかる．

6. ICCG 法による収束性の改善

ICCG 法における不完全なコレスキー分解の方法としては種々の方法が採られるが，いずれの場合も収束性の大幅な改善につながるのが普通である．ここでは，不完全コレスキー分解を介することがどのように収束性の改善につながり，不完全コレスキー分解の仕方が収束性にどのように影響するかを具体的な計算結果を示して確認する．

解析対象は自由度の大きな解析対象（2）の平面応力問題および平板曲げ問題である．不完全コレスキー分解の仕方としては，対角要素または対角ブロックを書き換える方法を対象とし，加速係数の値は 0.1 刻みで対角要素が負にならない領域の 1.0 に最も近い値とする．また，対角要素または対角ブロックによってスケーリングを施しただけの CG 法による結果も比較して示す．

収束状況を図—11 に示す．採られた加速係数の値は図中に示した通りである．いずれの問題においても，不完全コレスキー分解を介することにより収束に要する繰り返し演算回数は約 1/2～1/3 に改善され，不完全コレスキー分解の仕方としては対角要素だけを書き換える方法よりも対角ブロックを書き換える方法の方がより収束

表—2 各解法で要する計算時間の比較

	解法の種類	記憶容量	データ入力	剛性行列	前処理過程(s) SCALING	前処理過程(s) コレスキー	反復過程(s) 回数	Ap1回	前後代入	反復1回	全過程	応力反力	全計算過程(s)
平面応力問題	SCG	447KB	1.83	4.22	0.073	——	449	0.0854	——	0.0859	38.569	8.51	53.40
	BSCG	491KB	1.84	4.26	1.591	——	405	0.0886	——	0.0891	36.096	10.10	54.07
	IC	491KB	1.84	4.20	0.073	0.035	153	0.0835	0.0963	0.1810	27.584	8.61	42.53
	BIC	535KB	1.84	4.20	0.073	0.109	115	0.0833	0.1032	0.1872	21.487	8.60	36.58
	SKY	1.44MB	1.84	4.43	——	1.100	—	——	0.0290	——	——	5.60	13.1
平板曲げ問題	SCG	447KB	1.83	2.89	0.073	——	5413	0.0854	——	0.0859	464.954	5.84	475.788
	BSCG	491KB	1.85	2.92	1.549	——	4485	0.0886	——	0.0891	399.697	7.36	413.577
	IC	491KB	1.84	2.87	0.073	0.041	2670	0.0835	0.0968	0.1810	482.576	5.95	493.563
	BIC	535KB	1.84	2.88	0.073	0.185	1472	0.0833	0.1040	0.1879	276.071	5.94	287.215
	SKY	1.44MB	1.84	3.10	——	1.100	—	——	0.0290	——	——	4.27	10.4

SCG ：対角要素によるスケーリングを施したCG法
BSCG：対角ブロックによるスケーリングを施したCG法
IC ：対角要素を書き換えるICCG法
BIC ：対角ブロックを書き換えるICCG法
SKY ：スカイライン法とコレスキー分解

性の改善に効果があることがわかる．

7. ICCG法に要する計算時間について

ICCG法において不完全コレスキー分解を介することによって収束性がどのように改善されるかを比較検討し，収束に要する繰り返し演算回数に対してICCG法が相当に有効であることを確認した．ICCG法では各反復計算ごとに不完全コレスキー分解に基づく解を求めるための前進代入および後退代入計算を行い，手間をかけるのであるから，ある程度の改善は当然のことである．

ここでは，ICCG法を適用するために必要とする付加的な計算時間に注目して，より現実的な評価基準としての計算時間に対するICCG法の有効性を具体的な数値例に基づいて議論する．

(1) ICCG法に要する計算時間の比較

解析対象（2）の平面応力問題と平板曲げ問題を対象として，対角要素または対角ブロックによるスケーリングを施しただけのCG法と，対角要素または対角ブロックを書き換える不完全コレスキー分解を用いたICCG法の4通りの方法を適用して計算を行った．使用した計算機は東京工業大学総合情報処理センターのスーパーコンピュータETA 10-Eである．

それぞれの方法に対応する収束演算回数，各計算過程に要した計算時間，ソルバー部で必要とする記憶容量を表—2にまとめて示した．スカイライン法とICCG法におけるコレスキー分解および前進代入・後退代入計算に要する計算時間の違いは，フィルイン部が対象となるかならないかの違いの他に，スカイライン法においては三角マトリックスの対角上の値が1となるようにコレスキー分解することが可能であるのに対し，ICCG法では三角マトリックスの非対角要素に元の係数をそのまま用いる関係で，三角マトリックスの対角上の値が1となるようにコレスキー分解することができないことによるものである．

ICCG法では反復過程ごとに1回の前進代入計算および後退代入計算を必要とするために，ICCG法の反復計算1回当りに要する計算時間はCG法の2倍以上となっている．このため，ICCG法では反復計算回数の改善ほどには計算時間の効率化につながっていないことがわかる．例えば，平板曲げ問題における対角要素だけを書き換える不完全コレスキー分解を用いたICCG法では最適な加速係数の値が1.7と大きいために収束性の改善の効果がそれほど大きくないことも関係し，スケーリングを施すだけのCG法よりも全計算過程においてむしろ多くの計算時間を要していることがわかる．

(2) ICCG法の効率化の可能性について

CG法の効率化をはかったICCG法においてもなお，直接法の1つであるスカイライン法と比較して相当多くの計算時間を要するので，CG法の実用化のためには更に何らかの計算時間の改善の方策を考えることが重要な課題である．

計算時間を支配する要因には種々の事項が考えられるが，プログラムの書き方にも相当依存する．例えばループ長が定数で短いDOループについてDOループを用いるか，またはインライン展開で処理するかといったプログラミングの仕方の違いには通常はそれ程の注意が払われていない．特にスカラー計算機を用いる場合には全くと言ってよい程，その違いを意識していないのが普通である．

表—3は，対角要素によるスケーリングを施すCG法の反復過程のうち，係数マトリックスと修正ベクトルの積 Ap の節点の自由度に関するループに，図—12に示すようなDOループを用いる場合とインライン展開を用いる場合に要した計算時間を比較して示したものである．対象とした問題は，解析対象（2）の平面応力問題であり，用いた計算機は東京工業大学総合情報処理センターのスーパーコンピュータETA-10Eとスカラー計算機HITAC M-660Kであり，特にETA-10の場合にはベクトル化演算とあえてスカラー演算を実行した場合の2通りの結果を示した．計算の内容が対称性を考慮し

表—3　DO ループとインライン展開で要する計算時間の比較

計算機	積 Ap 1回 (s)	
	DO ループ	インライン展開
ETA-10 (V)	0.0854	0.0604
ETA-10 (S)	0.0544	0.0375
M-660 K	0.1182	0.0805

(V)：ベクトル化演算　(S)：スカラ演算

```
      DO 20 K = 1 , N1
        DO 20 L = 1 , 3
          SS=SS+A(LP+L, J)*p(KP+L)
   20 CONTINUE
```

```
      DO 20 K = 1 , N1
          SS=SS+A(LP+1, J)*p(KP+1)
               +A(LP+2, J)*p(KP+2)
               +A(LP+3, J)*p(KP+3)
   20 CONTINUE
```

(係数マトリックス上半分に対応するループ)

```
      DO 30 K = 1 , N1
        DO 30 L = 1 , 3

          Ap(KP+L)=Ap(KP+L)
                  +A(LP+L, J)*p(IP+J)

   30 CONTINUE
```

```
      DO 30 K = 1 , N1
          Ap(KP+1)=Ap(KP+1)
                   +A(LP+1, J)*p(IP+J)
          Ap(KP+2)=Ap(KP+2)
                   +A(LP+2, J)*p(IP+J)
          Ap(KP+3)=Ap(KP+3)
                   +A(LP+3, J)*p(IP+J)
   30 CONTINUE
```

(係数マトリックス下半分に対応するループ)

(a) DO ループを用いた場合　(b) インライン展開した場合

図—12　内側の DO ループのインライン展開

て係数マトリックスの上半分だけを記憶した場合の乗算で，ベクトル計算に適していないためではあるが，計算機の種類やベクトル化演算かスカラー演算かの違いによらず，いずれの場合も，インライン展開を用いた場合の方が DO ループを用いた場合より相当計算時間が短縮されている．これは DO ループの判断に要する時間のためであると考えられる．特に M-660 K においては計算時間が約 1/2 に減少しており，DO ループ一つをはずしただけでこれ程の計算時間の短縮につながるという事実は，効率化を考えるに当たっての思いがけない示唆を与えている．

科学計算を考える場合，スーパーコンピュータの特徴を念頭に置くことは重要である．スーパーコンピュータの特殊性は大きく分けてベクトル化と並列化の 2 つである．従って，CG 法の効率化を考える場合に，ベクトル化およびベクトル長に関する議論は重要である．CG 法の反復計算過程の中で積 Ap 以外の計算部分は，主に総自由度数を次元にもつベクトルの内積計算 5 回とベクトルの和 2 回からなり，いずれも最内側のループ長は総自由度数である．反復計算過程を積 Ap とそれ以外の計算部分に分け，それぞれに要した計算時間をまとめて示したものが表—4 である．反復計算の中の積 Ap 以外の計算部分 1 回に要する計算時間について ETA-10 のベクトル化演算の場合とスカラー演算の場合を比較すると，ベクトル化演算により計算時間は約 1/18 に短縮されていることがわかる．このことは，ベクトル長の長い問題におけるベクトル計算機の有効性が相当大きいことを示

表—4　ベクトル化計算とスカラ計算で要する計算時間の比較

	計算機	回数	反復計算 1 回			全反復 (s)
			Ap 1回 (s)	他 1 回 (s)	反復 1 回 (s)	
インライン DO	ETA-10 (V)	449	0.0854	0.0005	0.0859	38.570
	ETA-10 (S)	450	0.0544	0.0092	0.0636	28.641
	M-660 K	449	0.1182	0.0150	0.1332	59.822

(V)：ベクトル化演算　(S)：スカラ演算

している．

積 Ap の計算部分でベクトル化計算の効果が発揮されていないのは，節点の自由度に関する DO ループを用いた場合の最内側のループ長は 3 であり，またインライン展開を用いた場合の新しい最内側のループ長はその節点に関係する節点のうちその節点よりも節点番号の小さくないものの数，すなわち 1 以上 5 以下の値であり，いずれの場合もベクトル長が短いためである．

また，積 Ap の計算 1 回に要する計算時間についてベクトル化演算の場合とスカラ演算の場合とを比較すると，ベクトル化計算によって逆に約 1.6 倍の計算時間を要する結果となっており，ベクトル化計算に不向きなプログラムをベクトル計算機で実行することはむしろ不利になる場合があることがわかる．このことは，コンパイラーのベクトル化の水準，などにもよるが，用いる計算機の特性とプログラムの内容との関係が効率化を考えるに当たっての重要な要因となることを示している．

積 Ap の計算はマトリックスとベクトルの積であるからデータに依存関係がなく，ベクトル長の長いベクトル計算機向きのアルゴリズムに書き換える方策を考えることは容易であり，積 Ap の計算に要する計算時間をベクトル計算機によって大幅に改善することができる可能性があると考える．一方，ICCG 法では更に前進代入および後退代入計算が必要となるが，これらの計算部分にはデータに依存関係が生ずるので，ベクトル化計算向きのアルゴリズムを開発することは難しいように思われる．

8. おわりに

以上，有限要素法で対象とすることとなる疎な大次元連立一次方程式の効率的な解法のアルゴリズムを構築することを目的として，CG 法を基礎とする種々の連立一次方程式のアルゴリズムの収束性，要する計算時間，などその特性を比較検討し，以下の結果を得た．

CG 法の収束性は係数マトリックスの条件数に依存することが知られている．ここでは数値例により，係数マトリックスの条件数が問題の種類やスケーリングの有無，方法により大きく変わり，その結果 CG 法の収束性も大幅に異なってくることを改めて確認すると同時に，係数マトリックスの固有値の分布特性がスケーリングを施すことにより一様分布に近づくこと，特に対角ブロックによるスケーリングを施した場合この傾向が顕著になり，収束性も改善されることを確認した．

不完全コレスキー分解を介するICCG法では，スケーリングを施すだけのCG法と比較して大幅な収束性の改善がみられるが，不完全コレスキー分解の過程で必要になる加速係数の値の決め方に決定的な方法はなく，経験に基づいて決めるものと認識されている一面がある．ここでは，加速係数が不完全コレスキー分解の過程で対角要素の値が負にならないようにその値を嵩上げする意味合いがあり，加速係数の値は対角要素の値が負にならない範囲で1.0に近いほど収束性がよいことを明らかにした．このような条件を満たす加速係数の値を求める具体的な方法として，加速係数の値を1.0から0.1刻みで漸増させ，不完全コレスキー分解の書換え過程で得られる対角要素の値の符号を判定しながら不完全コレスキー分解を繰り返し，対角要素が負とならない最も1.0に近い加速係数の値を求める方法をとったが，このような単純で，手間がかかる方法の場合でも，実際に要する計算時間は数回の反復計算に相当する程度であり，加速係数の値のわずかな差が収束性に大きく影響することを考えると，このような基礎的な方法によってでも最適な加速係数を求めることの効果は大きく，そのための手立てを与えた意義は大きいと考えている．

ICCG法ではその収束性は大幅に改善されるが，反復計算ごとに前進代入および後退代入計算を必要とするために，要する計算時間に関しては収束回数に見られるほどの大幅な改善にはつながらないことを確認した．特にベクトル計算を前提とした効率化を考える場合には，ベクトル化率を高くし，ベクトル長を長くとることが重要になるが，前進代入および後退代入計算の部分はデータに依存関係が存在するためにベクトル長を長くする方策を考えることは難しく，スーパーコンピュータを対象として効率化を計る場合にはICCG法がスケーリングを施すだけのCG法に比べて有効であるかどうかは定かではない．

参考文献

1) Hestenes, M.R. and Stiefel, E. : Methods of conjugate gradients for solving linear systemes, J. Res. Nat. Bur. Stand., Vol.49, No.6, pp.409~436, Dec. 1952.
2) Jennings, A. and Malik, G.M. : The solution of sparse linear equations by the conjugate gradient method, Int. j. Numer. Methods Eng., Vol.12, pp.141~158, 1978.
3) Saad, Y. and Schultz, M.H. : Conjugate gradient-like algorithms for solving nonsymmetric linear systems, Math. Comp., Vol.44, No.170, pp.417~424, Apr. 1985.
4) Johnsson, S.L. and Mathur, K.K. : Experience with the conjugate gradient method for stress analysis on a data parallel supercomputer, Int. J. Numer. Methods Eng., Vol.27, pp.523~546, 1989.
5) Ralston, A. and Wilf, H.S. : Mathematical Methods for Digital Computers, pp.62~72, John Wiley & Sons, Inc., 1967.
6) 戸川隼人：マトリックスの数値計算，pp.95~102，オーム社，1971.
7) 戸川隼人：共役勾配法 シリーズ新しい応用の数学17，教育出版，1977.
8) Jennings, A. : Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method, J. Inst. Maths. Applics., Vol.20, pp.61~72, 1977.
9) 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. Comp., Vol.31, No.137, pp.148~162, Jan. 1977.
10) Kershaw, D.S. : The incomplete Cholesky-conjugate gradient method for the iterative solution of systemes of linear equations, J. Comp. Phys., Vol.26, pp.43~65, 1978.
11) Gustafsson, I. : A class of first order factorization methoda, bit, Vol.18, pp.142~156, 1978.
12) Van der Vorst, H.A. : Iterative solution methods for certain sparse linear systems with a non-symmetric matrix arising from PDE-ploblems, J. Comp. Phys., Vol.44, pp.1~19, 1981.
13) Meijerink, J.A. and van der Vorst, H.A. : Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems, J. Comp. Phys., Vol.44, pp.134~155, 1981.
14) Ajiz, M.A. and Jennings, A. : A robust incomplete Choleski-conjugate gradient algorithm, Int. J. Numer. Methods Eng., Vol.20, pp.949~966, 1984.
15) Axelsson, O. and Brinkkemper, S. : On some versions of incomplete block-matrix factorization iterative methods, Lin. Alg. Appl., Vol.58, pp.3~15, 1984.
16) Concus, P., Golub, G.H. and Meurant, G. : Block preconditioning for the conjugate gradient method, SIAM J. Sci. stat. Comput., Vol.6, No.1, Jan. 1985.
17) Axelsson, O. : A general incomplete block-matrix factorization method, Lin. Alg. Appl., Vol.74, pp.179~190, 1986.
18) Axelesson, O. and Polman, B. : On approximate factorization methods, for block matrices suitable for vector and parallel processors, Lin. Alg. Appl., Vol.77, pp.3~26, 1986.
19) 藤掛・村野：不完全コレスキ分解共役勾配法とスカイライン法との連立一次方程式解法効率の比較，構造工学における数値解析法シンポジウム論文集，Vol.10，pp.28~33，1986.
20) 三好・高野：構造解析における反復解法について—ICCG法とSCGの比較—，構造工学における数値解析法シンポジウム論文集，Vol.12，pp.19~22，1988.
21) Vassilevski, P.S. : Algorithms for construction of preconditioners based on incomplete block-factorizations of the matrix, Int. J. Numer. Methods Eng., Vol.27, pp.609~622, 1989.
22) Brussion, G. and Sonnad, V. : A comparison of direct and preconditioned iterative techniques for sparse, unsymmetric systems of linear equations, Int. J. Numer. Methods Eng., Vol.27, pp.801~815, 1989.
23) 吉田・依知川・中川：連立一次方程式算法の計算機シス

テム適合性に関する一検討，構造工学における数値解析法シンポジウム論文集，Vol. 13，pp. 149～154，1989.
24) 吉田・依知川・中川：ICCG 法の有限要素方程式への適用に関する一検討，構造工学論文集，Vol. 36 A，pp. 255～262，1990.
25) 吉田・中川：CG 法を基礎とする連立一次方程式解法の効率化のアルゴリズム，構造工学における数値解析法シンポジウム論文集，Vol. 14，pp. 371～376，1990.
26) 村田・小国・唐木：スーパーコンピュータ 科学技術計算への適用，pp. 141～149，丸善，1985.
27) Van der Vorst, H.A. : A vectorizable variant of some ICCG methods, SIAM J. Sci. Stat. Comp., Vol.3, No.3, pp.350～356, Sep. 1982.
28) 速水謙：SCG 法による拡散方程式の解法，コンピュートロール，No. 26，pp. 32～39，1989.
29) 渡部・名取・小国：Fortran 77 による数値計算ソフトウェア，pp. 289～292，丸善，1989.
30) 森・名取・鳥居：数値計算 岩波講座情報科学—18, p. 88, 岩波書店，1982.

(1990. 11. 8 受付)

ON REFINEMENT OF THE METHODS OF CONJUGATE GRADIENTS FOR THE SOLUTION OF LARGE SPARSE FINITE ELEMENT SYSTEMS

Yutaka YOSHIDA, Masaya NAKAGAWA and Tomoyuki TANAKA

In this paper, the storage requirements and performance consequences of different implementations of the conjugate gradient methods for the solution of sparse systems arising from finite element structural analysis are reviewed.
Although the ICCG method has good convergence rates, it is not so effective in computing times because of the expence of computing forward and backward substitution at each iteration. Since algorithms for using the incomplete Choleski procedure are unsuitable for vectorization, it is concluded that the SCG method will be of advantage over the ICCG method on supercomputer systems.
The convergence characteristics of the ICCG method depends on the accelerating factor introduced into the resulting diagonal elements. This paper presents a possible means to decide the parameter which achieves the optimal convergence.