

有限要素解析用並列化ICCGソルバの開発について

岩下, 武史
京都大学大型計算機センター

金澤, 正憲
京都大学大型計算機センター

島崎, 眞昭
京都大学大学院工学研究科

<https://doi.org/10.15017/1470706>

出版情報 : 九州大学情報基盤センター年報. 2, pp.55-62, 2002-03. Computing and Communications Center, Kyushu University

バージョン :

権利関係 :



有限要素解析用並列化ICCG ソルバの開発について

岩下 武史[†]

Takeshi Iwashita[†]

金澤 正憲[†]

Masanori Kanazawa[†]

島崎 眞昭[‡]

Masaaki Shimasaki[‡]

[†] … 京都大学大型計算機センター

[†] … Data Processing Center, Kyoto University

[‡] … 京都大学大学院工学研究科

[‡] … Graduate School of Engineering, Kyoto University

要旨 有限要素解析などにおいて生ずるランダム・スパースな係数行列を持つ連立一次方程式に対するブラックボックス的な並列化 ICCG 法について述べる。従来よく用いられるブロック ICCG 法 (Localized ICCG 法) に加えて、ソルバの中でリオーダーリング処理を自動的に行う手法について述べる。約 100 万自由度の電磁界有限要素解析を例にとり、多色順序付け法を非構造型解析に応用した代数学的多色順序付け法により高い速度向上が得られることを示す。

Abstract The present paper discusses a black-box type parallel ICCG solver for a linear system of equations with a random sparse coefficient matrix, which arises in finite element analyses. The algebraic multi-color ordering method is mainly discussed. The method achieves a high speed-up ratio on a electromagnetic field analysis with one million degrees of freedom.

1 はじめに

偏微分方程式の境界値問題を有限要素法や差分法で解く際に生ずる正値対称な係数行列を持つ連立一次方程式の一般的な解法として ICCG (Incomplete Cholesky Conjugate Gradient) 法 [1] がある。ICCG 法は、共役勾配ソルバと不完全コレスキー分解に基づいた前処理部により構成される。このうち、共役勾配ソルバ部分については、係数行列や残差ベクトルを行方向に分割することで容易に並列化することができる。一方、不完全コレスキー分解と前処理として行われる代入計算は、逐次型計算であるためにその並列化が困難である。なかでも、代入計算は反復の度に実行されるので、全体の計算コストに対する影響が大きく、その並列化は課題である。これまでも、代入計算を含む ICCG 法の並列化に関して様々な研究 [2] がなされている。本稿では、これらの並列化手法のうち、有限要素解析などにおいて生ずるランダム・スパースな係数行列に対するブラックボックスソルバ的な手法について述べる。

実アプリケーションにおける並列化 ICCG ソルバの代表的なものとして、ブロック ICCG 法 (Localize

ICCG) がある。本手法は、係数行列を各プロセッサにブロック分割し、不完全コレスキー分解時にプロセッサにまたがる要素を無視する手法である。このことにより、不完全分解、代入計算ともに並列処理が可能となる。同手法は、分散メモリ型並列計算機に適合した方法であるが、共有メモリ型並列計算機にも実装が可能で、実装方法も容易であり、広く用いられている。しかしながら同手法はプロセッサ数が増加するに従って前処理において無視される係数行列内の要素が増加し、前処理の効果が薄れるという問題点がある。例えば、電磁界有限要素解析を対象とした著者らの数値実験においても、本現象が確認されている [3]。そこで、前処理効果の低下を補償する手法として、ブロック間のオーバーラップを用いる手法がある。このオーバーラップを用いる手法は、非対称係数行列用の GMRES 法の ILU 前処理において効果的であるという報告がされている [4]。また、ICCG 法においてもその効果は確認され、Overlapped localized ICCG 法と呼ばれている。しかし、どの程度オーバーラップさせればよいかなど、対象とする問題に依存する部分がある。また、オーバーラップさせることにより前処理効果がかえって低下する例も報告されており、現状では問題点もある。

上記のブロック ICCG 法による数値実験結果を踏まえ、著者らはソルバ内で自動的にリナンバリング処理を行うブラックボックス型並列化 ICCG ソルバ（リナンバリング処理付き並列化 ICCG 法）を提案した [3]。同手法では、係数行列は自動的に並列処理に適した形にリナンバリングされる。但し、リナンバリングの種類に応じて、演算の順序をかえることにより対応する場合、明示的に行う場合に分かれ、後者の場合、メモリに関して元の係数行列を保持するか否かの二種類にさらに分かれる。本手法の場合、前処理行列の構築において無視される行列要素がないため、ブロック ICCG 法よりも高い前処理効果が期待される。

本稿では、リナンバリング処理付き並列化 ICCG 法の並列処理向きオーダリングとして、二種類のオーダリングを紹介する。

一つ目はバンド行列に対する並列化オーダリングである。 N_p 台のプロセッサに対して、係数行列を N_p 個の内部領域と $N_p - 1$ 個のインターフェース領域に分ける。インターフェース領域の行数はバンド幅と同じとする。インターフェース領域を内部領域の後に順序づけると、各領域で代入計算は並列に行うことができる。このリオーダリング処理は明示的に行う必要はなく、元の係数行列に対する不完全コレスキー分解、代入計算の計算順序を変更することで行うことができる。本手法は、プロセッサ数が少数の場合（およそ 8 プロセッサ以下）の場合に有効である。それ以上のプロセッサでは、通常内部領域が十分にとれなくなったり、収束性の低下を招いたりする場合が多い。

そこで、二つ目のオーダリングとして、プロセッサ数が多い場合に有効な代数学的多色順序付け法を提案している [5]。従来、並列処理向きオーダリングの研究は主として差分格子を対象として行われているが、その過程において並列度と前処理効果のトレードオフが発見されている [6]。即ち、多くのプロセッサが使用可能な（並列度が高い）オーダリングほど、不完全コレスキー分解前処理の効果が小さくなる。そこで、土肥らは並列オーダリングの収束性を定量的に評価するためにオーダリンググラフを利用することを提案し、高い前処理効果と並列度を同時に実現する大きな色数による多色順序付けを提案した [7][8]。代数学的多色順序付けは、この多色順序付けをブラックボックス的に行う手法で、非構造型の解析への実装を可能にするものである。従って、適用範囲はバンド行列に限定されず、一般のランダム・スパース行列に適用できる。このような、多色順序付け法の特徴を非構造型の解析に導入する手法としては、他に未知数の Independent Set に

基づく手法がある [11] [12]。

本稿では、約 100 万自由度の有限要素解析を数値計算例として用い、代数学的多色順序付け法の有効性について示す。

2 並列化 ICCG 法

本解析では、有限要素解析における正値対称な係数行列を持つ連立一次方程式を対象とする。係数行列は、ランダム・スパース行列とし、係数行列の非ゼロ要素のみをメモリに格納する。

本稿では、全体マトリクスの次元数、バンド幅、使用プロセッサ台数をそれぞれ n , n_b , N_p で表記する。

2.1 前処理付き共役勾配法の並列化

前処理付き共役勾配法 (PCG 法) は正値対称な線形システムの反復解法として最も一般的なソルバである。PCG 法は、(i) 前処理、(ii) 内積計算、(iii) 行列・ベクトル積計算、(iv) ベクトルの更新 の 4 つの部分から構成され、これらは他の多くの反復解法も同様である。このうち、(ii), (iii), (iv) の計算は係数行列や各種のベクトルを行ブロックに分割することで並列化できる。一方、前処理部の並列化は前処理の種類によってその困難さは異なる。例えば、対角前処理（スケールリング）ではプロセッサ間の通信は必要とされず、並列化に関するトレードオフはない。しかし、逐次型の解析コードで通常よく用いられる IC(不完全コレスキー) 分解前処理の場合、前処理部は逐次的な前進・後退代入計算により構成され、その並列化は困難である。そこで、次節以降で ICCG 法に伴う前進・後退代入計算の並列化に関する手法について述べる。

2.2 ブロック ICCG 法

ICCG 法に伴う前進・後退代入計算の並列化手法の一つとして、ブロック ICCG 法 (BICCG 法) [13] がある。BICCG 法の特徴は、IC 分解においてプロセッサ間にまたがる行列要素を無視することである。この場合、前進・後退代入計算に用いられる分解行列、下三角行列 $\bar{L}$ 、上三角行列 $\bar{L}^T$ は図 1 に示されるような形となり、代入計算は各プロセッサで通信を伴うことなく並列に行うことができる。

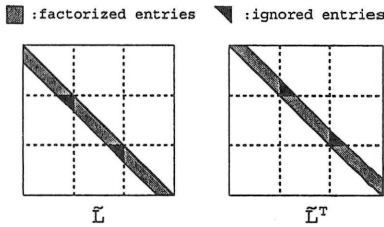


図 1: ブロック ICCG 法による分解行列 (3 プロセッサ)

2.3 リナンバリング処理付き ICCG 法 [3]

前節で述べた BICCG 法は、代入計算を通信なく並列処理できるが、使用プロセッサ数が増えるにつれて IC 分解時に無視される行列要素が単調増加する問題点がある。このことは、プロセッサ数が増加するに従い、IC 分解の近似度が下がり、前処理の効果が弱まることを意味する。

そこで、著者らは分解中に行列要素を無視することなく代入計算を行う方法として、リナンバリング処理付き並列化 ICCG 法 (PICCG-RP 法) を提案している。本手法は、ソルバ側で全体マトリクスを並列処理に適した形状にリナンバリングして処理する手法である。本手法はバンド係数行列を対象とするもので、まず、全体マトリクスを図 2(a) に示されるように、 N_p 個の n_{in} 行ブロック (ここでは内部領域部と呼ぶ) と $N_p - 1$ 個の n_b 行ブロック (ここではインターフェース部と呼ぶ) に分割する。内部領域部の行数 n_{in} は次式で与えられる。

$$n_{in} = \{n - (N_p - 1)n_b\} / N_p \quad (1)$$

ここで、 i 番目のプロセッサは i 番目の内部領域部とインターフェース部に関する行ブロックを担当する。即ち、 i 番目のプロセッサは、全体マトリクスのうち $(n_{in} + n_b)(i - 1) + 1 \sim (n_{in} + n_b)i$ 行を担当する (図 2(a) 参照)。次に、リナンバリング処理により、図 2(b) のように、内部領域から先に番号がつけられるように変更する。 i 番目のプロセッサの担当行のうち、 $(n_{in} + n_b)(i - 1) + 1$ から $n_{in}i + n_b(i - 1)$ の行ブロックは、 $n_{in}(i - 1) + 1$ から $n_{in}i$ の行ブロックに移され、 $n_{in}i + n_b(i - 1) + 1$ から $(n_{in} + n_b)i$ の行ブロックは、 $n_{in}N_p + n_b(i - 1) + 1$ から $n_{in}N_p + n_b i$ の行ブロックに移される。この処理を行なうことにより、全体マトリクスは dissection 分割法により得られるような図 2(b) 左に示される形状となる。

全体マトリクスが前述のような形に変換された後、

前進後退代入計算は以下のような手順で並列に実行される。前進代入計算では、まず内部領域部に関する処理が各プロセッサで並列に行なわれる。次に、 i 番目のプロセッサは隣接の $i+1$ 番目のプロセッサに n_b 個の配列要素を送信する。次に、インターフェース部に関する処理が各プロセッサで並列に行なわれる。後退代入計算では、まずインターフェース部に関する処理が行なわれ、隣接のプロセッサへの通信を経て、内部領域部に関する処理が行なわれる。従って、並列化が困難な代入計算を、 n_b 個の配列要素を隣接のプロセッサに送る通信を 2 度行なうだけで並列に処理できる。

本手法において、係数行列のリナンバリング処理に必要な計算コストは、一回の行列・ベクトル積計算と同程度であり、以降の繰り返し部分と比べて十分小さい。また、プログラミングを工夫することで、リナンバリング処理を明示的行わず、演算順序を変更することにより等価な前処理を行うこともできる。

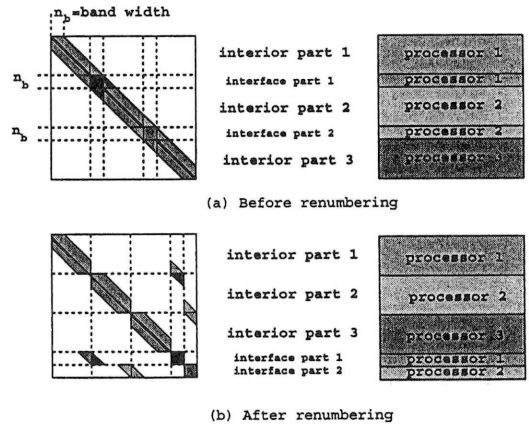


図 2: リナンバリング処理 (3 プロセッサ)

2.4 代数学的多色順序付け [5]

前節で述べた並列化リオーダーリングは、バンド幅が係数行列に対して十分に小さい場合に有効な手法であり、一般のランダム・スパース係数行列への適用は困難である。また、十分な収束性が得られる並列度には限界があることも分かっている。そこで、本節では、差分解析においてその高い有用性が示されている多色順序づけを一般のランダム・スパース係数行列に拡張する代数学的多色順序付け法 (Algebraic Multi-Color Ordering Method) について述べる。

多色順序付け法では、互いにデータ依存関係のない未知変数をまとめ、一つの色とみなす。色の順に未知変数を並べ替えることにより、代入計算は各色ごとに並列化される。代数学的多色順序付けの特徴は、係数行列より得られる情報のみに基づいて、未知変数の色づけを行うことにある。本手法の詳細を以下に示す。

手法の説明のため、以下の表記を導入する。解くべき連立一次方程式を $Ax = b$ とする。係数行列 A の狭義下三角行列を L とし、その i 行の非ゼロ要素の列番号を $\text{lnzc}(i, j)$ ($j=1, \dots, \text{lnz}(i)$) とする。ここで、 $\text{lnz}(i)$ は i 行の非ゼロ要素数である。

まず、代数学的多色順序付けで用いる色数 n_{color} を与える。ここで、色数としては、差分解析における結果から 30 から 100 程度が適当である。但し、色数は色付けの過程で自動的に増加される場合がある。未知変数に順次色付けを行うが、このとき以下の条件をみたすようにする。

条件 (MC: Multi-Color) — 同色の未知変数がデータ依存関係を持たない。

配列 $\text{color}(i)$ ($i=1, \dots, n$) により、未知変数 i の色を表すものとする。各未知変数の色は Fig. 3 に示されるプログラムで決められる。与えられた色数 n_{color} に対して、一番目の未知変数より順次 1 から n_{color} までの色を再帰的につけていく。このとき、上記の条件 (MC) が満たされるように、 i 番目の未知変数の色づけは以下のように行う。ここで、直前に付けられた色を $\text{icolor} - 1$ とする。

- Step 1: $\text{lnzc}(i, j)$ 番目の未知変数の色を調べ、 icolor が付けられていなければ、 icolor を i 番目の未知変数の色として色づけし、 $i+1$ 番目の要素に移る。そうでない場合、Step 2 へ。
- Step 2: icolor を $\text{mod}(\text{icolor}, n_{\text{color}})+1$ に更新し、Step 1 へ

この手順を全ての未知変数に対して行う。ここで、 $\text{lnzc}(i, j)$ ($j=1, \dots, \text{lnz}(i)$) の色として、0 から $n_{\text{color}} - 1$ までの色が全て使用されていた場合には、 n_{color} を 1 増加する。上記の手順では、色を Step 2 においてインクリメントしても新たな色がすでに割り当てられている場合、Step 1 と Step 2 を数回繰り返すことになる。そこで、各未知変数の割り当て時において、Step 1 ですでに使用されている色をメモリに保持し、これと異なる色を割り当てる方法もある。この場合はプログラムが若干複雑になるが、無駄な演算が除去できる。

全ての未知変数の色付けが終了したら、未知変数を色の順にリオーダーリングする。色の設定、係数行列や右辺ベクトルのリオーダーリングにかかる計算コストは、およそ一回の行列・ベクトル積計算程度であり、反復計算のコストに比べて通常はるかに小さい。

代数学的多色順序付けにおける代入計算は以下のようになら行される。 $\tilde{A}\tilde{x} = \tilde{b}$ をリオーダーリングされた連立一次方程式とする。このとき、解ベクトル $\tilde{x}$ と係数行列 $\tilde{A}$ は

$$\tilde{x} = \begin{pmatrix} \tilde{x}_{C(1)} \\ \tilde{x}_{C(2)} \\ \vdots \\ \tilde{x}_{C(n_{\text{color}})} \end{pmatrix}, \quad (2)$$

$$\tilde{A} = \begin{pmatrix} \tilde{C}_{1,1} & \tilde{C}_{1,2} & \dots & \tilde{C}_{1,n_{\text{color}}} \\ \tilde{C}_{2,1} & \tilde{C}_{2,2} & \dots & \tilde{C}_{2,n_{\text{color}}} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{n_{\text{color}},1} & \tilde{C}_{n_{\text{color}},2} & \dots & \tilde{C}_{n_{\text{color}},n_{\text{color}}} \end{pmatrix} \quad (3)$$

のように色毎に分割される。ここで、 $\tilde{x}_{C(ic)}$ は ic 番目の色に対応し、 $\tilde{C}_{l,m}$ は $\tilde{x}_{C(l)}$ と $\tilde{x}_{C(m)}$ の関係を表す。未知変数の色が条件 (MC) を満たすように決められたとき、小行列 $\tilde{C}_{ic,ic}$ ($ic = 1, \dots, n_{\text{color}}$) は対角行列で与えられる (図 4 参照)。従って、不完全分解行列は

$$\tilde{L} = \begin{pmatrix} \tilde{D}_1 & & & \\ \tilde{L}_{2,1} & \tilde{D}_2 & & 0 \\ \vdots & \ddots & \ddots & \\ \tilde{L}_{n_{\text{color}},1} & \tilde{L}_{n_{\text{color}},2} & \dots & \tilde{D}_{n_{\text{color}}} \end{pmatrix}, \quad (4)$$

$$\tilde{D} = \begin{pmatrix} \tilde{D}_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \tilde{D}_{n_{\text{color}}}^{-1} \end{pmatrix} \quad (5)$$

where

$$\tilde{A} \simeq \tilde{L}\tilde{D}\tilde{L}^T, \quad (6)$$

と書け、 $\tilde{D}_{ic}$ ($ic = 1, \dots, n_{\text{color}}$) は対角行列である。

残差ベクトル $\tilde{r}$ を用いて、前進代入計算を

$$\tilde{L}\tilde{y} = \tilde{r}, \quad (7)$$

のように書くと、(4) より、 ic 色に対する前進代入計算は

$$\tilde{y}_{C(ic)} = \tilde{D}_{ic}^{-1}(\tilde{r}_{C(ic)} - \sum_{k=1}^{ic-1} \tilde{L}_{ic,k} \tilde{r}_{C(k)}), \quad (8)$$

```

!-----
ncolor=some value      ! Set number of used colors
color(1:n)=0           ! Initialize the array color
icolor=1
do i=1,n
  j=1
  do while(j <= lnz(i))
    if (color(lnz(i,j))==icolor) then
      icolor=mod(icolor+1,ncolor)+1 !To next color
      j=0
    endif
    j=j+1
  enddo
  color(i)=icolor        ! Assignment of color
  icolor=mod(icolor+1,ncolor)+1 !To next color
enddo
!-----

```

図 3: 代数学的色付け

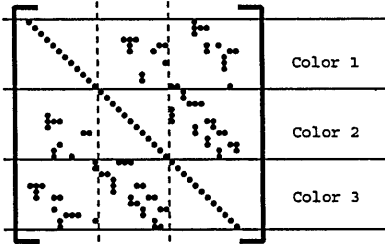


図 4: 代数学的多色順序付けにおける係数行列 (3色)

のように書ける。但し、 $\tilde{y}_{C(k)}$ および $\tilde{r}_{C(k)}$ は、 $\tilde{y}$, $\tilde{r}$ の k 番目の色に対応した部分である。 $\tilde{r}_{C(k)} (k=1, \dots, ic-1)$ がすでに計算され、その結果を全てのプロセッサが共有しているとすると、前進代入計算 (8) は行列・ベクトル積で構成され、並列化できる。即ち、前進代入計算は色毎に並列化できる。また、後退代入計算も同様に並列化できる。図 5 に、代数学的多色順序付けによる並列化代入計算の手順を示す。一回の前進・後退代入計算において、 $ncolor-1$ 回の通信 (同期) が必要となる。本手法によって得られる並列度は、一色あたりの未知変数の数に等しく、およそ $n/ncolor$ である。

3 解析結果

3.1 定式化

本解析では線形渦電流問題を対象とし、時間的な繰返し計算により、電磁界の過渡的な振る舞いについて調べる。解析対象内の電磁界を記述する方程式は、マクスウェル方程式において変位電流の項を無視することにより与えられる。本解析では、辺要素を使用し、磁気ベクトルポテンシャルのみによる定式化を行う A -

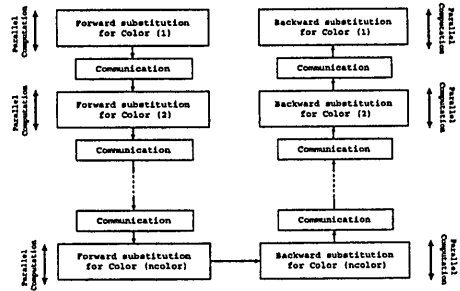


図 5: 代数学的多色順序付けにおける並列化代入計算

法を用いるので、支配方程式は次式で与えられる。

$$\nabla \times (\nu \nabla \times A_m) = -\sigma \frac{\partial A_m}{\partial t} + J_0 \quad (9)$$

ここで、 A_m は磁気ベクトルポテンシャル、 J_0 は強制電流の電流密度、 ν は磁気抵抗率、 σ は導電率を表す。磁気ベクトルポテンシャルをベクトル補間関数により近似展開し、式 (9) にガラーキン法を適用することにより、次式が得られる。

$$[K]\{A_m\} + [M] \frac{\partial \{A_m\}}{\partial t} - \{J\} = 0 \quad (10)$$

ここで、 $\{A_m\}$ は未知変数 A_{mi} からなる列ベクトルを表す。 $[K]$, $[M]$ は行列、 $\{J\}$ は列ベクトルを表し、以下のように与えられる。

$$K_{ij} = \sum_e \iiint_e (\nabla \times N_i) \cdot (\nu \nabla \times N_j) dV \quad (11)$$

$$M_{ij} = \sum_e \iiint_e \sigma N_i \cdot N_j dV \quad (12)$$

$$J_i = \sum_e \iiint_e N_i \cdot J_0 dV \quad (13)$$

ここで、 e は各要素、 m は全要素数、 N はベクトル補間関数を表す。未知変数の総数を n として、行列 $[K]$, $[M]$ は n 次正方行列、 $\{A_m\}$ および $\{J\}$ は n 次元ベクトルである。

式 (10) 中の時間微分項は、後退差分法を用いて以下のように離散化して解く。

$$\frac{\partial \{A_m\}}{\partial t} = \frac{\{A_m\} - \{A_{mold}\}}{\Delta t} \quad (14)$$

ここで、 $\{A_{mold}\}$ は 1 タイムステップ前の $\{A\}$, Δt は時間刻み幅である。

式 (10)(14) より、各タイムステップにおいて解かれる全体方程式は次式で与えられる。

$$[Q]\{A_m\} = \{b\} \quad (15)$$

表 1: 解析の諸元

要素数	327680
節点数	342225
未知数の総数	1011920
時間刻み幅	1 msec

表 2: 解析結果

(a) 逐次型 ICCG 法 (辞書式順序付け)

N_p	計算時間 (秒)	反復回数
1	2119	366

(b) ブロック ICCG 法

N_p	計算時間 (秒)	反復回数
2	1394	530
4	1028	779
8	550.5	834
16	251.6	675
32	120.4	726

(c) 代数学的多色順序付け法 (60 色)

N_p	計算時間 (秒)	反復回数
2	1194.4	390
4	606.3	390
8	308.4	390
16	164.3	390
32	99.83	390

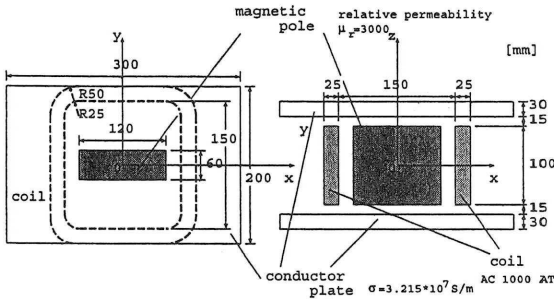


図 6: 電気学会 3 次元渦電流解析検証モデル (unit: [mm])

但し,

$$[Q] = ([K] + \frac{1}{\Delta t}[M]), \quad \{b\} = \frac{1}{\Delta t}[M]\{A_{old}\} + \{J\} \quad (16)$$

解析領域中に非導電性の部分 (空気領域) が含まれる場合, 係数行列 $[Q]$ は半正定値となる。

3.2 解析モデル

本解析では, 電気学会 3 次元渦電流解析検証モデルを用いる [14]. 表 1 に解析の諸元を示す. 解析は京都大学大型計算機センター VPP-800 上でおこなった. MPI ライブラリによる並列化を行い, ICCG 法の収束判定基準は, 右辺ベクトルと残差ベクトルの差が 10^{-7} 以下となる条件を用いる. ICCG 法の加速係数は 1.03 とする. 各手法の性能評価は一タイムステップあたりの計算時間により行う。

3.3 解析結果

表 2, 図 7 にブロック ICCG 法と代数学的多色順序付け法による計算 (経過) 時間, 反復回数, 収束の振る舞いを示す. ブロック ICCG 法では, プロセッサ台数が増加するにつれ, 前処理効果が薄れ, 反復回数が増加している。

一方, 代数学的多色順序付け法では, ICCG 法の収束性はプロセッサ台数とは無関係で, 色数に依存する。

表 3: 代数学的多色順序付け法における色数, 並列度, 反復回数

色数	並列度	反復回数
50	20200	426
60	16800	390
70	14500	393
100	10200	440
500	2000	383

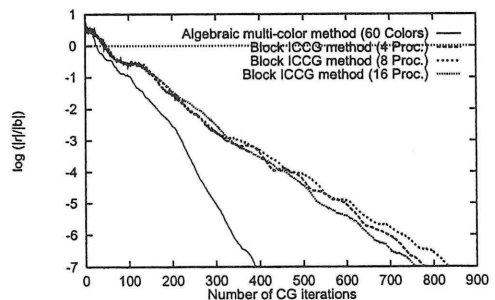


図 7: 収束の比較

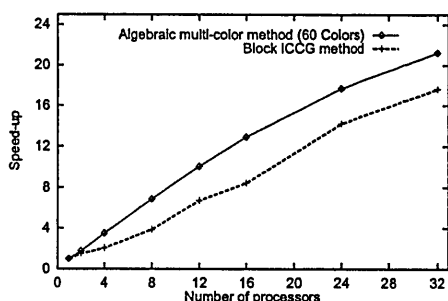


図 8: 代数学的多色順序付け法による速度向上

表 3 に色数を変化させた場合の、並列度と反復回数を示す。ブロック ICCG 法と比べて、少ない反復数の増加で高い並列度を実現していることが分かる。色数を増やすと一般に収束性が増すが、通信（同期）の回数が増えるため、それらのコストにより速度向上が飽和する。従って、最適な色数の選択は困難であるが、著者らの検討によると、差分解析で推奨される 30~100 色という値は有限要素解析においても有効であり、この範囲内であればどのような色を選択したとしても大幅な収束性の低下は避けられると考えている。本解析では、30~100 色の間でいくつかの色をテストしたが、60 色の場合にもっとも良い結果を得た。このとき、反復数の増加は逐次型ソルバと比べて 10% 程度であり、並列度は約 16800 に達する。その結果、図 8 に見られるようにブロック ICCG 法と比べて高い速度向上を得る。

4 結論

本稿では、非構造型の有限要素法における並列化 ICCG 法としてブラックボックス型のソルバについて述べた。その中で、差分法における多色順序付け法を拡張した代数学的多色順序付け法による解析結果を示した。同手法により、約 100 自由度の電磁界解析において 32 プロセッサで 20 倍の速度向上を得た。

参考文献

[1] J. Meijerink and H. A. van der Vorst, "An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix Is a Symmetric M-matrix," *Mathematics of Computation*, 31, (1977), pp. 148-162.

[2] H. A. van der Vorst and T. F. Chan, "Parallel Preconditioning for Sparse Linear Equations," *ZAMM. Z. angew. Math. Mech.*, 76, (1996), pp.167-170.

[3] T. Iwashita and M. Shimasaki, "Parallel Processing of 3-D Eddy Current Analysis with Moving Conductor Using Parallelized ICCG Solver with Renumbering Process," *IEEE Transactions on Magnetics*, 36, (2000), pp. 1504-1509.

[4] G. Radicati di Brozolo and Y. Robert, "Parallel Conjugate Gradient-like Algorithms for Solving Sparse Nonsymmetric Linear Systems on a Vector Multiprocessor," *Parallel Computing*, 11, (1989), pp. 223-239.

[5] T. Iwashita and M. Shimasaki, "Algebraic Multi-Color Ordering for Parallelized ICCG Solver in Finite Element Analyses," (2002), to appear.

[6] I. S. Duff and G. A. Meurant, "The Effect of Ordering on Preconditioned Conjugate Gradients," *BIT*, 29, (1989), pp.635-657.

[7] S. Doi and A. Lichnewsky, "A Graph-Theory Approach for Analyzing the Effects of Ordering on ILU Preconditioning," *INRIA report 1452*, (1991).

[8] S. Doi and T. Washio, "Ordering Strategies and Related Techniques to Overcome the Trade-off Between Parallelism and Convergence in Incomplete Factorization", *Parallel Computing*, 25, (1999), pp. 1995-2014.

[9] T. Iwashita and M. Shimasaki, "Construction and Ordering of Edge Elements for Parallel Computation," *IEEE Transactions on Magnetics*, 37, (2001), pp. 3498-3502.

[10] I. S. Duff and H. A. van der Vorst, "Developments and Trend in the Parallel Solution of Linear Systems," *Parallel Computing*, 25, (1999), pp. 1931-1970.

[11] 襲田 勉, 丸山 訓英, 鷲尾 巧, 土肥 俊, 山田 進, 「非構造メッシュ用 BILU 前処理付き反復法のベクトル・並列化手法」, 情報処理学会論文誌, Vol.41 No. SIG 8 (HPS 2), (2000), pp. 92-99.

- [12] M. T. Jones and P. E. Plassmann, "The efficient parallel iterative solution of large sparse linear systems," *Graph Theory and Sparse Matrix Computations*, IMA, 56, (1994), pp. 229-245.
- [13] M. L. Barton, "Three-dimensional magnetic field computation on a distributed memory parallel processor," *IEEE Trans. Magn.*, vol. 26, No. 2, March 1990, pp.834-836.
- [14] T. Nakata, N. Takahashi, T. Imai, and K. Muramatsu, "Comparison of Various Methods of Analysis and Finite Elements in 3-D Magnetic Field Analysis," *IEEE Trans. Magn.*, vol. 27, (1991), pp.4073-4076.
- [15] T. Washio and K. Hayami, "Overlapped Multicolor MILU Preconditioning," *SIAM Journal of Scientific Computing*, 16, (1995), pp. 636-650.
- [16] J. J. Dongarra, I. S. Duff, D. C. Sorensen and H. A. van der Vorst, "Solving Linear Systems on Vector and Shared Memory Computers," *SIAM*, (1991).
- [17] R. Barrett, et al., "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods," *SIAM*, (1994).
- [18] 小国 力 編著, 村田 健郎, 三好 俊郎, ドンガラ J. J., 長谷川 秀彦著, 「行列計算ソフトウェア WS, スーパーコン, 並列計算機」, 丸善, (1991).
- [19] 岩下 武史, 島崎 眞昭, 「多色順序付けを用いた並列化 ICCG ソルバに関する検討 ―ブロック化による性能向上と工学的応用―」, 第8回「ハイパフォーマンスコンピューティングとアーキテクチャの評価」に関する北海道ワークショップ (2001), pp. 55-60.