

## ブラックボックス型並列化ICCGソルバの開発について

岩下, 武史  
京都大学学術情報メディアセンターコンピューティング研究部門

<https://doi.org/10.15017/1470413>

---

出版情報：九州大学情報基盤センター広報：全国共同利用版. 2 (2), pp.117-127, 2002-07. 九州大学  
情報基盤センター  
バージョン：  
権利関係：

# ブラックボックス型並列化 ICCG ソルバの開発について

岩下 武史<sup>1</sup>

## 1 はじめに

この度、九州大学の情報基盤センター広報 全国共同利用版 において研究紹介をする機会を得た。そこで、本稿では著者とその研究グループが研究を行っている HPC 分野に関する研究の中で、並列化 ICCG 法に関して簡単に紹介する。本稿では、紙面の都合もあり、研究の背景、従来研究などは文献引用中心とし、2 種類の並列化 ICCG 法の手法を紹介し、実応用での計算例を示す。また、京都大学の学術情報メディアセンターで公開中の並列化ソルバについてごく簡単に紹介する。

スーパーコンピュータの実用例として重要な数値シミュレーションは、多くの場合大規模な連立一次方程式に帰着する。この連立一次方程式の求解法として最も一般的なものに、ILU 分解前処理付き反復法 [1][2] がある。ICCG (Incomplete Cholesky Conjugate Gradient) 法 [3] はこれらの手法の一つで、対称な係数行列に対する ILU 分解 (不完全コレスキー分解ともいう) を前処理とする共役勾配法である。しかし、ILU (IC) 分解前処理は、反復中で前進・後退代入計算として実装されるためその並列化が困難である。そこで、代入計算を含む ICCG 法の並列化に関して様々な研究 [1] [4] [5] 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しても適用可能なブラックボックスソルバ的な 2 種類の手法について取り上げる [6]。ここで紹介する手法は、非対称係数行列用の ILU 分解前処理についても応用が可能である。

まず、1 つ目の手法として、実アプリケーションにおける代表的な並列化 ICCG ソルバであるブロック ICCG 法 (Localized ICCG 法ともいう) [2] [7] について述べる。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。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。この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もあるが、対象とする問題によってはかえって前処理効果が低下する例も報告されており、ブロック ICCG 法の改良法として必ずしも有効でない。ただし、非対称用のブロック ILU 分解前処理では、オーバーラップは有効となる場合が多いようである [8]。

2 つ目の手法は、著者らが提案している手法で、ソルバ内で自動的に連立一次方程式の係数行列をリオーダーリングする手法である [9]。様々なリオーダーリング手法が考えられるが、本稿では多色順序付け法を代数的に行う代数的多色順序付け法 [10] [11] について述べる。多色順序付け法は従来差分解析の並列処理手法の一つとして知られているものであり、同手法はこれ

<sup>1</sup> 京都大学学術情報メディアセンター コンピューティング研究部門  
iwashita@media.kyoto-u.ac.jp

をランダム・スパース行列用に拡張したものである。このような多色順序付け法の非構造型解析への拡張では、他に [12] [13] の研究がある。このように、代数学的な多色順序付けにはその色分け手法において様々な方法が考えうるが、本稿で紹介する著者らが提案している手法は従来のものと若干異なる。従来法との優劣は対象とする問題に応じてかわってくるため一概に述べることはできないが、提案手法は実装が容易であるという利点をもつ。

本稿では、約 100 万自由度の電磁界有限要素解析を数値計算例として用い、並列化 ICCG ソルバ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。

### 2.1 ICCG 法の並列化

ICCG 法は正値対称な係数行列を持つ連立一次方程式の反復解法として最も一般的なソルバである。ICCG 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。従って、ICCG 法の並列化に関する研究では、主としてこの代入計算の並列化アルゴリズムが検討される。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ことにする。

### 2.2 ブロック ICCG 法 [2] [7]

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)[7]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  $\tilde{L}$ 、上三角行列  $\tilde{L}^T$  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。しかしながら、BICCG 法は、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。仮に係数行列が帯行列で、バンド内が密であるとする、バンド幅  $n_b$ 、プロセッサ数  $N_p$  に対して、無視される行列要素の数は  $n_b^2 N_p / 2$  となり、プロセッサ数に正比例する。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。

### 2.3 代数学的多色順序付け [10]

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付け法を一般のランダム・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

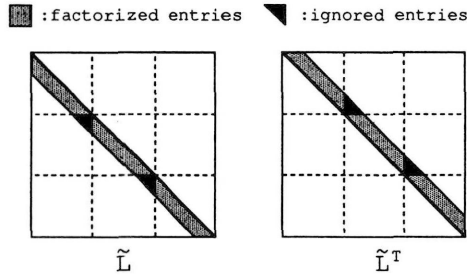


図 1: ブロック ICCG 法による分解行列 (3 プロセッサ)

Method) について述べる. 代数学的多色順序付け法では, ソルバ内で自動的に係数行列や各種ベクトルを並列処理に適した形に再順序付ける. 同手法において最も重要な点は, 未知変数をどのように色分けし再順序付けるかにあり, 下記にその詳細を述べる.

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では, この未知変数の色分けを係数行列より得られる情報のみに基づいて行う. 手法の説明のため, 以下の表記を導入する. 解くべき  $n$  元連立一次方程式を  $Ax = b$  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くする.

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. 各未知変数の色は図 2 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,  $\text{icolor}$  が付けられていなければ,  $\text{icolor}$  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:  $\text{icolor}$  を  $\text{mod}(\text{icolor}, n_{color}) + 1$  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  $n_{color} - 1$  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並列実行される。 $\tilde{A}\tilde{x} = \tilde{b}$ 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

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

$$\tilde{A} = \begin{pmatrix} \tilde{C}_{1,1} & \tilde{C}_{1,2} & \cdots & \tilde{C}_{1,ncolor} \\ \tilde{C}_{2,1} & \tilde{C}_{2,2} & \cdots & \tilde{C}_{2,ncolor} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{ncolor,1} & \tilde{C}_{ncolor,2} & \cdots & \tilde{C}_{ncolor,ncolor} \end{pmatrix}. \quad (2)$$

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  $\tilde{C}_{ic,ic}(ic = 1, \dots, ncolor)$  は対角行列で与えられる (図 3 参照)。従って、不完全分解行列は

$$\tilde{L} = \begin{pmatrix} \tilde{D}_1 & & & \\ \tilde{L}_{2,1} & \tilde{D}_2 & & O \\ \vdots & \ddots & \ddots & \\ \tilde{L}_{ncolor,1} & \tilde{L}_{ncolor,2} & \cdots & \tilde{D}_{ncolor} \end{pmatrix}, \quad (3)$$

$$\tilde{D} = \begin{pmatrix} \tilde{D}_1^{-1} & & & O \\ & \ddots & & \\ O & & & \tilde{D}_{ncolor}^{-1} \end{pmatrix} \quad (4)$$

但し

$$\tilde{A} \simeq \tilde{L}\tilde{D}\tilde{L}^T, \quad (5)$$

と書け、 $\tilde{D}_{ic}(ic = 1, \dots, ncolor)$  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

$$\tilde{L}\tilde{y} = \tilde{r}, \quad (6)$$

のように書くと、(3) より、 $ic$  色に対する前進代入計算は

$$\tilde{y}_{C(ic)} = \tilde{D}_{ic}^{-1}(\tilde{r}_{C(ic)} - \sum_{k=1}^{ic-1} \tilde{L}_{ic,k}\tilde{y}_{C(k)}), \quad (7)$$

```

!-----
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 <= lnzc(i))
    if (color(lnzc(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 2: 代数学的色付け

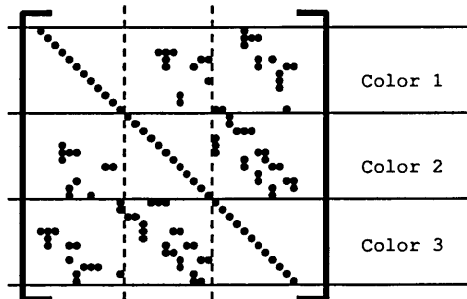


図 3: 代数学的多色順序付けにおける係数行列 (3色)

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.  $\tilde{y}_{C(k)} (k = 1, \dots, ic - 1)$  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 (7) 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 4 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, 時間的な繰り返し計算により, 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

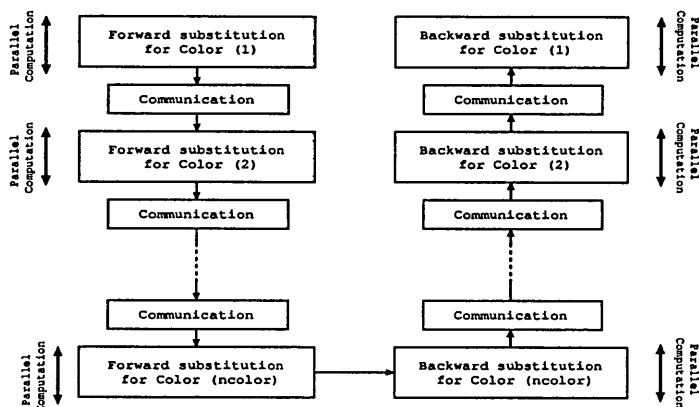


図 4: 代数学的多色順序付けにおける並列化代入計算

ルポテンシャルのみによる定式化を行う A-法を用いるので、支配方程式は次式で与えられる。

$$\nabla \times (\nu \nabla \times \mathbf{A}_m) = -\sigma \frac{\partial \mathbf{A}_m}{\partial t} + \mathbf{J}_0 \quad (8)$$

ここで、 $\mathbf{A}_m$  は磁気ベクトルポテンシャル、 $\mathbf{J}_0$  は強制電流の電流密度、 $\nu$  は磁気抵抗率、 $\sigma$  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 (8) にガラーキン法を適用することにより、次式が得られる。

$$[K]\{A_m\} + [M] \frac{\partial \{A_m\}}{\partial t} - \{J\} = 0 \quad (9)$$

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。

$$K_{ij} = \sum_e^m \iiint_e (\nabla \times \mathbf{N}_i) \cdot (\nu \nabla \times \mathbf{N}_j) dV \quad (10)$$

$$M_{ij} = \sum_e^m \iiint_e \sigma \mathbf{N}_i \cdot \mathbf{N}_j dV \quad (11)$$

$$J_i = \sum_e^m \iiint_e \mathbf{N}_i \cdot \mathbf{J}_0 dV \quad (12)$$

ここで、 $e$  は各要素、 $m$  は全要素数、 $\mathbf{N}$  はベクトル補間関数を表す。未知変数の総数を  $n$  とし、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 (9) 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。

$$\frac{\partial \{A_m\}}{\partial t} = \frac{\{A_m\} - \{A_{mold}\}}{\Delta t} \quad (13)$$

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

式 (9)(13) 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。

$$[Q]\{A_m\} = \{b\} \quad (14)$$

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

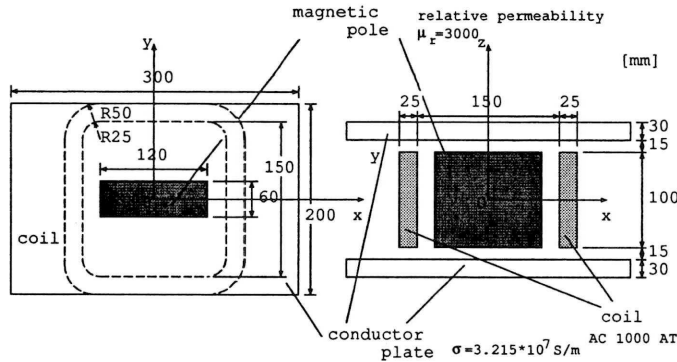


図 5: 電気学会 3 次元渦電流解析検証モデル (unit: [mm])

但し,

$$[Q] = ([K] + \frac{1}{\Delta t}[M]), \quad \{b\} = \frac{1}{\Delta t}[M]\{A_{old}\} + \{J\} \quad (15)$$

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, 図 5 に示す電気学会 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. 解析は京都大学学術情報メディアセンター 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 1 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 6 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 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



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

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

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（通常の多色順序付け法）で推奨されている 30～100 色という値は有限要素解析（非構造型解析）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 7 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。

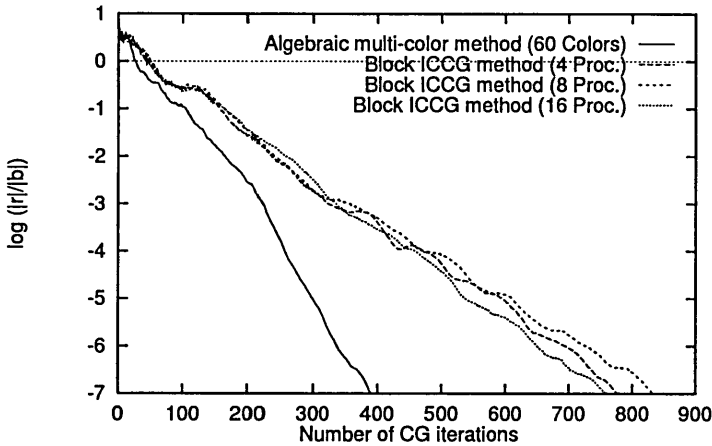


図 6: 収束の比較

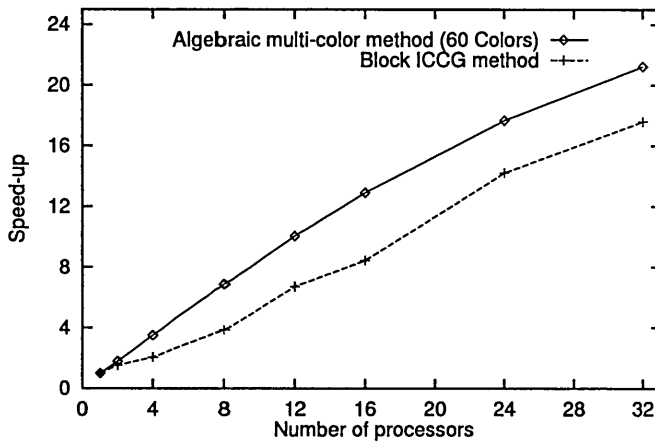


図 7: 代数学的多色順序付け法による速度向上

## 4 おわりに

本稿では、ブラックボックス型の並列化 ICCG 法として 2 種類の手法を紹介した。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。

(追記) 著者らの研究グループでは、開発した並列化アルゴリズム等を元に前処理付き反復解法ライブラリの作成を行っている。例えば、ここで紹介したブロック ICCG 法のプログラムなどを <http://fem.kuee.kyoto-u.ac.jp/~take/parasolver/piccg/index.html> において公開している。代数学的多色順序付け法によるソルバもなんらかの形で公開できるように検討を進めている。また、非対称係数行列向けの並列化 ILU-BiCGSTAB ソルバの開発 [15]、差分解析における新たな並列化オーダリングの提案 [16] とその非構造型への拡張を行っている [17]。ご興味のある方は引用論文を見ていただくか、直接著者に問い合わせ願いたい。

## 参考文献

- [1] R. Barrett, et al., "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods," SIAM, (1994).
- [2] 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).
- [3] 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*, vol. 31, (1977), pp. 148-162.
- [4] H. A. van der Vorst and T. F. Chan, "Parallel Preconditioning for Sparse Linear Equations," *ZAMM. Z. angew. Math. Mech.*, vol. 76, (1996), pp.167-170.
- [5] I. S. Duff and H. A. van der Vorst, "Developments and Trend in the Parallel Solution of Linear Systems," *Parallel Computing*, vol. 25, (1999), pp. 1931-1970.
- [6] 岩下 武史, 金澤 正憲, 島崎 眞昭, 「有限要素解析用並列化 ICCG ソルバの開発について」, 九州大学情報基盤センター 年報, ISSN 1346-9010, (2002), pp. 55-62.
- [7] M. L. Barton, "Three-Dimensional Magnetic Field Computation on a Distributed Memory Parallel Processor," *IEEE Transactions on Magnetics*, vol. 26, no. 2, (1990), pp.834-836.
- [8] G. Radicati di Brozolo and Y. Robert, "Parallel Conjugate Gradient-like Algorithms for Solving Sparse Nonsymmetric Linear Systems on a Vector Multiprocessor," *Parallel Computing*, vol. 11, (1989), pp. 223-239.
- [9] 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*, vol. 36, (2000), pp. 1504-1509.

- [10] T. Iwashita and M. Shimasaki, "Algebraic Multi-Color Ordering for Parallelized ICCG Solver in Finite Element Analyses," *IEEE Transactions on Magnetics*, (2002), to appear.
- [11] 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.
- [12] 襲田 勉, 丸山 訓英, 鷺尾 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.
- [13] 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.
- [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 Transactions on Magnetics*, vol. 27, (1991), pp.4073-4076.
- [15] 田見 智宏, 岩下 武史, 金澤 正憲, 「共有メモリ型並列計算機向け並列化 ILU-BiCGSTAB ソルバの開発」, 京都大学大型計算機センター研究開発部 研究発表報告集 第 17 号, ISSN 0912-2442, (2002), pp. 57-62.
- [16] 岩下 武史, 島崎 眞昭, 「同期点の少ない並列化 ICCG 法のためのブロック化赤-黒順序付け」, *情報処理学会論文誌*, vol. 43, no. 4, (2002), pp. 893-904.
- [17] T. Iwashita and M. Shimasaki, "Algebraic Block Red-Black Ordering Method for Parallelized ICCG Solver with Fast Convergence and Low Communication Costs", *Proc. IEEE CEFC'2002*, to appear.
- [18] I. S. Duff and G. A. Meurant, "The Effect of Ordering on Preconditioned Conjugate Gradients, BIT, vol. 29, (1989), pp.635-657.
- [19] S. Doi and A. Lichnewsky, "A Graph-Theory Approach for Analyzing the Effects of Ordering on ILU Preconditioning," *INRIA report 1452*, (1991).
- [20] S. Doi and T. Washio, "Ordering Strategies and Related Techniques to Overcome the Trade-off Between Parallelism and Convergence in Incomplete Factorization", *Parallel Computing*, vol. 25, (1999), pp. 1995-2014.
- [21] T. Washio and K. Hayami, "Overlapped Multicolor MILU Preconditioning," *SIAM Journal of Scientific Computing*, vol. 16, (1995), pp. 636-650.