

代数ブロック化多色順序付け法による 並列化ICCG ソルバの性能評価

岩下 武史^{†1} 高橋 康人^{†2} 中島 浩^{†1}

本論文では、ランダムスパース係数行列を対象とした並列化 ICCG ソルバについて述べる。差分解析において提案されたブロック化多色順序付け法をランダムスパース係数行列用に拡張した代数ブロック化多色順序付け法を提案する。代入計算におけるキャッシュヒット率の向上と ICCG 法の収束性改善を目的とするブロック化と色付けの方法を提示する。提案手法により得られたソルバを 3 種の数値例で検証した。その結果、Fujitsu HX600 1 ノード (16 コア) 使用時において従来法である代数多色順序付け法と比較して、計算時間を約半分に短縮することに成功した。

Performance Evaluation of Parallel ICCG Solver based on Algebraic Block Multi-color Ordering Method

TAKESHI IWASHITA,^{†1} YASUHIITO TAKAHASHI^{†2}
and HIROSHI NAKASHIMA^{†1}

This paper deals with parallel ICCG solvers for a linear system of equations having a random sparse coefficient matrix. We propose algebraic block multi-color ordering, which is an enhanced version of block multi-color ordering for a structured finite difference analysis. We present blocking and coloring procedures for improving cache hit ratio and convergence. The proposed method is examined in three numerical tests on a node of Fujitsu HX600, which show that the computational time is successfully reduced to half of that of the conventional algebraic multi-color ordering method.

^{†1} 京都大学学術情報メディアセンター

Academic Center for Computing and Media Studies, Kyoto University

^{†2} 京都大学大学院情報学研究所システム科学専攻

Graduate School of Informatics, Kyoto University

1. はじめに

有限要素解析や差分解析による数値シミュレーションでよく用いられる線形ソルバの一つに ICCG (Incomplete Cholesky Conjugate Gradient) 法¹⁾がある。ICCG 法は共役勾配法に IC 分解前処理を施した手法である。同手法はデスクトップ PC 等を用いた数十万自由度までの中・小規模シミュレーションでは単体のソルバとしてよく用いられている他、領域分割法における部分領域の解法や IC 前処理と同様の手順をもつマルチグリッド法における IC スムーザという形で大規模シミュレーションの building block としても頻りに活用されている。従って、ICCG 法もしくは IC 前処理 (IC スムーザ) の並列化は重要であるが、その解法手順において並列化の阻害要因となる前進・後退代入計算を含んでいる。そこで、これまでに ICCG 法、特に代入計算を対象としてその並列化に関して様々な研究が行われている²⁾。本論文ではこれらの並列化手法のうち、並列オーダリング法と呼ばれる手法に注目する。

並列オーダリング法は、未知変数や解析領域内で未知変数が置かれる節点を並列計算に適した形式に並び替える手法である。未知変数の並び替えによって得られた連立一次方程式に対して通常の ICCG 法の手順が適用される。並列オーダリングによる代入計算の並列化に関する研究では、差分解析における節点オーダリングの研究が早くから行われている。代表的な節点オーダリングとして、赤 - 黒順序付け法、多色順序付け法、Dissection 順序付け法、領域分割型オーダリング法などが知られている³⁾。こうした背景の下、著者らは文献 [4][5] において、ブロック化赤 - 黒順序付け法、ブロック化多色順序付け法と呼ぶ方法を提案している。これらの手法は多色順序付け法を基礎に、節点をブロック化することにより収束性の改善、スカラプロセッサにおけるキャッシュヒット率の向上を狙った方法で、未知変数間の関係が Stencil で表現される差分解析では 2 色を用いるブロック化赤 - 黒順序付け法が最も有効性が高いことを示した。

上記の節点オーダリングに関する研究に対して、これらの節点オーダリングを拡張する形で、非構造型解析におけるランダムスパース行列を係数とした場合の未知変数のオーダリングに関する研究が行われている。実用上の解析では、差分解析よりもこれらの非構造型解析の方がむしろ多く用いられており、重要な課題であるといってよい。例えば、多色順序付け法の色分けを代数的に行う Jones, Plassmann の方法⁶⁾や著者らの代数多色順序付け法⁷⁾、領域分割型オーダリングのランダムスパース行列向き拡張といえる Henon, Saad らの PHID 法がある。また、著者らはブロック化赤 - 黒順序付け法をランダムスパース行列

用に拡張した代数ブロック化赤 - 黒順序付け法と呼ぶ手法をこれまでに提案している⁸⁾。しかしながら、同手法は一般的に並列性の小さい計算環境でないと有効性が低いという問題点があった。これは、代数ブロック化赤 - 黒順序付け法では色数を 2 と限定していたために、未知変数のブロック化に制約が生じていたことに起因している。井上、染原、藤野は文献 [9][10] において代数ブロック化赤 - 黒順序付け法における本問題を論じ、マルチブロック法と呼ぶ手法を提案している。一方、本論文では色数の制限のないブロック化多色順序付け法を一般の係数行列に対して適用可能とした代数ブロック化多色順序付け法を新たに提案する。同手法を従来の提案手法である代数多色順序付け法と比較し、3 種の数値解析例においてキャッシュヒット率の改善等の効果に伴いより良好な並列台数効果が得られることを示す。

2. 対象とする問題と ICCG 法

本論文では、次式で表される実対称な係数行列を有する n 元連立一次方程式を対象とする。

$$Ax = b \quad (1)$$

ここで、係数行列 A が疎行列で、対称正定値である場合の連立一次方程式の解法として最も一般的なものに ICCG 法がある。ICCG 法は前処理付き共役勾配法の一つで、前処理行列として係数行列 A の不完全コレスキー分解行列を用いる。本論文では、この ICCG 法（より正確には ICCG(0) 法）の並列化について議論の対象とする。ICCG 法の反復過程は、前処理部である前進・後退代入計算と共役勾配ソルバ部である内積計算、行列・ベクトル積計算、ベクトル更新により構成される。これらの計算部分において、共役勾配ソルバ部については、係数行列、各種ベクトルを分割することにより比較的容易に並列化できる。一方、反復中の前進・後退代入計算は、一般的に逐次計算であるためにその並列化が困難である。また、本代入計算はマルチグリッド法における IC, ILU スムーザやガウスザイデル法, SOR 法にも共通な計算手順であり、その並列化は重要である。そこで、本稿では代入計算の並列化を中心に述べることにする。

3. 代数ブロック化多色順序付け法 (ABMC 法)

3.1 ブロック化多色順序付け法

本稿で提案する代数ブロック化多色順序付け法（以下、ABMC: Algebraic Block Multi-Color ordering 法）は著者らが以前に提示したブロック化多色順序付け (BMC: Block Multi-Color ordering) 法を一般のランダムスパース係数行列をもつ連立一次方程式に適用可能な形に拡張した方法である。そこで、差分解析における節点のオーダリングとして提案

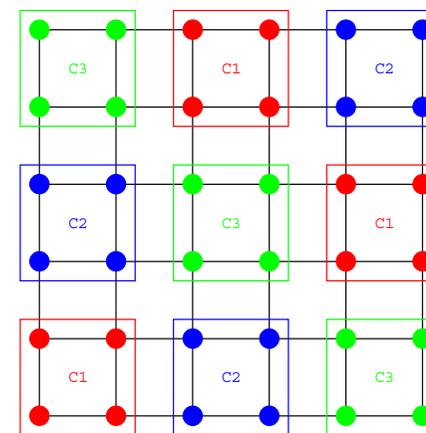


図 1 ブロック化多色順序付け
 Fig. 1 Block multi-color ordering

された BMC 法について簡単に述べる。同手法では、まずいくつかの節点をブロック化し、これらのブロックに対して多色順序付け法を適用する。このとき、同色の異なるブロック内の節点間にデータ依存関係が生じないようにする。以上の操作により、ICCG 法の前進・後退代入計算は各色内においてブロック毎に並列化可能となる。ただし、ブロック内の計算は逐次的に行われる。図 1 は 2 次元差分格子におけるブロック化多色順序付け（ブロックサイズ：4、色数：3）の例である。多色順序付け法と比較した場合、ブロック化により以下のような効果が得られる。効果 1: ブロック内の節点は近接しているために、代入計算におけるキャッシュヒット率の向上が得られる（多色順序付け法の場合、並列処理される同色の節点は近接していない。）効果 2: 節点をブロック化することにより ICCG 法の収束性が改善できる。これらの効果については、文献 [4] において数値実験により確認されている。また、差分解析における ICCG 法の収束性はある程度解析的に分析が可能で、土肥らが提案している Incompatibility ratio¹¹⁾ や著者らが提案した S.R.I.¹²⁾ といった指標からも効果 2 について検証されている。本稿で提案する ABMC 法では、未知変数のブロック化と各ブロックの色付けを与えられた係数行列の情報のみによって行う。次小節以降にその詳細を示す。

3.2 代数ブロック化多色順序付け (ABMC) 法

3.2.1 ブロック化の方法

ここでは、 n 個の未知変数をブロックサイズ $n_b > 1$ の複数のブロックにわけるとを考

える．簡単のために未知変数はブロックサイズで割り切れるとする．未知変数のブロック化には多数の方法が考えられる．最も単純な例は，元の未知変数の順序に従って，上から順にブロックに割り当てて方法である．即ち， i 番目のブロックは $n_b \cdot (i - 1) + 1$ から $n_b \cdot i$ 番目の未知変数により構成される．ABMC 法では，差分解析における BMC 法の長所である，ブロック化によるキャッシュヒット率の改善や IC 分解前処理効果の向上を目的としている．そこで，BMC 法と同様に，なるべく互いに関連性のある未知変数をブロック化することを考える．ここで，未知変数間の関連性は， i 番目の未知変数と j 番目の未知変数を考えた場合， a_{ij} が非零要素かどうかで決まる．即ち，係数行列の i 行 j 列要素 a_{ij} が非零の場合， i 番目の未知変数と j 番目の未知変数の間に関連性があるとする．具体的なブロック化アルゴリズムを以下に示す．今， i_b 番目のブロックに割り当てられる未知変数の選択を行っているとする．

Step 1. 未だブロックに割り当てられていない未知変数のうち最も番号の小さい未知変数 i_o を選択し，ブロック i_b に入れる．

Step 2. 未知変数 i_o と関連する未知変数のうち未だブロックに割り当てられていない未知変数の集合を $j_1, j_2, j_3, \dots, j_m$ とする．これらの未知変数を順にブロック化候補スタックに入れる．

Step 3. ブロック化候補スタックの先頭の未知変数 j_s をブロック i_b に入れ，当該未知変数をスタックから消去する．ここで，この操作によりブロックに n_b 個の未知変数が割り当てられた場合には，ブロック化候補スタックをクリアし，次ブロックの割り当てのために Step 1 にもどる．ブロック内の未知変数が n_b に満たない場合には， j_s に関連のある未知変数のうち未だブロックに割り当てられていないものをブロック化候補スタックの最後尾に追加する．

Step 4. ブロック化候補スタックが空の場合には Step 1 にもどる．それ以外の場合には Step 3 にもどる．

上記の手順を i_b が 1 から n/n_b となるまで繰り返す．これらの操作により，すべての未知変数がいずれかのブロックに割り当てられる．ここで以降の議論のために， i_b 番のブロックに割り当てられた未知変数の集合を S_{i_b} と表すものとする．

3.2.2 ブロックに対する色付けの方法

前節のブロック化の後，ブロック間の依存関係を表す $n/n_b \times n/n_b$ の行列 M_b を作成する．ここで， M_b の i 行 j 列要素は以下のように与えられる．

$$M_{bij} = \begin{cases} 1 & \text{if } \exists k \in S_i, \exists l \in S_j \text{ s.t. } a_{kl} \neq 0 \vee a_{lk} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

各ブロックに対する色づけはこの行列 M_b を係数とする連立一次方程式の未知変数に対して多色順序付けを適用することで行うことができる．ここで，多色順序付けを代数的に行う方法としては Jones らの方法⁶⁾，文献 [2] に記載の Greedy Algorithm 等があるが，ここでは著者らが提案している AMC 法⁷⁾を用いる．AMC 法は Jones らの手法と異なり，色数を外部的に指定することができる．AMC 法の詳細は文献 [7] に述べられているが以下に簡単に述べる．まず，行列 M_b は元の係数行列に関わらず対称行列となる．そこで，対称行列用の AMC 法を使用することができる．まずはじめにソルバの外部パラメータとして色数 c_s を設定する．ここで，与えられた色数が色付けの過程で不十分な場合には，色数を増加させ再度色付け操作を実行する．但し，AMC 法のアルゴリズムでは，行列 M_b の下三角部分行列 L_b の 1 行あたりの非零要素数の最大値 l_m の色数で色分けが可能となる．そこで，予め c_s と l_m の両者を比較し，その大きい方の値をソルバで使用できる色数 n_c とすれば，色分けの手順は一度で完了する．実際に開発したプログラムでは本手法により色付けの過程は一度だけ実行している．次に，未知変数に対して，元の順序に従って，順次 1 から n_c までの色をなるべく周期的に割り当てることとするが，その際に以下の条件が満たされるようにする．

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．

具体的には，以下の手順により 1 番目の未知変数から順次 1 から n_c までの色をつけていく．ここで， i 番目の未知変数に対する色付けは直前に付けられた色番号を $i_c - 1$ として，以下のように定める．

Step 1 未知変数 $j = \{j : L_{bij} \neq 0, j \neq i\}$ (未知変数 i と関連性をもち i 未満の番号をもつ未知変数) の色を調べ，使用されていない色番号の集合 S_c を作成する．

Step 2 $i_t = \min_{i_c \in S_c} \text{mod}(i_c - i_c + n_c, n_c)$ で定められる i_t に対して， $\text{mod}(i_t + i_c, n_c) + 1$ を色番号とする．

上記の手順を 1 から n/n_b の未知変数に適用することにより，各ブロックの色が定められる．

3.2.3 並列化代入計算

ブロック化と色付けが終了した後，色の順にブロックを並び替える．本研究では，ブロック内の未知変数の順序について元の順序付けに基づくものとしたが，なんらかの順序付け法を適用してもよい．これらの未知変数の再順序付けの結果，解くべき方程式 (1) は図 2 に示

すような係数行列を持つ方程式に変換される．本係数行列に対して，前進・後退代入計算は色毎にブロックを単位として並列化できる．即ち，各プロセッサ(コア)は，各色において1つまたは複数のブロックを担当する．ブロック内の計算は逐次計算として処理されるが，同色のブロック間にデータ依存関係がないため，ブロック単位で並列処理がなされる．以下に，前進代入計算を例にその詳細を示す．

再順序付けされた方程式を $\tilde{A}\tilde{x} = \tilde{b}$ と表記すると，係数行列と解ベクトルは以下のように色を単位として分割することができる．

$$\tilde{x}^T = (\tilde{x}_{C(1)}, \tilde{x}_{C(2)}, \dots, \tilde{x}_{C(n_c)})^T, \quad (3)$$

$$\tilde{A} = \begin{pmatrix} \tilde{C}_{1,1} & \tilde{C}_{1,2} & \dots & \tilde{C}_{1,n_c} \\ \tilde{C}_{2,1} & \tilde{C}_{2,2} & \dots & \tilde{C}_{2,n_c} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_{n_c,1} & \tilde{C}_{n_c,2} & \dots & \tilde{C}_{n_c,n_c} \end{pmatrix} \quad (4)$$

ここで， $C(i)$ は i 番目の色を表す． i 番目の色に割り当てられたブロックの個数を nb_{c_i} と表記するものとする．ABMC法により，係数行列 \tilde{A} の対角ブロック $\tilde{C}_{i,i}$ は，次式のようなブロック対角行列となる．

$$\tilde{C}_{i,i} = \begin{pmatrix} \tilde{B}_{i(1)} & & & \mathbf{O} \\ & \tilde{B}_{i(2)} & & \\ & & \ddots & \\ \mathbf{O} & & & \tilde{B}_{i(nb_{c_i})} \end{pmatrix} \quad (5)$$

ICCG法では，係数行列 \tilde{A} を次式のように不完全コレスキー分解して得られる行列を前処理行列として使用する．

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

$$\tilde{L} = \begin{pmatrix} \tilde{L}_{1,1} & & & \mathbf{O} \\ \tilde{L}_{2,1} & \tilde{L}_{2,2} & & \\ \vdots & \ddots & \ddots & \\ \tilde{L}_{n_c,1} & \tilde{L}_{n_c,2} & \dots & \tilde{L}_{n_c} \end{pmatrix}, \quad (7)$$

$$\tilde{D} = \begin{pmatrix} \tilde{D}_1 & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \tilde{D}_{n_c} \end{pmatrix} \quad (8)$$

ここで， \tilde{L} ， \tilde{D} はそれぞれ下三角行列，対角行列で，行列 \tilde{L} の対角成分を $diag(\tilde{L})$ と表すと，

$$diag(\tilde{L}) = \tilde{D}^{-1} \quad (9)$$

の関係がある．また，式(5)より \tilde{L} の対角ブロック成分 $\tilde{L}_{i,i}$ は次式のように書ける．

$$\tilde{L}_{i,i} = \begin{pmatrix} \tilde{L}_{B_{i(1)}} & & & \mathbf{O} \\ & \tilde{L}_{B_{i(2)}} & & \\ & & \ddots & \\ \mathbf{O} & & & \tilde{L}_{B_{i(nb_{c_i})}} \end{pmatrix} \quad (10)$$

ICCG法の反復中の前処理部における前進代入計算は，並び替えられた方程式に対して，

$$\tilde{L}\tilde{y} = \tilde{r} \quad (11)$$

のように与えられる．ここで， \tilde{r} は残差ベクトルである． i 番目の色に対応する \tilde{y} ， \tilde{r} の部分ベクトルをそれぞれ $\tilde{y}_{C(i)}$ ， $\tilde{r}_{C(i)}$ で表すと，これらのベクトルはブロックを単位として

$$\tilde{y}_{C(i)} = \begin{pmatrix} \tilde{y}_{b_{i(1)}} \\ \tilde{y}_{b_{i(2)}} \\ \vdots \\ \tilde{y}_{b_{i(nb_{c_i})}} \end{pmatrix}, \tilde{r}_{C(i)} = \begin{pmatrix} \tilde{r}_{b_{i(1)}} \\ \tilde{r}_{b_{i(2)}} \\ \vdots \\ \tilde{r}_{b_{i(nb_{c_i})}} \end{pmatrix} \quad (12)$$

のように書ける．ここで， $\tilde{y}_{C(i)}$ を計算する前進代入計算は式(7)(11)より，

$$\tilde{y}_{C(i)} = \tilde{L}_{i,i}^{-1} \tilde{d}_{C(i)} \quad (13)$$

$$\tilde{d}_{C(i)} = \tilde{r}_{C(i)} - \sum_{k=1}^{i-1} \tilde{L}_{i,k} \tilde{y}_{C(k)} \quad (14)$$

で与えられる．ここで， $\tilde{d}_{C(i)}$ は $i-1$ までの色に対応する \tilde{y} の部分ベクトルから計算され，式(13)の前進代入計算において既知である．そこで，ベクトル $\tilde{d}_{C(i)}$ を式(12)と同様に

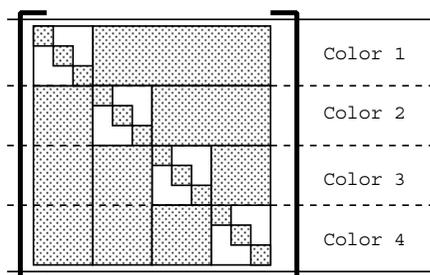


図 2 代数ブロック化多色順序付け法における係数行列

Fig. 2 Coefficient matrix form arising in algebraic block multi-color ordering method

ブロック毎に分割し、その各部分ベクトルを式 (12) と対応付ける形で $\tilde{d}_{b_i(m)}$ と表すと、式 (10) より、式 (13) は

$$\tilde{y}_{b_i(m)} = \tilde{L}_{Bi(m)}^{-1} \tilde{d}_{b_i(m)} \quad (m = 1, \dots, nbc_i) \quad (15)$$

のように書ける。式 (15) 中の $\tilde{y}_{b_i(m)}$ の計算において、同色内の他のブロックの未知変数に対する部分ベクトル $\tilde{d}_{b_i(l)}$, $l \neq m$ の値は必要とされないため、本代入計算 (15) はブロック毎に並列化が可能となり、その並列度は同色内のブロック数 nbc_i で与えられる。また、後退代入計算においても同様の手順により、同色内のブロック毎に並列計算が可能となる。

4. 数値実験結果

4.1 使用計算環境とテスト問題

提案手法を評価するために、数値実験を行った。数値実験に使用する計算機は、京都大学学術情報メディアセンターの富士通 HX600 である。HX600 の各ノードは 4 個の AMD 社製クアドコア Opteron プロセッサと 32GB (DDR2-667) のメモリを有している。当該計算機の 1 ノードを使用して数値実験を行った。プログラミング言語として FORTRAN90 を使用し、並列処理の API として OpenMP を使用した。コンパイラの最適化オプションには -Kfast、Opteron を指定した。ICCG 法の収束判定基準は相対残差ノルムが 10^{-7} 以下となる条件とし、反復開始から残差ノルムが本基準を満たすまでの経過時間を計測した。本数値実験には、著者らが開発した有限要素電磁場解析プログラムにおいて生ずる連立一次方程式 1 種と The University of Florida Sparse Matrix Collection (<http://www.cise.ufl.edu/research/sparse/matrices/>) より入手した連立一次方程式 2 種を用いる。なお、UF matrix collection における係数行

列の選定において、正値対称行列であることに加え、係数行列と右辺ベクトルの両者が提示されていること、非構造型の解析であること、比較的大きな次元数をもつ係数行列であることの 3 点を考慮した。

4.2 電磁場解析テスト問題

本小節では 3 次元静磁場有限要素解析において生ずる連立一次方程式を対象とする。解析対象モデルは、電気学会のベンチマークモデルとして利用され、文献 [13] にその詳細が示されているボックスシールドモデルである。解析対象内の電磁界を記述する方程式は、以下のように与えられる。

$$\nabla \times (\nu \nabla \times \mathbf{A}_m) = \mathbf{J}_0 \quad (16)$$

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、 ν は磁気抵抗率を表す。本解析では、式 (16) を 6 面体辺要素による辺要素有限要素法により解く。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 (16) にガラーキン法を適用することにより、テスト問題となる連立一次方程式が得られる。辺要素を用いた電磁場解析では、係数行列は正値性を失っている場合がほとんどあり、IC(0) 分解前処理をそのまま用いることができず、シフト付き ICCG 法が用いられる。そこで、本解析ではシフト量として係数行列の対角要素に 0.1 を乗じた値による対角行列を用いる。本解析モデルでは要素数を 253,500 とし、その結果、対象とする連立一次方程式の係数行列の次元数は 786,050、非零要素数は 22,345,344 となっている。

表 1 に、逐次 ICCG 法、代数多色順序付け (AMC) 法、ABMC 法の各手法における計算時間、反復回数、1 反復あたりの計算時間、速度向上を示す。なお、以下の記述において m 色による多色順序付け法を AMC(m) と表記し、ブロックサイズを n_b 、色数を m とした場合の代数ブロック化多色順序付け法を ABMC(m, n_b) と表記する。速度向上については、逐次型 ICCG 法の計算時間に対する比率で表すものとする。また、数値実験結果において使用コア数を 2, 4, 8, 12 とした場合の使用ソケット (プロセッサ) 数はそれぞれ、2, 4, 4, 4 である。まず、従来法である AMC 法について考察する。多色順序付け法では、差分解析を対象とした研究において、色数を増加させるに従って反復回数が減少するが、並列度も低下することが指摘されている。また、色数の増加に対して並列計算における粒度が減少する。表 1(b), (c) によると、これらの定性的な傾向がランダムスパース係数行列を対象とする本数値実験においても同様に確認される。即ち、収束性を向上させるために色数を増加させた場合、並列計算における粒度が低下するために 1 反復あたりの計算時間が増加し、台数効果の向上が得られず、比較的小さい色数である 30 色の場合の方が良い結果を得ている。

次に、提案手法である ABMC 法について考察する。これまでに述べたように ABMC 法ではブロック化を行うことにより色数を増加させる（粒度を低下させる）ことなく、収束性の改善を行う。表 1(d), (e) において、ブロック化を行うことにより AMC(30) と比べて収束性が向上していることが分かる。また、1 反復あたりの計算時間を AMC(30) と比較した場合、ABMC(30, 64), ABMC(30, 512) のいずれの場合も短縮されている。これは代入計算におけるキャッシュのヒット率が向上したことによると考えられる。次に、ABMC(30, 64), ABMC(30, 512) の両者を比較した場合、ブロックサイズの大きい ABMC(30, 512) が反復回数、計算時間のいずれの点でも優っている。ブロックサイズの増加は一般的に収束性の向上に寄与し、代入計算におけるキャッシュヒット率についても悪影響を及ぼすことは少ないと考えられるため、ブロックサイズはなるべく大きくとることが望ましいと考え得る。しかしながら、ブロックサイズの増加は並列度の低下を招く他、ランダムスパース係数行列を対象とする場合には、逆に収束性が低下する可能性がある。そこで、ブロックサイズの推奨値として、現時点で著者らは、128~512 程度のブロックサイズを基準に、各コアが少なくとも数個程度のブロックを担当するようなブロックサイズが適当であると考えている。

4.3 UF matrix collection テスト問題

本小節に The University of Florida Sparse Matrix Collection より取得した parabolic_fem, 及び thermal2 の 2 種類の係数行列、右辺ベクトルを使用した数値実験結果を示す。表 2 に実験で使用したデータの諸元を示す。表 3, 4 にこれらの係数行列データによる数値実験結果を示す。なお、紙面の都合から本実験結果は逐次解析結果と 16 コア使用時（ノード内の全コア使用時）の結果のみを示す。表 3 によると、parabolic_fem による数値実験では、リオーダーリングしても収束性にあまり変化がでない結果となっている。そのために、AMC 法を用いた場合、色数を増加してもよい効果は得られず、粒度とキャッシュヒット率の低下からソルバの性能は下がっている。次に、ABMC 法を用いた場合について考察する。本手法では 30 色を指定したが、色付けの過程で色数が十分ではなく、ブロックサイズ 64 の場合に 193 色、ブロックサイズ 128 の場合に 171 色が使用されている。いずれのブロックサイズを用いた場合でも AMC 法と比べて 1 反復あたりの計算時間が短縮され、提案手法の有効性が確認される。

次に、表 4 に示される係数行列データ thermal2 を用いた数値実験結果について考察する。本実験においても、parabolic_fem による数値実験と同様にリオーダーリングに対する収束性の変化が鋭敏ではなかった。AMC 法を用いた場合には、キャッシュヒット率の低下から反復あたりの計算時間が増え、色数を増やすことによる性能改善効果はみられなかった。

表 1 計算時間, 反復回数, 反復あたりの計算時間, 速度向上
 Table 1 Computation time, Iteration, Computation time per iteration, Speed-up

(a) 逐次型 ICCG 法 (1CPU) の実行結果

計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)
232	577	0.403

(b) AMC(30) による実行結果

コア数	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)	速度向上
2	501	1366	0.367	0.463
4	257	1356	0.189	0.904
8	140	1372	0.102	1.66
12	103	1349	0.0762	2.26
16	92.7	1356	0.0683	2.51

(c) AMC(500) による実行結果

コア数	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)	速度向上
2	501	1044	0.480	0.463
4	250	1049	0.238	0.928
8	155	1051	0.148	1.49
12	139	1051	0.132	1.68
16	129	1046	0.123	1.80

(d) BAMC(30, 64) 法による実行結果

コア数	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)	速度向上
2	307	1148	0.268	0.756
4	161	1158	0.139	1.44
8	89.4	1160	0.0770	2.60
12	70.5	1161	0.0607	3.30
16	65.4	1190	0.0550	3.550

(e) BAMC(30, 512) 法による実行結果

コア数	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)	速度向上
2	214	920	0.233	1.08
4	113	925	0.123	2.04
8	64.9	921	0.0705	3.58
12	51.1	914	0.0559	4.55
16	47.1	926	0.0508	4.93

表 2 UF matrix collection テスト問題の諸元

Table 2 Properties of test problem matrices from the UF matrix collection

係数行列名	問題領域	次元数	非ゼロ要素数
parabolic_fem	計算流体力学	525,825	3,674,625
thermal2	定常熱問題	1,228,045	8,580,313

表 3 係数行列データ parabolic_fem による数値実験結果

Table 3 Numerical results of parabolic_rem test problem

(a) 逐次型 ICCG 法 (1CPU) の実行結果

計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)
101	1178	0.0857

(b) 並列化 ICCG ソルバによる実行結果

並列化手法	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)
AMC(30)	37.0	1163	0.0317
AMC(500)	40.9	1157	0.0354
ABMC(30,64)	20.6	1137	0.0181
ABMC(30,128)	20.0	1158	0.0172

表 4 係数行列データ thermal2 による数値実験結果

Table 4 Numerical results of thermal2 test problem

(a) 逐次型 ICCG 法 (1CPU) の実行結果

計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)
483	2122	0.227

(b) 並列化 ICCG ソルバによる実行結果

並列化手法	計算時間 (sec)	反復回数	1 反復あたりの計算時間 (sec)
AMC(30)	146	2107	0.0690
AMC(90)	152	2103	0.0722
AMC(500)	157	2114	0.0745
ABMC(30,64)	71.7	2082	0.0345
ABMC(30,128)	71.5	2087	0.0342
ABMC(30,512)	66.9	2097	0.0319

参 考 文 献

- 1) Meijerink, J. and van der Vorst, H. A.: An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix Is a Symmetric M-matrix, *Mathematics of Computation*, Vol.31, pp.148-162 (1977).
- 2) Y. Saad, "Iterative Methods for Sparse Linear Systems", Second ed., SIAM, Philadelphia, PA, 2003.
- 3) Duff, I. S. and Meurant, G. A.: The Effect of Ordering on Preconditioned Conjugate Gradients, *BIT*, Vol.29, pp.635-657 (1989).
- 4) 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 - 」, 情報処理学会研究会報告集 ハイパフォーマンスコンピューティング, HPC-85, pp. 55-60, (2001).
- 5) Takeshi Iwashita and Masaaki Shimasaki : "Block Red-Black Ordering: A New Ordering Strategy for Parallelization of ICCG Method," *International Journal of Parallel Programming*, Vol. 31, No. 1, pp. 55-75, (2003).
- 6) M.T. Jones and P.E. Plassmann : "The efficient parallel iterative solution of large sparse linear systems", in *Graph Theory and Sparse Matrix Computations*, A. George, J. R. Gilbert and J. W. H. Liu, eds., IMA 56, Springer, Berlin, 1994, pp.229-245.
- 7) Takeshi Iwashita and Masaaki Shimasaki : "Algebraic Multi-Color Ordering for Parallelized ICCG Solver in Finite Element Analyses", *IEEE Transaction on Magnetics*, Vol. 38-2, (2002), pp. 429-432.
- 8) Takeshi Iwashita and Masaaki Shimasaki : "Algebraic Block Red-Black Ordering Method for Parallelized ICCG Solver With Fast Convergence and Low Communication Costs",

一方, 提案手法である ABMC 法では, parabolic_fem による実験結果と同様に AMC 法と比べてキャッシュ利用効率が改善されることにより反復あたりの計算時間が減少し, 求解に要する計算時間が削減されている.

5. おわりに

本論文では, 差分解析において提案されたブロック化多色順序付け法を一般のランダムスパース行列に向けて拡張した代数ブロック化多色順序付け法を提案した. 同手法により, 同色内の各ブロックは互いに独立に計算可能となり, ICCG 法の代入計算の並列化が可能となる. 並列代入計算をなるべく効率よく行うために, 互いに関連性のある未知変数が同一のブロック内に割り当てられるように考慮したブロック化アルゴリズムを提示した. 提案手法を 3 種のテスト問題により評価し, いずれの実験例においても従来の代数多色順序付け法と比べて 2 倍程度高速な求解性能を示した.

本研究の一部は, 日本学術振興会 科学研究費補助金 (基盤研究 (B) , 課題番号 20300011) の助成を受けている.

IEEE Transaction on Magnetics, Vol. 39-3, (2003), pp. 1713-1716.

- 9) 井上明彦, 藤野清次: フィルインの選択に基づく改良版 ABRB 順序付け法による ICCG 法の並列化, 情報処理学会論文誌: コンピューティングシステム, Vol. 46, No. SIG16(ACS12), pp. 119-127, (2005).
 - 10) 染原一仁, 藤野清次: 代数マルチブロック技法による ICCG 法の並列性能の向上, 情報処理学会論文誌: コンピューティングシステム, Vol. 47, No. SIG18(ACS16), pp. 21-30, (2006).
 - 11) Doi, S. and Lichnewsky, A.: A Graph-Theory Approach for Analyzing the Effects of Ordering on ILU Preconditioning, *INRIA report* 1452, (1991).
 - 12) Iwashita, T., Nakanishi, Y. and Shimasaki, M.: Comparison Criteria for Parallel Orderings in ILU Preconditioning, *SIAM J. Sci. Comput.*, Vol.26, No.4, pp.1234-1260 (2005).
 - 13) 実規模電磁界解析のための数値計算技術調査専門委員会: 実規模電磁界解析のための数値計算技術, 電気学会技術報告 No. 1129 (2008).
-