

HPC2500 による興奮伝播シミュレーションにおける 並列化 ICCG ソルバの性能評価

荒井 宗範,[†] 岩下 武史,^{††} 金澤 正憲^{††}

本稿では、FitzHugh-Nagumo model を用いた心臓の興奮伝播シミュレーションの高速化について述べる。このような興奮伝播シミュレーションでは細胞膜内の電位のみを考慮するモノドメインモデル、細胞膜内外の電位を独立させて考えるバイドメインの2種の方法が存在するが、いずれの場合においても最終的に大規模な連立一次方程式を解くことに帰着される。今回はモノドメインモデルを用いたシミュレーションにおいて、並列化 ICCG 法 (Incomplete Cholesky Conjugate Gradient method) を利用することで高速化を図る。ICCG 法の並列化の阻害要因である前進・後退代入計算に対し2つの手法の有効性を検討した。

Performance evaluation of fast parallel ICCG solvers for simulation of excitement propagation with HPC 2500

MUNENORI ARAI,[†] TAKESHI IWASHITA ^{††} and MASANORI KANAZAWA^{††}

In this paper, the authors deal with a linear system of equations arising in an excitement propagation of the mono-domain using the FitzHugh models and describe accelerating the solution process by means of parallel processing techniques. Although our final objective is to develop a simulation code based on the bidomain method with a precise cell model Kyoto model, the monodomain can be used for evaluation of our parallel iterative solver since the linear system has a similar property as that of the bidomain method. We have focused on the forward and backward substitutions which spend so much time on calculation and evaluate parallel ICCG (Incomplete Cholesky Conjugate Gradient) solver with two parallel techniques in order to solve them.

1. はじめに

現在、著者らとその研究グループは高精細な心筋細胞モデルである『KYOTO model』¹⁾を用いた心臓のシミュレーションコード開発プロジェクトに参画している²⁾。このプロジェクトの目的は心臓の動きを細胞レベルからコンピュータ上に実現することである。心筋の興奮伝播シミュレーションを行う際にはバイドメインモデルとモノドメインモデルという2つのアプローチがある。バイドメインモデルは細胞膜内と細胞膜外の電位を独立させて考えるモデルであり、モノドメインモデルは細胞膜内の電位のみを考慮するモデルである。より高精細な興奮伝播のシミュレーションのためにはバイドメインモデルを用いることが望ましいが、その

ためには計算時間が掛かる大規模な連立一次方程式を解く必要がある。そこで本研究ではバイドメインモデルを用いた興奮伝播シミュレーションコード開発に先立ち、モノドメインモデルを用いた連立一次方程式を対象にその高速化に取り組んだ。モノドメインモデルで得られる係数行列はバイドメインモデルで得られるそれと構造・性質は似ているが計算時間は短くなるため、ソルバの開発を効率よく行うことができる。そこで、古くから用いられている細胞の興奮性・振動性を示す FHN(FitzHugh-Nagumo)model³⁾を用い、並列化された ICCG 法 (Incomplete Cholesky Conjugate Gradient method)⁴⁾ による連立一次方程式の求解の高速化を行った。ICCG 法では大規模な前進・後退代入計算を行う必要があるが、一般的にその計算は逐次計算となり並列化を妨げる要因となる。この前進・後退代入計算を並列計算するために、今回は2種の手法を用いた。興奮伝播シミュレーションにおいてこれらの手法による並列化効果の検証を行った。

[†] 京都大学大学院 情報学研究所
Graduate School of Informatics, Kyoto University
^{††} 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. 心筋モデル

2.1 心筋細胞モデル

今回用いた細胞モデルである FHN model³⁾ は、Hodgkin-Huxley の微分方程式系を 2 変数 1 次微分方程式系に簡略化したものである。同モデルは心筋細胞の電気的特性を表したモデルではないが、非線形性を有する興奮性あるいは振動性細胞の一般的な特性をよく記述できるモデルとして知られている。2 変数に単純化されているため、数学的扱いが容易で少ない計算機パワーでも大規模なシミュレーションが可能になる¹⁾。このモデルはいろいろな表記方法がなされているが、FitzHugh の原論文では以下のように定式化されている。

$$\frac{dx}{dt} = c(y + x - \frac{x^3}{3} + z) \quad (1)$$

$$\frac{dy}{dt} = -\frac{(x - a + by)}{c} \quad (2)$$

$$(1 - \frac{2b}{3} < a < 1, 0 < b < 1, b < c^2)$$

x は膜電位を、 y は不応性を表す変数である。 a, b, c は上記の条件を満たすパラメータ、 z は外部刺激になる。ここで、1 個の FHN model による数値シミュレーションを行った場合について述べる。1 個のモデルを用いて、定常状態から刺激 z を与えたときの時間変化に対する膜電位の動きを示したものが図 1 になる。横軸に時間 (t)、縦軸に膜電位 (x) をとっている。パラメータは以下の通り。

x の初期値: $x_0 = 1.200181$

y の初期値: $y_0 = -0.624039$

微小時間: $\Delta t = 1.0 \times 10^{-3}$

$a = 0.7, b = 0.8, c = 20.0, 30.0, 40.0$

$z = 30$ (一定, $t = 10.0 \sim 10.5$ の間)

図 1 をみると、ある時間に刺激を与えるとそれまで定常状態にあった膜電位が興奮を起こすことがわかる。興奮とは、細胞の膜電位が急激に立ち上がることを指す。刺激を与えるのをやめてもしばらくは興奮が持続したままである。ある一定時間たつと膜電位は急激に変化し元の定常状態に戻る。この細胞の特性は心電図などにも利用されている。また、この挙動はパラメータを変更することで変化する。例えば図中に示すように c を大きくすると活動電位持続時間が長くなる。今回の評価には $c = 20.0$ の場合を用いた。

2.2 心筋細胞の結合組織モデル

現在、心臓の興奮伝播シミュレーションでは、パイ

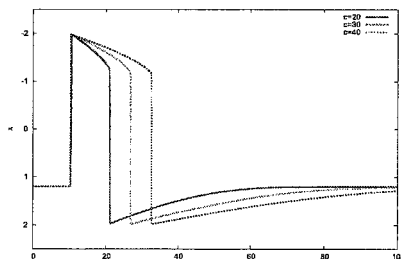


図 1 FHN-model の膜電位 (x) の時間変化

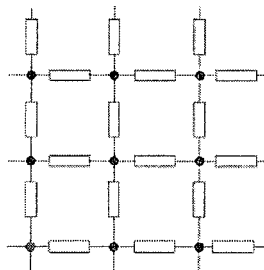


図 2 2次元電気回路

ドメインモデルを用いるのが一般的である。しかし、今回はモノドメインモデルを用いて計算を行った。それはバイドメインモデルに比べてコードを構築しやすいことや解くべき計算量が少なくて済むことから効率的に高速ソルバの開発に取り組みると考えたからである。モノドメインモデルは細胞の膜電位 V_m を $V_m = \phi_i - \phi_e = \phi_i - 0 = \phi$ とし、以下のように定式化される。

$$\frac{\partial V_m}{\partial t} = \frac{1}{\beta C_m} (\nabla \cdot \sigma \nabla \phi) - \frac{1}{C_m} (I_{ion} - I_{ext}) \quad (3)$$

ここで、 ϕ_i は細胞内電位、 ϕ_e は細胞外電位、 σ は細胞内空間の導電率、 I_{ion} はイオンチャネルを通る電流密度、 I_{ext} は刺激電流密度、 V_m は膜電位、 C_m は膜容量、 β は電流単位の変換定数、 ∇ は空間微分演算子をそれぞれ表す。また電場を 5 点差分で近似し、離散化を行うと、心筋細胞同士を抵抗のみで接続し、電気回路的に平面の仮想心筋を構成したものと等価になる (図 2)。図 2 において灰色の丸がひとつの細胞を、白い四角が抵抗を表す。ある時刻での細胞膜電位はそのとき隣接する細胞から流入する電流、隣接する細胞へと流出する電流を考慮し計算される。今回のシミュレーションにおける設定条件を示す。

メッシュサイズ: 10000×10000

横方向の抵抗値: $R_\alpha = 2.0$

縦方向の抵抗値: $R_\beta = 2.0$

以上のパラメータで、各細胞の膜電位を計算し、それに有する計算時間及び反復回数を計測した。

2.3 モノドメイン × FHN model

簡単のため一次元回路モデルを用いてモノドメインと FHN model を組み合わせる方法を説明する。式 (3) の I_{ion} は任意の細胞モデルによって決定する。今回は FHN model を適用した。式 (1) の x に細胞膜電位 V_i を代入すると

$$\frac{dV_i}{dt} = c(y + V_i - \frac{V_i^3}{3} + z) \quad (4)$$

となる。また抵抗 R_α で細胞同士がつながっているとすると式 (4) は式 (3) と対応させて、

$$\frac{dV_i}{dt} = c(I'_m + I'_{ion} + I_{ext}) \quad (5)$$

$$I'_{ion} = y_i + V_i - \frac{V_i^3}{3} \quad (6)$$

$$z = I'_m + I_{ext} \quad (7)$$

$$I'_m \simeq \frac{V_{i-1} - V_i}{R_\alpha} - \frac{V_i - V_{i+1}}{R_\alpha} \quad (8)$$

と解釈できる。以下、この考えのもとシミュレーションを行った。ここで、 $I'_m = (\nabla \cdot \sigma \nabla \phi) / \beta$, $I'_{ion} = -I_{ion}$, $c = 1/C_m$ となる。 I'_m は隣り合う細胞から流入してくる電流の総和を表す。なお今回の評価では縦方向と横方向を考慮した図 2 に示すような 2 次元モデルを用いて各点での膜電位の計算を行った。また I_{ext} は外部刺激であり興奮のきっかけのための電流であるので、端のひとつの細胞に $I_{ext} = 200.0$ を $t = 0 \sim 1.5$ のときに加えた。それ以外の細胞・時間では $I_{ext} = 0$ とした。

3. ICCG 法

本研究では生成された連立一次方程式の求解に ICCG 法 (不完全コレスキー分解付き共役勾配法) を用いた。これは係数行列が疎行列・正値対称行列のときに有効な解法とされてきたからである。連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ を ICCG 法により解く場合、その解法は前処理と共役勾配法 (CG 法) に大きく分けることができる。係数行列 \mathbf{A} を前処理行列として不完全コレスキー分解を行うことで下三角行列 \mathbf{L} を得る。

$$\mathbf{q} = (\mathbf{LL}^T)^{-1} \mathbf{r} \quad (9)$$

により近似解 \mathbf{q} を得る。ここで、 \mathbf{r} は残差ベクトルである。上の \mathbf{q} は以下のように求める。

$$(\mathbf{LL}^T) \mathbf{q} = \mathbf{r}$$

$$\mathbf{LL}^T \mathbf{q} = \mathbf{r}$$

$\mathbf{L}^T \mathbf{q} = \mathbf{y}$ とすることで、

$$\begin{cases} \mathbf{Ly} = \mathbf{r} \\ \mathbf{L}^T \mathbf{q} = \mathbf{y} \end{cases} \quad (10)$$

となり、式 (10) の上側の式は前進代入法、下側の式は後退代入法で求めることができる。ここまでの前処理となり、次に \mathbf{q} を用い CG 法により解を求めることになる。

———— CG 法アルゴリズム ————

$$\begin{aligned} \mathbf{x}^{(0)} \\ \mathbf{r}^{(0)} = \mathbf{b} - \mathbf{Ax}^{(0)} \\ \mathbf{p}^{(0)} = \mathbf{q}^{(0)} \end{aligned}$$

DO k=1,2,...

$$\begin{aligned} \alpha_k &= \frac{(\mathbf{q}^{(k)}, \mathbf{r}^{(k)})}{(\mathbf{p}^{(0)}, \mathbf{Ap}^{(k)})} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)} \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k \mathbf{Ax}^{(k)} \end{aligned}$$

$$\begin{aligned} \beta_k &= -\frac{(\mathbf{q}^{(k+1)}, \mathbf{Ap}^{(k+1)})}{(\mathbf{q}^{(k)}, \mathbf{Ap}^{(k)})} \\ \mathbf{p}^{(k+1)} &= \mathbf{q}^{(k+1)} + \beta_k \mathbf{p}^{(k)} \end{aligned}$$

END DO

このとき k は反復回数を示す。このように、任意の $\mathbf{x}^{(0)}$ から探索を始め、定めた収束条件を満たすまで反復を繰り返す。またこのとき残差ベクトル $\mathbf{r}^{(k)}$ と探索方向を表す $\mathbf{p}^{(k)}$ は、直交性と共役性を持つ。すなわち、

$$\begin{aligned} (\mathbf{r}^{(i)}, \mathbf{r}^{(j)}) &= 0 \quad (i \neq j) \\ (\mathbf{p}^{(i)}, \mathbf{Ap}^{(j)}) &= 0 \quad (i \neq j) \end{aligned}$$

を満たす。

4. 並列化手法

いかに ICCG 法が優れた解法であったとしても、問題の規模が大きくなれば計算時間が増加することは避けられない。そこで近年、並列処理により問題を高速に解くための様々な手法が研究・提案されている。そのひとつであるのが今回用いた AMC 法 (algebraic multicolor ordering method)⁵⁾⁶⁾ 及び BJ 型 IC 法 (block Jacobi incomplete Cholesky method)⁷⁾ である。いずれも並列効果の妨げとなる前進・後退代入計算を並列化し高速に計算するための方法である。今回は OpenMP によりこれらの方法を実装した。

4.1 ICCG 法の並列化

OpenMPは共有メモリ型並列計算機で用いられる並列処理のためのAPIである。ユーザがデータのやりとりを明示してやる必要はなく、ディレクティブをプログラムに挿入することによって使用できる。ICCG法を用いての計算において求められる計算要素は、ICCG法のアルゴリズムに示すように、主に4つである。すなわち、(a)内積計算、(b)行列とベクトルの積、(c)ベクトルの更新、(d)前進・後退代入計算、である。この中で前の3つの計算要素はOpenMPにより容易に並列化できる。それは用いる情報が1つ前の段階で計算されたものであり、並列化を行ってもデータ同士が干渉しあわないためである。一方で前進・後退代入計算に関しては簡単には並列化できない。これはデータ同士の干渉が生じるためである。そのため前進・後退代入計算は逐次計算にならざるを得なくなり、これが並列効果の妨げになるという問題が生じる。今回この問題を解決し、よりよい並列効果を実現するために2つの手法を用いた。ともに不完全コレスキー分解の対象となる係数行列 A に予め処理を施し \bar{A} へと変換するための手法である。変換後、前処理のための行列として A の代わりに \bar{A} を使用することになる。

4.2 AMC 法

AMC法は計算をする上で独立な要素を集めてひとつのグループとし、グループ単位で並列計算を行えるようにする手法である。ナンバリングと呼ばれるグループ分けを行った後に並べ替えを行う(リオーダーリング)。

4.2.1 ナンバリング

ナンバリングのポイントは関係のある要素同士を別のグループに割り当てるということである。手順は大きく3つに分けることができる。まず、係数行列の下三角部分が持つ一行あたりの最大要素数を色数に指定する。図3の左図は 4×4 のメッシュを用いたときの係数行列を表しており、今回のような5点中心差分問題では色数は3色になる。次にその色を基本的には順番に要素に割り当てていく。このとき割り当てられる色がその要素と関係性のある場合は次の色を割り当てる。この手順に従うことで図3の左図・右横の数字のように番号を割り当てることができる。

4.2.2 リオーダーリング

係数行列を色分けすることができれば次にそれら色順に並べ替える必要がある。基本的に色1のグループから上位に並べる。対称性を崩さないために行に対する処理を施した後は列に対する処理も同様に施す必要がある。並び替えを行ったものが図3の右図である。

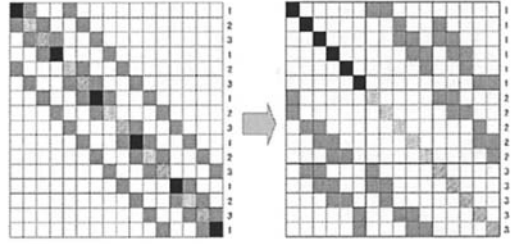


図3 色分けの例

これで $\bar{A}(= \bar{A}_{amc})$ が完成する。

4.2.3 AMC 法適用後の不完全分解

並べ替えた後の連立一次方程式を

$$\bar{A}_{amc} \bar{x}_{amc} = \bar{b}_{amc}$$

とおく。ここで

$$\bar{x}_{amc} = [\bar{x}_{C(1)}, \bar{x}_{C(2)}, \dots, \bar{x}_{C(dlr)}]^T \quad (11)$$

$$\bar{b}_{amc} = [\bar{b}_{C(1)}, \bar{b}_{C(2)}, \dots, \bar{b}_{C(dlr)}]^T \quad (12)$$

$$\bar{A}_{amc} = \begin{bmatrix} \bar{C}_{1,1} & \bar{C}_{1,2} & \dots & \bar{C}_{1,dlr} \\ \bar{C}_{2,1} & \bar{C}_{2,2} & \dots & \bar{C}_{2,dlr} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}_{dlr,1} & \bar{C}_{dlr,2} & \dots & \bar{C}_{dlr,dlr} \end{bmatrix} \quad (13)$$

となる。ここで dlr は設定された色数である。また $\bar{x}_{C(ic)}$ は ic 番目の色を持った未知変数を要素にするベクトル、 $\bar{C}_{i,m}$ は $\bar{x}_{C(i)}$ と $\bar{x}_{C(m)}$ の関係をあらわす行列である。 \bar{b}_{amc} は \bar{x}_{amc} に対応させて b を並び替えたものである。 \bar{A}_{amc} が決定すれば、不完全コレスキー分解により、前進・後退代入計算に用いる $\bar{L}_{amc}, \bar{D}_{amc}$ を決定することができる。

$$\bar{L}_{amc} = \begin{bmatrix} \bar{D}_{a(1)} & & & \\ \bar{L}_{2,1} & \bar{D}_{a(2)} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{L}_{dlr,1} & \bar{L}_{dlr,2} & \dots & \bar{D}_{a(dlr)} \end{bmatrix} \quad (14)$$

$$\bar{D}_{amc} = \begin{bmatrix} \bar{D}_{a(1)}^{-1} & & 0 \\ & \ddots & \\ 0 & & \bar{D}_{a(dlr)}^{-1} \end{bmatrix} \quad (15)$$

ここで $\bar{D}_{a(i)}$ は対角行列であり、

$$\bar{A}_{amc} \simeq \bar{L}_{amc} \bar{D}_{amc} \bar{L}_{amc}^T \quad (16)$$

という関係になる。不完全分解ができれば、

$$\bar{L}_{amc} \bar{y}_{amc} = \bar{r}_{amc} \quad (17)$$

により、前進代入で \bar{y}_{amc} が求まる。ここで

$$\bar{y}_{amc} = [\bar{y}_{C(1)}, \bar{y}_{C(2)}, \dots, \bar{y}_{C(dlr)}]^T \quad (18)$$

残差ベクトルを \tilde{r}_{amc} とすると、式(14)、式(15)より ic 色における前進代入計算は以下のように計算できる。

$$\tilde{y}_{C(ic)} = \tilde{D}_{\alpha(ic)}^{-1} \left(\tilde{r}_{C(ic)} - \sum_{k=1}^{ic-1} \tilde{L}_{ic} \tilde{r}_{C(k)} \right) \quad (19)$$

同様に後退代入計算も行うことができる。このとき同じ色の中ではデータの干渉が生じないために、前進・後退代入計算の部分も他の計算要素と同様 OpenMP を用いての並列化が可能であり、並列効果の向上が見込まれる。なお AMC 適用後の CG 法において \mathbf{A} の代わりに $\tilde{\mathbf{A}}_{amc}$ を用いることになる。

4.3 BJ 型 IC 法

BJ 型 IC 法による前処理アルゴリズムは AMC 法に比べて単純である。係数行列をいくつかのブロックに分割し対角ブロック以外を全てゼロ行列にすることで複数ブロックを用いて同時に並列計算することができる。

4.3.1 変換方法

前処理の方法について述べる。まず、並列に用いる CPU 数 p を決定する。つぎに係数行列を行方向に p 等分、列方向に p 等分する。これで係数行列は $p \times p$ のブロックに分割されたことになる (式 (20))。

$$\mathbf{A} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,p} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \cdots & \mathbf{B}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{p,1} & \mathbf{B}_{p,2} & \cdots & \mathbf{B}_{p,p} \end{bmatrix} \quad (20)$$

このとき、 $\mathbf{B}_{i,j}$ の (l, m) 成分は $a_{((i-1)N/p+l, (j-1)N/p+m)}$ となる。ここで $a_{(\alpha, \beta)}$ は \mathbf{A} の (α, β) 成分、 N は係数行列 \mathbf{A} の次元数である。次に、式 (20) において、 $\mathbf{B}_{i,j} = \mathbf{0} (i \neq j)$ とすることで $\tilde{\mathbf{A}} (= \tilde{\mathbf{A}}_{bj})$ ができる。ここで $\mathbf{0}$ はその成分が全て 0 のゼロ行列である。

$$\tilde{\mathbf{A}}_{bj} = \begin{bmatrix} \tilde{\mathbf{B}}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \tilde{\mathbf{B}}_p \end{bmatrix} \quad (21)$$

ここで $\tilde{\mathbf{B}}_i = \mathbf{B}_{i,i}$ である。図 4 の右図は 4×4 のメッシュで作られる係数行列を $p = 2$ で分割したときの例である。

4.3.2 BJ 型 IC 法適用後の不完全分解

次に式 (21) を生成したあとの不完全分解及び前進・後退代入計算について述べる。式 (21) で表される $\tilde{\mathbf{A}}_{bj}$ を不完全分解し、

$$\tilde{\mathbf{A}}_{bj} \simeq \tilde{\mathbf{L}}_{bj} \tilde{\mathbf{D}}_{bj} \tilde{\mathbf{L}}_{bj}^T \quad (22)$$

を得る。このとき $\tilde{\mathbf{L}}_{bj}$ は以下のように書ける。

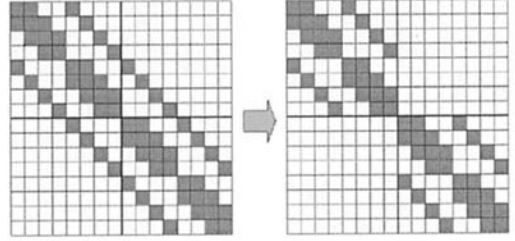


図 4 2CPU の場合

$$\tilde{\mathbf{L}}_{bj} = \begin{bmatrix} \tilde{\mathbf{L}}_{b(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \tilde{\mathbf{L}}_{b(p)} \end{bmatrix} \quad (23)$$

ここで $\tilde{\mathbf{L}}_{b(1)} \sim \tilde{\mathbf{L}}_{b(p)}$ は下三角行列となる。従って、これらと

$$\tilde{\mathbf{y}}_{bj} = [\tilde{\mathbf{y}}_{b(1)}, \tilde{\mathbf{y}}_{b(2)}, \dots, \tilde{\mathbf{y}}_{b(p)}]^T \quad (24)$$

$$\tilde{\mathbf{r}}_{bj} = [\tilde{\mathbf{r}}_{b(1)}, \tilde{\mathbf{r}}_{b(2)}, \dots, \tilde{\mathbf{r}}_{b(p)}]^T \quad (25)$$

を用いて

$$\tilde{\mathbf{L}}_{bj} \tilde{\mathbf{y}}_{bj} = \tilde{\mathbf{r}}_{bj} \quad (26)$$

の前進代入の計算は以下のように与えられる。

$$\begin{cases} \tilde{\mathbf{y}}_{b(1)} = \tilde{\mathbf{L}}_{b(1)}^{-1} \tilde{\mathbf{r}}_{b(1)} \\ \tilde{\mathbf{y}}_{b(2)} = \tilde{\mathbf{L}}_{b(2)}^{-1} \tilde{\mathbf{r}}_{b(2)} \\ \vdots \\ \tilde{\mathbf{y}}_{b(p)} = \tilde{\mathbf{L}}_{b(p)}^{-1} \tilde{\mathbf{r}}_{b(p)} \end{cases}$$

となり、 $\tilde{\mathbf{y}}_{b(1)} \sim \tilde{\mathbf{y}}_{b(p)}$ の計算は p 個の CPU を用いて並列に実行することが可能である。後退代入計算においても同様のことが言え、これにより前進・後退代入計算部分も並列化することが可能となる。なお、BJ 型法を適用後の CG 法部分に関しては、 $\tilde{\mathbf{A}}_{bj}$ ではなく、 \mathbf{A} をそのまま使うことになる。

5. 数値シミュレーション

本計算は、京都大学学術情報メディアセンターのスーパーコンピュータ HPC2500 上で行った。並列化手法はスレッド並列を行う OpenMP を用いた。計算結果を図 5、6 に示す。パラメータは前節までに示した通りである。評価には 11time step を計算するのに要した計算時間 (図 5) 及び並列度を上げたときの台数効果 (図 6) を用いる。両図ともに、横軸には並列計算で用いた CPU 数を 2 を底として対数表示している。図中の PART は 3.1 節における計算要素の

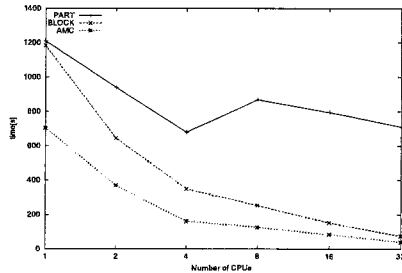


図5 計算時間

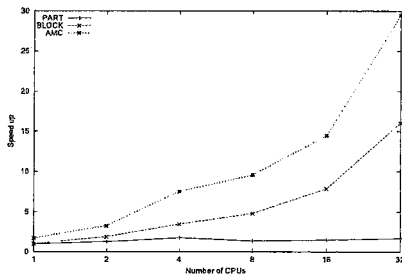


図6 台数効果

内、前進・後退代入計算以外を並列計算したときの結果である。BLOCKはPARTに加えBJ型IC法を、AMCはPARTに加えAMC法を適用し前進・後退代入計算も並列計算したときの結果になる。図5を見ると、PARTでは並列度を上げたとしても大した並列効果がみられず前進・後退代入計算が並列効果の阻害要因となっていることが確認できる。一方、AMC、BLOCKでは並列数の増加によって計算時間も堅実に減少していることがわかる。また図6では縦軸に台数効果をとっている。これは1CPUを用いてPARTで計算したときの時間を基準として、それぞれが何倍速度向上しているかを表す。図をみると明らかなように、AMC、BLOCKにおいては台数効果が確認できる。

6. 考 察

図5において、CPUを1つ用いて計算を行ったときにAMCが他の2つのソルバと比べ早く収束する結果となった。計測を行うと、この計算時間の差はやはり前進・後退代入計算部分にあった。CPUが1つのときはBLOCKとPARTはまったく同じ計算になるために、図5にあるようにほぼ同じ計算時間が掛かる。一方で、AMCを用いる場合には係数行列を並び替え前処理行列を変化させている。このことが計算速度に差を生んでいると考えられる。つまり、並び替え

によって要素同士が比較的近くに配置されメモリアクセスに変化が生じたことによる可能性や、色ごとの代入計算がすべて独立しているためにプロセッサがパイプラインをより効率的に活用できた可能性などが考えられる。この詳細に関しては現在調査中である。

7. おわりに

FHN-modelを用いてモノドメイン環境で細胞の興奮伝播シミュレーションを行った。またこの問題に対して並列化ICCGソルバの評価を行った。具体的にはICCG法に不可欠な前進・後退代入計算を並列計算するために、AMC法及びBJ型IC法を実装した。その結果、共有メモリ型並列計算機であるHPC2500を用いて、32CPUで並列化したところAMC法では約29倍、BJ型IC法では約16倍の並列効果を得ることができた。このことよりハイドメインモデルのコードの構築は困難ではあるが、今回の並列化手法、特にAMC法を用いれば大きな並列化効果を期待できると考えられる。

参 考 文 献

- 岡本良夫編著, "心臓のフィジオーム", 森北出版株式会社, 2003
- "細胞・生体機能シミュレーションプロジェクト", 文部科学省リーディングプロジェクト
<http://www.lp-biosimulation.com/index.shtml>
- FitzHugh R, "Impulses and physiological states in theoretical models of nerve membrane", Biophys.J. pp445-466, 1961
- "第5章 共役勾配法", pp.39-51
<http://www.coins.tsukuba.ac.jp/itosho/NC05>
- Takeshi Iwashita and Masaki Shimasaki, "Algebraic Multicolor Ordering for Parallelized ICCG Solver in Finite-Element Analyses", IEEE TRANSACTION ON MAGNETICS, VOL.38, NO.2, MARCH 2002, pp.429-431
- 岩下武史, 島崎真昭, "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, 43巻, 4号, 2002, pp.893-904
- Yousef Saad, "Iterative Methods for Sparse Linear Systems", SIAM, 2003