社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

分布定数系生体機能モデルの並列シミュレーションにおけるデータ分割改善手法

桑　　幸生† 　前濱　貴哉† 　プンザラン フロレンシオ ラスティ†† 　桑原　寛明††† 　國枝　義敏†††
天野　　晃††

† 立命館大学大学院情報理工学研究科
〒 525–8577 滋賀県草津市野路東 1 丁目 1–1
†† 立命館大学生命科学部
††† 立命館大学情報理工学部
E-mail: †{y-kuwa,t-maehama}@hpcss.is.ritsumei.ac.jp, ††{floren,a-amano}@fc.ritsumei.ac.jp,
†††{kuwabara,kunieda}@cs.ritsumei.ac.jp

あらまし　本研究では MPI 向けの生体機能シミュレーションプログラムのデータ分割を自動で変更することでシミュレーションを高速化する手法を提案する．生体機能シミュレーションでは，通常偏微分方程式系で記述された細胞モデルを扱う．細胞モデルとして与えられた偏微分方程式系を，多数の細胞ごとに，時間発展計算の形で大規模な数値計算を行う．この計算時間を短縮するために，複数の PE(Processor Element) からなる並列計算機による並列化が求められる．MPI(Message Passing Interface) などを用いて，各 PE が相互に必要となるデータをやりとりしつつ，各 PE に細胞群を分担させ計算させる．細胞群の分割次第では，それぞれの PE 間で実行時間に差異が生まれる．PE 間に実行時間の差があると，シミュレーションに必要な実行時間が大きくなる．そこで本論文では，細胞群の分割を自動で変更することで，PE 間の実行時間を平準化する手法を提案する．PE 間の実行時間が平準化することで，実行時間が遅い PE をなくし，生体機能シミュレーション全体の高速化を実現することを目的とする．
キーワード　生体機能シミュレーション，並列シミュレーションプログラム，CellML，データ並列，データ分割

A Refinement Method of Biological Morphology Data Division for Parallel Computing of Distributed Parameter Systems

Yukio KUWA†, Takaya MAEHAMA†, Florencio RUSTY PUNZALAN††, Hiroaki KUWABARA†††, Yoshitoshi KUNIEDA†††, and Akira AMANO††

† Graduate School of Information Science and Engineering, Ritsumeikan University
Nojihigashi 1-1-1, Kusatsu, Shiga, 525–8577,Japan
†† Faculty of Life Science, Ritsumeikan University
††† Faculty of Information Science and Engineering, Ritsumeikan University
E-mail: †{y-kuwa,t-maehama}@hpcss.is.ritsumei.ac.jp, ††{floren,a-amano}@fc.ritsumei.ac.jp,
†††{kuwabara,kunieda}@cs.ritsumei.ac.jp

Abstract　In this research, we propose a method to improve the simulation speed of MPI parallel program by automatically repartitioning the target simulation space. By using the MPI parallel program, the total simulation time is reduced compared to the sequential program, however, the performance strongly depends on the simulation space partition. In this method, we used the measured computation time of each processing element (PE) to refine the assigned area of each PE which leads to the balanced computational time of all PE.
Key words　biological function simulation, parallel simulation program, CellML, data parallel, data divide

1. はじめに

生命科学研究の分野において，多階層生体機能シミュレーションは各階層の細胞活動を包括的に解析できるため注目が集まっている [1]．多階層生体機能シミュレーションは下位階層の細胞単位のモデルを多数用いて上位階層である組織や臓器をシミュレートするマルチスケールなシミュレーションである [2]．多階層生体シミュレーションを行うことで，上位階層における生体機能の挙動を，最下層の細胞レベルのシミュレーションで精密に再現し解析できる．本研究は特に断りがない限り多階層生体機能シミュレーションを生体機能シミュレーション, または単にシミュレーションと表現する．多階層生体機能シミュレーションを行うための細胞モデルとして，分布定数系モデルがある．本研究では分布定数系モデルを用いた生体機能シミュレーションを対象とする．

多数の偏微分方程式系で構成された細胞モデルから手作業で生体機能シミュレーションプログラムを作成するには多大な労力を要するので，生体機能シミュレーションプログラムを自動生成するソフトウェアの需要が高まっている．細胞モデルから生体機能シミュレーションプログラムを自動生成するツールの開発はすでに行われており，いくつかの自動生成ツールが存在する [3]〜[7]．これらのツールを使うことにより，プログラミングの知識なしに細胞モデルから生体機能シミュレーションを行うことが可能である．

生体機能シミュレーションでは，通常細胞モデルとして与えられた偏微分方程式系を，多数の細胞ごとに，時間発展計算の形で大規模な数値計算を行う．この計算時間を大幅に短縮するために，複数のコンピュータや PE(Processor Element) からなる並列計算機 (例えば複数のパソコンをネットワークで結合した PC クラスタなどが一例) による並列処理が求められる．この場合，例えば MPI(Message Passing Interface) などを用いて，各 PE が相互に必要となるデータをやりとりしつつ，各 PE に細胞群を分担させ計算させる．一般に，PC クラスタで並列計算を行うために入力である並列処理するデータ群を分割し，分割した数と同じ数のプロセスを生成し，生成したプロセスを個々のコンピュータに割り当てる．データ分割の時，割り当てられるデータによって計算量が異なるため，各プロセスの実行時間に差異が生じる．各プロセスの実行時間に差異がある場合，実行時間が速いプロセスは実行時間が遅いプロセスを待つ必要があり，待っているプロセスは計算を行っていない時間があるため無駄が生じる．実行時間が遅いプロセスに割り当てられているデータの一部を他のプロセスに配置し，最も時間がかかっているプロセスの実行時間を短縮することで，生体機能シミュレーションの実行時間を短縮できる．

本研究では，クラスタを用いた生体機能シミュレーションコードのデータ分割を自動で行う手法を提案する．提案する手法は，計算時間が遅いプロセスの計算するべき細胞群を他のプロセスに再配置し，プロセスの最大実行時間を短縮することで，生体機能シミュレーションの全体の実行時間を短縮する．生体機能シミュレーションは時間発展をループで表現し，微分方程式系に対して時間発展計算を行う．時間発展のループを小さくすることで，各プロセスの実行時間の割合を変化させずシミュレーションの規模を小さくできる．規模を小さくした生体機能シミュレーションを実環境で動作させ，計算時間が長いプロセスが計算するべき細胞群を再配置し，計算時間が最も長いプロセスの計算時間が短縮される分割方法を探す．規模を小さくしたシミュレーションで発見された最適な分割法を，本来の大きさの時間発展ループの生体機能シミュレーションに適用させることで，シミュレーション時間を短縮する．ユーザである生体分野の研究者は提案する手法を実装したツールを利用することで，細胞群の分割方法を考えることなく，MPI で動作する生体機能シミュレーションプログラムを作成することができる．

2. 生体機能シミュレーション

2.1 生体機能シミュレーションコードの自動生成系 CellCompiler

生体機能シミュレーションのシミュレーションコード自動生成系として CellCompiler が提案されている [8]〜[10]．

CellCompiler は解析器とコード生成器から構成されている．解析器の入力は生体機能モデルである CellML モデル，方程式の解法が記述された解法スキームである TecML，時間や空間に関する境界条件である RelML, 形状情報である．

- CellML モデル

CellML [11] はオークランド大学で開発された細胞の機能と動きを定量的に表す XML ベースの細胞生理学モデルの記述言語であり，CellML で記述された多くの細胞生理学モデルが CellML Web レポジトリ [12] に登録されている．

- TecML

TecML [8] は方程式の解法を示す数式の集合（解法スキーム）で XML で記述する．ユーザは TecML を用意するか事前に用意されているものを選択する．

- RelML

RelML [8] は初期条件，終了条件，境界情報などのシミュレーションに必要なデータで XML で記述する．CellML と TecML の両者で使われている変数間の関係を記述する．

- 形状情報

形状情報は直交格子形状内で細胞をどの位置に配置するかを示す情報で，画像ファイルの形式である BMP ファイルである．細胞はメッシュ番号という番号をつけて管理する．

CellCompiler の解析器はこれらの入力から各細胞に関して計算すべき数式集合（数式セット）を作成する．数式セット作成の際，システムはまず，解法スキームや境界条件，形状情報を細胞モデルに適用し，計算順序が決定していない数式セットを作成する．次に，数式間の依存関係を調べ計算順序を決定し，数値的に解くことができる数式セットを作成する．これらの数式セットと細胞モデルの形状から中間形式である StructuredRecMLPDE を生成する．StructuredRecMLPDE には 4 つの要素が含まれる．

（1） 細胞の隣接情報

それぞれの次元方向への隣接する細胞のメッシュ番号が記述さ

れ，入力された形状情報より作成される．

（2） 入力ファイルより作成した数式セット

入力ファイルより数式間の依存関係が解析され，計算順序が決定した数式セットが記述される．

（3） 格子点で必要な数式セットの関連付けに関する情報

格子点に対して，どの数式セットが細胞のシミュレーションの計算に必要かを記述する．数式セットには数式セット間で依存関係がある場合があり，書かれた順で計算を行うことにより依存関係を気にせず実行することができる．

（4） 実験手順や初期値に関する情報

シミュレーション対象の細胞の個数や実験開始時間と実験終了時間，時間の刻み幅が記述されている．時間の刻み幅は時間発展計算における時間の増加分を示す．

コード生成器は StructuredRecMLPDE からシミュレーションコードを作成する．シミュレーションコードは，時間発展計算をループにより計算させることで，生体機能シミュレーションを行うため，実行時間は時間発展の数にほぼ比例する．コード生成器を変更することで，様々な言語に対応することが可能となる．先行研究 [9] では C，Java, CUDA のシミュレーションコードが生成可能となっている．ユーザはコード生成器から出力されたコードを最適なコンパイラでコンパイルすることで様々な実験環境でシミュレーションを実行できる．

2.2 並列生体機能シミュレーション

生体機能シミュレーションプログラムは計算量が膨大になり，実行時間が長くなるので，並列化することが求められている．大規模なシミュレーションを行う方法として，コンピュータクラスタを利用する方法が挙げられる．コンピュータクラスタとはコンピュータ同士を Ethernet に代表されるネットワークで接続し同時に動作させることにより論理的に 1 台のコンピュータとして扱う技術である．クラスタに適したプログラムとして MPI という並列プログラムを記述するライブラリインターフェースがある．MPI は指定した数のプロセスをコンピュータ上で生成し，クラスタのプロセッサに分配される．分配されたプロセスを認識するための番号を PE(Processing Element) と呼ぶ．

本論文で扱う生体シミュレーションの分野では，並列処理されるデータとは，シミュレートされる細胞群がそれぞれ持つ各種変数である．本稿でデータ分割という場合，これらの変数を保持する細胞群を分割することを指す．細胞群の領域の分割方法と PE への割り当て方法を本論文では細胞の分割手法と呼ぶ．

分布定数系モデルの生体機能シミュレーションでは細胞間の影響を含む方程式を計算する必要があるため，各細胞の計算に隣接する細胞の情報を必要とする．各 PE は演算を行う細胞群のみ正しい情報を保持しているため，各 PE に割り当てられた細胞の境界にある細胞の情報を時間発展ごとに PE 間で通信を行い，得た情報をもとに計算を行う必要がある．各 PE の境界にある細胞は，分割手法により通信量や通信回数が変化する．

2.3 並列生体機能シミュレーションコードの生成

文献 [13] では生体機能シミュレーションの MPICH のシミュレーションコードの自動生成方法について提案している．MPICH のシミュレーションコードを自動生成するために StructuredRecMLPDE に以下の二つを追加する．

- commvars

ある格子点の計算に必要な隣接格子点の情報である．

- cpulist

cpulist は各 PE が計算すべき格子番号のリストである．

既存研究 [13] では cpulist と commvars を手動で作成することで MPICH のシミュレーションコードを自動生成できるようにコード生成系を変更している．MPICH のシミュレーションコードを作成するためには主に二つの作業を行う．一つ目は，細胞群を分割し各 PE で計算するべき細胞群を配置する．この細胞群の分割と配置方法は人間が手作業で作成する．二つ目は，通信部分の作成である．通信を行うコードの作成のため commvars と呼ばれる細胞のシミュレーションの際に隣接している細胞の情報がないと計算できない変数のリストを作成する．cpulist の情報から各 PE が計算するべき細胞の要素がわかるため，commvars から得られる隣接している細胞の情報との論理積を求めることで，正しい値を得るために必要な通信先を求める．これを全組み合わせで行うことで正しい通信を行うコードが作成される．

2.4 細胞群の領域を分割する際の問題点

最も遅い PE の計算量を削減し，削減した計算を他の PE に再配置することで，最も計算時間が遅い PE の実行時間は短縮される．これにより，再配置された他の PE の計算量は増加するが，最も遅い PE を待っていた時間で計算できるので，実行時間は増加しない．よって，シミュレーション全体の実行時間が短縮できる．通信時間を考慮することも必要である．通信時間は通信回数が増えると長くなる．各 PE に割り当てられた細胞群の境界にある細胞の隣接している細胞の数が増えると通信回数が増加する．

生体機能シミュレーションにおいて細胞の領域は複雑な形状になる．細胞領域が複雑な場合, 実行時間が均等になるように格子状に分割するのは困難である．また，細胞群を分割するのが困難な場合がある．クラスタを構成するコンピュータは常に性能が同じであるとは限らない．性能に違いがある場合，性能が低いコンピュータに割り当てられた PE の実行時間が遅くなる．生体機能シミュレーションにおいて人間が考えた細胞の分割法を用いたシミュレーションコードでは高速なシミュレーションを行うことができない．

3. 細胞分割手法とその自動化

3.1 計算するべき細胞群の配置の手法

生体機能シミュレーションコードを実環境で実行し，実行にかかった時間を記録する．記録したデータから一番実行時間が長い PE を特定し，その PE の実行時間が短くなるように分割手法を変更する．PE の実行時間を短くするためには，割り当てられている細胞群の量を小さくし，他の PE に配置する．配置の時，実行時間を短くしたい PE に割り当てられた細胞のみを移動すると通信を行わなければならない PE の数が増えるため通信時間が増加する．通信時間が極端に変化してしまうこと

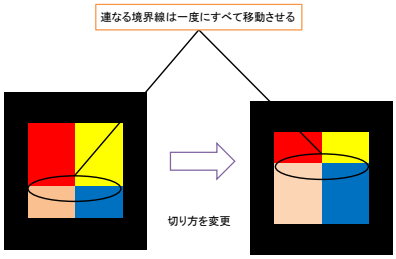

図 1　分割方法変更の例

を避ける為，図 1 のように連なる境界線を全て移動する．格子状の分割を行うという制約のもとで分割法を決定すると全ての PE の実行時間が同じになる分割法が存在しない場合があるので完全に PE の実行時間を平準化することはできない．

移動させる境界線と移動量の決定方法を示す．

- 移動させる境界線の決定

境界線を移動することにより，計算時間が遅い PE 以外の PE の実行時間も同時に変化するので，計算時間が遅い PE 以外の PE の実行時間も考慮して移動させる境界線を決定する．まず，最も実行時間が長い PE を求める．最も実行時間が長い PE を中心として，縦に区切る境界線か横に区切る境界線のどちらを移動させるかを決定するために，最も計算時間の長い PE が計算する細胞群を含む縦一列と横一列の PE の計算時間の合計を算出し，計算時間の合計が長い方向を移動する．分割線の移動方向が決定すると、移動させる分割線に接する PE 列の合計計算時間を算出し，移動させる分割線に隣接した PE 列の合計計算時間が短い方の分割線を最も計算時間が長い PE が計算する細胞群の量が小さくなるように移動させる．

- 移動量の決定

最初に移動する量はユーザが決定する．2 回目以降は，最初に移動した PE の実行時間の短縮量を元に，移動したい PE の計算時間が，全 PE の計算時間の順位の半分の位置になる量を求める．決定した分割線を決定した移動量で移動する．最低の移動量は細胞ひとつ分である．

3.2　時間発展の数によるシミュレーションの計算時間

生体機能シミュレーションにおいて，一度の時間発展計算で計算できるシミュレーションの時間は短いので，ループを大きくし，何度も実行する必要がある．高速化を実現するまでの時間的コストを削減するために，まず目標とする生体機能シミュレーションの時間発展のループを縮小した小規模なシミュレーションを用意する．小規模なシミュレーションに対して再分割手法の探索による高速化を実施し，高速な分割法を確定する．目標とするシミュレーションに確定した分割法を適用することで実行時間の短縮ができる．時間発展計算の途中で細胞モデルを変更したり，細胞の死滅により計算する細胞量が変化したりするような実行時間に時間発展の数が比例しないシミュレーション方法も存在するが，本研究では対象としていない．

表 1　実 験 環 境

OS	CentOS 6.5
CPU	Intel(R)Xeon(R)X5680(3.33GHz)(1 台は 1.6GHz)
RAM	24GB
ハブ	ギガビットスイッチングハブ (通信速度:1Gbps)
LAN	1000BASE-T CAT6 ケーブル

3.3　AutoDivide の実装

プンザランらは，PDE で記述された分布定数系モデルに対し，CellCompiler を用いたシミュレーションコードの自動生成系を提案している [10]．天野らは CellCompiler を改変し，クラスタ向けシミュレーションコードの自動生成を確認している．本研究で用いた CellCompiler は天野らにより開発されたもの [13] を改変し，用いている．分割方法を自動で変更するツール（AutoDivide）について説明する．

3.3.1　初期分割手法

CellCompiler のコード生成系は形状情報を画像ファイルである BMP ファイルから取得している．ユーザが BMP ファイルのデータに色を塗り分けることで分割の指定ができるよう CellCompiler の仕様を変更する．AutoDivide の 1 回目の入力はこれにより作成されたユーザが作成した細胞分割手法である．ただし，AutoDivide の入力とするためには直交格子状に分割する必要がある．

3.3.2　AutoDivide の構成

AutoDivide は時間発展の数を変更したシミュレーションコードの実行時間を取得し，cpulist を変更することで，分割方法を変更する．この操作を極端に計算時間が遅い PE がなくなるか，複数回実行してもシミュレーション時間が変わらなくなるまで繰り返すことで，シミュレーションの実行時間が短い細胞分割手法を見つける．AutoDivide の実装にあたり，シミュレーションプログラムの処理時間を測定し，ファイルに出力するプログラムが生成されるように CellCompiler を変更する．実行の最後に csv 形式で各 PE ごとに計算している部分の実行時間と，待ち時間を含む通信に必要な時間をログファイルとして出力する．この変更により実行時間はほぼ変わらない．

4.　評　　価

4.1　実 行 環 境

表 1 に示す環境において実験を行う．本実験では，6 台のコンピュータを Ethernet ケーブルでつなぎクラスタを構築した．6 台のうち 1 台のコンピュータの CPU のクロック数を 1.6GHz に変更し，実験を行った．クロック数を変更することで擬似的にクラスタを構成しているコンピュータの構成を変更している．

4.2　実　　験

細胞のモデルは HundRudy モデル [14]，時間発展方向のループ回数を 100 回で実験を行う．評価のため 2 つの実験を行う．1 つめは，細胞数が 400 × 400 の直交格子形状を図 2 のように 4 分割し，AutoDivide を実行し分割手法を変更した．2 つめは，細胞数が，約 31000 の心臓の形を輪切りにした形状を図 3 のように 16 分割し，AutoDivide を実行し分割手法を変更した．

図 2　400 × 400 の直交格子形状を 4 分割する初期分割方法

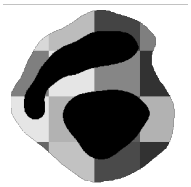

図 3　心臓の形状を 16 分割する初期分割方法

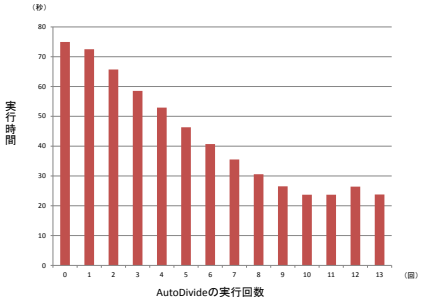

図 4　400 × 400 形状の 4 分割のシミュレーション実行時間の変化

4.3　実行時間と評価

4.3.1　直交格子形状の 4 分割

1 つめの実験では細胞数が 400 × 400 の直交格子形状を 4 分割して実行時間を計測する．6 台あるクラスタのうち 4 台を使って実行する．PE を 1 台につき 1 つずつ配置し，実験を行った．4 台のうち 1 台はクロック数を 1.6GHz に変更したコンピュータである．図 4 に AutoDivide を実行した回数と生体機能シミュレーションの実行時間の関係を示す．横軸に AutoDivide の実行回数，縦軸にシミュレーションの実行時間を表している．

AutoDivide で分割手法を変更する前は実行時間が 74.93 秒であったが，10 回 AutoDivide を実行した結果 23.71 秒となった．図 5 に AutoDivide を適用した回数と各 PE の計算時間の関係を示す．横軸に AutoDivide の実行回数，縦軸に各 PE の実行時間を表している．AutoDivide を実行する前は 2 番の PE の計算時間が 73.46 秒であったが，10 回 AutoDivide を実行した結果 2 番の PE の計算時間は 23.43 秒となった．10 回の AutoDivide の実行には約 20 分かかる．この後，AutoDivide を 100 回まで実行したが，100 回までの間でシミュレーションの実行時間が大きく変わることはなかった．

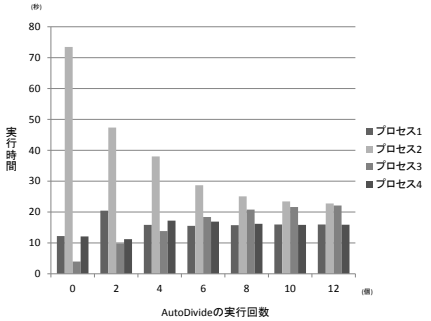

図 5　400 × 400 形状の 4 分割の PE の計算時間の変化

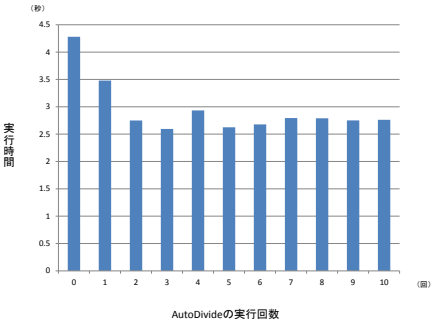

図 6　心臓の形状のシミュレーション実行時間の変化

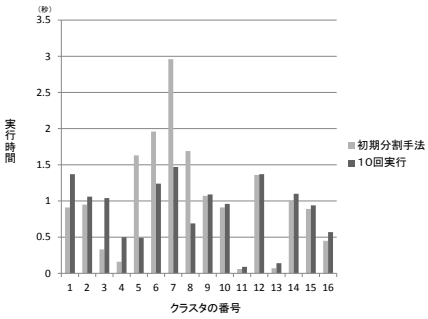

図 7　心臓の形状の PE の計算時間の変化

4.3.2　心臓の形状を 16 分割

細胞数が，約 31000 の心臓を輪切りにした形状のモデルを 16 分割して実行時間を計測する．6 台のクラスタを用いて実行する．1 台につき PE を 2 もしくは 3 ずつ配置し，実験を行った．図 6 に AutoDivide を実行した回数とシミュレーションの実行時間の関係を示す．横軸に AutoDivide の実行回数，縦軸にシミュレーションの実行時間を表している．AutoDivide を実行する前は 4.28 秒であったが，10 回 AutoDivide を実行した結果 2.75 秒となった．AutoDivide の 10 回の実行には約 20 分かかる．この後，AutoDivide を 100 回まで実行したが，100 回までの間でシミュレーションの実行時間が大きく変わることはなかった．図 7 に AutoDivide を適用した回数と各 PE の計算時間の関係を示す．横軸に PE の番号, 縦軸に PE の計算時間を表す．

4.4　考　　察

AutoDivide を実行することで，400 × 400 の直交格子形状

の生体機能シミュレーションプログラムの実行時間が 4 分割の場合 3.1 倍高速になる．心臓の形状の生体機能シミュレーションの実行時間は 1.5 倍高速になる．実験において AutoDivide を実行することで計算時間が遅い PE の計算量を減らし，高速化を実現できた．分割線が直線でなくてはならないという分割手法に制限があるため，完全な分散が実現できない．心臓の形状のように極端に実行時間が短い PE がある場合，形状と分割手法によっては負荷を分散できない PE が存在する．初期分割で極端に実行時間が短い PE を作らないようにすることでこの問題を解決できる．

AutoDivide で分割手法を変更すると，稀にシミュレーション時間が長くなる場合が存在する．最も実行時間が遅い PE を小さくなるように分割手法を変えるため，実行時間が遅い PE に隣接している PE の実行時間が長くなり，シミュレーション時間が長くなることがある．次の AutoDivide の実行により分割手法が変更され，配置されるためであり，一時的にシミュレーションの実行時間が増えることは問題ではない．

AutoDivide の実行により移動する境界線の決定は成功していると言える．提案手法により決定された境界線を移動することにより実行時間が短い PE の計算の配置が実現できている．決定した境界線の移動量については改善する必要があると考える．実験の全ての場合において，AutoDivide の複数回実行で同じ境界線を続けて移動させることがある．移動量を増やすことで同じ境界線を連続で移動させることが少なくなるため，分割手法の決定に AutoDivide の実行回数を少なくできる．

スーパーコンピュータなどの大規模な HPCC では細胞数と PE の数，時間発展の数を増やした大規模な生体機能シミュレーションを行う．本手法では大規模なシミュレーションを行う場合に提案手法を適用することで実行時間に大きな差が生まれる可能性がある．

5. おわりに

本論文では分布定数系モデルの並列シミュレーションにおけるデータ分割手法の改善について提案，実装を行った．本手法は，シミュレーションを行う前実験として，時間発展方向のループを少なくしたシミュレーションコードを実機で実行し，提案する手法を用い分割手法を変更する．変更した分割手法を用いて目標となる規模の生体機能シミュレーションを行うことで，実行時間を短縮する．

2 つの形状において本手法を適用した分割手法が改善されることを確認し，初期分割と比較し，最大で 3.5 倍高速化された．

今後の課題として，大規模な HPCC での実験を行うことや，3 次元として表現された細胞の形状情報を用いた生体機能シミュレーションの高速化が挙げられる．

謝辞 本研究は科研費 22136004 の助成を受けたものです．

文　献

[1] P.J. Hunter, P. Robbins, and D. Noble, “The IUPS human physiome project,” Pflügers Archiv European Journal of Physiology, vol.445, no.1, pp.1–9, Oct. 2002.

[2] 嶋吉隆夫，堀　謙太，陸　建銀，皿井伸明，天野　晃，“マルチフィジックス細胞・生体機能シミュレーションのための統合シミュレーション環境,” 電子情報通信学会技術研究報告. MBE, ME とバイオサイバネティックス，vol.103，no.730，pp.97–102，mar 2004． http://ci.nii.ac.jp/naid/110003286631/

[3] A. Garny, D.P. Nickerson, J. Cooper, R.W.D. Santos, A.K. Miller, S. McKeever, P.M.F. Nielsen, and P.J. Hunter, “Cellml and associated tools and techniques.,” Philos Transact A Math Phys Eng Sci, vol.366, no.1878, pp.3017–3043, Sept. 2008.

[4] A. Garny, D. Noble, P.J. Hunter, and P. Kohl, “Cellular open resource (cor): current status and future directions.,” Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, vol.367, no.1895, pp.1885–1905, May 2009.

[5] D.P. Nickerson, A. Corrias, and M.L. Buist, “Reference descriptions of cellular electrophysiology models,” Bioinformatics, vol.24, pp.1112–1114, April 2008.

[6] S. Missan and T.F. McDonald, “Cese: Cell electrophysiology simulation environment.,” Appl Bioinformatics, vol.4, no.2, pp.155–6, 2005.

[7] Y. Suzuki, Y. Asai, H. Oka, E. Heien, T. Urai, T. Okamoto, Y. Yumikura, K. Tominaga, Y. Kido, M. Nakanishi, K. Hagihara, Y. Kurachi, and T. Nomura, “A platform for in silico modeling of physiological systems iii.,” Conf. Proc. IEEE Eng. Med Biol Soc, vol.1, pp.2803–2806, 2009.

[8] A. Amano, N. Soejima, T. Shimayoshi, H. Kuwabara, and Y. Kunieda, “A general cellml simulation code generator using ode solving scheme description,” Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE, pp.940–944, 30 2011-sept. 3 2011.

[9] 山下義陽，副島直樹，川端真成，プンザラン フロレンシオ ラスティ，嶋吉隆夫，桑原寛明，國枝義敏，天野　晃，“形式的に記述された ODE 解法スキームに基づく CellML シミュレーションコード生成システム,” 生体医工学，vol.50，no.1，pp.68–77，2012.

[10] プンザラン フロレンシオ ラスティ，山下義陽，川端真成，“形式的に記述された PDE 解法スキームに基づく分布定数系生体機能モデルシミュレーションコード生成システム,” 生体医工学 = Transactions of Japanese Society for Medical and Biological Engineering : 日本エム・イー学会誌，vol.50，no.6，pp.666–674，dec 2012． http://ci.nii.ac.jp/naid/40019607522/

[11] A.A. Cuellar, C.M. Lloyd, P.M.F. Nielsen, D.P. Bullivant, D.P. Nickerson, and P.J. Hunter, “An overview of cellml 1.1, a biological model description language.,” Simulation, vol.79, no.12, pp.740–747, 2003.

[12] “Cellml model repository,” 2011. http://models.cellml.org/cellml

[13] 天野　晃，“Program code generation system for multi-scale biological function models that require complex coupling calculation schemes,” 第 53 回生体医工学会大会 OS2-02-2, p.105, 2014.

[14] T.J. Hund and Y. Rudy, “Rate dependence and regulation of action potential and calcium transient in a canine cardiac ventricular cell model,” Circulation, vol.110, no.20, pp.3168–3174, Nov. 2004. http://dx.doi.org/10.1161/01.CIR.0000147231.69595.D3