

自動チューニング化手法

井上裕太

平成 29 年 12 月 29 日

目 次

1	序論	2
1.1	神経科学においてシミュレーションを行う意義	2
1.2	神経回路シミュレーションの高速化・最適化への需要	2
1.3	先行研究	3
1.3.1	宮本さんの	3
1.3.2	片桐先生の	3
1.4	研究の目的と手法	3
1.5	本論文の構成	3
2	シミュレーションモデル	4
2.1	Hodgkin Huxley モデル	4
2.2	Purkinje Cell モデル	4
2.3	ベンチマークモデル	4
3	シミュレーション環境	4
3.1	京コンピュータ	4
3.1.1	CPU アーキテクチャ	4
3.1.2	キャッシュ・メモリ	4
3.2	研究室クラスター	4
3.2.1	CPU アーキテクチャ	4
3.2.2	キャッシュ・メモリ	4
4	自動チューニングスクリプトと MOD トランスパイラの構築	4
4.1	環境設定スクリプト	5
4.2	シミュレータ	5
4.3	トランスパイラ	5

5	シミュレーション結果	5
6	考察	5
7	結論	5

図目次

表目次

1 序論

1.1 神経科学においてシミュレーションを行う意義

- ・ボトムアップアプローチ

- ・また, 当研究室ではそうしたモデル構築のために実験などを行い, そうしたデータを元に様々なシミュレーション系を構築してきた

脳機能の理解を目的として, スーパーコンピュータを用いた神経回路のシミュレーションが行われている. また, 消費電力やシミュレーションの割り当て時間といったリソースの問題やリアルタイムデータ同化への需要からシミュレーションの高速化・最適化が求められている.

また, 現代の計算機にも多様な種類が存在し, それぞれに対する最適化も個別に行われてきた. 本研究の目的はそれぞれの細胞モデルのシミュレーションコードを個々のアーキテクチャに合わせて, 自動又は半自動的に最適化を行う手法を確立することである.

1.2 神経回路シミュレーションの高速化・最適化への需要

- ・神経回路シミュレーションには非常に大きな計算力が必要である一方で, こうした神経回路シミュレーションには非常に大きな計算能力が必要とされてきた.

本研究はスーパーコンピュータ京に関連するポスト京プロジェクトの一環として行われているが,

スーパーコンピュータを用いてもなお計算には多くの時間がかかっている.

そうした状態を踏まえ, 系の構築だけでなくシミュレーション自体の高速化・最適化が求められている.

- ・神経回路シミュレーションの最適化の難しさしかし、神経細胞には様々な種類のも
のが存在するため、個々の神経細胞のイオンチャンネルのモデルを最適化された形
で実装するために、これまでそれぞれのモデルに対して多大な努力が行われてきた。
- ・本研究の意義そこで、本研究では個々のイオンチャンネルモデルを自動で最適化す
るソフトウェアを作成することで、これまで人の手で逐次行われてきた最適化の汎
用化を目指す。

1.3 先行研究

1.3.1 宮本さんの

すごい

1.3.2 片桐先生の

すごい

メモリが大事

1.4 研究の目的と手法

高速化・最適化への需要への項で述べたように、本研究は個々の神経細胞のイオンチャンネルモデルに対し汎用的な最適化手法を開発することである。

神経回路シミュレーションを行うソフトウェアは多数存在するが、本研究では先行研究で用いられていた NEURON というソフトウェアを利用する。

NEURON では、神経細胞のモデルとそれぞれの細胞の関係を微分方程式の形でモデルファイル (MOD ファイル) として記述することができ、nmodl というトランスパイラが MOD ファイルを C ファイルに変換することで実行している。

先行研究では、この生成された C ファイルに着目し手動での最適化を行っていたが、本研究では C ファイルの生成と実行、結果の集約を自動で行うことで複数のパラメータを試し、シミュレーションを実行する上で最適なパラメータを選択することを目指す。

1.5 本論文の構成

本論文は全 6 章から構成されている。

本章では本研究の背景と目的を示した。

第 2 章では、本研究が対象とする神経回路シミュレーションの系、そしてシミュレーションを行う環境について述べる。

第3章では, 本研究で作成したプログラムについての詳細を述べる.
第4章では, シミュレーションの結果を示す.
第5章では, シミュレーション結果の考察を述べる.
第6章では, 本研究のまとめ, 成果を示した上で将来の課題について述べる.

2 シミュレーションモデル

2.1 Hodgkin Huxley モデル

2.2 Purkinje Cell モデル

2.3 ベンチマークモデル

3 シミュレーション環境

3.1 京コンピュータ

3.1.1 CPU アーキテクチャ

3.1.2 キャッシュ・メモリ

3.2 研究室クラスタ

3.2.1 CPU アーキテクチャ

3.2.2 キャッシュ・メモリ

4 自動チューニングスクリプトとMODトランスパイラの構築

本研究では環境・イオンチャンネルモデルに関わらない自動最適化を目的としている.

そのため, スーパーコンピュータ京・研究室クラスタ以外のマシンを用いる場合においても環境構築, プログラムの修正・実行にかかるコストは最小限になるべきである. 上記の要件を満たすため, 以下にあげる3種類のプログラムを作成した.

また, それぞれのプログラムは主として Python, Shell Script(TODO: reference) を使用している.

4.1 環境設定スクリプト

作成したシミュレータ・トランスパイラはPython(TODO: reference) のモジュールとして作成したが, pip(TODO: reference) のようなモジュール管理ツールが存在しない環境 (スーパーコンピュータ京) においては, モジュールとして公開するだけでは不十分である.

特にスーパーコンピュータ京では, デフォルトの Python(TODO: reference) のバージョンが2.6.6, sudo 権限を有しないため外部プログラムのインストールが難しいという環境であったため, Pyenv を利用して汎用的な環境を作成することにした.

4.2 シミュレータ

- ・シミュレータの説明

4.3 トランスパイラ

- ・MOD から C のトランスパイラの説明

5 シミュレーション結果

- ・結果を書きます

6 考察

- ・考察を書きます

7 結論

頑張りました

Appendix