

空間形状を持つニューロンモデルを用いた  
神経回路シミュレーション研究

小林 泰良

電気通信大学 大学院 情報理工学研究科  
博士（工学）の学位申請論文

2022年3月

空間形状を持つニューロンモデルを用いた  
神経回路シミュレーション研究

博士論文審査委員会

情報・ネットワーク工学専攻 准教授 山崎 匡 (主査)

情報・ネットワーク工学専攻 准教授 龍野 智哉

情報・ネットワーク工学専攻 教授 山本 野人

基盤理工学専攻 准教授 松田 信爾

九州工業大学 大学院 生命体工学研究科

人間知能システム工学専攻 准教授 立野 勝巳

著作権所有者

小林泰良

2022年

# Study on neural network simulation using neuron models with morphological structure

Taira Kobayashi

## Abstract

In the central nervous system, neurons constitute a network via synapses, and perform information processing by exchanging spikes. In neuroscience, it is generally assumed that individual neurons are simple elements that only receive/send spikes from/to other neurons, whereas complex brain functions emerge from networks of neurons. In contrast, recent studies indicate that even a single neuron can perform complex information processing by harnessing its morphological structure. Therefore, it is important to investigate functional roles of microscopic morphological structure of individual neurons on mesoscopic network dynamics and macroscopic brain functions.

Network simulation with single-compartment models, which are neuron models without morphological structure, has already reached the human scale by using the supercomputer Fugaku. However, if multi-compartment models that consider the morphological structure explicitly are used, the computational load will increase 100 times or more. To address this high computational load on modern supercomputers, we must employ efficient numerical methods.

In this study, we investigated an efficient numerical method for multi-compartment model simulation on next-generation supercomputers. Because dynamics of a multi-compartment model is described by a system of diffusion equations, an implicit method such as Crank-Nicolson method has been widely used for numerical simulation. In contrast, in this study, we proposed to use an explicit method called Runge-Kutta-Chebyshev (RKC) method. We confirmed that the RKC method implemented on a graphics processing unit (GPU) achieved significant reduction in memory access, and succeeded in demonstrating faster simulation and good scaling performance. Furthermore, using the



RKC method, we were able to conduct large-scale simulation of a cerebellar network model, which demonstrates importance of morphological structure of Purkinje cell dendrites for information processing in the cerebellum, in a reasonable computational time.

In summary, the present study provides an efficient numerical method for multi-compartment model simulation, which will be used as a scaffold to investigate neuronal morphological functions, towards next-generation simulation neuroscience.

## 概要

脳は、ニューロンと呼ばれる神経細胞がシナプスと呼ばれる構造を介して結合したネットワークである。個々のニューロンは単純な素子だが、それが多数集まることでネットワークのダイナミクスとして複雑な機能が生じるというのが、従来の神経科学の中心仮説である。しかし近年の研究では、ニューロン単体でも高度な情報処理能力を持ちうるということが明らかになり、特に、ニューロンの空間形状が重要な役割を担っていることがわかってきた。よって、神経科学分野における今後の重要な課題は、ネットワークダイナミクスや脳機能への単体のニューロンの空間形状が持つ機能的役割の調査である。このようなミクロな脳の空間形状とマクロな脳機能の因果関係を調査するに当たり、形状をパラメータとして任意に仮定・変更可能な計算機シミュレーションは有用な研究手法となる。

2022年現在、神経回路モデルの計算機シミュレーションは、ニューロンの空間形状を無視した場合、スパコン「富岳」を用いることで既にヒトスケールに到達している。しかし、個々のニューロンの空間形状を考慮した場合は、その演算規模が数十倍から数百倍に及ぶため、次世代のスパコンが必要となる。よって、計算機シミュレーションによって脳の空間形状と機能の関係性を考察するためには、シミュレーションの高速化技術が必要となる。

そこで、本研究では、ミクロな脳の空間形状とマクロな脳機能の因果関係を考察するため、2つの研究を行った。1つ目の研究では、次世代のスパコンの利用を見据えて、より一般的な並列演算用ハードウェアであるグラフィックスプロセッシングユニット（GPU）を用いて、空間形状を考慮したニューロンモデル（マルチコンパートメントモデル）のための、高効率な数値シミュレーション法について検討した。マルチコンパートメントモデルの挙動は拡散方程式で記述されるため、その数値シミュレーションには、Crank-Nicolson（CN）法のような陰解法が用いられるが、本研究では拡散方程式向けの陽解法である Runge-Kutta-Chebyshev（RKC）法を適用した。その結果、数値シミュレーションにおいて計算のボトルネックとなりうる、メモリアクセスの大幅な低減をもたらすことを確認し、シミュレーションの高速化と良好なスケーリング性能を示すことに成功した。

2つ目の研究では、RKC法によってマルチコンパートメントモデルを用いた小脳神経回路モデルのシミュレーションを高速に実施し、個々のニューロンの形状変化とネットワークダイナミクスとの関係性を調査、脳機能に与える影響について考察した。特に、小脳神経回路の情報処理機構における、プルキンエ細胞の神経変性がプルキンエ細胞間の活動を非同期化し、深部小脳核ニューロンの活動を変調させることで運動失調を引き起こす可能性を示唆する結果を得た。

本研究は、次世代のシミュレーション神経科学の確立に向けて、ニューロンの空間形状の機能的役割を解明するための、高性能な数値シミュレーション手法を提供するものである。

# 目次

第 1 章	序論	3
1.1	本論文の構成 . . . . .	6
第 2 章	基礎知識と手法	7
2.1	生理学的背景とニューロンモデル . . . . .	7
2.2	マルチコンパートメントモデルの数値計算法 . . . . .	15
2.3	GPU による計算の並列化とシミュレーション環境 . . . . .	17
第 3 章	陽解法によるマルチコンパートメントモデルシミュレーションの高速化	23
3.1	はじめに . . . . .	23
3.2	方法 . . . . .	25
3.3	結果 . . . . .	33
3.4	考察 . . . . .	44
3.5	結論 . . . . .	48
第 4 章	マルチコンパートメントモデルシミュレーションによるプルキンエ細胞の空間形状が持つ機能的役割の検証	49
4.1	はじめに . . . . .	49
4.2	方法 . . . . .	50
4.3	結果 . . . . .	57
4.4	考察 . . . . .	61
4.5	結論 . . . . .	65
第 5 章	総括	67

付録	71
A.1 本研究で使⽤したマルチコンパートメントモデル . . . . .	71
A.2 RKC 法のア⽼リズムと安定性 . . . . .	83
A.3 CUDA プログラミングの詳細 . . . . .	89
参考文献	97

# 第 1 章

## 序論

脳は、呼吸などの生命維持活動から、記憶・学習・思考・情動・意識といった高次機能まで担う器官である。ヒトを含む一部の霊長類は、文字や論理など抽象的な概念を扱えるほど高度な情報処理能力を持ち、これらの機能の動作原理を解明することが神経科学の最大の課題である。近代の神経科学は、19 世紀の Camillo Golgi と Santiago Ramón y Cajal らによる解剖学的知見から確立された [1]。20 世紀には、電子顕微鏡などの開発によりさらに精緻な解剖学的知見が得られたが、脳の構造や情報処理の機序は非常に複雑であり、その本質を理解するためには、解剖学・発生学・生理学・薬理学・心理学などの伝統的な実験科学をはじめ、遺伝子工学や計算機科学、人工知能といった様々な分野を統合した学際的な視点から理解を試みる必要があると考えられはじめた。21 世紀には学際的なプロジェクトが各国において本格的に始動し、2005 年にはヒト脳をスーパーコンピュータ上に構築することを目標とした欧州の Blue Brain Project [2]、2013 年には脳機能障害の治療方法の改善や人間の思考、学習、記憶などへの理解を試みるアメリカのブレイン・イニシアチブ [3]、そして 2014 年には日本においても「革新的技術による脳機能ネットワークの全容解明プロジェクト」が発足され、多分野の横断的連携による脳理解へのアプローチが開始されている。

このような取り組みの 1 つとして、脳を計算論的に理解する試みである、計算神経科学がある。脳を構成する神経細胞（ニューロン）とその結合（シナプス）を実験データに基づき微分方程式として記述、それらを統合したネットワークモデルをコンピュータに解かせることで脳内ダイナミクスを再現・可視化し理解することを目指している。計算論的手法による取り組みは Warren Sturgis McCulloch と Walter J. Pitts による形式ニューロンモデル [4] や Frank Rosenblatt によるパーセプトロン [5]、Alan Hodgkin と Andrew Huxley

による Hodgkin-Huxley model モデル [6] などの単一のニューロンの数理モデル化から始まり、ニューロン間のシナプス結合を導入したネットワークモデルの数値シミュレーション研究に発展した [7]。21 世紀に入ると、コンピュータの性能向上がネットワークモデルの大規模化をもたらした [8-13]、2021 年現在ではスパコン「富岳」においてヒトスケール全脳シミュレーションを達成するまでに至っている [14]。このようなシミュレーション研究はヒトスケールの脳内のダイナミクスを可視化し、さらに脳機能障害などの病態を再現することで、神経回路の構造とダイナミクス、脳機能の関係性の調査などへの応用が期待できる。

一方で、五十嵐らのシミュレーション [14] に用いられたニューロンモデルは、ニューロンの空間形状を考慮していない簡易的なモデル（シングルコンパートメントモデル）である。ニューロンの空間形状の重要性は近年の研究により多数報告されており、他のニューロンからの刺激シーケンスの弁別や非線形な論理演算の演算などの単体のニューロンレベルにおける計算能力の向上から、脊髄小脳変性症 I 型をはじめとした神経変性疾患によるニューロンの変性と運動失調のような脳機能障害の相関性などが明らかにされている [15-20]。計算機シミュレーションはニューロンの空間形状をパラメータとして任意に操作でき、且つ脳の活動を可視化することでその因果関係を調査するために有用であることから、次世代の計算神経科学分野の目標としてニューロンの形状まで考慮したモデル（マルチコンパートメントモデル）を用いたヒトスケール全脳シミュレーションが期待される。マルチコンパートメントモデルはシングルコンパートメントモデルの数十から数百倍の計算量を要するが、ムーアの法則に従い集積回路あたりの部品数が順調に 2 年で 2 倍ずつ増えることで、2030 年代にはマルチコンパートメントモデルによるヒト全脳シミュレーションを達成する規模の計算機環境の構築が予想される。さらに言えば、十数年後にはニューロン間のシナプス結合を解析するコネクトーム研究による知見から、霊長類における詳細な配線データが得られ、精緻な神経回路マップを持った全脳モデルの完成が見込まれる。このような詳細な空間形状を持つネットワークモデルは、ニューロンやネットワークの微細な構造の変化がネットワークダイナミクスにどのような影響をもたらすか、脳機能を変調させるか、その因果関係の調査や病態の予測などを可能とし、脳の動作原理解明に重要な進展をもたらすことが期待される。

2022 年現在のマルチコンパートメントモデルを用いた計算機シミュレーションの進展として、各脳部位のネットワークモデルの開発 [18, 19, 21] や、スパコンや GPU のような最新の並列計算用ハードウェアをサポートした新たな神経回路シミュレータが開発されていることが挙げられる [22, 23]。しかしながら、これら最新のシミュレータを用いても、上記のような大規模ネットワークモデルにおいて、学習のような長時間タスクのシミュレーションには

莫大な計算時間を要するため現実的ではないといった問題が存在する。これは現状のシミュレータが数値計算法の改良に未着手であり、最新のハードウェアの性能を十全に引き出せていないことに起因すると推測される。一般的にマルチコンパートメントモデルのシミュレーションには、メモリアクセスを多用する連立方程式の求解が必要な陰解法（Crank-Nicolson法など [24]）が用いられるが、これは最新の計算機が抱える、メモリのデータ転送速度がプロセッサの計算速度に対して非常に遅い「メモリウォール問題 [25]」に直結する。さらに言えば、このメモリウォール問題は次世代のスパコンにおいてより顕著に現れると予想される。以上より並列計算用ハードウェアに合わせたマルチコンパートメントモデル用のシミュレータの準備、特に次世代のスパコンではより顕著に現れるであろうメモリウォール問題を回避し、良好なスケーリング性能が得られるような数値計算法の改良が求められる。

このような背景から、今後十年間の神経解剖学や計算機科学分野の進展を見据え、計算神経科学者が達成すべき具体的な目標は、次世代のスパコンの性能を十全に引き出しヒトスケール全脳シミュレーションを達成するシミュレータを準備すること、そして現在行える最大規模でミクロスケールな脳の空間形状とメゾスケールなネットワークダイナミクスやマクロスケールな脳機能に関する基礎的な知見を提供することである。

そこで本研究では、「ニューロンの空間形状を考慮した計算機シミュレーションの高速化」と「ニューロンの空間形状と脳機能の関係性の考察」に関する2つのテーマで研究を行った。テーマ1では、マルチコンパートメントモデルに対して、連立方程式の求解を必要としない陽解法の適用を試みた。具体的には、数値解析分野で用いられる偏微分方程式用の陽解法を神経科学分野のモデルに適用し、その性能を一般的な並列計算用ハードウェアであるGPU上で評価した。特に、複雑な空間形状を持つ小脳のプルキンエ細胞を中心に計算時間、メモリ性能、スケーリング性能をテストし、最終的には小脳の神経回路モデルにおける性能評価を行った。テーマ2では、テーマ1で実装した小脳神経回路モデルを用い、ニューロンの空間形状が持つ機能的役割について検討した。特に、神経変性疾患の一種である脊髄小脳変性症I型にみられるプルキンエ細胞の樹状突起縮退を再現し、ニューロンの空間形状変化がネットワークダイナミクスに与える影響について調査し、脳機能との関連性について考察した。最後に、得られた結果と今後の展望についてまとめた。



## 1.1 本論文の構成

第1章（本章）では、論文全体にかかわる研究の背景と目的について述べた。

第2章では、神経科学全般とニューロンの数理モデルならびに数値計算法に関する基礎的知識について述べる。また、GPUによる計算の並列化手法と本研究で用いたシミュレーションの環境についても記述する。

第3章では、テーマ1のマルチコンパートメントモデルに対する陽解法の適用に関する研究について述べる。特に、特異な空間形状を持つニューロンが存在する小脳に着目し、小脳ニューロンのマルチコンパートメントモデルをGPU上に実装、陽解法を適用しGPU上での性能について一般的な数値計算法である陰解法と比較・評価する。さらに、小脳神経回路モデルを作成し、大規模な神経回路モデルにおけるRKC法の適用可能性について検討する。

第4章では、第3章で実装した小脳神経回路モデルを用い、ニューロンの空間形状が持つ機能的役割について検討する。特に、脊髄小脳変性症I型で顕著な神経変性が見られる小脳プルキンエ細胞に着目し、プルキンエ細胞の形状変化がネットワークのダイナミクスに与える影響について調査する。さらに、具体的な運動タスクとして小脳が重要な役割を果たす瞬目反射条件づけのシミュレーションを行い、脳機能に対するニューロンの空間形状が持つ機能的役割について考察する。

第5章では、本研究で得られた結果についてまとめ、今後の展望について述べる。

## 第2章

# 基礎知識と手法

### 2.1 生理学的背景とニューロンモデル

#### 2.1.1 脳の構造と機能

脳は、呼吸などの生命維持活動から、記憶・学習・思考・情動・意識といった高次機能まで担う器官であり、ニューロンと呼ばれる神経細胞がシナプスと呼ばれる結合を介して情報を伝達する回路構造によって構成されている。ヒトの脳の大部分は意識や情動といった高次脳機能を司る大脳と、運動調整や運動学習などを司る小脳、他の神経系との情報をリレーする脳幹、抹消神経との連絡通路である脊髄によって構成されている（図 2.1）。

大脳には外層部分にあたる大脳皮質と深部にあたる海馬、扁桃核、大脳基底核が存在し、それぞれが異なる役割を持つとされている。大脳皮質は、知覚や推論、随意運動などを司り、カラムと呼ばれる規則正しい構造を持つ。大脳皮質は、その部位が果たしている機能的役割に応じて、運動野や感覚野といった名称が振り分けられている。例えば、頭頂葉と前頭葉の間に存在する一次運動野は運動の計画などの役割を持つ。大脳深部に存在する海馬は記憶の形成や貯蔵、扁桃核は情動に起因した反応の形成に関与する。また、大脳基底核は運動の制御に関与するとされる。大脳、特に大脳皮質は高度な情報処理機能を持つことから、霊長類において発達が著しく、ヒトの大脳は脳の体積の約 80 % を占めているとされる。

小脳は脳の体積のうち 20 % 程度を占めているとされるが、脳内に存在する細胞の約 80 % は小脳に存在するとされ、非常に密度の高い部位である。しかし、小脳を構成するニューロンの種類は比較的少なく、内部を構成する神経回路の構造は比較的解明が進んでいる（図

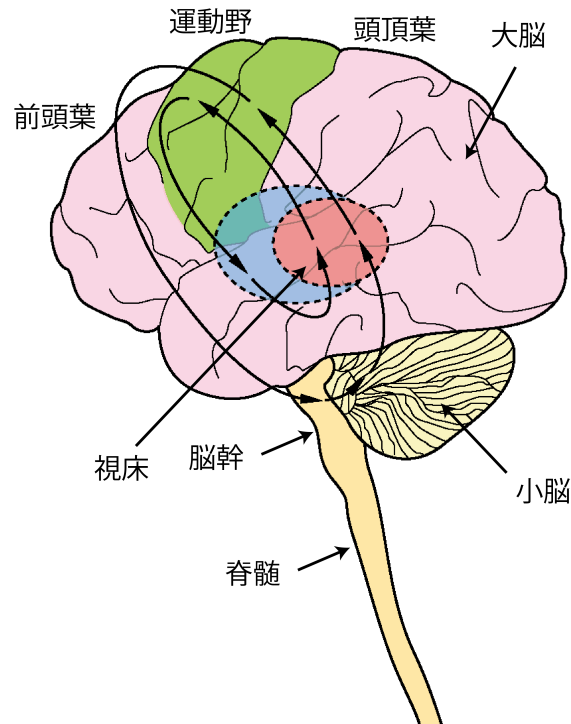


図 2.1: ヒトの中樞神経系。中樞神経系は大きく分けて、「高次脳機能を司る大脳」と「運動の学習や調整を行う小脳」、「間脳、中脳、橋、延髄からなり、他の神経系との情報をリレーする脳幹」、「皮膚や筋肉などの抹消の神経と脳との情報を連絡する脊髄」から構成されている。また、大脳皮質は小脳へと信号を送り、小脳は視床を介して大脳皮質へと信号を出力するループ構造を持つ。

2.2)。小脳は、機能的に異なる 3 つの部位によって構成されており、それぞれ前庭小脳、脊髄小脳、大脳小脳と呼ばれる。前庭小脳は前庭や視覚情報、頸部体性感覚情報を前庭神経核から受け、平行調節や前庭反射、眼球運動などに関する情報処理を行う。脊髄小脳は体幹や四肢の体性感覚情報や運動野からの信号を受け、脊髄や脳幹のニューロンへと出力し、姿勢や四肢の動きの制御に関与する。大脳小脳は大脳皮質からの情報を橋核を経由して受け、運動野へと出力をすることで、運動の計画や実行に関与する。また、大脳小脳はヒトなどの霊長類で発達しており、認知機能にも関与している可能性が報告されている [1]。橋核を経由して入力された信号は、苔状線維として小脳の顆粒細胞へと伝えられ、顆粒細胞は平行線維と呼ばれる長い軸索を用いてプルキンエ細胞に情報を伝える。深部小脳核のニューロンは、苔状線維からの直接興奮性の信号を受けると同時に、小脳回路内で処理された情報をプルキンエ細胞から抑制性の信号として受け取ることで、情報の比較を行い、他の脳部位へと出力す

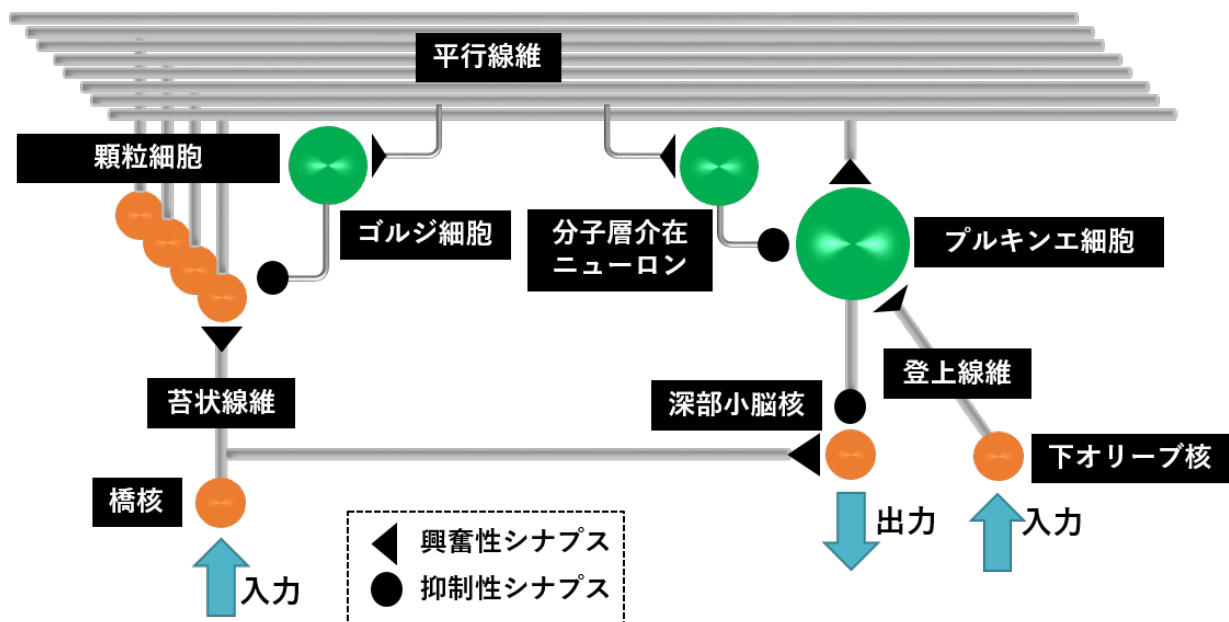


図 2.2: 小脳の神経回路構造の模式図。橋核および下オリーブ核から入力、深部小脳核から出力が行われる 2 入力 1 出力の回路構造となっている。橋核からの興奮性入力は苔状線維を介して顆粒細胞へ入る。顆粒細胞からの興奮性入力は平行線維を介してプルキンエ細胞へ入り、深部小脳核のニューロンを抑制することで小脳回路で処理された情報を出力する。また、ゴルジ細胞と分子層介在ニューロンは平行線維入力を介して顆粒細胞とプルキンエ細胞をそれぞれ抑制する。下オリーブ核のニューロンは登上線維入力を介してプルキンエ細胞を興奮させ、運動学習における教師信号を与える。

る。また、下オリーブ核のニューロンは抹消や大脳皮質から感覚・運動情報を受け、プルキンエ細胞に投射するが、下オリーブ核ニューロンの発火の頻度は低く、運動学習に必要な事象の検知に特化しているとされる。

これまでの解剖学的研究や生理学的研究によって、神経回路の構造や各脳部位の機能に関して、ある程度の知見が得られており、現在は電子顕微鏡などを用いたさらに詳細なスケールの研究が行われている。一方で、脳の機能原理には未だ不明な部分が多く、現在は解剖学的知見を基に、磁気共鳴機能画像法 (fMRI) やカルシウムイメージング、数値シミュレーションなど脳内ダイナミクスを可視化する技術を用いた脳機能の原理解明などが試みられている。

神経回路の構造と脳機能の相関的な関係を示す事例として、遺伝子疾患の一種である神経変性疾患による神経回路構造の変性と脳機能障害が挙げられる。例えば、脊髄小脳変性症 I 型では、小脳のプルキンエ細胞層に萎縮が見られ、同時に衝動性眼球運動や眼筋麻痺、運動

失調など小脳の主要な機能である運動制御への影響が現れる（図 2.3）。また、大脳皮質基底

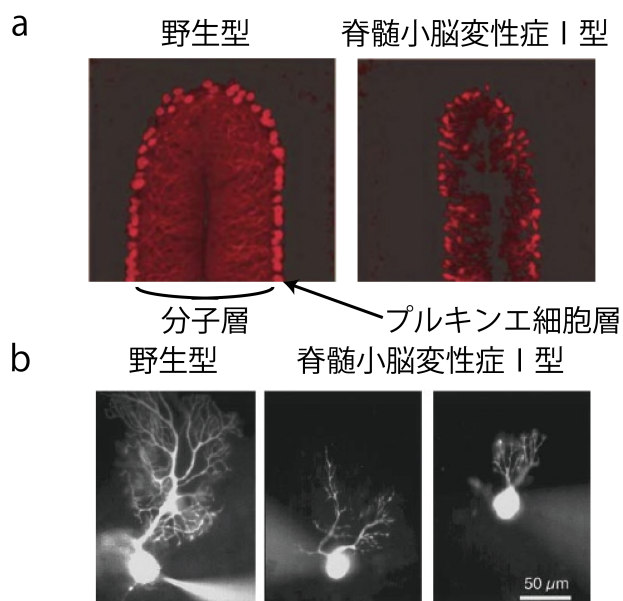


図 2.3: 脊髄小脳変性症 I 型モデルマウスにおけるプルキンエ細胞の神経変性 (a は [26]、b は [27] より改変)。**a**: 18 週間の野生型 (左) とゲノム改変を行ったモデルマウス (右) におけるプルキンエ細胞層および分子層の様子。脊髄小脳変性症 I 型によるプルキンエ細胞層の萎縮が確認できる。**b**: 単体のプルキンエ細胞の変性の様子。野生型 (左) において発達している樹状突起がゲノム改変を行ったモデルマウス (中および右) では樹状突起が縮退している様子を確認できる。

核変性症やアルツハイマー型認知症 (海馬変性) など、小脳以外の領域においても神経変性疾患は存在し、運動障害や記憶障害などの脳機能障害が報告されている。神経細胞の変性が起こる遺伝子学的機序は遺伝子工学研究によりある程度判明しつつあるが、神経変性による神経回路構造の変化と運動失調の因果関係については未だ研究が進んでいない。生理学的実験により、ニューロンの空間形状を操作しつつ、ネットワーク全体のニューロンのダイナミクスを記録・解析することは困難であるため、この因果関係の解明には、ニューロンの空間形状をパラメータとして仮定・変更可能な計算機シミュレーションによる研究が有用である。次節では、ニューロンの電気的特性とモデル化の説明を行う。

## 2.1.2 ニューロンの電気的特性と数理モデル

ニューロンは脳を構成する細胞であり、他のニューロンから神経スパイク（発火）と呼ばれる電気インパルス信号を受け取り、信号を積算、閾値を超えると他のニューロンへスパイクを発射する。即ち、ニューロンは電気信号によって相互に情報伝達を行う電気的な素子である。また、ニューロンは特徴的な空間形状を持ち、部位に応じた役割を持つ（図 2.4）。一般的に、ニューロンは他のニューロンから入力を受け取る樹状突起、入力情報を積算し、閾値を超えるとスパイクを生成する細胞体、他のニューロンにスパイクを伝達する軸索によって構成されている。ニューロン間でスパイクを用いた情報伝達をする部位をシナプスと呼び、情報を送る側のニューロンをシナプス前細胞、情報を受け取る側のニューロンをシナプス後細胞と呼ぶ。

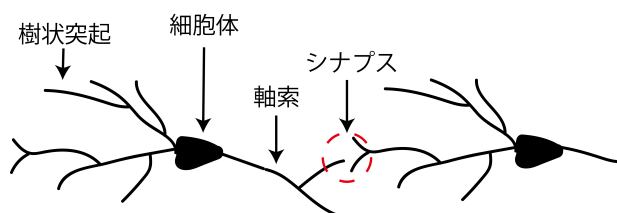


図 2.4: ニューロンの模式図。単一のニューロンは一般的に、樹状突起、細胞体、軸索から構成される。細胞体で生成された神経スパイクは軸索を通り、シナプス結合を介して他のニューロンの樹状突起に伝達される。

ニューロンの電気的な特性を記述するうえで重要な要素として、細胞膜とイオンチャンネルが挙げられる。ニューロンの細胞膜は脂質二重層の薄い膜で構成されており、絶縁性を持つ。細胞膜の内外には各種イオン ( $\text{Na}^+$ 、 $\text{K}^+$ 、 $\text{Ca}^{2+}$  など) が存在し、細胞膜に埋め込まれたイオンポンプと呼ばれるたんぱく質が ATP の化学エネルギーを消費して、各種イオンに濃度勾配を生じさせ、電位差が発生する。この電位差を膜電位と呼び、通常は細胞外よりも細胞内の方が電位が低く、静止状態では  $-65 \text{ mV}$  程度の差となる。また、細胞膜には特定のイオン種のみを選択的に透過させるイオンチャンネルと呼ばれるたんぱく質が存在する。イオンチャンネルは膜電位や  $\text{Ca}^{2+}$  濃度などに依存して開口し、それぞれのイオンは電気化学ポテンシャルに従って細胞膜を通過することで、電位差に変化が生じる。特に、 $\text{Na}^+$  チャンネルはある程度膜電位が上昇すると瞬間的に開口し、 $\text{Na}^+$  が細胞外から細胞内へ流れ込むことでスパイク状

の膜電位変化が生成される。K<sup>+</sup> チャンネルは Na<sup>+</sup> チャンネルに遅れて開口し、細胞内から細胞外へ流れ出ることによって、膜電位を下げ、スパイク生成後の膜電位を生成前の値まで戻す役割を持つ。このようなアクティブなイオンチャンネルに対して、開閉が膜電位などに依存せず、膜電位を一定の値（静止電位）まで戻す役割を持つパッシブなチャンネル（リークチャンネル）も存在する。

Hodgkin と Huxley はヤリイカの巨大軸索で電流を測定し、このスパイク生成のメカニズムを発見し、数理モデル（Hodgkin–Huxley (HH) モデル）として発表した [6]。HH モデルはニューロンの細胞膜をキャパシタ、アクティブなイオンチャンネルを可変抵抗に見立て電気回路として表現し（図 2.5）、次式のように定式化している。

$$c_m \frac{dv(t)}{dt} = I(t) - g_l(v(t) - E_l) - g_{Na}(t)(v(t) - E_{Na}) - g_K(t)(v(t) - E_K) \quad (2.1)$$

$c_m$  は膜のキャパシタンス、 $u(t)$  は膜電位、 $I(t)$  は入力電流、 $g_l$  と  $g_{Na}(t)$ 、 $g_K(t)$  はそれぞれ、リークチャンネルと Na<sup>+</sup> チャンネル、K<sup>+</sup> チャンネルのコンダクタンスである。 $E_l$  と  $E_{Na}$ 、 $E_K$  は反転電位と呼ばれるチャンネル固有の電位であり、この電位を境に細胞内外のイオンの電気化学ポテンシャルが逆転し、電流の向きが反転する。また、Hodgkin と Huxley による電位依存性のアクティブな Na<sup>+</sup> と K<sup>+</sup> チャンネルのコンダクタンス  $g_{Na}(t)$  と  $g_K(t)$  の挙動は次式のように定式化され、HH 型方程式と呼ばれる。

$$g_{Na}(t) = \bar{g}_{Na} m^3 h \quad (2.2)$$

$$g_K(t) = \bar{g}_K n^4 \quad (2.3)$$

ここで  $\bar{g}_{Na}$  と  $\bar{g}_K$  はそれぞれのチャンネルの最大コンダクタンスである。通常、チャンネルの一部がブロックされており、最大のコンダクタンスで電流が流れることはない。そこでゲート変数  $m$  と  $h$ 、 $n$  によってチャンネルが開いている確率を表現し、チャンネルに流れる電流の量を制御する。ゲート変数 ( $x$ ) は膜電位に依存して次式のように変化する。

$$\frac{dx}{dt} = -\frac{1}{\tau_x(v)}(x - x_0(v)) \quad (2.4)$$

$$x_0(v) = \frac{\alpha(v)}{\alpha(v) + \beta(v)} \quad (2.5)$$

$$\tau_x(v) = \frac{1}{\alpha(v) + \beta(v)} \quad (2.6)$$

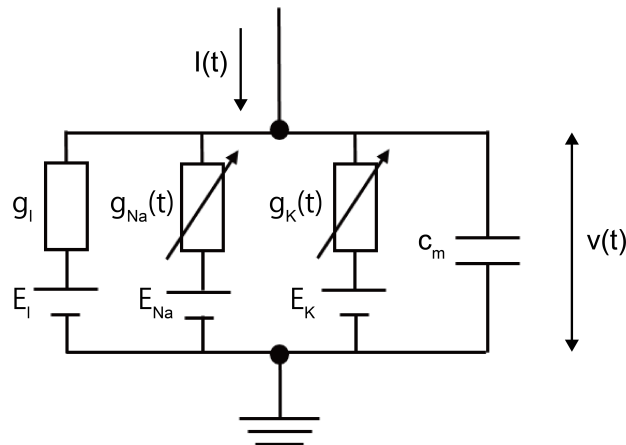


図 2.5: HH モデルにおける細胞膜の等価回路。Na<sup>+</sup>、K<sup>+</sup> のアクティブなチャンネルを可変抵抗、パッシブなチャンネルを固定抵抗、細胞膜をキャパシタとして表現した並列回路となっている。

ゲート変数 ( $x$ ) は膜電位  $v$  に対して、時定数  $\tau_x(v)$  で  $x$  が  $x_0(v)$  に近づくことを示しており、 $\alpha(v)$  と  $\beta(v)$  は生理学的実験などによってフィッティングされる。また、イオンチャンネルは Na<sup>+</sup> と K<sup>+</sup> 以外にも多数存在し、同じイオンのチャンネルによっても異なるダイナミクスを持つサブタイプが存在する。

HH モデル (式 (2.1)) はニューロンが持つイオンチャンネルの特性を考慮しているがニューロンそのものの空間形状は考慮していない。一方で、イオンチャンネルは通常、ニューロンの部位 (樹状突起や細胞体、軸索) によって異なる密度で存在している。例えば、小脳のプルキンエ細胞の細胞体には、Na<sup>+</sup> や Ca<sup>2+</sup> のチャンネルが存在するのに対し、樹状突起上にはカルシウム依存性の K<sup>+</sup> チャンネルが存在する。また、プルキンエ細胞が受ける登上線維からの興奮性シナプス入力、細胞体上でのスパイクを生成させると同時に、Ca<sup>2+</sup> の細胞内への流入を引き起こし、流入した Ca<sup>2+</sup> が樹状突起へと伝搬、カルシウム依存性の K<sup>+</sup> チャンネルを起動させるといった空間形状に起因したメカニズムが知られている [28]。

このような複雑な細胞内ダイナミクスを計算するために、ニューロンの空間形状を考慮した数理モデルとして、マルチコンパートメントモデルが提案されている (図 2.6) [29]。はじめに、ニューロンを半径や膜キャパシタンス、イオンチャンネルの密度 (コンダクタンス) といったパラメータを持つ複数のケーブルに分割する。各ケーブルが持つ膜電位  $v(x, t)$  は時間



と空間に関する拡散方程式によって次式のように記述される。

$$c_m \frac{\partial v(t, x)}{\partial t} = \frac{a}{2R} \frac{\partial^2 v(t, x)}{\partial x^2} + I_{\text{ext}}(t) - g_l(v(t, x) - E_l) - \sum_k g_k(t)(v(t, x) - E_k) \quad (2.7)$$

ここで、 $a$  と  $R$  はケーブルの半径と固有の細胞内抵抗、 $g_k$  と  $E_k$  はイオンチャンネルのコンダクタンスと反転電位 ( $k \in \{\text{Na}^+, \text{K}^+, \text{Ca}^{2+}, \dots\}$ )、 $I_{\text{ext}}(t)$  はシナプス入力や電極入力によって生じる外部からの入力電流を表す。式 (2.7) 右辺の第 1 項はケーブル内の電位の拡散を表す拡散項、第 4 項は膜電位とイオンチャンネルのコンダクタンスによって細胞膜を流れる電流に起因する反応項である。よって、式 (2.7) は拡散反応方程式と呼ばれる偏微分方程式となる。

次に、コンピュータは微分項を直接取り扱うことができないため、各ケーブルを空間刻み幅  $\Delta x$  によってコンパートメントと呼ぶ小さなセグメントに分割する。具体的には、数値シミュレーションのために拡散項を  $x$  に関する 2 階の中心差分によって離散化し、次式のよう

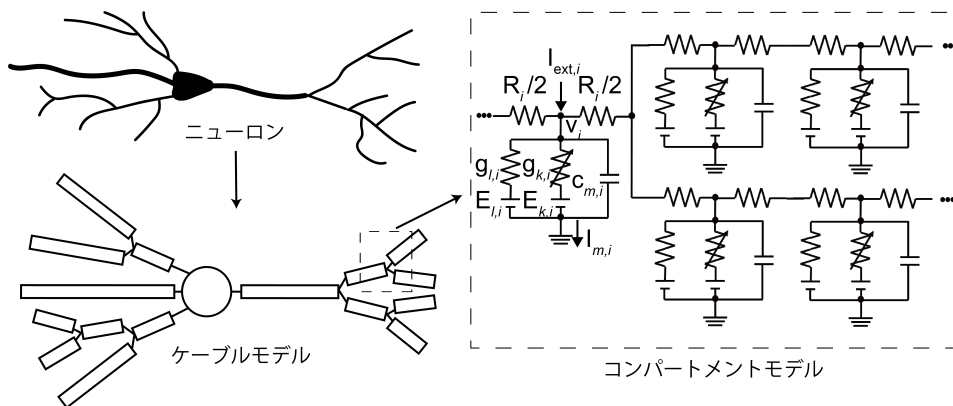


図 2.6: マルチコンパートメントモデルの模式図。マルチコンパートメントモデルを生成するためには、ニューロンを複数のケーブルに分割する。分割されたケーブルをさらにコンパートメントと呼ばれる小さなセグメントに分割する。各コンパートメントの膜電位は隣合うコンパートメントから流入出する電流と、細胞膜を通過するイオン電流によって変化する。コンパートメント間の電流による電位の変化はケーブル方程式 (拡散方程式) によって記述され、イオン電流による電位の変化は HH 型方程式によって記述される。

に展開する。

$$c_m \frac{\partial v_i(t)}{\partial t} = \frac{a}{2R} \frac{v_{i+1}(t) - 2v_i(t) + v_{i-1}(t)}{(\Delta x)^2} + I_{\text{ext}}(t) - g_1(v_i(t) - E_1) - \sum_k g_k(t)(v_i(t) - E_k) \quad (2.8)$$

また、ケーブルが分岐する場合、即ちあるコンパートメント  $i$  が 3 つ以上のコンパートメント  $j$  ( $j = 0, 1, 2, \dots$ ) と接続している場合は、次式のように展開される。

$$c_{m,i} \frac{\partial v_i(t)}{\partial t} = \sum_j \frac{v_j(t) - v_i(t)}{r_{ij}} + I_{\text{ext},i}(t) - g_{1,i}(v_i(t) - E_{1,i}) - \sum_k g_{k,i}(t)(v_i(t) - E_{k,i}) \quad (2.9)$$

ただし、 $g_{k,i}$ 、 $E_{k,i}$ 、 $I_{\text{ext},i}(t)$  はそれぞれ  $i$  番目のコンパートメントが持つイオンチャネルのコンダクタンスと反転電位、外部入力電流を表す。また、 $r_{ij}$  は  $i$  番目のコンパートメントと  $j$  番目のコンパートメントを繋ぐ内部抵抗であり、次式で定義される。

$$\frac{1}{r_{ij}} = \frac{1}{2} \left( \frac{a_i}{2R_i(\Delta x)^2} + \frac{a_j}{2R_j(\Delta x)^2} \right) \quad (2.10)$$

ここで、 $a_i$  と  $R_i$  ならび  $a_j$  と  $R_j$  はそれぞれ  $i$  番目と  $j$  番目のコンパートメントの半径と固有の細胞内抵抗である。

以上より、式 (2.7) は線の方法 (method of lines) によって常微分方程式 (式 (2.8)、(2.9)) のシステムとして記述され、次節で説明する数値計算法によって解くことができる。

## 2.2 マルチコンパートメントモデルの数値計算法

### 2.2.1 陽解法

常微分方程式の数値計算を実行するために用いられる最も一般的な数値計算法は陽解法である。陽解法は、現在の時刻  $t$  の状態のみを用いて時間刻み  $\Delta t$  後の時刻  $t + \Delta t$  の状態を計

算する手法である。即ち、式 (2.8) の左辺を陽的に離散化すると、次式のようにになる。

$$c_m \frac{v_i(t + \Delta t) - v_i(t)}{\Delta t} = \frac{a}{2R} \frac{v_{i+1}(t) - 2v_i(t) + v_{i-1}(t)}{(\Delta x)^2} + I_{\text{ext}}(t) - g_1(v_i(t) - E_1) - \sum_k g_k(t)(v_i(t) - E_k) \quad (2.11)$$

このような 1 階の前進差分を用いた離散化による計算方法を陽的オイラー法と呼び、誤差は  $O(\Delta t) + O((\Delta x)^2)$  となる [29]。陽的オイラー法は過去の変数の値のみを使用するため、実装や演算が容易である一方で、数値計算を安定化させるためには  $\Delta t \leq RC_m(\Delta x)^2/a$  を満たす必要がある。マルチコンパートメントモデルはその性質上、ニューロンの詳細な空間形状を再現するために空間刻み  $\Delta x$  を小さくとることがあるため、計算の安定化には極めて小さな  $\Delta t$  が必要となる。よって、マルチコンパートメントモデルのシミュレーションには次節で説明する安定的な数値計算を可能とする陰的解法が一般的に用いられる。

## 2.2.2 陰解法

前節で述べたように、偏微分方程式はモデルの空間的な離散化により、数値的に不安定になることがある。そこで、時刻  $t + \Delta t$  における状態を計算するために、同じく時刻  $t + \Delta t$  の変数を用い、安定的な計算を行う陰解法が広く用いられる。即ち、式 (2.8) の左辺を陰的に離散化すると次式のようにになる。

$$c_m \frac{v_i(t + \Delta t) - v_i(t)}{\Delta t} = \frac{a}{2R} \frac{v_{i+1}(t + \Delta t) - 2v_i(t + \Delta t) + v_{i-1}(t + \Delta t)}{(\Delta x)^2} + I_{\text{ext}}(t + \Delta t) - g_1(v_i(t + \Delta t) - E_1) - \sum_k g_k(t + \Delta t)(v_i(t + \Delta t) - E_k) \quad (2.12)$$

このような 1 階の後退差分を用いた離散化による計算方法を陰的オイラー法と呼び、誤差は陽的オイラー法と同様に  $O(\Delta t) + O((\Delta x)^2)$  となる [29]。時刻  $t$  における膜電位  $v_i(t)$  は  $v_{i-1}(t + \Delta t)$ 、 $v_i(t + \Delta t)$ 、 $v_{i+1}(t + \Delta t)$  に影響するため、どのような時間刻み  $\Delta t$  を用いても、 $\mathbf{v}(t + \Delta t) = \{\dots, v_{i-1}(t + \Delta t), v_i(t + \Delta t), v_{i+1}(t + \Delta t), \dots\}$  の連立方程式を解くことで、安定的な計算を行うことができる。

また、式 (2.8) の左辺を時刻  $t + \Delta t/2$  において中心差分をとることにより、次式のように

離散化することができる。

$$c_m \frac{v_i(t + \Delta t) - v_i(t)}{\Delta t} = \frac{1}{2} \left( \frac{a}{2R} \frac{v_{i+1}(t + \Delta t) - 2v_i(t + \Delta t) + v_{i-1}(t + \Delta t)}{(\Delta x)^2} + I_i(t + \Delta t) \right) + \frac{1}{2} \left( \frac{a}{2R} \frac{v_{i+1}(t) - 2v_i(t) + v_{i-1}(t)}{(\Delta x)^2} + I_i(t) \right) \quad (2.13)$$

ただし、 $I_i(t) = I_{\text{ext}}(t) - g_l(v_i(t) - E_l) - \sum_k g_k(t)(v_i(t) - E_k)$  である。式 (2.13) は陽解法と陰解法の平均値をとるような離散化方法であり、この離散化による計算手法を Crank-Nicolson (CN) 法と呼ぶ。CN 法は数値的に安定的な計算手法であり、その誤差は  $O((\Delta t)^2) + O((\Delta x)^2)$  となる。マルチコンパートメントモデルのシミュレーションには計算の安定性と時間と空間に関する 2 次精度から CN 法が適用されることが多く、時間刻み幅は  $\Delta t = 25 \mu\text{s}$  が適切とされる [24]。

陰解法は時刻  $t + \Delta t$  における状態の計算に未知の変数を用いるため、各時間ステップごとに連立方程式を解かなければならない。このように、陰解法は複雑な計算を要するため、マルチコンパートメントモデルのシミュレーションは一般的に、専用のシミュレータ [22–24, 30] を用いて実行される。また、連立方程式の求解方法として、ガウスの消去法と共役勾配法 (conjugate gradient method、CG 法) が代表として挙げられる。ガウスの消去法は連立方程式の係数を操作し直接的に求解する手法であり、既存のマルチコンパートメントモデル用シミュレータに採用されることが多い [22, 23]。対して、CG 法は残余誤差を徐々に最小化することによって最適解を見つける反復法である。本研究で用いるマルチコンパートメントモデル (3.2.2 節のモデル) は大規模且つ疎な正定値対称行列であり、このモデルに CN 法を適用した際に現れる連立方程式の求解には CG 法を適用した。

## 2.3 GPU による計算の並列化とシミュレーション環境

一般的に、マルチコンパートメントモデルはシングルコンパートメントの数十から数百倍の計算量を必要とする。例えば、小脳のプルキンエ細胞は非常に複雑な空間形状を持つため、そのマルチコンパートメントモデルは 1600 個のコンパートメントによって構成されており [31, 32]、このモデルを含むようなネットワークモデルのシミュレーションには長い計算時間を要する。そこで、計算機科学分野において発展が著しい並列計算用ハードウェアである GPU を用いた、マルチコンパートメントモデル用の並列シミュレータが開発されている [22, 23]。GPU はもともと、コンピュータゲームに用いられる 3D グラフィックスなどを

リアルタイムで処理するために開発された、画像処理に特化した演算装置（プロセッサ）である。近年では、GPU が持つ 1 つの命令に対して類似した多数の計算を並列的に処理する性能が着目され、深層学習や天体物理シミュレーションなど様々な分野において応用が試みられている。このような GPU を用いた汎用的な数値計算の並列化技術を General Purpose computing on GPUs (GPGPU) と呼び、そのためのプラットフォームとして、NVIDIA 社が提供する Compute Unified Device Architecture (CUDA) [33] などが開発されている。

本節では、GPU と CUDA の基礎的事項と本研究で用いたシミュレーション環境を説明する。

### 2.3.1 GPU と CUDA の基礎的事項

GPU は 1 つ 1 つの演算装置にあたる CUDA コアを多数搭載した、並列演算用ハードウェアである。この CUDA コアの本数は GPU の演算性能を決める重要な要素である。CUDA コアをいくつかまとめたものを Streaming Multiprocessor (SM) ユニット (図 2.7) と呼ぶ。各 SM ユニットに配置されているコア数や SM ユニットの数によって並列計算の能力が異なり、例えば Tesla V100 には 80 個の SM ユニット (図 2.8) が搭載され、各 SM ユニットには 64 個の単精度 CUDA コアが搭載されているため、合計 5120 個の単精度 CUDA コアを持つ。

このような物理的な演算装置の構造に対して、CUDA プログラミングでは、スレッド・ブロック・グリッドと呼ばれるソフトウェア上の論理的な階層構造を考える (図 2.9)。スレッドは並列化された個々の計算を割り当てられる、演算の最小単位である。GPU 上で実行する 1 つの関数 (カーネル) に対して、CUDA で同時に並列化可能なスレッド数は環境に依存するが、例えば  $65535 \times 65535 \times 512$  個のような膨大な数となる。物理的な CUDA コア数よりもはるかに多いため、CUDA では内部でスケジューリングを行うことになる。このような多数のスレッドを管理するために、スレッドの集合体であるブロックとブロックの集合体であるグリッドと呼ばれる階層構造を用いる。関数の実行時に、グリッド当たりのブロック数とブロック当たりのスレッド数を指定することで、CUDA は自動的にブロック 1 つあたりに 1 つの SM を割り当て、計算を実行する。

GPU はメモリにも階層構造を持つ。メモリの階層はデバイスによって異なるが、おおむね図 2.10 のような階層と以下のような性質を持つ。



図 2.7: Tesla V100 の SM ユニット ( [34] から引用)。64 個の単精度 CUDA コア、32 個の倍精度 CUDA コアなどから構成されている。



図 2.8: Tesla V100 チップのブロック図 ([34] から引用)。80 個の SM ユニット、6,144 KB の L2 キャッシュなどを搭載している。

- プログラマブルメモリ (明示的に制御可能)
  - グローバル (デバイス) メモリ：GPU において最も大きく、高遅延のメモリ空間。デバイス上のすべての SM からアクセス可能である。
  - テクスチャメモリ：デバイスメモリの一部であり、読み込み専用のメモリ空間。SM ごとにキャッシュされるため、スレッド間で連続したデータにアクセスする場合に性能を発揮する。
  - コンスタントメモリ：デバイスメモリの一部であるが、読み込み専用且つ高速である。複数のスレッドが共有する定数を保存することで性能を発揮する。
  - ローカルメモリ：レジスタに収まりきれない変数の退避先となるメモリ空間。
  - シェアードメモリ：SM ごとに割り当てられた高速なメモリ。同一ブロック内のスレッド間で共有可能だが容量が小さい。
  - レジスタ：CUDA コアに近く、GPU において最も高速なメモリ。スレッド間で



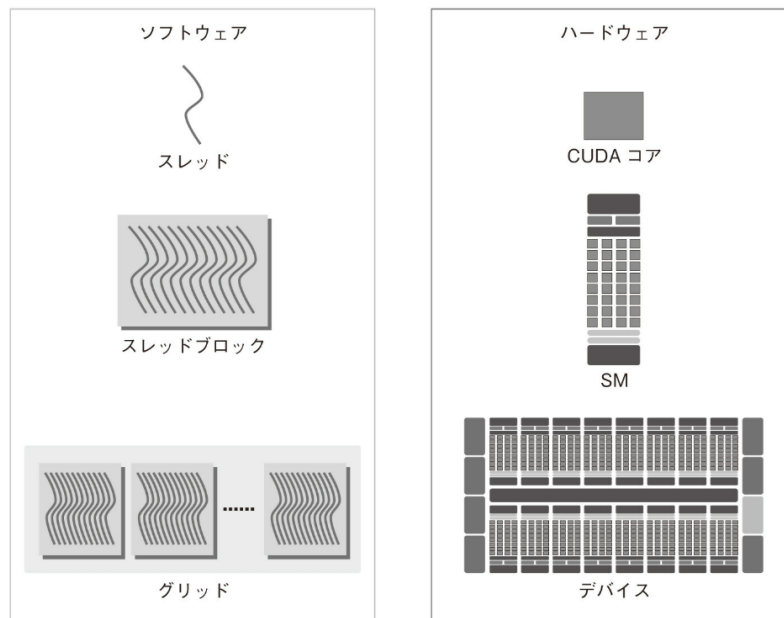


図 2.9: CUDA で取り扱う論理的な階層構造 (左) と GPU 上の物理的な階層構造の模式図 ([35] から引用)。スレッドは計算の最小単位である CUDA コアに対応し、スレッドの集合体であるブロックには 1 つの SM が割り当てられる。グリッドはブロックの集合体である。CUDA では、多数のスレッドを管理するためにこのようなグリッドとブロックの論理的な階層構造を用いる。

の共有ができない。

- ノンプログラマブルメモリ (明示的に制御できず、自動で確保される)
  - L1 キャッシュ：ローカルメモリとグローバルメモリのキャッシュとして SM ごとに存在する。高速であるが容量が小さい。
  - L2 キャッシュ：ローカルメモリとグローバルメモリのキャッシュとしてデバイス上に存在しすべての SM で共有可能。

グローバルメモリは、GPU 上に配置された最も大きく使いやすいメモリであるが、高遅延であることから、高速な演算を実行するためには、如何にグローバルメモリに頼らずカーネルを実行するかが鍵となる。グローバルメモリへのアクセスを多数行うような計算では、メモリアクセスの待機時間が計算時間のボトルネックとなることが知られているため、このような操作を避けるような工夫が必要となる。即ち、CUDA のコア数と同様に、メモリの性能と適切な並列演算の実装が GPGPU における性能を決める重要な要素となる。



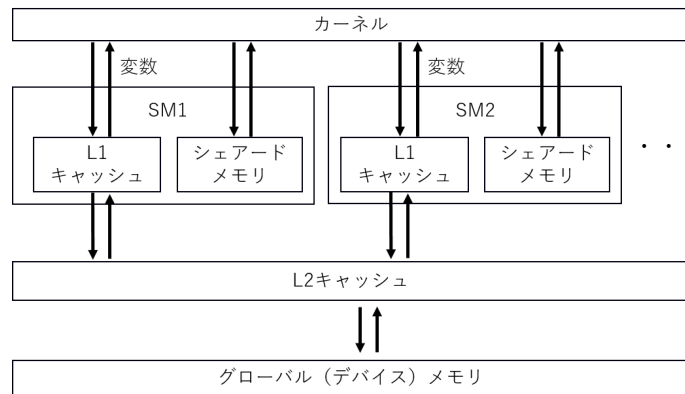


図 2.10: GPU メモリの階層構造

表 2.1: 本研究で用いたシミュレーション環境

CPU	AMD EPYC 7552
GPU	Tesla V100 32GB
Memory	125 GB
OS	Ubuntu 16.04
CPU Compiler	GCC 7.5
CUDA Toolkit	11.1

### 2.3.2 シミュレーション環境

GPU を用いた並列シミュレーションの実装には、NVIDIA が提供する GPGPU プログラミング用のプラットフォームである CUDA [33] を使用した。陰解法を用いた際に生じる連立方程式の求解には CG 法を使用し、またその実装には NVIDIA が提供する線形代数計算用ライブラリである cuBLAS [36] を使用した。cuBLAS は、GPU 上での行列やベクトルの演算を効率化し、高速な計算を可能としている。また、一部のイオンチャンネルが持つ反応方程式などの小規模な連立方程式の求解にはガウスの消去法を用いた。その他のコーディングには C 言語を用い、誤差の解析などには数値解析ソフトウェアである MATLAB を用いた。

最後に、本研究で用いた GPU を含むシミュレーション環境を表 2.1 に示す。

## 第3章

# 陽解法によるマルチコンパートメントモデルシミュレーションの高速化

### 3.1 はじめに

計算神経科学分野において、ニューロンの空間形状を考慮しないシングルコンパートメントモデルを用いたスパイクネットワークモデルは年々大規模化しており [8–12]、2020年にはヒトスケール小脳モデル [13]、2021年にはついにヒトスケール全脳シミュレーション [14] が達成された。これにより、ヒト全脳スケールでのネットワークダイナミクスの可視化や疾患の再現による病態予測などへの発展が期待される。一方で、シングルコンパートメントモデルによる計算機シミュレーションでは神経情報処理におけるニューロンの空間形状の影響を考察できない。近年の研究では、ニューロンはその空間的な形状を用いて、単体でも高度な情報処理能力を持つことが示唆されている [17, 28, 37]。そこで、次の計算神経科学分野の試みとして、ネットワークダイナミクスや脳機能に対する個々のニューロンの微細な形状が持つ機能的役割を検討するため、ニューロンの空間形状を考慮したマルチコンパートメントモデルによる大規模シミュレーションの実行が試みられている [18, 19, 21, 38–41]。微細な細胞形状のモデル化は難しい技術であるが、長年にわたるスパコンの性能向上やそれをサポートするソフトウェアツール [42, 43] の開発によって実現されつつあり、着実にマルチコンパートメントモデルを用いたヒトスケールシミュレーションに近づきつつある。

一方で、マルチコンパートメントモデルは、ケーブル方程式と呼ばれる拡散方程式によって記述されることや、各コンパートメントはそれぞれシナプス結合やイオンチャネルを

持つため数値計算が煩雑且つ計算コストが高い。そこで、一般的には専用のシミュレータ (NEURON [24] や GENESIS [30] など) を用いて計算される。しかしながら、NEURON や GENESIS シミュレータは古く、大規模なネットワークシミュレーションには向かないため、スパコンやより汎用的な GPU に代表される並列計算用ハードウェアをサポートした新たなシミュレータ (CoreNEURON [22] や Arbor [23]) が開発されている。

CoreNEURON や Arbor はマルチコンパートメントモデルの拡散方程式の計算に Crank-Nicolson (CN) 法や陰的オイラー法などのスタンダードな陰解法を用いている。陰解法は拡散方程式を安定的に計算することが可能であるが、シミュレーションの各時間ステップごとに連立方程式の求解が必要となる。一般的に、GPU において大規模な連立方程式を解く際には、大域的なメモリアクセスが必要となるため、高速なキャッシュメモリから低速なグローバルメモリへとデータが溢れてしまうことが問題として挙げられる。即ち、並列計算による高速化において、多数のメモリアクセスを要する連立方程式の求解は計算時間のボトルネックとなる可能性がある。このボトルネックを解消するための方法の 1 つは、連立方程式の求解を必要としない、陽解法を用いることである。一方で、陽解法はスパイク生成時の膜電位計算が不安定になることが知られており、これを回避するためには極めて小さい時間刻み幅を要するため、マルチコンパートメントモデルの数値計算法は陰解法を使わざるを得ない状況となっている。

そこで本研究では、連立方程式の求解を必要としないが、安定的な数値計算を実行する手法として、拡散方程式用の陽解法の一つである、Runge-Kutta-Chebyshev (RKC) 法 [44–48] に着目した。RKC 法は適応的な時間刻み幅と可変ステージ数を持ち、分子の拡散と反応計算などを解くために使用される数値計算法である。これを膜電位の拡散とアクティブなイオンチャンネルを持つマルチコンパートメントモデルに適用し、その GPU 上での性能を評価する。特に、特徴的な空間形状を持つ小脳ニューロンのマルチコンパートメントモデルのシミュレーションに用いて、計算精度、時間、メモリ使用状況などを CN 法と比較した。更に、小脳ネットワークモデルを構築し、RKC 法を適用、性能評価を行った。

## 3.2 方法

### 3.2.1 小脳ニューロンのマルチコンパートメントモデル

本研究では、小脳プルキンエ細胞 [31,32]、顆粒細胞 [49]、ゴルジ細胞 [50]、下オリーブ核ニューロン [51] の4種類のマルチコンパートメントモデルを使用した (図 3.1)。上記のモデルを先行研究のパラメータを用いて、CN 法および RKC 法を使用して GPU 上でシミュレーションできるように再実装した。

プルキンエ細胞モデルは細胞体および3種類の樹状突起 (spiny、smooth、main) からなる合計 1600 個のコンパートメントによって構成される複雑な形状を持つ (図 3.1a)。各コンパートメントは種類に応じたいくつかのイオンチャンネルを持つ。チャンネルのダイナミクスは Hodgkin-Huxley (HH) 型方程式で記述される。また、コンパートメントおよびイオンチャンネルのパラメータは De Schutter らのモデルで最も標準的な PM9 型モデルを採用した [31]。

顆粒細胞モデルは樹状突起、細胞体、軸索小丘、軸索初節、上行性軸索、平行線維からなる合計 578 個のコンパートメントによって構成されている (図 3.1b)。イオンチャンネルはプルキンエ細胞モデルと同様に、HH 型方程式で記述されているが、例外的にスパイク生成の主成分となる  $\text{Na}^+$  のチャンネルのみ電位依存性の 12 状態の反応方程式によって記述される [49] (詳細は 3.2.5 節)。

ゴルジ細胞は先端樹状突起、基底樹状突起、細胞体、軸索の 131 コンパートメントから構成される (図 3.1c)。イオンチャンネルは細胞体に集中しており、樹状突起及び軸索はパッシブなリークチャンネルのみが存在している。

下オリーブ核ニューロンモデルは樹状突起、細胞体、軸索のコンパートメントの3コンパートメントのみから構成される (図 3.1d)。

各ニューロンモデルのイオンチャンネル式とパラメータは付録 A.1 に記載する。

### 3.2.2 ネットワークモデル

図 3.2 に本研究で用いた小脳皮質のネットワークモデルを示す。本ネットワークモデルは小脳皮質における最低限の構成要素にあたる顆粒細胞、ゴルジ細胞、プルキンエ細胞から構成され、各ニューロンの細胞数は計算時間の観点から可能な限り最大数に設定した (顆粒細胞、ゴルジ細胞、プルキンエ細胞がそれぞれ 2048 個、512 個、512 個)。本研究では、学習

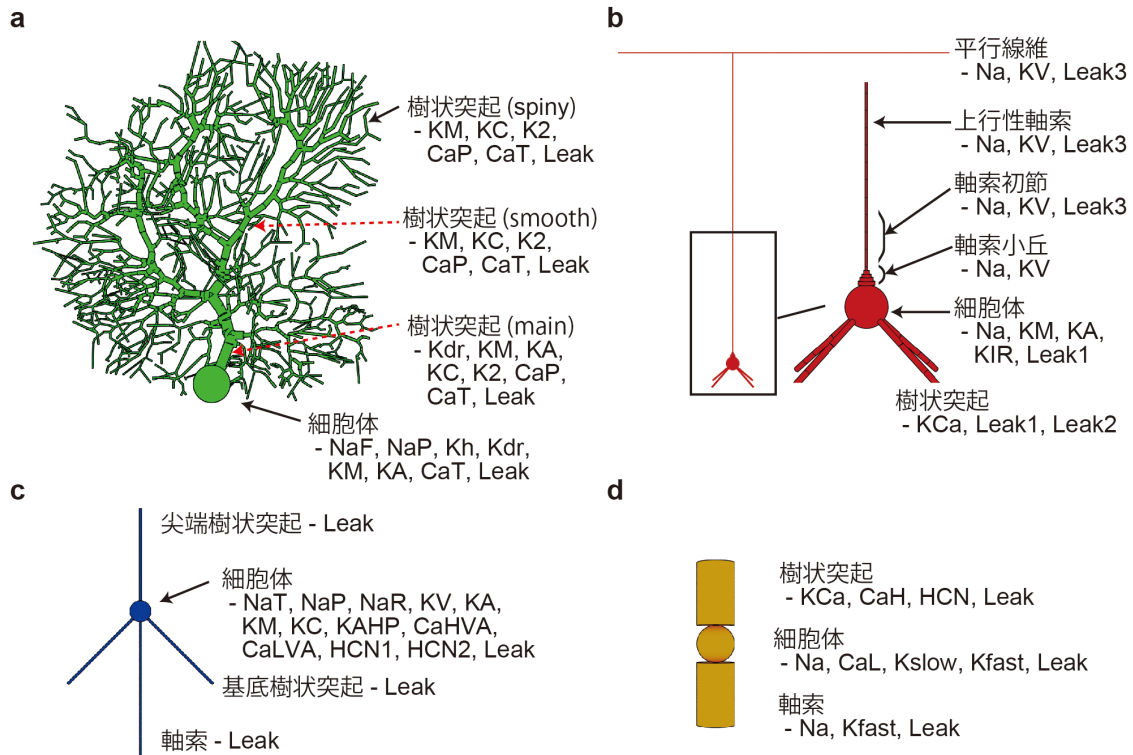


図 3.1: 小脳ニューロンのマルチコンパートメントモデル。a: 小脳プルキンエ細胞のマルチコンパートメントモデル [31,32]。spiny (1485 個)、smooth (105 個)、main (9 個) と呼ばれる樹状突起と細胞体 (1 個) の合計 1600 コンパートメントからなる。b: 小脳顆粒細胞モデル [49]。4 本の樹状突起 (それぞれ 4 個のコンパートメント)、細胞体 (1 個)、軸索小丘 (5 個)、軸索初節 (3 個)、上行性軸索 (53 個)、平行線維 (500 個) の合計 578 コンパートメントからなる。拡大図: 細胞体周辺のコンパートメント。視認性向上のため、軸索と平行線維の半径を実際のパラメータ値より大きく表示している。c: ゴルジ細胞のマルチコンパートメントモデル [50]。3 本の樹状突起 (それぞれ 10 個のコンパートメント)、細胞体 (1 個)、軸索 (100 個) の合計 131 コンパートメントからなる。d: 下オリーブ核ニューロンのマルチコンパートメントモデル [51]。樹状突起、細胞体、軸索がそれぞれ 1 個、合計 3 個のコンパートメントからなる。ハイフン以下は各種類のコンパートメントが持つイオンチャネルを表す。

のシミュレーションを行わないため、運動学習において教師信号の役割を持つ下オリーブ核ニューロンモデルは含めなかった。

このネットワークモデルにおいて、顆粒細胞は4本の樹状突起にそれぞれ苔状線維からの興奮性シナプス結合を受け、近隣のゴルジ細胞から抑制性のシナプス結合を受ける。また、ゴルジ細胞とプルキンエ細胞は顆粒細胞の平行線維から興奮性のシナプス結合を受ける。

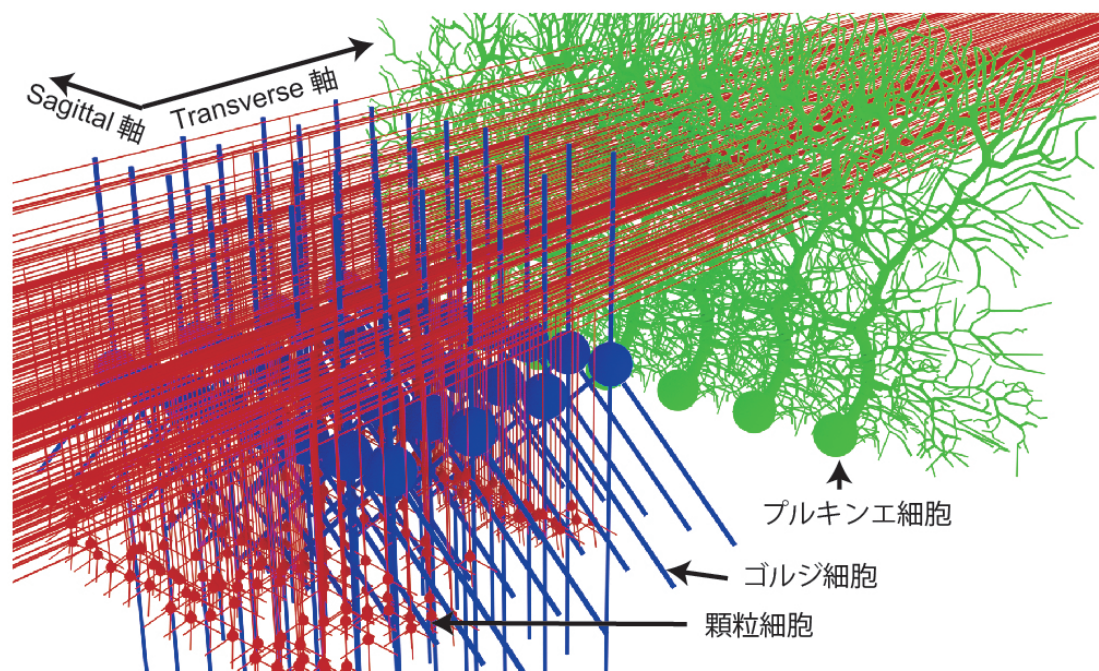


図 3.2: 顆粒細胞、ゴルジ細胞、プルキンエ細胞からなる小脳ネットワークモデル。各ニューロンは3次元空間上に配置されている。顆粒細胞は苔状線維を模した興奮性のシナプス結合と近隣のゴルジ細胞からの抑制性のシナプス結合を受ける。ゴルジ細胞とプルキンエ細胞は顆粒細胞の平行線維から興奮性のシナプス結合を受ける。ネットワークモデルのサイズは、2048 個の顆粒細胞 (1183744 コンパートメント)、512 個のゴルジ細胞 (68513 コンパートメント)、512 個のプルキンエ細胞 (819200 コンパートメント) である。Sagittal 軸と Transverse 軸はそれぞれ、冠状面と矢状面に垂直な方向の軸を表す。

顆粒細胞は興奮性シナプスの受容体として  $\alpha$ -amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) 受容体と N-methyl-D-aspartate (NMDA) 受容体、抑制

表 3.1: 各種シナプスにおけるパラメータと減衰関数

ニューロン	シナプス	減衰関数	$E_c$ [mV]	$\bar{g}_c$ [pS]
顆粒細胞	AMPA	$\alpha_{\text{AMPA}}(t) = e^{-t/1.2}$	0.0	0.18
	NMDA	$\alpha_{\text{NMDA}}(t) = e^{-t/52.0}$	0.0	0.025
	GABA <sub>A</sub>	$\alpha_{\text{GABA}_A}(t) = 0.43e^{-t/7.0} + 0.57e^{-t/59.0}$	-82.0	0.028
ゴルジ細胞	AMPA	$\alpha_{\text{AMPA}}(t) = e^{-t/1.5}$	0.0	45.5
	NMDA	$\alpha_{\text{NMDA}}(t) = 0.33e^{-t/31.0} + 0.67e^{-t/170.0}$	0.0	30.0
プルキンエ細胞	AMPA	$\alpha_{\text{AMPA}}(t) = e^{-t/6.0}$	0.0	0.7

性シナプスの受容体として  $\gamma$ -aminobutyric acid type A (GABA<sub>A</sub>) 受容体を持つ。同様に、ゴルジ細胞は AMPA 受容体と NMDA 受容体を、プルキンエ細胞は AMPA 受容体のみを持つ。これらの受容体をもたらすシナプス性電流は、シナプス後細胞  $a$  側のコンパートメント番号を  $i$ 、シナプス前細胞  $b$  側のコンパートメント番号を  $j$  とすると、以下のように定式化できる。

$$I_{\text{syn},a,i;b}(t) = \bar{g}_c w_{a,i;b} (v_{a,i}(t) - E_c) \int_{-\infty}^t \alpha_c(t-s) \delta_b(s) ds. \quad (3.1)$$

ただし、 $w_{a,i;b}$  はニューロン  $b$  からニューロン  $a$  の  $i$  番目のコンパートメントへのシナプス結合荷重であり、 $v_{a,i}(t)$  はニューロン  $a$  の  $i$  番目コンパートメントの膜電位、 $\delta_b(t)$  はニューロン  $b$  の発火活動を表す。また、 $\bar{g}_c$ 、 $E_c$ 、 $\alpha_c(t-s)$  はそれぞれ、シナプスの最大コンダクタンス、反転電位、減衰関数を表す（ただし  $c \in \{\text{AMPA}, \text{NMDA}, \text{GABA}_A\}$ ）。各ニューロンの各種シナプスにおけるパラメータおよび減衰関数を表 3.1 に示す。また、シミュレーションを実行する上で、シナプス結合荷重の値は表 3.2 を基にシナプス後細胞が受ける結合数によって正規化している。

ゴルジ細胞のダイナミクスの特徴として、近隣のゴルジ細胞は樹状突起同士がギャップ結合と呼ばれる電氣的な結合によって互いに接続し、同期的な活動を示すことが知られている [52–54]。Solinas らによるゴルジ細胞モデル [50] ではギャップ結合がモデル化されていなかったため、本研究ではゴルジ細胞間のギャップ結合として、最も簡易的なモデル化として

表 3.2: シナプス結合荷重の値

シナプス前細胞	シナプス後細胞	値
苔状線維	顆粒細胞	40000.0
顆粒細胞	ゴルジ細胞	16.0
ゴルジ細胞	顆粒細胞	640000.0
顆粒細胞	プルキンエ細胞	1000000.0

式 (3.2) を実装した。

$$I_{gj,a,i;b,j} = \bar{g}_{gj}(v_{b,j} - v_{a,i}), \quad (3.2)$$

ただし、 $i$  と  $j$  はそれぞれギャップ結合で接続しているゴルジ細胞  $a$  と  $b$  の樹状突起のコンパートメント番号を、 $\bar{g}_{gj}$  はギャップ結合の最大コンダクタンス (2.5 nS [52, 53]) を表す。

ゴルジ細胞と同様に、下オリーブ核ニューロンの樹状突起もギャップ結合によって接続し、同期的な活動を示すことが知られている [51]。De Gruijl らによる下オリーブ核ニューロンモデルではギャップ結合がモデル化されていたため、それに従い本研究では式 (3.3) を実装した [51]。

$$I_{io} = \bar{g}_{io}(0.8e^{-0.01v_{diff}^2} + 0.2)v_{diff} \quad (3.3)$$

ただし、 $\bar{g}_{io}$  は IO モデル間のギャップ結合のコンダクタンスであり、 $v_{diff}$  は結合しているコンパートメント間の電位差を表す。

本研究では特定領域内のプルキンエ細胞同士の同期的なスパイク活動 [55, 56] を再現するために、式 (3.2) を用いて隣接するプルキンエ細胞の樹状突起間にもギャップ結合を実装した。具体的には、sagittal 軸方向に 50  $\mu\text{m}$  間隔でプルキンエ細胞を配置し、それらの樹状突起がコンパートメントの半径よりも小さい場合に接続を想定し結合した。このプルキンエ細胞間のギャップ結合モデルについては 3.4.3 節にて考察を行う。また、De Schutter らはこのプルキンエ細胞モデルは強い活動を示し過ぎる傾向がある [31, 32] と記述したことから、本研究のネットワークシミュレーションでは発火活動を抑制するために、CaP チャネル (持続性の  $\text{Ca}^{2+}$  チャネルであり興奮性の電流が流れる) のコンダクタンスを 4.5 nS から 3.0 nS に、K2 チャネル (低閾値の  $\text{Ca}^{2+}$  依存性  $\text{K}^+$  チャネルであり抑制性の電流が流れる) のコンダクタンスを 0.39 nS から 0.9 nS に変更した。



### 3.2.3 拡散方程式用の陽解法

2.2 節で述べたように、一般的に陽解法は拡散方程式を含む偏微分方程式の求解には数値計算の不安定性から使用が避けられる。最も一般的な陽解法である、陽的オイラー法や古典的 Runge-Kutta 法をマルチコンパートメントモデルのシミュレーションに適用した場合、スパイク活動生成時の膜電位やイオンチャネルの計算が発散しやすい。このような計算の発散を回避するためには、時間刻み幅を極めて小さく設定する必要があるが、これは丸め誤差や計算時間の増加を引き起こす。しかしながら、陽解法は連立方程式の求解を必要しないため、GPU などのメモリバンド幅が小さく大規模なデータの転送に時間がかかるハードウェアを用いる上で非常に大きなメリットがある。そのため数値解析分野などにおいて、偏微分方程式用の陽解法の研究が進められてきた。そこで、反応項を含む拡散方程式用の陽的解法として、Runge-Kutta-Chebyshev (RKC) 法 [44–48] が開発された。RKC 法は “mildly stiff (適度に硬い)” モデルのシミュレーション向けの数値計算法として設計されている。ここで、“stiffness (硬さ)” とは、システム内で短時間の高速なダイナミクスが長時間の低速なダイナミクスにどの程度強く影響するかを意味する。RKC 法を使用する利点として、RKC 法は適応的な時間刻み幅とステージ数を持つことで、スパイク活動生成時に時間刻み幅を一時的に小さくしステージ数を大きくすることで局所誤差を抑え込み、安定的な数値計算が行えることが挙げられる。また、マルチコンパートメントモデルのシミュレーションにおいて一般的な解法である CN 法は時間に関する 2 次精度を持つが、RKC 法は時間に関する精度を 2 次に拡張することが可能であり、CN 法の代替手法として有用である。RKC 法のアルゴリズムと安定性については付録 A.2 に記載する。

また、RKC 法は誤差の許容値 (tolerance) と呼ばれるパラメータを持ち、誤差の値をコントロールすることができる。一般的に、tolerance を厳しく設定することで、誤差の値を小さくすることが可能であるが、一方で計算にかかる時間は増加する [47]。本研究では、この tolerance による誤差の値と計算時間の関係性を評価するために、RKC 法と CN 法による膜電位計算における平均二乗誤差と計算時間を調査した (シミュレーションの条件は 3.2.7 節)。

### 3.2.4 数値計算法の実装

本研究では、RKC 法の GPU 上での性能評価を行うため、マルチコンパートメントモデルの一般的な数値計算法である CN 法と比較した。比較のために、C 言語と CUDA を用いて CN 法と RKC 法を独自に実装した。実装にあたり使用した CUDA 用のライブラリや特殊な並列化方法は付録 A.3 に記載する。

### 3.2.5 硬さを回避したマルチコンパートメントモデル

RKC 法などの動的な時間刻みを持つ陽解法を硬いモデル (3.2.4 節を参照) に適用する場合、計算を安定化させるため細かい時間刻みを必要とする。マルチコンパートメントモデルにおける硬さとは、短時間で急激な変化をする反応項によって膜電位全体のダイナミクスが変化するようなモデルを指す。ニューロンモデルはスパイク活動を生成するために比較的短時間で変化する  $\text{Na}^+$  チャンネルを必要とすることから適度な硬さを持つ。通常、イオンチャンネルは開口と閉口の 2 状態を持つ HH 型方程式で記述されているが、Dover らの顆粒細胞モデルは  $\text{Na}^+$  チャンネルを 12 状態の反応方程式で記述している (図 3.3) ため、HH 型方程式よりも短い時定数を持つ。そのため、Dover らの顆粒細胞モデルは硬いモデルであることが予想される。そこで、この  $\text{Na}^+$  チャンネルを通常の HH 型方程式で記述することで硬さ問

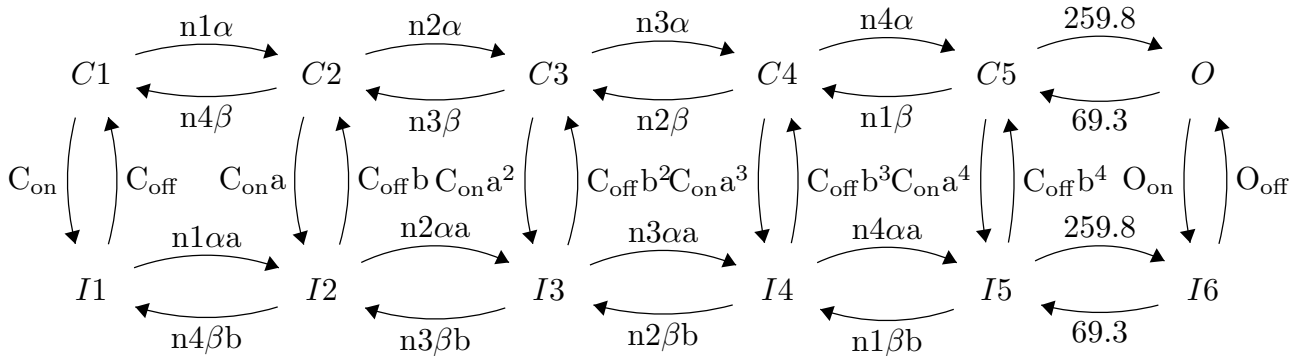


図 3.3: Dover らの顆粒細胞マルチコンパートメントモデルが持つ 12 状態の反応方程式で記述された  $\text{Na}^+$  チャンネル [49]。反応方程式を解くことによってゲート変数 ( $O$ ) が得られる。このイオンチャンネルモデルは Fibroblast growth factor homologous factor (FHF) たんぱく質による顆粒細胞モデルの性質変化を示すために使用されている。

表 3.3: Gabbiani らの顆粒細胞モデルの HH 型  $\text{Na}^+$  チャネル方程式 [57]。最大コンダクタンスは  $\bar{g}_{\text{Na}} = 70[\text{mS}/\text{cm}^2]$ 、反転電位は  $E_{\text{Na}} = 55 [\text{mV}]$  である。

状態変数	変化率 ( $\alpha(v)$ )	変化率 ( $\beta(v)$ )	時定数の下限値 [ms]
開口状態 $m$	$1.5e^{0.081(v+39.0)}$	$1.5e^{-0.066(v+39.0)}$	0.05
閉口状態 $h$	$0.1e^{-0.089(v+50.0)}$	$0.12e^{0.089(v+50.0)}$	0.225

題を回避することを検討した。具体的には、Gabbiani らの顆粒細胞マルチコンパートメントモデル [57] の HH 型  $\text{Na}^+$  チャネル方程式を適用した。表 3.3 にチャネルのパラメータを示す。ただし、変化率  $\alpha(v)$  と  $\beta(v)$  はそれぞれ、式 (2.4) において  $x_0(v) = \frac{\alpha(v)}{\alpha(v)+\beta(v)}$ 、 $\tau_x(v) = \frac{1}{\alpha(v)+\beta(v)}$  とした際のゲート変数の状態遷移率を表す。また、Gabbiani らの顆粒細胞モデルは、樹状突起及び細胞体のみからなる 13 コンパートメントのモデルであり、Dover らによるモデルと異なるダイナミクスを持つため、入出力関係とスパイクの伝搬速度が保たれるように各コンパートメント上のチャネル密度を再設定した。この HH 型の  $\text{Na}^+$  チャネルは時定数の下限値を持つため、モデルの硬さを制限できる。

本研究では、通常、顆粒細胞モデルは Dover らのオリジナルモデルを指し、硬さを回避した顆粒細胞のマルチコンパートメントモデルを使用する場合はその都度「改良したマルチコンパートメントモデル」などの記述を行う。

### 3.2.6 メモリ使用状況の測定方法

シミュレーションにおける、GPU 上のメモリの使用状況を解析するために、本研究では NVIDIA プロファイラ [58] の “dram\_read\_transactions” と “dram\_write\_transactions” メトリクスを用いてメモリ情報を測定した。これは L2 キャッシュとデバイスメモリ間のデータ転送の測定に相当する (図 2.10 参照)。メモリ情報のプロファイリングには非常に時間がかかるため、本研究ではプルキンエ細胞モデルに 2.0 nA の定電流を与えた 10 ms 間のシミュレーションを解析し、デバイスのグローバルメモリの読み込みと書き込みのデータ量を測定した。本研究で使用した GPU である Tesla V100 の使用データ [59] に従って、測定データを MByte に変換して評価した。

### 3.2.7 シミュレーション条件

各ニューロンモデルのシミュレーションにおいて、入力期間中はニューロンの細胞体へ定電流を入力電流として与え、それ以外の期間は入力電流を与えていない。また、各ニューロンの膜電位は細胞体の膜電位を測定した。ネットワークモデルのシミュレーションにおいては、入力期間中は 30 Hz、入力期間外は 5 Hz で若状線維を活動させ、顆粒細胞へシナプス入力を与えた。シミュレーションの計算時間は顆粒細胞を除き、500 ms 間のフリーランを行った後の 1000 ms 間のシミュレーションにかかる時間を計測した。顆粒細胞においては、実行時間の都合上、フリーランを行わず、100 ms 間のシミュレーションにかかる時間を計測した。計算精度を比較するためのシミュレーションを除き、各ニューロンの膜電位は静止電位から最大 5 mV ほどランダムにずれた初期値を与えシミュレーションを行った。反応方程式で記述されたイオンチャネルはガウスの消去法を用いて計算した。

また、tolerance による誤差の値と計算時間の関係性を調査するために、RKC 法と CN 法による膜電位計算における平均二乗誤差と計算時間を計測した（シミュレーションの条件は 3.2.7 節）。CN 法は固定時間刻み幅（本研究では  $25\mu\text{s}$ 、2.2 節参照）であるのに対して、RKC 方は時間刻み幅が変動するため、tolerance の評価を行う際は、RKC 法の場合は適応時間ステップごとに膜電位の数値データを生成し CN 法のデータを線形補間しながら評価した。平均二乗誤差は入力電流を与えている期間中のみ計算し、計算時間はフリーランを除くシミュレーション期間全体で測定した。

## 3.3 結果

GPU 上に実装した小脳ニューロンのマルチコンパートメントモデルのシミュレーションを実行し、RKC 法の性能調査を行った。はじめに、脳内でも特に複雑な形状を持つ小脳のプルキンエ細胞のマルチコンパートメントモデルのシミュレーションに対して RKC 法を適用し、3.3.1 節ではその計算精度と時間、3.3.2 節ではメモリパフォーマンスを CN 法と比較・調査した。次に、3.3.3 節では、他のニューロンモデル（顆粒細胞、ゴルジ細胞、下オリーブ核ニューロン）に対する RKC 法の性能を調査した。3.3.4 節では、顆粒細胞、ゴルジ細胞、プルキンエ細胞からなる小脳ネットワークモデルのシミュレーションに RKC 法を適用し、大規模なシミュレーションにおける RKC 法の性能を評価した。最後に、3.3.5 節では、数値

計算法ではなくモデルの改良案として、既存の顆粒細胞モデルのイオンチャネルを改良し、RKC 法に適切したマルチコンパートメントモデルを作成、そのモデルに対する RKC 法の性能評価を行った。

### 3.3.1 プルキンエ細胞モデルシミュレーションにおける RKC 法の性能評価

はじめに、RKC 法によるシミュレーションがプルキンエ細胞モデルの挙動を再現しているか確認するため、モデルの原著論文 [31] に近い条件（約 1 秒間 0.5 nA、1.0 nA、2.0 nA、3.0 nA の細胞体への電流入力）を与えたシミュレーションを行った（図 3.4）。原著論文の結

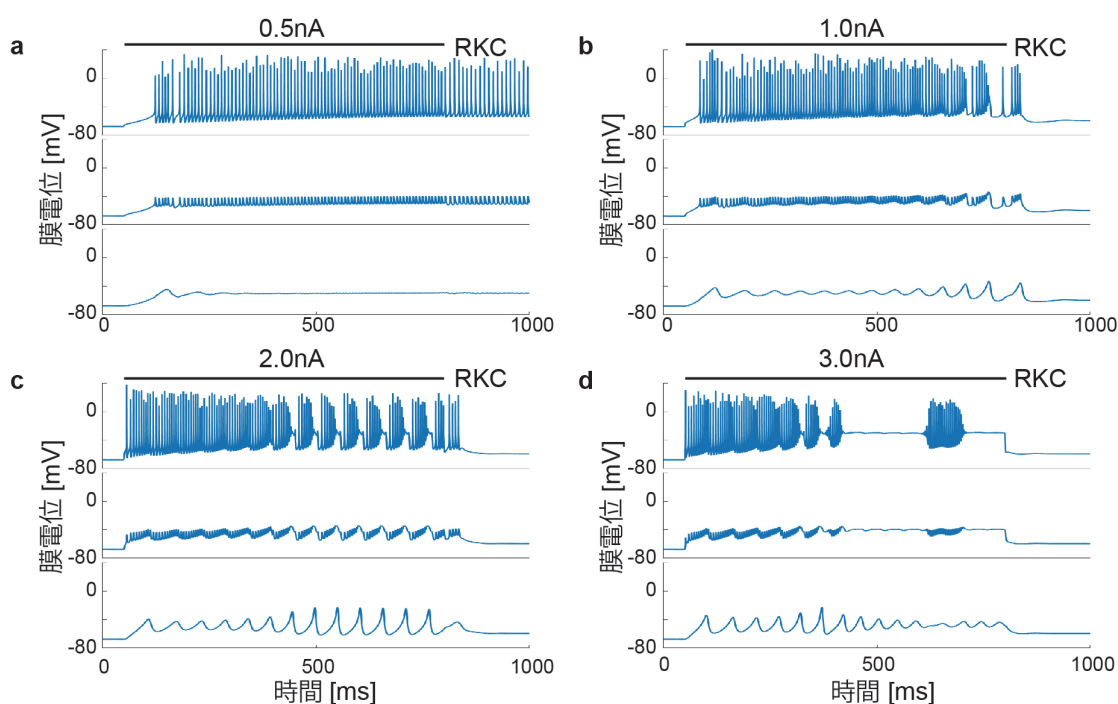


図 3.4: GPU 上に実装したプルキンエ細胞モデルの RKC 法によるシミュレーション結果。a–d: 細胞体（上）、樹状突起（main）、樹状突起（spiny）における膜電位応答（a: 0.5 nA、b: 1.0 nA、c: 2.0 nA、d: 3.0 nA）。電流入力は細胞体に与えている。

果（図 3.5）と比較して、低強度の電流に対しては持続的な発火活動（図 3.4a, b）、高強度の電流には休止と発火の繰り返しの応答（図 3.4c, d）といった特徴的な活動を再現した。

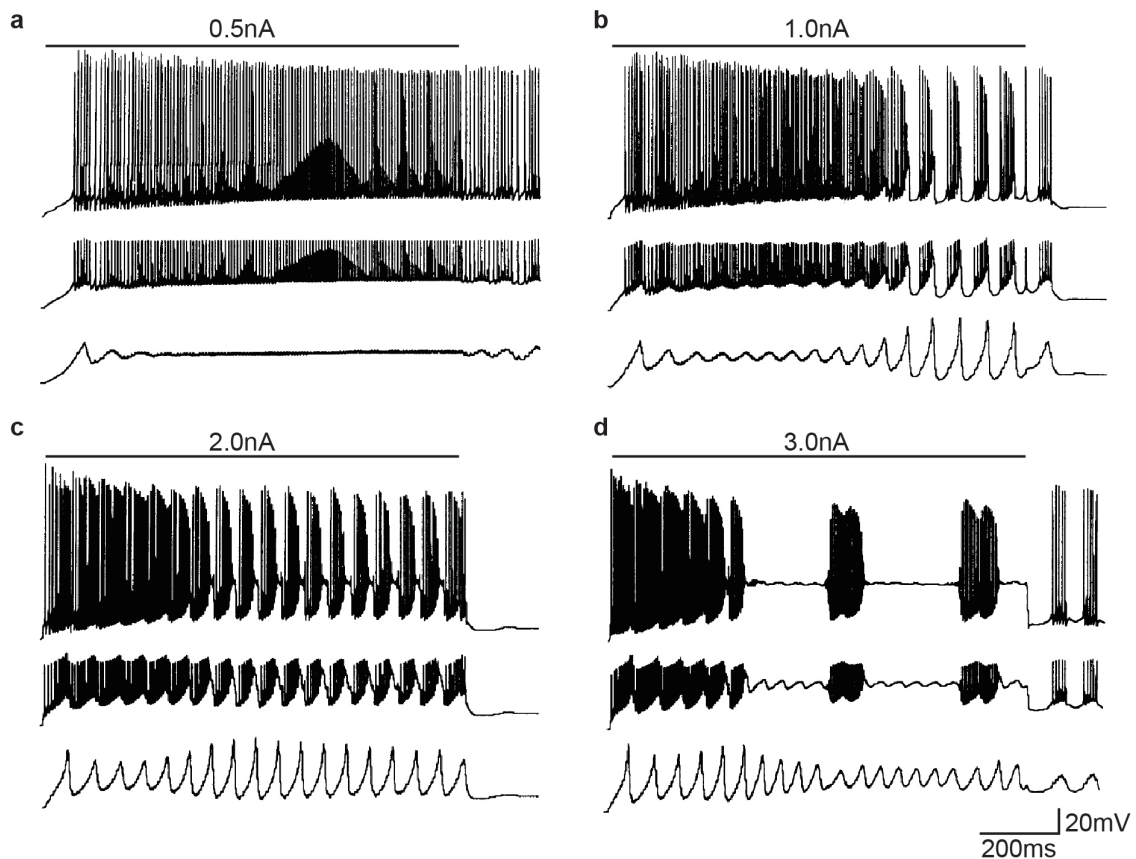


図 3.5: プルキンエ細胞モデルの原著論文におけるシミュレーション結果 ([31] から改変)。このプルキンエ細胞モデルは GENESIS シミュレータ上に実装されており、 $25\mu s$  の刻み幅を持つ Crank-Nicolson 法によって計算されている。a-d: 細胞体 (上)、樹状突起 (main)、樹状突起 (spiny) における膜電位応答 (a: 0.5 nA、b: 1.0 nA、c: 2.0 nA、d: 3.0 nA)。

次に、本研究で独自に実装した CN 法と RKC 法によるシミュレーション結果を比較するため、原著論文では検証していないより低い電流入力 (0.1 nA) に対するプルキンエ細胞モデルの膜電位応答をシミュレーションした (図 3.6)。図 3.6a と b から、本研究で実装した RKC 法と CN 法はほぼ同様の膜電位応答がシミュレーション可能であり、RKC 法の再現性が確認できた。

次に、ギャップ結合を介してプルキンエ細胞を電氣的に接続することで、プルキンエ細胞

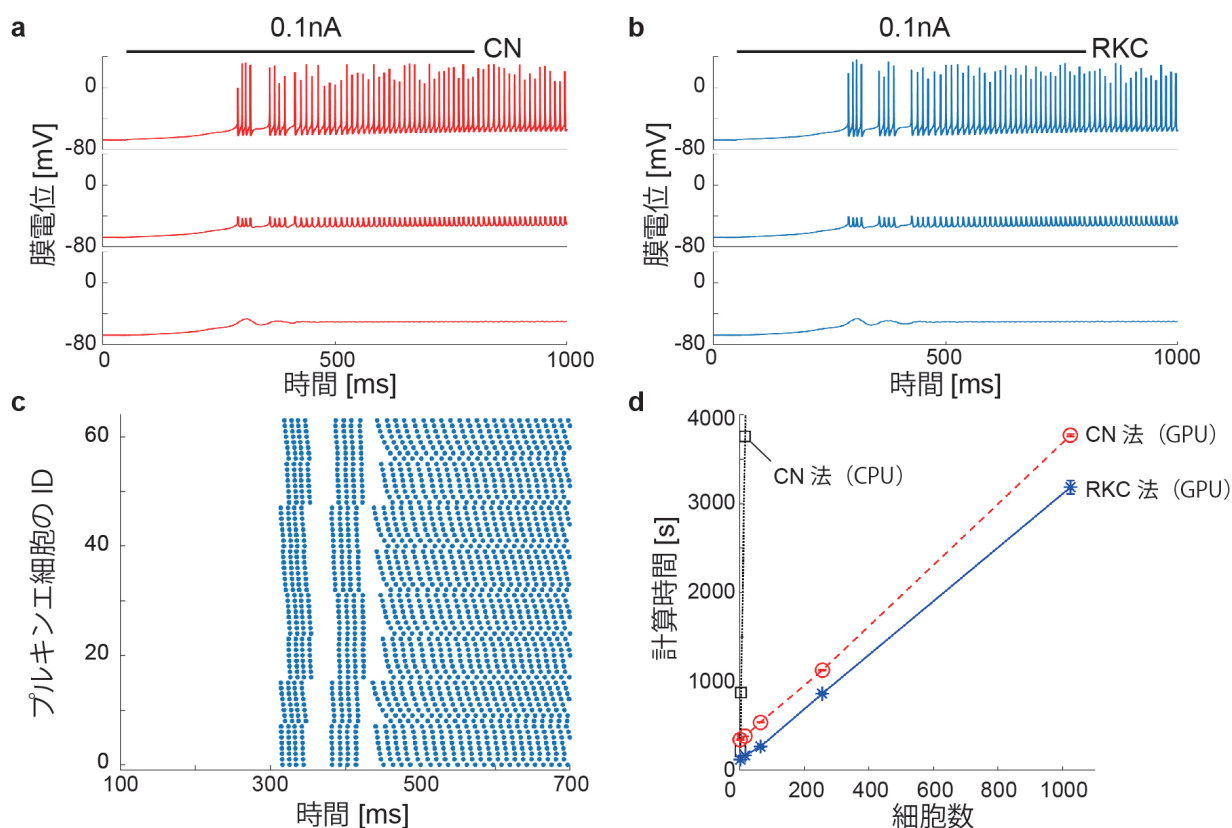


図 3.6: 0.1 nA の電流入力に対するプルキンエ細胞モデルのシミュレーション結果。**a, b:** CN 法と RKC 法による膜電位応答。**c:** RKC 法を用いた 64 個のプルキンエ細胞のシミュレーションにおけるラスタプロット (各プルキンエ細胞が発火活動を示した時刻に点を表示)。1000 ms のシミュレーションにおいて、50 ms から 800 ms 間に入力電流を与えているが、視認性向上のため 100 ms から 700 ms の結果のみを表示している。**d:** プルキンエ細胞の細胞数と数値計算手法に対する計算時間の変化。赤の点線 (丸印) と青の実線 (米印) はそれぞれ GPU 上で CN 法と RKC 法を用いた場合の計算時間を表す。黒の破線 (四角印) は CPU 上で CN 法を用いた場合の計算時間を表す (計算時間の都合上、最大 4 個のプルキンエ細胞までの結果を表示)。計算時間のデータは 5 回のシミュレーションにおける平均値であり、エラーバーは標準偏差を表す。

のネットワークモデルを構築した。64 個のプルキンエ細胞をシミュレーションした結果 (図 3.6c)、ギャップ結合によって接続されたプルキンエ細胞のグループは同期的な発火活動を示した。次に、プルキンエ細胞の数を 1 個から 1024 個まで変化させながら、RKC 法と CN 法の計算時間を比較した (図 3.6d)。GPU を用いたプルキンエ細胞のマルチコンパートメントモデルのシミュレーションにおいて、RKC 法 (青の実線) は CN 法 (赤の破線) よりも高速

であり、計算時間の観点から CN 法よりも RKC 法が優れていることがわかる。また、細胞数の増加に対して、CN 法による計算時間の増加は RKC 法より大きな傾きを示しており、スケーリング性能においても RKC 法が優れていることが示された。最後に確認として、CPU 上にもプルキンエ細胞モデルを実装し、計算時間を測定した (図 3.6d の黒の破線)。CPU を用いたシミュレーションの計算時間は GPU に対して非常に遅く、GPU による並列シミュレーションの有効性が確認された。

### 3.3.2 RKC 法のメモリ性能評価

プルキンエ細胞モデルのシミュレーションにおけるメモリ性能を調査するため、GPU 上のデバイスメモリへのデータの読み取りと書き込みのプロファイリングを実行した。メモリ性能のプロファイリングには非常に時間がかかる一方で、RKC 法は適応時間刻み幅を持つためスパイクが発生したときのシミュレーションデータを測定する必要があることから、比較的高強度 (2.0 nA) の入力電流を与えた短時間 (10 ms 間) のシミュレーションを実行した (図 3.7a)。スパイク活動を含むシミュレーションにおいて、RKC 法は CN 法よりも多くの倍精度浮動小数点演算を必要としたが (図 3.7b)、GPU のデバイスメモリのデータ転送量は

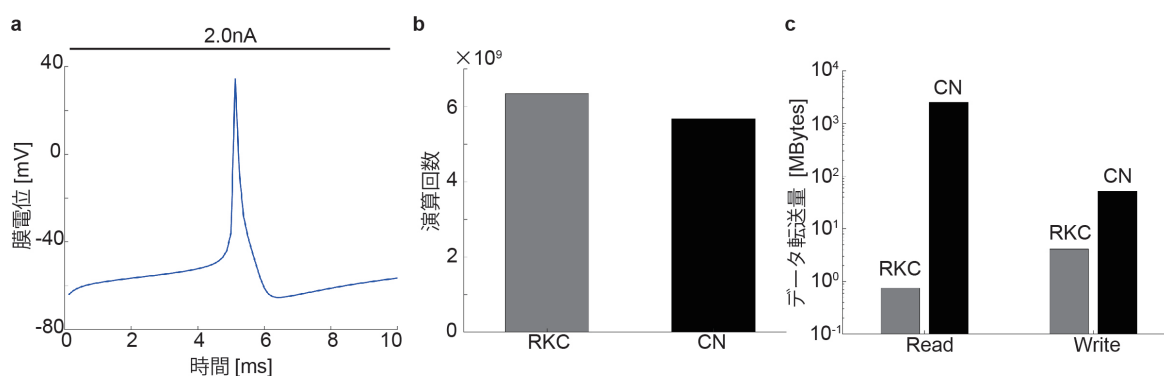


図 3.7: RKC 法および CN 法を用いたシミュレーションにおけるメモリ性能のプロファイリング。a: プロファイリング中の RKC 法によるプルキンエ細胞モデルの膜電位応答。b, c: 10 ms 間のシミュレーションにおける、RKC 法 (灰色) と CN 法 (黒) による倍精度浮動小数点の演算回数と GPU 上のデバイスメモリへの読み取りおよび書き込みデータ量。



CN 法よりも少なかった (図 3.7c)。特に、RKC 法による読み取りデータ量は CN 法に比べて 1/1000 未満、書き込みデータ量は 1/10 未満であった。この結果は、RKC 法はメモリのデータ転送時間が計算時間のボトルネックとなりうる GPU をはじめとした並列計算用ハードウェアに適した数値計算法であることを示す。

### 3.3.3 顆粒細胞、ゴルジ細胞、下オリブ核ニューロンモデルのシミュレーション

他のニューロンモデルにおいても、RKC 法がプルキンエ細胞モデルの時と同様の性能を発揮するかを評価するため、顆粒細胞、ゴルジ細胞、下オリブ核ニューロンモデルに適用した。各ニューロンモデルのシミュレーションにおいて、RKC 法と CN 法による膜電位応答の計算結果は同様の波形を示した (図 3.8a-c)。

一方で、RKC 法は内部パラメータとして、誤差の許容値 (tolerance) を持ち、誤差の値をコントロールすることができる。誤差の許容値を厳しく設定することで、より精密な膜電位の計算が可能となるが、計算時間は増加する。そこで、各ニューロンモデルのシミュレーションにおいて誤差の許容値を変化させ、「CN 法との膜電位の平均二乗誤差」と「計算時間」を調査した (図 3.8d-f)。全てのニューロンモデルにおいて、誤差の許容値の変化は膜電位の誤差値と計算時間のトレードオフの関係を示した。

次に、細胞数の増加に対する計算時間の推移を調査した (図 3.8g-i)。ゴルジ細胞および下オリブ核ニューロンモデルのシミュレーションではプルキンエ細胞モデルと同様、RKC 法は CN 法よりも高速なシミュレーションと細胞数の増加に対する良いスケール性能を示した (図 3.8h, i)。一方で、顆粒細胞モデルのシミュレーションでは、RKC 法の計算速度は CN 法よりも遅く、また細胞数の増加に対してその差は顕著に現れた (図 3.8g)。この理由を調査するため、顆粒細胞モデルのシミュレーションにおいて、RKC 法が動的に設定した時間刻み幅を計測したところ、スパイクの生成時に計算を安定させるため、最小の刻み幅 ( $2^{-12}$  ms) を設定し続けていることが判明した。これにより、膜電位計算のステップ数が増加し、莫大な計算時間を要したと考えられる。この問題に対しては、3.4.1 節で考察する。

上記の結果は、全てのニューロンモデルに対して盲目的に数値計算法を設定するのではなく、モデルの適正に応じて数値計算法を適切に切り替えることがシミュレーションの効率化につながることを示唆している。

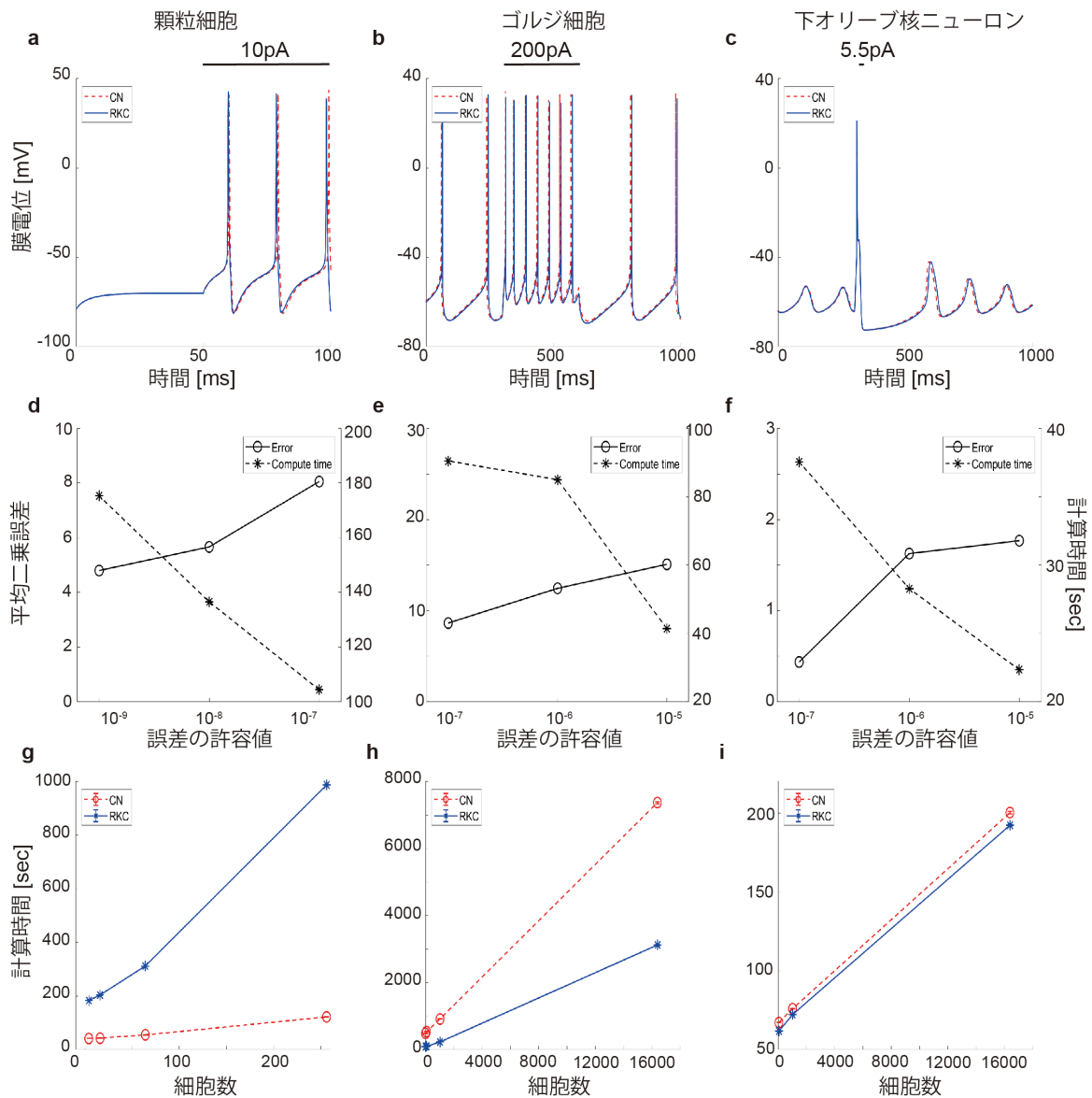


図 3.8: GPU 上に実装した他のニューロンモデルのシミュレーション結果。a-c: RKC 法 (青の実線) および CN 法 (赤の点線) による、顆粒細胞、ゴルジ細胞、下オリブ核ニューロンモデルの細胞体の膜電位応答。それぞれ細胞体に 10 pA、200 pA、5.5 pA の電流入力を与えた。図上部の実線は入力期間を表す。d-f: 各モデルのシミュレーションにおける、RKC 法の誤差の許容値に対する計算時間と平均二乗誤差。実線 (丸印、目盛りは左軸) は RKC 法と CN 法を使用したシミュレーションにおいてそれぞれ計算された膜電位間の平均二乗誤差。点線 (米印、目盛りは右軸) は RKC 法に使用した場合の計算時間。g: 顆粒細胞の細胞数に対する計算時間の推移。赤の点線 (丸印) と青の実線 (米印) はそれぞれ CN 法と RKC 法を用いた場合の計算時間を表す。各顆粒細胞の膜電位の初期値はランダムに設定し、シナプス結合を持たない。h, i: ゴルジ細胞、下オリブ核ニューロンの細胞数に対する計算時間の推移。g と同様、各細胞の膜電位の初期値はランダムに設定し、シナプス結合を持たないが、ゴルジ細胞と下オリブ核ニューロンモデルはそれぞれギャップ結合によってネットワークを構築している。a-c および g-i における RKC 法の誤差の許容値はそれぞれ  $10^{-8}$ 、 $10^{-6}$  および  $10^{-6}$  に設定している。

### 3.3.4 小脳ネットワークモデルシミュレーション

各ニューロンモデルにおいて適切な数値計算法を適用することでシミュレーションのパフォーマンスがどの程度向上するか調査するため、RKC 法と CN 法を切り替え、顆粒細胞、ゴルジ細胞、プルキンエ細胞からなる小脳ネットワークモデルのシミュレーションを実行した。

図 3.9a は、顆粒細胞モデルに CN 法を適用し、ゴルジ細胞およびプルキンエ細胞モデルに RKC 法を適用した混成手法 (Mix) によるネットワークモデルシミュレーションのラスタプロットを示している。小脳ネットワークの特徴である、顆粒細胞-ゴルジ細胞間のオシレーションとプルキンエ細胞の同期的な活動が再現されている。混成手法は、全てのニューロンモデルに CN 法のみを適用したシミュレーションよりも速い計算時間を示した (図 3.9b)。

最後に、計算時間の内訳を解析した (図 3.9c)。混成手法における計算時間の約 3/4 は 2048 個の顆粒細胞の計算によって費やされており、残りは 512 個のプルキンエ細胞と 512 個のゴルジ細胞がほぼすべての計算時間を占めていた。顆粒細胞はどちらのシミュレーションにおいても CN 法で計算したため、ゴルジ細胞とプルキンエ細胞の RKC 法による計算時間の短縮のみが、混成手法による計算時間の改善として現れている。

これらの結果は、マルチコンパートメントネットワークモデルの効率的なシミュレーションには、ニューロンモデルに適した数値計算法を適用する必要があることを示唆している。次節では、計算時間の根本的な問題になっている顆粒細胞モデルを RKC 法で解くための改良方法について提案する。

### 3.3.5 イオンチャネルモデルの変更による硬さの回避と RKC 法の性能検証

3.3.1 節と 3.3.3 節では既存の各マルチコンパートメントモデルに対する RKC 法の性能を CN 法と比較した。RKC 法は、ゴルジ細胞、プルキンエ細胞、下オリーブ核ニューロンモデルの GPU 上でのシミュレーションに計算時間の短縮やスケーリング性能の向上をもたらしたが、顆粒細胞モデルに対しては計算時間を増加させた。また、3.3.4 節の小脳ネットワークモデルのシミュレーションにおいては、RKC 法をゴルジ細胞とプルキンエ細胞に適用し、計算時間の短縮に成功したが、ネットワークモデルの計算時間のほとんどは CN 法を適用した顆粒細胞であったため、大きな改善にはいたらなかった。そこで、Dover らの顆粒細胞モデ

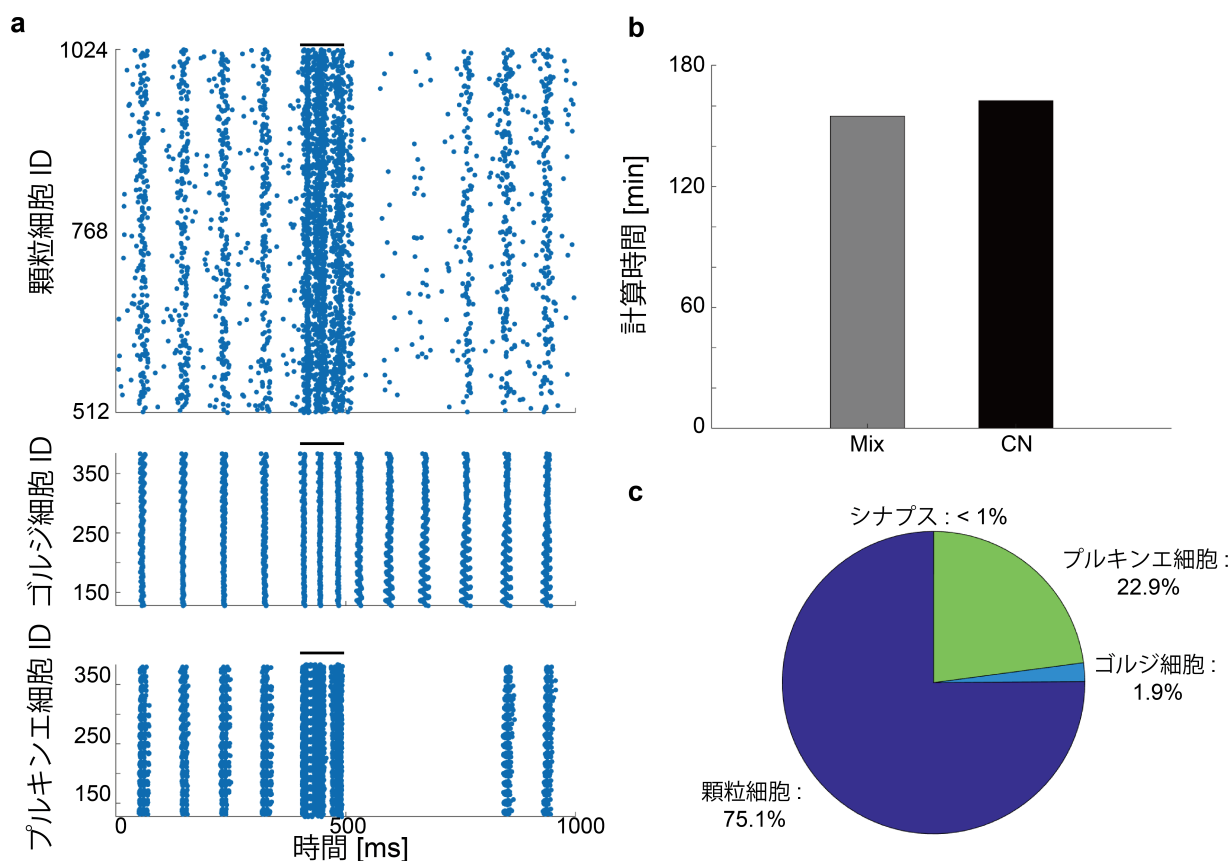


図 3.9: 小脳ネットワークモデルシミュレーション。ネットワークモデルは 2048 個の顆粒細胞（総計 1183744 コンパートメント）、512 個のゴルジ細胞（総計 68513 コンパートメント）、512 個のプルキンエ細胞（総計 819200 コンパートメント）から構成されており、総シナプス数は 780404 個である。**a**: 顆粒細胞モデルには CN 法、ゴルジ細胞とプルキンエ細胞モデルには RKC 法をそれぞれ適用した混成手法 (Mix) による各ニューロンモデルのラスタプロット。顆粒細胞への苔状線維入力が入力期間中は 30 Hz であり、期間外は 5 Hz である。図上部の実線は入力期間 (400–500 ms) を表している。**b**: 小脳ネットワークモデルシミュレーションの計算時間。灰色は混成手法、黒は全てのニューロンモデルに CN 法のみを適用したシミュレーションの計算時間を表す。**c**: 混成手法のシミュレーションにおける計算時間の内訳。

ル [49] を改良し (3.2.5 節)、そのモデルに対する RKC 法の性能を評価する。

改良した顆粒細胞マルチコンパートメントモデルに対して、RKC 法と CN 法を用いてシミュレーションを行ったところ、ほぼ一致した膜電位の波形を確認した (図 3.10a)。また、

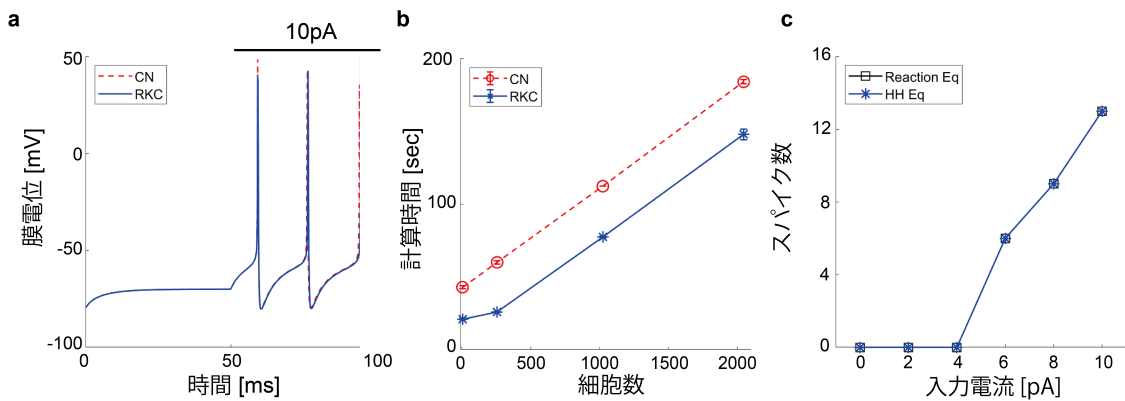


図 3.10:  $\text{Na}^+$  チャンネルを HH 型方程式に置き換えた顆粒細胞モデルのシミュレーション結果。a: RKC 法 (青の実線) および CN 法 (赤の点線) による顆粒細胞モデルの細胞体の膜電位応答。細胞体に 10 pA の電流入力を与えた。図上部の実線は入力期間を表す。b: 顆粒細胞の細胞数に対する計算時間の推移。赤の点線 (丸印) と青の実線 (米印) はそれぞれ CN 法と RKC 法を用いた場合の計算時間を表す。各顆粒細胞の膜電位の初期値はランダムに設定し、シナプス結合を持たない。c:  $\text{Na}^+$  チャンネルの違いによる入出力関係。黒の破線は  $\text{Na}^+$  チャンネルをオリジナルの 12 状態からなる反応方程式で記述した場合、青の実線は  $\text{Na}^+$  チャンネルを HH 型方程式に置き換えた場合における、入力電流強度に対する 250 ms 間のスパイク発射数をそれぞれ表す。

RKC 法は CN 法よりも速い計算時間 (図 3.10b) と、同等のスケーリング性能を示した。さらに、 $\text{Na}^+$  チャンネルに 12 状態からなる反応方程式を用いたオリジナルの顆粒細胞モデルと入力電流の強度に対するスパイク数を比較したところ、一致する結果を得られた (図 3.10c)。以上より、反応方程式を HH 型方程式に置き換え、モデルの硬さを取り除いたことで、入出力関係を変えることなく、計算時間の短縮に成功した。

次に、この顆粒細胞モデルを用いた小脳ネットワークモデルにおける RKC 法の性能評価を行った (図 3.11)。図 3.11a は図 3.9a と同様の顆粒細胞-ゴルジ細胞間のオシレーションとプルキンエ細胞の同期的な活動を示した。また、計算時間は全ての細胞モデルに RKC 法を適用したシミュレーションが CN 法よりも速いことを確認した (図 3.11b)。以上の結果から、モデルの硬さの原因となる短いダイナミクスを持つ反応方程式で記述されたイオンチャンネルを通常の HH 型方程式で置き換えることにより、モデルの硬さを回避し、RKC 法による計算時間の短縮が可能であることを提案する。

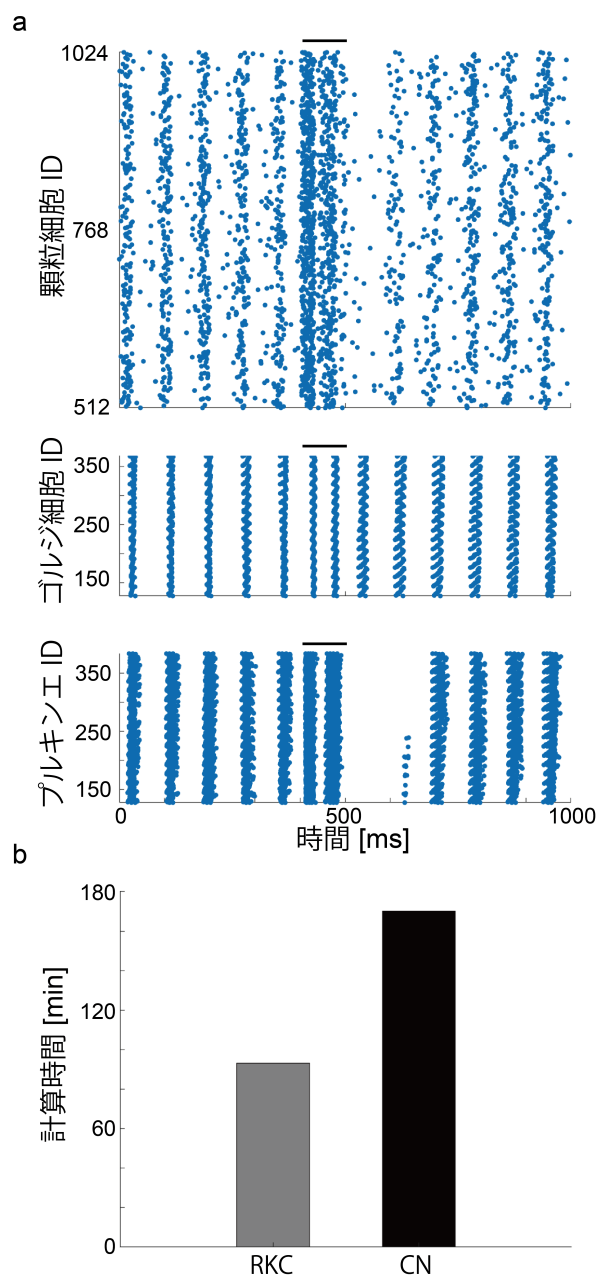


図 3.11: 改良した顆粒細胞モデルを用いた小脳ネットワークモデルシミュレーション。ネットワークモデルは 2048 個の顆粒細胞（総計 1183744 コンパートメント）、512 個のゴルジ細胞（総計 68513 コンパートメント）、512 個のプルキンエ細胞（総計 819200 コンパートメント）から構成されており、総シナプス数は 780404 個である。**a**: 各ニューロンモデルのラスタプロット。顆粒細胞への苔状線維入力が入力期間中は 30 Hz であり、期間外は 5 Hz である。図上部の実線は入力期間（400–500 ms）を表している。**b**: 小脳ネットワークモデルシミュレーションの計算時間。灰色は全てのニューロンモデルに RKC 法のみを適用した場合、黒は全てのニューロンモデルに CN 法のみを適用した場合のシミュレーションの計算時間を表す。

## 3.4 考察

### 3.4.1 マルチコンパートメントモデルシミュレーションへの陽解法の適用

本研究で行ったマルチコンパートメントモデルへの RKC 法の適用は、プルキンエ細胞、ゴルジ細胞、下オリーブ核ニューロンモデルの計算時間を短縮し、スケーリング性能を上昇させた。また、メモリのプロファイリング結果から、RKC 法は一般的な陰解法である CN 法と比較して、GPU 上のデバイスメモリへのデータ転送量を減少させた。これは CN 法を用いたシミュレーションは連立方程式の求解に大域的なメモリアクセスが必要となるため、データがキャッシュメモリから溢れてしまうが、RKC 法は連立方程式の求解を必要とせず、時間刻み幅の推定時はデータをキャッシュメモリに載せたまま何度も同じ計算を繰り返すことができることからデバイスメモリへのアクセスはほとんど起こらないことに起因する。以上の結果は、陽解法が同精度の陰解法よりも GPU の性能を引き出し、効率的な並列計算を可能とすることを示唆している。従来のシミュレータは、拡散方程式で記述されるマルチコンパートメントモデルを安定的に計算するために、陰解法を使用していたが [22–24, 30]、本研究は並列計算用ハードウェアのボトルネックとなるメモリアクセスを回避しうる陽解法が合理的な代替手段であることを提案する。

対して、RKC 法による顆粒細胞モデルのシミュレーションには、CN 法を用いた場合の 4.5 倍以上の計算時間を要し、細胞数の増加に対してこの傾向は悪化した。この結果は、RKC 法は全てのマルチコンパートメントモデルに対して盲目的に使用できるわけではないことを示している。RKC 法が本研究で用いた Dover らの顆粒細胞モデル [49] に適していない理由として、モデルの硬さが挙げられる。硬いモデルとは、モデルの全体的なダイナミクスに対して急激な変化を起こす項が含まれるモデルであり、陽的に数値計算をする場合、刻み幅を極めて小さくすることが求められる。マルチコンパートメントモデルの持つイオンチャネルは通常、開口と閉口の 2 状態を持つ HH 型方程式で記述されるが、この顆粒細胞モデルの  $\text{Na}^+$  チャネルは 12 状態の反応方程式によって記述されており [49]、特に  $\text{Na}^+$  チャネルはスパイクの生成に寄与するため、12 個の状態の遷移が瞬間的に引き起るような極めて短い時定数を持つ。よって、他のニューロンモデルと比較して、顆粒細胞モデルはより硬い性質を持ち、RKC 法は計算を安定させるため、特にスパイク生成時は最小  $2^{-12}$  ms 程の刻み幅を設定し、パフォーマンスが大幅に低下したと考えられる。一方で、ゴルジ細胞にも 6 状態の反

応方程式で記述された KAHP チャネルが存在するが、ゴルジ細胞は RKC 法による計算の高速化と良好なスケーリング性能を示している。これは、KAHP チャネルはスパイクを生成する  $\text{Na}^+$  チャネルよりも比較的遅いダイナミクスであることと状態数が少ないことが原因であると推定する。

これらの結果から、本研究は複数種類のニューロンからなるネットワークモデルをシミュレーションするにあたり、異なる数値計算手法を混合して適用した。具体的には、小脳ネットワークモデルのシミュレーションでは、ゴルジ細胞とプルキンエ細胞モデルにはそれぞれ RKC 法、顆粒細胞モデルに CN 法を適用した混合手法を用いて、全てのモデルを CN 法のみで計算した場合よりも計算時間の短縮に成功した。従来、マルチコンパートメントモデルのシミュレーションは単純化のために単一の数値計算手法のみを適用することが一般的であるが、本研究ではネットワークシミュレーションにおいて、各ニューロンの種類に応じ、異なる数値計算手法の適用がシミュレーションのパフォーマンスを向上させる可能性を提案する。

また、顆粒細胞のマルチコンパートメントモデルに対して RKC 法を適用するために、硬さの要因である反応方程式で記述された  $\text{Na}^+$  を HH 型方程式で記述しなおすことで、硬さを回避したモデルを作成し、RKC 法を適用したところ、計算時間の短縮に成功した。ネットワークモデルにおいても同様に、RKC 法によって計算時間を 45 % ほど短縮することに成功した。GPU などの並列計算用ハードウェアにおける陽解法の利点を考慮すれば、反応方程式を HH 型方程式に置き換えるようなモデルの変更によって硬さの回避を行うことは十分な意義が見込める。

本研究の結果は、今後更なる大規模化が必要とされるマルチコンパートメントネットワークモデルの効率的なシミュレーションには、モデルの性質（硬さ）に合わせた適切な数値計算法の選定や、ハードウェアと数値計算法の性質に合わせたモデルの変更が必要であることを示唆する。

### 3.4.2 GPU を用いたマルチコンパートメントモデルの並列シミュレーション

GPU は計算科学のあらゆる分野において、シミュレーションを高速化するために広く使用されている。計算神経科学の分野においても、GPU はシミュレーションの高速化にあたってすでに利用されている [22, 23]。特に、ニューロンの空間形状を考慮しないシングルコンパートメントモデルの膜電位ダイナミクスは常微分方程式によって記述され、陽解法によって簡単に計算できるためその並列化は容易である。一方で、マルチコンパートメントモデルのダ



イナミクスは偏微分方程式（拡散方程式）によって記述される性質上、計算の安定化のために陰解法が用いられ、並列化にあたり、連立方程式の求解がメモリアクセスを多用し、計算のボトルネックとなっていた。本研究において提案した、拡散方程式用に開発された陽解法をマルチコンパートメントモデルへ適用する試みは、連立方程式の求解を回避しメモリアクセス回数を低減することで、GPU の性能を十分に引き出すことを可能とする。特に、ニューラルネットワークモデルのシミュレーションは、多数（ヒトであれば 860 億ニューロン）のモデルの同時計算であるため、並列化と相性が良い。今後の計算神経科学分野では、計算用ハードウェアと解くべきモデルに適切な数値計算法を適用・開発することが重要な課題になると考える。

### 3.4.3 マルチコンパートメントモデルを用いたプルキンエ細胞間の同期活動

マルチコンパートメントモデルの利点として、ニューロン間にギャップ結合を導入し、直接的な膜電位の伝達をシミュレーションできることが挙げられる。ギャップ結合はニューロンの種類によって特定の部位にのみ存在するため、空間形状を考慮していないシングルコンパートメントではその影響を十全にシミュレーションすることが難しい。この利点を生かし、本研究では同一の Sagittal 軸方向に（同一矢状面上）に存在する隣接したプルキンエ細胞をギャップ結合で接続した。結果として、プルキンエ細胞群は約 60 Hz から 80 Hz の発火率で同期的な活動を示し、ギャップ結合がニューロン群を同期させたことを示唆している。

ギャップ結合によるニューロンの同期活動はすでに深く研究がなされているが [60, 61]、プルキンエ細胞間にギャップ結合が存在する可能性を示している研究は Middleton らによる生理学実験のみである [62]。本研究では、密接するプルキンエ細胞の樹状突起間は膜電位を直接的に伝達していると仮定し、ギャップ結合を導入した（具体的な設定方法は 3.2.2 節を参照）。プルキンエ細胞の同期活動については、ギャップ結合のような直接的な経路だけではなく、入力同期や細胞外のイオン濃度による同期など様々な説が存在しており、またその役割に関しても更なる研究が必要とされている [55, 56, 62–65]。

### 3.4.4 マルチコンパートメントモデルによるネットワークシミュレーション

顆粒細胞、ゴルジ細胞からなる小脳皮質の顆粒細胞層モデルのシミュレーションは Sudhakar らによって行われており、10 Hz 程度の同期的な自発活動を持つことが示されてい

る [40]。図 3.9a における入力期間外の顆粒細胞とゴルジ細胞は 10 Hz 程度の同期的な活動を示しており、ネットワークダイナミクスの再現性が示されている。本研究で構築した小脳ネットワークモデルは、顆粒細胞とゴルジ細胞、そしてプルキンエ細胞のマルチコンパートメントモデルによって構成されており、今後は顆粒細胞における苔状線維入力の情報統合 [66]、ゴルジ細胞の周期的な活動 [50]、プルキンエ細胞の記憶容量 [37]、瞬目反射条件づけ中のプルキンエ細胞樹状突起内の過渡的情報処理 [67] といった様々な小脳の機能をネットワークレベルで調査するために使用できる。一方で、より大規模なネットワークモデルを用いたシミュレーションを実行するためには、更に効率的な数値計算法の適用・開発や、適切なニューロンモデルの選択がより重要になる。本研究は、これらの問題に取り組むための基礎的知見を提供した。

### 3.4.5 今後の課題

マルチコンパートメントモデルへの RKC 法の適用にはいくつかの制限が存在するが、最も重要な問題は硬いニューロンモデルに適用できない点である。本研究では、この問題の解決方法の 1 つとして、イオンチャネルの変更によるモデルの改良案を提示したが、反応方程式を用いることでしかニューロンのダイナミクスを再現・調査することができないような状況ではこの方法をとることができない。硬いモデルを計算するための手法の開発は、計算科学における主要な問題であり、これまでに様々な方法が試みられている [68, 69]。硬いマルチコンパートメントモデルを効率的に計算するためのアイデアの 1 つとして、計算途中で陰解法と陽解法を切り替えるようなスイッチング法 [70] などが挙げられる。これは、基本的には陽解法を用いて計算し、システムの硬さが増すタイミング（例えばスパイク生成時など）に陰解法へと切り替える手法であるが、精度を維持しながら解法を切り替える適切なタイミングの検知や初期値の設定方法といった諸問題が存在する。

別の問題点として、本研究におけるネットワークモデルのシミュレーションでは、シナプスの計算時間は非常に小さい割合であったが、コンパートメント数が少ないニューロンモデルで構成される高密度なネットワークモデルのシミュレーションでは、シナプスの計算が主要な要因になる可能性が上げられる [71]。そのような場合、膜電位、イオンチャネル、そしてシナプスの計算を並列化するような手法が必須となる。数値計算法の改善や並列化の技術問題などは、神経科学における本質ではないが、次世代のスーパーコンピューティングに必要となるマルチコンパートメントモデルのシミュレーションをより効率的に実行するため、学

際的な視点で計算科学などの他分野と協力し、技術を組み込むことが、今後の計算神経科学分野における重要な課題であると考えられる。

## 3.5 結論

本研究では、マルチコンパートメントモデルのシミュレーションに陽解法を適用することを提案した。陽解法は陰解法と比較して、GPU 上のデバイスメモリへのアクセス回数を低減し、計算時間を短縮、適度な硬さを持つニューロンモデルのスケーリング性能を向上させた。マルチコンパートメントモデルのための高性能な数値計算法は、空間形状を考慮した全脳モデルの大規模シミュレーションを実現するために有効な手段となりうる。

## 第4章

# マルチコンパートメントモデルシミュレーションによるプルキンエ細胞の空間形状が持つ機能的役割の検証

### 4.1 はじめに

中枢神経系における情報処理は、ニューロンと呼ばれる神経細胞がネットワークを構築し、スパイク活動を伝達することで情報を伝搬・処理している。従来の神経科学分野では、個々のニューロンは情報を積算する素子であり、複雑な脳の機能はネットワークのダイナミクスによって生じるとされてきた。しかし近年の研究により、ニューロンはその複雑な空間形状を用いて、単体でも非線形な論理演算や刺激シーケンスの弁別といった高度な情報処理を行っていることが明らかにされている [15–17, 20, 28, 37]。一方で、このようなニューロンの空間形状の機能的役割について、ネットワークダイナミクスや脳機能のような視点からの考察はほとんど行われていない。よって、このニューロンのマイクロな空間形状が、ネットワークダイナミクスを介してマクロな脳機能にどのような影響を与えているか調査することは、今後の神経科学分野における重要な課題である。

具体的な空間形状と脳機能の関係性として、神経変性疾患におけるニューロンの形状の変性と脳機能障害が報告されている。例えば、神経変性疾患の一種である脊髄小脳変性症 I 型

では、運動失調と同時に、運動の調整などを司る小脳のプルキンエ細胞の樹状突起が縮退すると報告されている [27]。プルキンエ細胞は小脳皮質で処理された運動情報を出力するニューロンであり、複数のプルキンエ細胞からの出力信号は深部小脳核ニューロンで統合され、運動系を調節する信号として大脳皮質や脳幹へと出力される [1]。この時、正常時におけるプルキンエ細胞は矢状面方向のプルキンエ細胞間で同期的な活動を示しており、この同期活動が深部小脳核に規則的な活動を促し、その情報を運動前野など他の脳部位へ中継することで、運動の開始や調節に関与していることが報告されている [72-74]。よって、プルキンエ細胞の神経変性は小脳における情報処理を変調させ、深部小脳核ニューロンの運動調整機能に影響を与えることが予想されるが、脊髄小脳変性症 I 型における神経変性と運動失調の関係性は相関関係であり、この因果関係は完全には解明されていない。

このようなニューロンの空間形状と脳機能やネットワークダイナミクスの因果関係を調査するには、実際に形状を変更しつつニューロン全体の活動を記録する必要がある。空間形状をパラメータとして任意に仮定・変更可能な数値シミュレーション手法はこのような因果関係の調査に有用である。そこで本研究では、ニューロンの空間形状を考慮したモデル（マルチコンパートメントモデル）を用いて、樹状突起が縮退したプルキンエ細胞モデルを作成し、神経変性と運動失調の因果関係を考察する。具体的には、「小脳への入力情報をリレーする顆粒細胞」、「顆粒細胞の活動を調整するゴルジ細胞」、「小脳で処理された運動情報を出力するプルキンエ細胞」、「複数のプルキンエ細胞からの信号を統合し運動系の調節信号を担う深部小脳核ニューロン」からなる小脳ネットワークモデルを構築し、数値シミュレーションを行う。特に、深部小脳核ニューロンはプルキンエ細胞の同期的な活動によってその活動が変調され、運動行動などを調整するため、プルキンエ細胞の形状変化がプルキンエ細胞間の同期活動にどのような影響を与えるか、そして深部小脳核ニューロンの活動がどのように変調されるかに着目する。これにより、プルキンエ細胞が持つ空間形状の機能的役割について考察する。

## 4.2 方法

### 4.2.1 ニューロンモデルとネットワークモデル

本研究で用いた、小脳の顆粒細胞 [49]、ゴルジ細胞 [50]、プルキンエ細胞 [31, 32] モデルは第 3 章で用いたマルチコンパートメントモデルと同様である (図 3.1)。ただし、顆粒細胞モデルには、HH 型方程式で記述された  $\text{Na}^+$  チャネル [57] を反応方程式の代わりに使用してい

表 4.1: 各種シナプスにおけるパラメータと減衰関数。

パラメータ名	値	単位
$c_m$	122.3	[pF]
$g_l$	1.63	[nS]
$E_l$	-56.0	[mV]
$\alpha_{\text{GABA}_A}(t)$	$e^{-t/42.3}$	
$\bar{g}_{\text{GABA}_A}$	30	[nS]
$E_{\text{GABA}_A}$	-88.0	[mV]
$\theta$	-38.8	[mV]
$I_{\text{ext}}(t)$	150	[pA]

る。また、プルキンエ細胞モデルは正常状態のモデルと、樹状突起のコンパートメント数を半分ほど減らした縮退状態のモデルを用意し、活動を比較した (図 4.1a)。

また、本研究では深部小脳核ニューロンには積分発火モデル (シングルコンパートメント) を用いた。積分発火モデルは次式のように、膜電位  $v(t)$  のダイナミクスを膜のキャパシタンス  $c_m$ 、リークチャネルのコンダクタンス  $g_l$  と反転電位  $E_l$  を用いて 1 次の微分方程式で表す。

$$c_m \frac{dv(t)}{dt} = -g_l(v(t) - E_l) - I_{\text{GABA}_A}(t) + I_{\text{ext}}(t) \quad (4.1)$$

ただし、 $I_{\text{GABA}_A}(t)$  と  $I_{\text{ext}}(t)$  はそれぞれプルキンエ細胞からの抑制性シナプス性電流と自発活動を促す外部入力を表す。膜電位  $v(t)$  が閾値  $\theta$  を超えると発火したとみなし、膜電位を  $E_l$  までリセットする。各パラメータの具体的な値は表 4.1 に示す。

本研究で用いたネットワークモデル (図 4.1b) は、小脳皮質において最も重要である顆粒細胞 (4096 個)、ゴルジ細胞 (4 個)、プルキンエ細胞 (16 個) の 3 種類のニューロンと、小脳の最終的な出力にあたる深部小脳核ニューロン (1 個) の合計 4 種類のニューロンモデルから構築され、3 次元空間上に配置されている。本研究では、計算時間の観点から、学習中のシミュレーションを行わないため、運動学習において重要な役割を持つ下オリーブ核ニューロンは導入していない。また、顆粒細胞とゴルジ細胞の細胞数の比は Casali らのマウス小脳モデルに基づき、1000:1 の割合で設定した [75]。

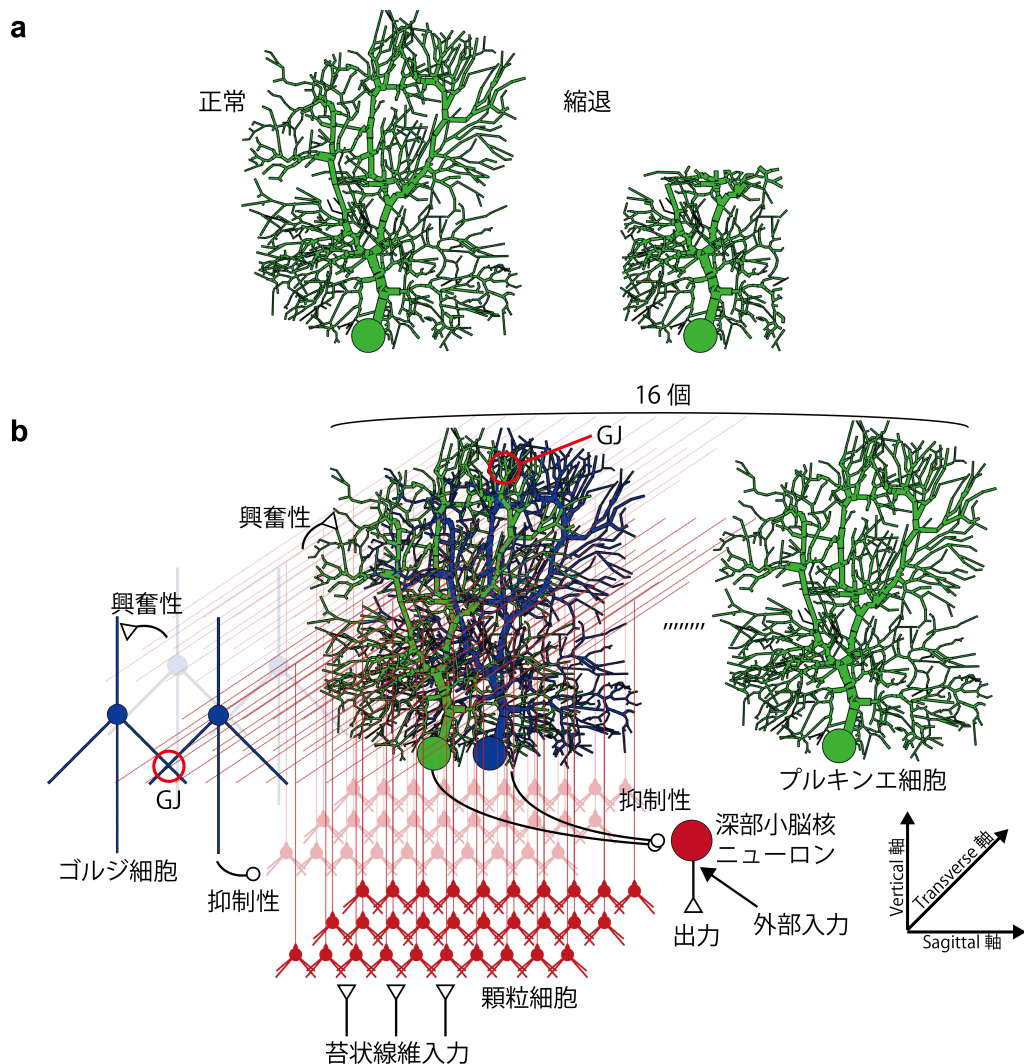


図 4.1: モデルの概要図。a: プルキンエ細胞のマルチコンパートメントモデル。左は 1600 コンパートメントからなる正常状態のプルキンエ細胞モデル [31,32] であり、右は樹状突起の縮退を再現するため、735 個までコンパートメント数を減らしたモデルである。b: 小脳ネットワークモデル。4096 個の顆粒細胞、4 個のゴルジ細胞、16 個のプルキンエ細胞、1 個の深部小脳核ニューロンから構成される。各ニューロンは 3 次元空間上に配置されている。顆粒細胞は苔状線維からの興奮性のシナプス結合（三角印）と近隣のゴルジ細胞からの抑制性のシナプス結合（丸印）を受ける。ゴルジ細胞とプルキンエ細胞は顆粒細胞の平行線維から興奮性のシナプス結合を受ける。深部小脳核ニューロンは興奮性の外部入力とプルキンエ細胞からの抑制性のシナプス結合を受ける。ゴルジ細胞は隣接するゴルジ細胞間で樹状突起上にギャップ結合（Gap Junction: GJ）を形成している。同様に、Sagittal 軸方向に 16 個並んで配置されたプルキンエ細胞もギャップ結合によって、電氣的に結合している。Sagittal 軸と Transverse 軸、Vertical 軸はそれぞれ、冠状面と矢状面、横断面に垂直な方向の軸を表す。また、視認性向上のため、ニューロン数は本来よりも少なく表示している。

表 4.2: シナプス結合荷重の値

シナプス前細胞	シナプス後細胞	値
苔状線維	顆粒細胞	40000.0
顆粒細胞	ゴルジ細胞	16.0
ゴルジ細胞	顆粒砂防	640000.0
顆粒細胞	プルキンエ細胞（正常）	1000000.0
顆粒細胞	プルキンエ細胞（縮退）	500000.0
プルキンエ細胞	深部小脳核ニューロン	12000.0

ネットワークモデルにおいて、顆粒細胞は苔状線維、ゴルジ細胞とプルキンエ細胞は顆粒細胞から興奮性のシナプス結合を受ける。また、顆粒細胞はゴルジ細胞、深部小脳核ニューロンはプルキンエ細胞から抑制性のシナプス結合を受ける。プルキンエ細胞から深部小脳核ニューロンへのシナプス電流は、結合荷重  $w$ 、最大コンダクタンス  $\bar{g}_{\text{GABA}_A}$ 、反転電位  $E_{\text{GABA}_A}$ 、減衰関数  $\alpha_c(t)$ 、発火活動  $\delta_b$  を用いて次式のように表す。

$$I_{\text{GABA}_A}(t) = \bar{g}_{\text{GABA}_A} w (v(t) - E_{\text{GABA}_A}) \int_{-\infty}^t \alpha_c(t-s) \delta_b(s) ds \quad (4.2)$$

その他のシナプス結合モデルは第 3 章と同様であり、式 (3.1) と表 3.1 に示すとおりである。

本ネットワークモデルで用いたシナプス結合荷重の値を表 4.2 に示す。実際にシミュレーションを行う場合、表 4.2 の値を基準にシナプス後細胞が受け取るシナプス数によって正規化している。樹状突起を縮退させたプルキンエ細胞モデルは、正常なモデルよりも顆粒細胞から受け取るシナプス数が減少し、活動が極端に強くなりすぎるため、結合荷重を半分に行っている。また、顆粒細胞からプルキンエ細胞へのシナプス結合は、表の値を平均とした正規分布に基づく乱数によって値に幅を持たせている。

近隣のゴルジ細胞は樹状突起同士がギャップ結合と呼ばれる電気的な結合によって互いに接続し、同期的な活動を示す [52–54]。本研究ではゴルジ細胞間のギャップ結合として、最も簡易的なモデル化として、第 3 章同様に式 (4.3) を実装した。

$$I_{\text{gj},a,i;b,j} = \bar{g}_{\text{gj}} (v_{b,j} - v_{a,i}), \quad (4.3)$$



ただし、 $i$  と  $j$  はそれぞれギャップ結合で接続しているゴルジ細胞  $a$  と  $b$  の樹状突起のコンパートメント番号を、 $\bar{g}_{gj}$  はギャップ結合の最大コンダクタンス (2.5 nS [52,53]) を表す。

本研究では、プルキンエ細胞間の同期現象 [55,56] を再現するため、隣合うプルキンエ細胞の樹状突起がそのコンパートメントの半径よりも小さい場合、ギャップ結合 (式 (4.3)、2.5 nS) していると仮定した。これにより、小脳ネットワークモデルの Sagittal 軸方向に並んだ 16 個のプルキンエ細胞はギャップ結合によってネットワークを構築している。また、本研究で用いたプルキンエ細胞モデルは強い活動を示し過ぎる傾向がある [31,32] ため、第 3 章と同様に、CaP チャネルのコンダクタンスを 4.5 nS から 3.0 nS に、K2 チャネルのコンダクタンスを 0.39 nS から 0.9 nS に変更した。

#### 4.2.2 相互相関解析

本研究では、プルキンエ細胞間が持つ空間形状の機能的役割について、プルキンエ細胞の神経変性が細胞間の同期活動にどのような影響を与えるのかに着目する。プルキンエ細胞間の同期活動を定量的に評価するため、本研究では神経スパイクの相互相関解析を行う [76]。スパイク活動の相互相関解析は、計測期間を  $T$  [ms] の時間ビンに区切り、ある時刻  $t$  における 2 つのニューロンの活動を  $x(t)$  と  $y(t)$ 、活動の時間差を  $\tau$  として次式のように評価する。

$$C_{xy}(\tau) = \sum_{t=1}^T x(t)y(t+\tau), \quad (4.4)$$

ここで、ニューロンの活動  $x(t)$  と  $y(t)$  は神経スパイクであるため、スパイク発生時刻では 1、そうでなければ 0 を持つ。相互相関関数  $C_{xy}(\tau)$  は 2 つの活動の時系列における類似度を示しており、同期傾向にある場合は時間差 0 においてピークを持ち、そうでなければ全ての時間差に対して平坦な値を持つ。本研究では、500 ms 間の活動量を測定し、時間ビン  $T$  は 1 ms とした。解析には数値解析ソフトウェアである MATLAB を用いた。

また、本研究で用いる小脳ネットワークモデルには、Sagittal 軸方向に並んだ 16 個のプルキンエ細胞が存在する。そこで本研究では、両端 (1 番目と 16 番目) のプルキンエ細胞間の相互相関、および 1 番目のプルキンエ細胞に対する他の 15 個のプルキンエ細胞間との相互相関の平均値を用いて解析を行う。

### 4.2.3 瞬目反射条件づけ

本研究では、プルキンエ細胞の空間形状の機能的役割を具体的な運動タスクの観点から考察するために、小脳が重要な役割を果たす、瞬目反射条件づけのタスクに着目しシミュレーションを行う。このタスクは、マウス、ウサギ、ヒトなど様々な動物で研究が進んでおり、また多様な神経疾患における研究が行われているため [77,78]、詳細な神経メカニズムが解明されており、疾患モデルの機能評価として有用である。

瞬目反射条件づけは、聴覚への音刺激などを条件刺激（CS：conditioned stimulus）とし、眼球へのエアパフ刺激などを無条件刺激（US：unconditioned stimulus）として、同時に繰り返し提示し続けることで、条件刺激だけでまばたきなどの条件反射（CR：conditioned response）を引き起こすように学習する古典的条件づけである。特に、CSを与えている間にUSを与える遅延課題においては、その運動学習のメカニズムが小脳に起因することが知られており、平行線維とプルキンエ細胞間のシナプス結合荷重の長期抑圧（LTD：Long Term Depression）が重要な役割を果たすとされている [79]。具体的な神経メカニズムを図 4.2 に示す。学習前ではプルキンエ細胞は顆粒細胞を経由して音刺激（CS）の入力によるバースト発火が起こり、深部小脳核のニューロンはプルキンエ細胞からの抑制を受け、瞬きが起きない。瞬きが起きない場合、プルキンエ細胞はエアパフ（US）刺激を下オリーブ核ニューロンを経由して教師信号として受けることで、CS 入力を運ぶ平行線維（顆粒細胞軸索）とプルキンエ細胞間のシナプス結合荷重が長期抑圧（LTD）される。これを繰り返すことで、学習後は CS 入力に対してプルキンエ細胞は活動を示さなくなるため、深部小脳核ニューロンへの抑制が外れ活性化されるため、瞬目の主動筋である眼輪筋へと情報が伝達され、瞬き（CR）が起こるようになる。

本研究では、この瞬目反射条件づけのシミュレーションを行うため、4096 個の顆粒細胞の内、256 個の顆粒細胞に CS 刺激に対応する入力として、300 ms の間 30 Hz の苔状線維入力を与えた。入力期間外とその他の顆粒細胞は背景活動として常に 5 Hz の苔状線維入力を受けている。一方で、現状のシミュレータでは、計算時間の観点から学習状態の長時間のシミュレーションを行うことは困難である。そこで、LTD による学習を再現するために、学習後のシミュレーションでは CS 刺激に対応する顆粒細胞とプルキンエ細胞間の結合荷重を 0.1 倍に変更しシミュレーションを行った。よって、本研究では教師信号に対応する下オリーブ核ニューロンモデルはネットワークモデルに導入していない（図 4.1b）。また、本研究では

CS 刺激に対応する顆粒細胞-プルキンエ細胞間の全シナプス結合の荷重を一律で変化させたため、瞬きのタイミング学習ではなく、CS 刺激に対して CR を起こす古典的条件づけを学習している。

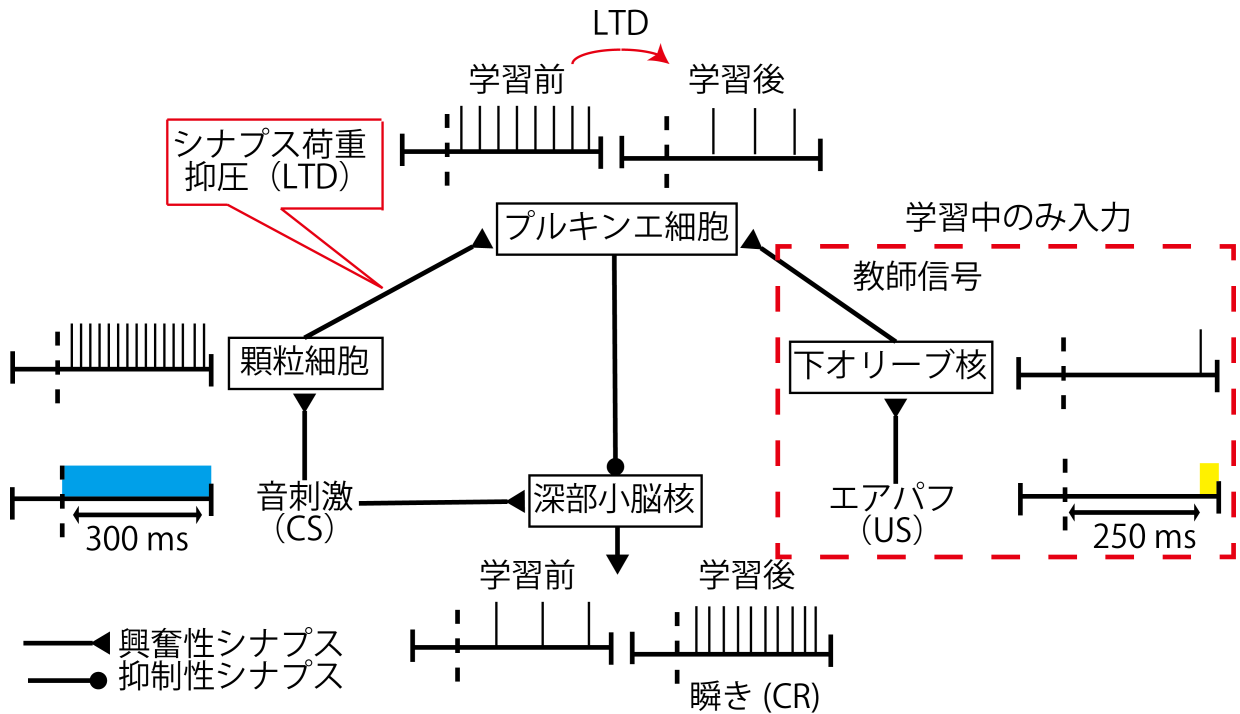


図 4.2: 小脳神経回路内における瞬目反射条件付けの神経メカニズム。学習前では、プルキンエ細胞は音刺激 (CS) によって活動し、深部小脳核ニューロンは抑制を受けるため瞬きが起こらない。学習中では、プルキンエ細胞はエアパフ (US) 刺激を下オリーブ核ニューロンを経由して受けることで、平行線維-プルキンエ細胞間のシナプス結合荷重が長期抑圧 (LTD) される。学習後は CS 入力に対してプルキンエ細胞は活動を示さなくなるため、深部小脳核ニューロンへの抑制が外れ活性化されるため、瞬き (CR) が起こるようになる。

#### 4.2.4 数値計算法とシミュレーション条件

本研究では、第 3 章で有効性を示した、Runge-Kutta-Chebyshev (RKC) 法をマルチコンパートメントモデル用の数値計算法として採用した。また、積分発火モデルを用いている深部小脳核ニューロンモデルの計算には、陽的オイラー法を用いており、刻み幅は 0.005 ms としている。

ネットワークモデル以外のシミュレーションにおいて、外部入力<sup>1</sup>は細胞体への定電流入力として与え、細胞体の膜電位を測定している。ネットワークモデルのシミュレーションでは、苔状線維を 5 Hz で活動させ、250 ms のフリーランの後、膜電位などを測定した。また、ニューロンの発火時刻を示すラスタプロットは、細胞体の膜電位が一定の閾値 (-10 mV) を超えた時刻を出力した。全てのシミュレーションにおいて、各ニューロンの膜電位は静止電位から最大 5 mV ほどランダムにずれた初期値を与えシミュレーションを行った。

## 4.3 結果

### 4.3.1 プルキンエ細胞モデルのギャップ結合による同期活動

はじめに、ギャップ結合によって、異なる入力を与えた 2 つのプルキンエ細胞の活動が同期するか調査した (図 4.3)。正常なプルキンエ細胞モデルにおいて、ギャップ結合は異なる位相の正弦波状の入力電流を与えたプルキンエ細胞間の同期的なスパイク活動を促した (図 4.3a)。一方で、縮退モデルでは、ギャップ結合は 2 つのプルキンエ細胞を十分に同期させることができなかった (図 4.3b)。これは、樹状突起の縮退により、プルキンエ細胞間のギャップ結合の数が減少したことに起因すると考えられる。正常なプルキンエ細胞の複雑に分岐した樹状突起は、隣接するプルキンエ細胞とのギャップ結合を介した同期的な活動を促進する役割を持つと考えられる。また、縮退モデルのシミュレーション結果は、正常なモデルと同じ強度の入力を与えたにもかかわらず、スパイク数が増加した。これは樹状突起の縮退により、樹状突起上の  $K^+$  チャネルが欠落したことに起因すると考えられる。そこで、入力の強度を半分にしてシミュレーションしたところ、スパイク数が正常なモデルと一致した (図 4.3c)。よって、以降のシミュレーションは、縮退モデルへの電流の強度およびシナプス結合荷重の値を正常モデルの半分に設定する。

### 4.3.2 プルキンエ細胞の空間形状変化による深部小脳核ニューロンへの影響

プルキンエ細胞の樹状突起が、ギャップ結合による同期的な活動を介して、深部小脳核ニューロンへどのような影響を与えるか調査するために、正常なプルキンエ細胞モデルと樹状突起を縮退させたプルキンエ細胞モデルを用いて、小脳ネットワークモデルのシミュレーションを行った (図 4.4)。10 から 20 Hz 程の顆粒細胞の振動的な活動 (図 4.4a) に対して、正常なプルキンエ細胞モデルを用いたシミュレーション (図 4.4 左) では、プルキンエ細胞

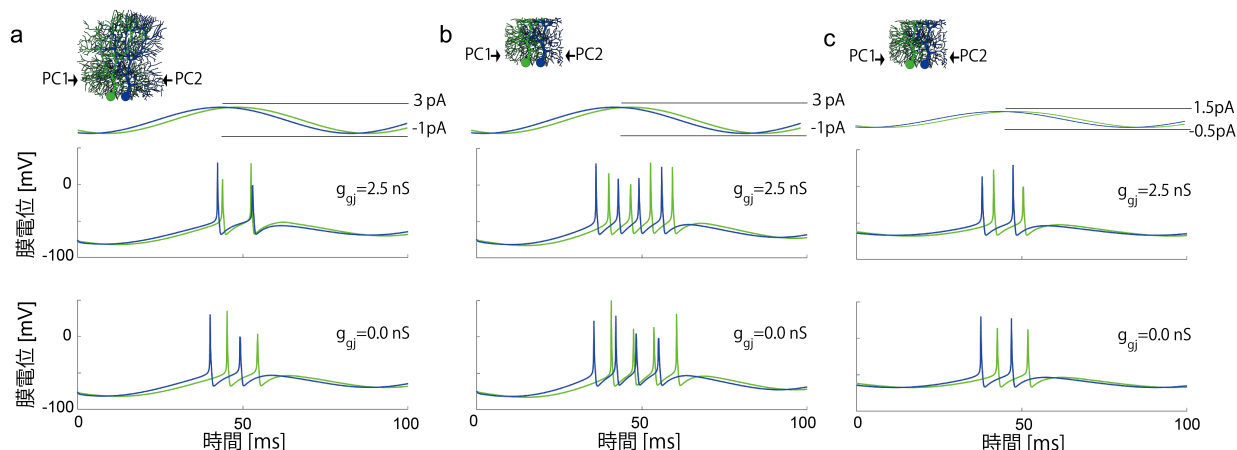


図 4.3: 位相の異なる正弦波上の電流入力を与えた2つのニューロンのギャップ結合によるスパイクタイミングの同期。図上部に示す、位相の異なる正弦波入力をもつ2つのプルキンエ細胞 (PC1 (緑) と PC2 (青)) に与えた。正常なプルキンエ細胞モデル (a) と縮退モデル (b) における膜電位応答。上がギャップ結合 ( $g_{gj} = 2.5 \text{ nS}$ ) によって2つのプルキンエ細胞 (PC1 と PC2) が同期する場合、下がギャップ結合のコンダクタンスを  $0 \text{ nS}$  に設定し、同期をしない場合の応答。また、c は縮退モデルにおける正弦波入力の振幅を半分にした場合の膜電位応答を表す。

間の活動に同期が見られ、プルキンエ細胞-深部小脳核ニューロン間のシナプス結合によるリズムミカルな抑制は深部小脳核ニューロンに規則的な活動を示している。一方で、縮退したプルキンエ細胞モデルを用いたシミュレーション (図 4.4 右) では、プルキンエ細胞は非同期的な活動を示している。そのため、プルキンエ細胞による抑制性電流が常時与えられるため、深部小脳核ニューロンは不規則的な発火活動を示す。

次に、小脳ネットワークモデルシミュレーションにおける、プルキンエ細胞間の活動の同期を定量的に評価するため、スパイク活動の相互相関解析を行った (図 4.5)。正常なプルキンエ細胞モデルを用いたシミュレーションにおいて、ネットワークモデルの両端に存在するプルキンエ細胞間の相互相関関数は時間差  $0 \text{ ms}$  付近でピークを持ち、神経スパイクの発生時刻が同期している (図 4.5a 左)。一方で、樹状突起が縮退したモデルではピークが存在しない平坦な値を示すため、活動の時間的な相関関係が示されない (図 4.5a 右)。また、ネットワークモデル内のすべてのプルキンエ細胞間との相互相関関数の平均を計算した結果においても同様の傾向が示された (図 4.5b)。

これらの結果は、プルキンエ細胞の複雑に分岐した樹状突起がプルキンエ細胞間の同期的

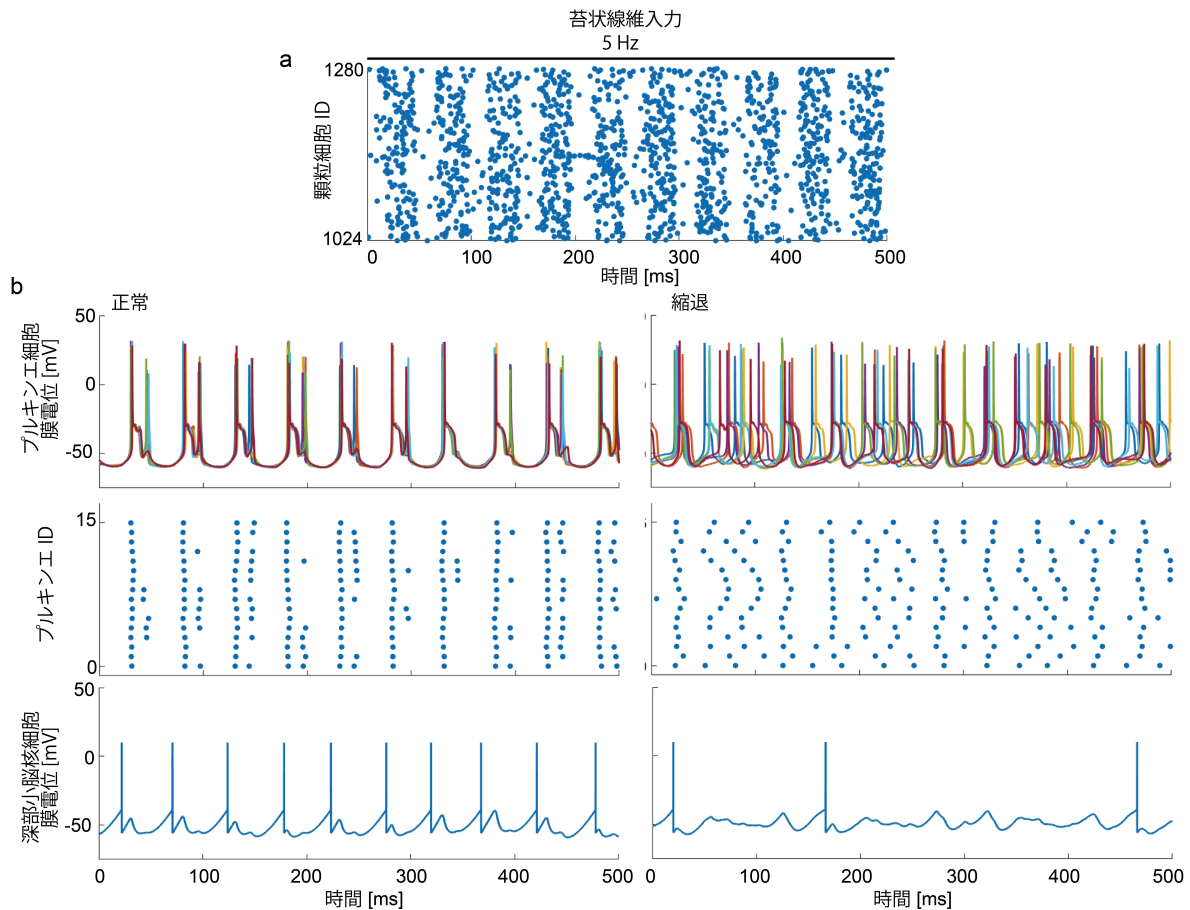


図 4.4: 小脳ネットワークモデルのシミュレーション。**a**: 顆粒細胞のラスタプロット。シミュレーション期間中、顆粒細胞は 5 Hz の苔状線維入力を受けている。4096 個の顆粒細胞に対して、1024 番目から 1280 番目のみをプロットしている。**b**: 正常なプルキンエ細胞モデルを用いた場合 (左) と縮退モデルを用いた場合 (右) のプルキンエ細胞と深部小脳核ニューロンの応答。上: プルキンエ細胞の膜電位変化。小脳ネットワーク内の 16 個のプルキンエ細胞の内、8 つの細胞の膜電位をプロットしている。中: 16 個のプルキンエ細胞のラスタプロット。下: 深部小脳核ニューロンの膜電位応答。深部小脳核ニューロンはシミュレーション期間中、外部から 175 pA の電流入力を受けている。

な活動を介し、深部小脳核ニューロンの規則的な活動を促進することを示唆する。また、樹状突起の縮退など、ニューロンの空間形状の変化は、空間形状に依存するギャップ結合を減少させ、ニューロンの同期活動や規則的な活動を阻害し、脳機能に影響を与える可能性を提案する。

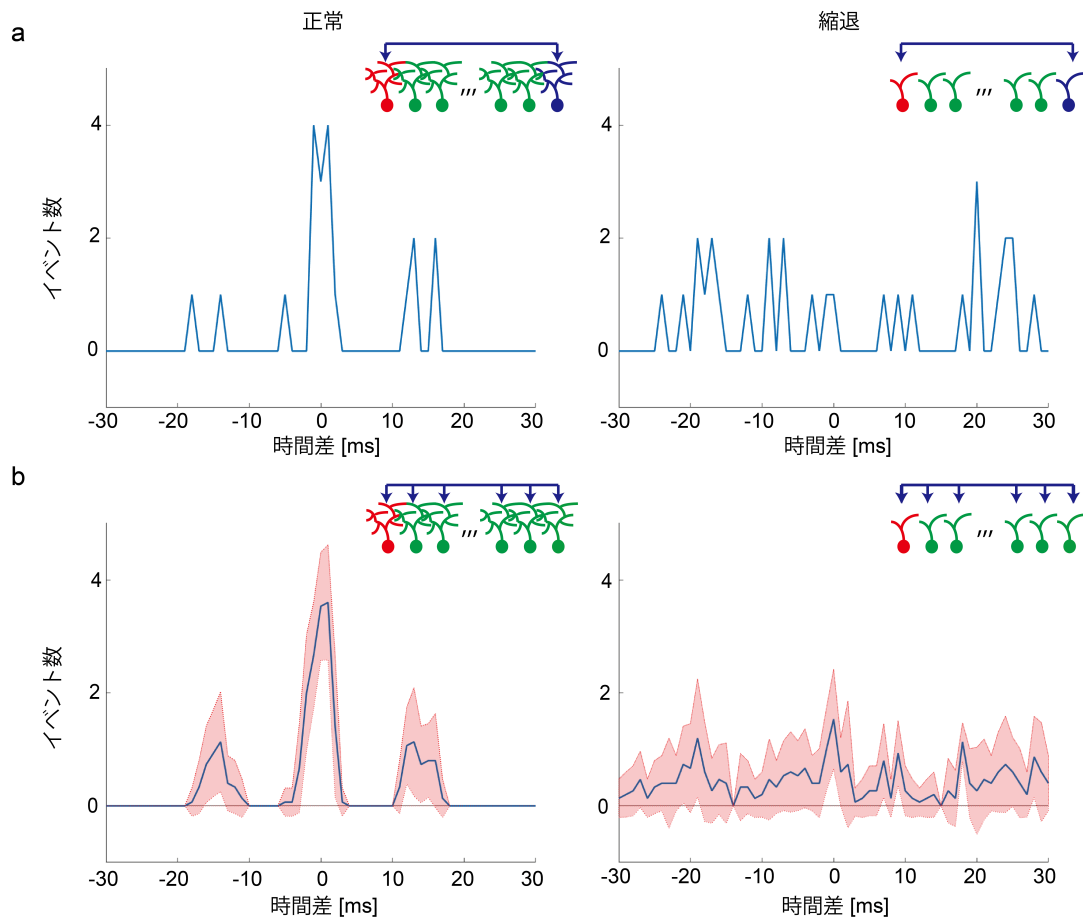


図 4.5: プルキンエ細胞の神経スパイクの相互相関解析。a: Sagittal 軸方向に並んだ 16 個の内、両端 (1 番目と 16 番目の) プルキンエ細胞の相互相関解析。b: 1 番目のプルキンエ細胞に対する他の 15 個のプルキンエ細胞間との相互相関の平均値 (実線) と標準偏差 (赤い領域)。a、b ともに左は正常なプルキンエ細胞、右は樹状突起が縮退したプルキンエ細胞モデルを用いたシミュレーション結果に対する解析。

### 4.3.3 瞬目反射条件づけにおけるプルキンエ細胞の空間形状が持つ機能的役割

具体的な運動タスクにおけるプルキンエ細胞の空間形状の機能的役割について検討するため、小脳が重要な役割を果たす瞬目反射条件付けを想定したシミュレーションを行った。

図 4.6 に学習前における、CS 入力に対する小脳ネットワークモデルのシミュレーション結

果を示す。CS 入力に対応する顆粒細胞は入力期間中、65 Hz 程度で応答を示している (図 4.6a)。学習前の場合、正常なプルキンエ細胞モデルと縮退モデルは同様に CS 入力に対して 60 Hz 程度のバースト応答を示し、深部小脳核ニューロンは常に抑制されるため活動が不活性化される (図 4.6b)。深部小脳核ニューロンは小脳の実出力として、瞬きを行う筋線維へと情報をリレーできず、瞬きが起きない。

次に、瞬目反射条件付けの学習後として、CS 入力に対応する顆粒細胞とプルキンエ細胞間のシナプス結合に長期抑圧 (LTD) を模した結合荷重の減弱 (0.1 倍) を行いシミュレーションを実行した (図 4.7)。正常なプルキンエ細胞モデルを使用した場合、LTD によって CS 入力に対応する顆粒細胞からの入力なくなり、プルキンエ細胞は 20–30 Hz 程度の同期的な自発活動のみを行うため、深部小脳核ニューロンは 60 Hz 程度で活動を示す (図 4.7 左)。これは、学習によって瞬きを行う筋線維へと情報がリレーされ瞬目反射条件付けが成功したことを示唆する。対して、縮退したプルキンエ細胞モデルを使用した場合も同様に、LTD によってプルキンエ細胞は 20–30 Hz 程度の自発活動のみを行うが、プルキンエ細胞間の活動が非同期的であるため、学習前同様に深部小脳核ニューロンが持続的に抑制されており、十分な発火活動を促すことができていない (図 4.7 右)。

よって、プルキンエ細胞の樹状突起の縮退はニューロン間の活動を非同期にすることで、深部小脳核ニューロンを持続的に抑制し続けるため、小脳における重要な LTD の効果が薄れ、運動学習による十分なゲインの増加を妨げる可能性を提案する。

## 4.4 考察

### 4.4.1 プルキンエ細胞の空間形状が持つ機能的役割

単一のプルキンエ細胞が持つ樹状突起は非常に複雑に分岐し、且つそれぞれの樹状突起は互いに重ならないように配置されている [80]。この特徴的な空間形状の機能的役割について、Zang らはそれぞれの樹状突起が固有の状態を保持することで、単体のプルキンエ細胞の計算能力が増加する可能性について指摘した [28]。また Brunel らは、プルキンエ細胞が樹状突起上の十数万ものシナプスを使用することで、最大 5 KBytes もの情報を学習可能であることを示唆している [37]。一方で、プルキンエ細胞のネットワークにおける空間形状の機能的役割についてはあまり検討されていない。

本研究では、Middleton らによる生理学実験 [62] を基に、マルチコンパートメントモデル



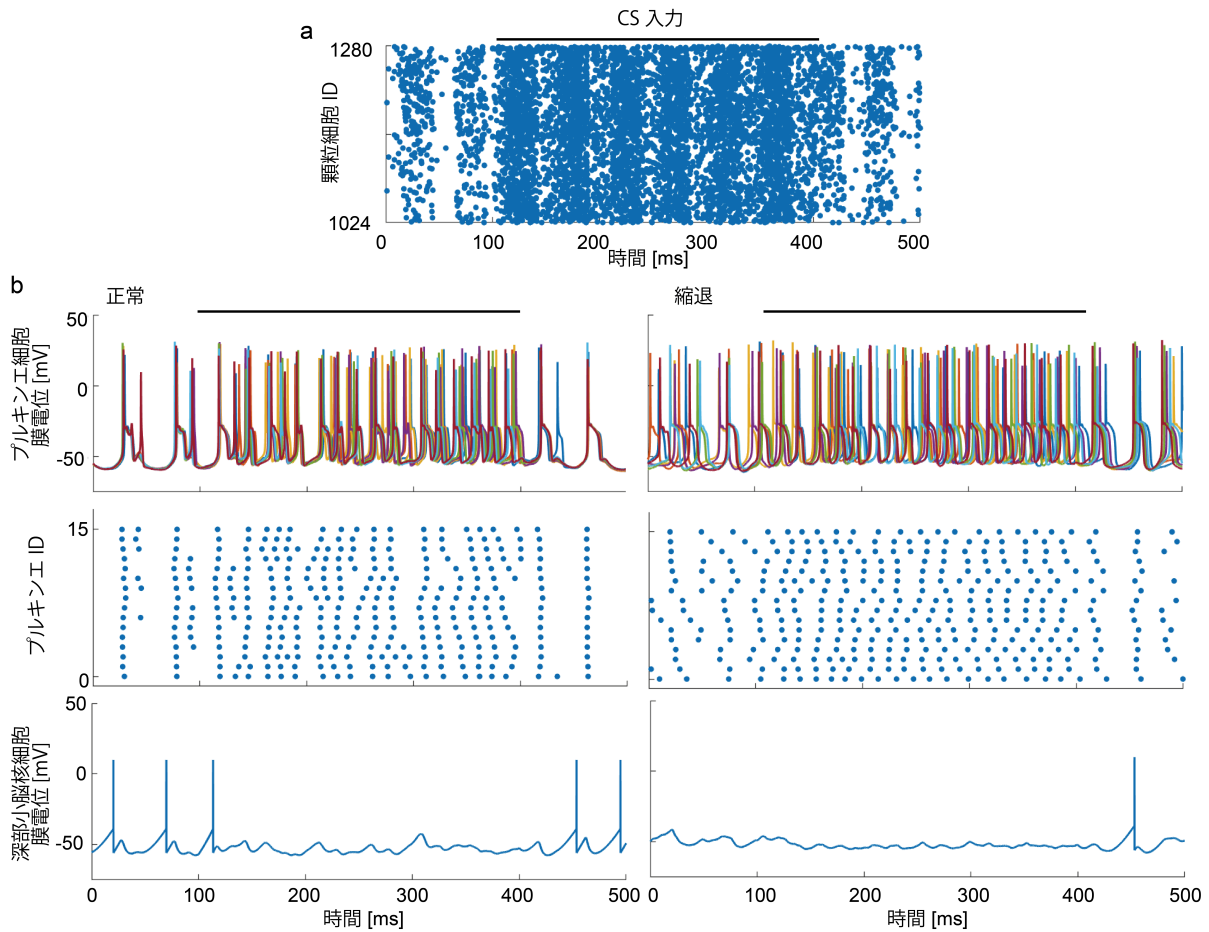


図 4.6: 瞬目反射条件付けにおける学習前のシミュレーション結果。a: 顆粒細胞のラスタプロット。4096 個の顆粒細胞に対して、1024 番目から 1280 番目のみをプロットしている。シミュレーション期間中、顆粒細胞は 5 Hz の苔状線維入力を受けており、CS 入力期間中（上部のバー）は 1024 番目から 1280 番目の顆粒細胞のみ 30 Hz の入力を受けている。b: 正常なプルキンエ細胞モデルを用いた場合（左）と縮退モデルを用いた場合（右）のプルキンエ細胞と深部小脳核ニューロンの応答。上: プルキンエ細胞の膜電位変化。小脳ネットワーク内の 16 個のプルキンエ細胞の内、8 つの細胞の膜電位をプロットしている。中: 16 個のプルキンエ細胞のラスタプロット。下: 深部小脳核ニューロンの膜電位応答。深部小脳核ニューロンはシミュレーション期間中、常に 175 pA の入力を受けており、CS 入力期間中はさらに 100 pA の追加入力を受けている。

の特性を利用して、同一矢状面上のプルキンエ細胞間の樹状突起にギャップ結合を仮定し、この電気的な結合によってスパイク活動が同期する可能性について検討した。結果として、豊富な樹状突起を持つ正常なプルキンエ細胞モデルを使用したシミュレーションではプルキ

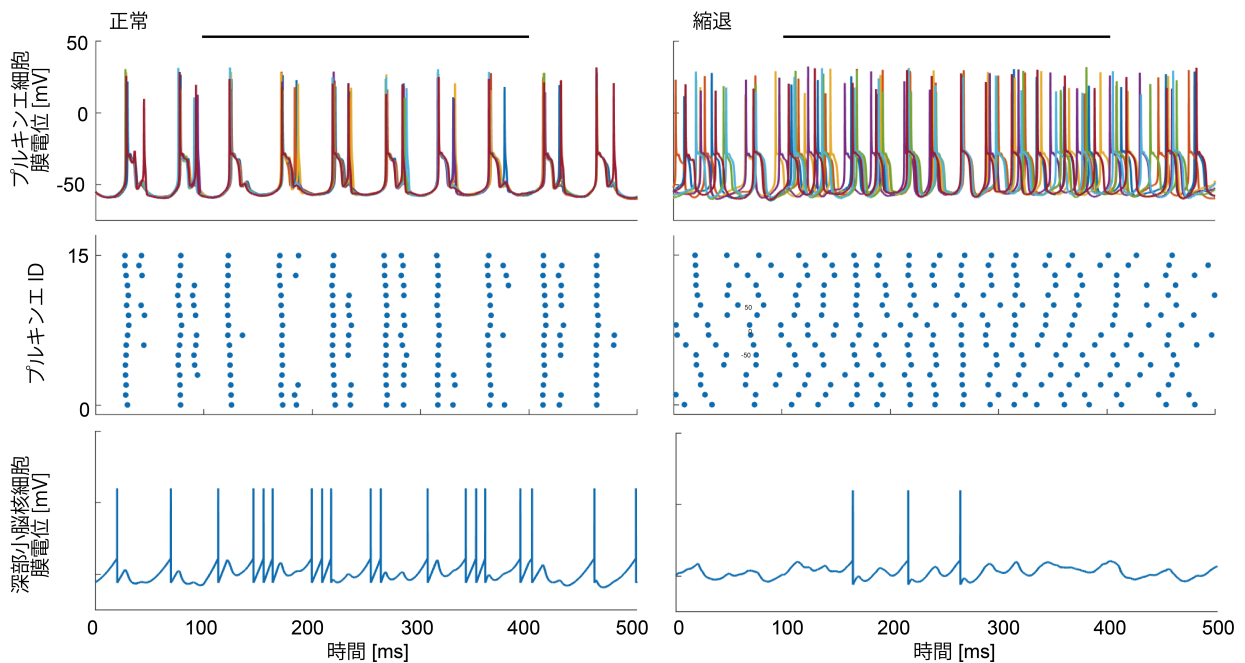


図 4.7: 瞬目反射条件付けにおける学習後のシミュレーション結果。正常なプルキンエ細胞モデルを用いた場合（左）と縮退モデルを用いた場合（右）のプルキンエ細胞と深部小脳核ニューロンの応答。上: プルキンエ細胞の膜電位変化。小脳ネットワーク内の 16 個のプルキンエ細胞の内、8 つの細胞の膜電位をプロットしている。中: 16 個のプルキンエ細胞のラスタプロット。下: 深部小脳核ニューロンの膜電位応答。深部小脳核ニューロンはシミュレーション期間中、常に 175 pA の入力を受けており、CS 入力期間中はさらに 100 pA の追加入力を受けている。顆粒細胞の応答は図 4.6a と同様。上部のバーは CS 入力期間を表している。

ンエ細胞間の同期現象が再現された。一方で、樹状突起のコンパートメント数を減少させた縮退モデルによるシミュレーションでは、ギャップ結合が失われたことでプルキンエ細胞間の活動が非同期化された。即ち、プルキンエ細胞の空間形状が同一矢状面上の細胞間の同期活動を促す役割を持つ可能性について提案する。

#### 4.4.2 プルキンエ細胞の同期活動のメカニズム

プルキンエ細胞間の同期活動に関して、顆粒細胞活動の源となる苔状線維が共通の入力によって同期的な活動を示すといった報告や、バスケット細胞からの抑制性入力による同期、局所的な細胞外電位による同期など、様々なメカニズムが提案されている [56,65]。一方で、

本研究で提案した、プルキンエ細胞間の樹状突起におけるギャップ結合による同期メカニズムは既存のメカニズムに矛盾しない。例えば、局所的な細胞外電位による同期 [65] は、隣り合うプルキンエ細胞間のスパイク同期を説明可能だが、実際には数十にわたるプルキンエ細胞の集団が同期的な活動を示している。バスケット細胞による抑制性入力は、最大 100 ms ほどのスパイク活動の停止期間を同期させることはできるが、背景活動におけるプルキンエ細胞は 30 Hz 程度でスパイクを発射するため、スパイクの同期を説明することができない。顆粒細胞の活動の源となる、苔状線維が共通の求心性入力を持つことでプルキンエ細胞の活動が同期するメカニズムは、遠方のプルキンエ細胞間の同期や感覚入力に対する同期などを説明可能であるが、本研究におけるギャップ結合は、特に背景活動における低頻度の顆粒細胞の活動に対してプルキンエ細胞間の同期を促す理由を説明可能となる。

#### 4.4.3 プルキンエ細胞の同期活動と運動制御

プルキンエ細胞間の活動は、小脳から脊髄下の運動ニューロンや大脳皮質などの他の脳領域のニューロンへの出力を担う深部小脳核ニューロンの活動を調整する役割を持つ [1]。特に、プルキンエ細胞間の同期的な活動は、深部小脳核ニューロンの規則的な活動や時間に精密な活動を誘発することが報告されている [55, 72]。また、このような下流のニューロンへの出力を介して、プルキンエ細胞の同期活動が四肢や体幹の運動の開始と調整を促進することが示唆されている [74]。本研究では、プルキンエ細胞の樹状突起の縮退が細胞間の同期活動を阻害し、深部小脳核ニューロンの活動を不規則的に変調させる可能性を示した。これにより、深部小脳核ニューロンは大脳皮質や脳幹からの運動指令や感覚入力の有無によらず、常に活動が抑制されることで、入力情報の判別を付けることができず、運動の開始時における活動などが阻害される。これは運動タイミングの制御といった時間に精密な調整機能を阻害するなど、脊髄小脳変性症 I 型における、プルキンエ細胞の神経変性と運動失調の因果関係を説明するうえで、重要な知見であることを主張する。

また、本研究では多様な神経疾患における研究が行われている瞬目反射条件づけのシミュレーションを行った [77, 78]。瞬目反射条件づけは詳細な神経メカニズムが解明されており、疾患モデルの機能評価として有用である。本研究でも正常なプルキンエ細胞モデルでは LTD による条件づけ学習に対応する応答が得られたが、縮退モデルではプルキンエ細胞間の同期現象の阻害によって適切な学習状態が再現されず、神経変性が運動学習の影響を阻害する可能性が示唆された。これは生理学実験では困難な操作である、ニューロン形状の縮退と運動

学習の関係性を考察するうえで重要な知見となる。

#### 4.4.4 今後の課題

本研究では、小脳皮質の最も主要なニューロンとして顆粒細胞とゴルジ細胞、プルキンエ細胞の3種類と小脳の最終的な出力を担う深部小脳核ニューロンからなるネットワークモデルを構築した。一方で、小脳皮質には、星状細胞とバスケット細胞からなる分子層介在ニューロンが存在し、それぞれプルキンエ細胞の樹状突起や細胞体といった異なる部位に抑制性のシナプス結合を与えている。また、分子層介在ニューロンはギャップ結合によってネットワークを構築していることも知られており、これらの活動をマルチコンパートメントモデルを用いてシミュレーションすることにより、プルキンエ細胞の空間形状が持つ役割について更なる知見を得ることが期待できる。さらにいえば、本研究で作成した小脳モデルを拡張し、大脳皮質や海馬などの他の脳部位のネットワークを含む全脳モデルや、身体モデルとの統合を行うことで具体的な運動タスクにおける挙動からの考察が可能となり、神経変性疾患における神経変性と脳機能障害の因果関係をさらに深く考察することが可能となる。よって、今後はマルチコンパートメントモデルによる全脳-身体モデルの作成と、このモデルのシミュレーションを実行可能なシミュレータの開発が必要となる。

### 4.5 結論

本研究では、マルチコンパートメントモデルを用いた小脳神経回路モデルを作成し、シミュレーションすることでプルキンエ細胞の空間形状が持つ機能的役割について考察した。プルキンエ細胞間の樹状突起にギャップ結合を仮定することで、細胞間の同期活動が促進され、深部小脳核ニューロンに規則的な活動が現れた。一方で、プルキンエ細胞の樹状突起の縮退は、ギャップ結合の損失により、細胞間の同期活動を阻害し、深部小脳核ニューロンの活動は不規則に変調された。このことから、脊髄小脳変性症 I 型などにみられるプルキンエ細胞の神経変性は、電気的な結合の消失による同期活動の阻害から、運動失調を引き起こす可能性が示唆される結果が得られた。一方で、本研究で行ったようなニューロン形状の変更とネットワークダイナミクスの記録は生理学実験では困難な操作である。よって、マルチコンパートメントモデルによる神経回路シミュレーションは、ニューロンの空間形状が持つ機能的役割について考察するために有用であり、今後の計算神経科学分野の発展による、マルチコン

パートメントモデルを用いた全脳シミュレーションの達成が神経変性疾患などの運動失調のメカニズム解明に大きく寄与することが期待される。

## 第 5 章

# 総括

本研究では、ニューロンの空間形状を考慮したマルチコンパートメントモデルの高效率な数値計算法について検討し、GPU を用いて性能の検証を行った。その後、この数値計算法を用いて、大規模な小脳神経回路モデルのシミュレーションを行い、小脳の情報処理におけるプルキンエ細胞の空間形状の重要性に関する知見を提供した。

第 2 章では、まず、生理学的背景とニューロンの数理モデルについて説明した。次に、ニューロンの空間形状を考慮したモデルとして、マルチコンパートメントモデルを説明し、実際のマルチコンパートメントモデルシミュレーションに用いられている数値計算法について説明した。また、マルチコンパートメントモデルは、空間形状を考慮しないニューロンモデルよりも多くの計算量が必要であることから、現在主流となっている並列計算用ハードウェアである GPU とそれを用いた並列シミュレーションについても述べた。

第 3 章では、マルチコンパートメントモデルの効率的な数値計算法として、拡散方程式用の陽解法である、RKC 法をマルチコンパートメントモデル用の数値計算法として採用した。RKC 法を GPU 上に実装した小脳ニューロンのマルチコンパートメントモデルに適用して、その性能を一般的な解法である CN 法と比較した。プルキンエ細胞のシミュレーションにおいて、連立方程式の求解によって低速な GPU のグローバルメモリへ多数のアクセスを必要とする陰解法 (CN 法) に対して、連立方程式の求解を必要としない RKC 法はメモリアクセス回数を低減させ、計算時間の高速化と細胞数の増加に対するスケーリング性能の向上を確認した。この傾向は、ゴルジ細胞と下オリーブ核ニューロンにも共通したが、顆粒細胞のシミュレーションにおいて、RKC 法は CN 法よりも計算時間が増加した。この原因として、顆粒細胞モデルが持つ 12 状態の反応方程式によって、モデルの硬さが増したことを挙げ、解決

法として 12 状態の反応方程式を 2 状態の HH 型方程式に書き直すことで硬さを回避することを試みた。結果として、HH 型方程式のみを持つ顆粒細胞モデルは RKC 法による計算時間の短縮に成功した。最後に、顆粒細胞、ゴルジ細胞、プルキンエ細胞からなる小脳神経回路モデルを構築し、RKC 法による計算時間の短縮を確認した。これらの結果から、陽解法による効率的な並列シミュレーションの実装や数値計算法に合わせたモデルの開発が、今後の計算神経科学分野の目標であるマルチコンパートメントモデルによる全脳シミュレーションを達成するうえで重要であることを主張した。

第 4 章では、第 3 章で採用した RKC 法を用いて、大規模な小脳神経回路モデルのシミュレーションを実行し、小脳の神経情報処理におけるプルキンエ細胞の空間形状が持つ機能的役割について考察した。特に、マルチコンパートメントモデルの特性を利用して、同一矢状面上のプルキンエ細胞間の樹状突起にギャップ結合を仮定した。結果として、同一矢状面上のプルキンエ細胞は同期的な活動を示し、プルキンエ細胞集団のリズミカルな抑制は、深部小脳核ニューロンに規則的な活動をもたらした。次に、プルキンエ細胞の樹状突起を縮退させたモデルを用い同様のシミュレーションを行ったところ、プルキンエ細胞の集団は非同期的な活動を示し、深部小脳核ニューロンに常に抑制入力を与え、不規則な活動をもたらした。これらの結果は、マウスを用いた生理学的研究からプルキンエ細胞の同期活動が運動制御に重要であるという報告ことや、脊髄小脳変性症 I 型におけるプルキンエ細胞の神経変性と運動失調の相関関係の報告などを踏まえ、プルキンエ細胞の空間形状は細胞間の同期活動に重要であり、樹状突起の縮退は深部小脳核ニューロンの変調を介して運動失調に影響を与える可能性を示唆した。

本研究における結果は、マルチコンパートメントモデルによる神経回路シミュレーションが、ニューロンの空間形状が持つ機能的役割について考察するために有用であること、そして大規模な神経回路モデルのシミュレーションを実行するためには数値計算法の改良に踏み込む必要があることを示唆する。特に、数値計算法の改良は神経科学分野の本質とは異なるため、これまでほとんど着目されてこず、1990 年代に開発されたシミュレータと同様の手法が今日まで用いられている。一方で、コンピュータの性能は 21 世紀に入り圧倒的に進化しており、スパコンや GPU といった最新のハードウェアはメモリウォール問題など固有の課題を抱えていることから、学際的な研究によって脳の本質の理解を試みるには、計算機科学や数値解析分野などの他分野における手法を積極的に取り入れ対応していく必要が存在する。本研究で実行した高効率な数値計算法を用いたマルチコンパートメントモデルによる小脳神経回路モデルシミュレーションは、生理学的実験では難しいニューロン形状の操作と全細胞

の同時記録を可能とし、プルキンエ細胞の空間形状が持つ機能的役割について細胞間の同期活動から考察することを可能とした。

本研究を発展させれば、次世代のスパコンを用いた、マルチコンパートメントモデルによるヒトスケール全脳シミュレーションの達成可能性は十分にあると考える。そのためには、「各脳部位モデルやシナプス結合マップの情報を統合し計算できる環境」、「ニューロンモデルに適切な数値計算法の更なる開発による計算時間の短縮と長時間シミュレーションの実現」、「身体モデルとの統合による具体的なタスクの実現」を準備する必要がある。このような準備をそろえ、実際にヒトスケール全脳シミュレーションが達成されることで、脳のミクロな形状がマクロな機能に対して影響を与えるか考察し、病理学的応用に寄与する基礎的知見を提供することが期待される。



# 謝辞

本研究をするにあたり、主任指導教員である電気通信大学大学院情報理工学研究科情報・ネットワーク工学専攻の山崎匡准教授には、終始適切なお助言、ご指導を賜りましたこと深く感謝を申し上げます。また、本論文の審査を引き受けてくださった、指導教員である龍野智哉准教授、山本野人教授、基盤理工学専攻の松田信爾准教授、他大学であるにも関わらず貴重なお助言とご教示をいただきました九州工業大学大学院生命体工学研究科の立野勝巳准教授に心より御礼申し上げます。研究遂行にあたり、様々なお助言、ご指摘をいただきました山崎研究室の皆様にも深く感謝いたします。

# 付録

## A.1 本研究で使用したマルチコンパートメントモデル

本節では、本研究で用いた小脳ニューロンのマルチコンパートメントモデルの詳細を記述する。なお本論を含め、全てのニューロンモデルの概要図は、空間形状のパラメータを使用し、オープンソースの統合型 3DCG 製作ソフトである Blender (2.8.0) を用いて再現している。

### A.1.1 プルキンエ細胞のマルチコンパートメントモデル

小脳のプルキンエ細胞のスタンダードなモデルとして、Eric De Schutter らが 1994 年に発表した非常に詳細な形態を持つマルチコンパートメントモデル [31, 32] が存在しており、GENESIS シミュレータ [30] 上で動作する。本研究では、このモデルのパラメータを基に CUDA で再実装し GPU 上で動作するようにした。モデルの模式図を図 A.1.1 に示す。

このモデルは、細胞体と樹状突起から構成されており、軸索は考慮されていない。1 コンパートメントの細胞体に対して、樹状突起は 1599 個のコンパートメントから成り、非常に複雑に分岐した樹状突起の空間形状を表現している。樹状突起は細胞体から伸び出ている main dendrite、そこから分岐する spiny dendrite、棘状に分岐した spiny dendrite の 3 種類から成り、spiny dendrite が最も多い。表 A.1.1 と A.1.2 にこのプルキンエ細胞モデルの各コンパートメントが持つイオンチャンネルと各チャンネルの最大コンダクタンスを示す。

NaF、NaP、CaP、CaT、KA チャンネルの式を表 A.1.3 に示す。これらのチャンネルは式 (2.4) から式 (2.6) に従う。残りのチャンネルは  $\alpha$  と  $\beta$  を用いずにゲート変数の時定数などがフィッティングされている (表 A.1.4)。

表 A.1.1: プルキンエ細胞モデルが持つイオンチャンネル

コンパートメント	コンパートメント数	イオンチャンネル
細胞体	1	Leak, NaF, NaP, CaT, Kh1, Kh2, Kdr, KM, KA
main dendrite	9	Leak, CaP, CaT, Kdr, KM, KA, KC, K2
smooth dendrite	105	Leak, CaP, CaT, KM, KC, K2
spiny dendrite	1485	Leak, CaP, CaT, KM, KC, K2

表 A.1.2: 各コンパートメントが持つイオンチャンネルの最大コンダクタンス [mS/cm<sup>2</sup>]

チャンネル名	反転電位 [mV]	細胞体	main dendrite	smooth と spiny dendrite
Leak	-55.0	0.1	0.033	0.033
NaF	45.0	7500.0	0.0	0.0
NaP	45.0	1.0	0.0	0.0
CaP	135.0	0.0	4.5	4.5
CaT	135.0	0.5	0.5	0.5
Kh	-30.0	0.3	0.0	0.0
Kdr	-85.0	600.0	60.0	0.0
KM	-85.0	0.040	0.010	0.013
KA	-85.0	15.0	2.0	0.0
KC	-85.0	0.0	80.0	80.0
K2	-85.0	0.0	0.39	0.39

表 A.1.3: プルキンエ細胞モデルのゲート変数のダイナミクス

チャンネル名	ゲート変数	乗数	$\alpha$ [1/ms]	$\beta$ [1/ms]
NaF	m	3	$35.0/\exp(-(v+5.0)/10.0)$	$7.0/\exp((v+65.0)/20.0)$
	n	1	$0.226/(1+\exp((v+80.0)/10.0))$	$7.5/\exp(-(v-3.0)/18.0)$
NaP	m	3	$200.0/(1+\exp(-(v-18.0)/16.0))$	$25.0/(1+\exp((v+58.0)/8.0))$
	m	1	$8.5/(1+\exp(-(v-8.0)/12.5))$	$35.0/(1+\exp((v+74.0)/14.5))$
CaT	n	1	$0.0015/(1+\exp((v+29.0)/8.0))$	$0.0055/(1+\exp(-(v+23.0)/8.0))$
	m	1	$2.60/(1+\exp(-(v+21.0)/8.0))$	$0.18/(1+\exp((v+40.0)/4.0))$
KA	n	1	$0.0025/(1+\exp((v+40.0)/8.0))$	$0.19/(1+\exp(-(v+50.0)/10.0))$
	m	4	$1.40/(1+\exp(-(v+27.0)/12.0))$	$0.49/(1+\exp((v+30.0)/4.0))$
	n	1	$0.0175/(1+\exp((v+50.0)/8.0))$	$1.30/(1+\exp(-(v+13.0)/10.0))$

表 A.1.4: 残りのゲート変数のダイナミクス

チャンネル名	ゲート変数	乗数	$\tau_x(v)$ [ms]	$x_0(v)$
Kh1	m	1	7.6	$0.8/(1+\exp((v+82.0)/7.0))$
	m	1	36.8	$0.2/(1+\exp((v+82.0)/7.0))$
Kdr	m	1	$\begin{cases} 1200.0 & (v < -25.0) \\ 10.0 & (\text{otherwise}) \end{cases}$	$\frac{\alpha_{\text{Kdr}}}{\alpha_{\text{Kdr}}+5.0\exp(-(v+127.0)/30.0)}$
			$\alpha_{\text{Kdr}} = \begin{cases} \frac{0.0235 \times (v-8.0)}{(\exp(-(v-8.0)/12.0)-1.0)} & ( v-8.0  > 10^{-6}) \\ 0.282 & (\text{otherwise}) \end{cases}$	
KM	m	1	$\frac{60.61}{\exp((v+35.0)/20.0)+\exp(-(v+35.0)/20.0)}$	$1.0/(1+\exp((v+25.0)/4.0))$
	m	1	$\frac{7.5+0.110/\exp((v-35.0)/14.9)}{1}$	$\frac{7.5}{7.5+0.110/\exp((v-35.0)/14.9)}$
K2	n	2	10.0	$1.0/(1.0+4.00/[\text{Ca}])$
	m	1	$\frac{1}{25.0+0.075/\exp((v+25.0)/6.0)}$	$\frac{25.0}{25.0+0.075/\exp((v+25.0)/6.0)}$
	n	2	10.0	$1.0/(1.0+0.20/[\text{Ca}])$

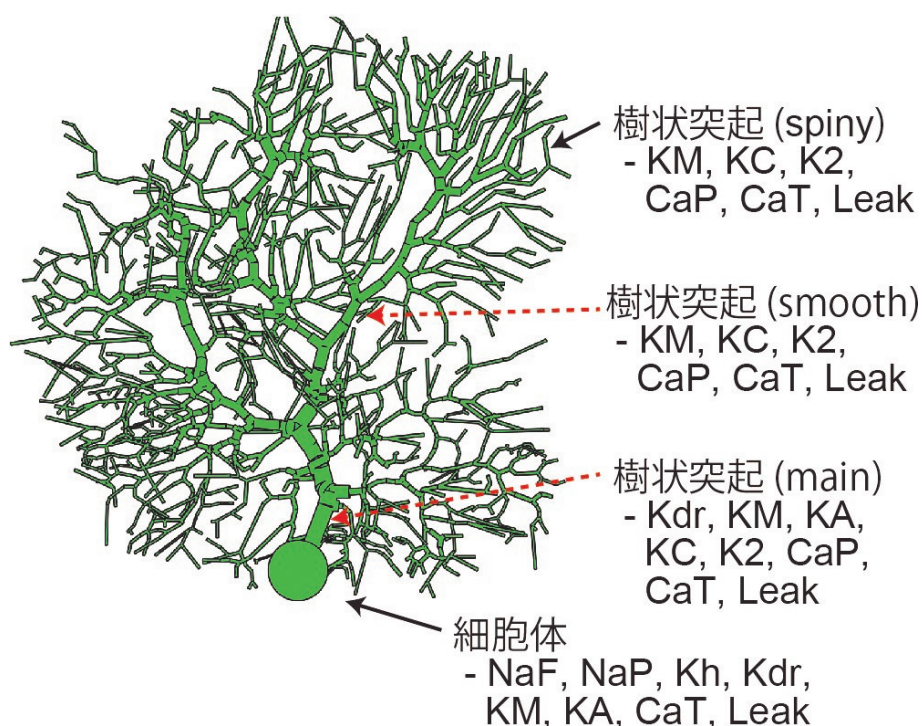


図 A.1.1: 小脳プルキンエ細胞のマルチコンパートメントモデル。De Schutter らのモデル [31, 32] のパラメータを基に CUDA を用いて再実装した。spiny (1485 個)、smooth (105 個)、main (9 個) と呼ばれる 3 種類の樹状突起と細胞体 (1 個) の合計 1600 コンパートメントからなる。10 個のイオンチャンネルと 1 個のリークチャンネル、13 個のゲート変数を持つ。

## A.1.2 顆粒細胞のマルチコンパートメントモデル

小脳顆粒細胞のモデルとして、Katarzyna Dover らが 2016 年に発表した非常に詳細なダイナミクスを持つマルチコンパートメントモデル [49] が存在しており、NEURON シミュレータ [24] 上で動作する。本研究では、このモデルのパラメータを基に CUDA で再実装し GPU 上で動作するようにした。モデルの模式図を図 A.1.2 に示す。

このモデルは樹状突起と細胞体、軸索、そして軸索が分岐した平行線維から構成されている。特に、スパイクを生成する軸索小丘と軸索初節は精緻にモデル化されており、生成されたスパイクは上行性軸索と並行線維でその形状を維持しながら伝搬されるようなダイナミクスを持つ。

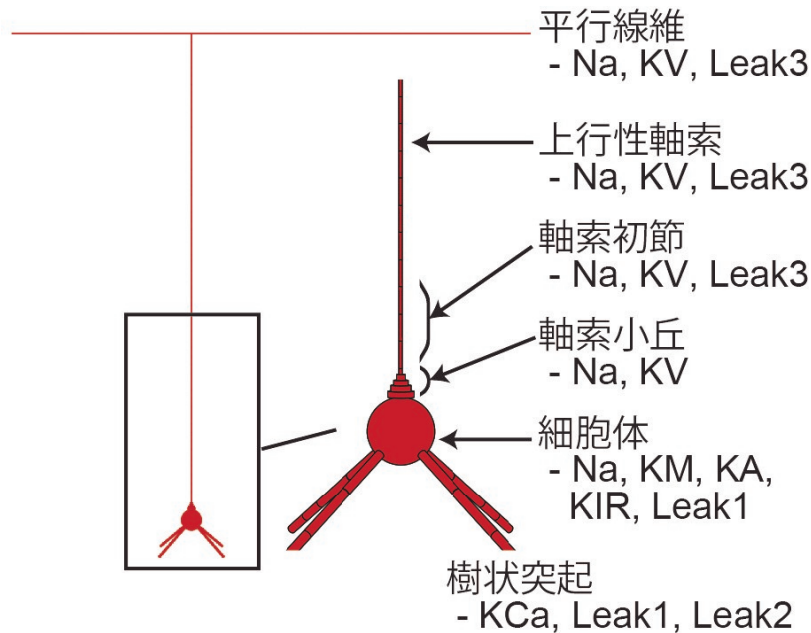


図 A.1.2: 小脳顆粒細胞のマルチコンパートメントモデル。Dover らのモデル [49] のパラメータを基に CUDA を用いて再実装した。4 本の樹状突起 (コンパートメント数 4 個)、細胞体 (1 個)、軸索小丘 (5 個)、軸索初節 (3 個)、上行性軸索 (53 個)、平行線維 (500 個) の合計 578 個のコンパートメントからなる。7 個のイオンチャンネルと 3 個のリークチャンネル、反応方程式で記述される  $\text{Na}^+$  チャンネルを除き、8 個のゲート変数を持つ。

表 A.1.5 と A.1.6 にこのモデルの各コンパートメントが持つイオンチャンネルと各チャンネルの最大コンダクタンスを示す。また、 $\text{Na}^+$  チャンネルを除いた各イオンチャンネルの式を表 A.1.7 に示す。

本モデルの  $\text{Na}^+$  チャンネルは 12 状態の反応方程式を解くことにより、ゲート変数  $O$  の値が得られる。 $\text{Na}^+$  チャンネルのスキームを図 A.1.3 に示す。また、 $\text{Na}^+$  チャンネルの各パラメータは Fibroblast growth factor homologous factor (FHF) たんぱく質の有無によって変化する。

表 A.1.5: 顆粒細胞モデルが持つイオンチャンネル

コンパートメント	コンパートメント数	イオンチャンネル
樹状突起	4	Leak1, Leak2, Kca
細胞体	1	Leak1, Na, KM, KA, KIR
軸索小丘	5	Na, KV
軸索初節	3	Leak3, Na, KV
上行性軸索	53	Leak3, Na, KV
平行線維	500	Leak3, Na, KV

表 A.1.6: 各コンパートメントが持つイオンチャンネルの最大コンダクタンス [ $\text{mS}/\text{cm}^2$ ]. 樹状突起の最尖端と 2 番目のコンパートメントが持つコンダクタンスは倍の値を持つ。

チャンネル名	反転電位 [mV]	細胞体	樹状突起	軸索小丘	軸索初節	上行性軸索	平行線維
Leak1	-16.5	0.9169	0.0225	-	-	-	-
Leak2	-65.0	-	0.0476	-	-	-	-
Leak3	-74.5	-	-	-	1.0	0.05	0.05
Na	87.39	-	-	500.0	800.0	900.0	300.0
KV	-84.69	-	-	30000.0	10000.0	1000.0	30.0
KA	-84.69	40.0	-	-	-	-	-
KIR	-84.69	2.0	-	-	-	-	-
KCa	-84.69	-	14.0	-	-	-	-
KM	-84.69	6.0	-	-	-	-	-

る [49]。このスキームは次式のように表すことができる。

$$\left\{ \begin{array}{l} \frac{dC1}{dt} = -(n_1\alpha + C_{\text{on}}) \cdot C1 + n_4\beta \cdot C2 + C_{\text{off}} \cdot I1 \\ \frac{dC2}{dt} = -(n_2\alpha + C_{\text{on}}a) \cdot C2 + n_3\beta \cdot C3 + C_{\text{off}}b \cdot I2 \\ \frac{dC3}{dt} = -(n_3\alpha + C_{\text{on}}a^2) \cdot C3 + n_2\beta \cdot C4 + C_{\text{off}}b^2 \cdot I3 \\ \frac{dC4}{dt} = -(n_4\alpha + C_{\text{on}}a^3) \cdot C4 + n_1\beta \cdot C5 + C_{\text{off}}b^3 \cdot I4 \\ \frac{dC5}{dt} = -(259.8 + C_{\text{on}}a^4) \cdot C5 + 69.3 \cdot O + C_{\text{off}}b^4 \cdot I5 \\ \frac{dI1}{dt} = -(n_1\alpha a + C_{\text{off}}) \cdot I1 + n_4\beta b \cdot I2 + C_{\text{on}} \cdot C1 \\ \frac{dI2}{dt} = -(n_2\alpha a + C_{\text{off}}b) \cdot I1 + n_3\beta b \cdot I3 + C_{\text{on}}a \cdot C2 \\ \frac{dI3}{dt} = -(n_3\alpha a + C_{\text{off}}b^2) \cdot I1 + n_2\beta b \cdot I4 + C_{\text{on}}a^2 \cdot C3 \\ \frac{dI4}{dt} = -(n_4\alpha a + C_{\text{off}}b^3) \cdot I1 + n_1\beta b \cdot I5 + C_{\text{on}}a^3 \cdot C4 \\ \frac{dI5}{dt} = -(259.8 + C_{\text{off}}b^4) \cdot I1 + 69.3 \cdot I6 + C_{\text{on}}a^4 \cdot C5 \\ \frac{dO}{dt} = -(69.3 + O_{\text{on}}) \cdot O + 259.8 \cdot C5 + O_{\text{off}} \cdot I6 \end{array} \right.$$

表 A.1.7: 顆粒細胞モデルのゲート変数のダイナミクス

チャネル名	ゲート変数	乗数	$\alpha$ [1/ms]	$\beta$ [1/ms]
KV	m	4	$\begin{cases} -0.17 \times \left(1 + \frac{v+25.0}{20.0}\right) & ( v + 25.0  < 10^{-6}) \\ \frac{-0.017 \times v + 25.0}{\exp(-(v+25.0)/10.0) - 1.0} & (\text{otherwise}) \end{cases}$	$0.2223 \times \exp\left(\frac{-(v+35.0)}{80.0}\right)$
		3	$\frac{0.071}{1 + \exp(-(v+9.17203)/23.32708)}$	$0.1567 \times \exp\left(\frac{v+18.27914}{19.47175}\right)$ $x_0 = \frac{1.0}{1.0 + \exp\left(\frac{-(v+38.0)}{17.0}\right)}$
KIR	m	1	$0.0348 / \left(1 + \exp\left(\frac{v-111.33208}{12.8433}\right)\right)$	$0.0327 / \left(1 + \exp\left(\frac{-(v+49.9537)}{8.90123}\right)\right)$
		1	$0.2302 \times \exp\left(\frac{-(v+83.94)/24.3902}{1.4433}\right)$	$x_0 = 1.0 / \left(1.0 + \exp\left(\frac{v+78.8}{8.4}\right)\right)$ $0.2943 \times \exp\left(\frac{(v+83.94)/35.714}{0.8660}\right)$
KCa	m	1	$\frac{1.0}{1 + (1.5 \times 10^{-3} \times \exp(-v/11.765)) / [\text{Ca}]}$	$\frac{[\text{Ca}]}{0.8660}$
		1	$0.0046 \times \exp((v+30.0)/40.0)$	$0.0046 \times \exp\left(\frac{-v/11.765}{1.0}\right)$ $x_0 = \frac{1.0}{1.0 + \exp(-(v+35.0)/6.0)}$



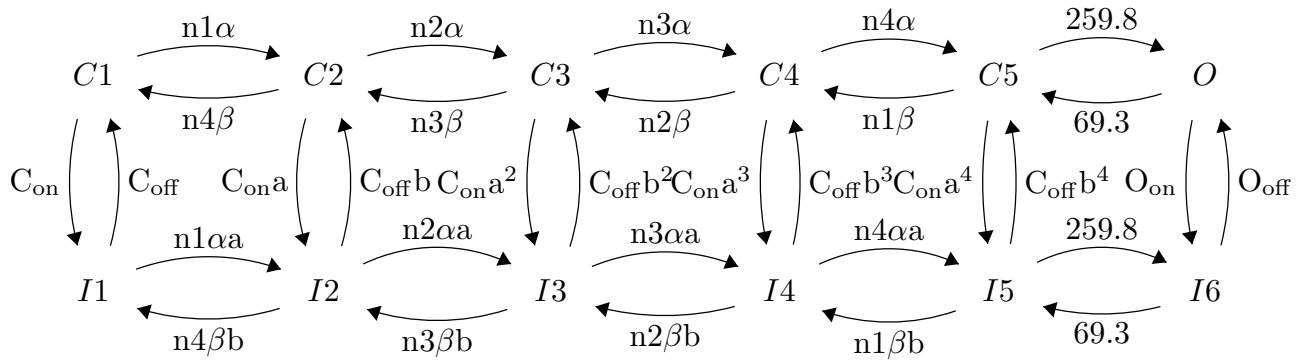


図 A.1.3: 12 状態の反応方程式からなる  $\text{Na}^+$  チャネル。このモデルでは、 $O$  がゲート変数にあたる。

また、各変数は次式に従う。

$$\alpha = A_\alpha \exp\left(\frac{v(t, x) - V_{\text{shift}}}{10.0}\right), \beta = A_\beta \exp\left(\frac{-v(t, x) + V_{\text{shift}}}{10.0}\right)$$

$$a = \left(\frac{O_{\text{on}}}{C_{\text{on}}}\right)^{1/4}, b = \left(\frac{O_{\text{off}}}{C_{\text{off}}}\right)^{1/4}$$

### A.1.3 ゴルジ細胞のマルチコンパートメントモデル

ゴルジ細胞のモデルとして、Sergio Solinas らが 2007 年に発表した詳細なダイナミクスを持つマルチコンパートメントモデル [50] が存在しており、NEURON シミュレータ上で動作する。本研究では、このモデルのパラメータを基に CUDA で再実装し GPU 上で動作するようにした。モデルの模式図を図 A.1.4 に示す。

このモデルは先端樹状突起と基底樹状突起と呼ばれる 2 種類の樹状突起、細胞体、軸索から構成されている。また、イオンチャネルは細胞体に集中しており、その他の部位には Leak チャネルのみが存在している。

表 A.1.8 と A.1.9 にこのモデルの各コンパートメントが持つイオンチャネルと各チャネルの最大コンダクタンスを示す。また、KAHP チャネルを除いた各イオンチャネルの式を表 A.1.10 に示す。本モデルの KAHP チャネルは 6 状態の反応方程式を解くことにより、ゲート変数  $O1$  と  $O2$  の値が得られる。KAHP チャネルのスキームを図 A.1.5 に示す。このス

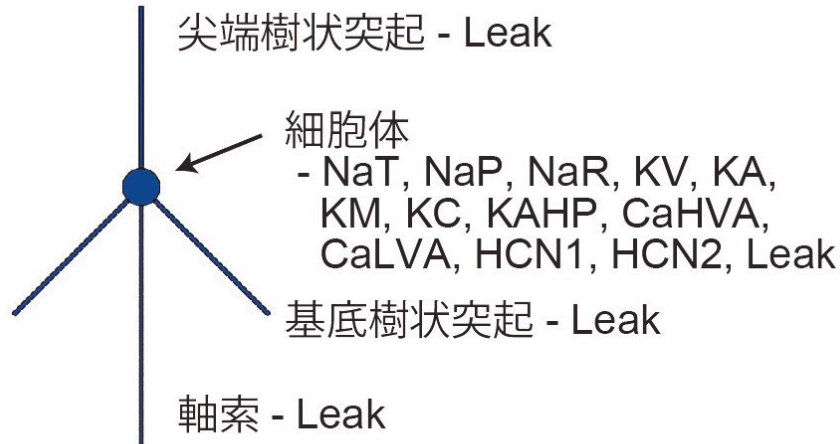


図 A.1.4: ゴルジ細胞のマルチコンパートメントモデル。Solinas らのモデル [50] のパラメータを基に CUDA を用いて再実装した。3 本の樹状突起 (コンパートメント数 10 個)、細胞体 (1 個)、軸索 (100 個) の合計 131 個のコンパートメントからなる。12 個のイオンチャンネルと 1 個のリークチャンネル、反応方程式で記述される KAHP チャンネルを除き、16 個のゲート変数を持つ。

表 A.1.8: ゴルジ細胞モデルが持つイオンチャンネル

コンパートメント	コンパートメント数	イオンチャンネル
樹状突起	10	Leak
細胞体	1	Leak, NaT, NaR, NaP, KV, KM, KA, KC, KAHP, CaHVA, CaLVA, HCN1, HCN2
軸索	100	Leak

キームは次式のように表すことができる。

$$\left\{ \begin{array}{l} \frac{dC1}{dt} = -d_1 \cdot C1 + i_1 \cdot C2 \\ \frac{dC2}{dt} = -(d_2 + i_1) \cdot C2 + d_1 \cdot C1 + i_2 \cdot C3 \\ \frac{dC3}{dt} = -(d_3 + i_2 + b_1) \cdot C3 + d_2 \cdot C2 + i_3 \cdot C4 + a_1 \cdot O1 \\ \frac{dC4}{dt} = -(i_3 + b_2) \cdot C4 + d_3 \cdot C3 + a_2 \cdot O2 \\ \frac{dO1}{dt} = -a_1 \cdot O1 + b_1 \cdot C3 \\ \frac{dO2}{dt} = -a_2 \cdot O2 + b_2 \cdot C4 \end{array} \right.$$

表 A.1.9: 各コンパートメントが持つイオンチャネルの最大コンダクタンス [mS/cm<sup>2</sup>]

チャネル名	反転電位 [mV]	樹状突起	細胞体	軸索
Leak	-55.0	0.021	0.021	0.021
NaT	87.39	0.0	48.0	0.0
NaR	87.39	0.0	1.7	0.0
NaP	87.39	0.0	0.19	0.0
KV	-84.69	0.0	32.0	0.0
KA	-84.69	0.0	8.0	0.0
KC	-84.69	0.0	3.0	0.0
KM	-84.69	0.0	1.0	0.0
CaHVA	120.0	0.0	0.46	0.0
CaLVA	120.0	0.0	0.25	0.0
HCN1	-20.0	0.0	0.05	0.0
HCN2	-20.0	0.0	0.08	0.0
KAHP	-84.69	0.0	38.0	0.0

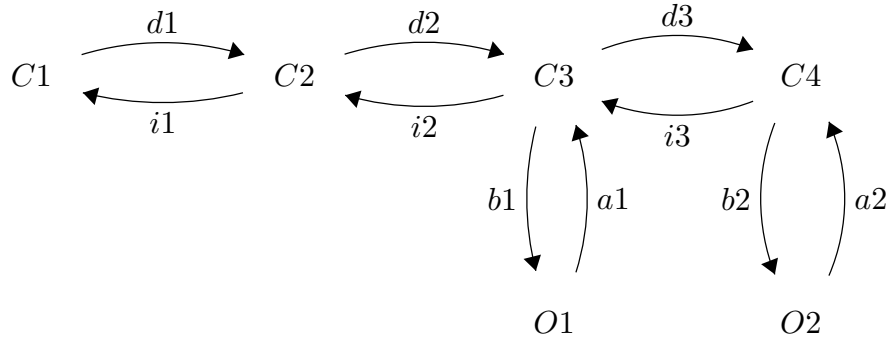


図 A.1.5: 6 状態の反応方程式からなる KAHP チャネル。このモデルでは、 $O_1$  と  $O_2$  の和がゲート変数にあたる。

また、各変数とパラメータは次式に従う。

$$\begin{aligned}
 d_1 &= 200 \times \frac{[\text{Ca}]}{3.0}, d_2 = 160 \times \frac{[\text{Ca}]}{3.0}, d_3 = 80 \times \frac{[\text{Ca}]}{3.0} \\
 i_1 &= 80 \times 10^{-3}, i_2 = 80 \times 10^{-3}, i_3 = 200 \times 10^{-3} \\
 a_1 &= 1.0, a_2 = 100 \times 10^{-3}, b_1 = 160 \times 10^{-3}, b_2 = 1.2
 \end{aligned}$$

表 A.1.10: ゴルジ細胞モデルのゲート変数のダイナミクス

チャンネル名	ゲート変数	乗数	$\alpha$ [1/ms]	$\beta$ [1/ms]
NaT	m	3	$0.417(v + 35)/(1 - \exp(-(v + 35)/10))$	$16.68 \exp(-0.055(v + 60))$
	n	1	$0.292 \exp(-0.3(v + 60))$	$4.17/(1 + \exp(-(v + 27)/5))$
NaR	m	1	$\left\{ \frac{1.11 - 68.5(v + 6.48)}{\exp(-(v + 6.48)/6.8) - 1} \right\} \times 10^{-4}$	$\left\{ \frac{\exp((v + 54)/0.11) - 1}{66 + 21.7(v + 54)} \right\} \times 10^{-3}$
	n	1	$0.443 \exp(-(v + 90)/62.5)$	$0.014 \exp((v + 93.3)/16.1)$
NaP	m	1	$0.421(v + 50)/[1 - \exp(-(v + 50)/5)]$	$-0.287(v + 50)/[1 - \exp((v + 50)/5)]$
			$x_0 = 1/(1 + \exp(-v + 53)/5)$	$\tau_x = 5/(\alpha + \beta)$
KV	m	4	$0.062(v + 36)/[1 - \exp(-(v + 36)/10)]$	$0.78 \exp(-(v + 46)/80)$
KA	m	3	$0.62/[1 + \exp(-(v + 19.17)/23.32)]$	$0.126[\exp(-(v + 28.28)/19.47)]$
	n	1	$x_0 = 1/(1 + \exp((v + 88.8)/8.4))$	
KC	n	1	$0.028/[1 + \exp((v + 121)/12.84)]$	$0.026/[1 + \exp(-(v + 59.95)/8.9)]$
			$x_0 = 1/(1 + \exp(-(v + 48)/17))$	
KM	m	1	$\frac{1 + 0.0015 \exp(-(v + 10)/11.7)/[Ca]}{1 + 0.0015 \exp(-(v + 10)/11.7)}$	$\frac{1 + [Ca]/(1.5 \times 10^{-4} \exp(-(v + 10)/11.7))}{0.0037 \exp(-(v + 40)/20)}$
	m	1	$0.0037 \exp((v + 40)/40)$	
CaHVA	m	2	$x_0 = 1/[1 + \exp(-(v + 45)/6)]$	$0.115 \exp(-0.039(v + 28.66))$
	n	1	$0.0687 \exp(0.063(v + 39))$	$1.8 \times 10^{-3} \exp((v + 58)/83)$
CaLVA	m	2	$1.8 \times 10^{-3} \exp(-(v + 58)/18)$	$3 + \frac{\exp(-(v + 37)/10) + \exp(-(v + 112)/15)}{0.85}$
			$x_0 = 1/(1 + \exp(-(v + 62)/7.4))$	$\tau_x = \frac{85 + \frac{\exp(-(v + 58)/4) + \exp(-(v + 417)/50)}{0.9}}{1}$
HCN1	n	1	$x_0 = 1/(1 + \exp(-(v + 90)/5))$	$\tau_x = \frac{0.9}{\exp(-(v + 58)/4) + \exp(-(v + 417)/50)}$
	m	1	$h_{fast} = r/[1 + \exp((v + 82.5) \times 0.11)]$	$\tau_{fast} = \exp((0.014v + 13.37)/2.3)$
HCN2	n	1	$h_{slow} = [1 - r]/[1 + \exp((v + 82.5) \times 0.11)]$	$\tau_{slow} = \exp((0.014v + 14.07)/2.3)$
			$r = 0.0021(v + 10) + 0.97$	
HCN2	m	1	$h_{fast} = r/[1 + \exp((v + 91.9) \times 0.16)]$	$\tau_{fast} = \exp((0.027v + 15.6)/2.3)$
	n	1	$h_{slow} = [1 - r]/[1 + \exp((v + 91.9) \times 0.16)]$	$\tau_{slow} = \exp((0.015v + 15.3)/2.3)$
			$r = -0.0227(v + 10) - 1.47$	

#### A.1.4 下オリブ核ニューロンのマルチコンパートメントモデル

下オリブ核ニューロンのモデルとして、Jornt R. De Gruijl らが 2012 年に発表したマルチコンパートメントモデル [51] が存在しており、Matlab 上で動作する。本研究では、このモデルのパラメータを基に CUDA で再実装し GPU 上で動作するようにした。モデルの模式図を図 A.1.6 に示す。



図 A.1.6: 下オリブ核ニューロンのマルチコンパートメントモデル。De Gruijl らのモデル [51] のパラメータを基に CUDA を用いて再実装した。樹状突起、細胞体、軸索がそれぞれ 1 個のコンパートメントからなる。7 個のイオンチャンネルと 1 個のリークチャンネル、10 個のゲート変数を持つ。

このモデルは、それぞれ 1 個のコンパートメントからなる細胞体と樹状突起、軸索から構成されている。本来、下オリブ核ニューロンは複雑な形状を持つことで知られている [81] が、下オリブ核ニューロン特有の形状まで考慮したマルチコンパートメントモデルは現状存在しないため、3 コンパートメントからなる本モデルを採用した。表 A.1.11 と A.1.12 にこのモデルの各コンパートメントが持つイオンチャンネルと各チャンネルの最大コンダクタンスを示す。

各イオンチャンネルの式を表 A.1.13 に示す。これらのチャンネルは式 (2.4) から式 (2.6) に従う。

表 A.1.11: 下オリブ核ニューロンモデルが持つイオンチャンネル

コンパートメント	コンパートメント数	イオンチャンネル
樹状突起	1	Leak, KCa, CaH, H
細胞体	1	Leak, Na, K, Kdr, CaL
軸索	1	Leak, Na, K

表 A.1.12: 各コンパートメントが持つイオンチャンネルの最大コンダクタンス [mS/cm<sup>2</sup>]

チャンネル名	反転電位 [mV]	樹状突起	細胞体	軸索
Leak	10.0	0.016	0.016	0.016
Na	-75.0	0.0	120.0	240.0
K	-75.0	0.0	5.0	20.0
Kdr	-75.0	0.0	9.0	0.0
KCa	-75.0	35.0	0.0	0.0
CaH	120.0	4.5	0.0	0.0
CaL	120.0	0.0	0.83	0.0
H	-43.0	0.15	0.0	0.0

## A.2 RKC 法のアルゴリズムと安定性

本節では、本研究において CN 法（陰解法）の代替手法として提案した RKC 法のアルゴリズムとその安定性について記述する。

### A.2.1 RKC 法のアルゴリズム

RKC 法は放物型の偏微分方程式を陽的に解くために開発された、動的な時間刻み幅  $\Delta t$  とステージ数  $s$  を持つ Runge-Kutta 法である [44–48]。本節では 2 次の RKC 法のアルゴリズムについて簡単に記述する。

表 A.1.13: 下オリーブ核ニューロンモデルのゲート変数のダイナミクス

チャンネル名	ゲート変数	乗数	$\alpha$ [1/ms]	$\beta$ [1/ms]
Na	m	3	$x_0 = \frac{1}{1+\exp(-(v+30)/5.5)}$	—
	n	1	$x_0 = \frac{1}{1+\exp(-(v+60)/-5.8)}$	$\tau_x = 1.5 \exp(-(v+40)/33)$
K	m	4	$\frac{0.13v+3.25}{1-\exp(-(v+25)/10)}$	$1.69 \exp(-(0.0125v+0.4375))$
	Kdr	1	$x_0 = \frac{1}{1+\exp(-(v+3)/10)}$	$\tau_x = 47 \exp(-(v+50)/900) + 5$
KCa	n	1	$x_0 = \frac{1}{1+\exp(-(v+51)/12)}$	$\tau_x = 47 \exp(-(v+50)/900) + 5$
	m	1	$\min(0.00002[\text{Ca}^{2+}], 0.01)$	0.015
CaH	m	2	$\frac{1.7}{1+\exp(-(v+5)/13.9)}$	$\frac{0.02v+0.17}{\exp(-(v+8.5)/5)-1}$
CaL	m	3	$x_0 = \frac{1}{1+\exp(-(v+61)/4.2)}$	$\tau_x = 1$
	n	1	$x_0 = \frac{1}{1+\exp((v+85.5)/8.5)}$	$\tau_x = \frac{20 \exp((v+160.0)/30.0)}{1.0+\exp((v+84.0)/7.3)} + 35.0$
H	m	1	$x_0 = \frac{1}{1+\exp((v+80.0)/4.0)}$	$\tau_x = \frac{1}{\exp(-(0.086v+14.6))} + \exp(0.07v - 1.87)$

まず、解くべき偏微分方程式を線の方法により常微分方程式（式 (2.8)）のシステムに分割し、次式のように表す。

$$\frac{dU(t)}{dt} = F(t, U(t)), \quad 0 < t \leq T, \quad U(0) = U_0 \quad (\text{A.1})$$

また、そのヤコビ行列を  $F'(t, U) = \partial F(t, U)/\partial U$  と表す。次に、変数  $U(t)$  を時間  $t_n$  から  $t_{n+1}(= t_n + \tau)$  に更新することを考える。

$$\begin{aligned} Y_0 &= U_n, \\ Y_1 &= Y_0 + \tilde{\mu}_1 \tau F_0, \\ Y_j &= (1 - \mu_j - v_j) Y_0 + \mu_j Y_{j-1} + v_j Y_{j-2} + \tilde{\mu}_j \tau F_{j-1} + \tilde{\gamma}_j \tau F_0, \quad j = 2, \dots, s, \\ U_{n+1} &= Y_s \end{aligned} \quad (\text{A.2})$$

ただし、 $U_n = U(t_n)$  であり、ステージ数  $s$  は 2 以上の値をとる。各パラメータは以下の通りに与えられる。

$$\begin{aligned} \varepsilon &= \frac{2}{13}, \quad w_0 = 1 + \varepsilon/s^2, \quad w_1 = \frac{T'_s(w_0)}{T''_s(w_0)}, \\ b_j &= \frac{T''_j(w_0)}{(T'_j(w_0))^2} \quad (2 \leq j \leq s), \quad b_0 = b_2, \quad b_1 = b_2 \\ \tilde{\mu}_1 &= b_1 w_1, \quad \mu_j = \frac{2b_j w_0}{b_{j-1}}, \quad v_j = \frac{-b_j}{b_{j-2}}, \\ \tilde{\mu}_j &= \frac{2b_j w_1}{b_{j-1}}, \quad \tilde{\gamma}_j = -(1 - b_{j-1} T_{j-1}(w_0)) \tilde{\mu}_j \quad (2 \leq j \leq s) \end{aligned} \quad (\text{A.3})$$

ここで、 $T_s(x)$  は第一種 Chebyshev 多項式であり、 $x = \cos(t)$  とすると、 $T_s(x) = \cos(st)$  と表される。即ち、

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ &\vdots \end{aligned} \quad (\text{A.4})$$

であり、多項式は漸化式  $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$  に従う。また、式 (A.2) において、



$F_j = F(t_n + c_j\tau, Y_j)$  であり、 $c_j$  は次式で表される。

$$c_j = \frac{T'_s(w_0) T''_j(w_0)}{T''_s(w_0) T'_j(w_0)} \approx \frac{j^2 - 1}{s^2 - 1} \quad (2 \leq j \leq s - 1), \quad c_1 = \frac{c_2}{T'_2(w_0)} \approx \frac{c_2}{4}, \quad c_s = 1 \quad (\text{A.5})$$

次に、新たな時間刻み幅  $\tau_{\text{new}}$  は現在の時間刻み  $\tau$  を用いて次式のように決定する。

$$\begin{aligned} \tau_{\text{new}} &= \min(10, \max(0.1, \text{fac}))\tau \\ \text{fac} &= 0.8 \left( \frac{\|\text{Est}_n\|^{1/(p+1)} \tau_n}{\|\text{Est}_{n+i}\|^{1/(p+1)} \tau_{n-1}} \right) \frac{1}{\|\text{Est}_{n+1}\|^{1/(p+1)}} \end{aligned} \quad (\text{A.6})$$

ここで、 $\text{Est}_{n+1}$  は局所誤差の推定値であり、次式のように表される。

$$\text{Est}_{n+1} = \frac{1}{15} [12(U_n - U_{n+1}) + 6\tau(F(U_n) + F(U_{n+1}))] \quad (\text{A.7})$$

また、ステージ数  $s$  は時間刻み  $\tau$  とヤコビ行列のスペクトル半径  $\sigma(F'(t, U))$  を用いて、次式において最小の整数に設定される。

$$\tau\sigma(F'(t, U)) \leq 0.653s^2 \quad (\text{A.8})$$

最後に、更新した時間刻み幅とステージ数が誤差を十分に抑えているか、パラメータとして与えられる誤差の許容値  $\text{atol}, \text{rtol}$  を用いて判定する。

$$\begin{aligned} \|\text{Est}_{n+1}\| &= \text{RMS}(w^{-1}\text{Est}_{n+1}) \\ w &= \sqrt{m} \text{diag}(\text{Tol}_1, \dots, \text{Tol}_m) \\ \text{Tol}_k &= \text{atol}_k + \text{rtol} |U_{n+1,k}| \end{aligned} \quad (\text{A.9})$$

ここで、 $m$  は常微分方程式のシステムの次元（本研究ではコンパートメントによる空間的な分割数）であり、 $U_{n+1,k}$  は  $k$  番目の常微分方程式の  $U_{n+1}$  の値、 $\text{RMS}(x)$  は  $x$  の二乗平均平方根である。最終的に、 $\|\text{Est}_{n+1}\| \leq 1$  であれば、この時間刻みとステージ数は許容され、誤差の値が 1 を超えた場合、次式に従って更新された時間刻み幅を用いてステージ数の推定を再度行う。

$$\tau_{\text{new}} = 0.8\tau / \|\text{Est}_{n+1}\|^{1/3} \quad (\text{A.10})$$

## A.2.2 RKC 法の安定領域

数値計算法の安定性を解析する方法として、Dahlquist のテスト方程式 (式 (A.11)) に対する 1 ステップ後の近似値にあたる、安定性関数  $R(z)$  を求める手法が挙げられる [82]。

$$y' = \lambda y, \quad y_0 = 1, \quad z = h\lambda \quad (\text{A.11})$$

このとき、集合

$$S = \{z \in \mathbb{C}; |R(z)| \leq 1\} \quad (\text{A.12})$$

は、数値計算法の安定領域と呼ばれる。

例えば、2 段 2 次の陽的ルンゲクッタ法は時間刻み幅を  $h$  とすると、

$$\begin{aligned} k_1 &= f(x_m, y_m) \\ k_2 &= f(x_m + h, y_m + hk_1) \\ y_{m+1} &= y_m + h \frac{k_1 + k_2}{2} \end{aligned} \quad (\text{A.13})$$

であり、これをテスト方程式に作用させると、

$$\begin{aligned} y_{m+1} &= y_m + (h\lambda)y_m + \frac{(h\lambda)^2}{2}y_m \\ y_{m+1} &= \left(1 + h\lambda + \frac{(h\lambda)^2}{2}\right)y_m \\ y_{m+1} &= \left(1 + z + \frac{z^2}{2}\right)y_m \end{aligned} \quad (\text{A.14})$$

となり、安定性関数は

$$R_2(z) = 1 + z + \frac{z^2}{2} \quad (\text{A.15})$$

となる。

一方で、RKC 法の安定化関数は次式のように与えられる [82]。

$$R_s(z) = a_s + b_s T_s(w_0 + w_1 z) \quad (\text{A.16})$$

RKC 法の安定領域はステージ数  $s$  によって可変であり、負の実軸方向におよそ  $0.653s^2$  で広がることから、同精度のルンゲクッタ法と比較しても図 A.2.7 のように広い数値的安定性を

持つ。

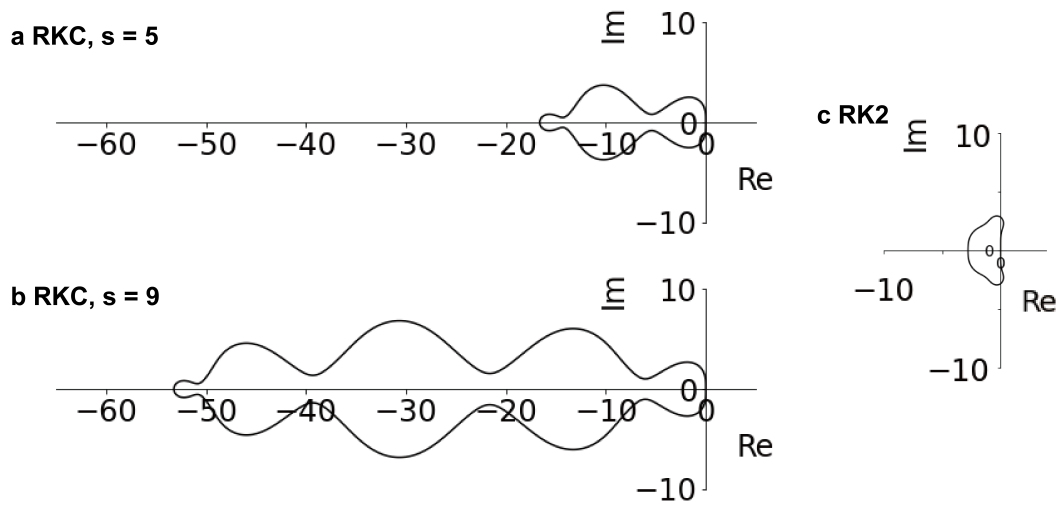


図 A.2.7: RKC 法の安定領域。a と b はステージ数  $s$  が 5 と 9 の場合の RKC 法における安定領域を示し、c は比較対象として、2 段 2 次の陽的ルンゲクッタ法における安定領域を示す。

## A.3 CUDA プログラミングの詳細

本節では、本研究で用いた CUDA プログラミングの詳細、特に CUDA 用の線形代数学ライブラリである cuBLAS と cuSPARSE を用いた CG 法の実装部分と、RKC 法におけるベクトル要素の二乗平均平方根を計算する際に使用した Parallel Reduction 技術の詳細を記述する。

### A.3.1 cuBLAS および cuSPARSE による CG 法の実装

CUDA には、線形代数用ライブラリである BLAS (Basic Linear Algebra Subprograms) を GPU プログラミング用に最適化した cuBLAS [36] と、疎行列・疎ベクトル向けの代数計算を最適化した cuSPARSE [83] が存在する。本節では、これらの CUDA に最適化されたライブラリを用いて、CG 法を実装するための詳細を記述する。

#### CG 法のアルゴリズム

CG 法は  $n$  次の連立方程式  $Ax = b$  の解を二次関数  $\phi(x) = \frac{1}{2}(x, Ax) - (b, x)$  の最小値探索問題であると考え、以下のアルゴリズムによって実装される。ただし、 $A$ 、 $x$ 、 $b$  はそれぞれ、 $n$  次の正定値対称行列、 $n$  次の未知ベクトル、 $n$  次の既知ベクトルである。

1. 初期近似解  $x_0$  を設定する
2. 初期近似解に対する残差ベクトル ( $r_0 = b_0 - Ax_0$ ) を計算し、探索方向ベクトル  $p_0$  を  $r_0$  で初期化する
3. 以下の計算を収束条件を満たすまで繰り返す (for  $k = 0, 1, 2, \dots$ )
  - (a)  $\alpha_k = \frac{\|r_k\|^2}{(p_k, Ap_k)}$  : 探索方向ベクトルの修正係数
  - (b)  $x_{k+1} = x_k + \alpha_k p_k$  : 近似解の更新
  - (c)  $r_{k+1} = r_k - \alpha_k Ap_k$  : 近似解の更新
  - (d)  $\|r_{k+1}\| \leq \epsilon$  : 収束条件
  - (e)  $\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$  : 探索方向ベクトルを更新するための修正係数
  - (f)  $p_{k+1} = r_{k+1} + \beta_k p_k$

CG 法は  $n$  次の正定値対称行列  $A$  とのベクトル行列演算を繰り返すため、 $A$  が疎行列である

場合、演算量が少ないため、ガウスの消去法や LU 分解よりも優れている場合が多い。特に、マルチコンパートメントモデルにおける行列  $A$  はケーブル方程式の係数からなるため非常にスパースである。

次節では、このような疎行列の格納方式について述べる。

### 疎行列格納方式

ガウスの消去法などのアルゴリズムと異なり、CG 法において疎行列  $A$  は計算の途中で要素が変化しない。そこで、 $A$  のゼロ要素を配列要素として確保しておく必要が存在しないため、疎行列用の格納方式を用いることでメモリの節約をすることができる。本研究では、最も代表的な疎行列格納方式である、CSR (Compressed Sparse Row) 方式を用いた。

CSR では、疎行列  $A$  (次数  $n$ 、非ゼロ要素の個数  $nnz$ ) を以下の 3 つの配列に格納することでメモリの削減を図る。

val : 行列の非ゼロ要素の値を格納するための行列

col : 非ゼロ要素の列番号を格納するための行列

row : 各行の先頭の非ゼロ要素が val 内で何番目に格納されているか示すための行列

ここで、val と col は要素数が  $nnz$  であり、row は  $n + 1$  である。また、row の最初の要素は 0 とし最後の要素は非ゼロ要素の個数とする。インデックスを 0 ベースとしたとき、次の行列  $A$  は以下のように格納される。

$$A = \begin{pmatrix} 0 & 0 & 5 & 0 & 3 \\ 0 & 2 & 0 & 7 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{val} = \{5, 3, 2, 7, 9, 1, 3, 2\}$$

$$\text{col} = \{2, 4, 1, 3, 2, 3, 0, 1\}$$

$$\text{row} = \{0, 2, 4, 5, 6, 8\}$$

### cuBLAS によるベクトル演算

cuBLAS は密行列や密ベクトルによる線形代数演算を CUDA で最適にサポートしたライブラリである。このライブラリでは、Level が 3 段階に分かれており、Level 1 ではベクトル

同士の演算、Level 2 ではベクトルと行列の演算、Level 3 では行列同士の演算である。本研究で扱う行列は疎行列であり、cuBLAS は主にベクトル同士の演算のみを行うため、Level 1 の API のみを用いた。

cuBLAS を使用する場合には、ライブラリを使用するための `cublasHandle_t` 型の変数と関数実行のステータスを格納するために必要な `cublasStatus_t` 型の変数を宣言する必要がある。

```
cublasStatus_t stat1;  
cublasHandle_t handle1;  
stat1 = cublasCreate_v2(&handle1);
```

倍精度浮動小数点の  $n$  次ベクトル  $x$  をベクトル  $y$  にコピーする API は次のようになる。

```
cublasStatus_t cublasDcopy(cublasHandle_t handle, int n,  
    const double *x, int incx, double *y, int incy)
```

ここで、`incx` と `incy` はベクトルの要素の処理間隔であるが、全ての要素のコピーであれば 1 で設定する。

同様に、倍精度浮動小数点のベクトル  $x$  のスカラー倍 ( $x = \alpha x$ ) は次のように行う。

```
cublasStatus_t cublasDscal(cublasHandle_t handle, int n,  
    const double *alpha, double *x, int incx)
```

ベクトル  $x$  を  $\alpha$  倍し、ベクトル  $y$  に加算する API ( $x = \alpha x + y$ ) は次のようになる。

```
cublasStatus_t cublasDaxpy(cublasHandle_t handle,  
    int n, const double *alpha, const double *x,  
    int incx, double *y, int incy)
```

最後に、2つのベクトルの内積 ( $\alpha = (x, y)$ ) の API は次のようになる。

```
cublasStatus_t cublasDdot (cublasHandle_t handle, int n,  
    const double *x, int incx, const double *y, int incy,  
    double *alpha)
```

### cuSPARSE による疎行列ベクトル演算

cuSPARSE は疎行列に対する線形代数演算を CUDA で最適にサポートしたライブラリである。このライブラリも cuBLAS 同様に、Level が 3 段階に分かれており、Level 1 ではベ

クトル同士の演算、Level 2 ではベクトルと行列の演算、Level 3 では行列同士の演算である。本研究では、疎行列  $A$  とベクトル  $x$  の演算のみであるため、Level 2 の API のみを用いた。

cuSPARSE を使用する場合、cuBLAS 同様にライブラリを使用するための `cusparseHandle_t` 型の変数と関数実行のステータスを格納するために必要な `cusparseStatus_t` 型の変数を宣言する必要がある。

```
cusparseStatus_t stat2;
cusparseHandle_t handle2;
stat2 = cusparseCreate(&handle2);
```

また、cuSPARSE では疎行列に関する情報を格納するための構造体である `cusparseMatDescr_t` を用意する必要がある。

```
typedef struct {
    cusparseMatrixType_t MatrixType;
    cusparseFillMode_t FillMode;
    cusparseDiagType_t DiagType;
    cusparseIndexBase_t IndexBase;
} cusparseMatDescr_t;
```

構造体の宣言と生成は以下のように行う。

```
cusparseMatDescr_t descrA;
cusparseCreateMatDescr(&descrA);
```

また、行列の種類を次のように指定する。

```
cusparseSetMatType(descrA, CUSPARSE_MATRIX_TYPE_GENERAL);
cusparseSetMatIndexBase(descrA, CUSPARSE_INDEX_BASE_ZERO);
```

1 行目は対称行列の指定、2 行目は行列のインデックスが 0 ベースであることの指定である。

最後に、CSR ( $y = \alpha Ax + \beta y$ ) を行うための API は次のようになる。

```
cusparseStatus_t cusparseDcsrsv(cusparseHandle_t handle,
    cusparseOperation_t transA, int m, int n, int nnz,
    const double *alpha, const cusparseMatDescr_t descrA,
    const double *csrValA, const int *csrRowPtrA,
    const int *csrColIndA, const double *x,
    const double *beta, double *y)
```

ここで、 $m$  と  $n$  は行列とベクトルの次元数、 $nnz$  は行列の非ゼロ要素、`csrValA`、`csrRowPtrA`、`csrColIndA` は CSR 形式における、`val`、`row`、`col` を格納したベクトルである。また、`transA`

は行列の転置の指定であり、転置しない場合は CUSPARSE\_OPERATION\_NON\_TRANSPOSE とする。

## CG 法の実装

最後に cuBLAS と cuSPARSE を用いた CG 法のコードを以下に示す。

```
__host__ static void cg_cusparses_crs (const int n,
    const int nnz, const double *d_val, const int *d_col,
    const int *d_row, double *d_x, double *d_b)
{
    static double *d_r, *d_p, *d_ap; //ap = A * p
    // Secure the array on the GPU
    cudaMalloc ( ( double ** ) &d_r, n * sizeof ( double ) );
    cudaMalloc ( ( double ** ) &d_p, n * sizeof ( double ) );
    cudaMalloc ( ( double ** ) &d_ap, n * sizeof ( double ) );

    double bnorm, rnorm_k, rnorm_k1;
    double alpha, beta, pap; //pap = (p, Ap)
    double epsilon = 1.0e-15;
    double cp1 = 1.0;
    double c0 = 0.0;
    double cm1 = -1.0;

    // initialization of cublas and cusparses
    cublasStatus_t stat1;
    cublasHandle_t handle1;
    cusparseStatus_t stat2;
    cusparseHandle_t handle2;
    cusparseMatDescr_t descrA;
    stat1 = cublasCreate_v2(&handle1);
    stat2 = cusparseCreate(&handle2);
    cusparseCreateMatDescr(&descrA);
    cusparseSetMatType(descrA, CUSPARSE_MATRIX_TYPE_GENERAL);
    cusparseSetMatIndexBase(descrA, CUSPARSE_INDEX_BASE_ZERO);

    stat1 = cublasDscal(handle1, n, &c0, d_x, 1); // x = 0
    stat1 = cublasDcopy(handle1, n, d_b, 1, d_r, 1); // r = b
    stat1 = cublasDcopy(handle1, n, d_r, 1, d_p, 1); // p = r
    stat1 = cublasDdot(handle1, n, d_b, 1, d_b, 1, &bnorm); // ||b||

    for (int k = 0; k < 100; k++) {
        stat2 = cusparseDcsrsv(&handle2,
```



```

CUSPARSE_OPERATION_NON_TRANSPOSE,
n, n, nnz, &cp1, descrA, d_val, d_row,
d_col, d_p, &c0, d_ap); // Ap
stat1 = cublasDdot(handle1, n, d_r, 1, d_r, 1, &rnorm_k);
// ||r_k||^2
stat1 = cublasDdot(handle1, n, d_p, 1, d_ap, 1, &pap); // pAp
alpha = rnorm_k / pap; // alpha
stat1 = cublasDaxpy(handle1, n, &alpha, d_p, 1, d_x, 1);
// x += alpha * p
alpha = -1.0 * alpha;
stat1 = cublasDaxpy(handle1, n, &alpha, d_ap, 1, d_r, 1);
// r -= alpha * ap
stat1 = cublasDdot(handle1, n, d_r, 1, d_r, 1, &rnorm_k1);
// ||r_{k+1}||^2

if (sqrt(rnorm_k1) <= epsilon * sqrt(bnorm)) { break; }

// p = r + beta * p
beta = rnorm_k1 / rnorm_k;
stat1 = cublasDscal(handle1, n, &beta, d_p, 1);
stat1 = cublasDaxpy(handle1, n, &cp1, d_r, 1, d_p, 1);
}

// finalization of cublas and cusparse
cusparseDestroyMatDescr(descrA);
cublasDestroy_v2(handle1);
cusparseDestroy(handle2);

//Free memory on the GPU
cudaFree ( d_r );
cudaFree ( d_p );
cudaFree ( d_ap );
}

```

### A.3.2 Parallel Reduction による効率的なベクトル要素の総和計算

RKC 法は陽解法であるため、連立方程式の求解が必要なく、CG 法のような煩雑なベクトル演算をほとんど必要としない。しかしながら、RKC 法における局所誤差の推定において、各コンパートメントの膜電位の二乗平均平方根を計算する必要がある。この二乗平均平方根を計算するには、ベクトル要素の総和 ( $x_{\text{sum}} = \sum_i x_i$ ) を計算する必要があるが、これは各スレッドが  $x_{\text{sum}}$  の値を読み取り、そこにそのスレッドが持つ  $x_i$  の値を足し、 $x_{\text{sum}}$  に新たに

書き込む操作が必要になる。しかし、この演算を単純に並列化してしまうと、あるスレッドが値を読み込むのと並行して別のスレッドが値を書き込むといった衝突が起こるなど、適切な演算が行えない。そこで CUDA では1つのスレッドが書き込む場合、他のスレッドからの操作をブロックする AtomicAdd などの API によって実装する必要があるが、これは計算が逐次実行されていることにほかならず、本研究のような大規模なモデルの計算には適していない。そこで、本研究では、Parallel Reduction (図 A.3.8) と呼ばれる、値の畳み込み計算の並列化技術を用いて効率的なベクトルの総和計算を実装した [84]。

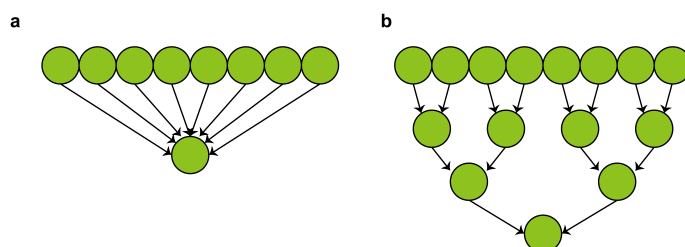


図 A.3.8: Parallel Reduction における総和演算の概念図。ベクトルの各要素を逐次的に足し合わせる場合 (a)、1つのスレッドが操作している間には他のスレッドからの操作をブロックする必要があり、その計算量は要素数  $n$  と等しい。Parallel Reduction (b) では、順番に2つの要素を足し合わせていき、各段を並列的に処理することが可能であり、その計算量は二分木の深さ ( $\log_2(n)$ ) となる。[84] を基に作成。

Parallel Reduction は、ベクトルの各要素を上から順番に2つずつ足し合わせることで、各和算を並列に処理する技術である。全ての要素を逐次的に足し合わせる場合、ベクトルの要素数 ( $n$ ) の計算回数が必要になるが、Parallel Reduction では、(図 A.3.8b) その計算量は二分木の深さ ( $\log_2(n)$ ) となる。Parallel Reduction における各スレッドの動作の概念図を図 A.3.9 に示す。また、各スレッドが計算した値をシェアードメモリを用いて共有することで、グローバルメモリへのアクセスを防ぎ、計算の遅延を回避する。

最後に、最も基本的な Parallel Reduction の実装コードを次に示す ([84] より改変)。i\_data、o\_data はそれぞれ入力データの配列であり、s\_data はシェアードメモリ上のデータ配列である。

```

__global__ void parallel_reduction(int *i_data, int *o_data) {
    extern __shared__ int s_data[];
    // Store the values from global memory to shared memory
    unsigned int tid = threadIdx.x;
    unsigned int id = blockIdx.x*blockDim.x + threadIdx.x;

```

```

s_data[tid] = i_data[id];
__syncthreads();
// parallel reduction in shared memory
for(unsigned int s=1; s < blockDim.x; s *= 2) {
//sum of the values with one skipped
  if (tid % (2*s) == 0) {
    s_data[tid] += s_data[tid + s];
  }
  __syncthreads();
}
// return the value of thread 0
if (tid == 0) o_data[blockIdx.x] = s_data[0];
}

```

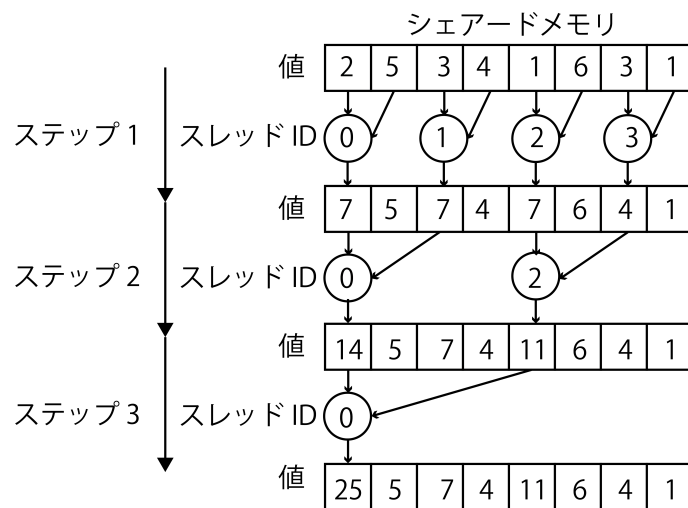


図 A.3.9: Parallel Reduction における各スレッドの動作の概念図。四角はベクトルの各要素の値であり、シェアードメモリを用いている。丸は各スレッドであり、隣り合う各要素の値を 2 つずつ畳み込んでいる。[84] を基に作成。

## 参考文献

- [1] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, and A James Hudspeth. *Principles of neural science*, volume 5. McGraw-hill New York, 2000.
- [2] Henry Markram. The blue brain project. *Nature Reviews Neuroscience*, 7(2):153–160, 2006.
- [3] National Institutes of Health (NIH). *The BRAIN Initiative*. <https://braininitiative.nih.gov/>. Accessed 15 Oct 2021.
- [4] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [5] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [6] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.
- [7] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [8] Rajagopal Ananthanarayanan, Steven K Esser, Horst D Simon, and Dharmendra S Modha. The cat is out of the bag: cortical simulations with  $10^9$  neurons,  $10^{13}$  synapses. In *Proceedings of the conference on high performance computing networking, storage and analysis*, pages 1–12, 2009.
- [9] Moritz Helias, Susanne Kunkel, Gen Masumoto, Jun Igarashi, Jochen Martin Epler, Shin Ishii, et al. Supercomputers ready for use as discovery machines for

- neuroscience. *Frontiers in neuroinformatics*, 6:26, 2012.
- [10] Susanne Kunkel, Maximilian Schmidt, Jochen M Eppler, Hans E Plesser, Gen Masumoto, Jun Igarashi, et al. Spiking network simulation code for petascale computers. *Frontiers in neuroinformatics*, 8:78, 2014.
- [11] Jakob Jordan, Tammo Ippen, Moritz Helias, Itaru Kitayama, Mitsuhsa Sato, Jun Igarashi, et al. Extremely scalable spiking neuronal network simulation code: from laptops to exascale computers. *Frontiers in neuroinformatics*, 12:2, 2018.
- [12] Jakob Jordan, Moritz Helias, Markus Diesmann, and Susanne Kunkel. Efficient communication in distributed simulations of spiking neuronal networks with gap junctions. *Frontiers in neuroinformatics*, 14:12, 2020.
- [13] Hiroshi Yamaura, Jun Igarashi, and Tadashi Yamazaki. Simulation of a human-scale cerebellar network model on the K computer. *Frontiers in neuroinformatics*, 14:16, 2020.
- [14] 五十嵐 潤, 山浦 洋, 野村 昂太郎, 山崎 匡. 富岳コンピュータによるヒト規模大脳皮質-小脳-視床回路シミュレーションに向けて. 第 44 回日本神経科学大会シンポジウム. 2021 年 7 月 30 日. 神戸国際会議場.
- [15] Tiago Branco, Beverley A Clark, and Michael Häusser. Dendritic discrimination of temporal input sequences in cortical neurons. *Science*, 329(5999):1671–1675, 2010.
- [16] Albert Gidon, Timothy Adam Zolnik, Pawel Fidzinski, Felix Bolduan, Athanasia Papoutsis, Panayiota Poirazi, et al. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 367(6473):83–87, 2020.
- [17] Toviah Moldwin and Idan Segev. Perceptron learning and classification in a modeled cortical pyramidal cell. *Frontiers in computational neuroscience*, 14, 2020.
- [18] Eugene M Izhikevich and Gerald M Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the national academy of sciences*, 105(9):3593–3598, 2008.
- [19] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W Reimann, Marwan Abdellah, Carlos Aguado Sanchez, et al. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015.
- [20] Eren Sezener, Agnieszka Grabska-Barwińska, Dimitar Kostadinov, Maxime Beau, Sanjukta Krishnagopal, David Budden, et al. A rapid and efficient learning rule for

- biological neural circuits. *bioRxiv*, 2021.
- [21] Marianne J Bezaire, Ivan Raikov, Kelly Burk, Dhrumil Vyas, and Ivan Soltesz. Interneuronal mechanisms of hippocampal theta oscillations in a full-scale model of the rodent CA1 circuit. *Elife*, 5:e18566, 2016.
- [22] Pramod Kumbhar, Michael Hines, Jeremy Fouriaux, Aleksandr Ovcharenko, James King, Fabien Delalondre, et al. CoreNEURON: an optimized compute engine for the NEURON simulator. *Frontiers in neuroinformatics*, 13:63, 2019.
- [23] Nora Abi Akar, Ben Cumming, Vasileios Karakasis, Anne Küsters, Wouter Klijn, Alexander Peyser, et al. Arbor – a morphologically-detailed neural network simulation library for contemporary high-performance computing architectures. In *2019 27th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pages 274–282. IEEE, 2019.
- [24] Michael L Hines and Nicholas T Carnevale. The NEURON simulation environment. *Neural computation*, 9(6):1179–1209, 1997.
- [25] Philip Machanick. Approaches to addressing the memory wall. *School of IT and Electrical Engineering, University of Queensland*, 2002.
- [26] Janghoo Lim, Juan Crespo-Barreto, Paymaan Jafar-Nejad, Aaron B Bowman, Ronald Richman, David E Hill, et al. Opposing effects of polyglutamine expansion on native protein complexes contribute to sca1. *Nature*, 452(7188):713–718, 2008.
- [27] Takafumi Inoue, Xi Lin, Kristi A Kohlmeier, Harry T Orr, Huda Y Zoghbi, and William N Ross. Calcium dynamics and electrophysiological properties of cerebellar purkinje cells in sca1 transgenic mice. *Journal of neurophysiology*, 85(4):1750–1760, 2001.
- [28] Yunliang Zang, Stéphane Dieudonné, and Erik De Schutter. Voltage-and branch-specific climbing fiber responses in Purkinje cells. *Cell reports*, 24(6):1536–1549, 2018.
- [29] Christof Koch and Idan Segev. *Methods in neuronal modeling: from ions to networks*. MIT press, 2nd edition, 1998.
- [30] Matthew A Wilson, Upinder S Bhalla, John D Uhley, and James M Bower. GENESIS: a system for simulating neural networks. *Advances in Neural Information*

*Processing Systems*, 1:485–492, 1989.

- [31] Erik De Schutter and James M Bower. An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. *Journal of neurophysiology*, 71(1):375–400, 1994.
- [32] Erik De Schutter and James M Bower. An active membrane model of the cerebellar Purkinje cell II. Simulation of synaptic responses. *Journal of neurophysiology*, 71(1):401–419, 1994.
- [33] NVIDIA. *CUDA C Programming Guide*. [https://docs.nvidia.com/cuda/pdf/CUDA\\_C\\_Programming\\_Guide.pdf](https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf), 2021. Accessed 24 March 2021.
- [34] NVIDIA. *NVIDIA Tesla V100 GPU Architecture*. <https://images.nvidia.com/content/pdf/tesla/Volta-Architecture-Whitepaper-v1.1-jp.pdf>, 2022. Accessed 16 January 2022.
- [35] John Cheng, Max Grossman, and Ty McKercher. *CUDA C プロフェッショナル プログラミング*. 株式会社インプレス. 株式会社クイープ (訳) , 森野 慎也 (監訳) .
- [36] NVIDIA. *CUBLAS Library*. [https://docs.nvidia.com/cuda/pdf/CUBLAS\\_Library.pdf](https://docs.nvidia.com/cuda/pdf/CUBLAS_Library.pdf), 2021. Accessed 24 March 2021.
- [37] Nicolas Brunel, Vincent Hakim, Philippe Isope, Jean-Pierre Nadal, and Boris Barbour. Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. *Neuron*, 43(5):745–757, 2004.
- [38] Roger D Traub, Diego Contreras, Mark O Cunningham, Hilary Murray, Fiona EN LeBeau, Anita Roopun, et al. Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. *Journal of neurophysiology*, 93(4):2194–2232, 2005.
- [39] Roger D Traub, Steven J Middleton, Thomas Knöpfel, and Miles A Whittington. Model of very fast ( $> 75$  Hz) network oscillations generated by electrical coupling between the proximal axons of cerebellar purkinje cells. *European Journal of Neuroscience*, 28(8):1603–1616, 2008.
- [40] Shyam Kumar Sudhakar, Sungho Hong, Ivan Raikov, Rodrigo Publio, Claus Lang, Thomas Close, et al. Spatiotemporal network coding of physiological mossy fiber inputs by the cerebellar granular layer. *PLoS computational biology*, 13(9):e1005754, 2017.

- [41] Yazan N Billeh, Binghuang Cai, Sergey L Gratiy, Kael Dai, Ramakrishnan Iyer, Nathan W Gouwens, et al. Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. *Neuron*, 106(3):388–403, 2020.
- [42] Samuel A Neymotin, Dylan S Daniels, Blake Caldwell, Robert A McDougal, Nicholas T Carnevale, Mainak Jas, et al. Human neocortical neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. *Elife*, 9:e51214, 2020.
- [43] Kael Dai, Sergey L Gratiy, Yazan N Billeh, Richard Xu, Binghuang Cai, Nicholas Cain, et al. Brain modeling toolkit: An open source software suite for multiscale modeling of brain circuits. *PLOS Computational Biology*, 16(11):e1008386, 2020.
- [44] Jan G Verwer, Willem H Hundsdorfer, and Ben P Sommeijer. Convergence properties of the Runge-Kutta-Chebyshev method. *Numerische Mathematik*, 57(1):157–178, 1990.
- [45] Jan G Verwer. Explicit Runge-Kutta methods for parabolic partial differential equations. *Applied Numerical Mathematics*, 22(1-3):359–379, 1996.
- [46] PJ Van der Houwen. The development of Runge-Kutta methods for partial differential equations. *Applied Numerical Mathematics*, 20(3):261–272, 1996.
- [47] Ben P Sommeijer, Laurence F Shampine, and Jan G Verwer. RKC: An explicit solver for parabolic PDEs. *Journal of Computational and Applied Mathematics*, 88(2):315–326, 1998.
- [48] Jan G Verwer, Ben P Sommeijer, and W Hundsdorfer. RKC time-stepping for advection–diffusion–reaction problems. *Journal of Computational Physics*, 201(1):61–79, 2004.
- [49] Katarzyna Dover, Christopher Marra, Sergio Solinas, Marko Popovic, Sathya Subramaniyam, Dejan Zecevic, et al. FHF-independent conduction of action potentials along the leak-resistant cerebellar granule cell axon. *Nature communications*, 7(1):1–11, 2016.
- [50] Sergio Solinas, Lia Forti, Elisabetta Cesana, Jonathan Mapelli, Erik De Schutter, and Egidio D’Angelo. Computational reconstruction of pacemaking and intrinsic electroresponsiveness in cerebellar Golgi cells. *Frontiers in cellular neuroscience*,



1:2, 2007.

- [51] Jornt R De Gruijl, Paolo Bazzigaluppi, Marcel TG de Jeu, and Chris I De Zeeuw. Climbing fiber burst size and olivary sub-threshold oscillations in a network setting. *PLoS Comput Biol*, 8(12):e1002814, 2012.
- [52] Guillaume P Dugué, Nicolas Brunel, Vincent Hakim, Eric Schwartz, Mireille Chat, Maxime Lévesque, et al. Electrical coupling mediates tunable low-frequency oscillations and resonance in the cerebellar Golgi cell network. *Neuron*, 61(1):126–139, 2009.
- [53] Koen Vervaeke, Andrea Lőrincz, Pádraig Gleeson, Matteo Farinella, Zoltan Nusser, and R Angus Silver. Rapid desynchronization of an electrically coupled interneuron network with sparse excitatory synaptic input. *Neuron*, 67(3):435–451, 2010.
- [54] Koen Vervaeke, Andrea Lőrincz, Zoltan Nusser, and R Angus Silver. Gap junctions compensate for sublinear dendritic integration in an inhibitory network. *Science*, 335(6076):1624–1628, 2012.
- [55] Soon-Lim Shin and Erik De Schutter. Dynamic synchronization of Purkinje cell simple spikes. *Journal of neurophysiology*, 96(6):3485–3491, 2006.
- [56] Andrew K Wise, Nadia L Cerminara, Dilwyn E Marple-Horvat, and Richard Apps. Mechanisms of synchronous activity in cerebellar Purkinje cells. *The Journal of physiology*, 588(13):2373–2390, 2010.
- [57] FABRIZIO Gabbiani, JENS Midtgaard, and THOMAS Knopfel. Synaptic integration in a model of cerebellar granule cells. *Journal of neurophysiology*, 72(2):999–1009, 1994.
- [58] NVIDIA. *CUDA Profiler User’s Guide*. <https://docs.nvidia.com/cuda/profiler-users-guide/index.html#nvprof-overview>, 2021. Accessed 16 July 2021.
- [59] NVIDIA. *CUDA C++ Best Practices Guide*. <https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html>, 2021. Accessed 16 July 2021.
- [60] Guillaume Pernelle, Wilten Nicola, and Claudia Clopath. Gap junction plasticity as a mechanism to regulate network-wide oscillations. *PLoS computational biology*, 14(3):e1006025, 2018.
- [61] Michael VL Bennett and R Suzanne Zukin. Electrical coupling and neuronal syn-

- chronization in the mammalian brain. *Neuron*, 41(4):495–511, 2004.
- [62] Steven J Middleton, Claudia Racca, Mark O Cunningham, Roger D Traub, Hannah Monyer, Thomas Knöpfel, et al. High-frequency network oscillations in cerebellar cortex. *Neuron*, 58(5):763–774, 2008.
- [63] Timothy A Blenkinsop and Eric J Lang. Block of inferior olive gap junctional coupling decreases purkinje cell complex spike synchrony and rhythmicity. *Journal of Neuroscience*, 26(6):1739–1748, 2006.
- [64] Camille de Solages, German Szapiro, Nicolas Brunel, Vincent Hakim, Philippe Isope, Pierre Buisseret, et al. High-frequency organization and synchrony of activity in the purkinje cell layer of the cerebellum. *Neuron*, 58(5):775–788, 2008.
- [65] Kyung-Seok Han, Chong Guo, Christopher H Chen, Laurens Witter, Tomas Osorno, and Wade G Regehr. Ephaptic coupling promotes synchronous firing of cerebellar Purkinje cells. *Neuron*, 100(3):564–578, 2018.
- [66] Taro Ishikawa, Misa Shimuta, and Michael Häusser. Multimodal sensory integration in single cerebellar granule cells in vivo. *Elife*, 4:e12916, 2015.
- [67] Daniel Majoral, Ajmal Zemmar, and Raul Vicente. A model for time interval learning in the Purkinje cell. *PLoS computational biology*, 16(2):e1007601, 2020.
- [68] Mark Sofroniou and Giulia Spaletta. Extrapolation methods in mathematica. *JNAIAM J. Numer. Anal. Indust. Appl. Math*, 3:105–121, 2008.
- [69] Kyle E Niemeyer and Chih-Jen Sung. Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs. *Journal of Computational Physics*, 256:854–871, 2014.
- [70] Artur I Karimov, Denis N Butusov, and Aleksandra V Tutueva. Adaptive explicit-implicit switching solver for stiff ODEs. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, pages 440–444. IEEE, 2017.
- [71] Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M Bower, et al. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of computational neuroscience*, 23(3):349–398, 2007.
- [72] Abigail L Person and Indira M Raman. Purkinje neuron synchrony elicits time-locked spiking in the cerebellar nuclei. *Nature*, 481(7382):502–505, 2012.

- [73] Shyam Kumar Sudhakar, Benjamin Torben-Nielsen, and Erik De Schutter. Cerebellar nuclear neurons use time and rate coding to transmit purkinje neuron pauses. *PLoS computational biology*, 11(12):e1004641, 2015.
- [74] Tycho M Hoogland, Jornt R De Gruijl, Laurens Witter, Cathrin B Canto, and Chris I De Zeeuw. Role of synchronous activation of cerebellar purkinje cell ensembles in multi-joint movement control. *Current Biology*, 25(9):1157–1165, 2015.
- [75] Stefano Casali, Elisa Marenzi, Chaitanya Medini, Claudia Casellato, and Egidio D’Angelo. Reconstruction and simulation of a scaffold model of the cerebellar network. *Frontiers in neuroinformatics*, 13:37, 2019.
- [76] Carlos D Brody. Correlations without synchrony. *Neural computation*, 11(7):1537–1551, 1999.
- [77] David A McCormick and Richard F Thompson. Cerebellum: essential involvement in the classically conditioned eyelid response. *Science*, 223(4633):296–299, 1984.
- [78] David J Krupa, Judith K Thompson, and Richard F Thompson. Localization of a memory trace in the mammalian brain. *Science*, 260(5110):989–991, 1993.
- [79] Masao Ito. Cerebellar long-term depression: characterization, signal transduction, and functional roles. *Physiological reviews*, 81(3):1143–1195, 2001.
- [80] Kazuto Fujishima, Kelly Kawabata Galbraith, and Mineko Kengaku. Dendritic self-avoidance and morphological development of cerebellar purkinje cells. *The Cerebellum*, 17(6):701–708, 2018.
- [81] Nora Vrieler, Sebastian Loyola, Yasmin Yarden-Rabinowitz, Jesse Hoogendorp, Nikolay Medvedev, Tycho M Hoogland, et al. Variability and directionality of inferior olive neuron dendrites revealed by detailed 3d characterization of an extensive morphological library. *Brain Structure and Function*, 224(4):1677–1695, 2019.
- [82] Ernst Hairer and Gerhardy Wanner. 常微分方程式の数値解法 2. シュプリンガー・ジャパン, 2008. 三井 斌友 (監訳) .
- [83] NVIDIA. *CUSPARSE Library*. [https://docs.nvidia.com/cuda/pdf/CUSPARSE\\_Library.pdf](https://docs.nvidia.com/cuda/pdf/CUSPARSE_Library.pdf), 2022. Accessed 16 January 2022.
- [84] NVIDIA. *Optimizing Parallel Reduction in CUDA*. <https://developer.download.nvidia.com/assets/cuda/files/reduction.pdf>, 2022. Accessed 16 January 2022.

# 関連論文の印刷公表の方法および 時期

## 論文 (査読有)

1. 著者名 : Taira Kobayashi, Rin Kuriyama, Tadashi Yamazaki  
論文題目 : Testing an explicit method for multi-compartment  
neuron model simulation on a GPU  
雑誌名 : Cognitive Computation  
発行時期 : 2021 年 11 月  
(本文第 3 章に関連)