

jh250014

宇宙地球惑星系連成シミュレーションに向けた計算技術開発

三宅洋平（神戸大学大学院システム情報学研究科）

概要

本研究の目的は、異なる時空間スケールや素過程をターゲットとする複数のシミュレーションモデルをプロセス間通信により協調動作させることで、次世代の宇宙プラズマ・惑星大気シミュレーションを実現することである。独自に開発した MPMD モデルに基づくコード間結合フレームワーク CoToCoA を活用し、柔軟かつ省コストに連成シミュレーションを実装する。今年度は連成対象の要素モデルとして 2 次元のモンテカルロ直接シミュレーション (DSMC) への動的負荷分散機能の実装と性能検証を実施した。また連成対象に Python プログラムを加えるための CoToCoA 拡張機能に基づき、主解析出力のポスト処理機能を結合した連成解析モデルを構築した。

1. 共同研究に関する情報

(1) 共同利用・共同研究を実施している拠点名

京都大学 学術情報メディアセンター

九州大学 情報基盤研究開発センター

(2) 課題分野

大規模計算科学課題分野

(3) 参加研究者一覧と役割分担

三宅・白井 (神戸大学)	宇宙プラズマ連成モデルの設計・応用
酒田・楠部 (神戸大学)	宇宙プラズマ・惑星大気連成モデルコードの開発
土田・寺西 (神戸大学)	宇宙プラズマ高効率計算手法の検討・開発
南里豪志 (九州大学)	連成フレームワーク CoToCoA の開発
深沢圭一郎 (総合地球環境学研究所)	グローバルプラズマシミュレーションモデルの開発
加藤雄人 (東北大学)	メソスケールプラズマシミュレーションモデルの開発
寺田直樹 (東北大学)	惑星上層中性大気計算の妥当性検証

2. 研究の目的と意義

本研究の目的は、異なる時空間スケールや素過程をターゲットとする複数のシミュレ

ーションモデルをプロセス間通信により協調動作させることで、マルチスケール/マルチフィジックスの宇宙プラズマ・惑星大気シミュレーションを実現することである。複合システムである宇宙地球惑星系の各領域や時空間スケールの物理現象計算の間を縮約化された物理情報で結合する「連成シミュレーション」を、本研究では独自のコード間結合フレームワーク CoToCoA の活用により柔軟かつ省コストに実装し、その有効性を探求する。連成フレームワークは、異なるシミュレーションプログラムを低作業コスト・高計算効率で結合する方法論と定義づけることができ、本課題で連成シミュレーションを実現する際に蓄積されたノウハウそのものが、幅広い応用分野への波及効果を持つ。

本申請は過年度からの継続課題であり、これまでに惑星上層における中性・電離大気統合数値シミュレーションの要素技術となるモンテカルロ直接シミュレーション (DSMC) への動的負荷分散機能実装の取り組みを開始し、1 次元モデルでの動作検証を完了した。これを引き継ぎ本年度は、2 次元のモンテカルロ直接シミュレーション (DSMC) への動

動的負荷分散機能の実装と性能検証に取り組む。また既存の DSMC シミュレータに実装されている光化学反応処理をより汎用性を有する外部開発モデル (PROTEUS) に連成計算の枠組みで置き換える改良も実施する。これに加えて、CoToCoA の機能拡充として Python プログラムへの対応と、当該機能を応用した連成解析の実装を推進し、より広範な用途に使用できるフレームワークに拡充を行う。

3. 当拠点の公募型共同研究として実施した意義

部分プロセス空間内で独立に動作する複数のコードが高効率かつ正しく情報伝達を行うには、メモリー貫性に配慮した上でリモートメモリアクセス方式に基づいた並列プログラムを記述する必要があり、計算機科学分野の知識と経験が必要である。一方で、CoToCoA のような、連成フレームワークを設計する上では、実際の連成アプリケーションでやり取りが発生するデータの種別やサイズ、頻度に応じて仕様を策定する必要があり、ソフトウェア設計者と利用者間のコデザインが必要である。また個別のシミュレーションコードの効率化作業においても、両分野の協働が極めて有効であることは、申請者のこれまでの JHPCN 課題の成果が示してきたとおりである。このように本課題は、計算科学と計算機科学分野の緊密な連携が必要不可欠な研究テーマであり、拠点公募型共同研究として実施する意義は大きい。

4. 前年度までに得られた研究成果の概要

独立に設計・開発されてきた科学技術計算プログラム同士を省コストに結合する数値技術の実証として、磁気圏グローバル MHD 計算と人工衛星帯電予測計算を CoToCoA に基づき連結することで、磁気圏環境の時間変動効果を反映した人工衛星の帯電予測技術の構築に取り組んだ。2024 年度は、CoToCoA

を活用した結合計算の基本実装、現実に即した衛星帯電評価を行うための機能拡充に加えて、空間方向の自由度を持たない衛星帯電モデルに対して Parareal 法に基づく時間方向並列化を実装した。粗い時間刻み幅の時間更新アルゴリズムを適切に選択することで、32 プロセス使用時に最大 29 倍の並列加速率を達成している。開発された連成モデルは、MHD と帯電計算の双方にマルチプロセスを柔軟に割り当てることができ、連結モデル間の計算負荷バランスの調整が可能な実装となった。これらの成果をもって、宇宙環境変動効果を範囲した衛星帯電評価計算の基盤技術が概ね確立された。

2024 年度から宇宙理学分野における中核的研究課題をターゲットに据えた数値技術開発を開始し、1 次元モンテカルロ直接シミュレーション (DSMC) への動的負荷分散機能 OhHelp の実装に着手した。2024 年度の検討では、OhHelp 導入による高速化効果が一定程度認められたと評価できる一方、1 プロセスが担当するセル数が非常に小さくなる 1 次元計算モデルにおいて、粒子負荷のプロセスへの割り付けに制約が生じたことで、必ずしも最適な負荷均衡が実現しない状況も確認された。

5. 今年度の研究成果の詳細

課題申請時に設定した計画に対応した研究項目とその成果を記載する。

(1) 2 次元モンテカルロ直接シミュレーション (DSMC) への動的負荷分散機構の実装 (申請書内研究計画項目 1a)

惑星上層大気過程向けに開発されたモンテカルロ直接シミュレーション (Direct Simulation Monte Carlo: DSMC) [Terada+, 2016] に関して、並列実行のために領域分割を実施した際に生じる粒子密度の非一様性に伴う負荷不均衡を是正するため、動的負荷分散手法 OhHelp を実装した。OhHelp は各

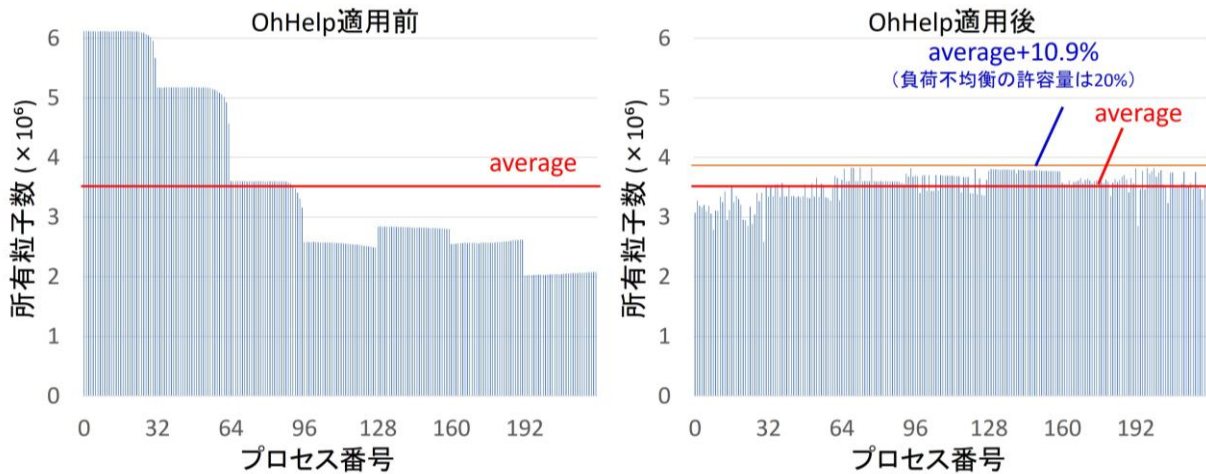


図 1. OhHelp 適用前（左図）と適用後（右図）の 2 次元 DSMC 計算におけるプロセス毎の粒子演算負荷ヒストグラム。「average」は「総粒子数／プロセス数」で計算した理想的な負荷均衡条件を示す。本テスト計算では、許容する最大の不均衡度を平均の 20%と規定しており、当該設定を満足する負荷均衡が実現されている。

プロセスが本来の担当領域以外に、粒子数が多い他の小領域の計算を手助けする手法である [Nakashima+, 2009]。今年度上半期の取り組みとして、2024 年度に OhHelp を試行的に実装した 1 次元 DSMC コード(以下、DSMC-OhHelp と記述する)を 2 次元計算に対応させるべく拡張実装を推進した。

DSMC-OhHelp の多次元化に際しては、将来の 3 次元計算対応を想定したデータ構造を採用しつつ、2 次元の計算条件を設定した。開発における第一の実装ポイントは OhHelp の拡張レイヤーの導入である。DSMC 法は同一セル内の粒子間で衝突計算を行う特性上、物理セル内の粒子がプロセス間に分散されてしまうとプロセスメモリ空間に閉じた粒子間衝突計算が実装困難となる。これに対し、所属セルに基づいて粒子を分類・集計・ソーティングする拡張レイヤーを実装することで、同一セルの全粒子が必ず単一のプロセスに格納されるよう制御した。

第二に、DSMC コードと OhHelp ライブラリ間でのグリッド規格の対応付け(座標変換)である。DSMC 側では物理的な高度に応じて不均等なグリッド幅を用いるが、

OhHelp 側は整数規格を前提とする。この差異を解消するため、DSMC の所属セル番号を OhHelp が認識可能な整数規格(論理座標)へ変換して渡す仕組みを構築した。また、粒子バッファに対して物理座標と論理座標の両方の構造体要素を含めることで、シミュレーションの本来の動作を損なうことなくライブラリとの整合性を確保した。

O+ピックアップイオンの降り込みによる火星上層大気スパッタリングを考慮した密度じょう乱解析に開発コードを適用した。図 1 に計算終了時点における OhHelp 適用前後の各プロセスの担当粒子数のヒストグラムを示す。本計算実験においては、水平方向に 32、鉛直方向に 7 のプロセスグリッドを割り当てており、高高度領域を担当する老番のプロセスは大気の希薄化に伴い、担当粒子数が小さくなる。OhHelp 適用後は物理セル内の粒子を同一プロセスメモリ空間内に保持した状態で、プロセス間の負荷分散が実現された。結果的に物理条件を踏まえた本計算実験条件において、OhHelp 導入前との比較で 2 倍以上の高速化を実現した。粒子密度分布に

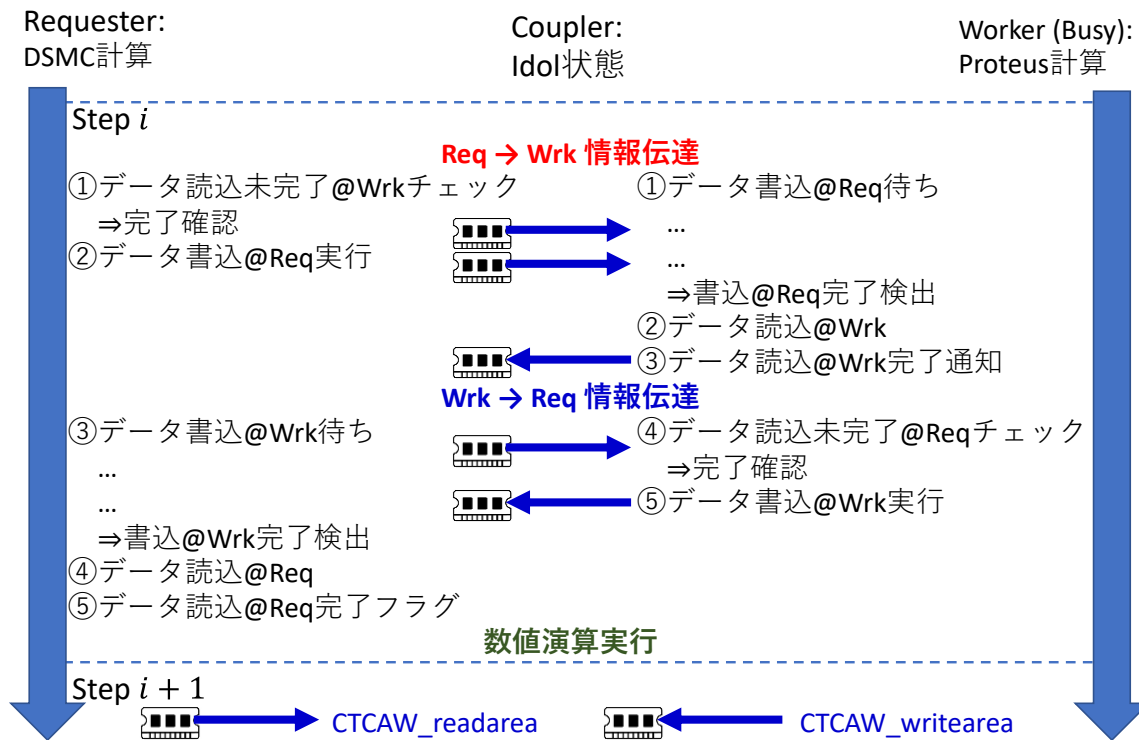


図 2. CoToCoA のリモートメモリアクセス API を活用した相互物理情報伝達の実装手順.

応じた適切な負荷分散効果に加え、粒子ソーティングに伴うメモリ参照局所性も計算高速化に寄与した結果と考えられる。

(2) 惑星上層大気 DSMC シミュレーションと汎用光化学モデル PROTEUS の連成結合物理モデルの構築 (申請書内研究計画項目 1b)

既存の DSMC コードには、数種の光化学反応に基づく粒子生成・消失過程が実装されているが、当該処理を標準化された汎用モデル PROTEUS [Nakamura+, 2023] に換装することにより、より多様な反応過程を網羅的に取り込むことに加え、PROTEUS を活用する他の物理モデルとの連携も柔軟に実施可能となる。このような連成モデルの開発に CoToCoA を活用し、DSMC-PROTEUS 結合コードのプロトタイプを開発した。初期段階の取り組みとして、PROTEUS の光化学反応に基づく粒子生成率・消失率演算結果を、DSMC 側から適時参照する実装を完了した。

当該実装においては、CoToCoA が提供する非同期通信処理によるリモートメモリアクセス API を基盤として、DSMC 計算側か

ら中性粒子密度情報を PROTEUS に送信し、その密度情報を元に算出した特定の化学種の生成率を PROTEUS 側から DSMC に返却することで、双方向の物理情報伝達を実現した (図 2)。このとき、DSMC 計算側は空間 2 次元の自由度を持つのに対し、使用した PROTEUS 計算カーネルは高度方向の空間 1 次元の自由度となっているため、DSMC の水平方向の各座標における高度方向 1 次元空間を PROTEUS と対応づけるアプローチを採用した。

当実装が双方のコードの計算カーネルの変更箇所を最小限に抑え、省工程で実現されたことが特筆すべき点である。

(3) CoToCoA フレームワークにおける Python 対応と連成解析実装 (申請書内研究計画項目 2)

初期の CoToCoA 開発においては、結合対象のプログラムは C および Fortran に記述されたものに限定されていた。近年その重要性が増している Python を基盤としたデータ解析・AI エコシステムと、既存の科学技術計算プログラムの直接結合や連成を目的に、

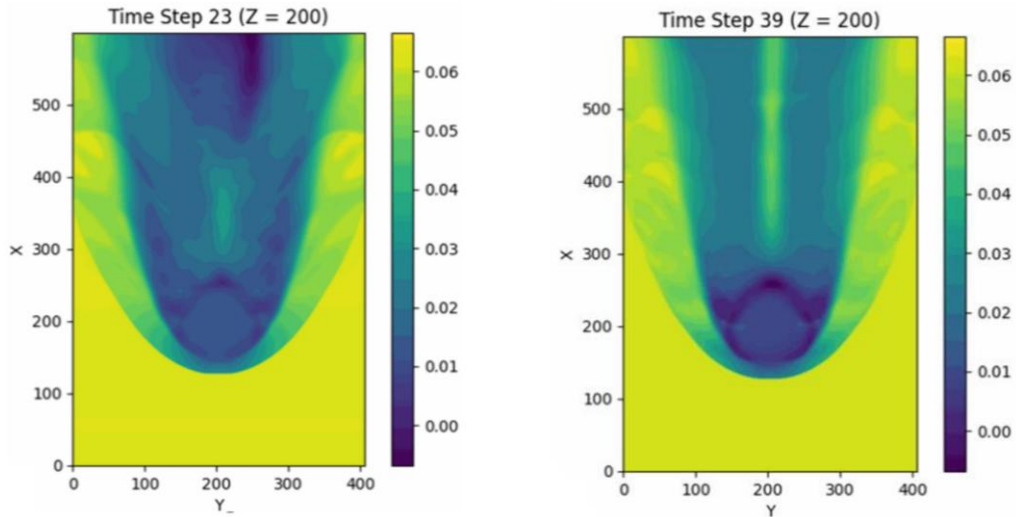


図 3. CoToCoA4Python を用いた磁気圏 MHD シミュレーションの In-situ 可視化結果例.

CoToCoA の Python インターフェースの開発を進めてきた。

当該インターフェースは、Python プログラムを CoToCoA フレームワークにおける Worker として機能させること前提に、以下の二つの手法により開発された。

第一の手法は `mpi4py` を用いた直接実装である。これは MPI 標準の Python バインディングを利用して通信機構を構築したもので、内部の配列操作に NumPy を採用することで、C や Fortran と互換性のあるデータ型を維持したまま高速なデータ処理を可能にしている。

第二の手法は `ctypes` ライブラリを利用して、C 言語で実装された既存の CoToCoA 関数を呼び出す方式である。具体的な手順として、まず Worker 機能を持つ C コードを共有ライブラリ形式（.so 等）でコンパイルし、Python 側からロードする。次に関数の引数と戻り値の型を厳密に定義することで、Python から C の機能をネイティブメソッドのように直接実行可能にしている。この手法は本体コードの大規模な改変が不要であり、C 言語ネイティブの実装に匹敵する高い通信性能を維持できるのが特徴である。

これら二方式の実装は共通のインターフェースクラス `CTCAWorker` としてカプセル

化されており、初期化、計算要求のポーリング、片方向通信といったメソッド群を提供している。この体系的な開発により、計算効率を損なうことなく、Python の広範な可視化・AI エコシステムとのシームレスな統合が達成された。

開発したインターフェースの通信効率を検証するため、京都大学スーパーコンピュータにて性能評価を実施した。計算要求のポーリングの性能に関して、`ctypes` 方式は、データサイズが大きくなるにつれて C 言語によるネイティブ実装に極めて近い性能を示した。特筆すべき点として、小規模データ（1～4MB）では `ctypes` 方式が C 言語実装を上回る実行速度を記録したが、これは C 言語側がより早く待機状態に入ることによるものと分析される。

MPI の一対多通信（RMA）を利用する片方向通信関数では、`ctypes` 方式がデータサイズ 16MB において C 言語実装と遜色ない速度を達成した。一方、`mpi4py` 方式は抽象化レイヤーに伴うオーバーヘッドにより、データサイズの増大に従って実行時間が延伸する傾向が見られた。

これらの評価から、Python インターフェースの導入による通信速度の差は、Coupler

が持つリクエストキューやノンブロッキング通信によって適切に隠蔽され、システム全体の運用に致命的な影響を与えないことが実証された。

実用性の検証として、惑星磁気圏の磁気流体力学 (MHD) シミュレーションにおける In-situ 可視化に適用した。本事例では、Fortran で記述された MHD コード (Requester) が計算を行い、Python プログラム (Worker) が Matplotlib ライブラリを用いて結果を可視化する構成をとった。

この適用により、以下の劇的なリソース削減が達成された。まずはストレージの節約である。対象としたグリッドサイズ (600×407×400) のデータは 1 回につき約 6 GB に達する。全 268,800 ステップをすべてディスクに出力した場合、約 1,600 TB という現実的ではないストレージ容量を要するが、本インターフェースによる通信により、この問題を完全に解消した。次に実行時間の短縮に関して検証を実施した。同ステップ数の I/O 処理のみで合計約 6 時間を要すると試算されるが、メモリ間通信の採用により、この I/O 待ち時間を排除することに成功した。

6. 進捗状況の自己評価と今後の展望

(1) 2 次元 DSMC シミュレーションへの動的負荷分散機構実装、および (2) DSMC 計算と汎用光化学モデル PROTEUS の連成結合物理モデルの構築：進捗率 85%

DSMC 法に基づく 2 次元粒子シミュレーションへの OhHelp 動的負荷分散技法の実装により、高度方向に著しい粗密構造を有する惑星上層希薄大気に起因する負荷不均衡の課題を解決に導いた。この成果は単に計算性能の向上に資するのみならず、計算実行前に予期していない粒子の集中とそれに伴うプロセスメモリ空間の不足による計算異常終了を防ぐ上で意義がある。今後、3 次元モデルや大規模計算への発展が期待される。

並列計算技法の観点では、元来 Particle-in-Cell 法に基づくプラズマ粒子シミュレーションを対象に開発された OhHelp を異種物理モデルに展開応用した点で意義が大きい。計算体系の類似性と相違点を整理した上で必要な適合化を成功させた事例であり、アルゴリズムの持つポテンシャルを活かしたさらなる汎用利用が視野に入る。

また CoToCoA を活用することで、異種物理モデルの連成結合計算の基盤の開発も推進した。両者は過去に独立に開発が進められてきたが、結合方式にプロセス間のデータ通信規格である MPI を活用することで、計算カーネルと連携実装が独立されているため、各種変数・手続きの名前空間が自然に分離される。連成物理モデルを大規模チームで開発するための標準アプローチとしての可能性を示す成果である。

(3) CoToCoA フレームワークにおける Python 対応と連成解析実装：進捗率 90%

太陽地球系物理分野における連成物理シミュレーションを効率化するため、既存の CoToCoA ライブラリへ新たに Python インターフェースを導入し、その有効性を示した。本研究により、C や Fortran の計算効率と Python の解析利便性を両立する枠組みが構築された。本インターフェースは単なる可視化に留まらず、シミュレーション中のデータを直接 AI モデルへ渡してリアルタイム学習や異常検知、将来予測を行うといった、より高度なデータ駆動型科学への応用が期待される。中間ファイルを介さない高度な連携は、今後の大規模並列シミュレーションにおけるデータ処理の標準的な手法となる可能性を秘めている。

両開発項目とも、宇宙惑星物理の実問題への適用を通してその優位性を示すことが今後の課題であり、その点を考慮して進捗率を 85~90% と自己評価している。