

データ駆動科学ソフトウェア・表面解析シミュレーションの高速化技術研究

工藤周平（電気通信大学）

概要

2次元物質とは原子数層の厚みしかない結晶表面構造であり、結晶そのものとは異なる機能を有するため、次世代電子デバイスや触媒開発などへの応用が期待されている。TRHEPDは陽電子線を結晶表面に当てた反射回折パターンを測定する技術であり、表面感度の高く、結晶表面構造解析に適しているが、実験データから直接結晶構造を逆算することが困難という問題があった。そこで実験データと計算モデルによるシミュレーションとを組み合わせ、データ駆動科学の手法を用いた結晶構造探索を行うことが検討されている。本研究では、計算手法の詳細を知らないユーザーであっても、簡単かつ即座に信頼に足る統計解析結果を得られるシステムを構築することを目的に、次の2項目に取り組む。1) 結晶構造探索が多数の候補結晶構造に対するシミュレーションを同時並列に実行するものとなっていることに着目し、シミュレーションの高スループット化による全体高速化に取り組む。具体的には、結晶表面回折シミュレーションアプリ sim-trhepd-rheed に対し、バッチ型計算のためのインターフェースを設計、また GPU などの並列性能の高い計算機を活用するための設計変更を行う。2) データ駆動科学ソフトウェア 2DMAT (ODAT-SE へと改称されたが本レポートでは旧名称を用いる) へ AutoTuning 技術の適用を行う。とくに、2DMAT が持つデータ解析アルゴリズムの一つ PAMC 法を対象に、実行時間の外挿的予測と信頼性尺度の構築を行い、AutoTuning に活用することで、統計精度を確保しつつ高速な探索を実現する。

1 共同研究に関する情報

1.1 共同研究を実施した拠点名

- 北海道大学 情報基盤センター
- 東京大学 情報基盤センター

1.2 課題分野

- 大規模計算科学課題分野

1.3 参加研究者の役割分担

工藤グループ: 結晶表面解析高スループット化手法研究開発 工藤周平・長井元輝

星グループ: 2DMAT AutoTuning 技術研究開

発 星建夫・寺地雄真・須田将成・中野陽斗・谷田秀哉

2 研究の目的と意義

■研究テーマ 1 Reflection High-Energy Electron Diffraction (RHEED) と Total-Reflection High-Energy Positron Diffraction (TRHEPD, 図 1) は 2次元物質向け構造解析手法として注目されている観測技術である。2次元物質とは原子数層の厚みしかない結晶表面構造であり、結晶そのものとは異なる機能を有するため、次世代電子デバイ

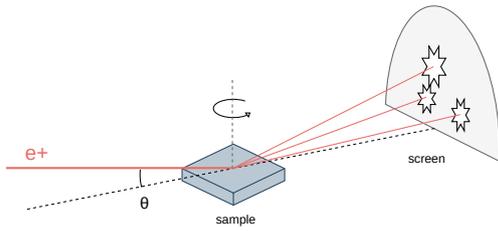


図1 TRHEPDによる表面構造解析の概略

スや触媒開発などへの応用が期待されている。RHEED/TRHEPDは非常に浅い角度から(陽)電子線を照射し、その反射波を観測することで、結晶表面付近に限定した情報を取得することが可能となる。

しかしながらこのような実験観測データから結晶構造のような潜在変数を計算する問題はそれ自体が困難な問題である。そこで実験観測データと計算モデルによるシミュレーション結果を突き合わせ、データ科学の手法を用いて解析を行う、データ駆動科学を取り入れることが進められている。副代表者らが開発したデータ解析フレームワーク2DMATは実験観測データとシミュレーションアプリを結合し、多数のデータ解析アルゴリズムを適用するシステムである。2DMATは解析の手間を減らすだけでなく、観測データからより多くの情報を得ることで、RHEED/TRHEPDに限らず様々な実験観測技術の潜在能力を引き出すことが可能とする。

データ駆動科学に基づくデータ解析は、簡単に言えば、(結晶構造などの)潜在変数 X と観測データ $D = f(X)$ に対し、解候補 $X^{(1)}, X^{(2)}, \dots, X^{(N)}$ の中から最も D に近いシミュレーション結果を出力するもの $D_{\text{cal}} = f_{\text{sim}}(X^{(i)})$ を選択する逆解析型問題である。2DMATが実装するデータ解析手法

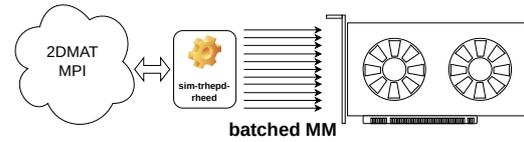


図2 2DMATとシミュレーションソフトウェアの連携による高スループット計算の実現

Particle Annealing Monte Carlo法(PAMC法)は多数の解候補を一度に生成し、同時並列にシミュレーション可能なため、個々のシミュレーション速度よりも単位時間あたりシミュレーション実行数が重要である。すなわち高スループット計算への転換が必要となっている。

本テーマの目的は、メニーコアCPUやGPUなどの主流なHPC環境を活用する高スループット計算に対応するための2DMATの再設計を行うことと、RHEED/TRHEPDシミュレーションアプリsim-trhepd-rhed(STR)の高スループット化を実現することである。(図2)

STRのアルゴリズムでは、逆格子数と呼ばれる変数の数 n に対し、 $2n \times n$ 行列についての行列ODEを計算する。そのため計算の大部分は行列計算(とくに行列積)となっており、近年のCPUやGPUの性能を活かしやすい。ただし n の増加に対して演算量が急増するため、 $n = O(10)$ や、極端には $n = 1$ などの小さな値を用いた計算が実用上重要である。このような小さな n において、1つ1つのシミュレーションの高速化には限界があるため、多数の解候補に対して同時にシミュレーションを行うバッチ化が、高速化に対して効果的であると考えられる。一方、これまで実用上は使われてこなかった大きな n に対しての計算においても、高速化によって現実的なものとなれば、シ

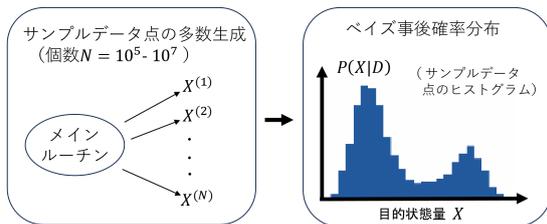


図3 ベイズ事後確率分布計算の概略

ミュレーションの高精細化に貢献できると考えられる。

これらの改良は直接、RHEED/TRHEPD データ解析に役立つだけでなく、2DMAT と組み合わせ可能な他のアプリの性能改善にも影響を与える。

■研究テーマ2 本テーマでは、2DMAT を対象として、AutoTuning 技術を開発する。2DMAT のようなデータ駆動科学計算では「信頼に足る統計解析結果を得る」ことが目標であり、できるだけ少ない実行時間で目標達成することが AutoTuning 技術の問題設定となる。とくに計算手法の詳細を知らないユーザーでも高速計算を容易に実現できることが求められている。2DMAT には、多数のデータ解析アルゴリズムが実装されているが、本テーマでは PAMC 法によるベイズ推定（ベイズ事後確率分布計算、図3）を対象とする。

第1課題は実行時間の予測である。実行時間予測は実測時間を学習データとした性能モデルパラメータ推定問題と理解できる。とくに小規模実行のデータから大規模実行の実行時間を予測する外挿的予測が重要な目標である。第2課題は、統計精度を自動的に確保するための信頼性尺度である。星は PAMC 法に即した信頼性尺度の提案を行っており（星他、日本応用数理学会、2022年9月）、これに基づき動的 AutoTuning 技術を実現する。さらに発展的課題として、精度制御型計算の実現である。順問

題計算の時・空間刻み幅や、演算精度（単精度・半精度）を制御し順問題計算の数値精度を意図的にさげることで高速化し、同じ実行時間でより多くのサンプルデータ点を生成する（統計精度を向上させる）ことである。

※ 本課題は新規課題であるが、今後継続課題として発展させていく予定である。

3 当拠点公募型研究として実施した意義

本研究は、実験科学におけるニーズに、高性能計算の技術で応えるため、データ駆動科学の方法で両者を結びつける共同研究の一端を担うものである。本研究のアプリの対象である TRHEPD 実験は日本独自の先端観測技術であり、その発展には大きな期待が寄せられており、シミュレーションの高速化は実験技術全体の高度化に必要不可欠なものであった。本公募型研究として本課題を実施したことにより、先端計測技術と先端高性能計算技術の橋渡しを行うことができた。

4 前年度までに得られた研究成果の概要

該当なし

5 今年度の研究成果の詳細

■STR アプリ高スループット化実現 本テーマでは研究代表らが提案した高速 STR 計算手法を基に、2DMAT・STR アプリ間の連携強化による高速化手法について検討した。とくに、PAMC 法の高並列性を活用し、複数解候補に対する順問題計算のバッチ型計算を行うことで、2DMAT・アプリ間通信のオーバーヘッドやアプリ起動コストを削減した。

はじめに、Ge(001) の表面構造解析を 2DMAT の PAMC アルゴリズムを用いて解

表 1 各種 STR 計算手法の概要

手法	計算アルゴリズム	並列化	アプリ起動コストの削減
new batch	高速 STR	バッチによる解候補並列	あり
new	高速 STR	アプリ内並列のみ	あり
new flatmpi	高速 STR	Flat MPI による解候補並列	あり
cso flatmpi	従来	Flat MPI による解候補並列	あり
orig flatmpi	従来	Flat MPI による解候補並列	なし
orig	従来	アプリ内並列のみ	なし

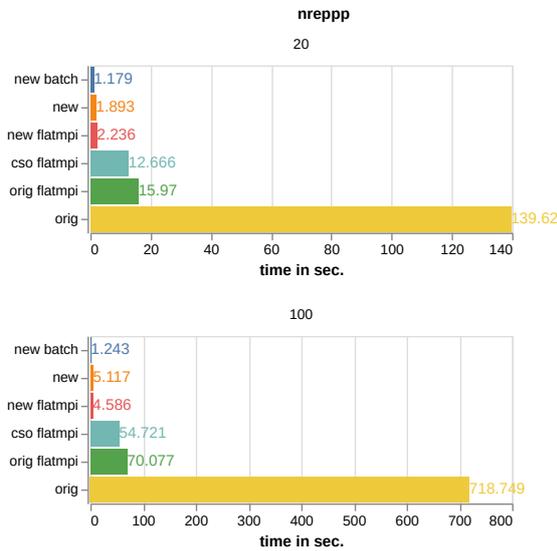


図 4 各種 STR 計算手法の実行時間比較. 1 プロセスあたりの解候補計算数は上段が 20, 下段は 100.

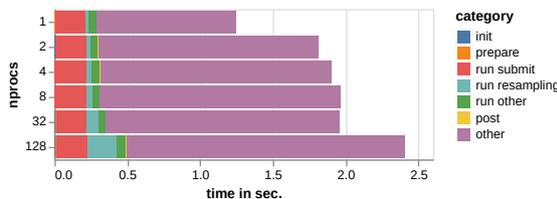


図 5 新規バッチ型計算手法のウィークスケーリング性能.

析するテスト問題に対する実行時間を示す. 本テスト問題は逆格子数 $n = 1$ という小規模問題であり, アプリの計算効率向上が難しいだけでなく, アプリ起動コストなどのオーバーヘッドが計算時間に大きな影響を及ぼすものとなっている. 図 4 は Grand Chariot の 1CPU 上で 2DMAT を用いテスト問題の構造解析を行ったときの実行時間を示している. ここでは従来アルゴリズムまたは高速 STR 計算手法を用いるかや, バッチ化の有無, 並列化の構成によって 6 種類の計算手法を比較している. 計算手法の詳細は表 1 に示す. 図より, 高速 STR 計算手法により計算の大幅な高速化が実現できていることがわかる. また, 高速 STR 計算手法を用いた場合は, 単に Flat MPI によって並列化した場合と比べて, バッチ化を行うことでさらに高速となっている. また new batch は計算する解候補数を増大させても計算時間がほとんど変化していない. これは, 2DMAT 自体の (python モジュールの読み込み等の) 起動コストが計算時間の中に入っているからである. そこで長いステップ数の計算においては, 実行時間の比率はより大きくなるものと考えられる.

図 5 は新規実装のウィークスケーリング性能を示している. この図では, Grand Chariot 上で, 1CPU あたり 1MPI プロセス, 1 プロセスあたりの解候補数を 100 と固定し, MPI プ

ロセス数を増大させたときの実行時間の内訳を示している。実行時間の大部分を占めているのは other の部分であり、主に python モジュールの読み込みなど、2DMAT の起動コストが含まれている。STR アプリの計算時間は run submit に相当し、理想的な並列性能を示している。一方、run resampling の実行時間は並列度の増大に従い増大している。これは 2DMAT 内における PAMC 法の計算時間であり、MPI 通信が多く含まれる部分である。これより、実用上重要な $n = 1$ の小規模問題においては十分な高速化が実現できており、今後は 2DMAT 自体の高速化が必要になってくると考えられる。

また、高並列性を活かした STR アプリの GPU 実装に取り組んだ。ここでは $n > 1$ のテスト問題として Si(111) 構造の回折シミュレーションを $n = 23$ と $n = 521$ の問題に対して行った。GPU 実装においては逆格子数 $n > 1$ の問題について、cuBLAS を用いた行列計算の GPU 化と、バッチ型行列計算ルーチンを用いた高速化を試みた。また、CUDA Stream を用いたホスト・デバイス間通信時間とホスト側計算時間の隠蔽を行っている。また Stream 数 n_{strm} とホスト側スレッド数を一致させることで、ホスト側計算の並列化を行っている。さらに 1 Stream あたり n_{xtal} 個の解候補を同時に計算させることで演算密度の向上を図っている。以上により、1 度に $n_{\text{strm}} \times n_{\text{xtal}}$ 個の解候補を同時計算するバッチ型計算となっている。

図 6 に $n = 23$ の問題に対する GPU 化 STR アプリの実行時間とその内訳を示す。実行時間は東大 Wisteria-a の NVIDIA A100 を用いたときの、8,192 個の解候補を計算するのに要した時間となっている。図より、適切な Stream 数と解候補数を設定した場合、Wisteria-a の

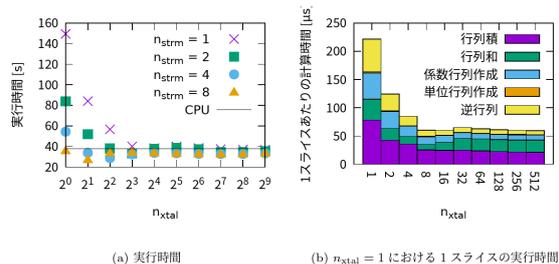


図 6 $n = 23$ における GPU 向け STR アプリの実行時間とその内訳。

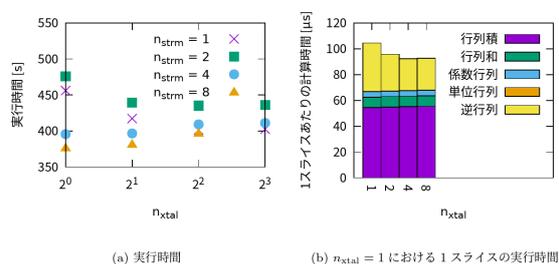


図 7 $n = 521$ における GPU 向け STR アプリの実行時間とその内訳。

ホスト側 CPU (Intel Xeon 8360Y 2 基) よりも高速に計算が可能となっており、最大で 1.4 倍高速となっている。計算時間はバッチサイズが大きくなるほど高速になる傾向があるが、 $n_{\text{strm}} \times n_{\text{xtal}} = 16$ となる特定のサイズでとくに高速となっている。また計算時間の内訳 (b) より、バッチサイズの増大に伴い行列積計算は高速化しているが、行列和計算はあるところから増大に転じている。これは計算データが GPU の L2 キャッシュに収まる程度になっているかどうかの原因であると考えられる。そこで $n = 23$ 程度のサイズの問題のさらなる高速化においては、バッチ型計算に加えて、データ再利用性を高めるような計算手順の再構築が必要であると考えられる。

また図 7 に $n = 521$ の大規模問題に対する GPU 化 STR アプリの実行時間とその内訳を示す。大規模問題に対してはバッチ化の効果は

小さいが、Stream 数の増加による高速化の効果が確認できる。また実行時間の内訳では行列積が計算の大部分を占めており、この部分においては GPU の理論性能の 8 割を超える性能に達している。また最も高速な設定においては GPU は CPU の約 4 倍の性能となっており、GPU の性能を有効に利用できていることがわかる。

以上の通り、新規高速計算アルゴリズムに加え、バッチ化や GPU 化のような高スループット化により、計算全体の高スループット化を実現した。とくに $n = 23$ の問題において、従来アプリでは 1 つの解候補あたり約 30 秒程度要していたものが、約 3.3ms 程度にまで削減されており、トータルで 1 万倍の高スループット化となっている。また、 $n = 521$ の大規模問題においても、従来アプリでは 1 つの解候補あたり約 2 万秒を超える計算時間となっていたものが、約 5.5 秒にまで削減されており、トータルで 1,000 倍以上の高スループット化が実現できている。

■自動チューニング技術による 2DMAT PAMC 法の改良とその検証 PAMC 法のパラメータ (レプリカ数, 不確かさ強度パラメータ変化率, リサンプリング操作発動条件, などの) 最適化手法の手法開発とテスト計算を行ってきた。レプリカ数については、Gellman-Rubin 型診断条件を PAMC に対して考案し、テスト計算で有用性を示した。不確かさ強度パラメータ変化率については、変化率が大きすぎると、正しい分布が得られない事例がみつき、対処方法の立案とテスト計算を行なった。リサンプリング操作発動条件については、少数レプリカが確率分布上の重みを寡占する事例がみつき、対処方法の立案とテスト計算を行なった。

6 今年度の進捗状況と今後の展望

■テーマ 1: 高スループット化実現 [段階 1] 2DMAT 内部でのスレッド並列化実装: 2DMAT 内において Python プログラムをスレッド並列化することで、多様なアプリの高スループット化を実現することが目標の 1 つである。今年度は Python プログラムのスレッド並列化の難易度とオーバーヘッドについて準備的調査を行った。とくに (a) について、スレッド並列化実現のためのプログラムコストが当初の想定を大きく上回っていることが判明し、また、原理上スレッド並列化の効果が薄い Python 側計算のコストも大きいことが判明した。以上のため、段階 1 を飛ばし、より効果の高い段階 2 を中心に研究を進めた。

[段階 2] RHEED/TRHEPD アプリのバッチ化: アプリ・2DMAT のバッチ化対応とともにアプリの GPU 化についても順調に進んでおり、当初の想定以上の結果を得ている。とくにバッチ化では、アプリ起動オーバーヘッドの大きな問題に対して非バッチ化構成と比べて約 7 倍の高速化を実現しており、また GPU 化においては、演算量の大きな問題に対して GPU 性能の 8 割以上の効率を実現するなど、良好な結果を得ている。今後も継続して GPU 実装の高速化に取り組むとともに、バッチ化手法の他アプリへの展開などへ発展させていく。

■テーマ 2: 2DMAT の AutoTuning の実現 [段階 1] 既存実装を用いた検証: 本年度は STR アプリだけでなく多様なアプリに対する 2DMAT の性能調査と検証を進めてきた。次年度においては、AutoTuning 手法の理論的解析を進め、さらに Hamiltonian Monte Carlo (HMC) 法などの近年の優れた手法を取り入れ、性能検証を行う。また 2DMAT 自体の大規模並列時における性能検証や高速化手法の研

究にも着手する予定である。

[段階 2] 精度制御型計算の実現：発展的課題として挙げていたが、現状では着手できていない。今後、AutoTuning 実装の進展に合わせて、精度制御型計算についての研究を進めていく予定である。