

jh180023-NAH

高性能・変動精度・高信頼性数値解析手法とその応用

中島研吾（東京大学情報基盤センター）

概要

本研究は、最先端のスパコン向けに開発された高性能数値アルゴリズムに対して、半精度から倍精度、倍々精度までの広範囲をカバーする変動精度演算を適用し、精度保証、そのための自動チューニング手法を開発する。開発された手法を様々なアプリケーションに適用することで、低精度を中心とした変動精度演算の科学技術シミュレーションへの有効性を検証する。開発したアルゴリズム、アプリケーションの消費電力の直接測定によって、各計算の特性と低精度演算の有効性を消費電力の観点から検討する。

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

- 東京大学（Oakforest-PACS, Reedbush-U/H/L）
- 東京工業大学（Tsubame-3）

(2) 共同研究分野

■ 超大規模数値計算系応用分野

□ 超大規模データ処理系応用分野

□ 超大容量ネットワーク技術分野

□ 超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

- 中島研吾（東京大学）（代表）全体統括・高性能アルゴリズム
- 市村 強（東京大学）（副代表）高性能アルゴリズム・地震シミュレーション
- 横田理央（東京工業大学）（副代表）高性能アルゴリズム・精度保証
- 岩下武史（北海道大学）高性能アルゴリズム
- 深谷 猛（北海道大学）高性能アルゴリズム
- 埴 敏博（東京大学）性能最適化・電力測定
- 伊田明弘（東京大学）高性能アルゴリズム
- 星野哲也（東京大学）高性能アルゴリズム・性能最適化
- 坂本龍一（東京大学）電力測定・自動チューニング
- 有間英志（東京大学）電力測定・自動チューニング
- 古村孝志（東京大学）地震シミュレーション
- 藤田航平（東京大学）高性能アルゴリズム・地震シミュレーション
- 近藤正章（東京大学）精度保証・電力測定・自動チューニング
- 奥田洋司（東京大学）工学シミュレーション・精度保証
- 森田直樹（東京大学）工学シミュレーション・精度保証
- 荻田武史（東京女子大学）精度保証・自動チューニング
- 尾崎克久（芝浦工業大学）精度保証・自動チューニング
- 片桐孝洋（名古屋大学）高性能アルゴリズム・精度保証・自動チューニング
- 八代 尚（理化学研究所）大気シミュレーション
- 河合直聡（理化学研究所）高性能アルゴリズム・量子科学シミュレーション
- 井上弘士（九州大学）精度保証・電力測定・自動チューニング
- 荒川 隆（RIST）大気シミュレーション
- 成瀬 彰（NVIDIA）精度保証・自動チューニング
- 堀越将司（インテル）精度保証・自動チューニング
- 大島聡史（九州大学）高性能アルゴリズム・性能最適化
- Gerhard Wellein（FAU Erlangen-Nürnberg, Germany）高性能アルゴリズム・量子科学シミュレーション・SELL-C-σ
- Achim Basermann（DLR, Germany）高性能ア

ルゴリズム・量子科学シミュレーション

2. 研究の目的と意義

エクサスケールシステムにおける、高性能数値アルゴリズム実現には通信最適化が必須である。通信には、メモリ上のデータ移動 (Serial 通信)、ノード間ネットワークを介した Parallel 通信がある。メモリ・ネットワークについては、技術革新により大容量化、広帯域化が進む一方で、階層がより深化し、局所化やレイテンシ対応が必要となり、Serial・Parallel 通信最適化のための手法が提案、実装されている。もう一つの技術的課題は消費電力、エネルギー (以下「消費電力」) である。ハードウェア的技術革新と共に、省電力・省エネルギー (以下「省電力」) のためのアルゴリズムの開発、実用化により、実計算時の消費電力の抑制が期待される。Approximate Computing (S. Mittal, A Survey of Techniques for Approximate Computing, ACM CSUR 48-4, 2016) は、低精度演算の積極的活用により計算時間短縮、消費電力削減を図る試みである。混合精度演算はその一種であり、既に多くの研究事例があるが、Approximate Computing では、半精度から四倍精度まで演算精度を動的に変動させる変動精度 (Transprecision) の研究が進められている。数値計算による近似解 (数値解) は様々な計算誤差を含み、計算結果の信頼性の観点から、数値解の正しさを数学的に保証する必要がある、低精度・変動精度使用時、悪条件問題には重要である。昨今は、スパコンによる大規模計算向け精度保証の研究も実施されているが、実問題で現れる大規模疎行列・H 行列 (階層化行列、密行列を低ランク近似等により階層化する手法) 系への応用例はほとんどない。本研究では、Oakforest-PACS (東大, Intel Xeon Phi (Knights Landing, KNL)), Reedbush-H/L (RBH, RBL) (東大, NVIDIA P100), Tsubame-3 (東工大, NVIDIA P100) のという JHPCN システム群の中でも消費電力当たり計算性能 (GFLOPS/W 値) の高いシステムを主たるターゲットとして、以下を実施する:

- 疎行列演算, H 行列演算, ステンシル演算等の代表的数値アルゴリズムについて、各参加者によって開発された Serial・Parallel 通信の最適化に着目した高性能手法 (1.SELL-C- σ 疎行列格納法 (SIMD 演算に適した疎行列格納法), 2.時空間タイリング (ステンシル演算の特性に基づき、通信削減・メモリアクセス最適化を実施する), 3.Pipeline 法 (MPI_Allreduce の通信オーバーヘッドを隠蔽する手法), 4.Dynamic Loop Scheduling (疎行列ベクトル積において OpenMP のマスタースレッドが通信を担当し、通信と計算をオーバーラップさせる手法), 5.hCGA 法 (Hierarchical Coarse Grid Aggregation, 並列多重格子法において段階的にプロセス数を削減し並列オーバーヘッドを削減する手法), 6.Adaptive CG (要素単位多重格子法前処理を適用した共役勾配法), 7.pK-Open-SOL (悪条件問題向け並列前処理ソルバー), 8.HACApK (世界初のオープンソース OpenMP/MPI ハイブリッドによる H 行列ライブラリ)) を各システムにおいて実装、最適化し、低精度演算・変動精度演算についての検討を実施する。
- 各アルゴリズムを各参加者によって開発された地震 (Seism3D, GHYDRA), 大気科学 (NICAM), 量子科学, 工学 (FrontIstr) 等の実シミュレーションに適用し、従来手法と性能・精度を比較するとともに、一部のアルゴリズム, シミュレーションについては計算時の消費電力測定を実施する。
- 疎行列演算や H 行列演算を対象として、特に悪条件問題における実用的な精度保証法を確立する。低精度演算による数値解の精度改善手法である反復改良法 (Iterative Refinement) を変動精度演算により実現し、精度保証法と融合させる。所望の結果精度達成という条件下で、計算時間や消費電力を最小化する最適な演算精度を自動チューニング技術によって動的に制御する手法を確立する。本研究では、単精度+倍精度の変動精度を前提とするが、

一部アルゴリズム, アプリケーションについては, 半精度, 四倍精度についても検討する。

- 本研究によって開発された高性能・変動精度・高信頼性数値解法を東大情報基盤センターで開発中の自動チューニング機構を有するアプリケーション開発・実行環境 ppOpen-HPC 及び pK-Open-HPC (ppOpen-HPC の発展形) (<http://ppopenhpc.cc.u-tokyo.ac.jp/>) に実装し, Oakforest-PACS (OFP), Reedbush (RBU, RBH, RBL), Tsubame-3 (TSB3) 等で公開する。将来的には, 他のセンターのスパコンへの導入も視野に入れる。

本研究は, 最先端のスパコン向けに開発された高性能数値アルゴリズムに対して, 半精度から倍精度, 倍々精度までの広範囲をカバーする変動精度演算を適用し, 精度保証, そのための自動チューニング手法を開発する試みとしては初めてのものであり, 開発された手法を様々なアプリケーションに適用することで, 低精度を中心とした変動精度演算の科学技術シミュレーションへの有効性を検証できる。開発したアルゴリズム, アプリケーションの消費電力の直接測定によって, 各計算の特性と低精度演算の有効性を消費電力の観点から検討可能となる

3. 当拠点公募型共同研究として実施した意義

JHPCN は多様な計算機環境を備え, 実用的なシステムとして最も GFLOPS/W 値の高い Intel Xeon Phi (OFP), NVIDIA P100 (Tsubame-3) による国内最大システムを有し, 本研究の目指す高性能・変動精度・高信頼性数値解法の研究には最適である。東大情報基盤センターの Reedbush-L では「ノード固定」における設定カスタマイズにより, 個別ノードの消費電力測定が可能である。JHPCN は疎行列・H 行列・ステンシル計算, 自動チューニング, 精度保証, 計算科学, 計算機科学等の様々な分野の専門家を擁し, 本研究のような学際的研究を推進する体制を容易に構築でき, 本研究にも北大, 東大, 東工大, 名大, 九大各センターから

様々な分野の研究者が参加している。JHPCN 各センターはオープンソースソフトウェア活用に積極的であり, 本研究の成果を公開, 各センターのスパコンにデプロイし, 講習会等の普及活動を協力して行い, 利用者拡大, ソフトウェアの更なる改良が可能となる。

4. 前年度までに得られた研究成果の概要

(該当せず)

5. 今年度の研究成果の詳細

①ステンシル演算：構造格子

本研究項目では, 三次元地震波動伝搬解析コード「Seism3D」を対象として, 時空間タイリング手法と変動精度演算の適用を検討し, 性能検証・精度評価を実施することが目標である。今年度前半では, まず, Seism3D のミニアプリ版 (シングルノード用の簡略版) に対して, プログラムコードの分析を行い, 時空間タイリング手法の適用に関する検討を行った。ミニアプリ版のプログラムコードを分析した結果, 時空間タイリング手法の適用に関して, 以下の点が明らかになった。

- オリジナルのプログラムコードは, 演算回数の削減とコードの可読性・保守性等の理由から中間変数を格納するための多数の配列を使用しており, このままの状態では時空間タイリング手法の適用は難しい。
- 処理の本質的な部分は, 速度の配列と応力の配列を交互に更新する構造であり, この点においては, 過去の時空間タイリングの研究 (3D FDTD 法に対する時空間タイリング, 深谷・岩下, HPC Asia 2018) と同じ構造である。
- 実アプリ特有の設定 (本課題では, 自由表面境界条件, 無反射境界条件, 非弾性減衰) への対応は, 上記の FDTD 法の研究で今後の課題としていたため, 今回, 新たに検討が必要である。

今年度後半では, 上述の課題を踏まえて, 実際

に時空間タイリング手法の適用を試みた。先行研究として、Seism3D に関して、中間変数の配列を削減することで、処理の演算密度を向上させるアプローチ (F. Mori et al. VECPAR 2014) が試されている。そこで、まずは、時空間タイリング手法の効果を見積もるために、境界条件等の処理を取り除き、同じ構造のステンシルに基づく格子点の更新のみを行うテストコードを作成し、そのコードに対して、時空間タイリング手法を適用した。具体的には、図 1 に示した平行四辺形型とダイヤモンド型のタイル形状を導出し、これを組み合わせることで空間 3 次元の反復型ステンシル計算に適用した。

東大 Reedbush-U の計算ノード (Intel Xeon E5-2695v4×2) を用いて行った性能評価の一例を図 2 に示す。図 2 は N^3 個の格子点に対して、512 時間ステップの計算を、36 スレッドにより行った場合の性能を示している。時空間タイリング手法としては、タイル内で並列化する手法 (Tiling(0D)), タイルレベルの並列性を有する手法 (Tiling(1D, 2D, 3D)) を比較している。図 2 が示すように、時空間タイリング手法で性能が向上していることが確認でき、特に、Tiling(2D)を用いた場合に最も高い効果が得られたことが分かる。この結果は、タイルレベルの並列性を持つ時空間タイリング手法が有効であることを示しており、3D FDTD 法に対する時空間タイリングの事例 (深谷・岩下, HPC Asia 2018) と同様の結果である。一方で、Seism3D では 4 次精度の中心差分を用いており、ある点を更新する際に、2 次精度の中心差分の場合よりも多くの点を必要とする。そのため、タイルの形状が空間方法に広がり、時間方法のタイル幅を大きく設定することが難しいことが、今回の評価により明らかになった。時間方法のタイル幅は、データの再利用性に影響するため、この点については、今後、より詳しい評価をする必要がある。

次年度以降の課題としては、今回作成したコードに境界条件等の設定を加えて、実アプリケーションに近い状況で性能評価を行うことが挙げられる。また、ステンシルの形状やデータ型による、

時空間タイリング手法の効果と最適なタイル幅の違いについて調査することも有益である。

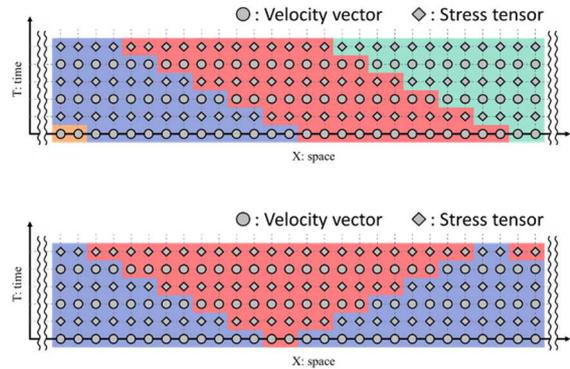


図 1: Seism3D を模した反復型ステンシル計算に対する時空間タイリング手法のタイル形状 (上: 平行四辺形型, 下: ダイヤモンド型)

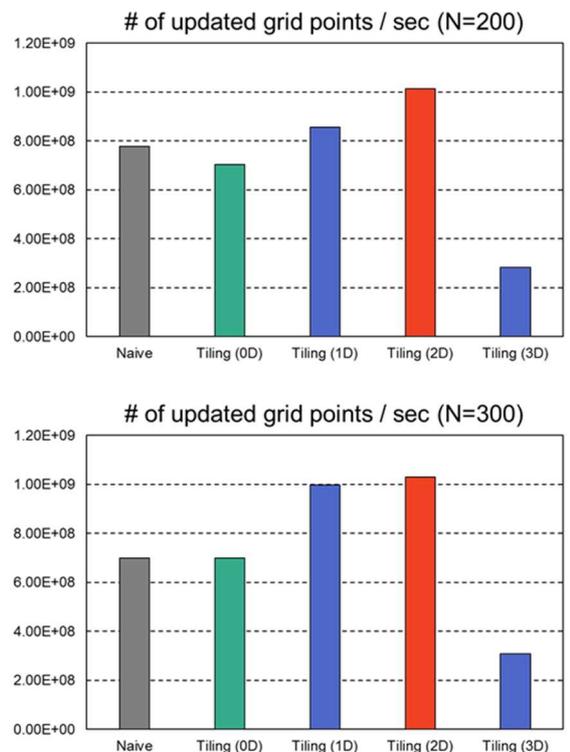


図 2: Seism3D を模した反復型ステンシル計算に対する各種時空間タイリング手法の効果の比較 (Reedbush-U, 1 ノード 36 スレッド使用)

②ステンシル演算: 準構造格子

一般的な構造格子ではないシミュレーションモデルでの評価を行うべく、本研究では正 20 面体全球大気モデル NICAM(Nonhydrostatic ICosahedral Atmospheric Model)を対象とした。NICAM は地球表面を均一に近い水平解像度のメッシュで覆うために、正 20 面体格子系を採用している。ただし、

間接参照を用いた完全な非構造格子ではなく、菱形の領域で格子群をまとめることでプロセス内のステンシル計算を構造格子と同様に扱い、複雑なトポロジーはそれぞれの菱形が持つ袖領域での P2P 通信に担当させる「準構造格子」のスタイルをとっている。これにより、プロセス単位では多くの構造格子ステンシルコードで用いられる最適化を容易に適用することが可能である。しかし、時空間タイリングを実現するアルゴリズムの適用は難しい。そこで本研究では NICAM を用いた混合精度でのシミュレーション性能評価を行った。NICAM の流体力学計算部分（力学コア）は NICAM-DC としてパッケージ化され、過去に OpenACC での最適化と GPU スパコンでの性能評価が行われている。その後単精度実行を可能にする変更を含む大幅なソースコードのリファクタリングが行われたために、GPU 最適化部分は一旦失われた形となった。本研究では更新された最新の NICAM の力学コアについて、OpenACC での GPU 最適化を行った。最適化に際して、ステンシル計算の一部は OpenMP を対象としたループ構造では性能が著しく低下することがわかったため、最適化方法を変えて分岐する方法を採用した。最適化されたソースコードを用いて、東大 Reedbush-L での計算性能の計測を行った。

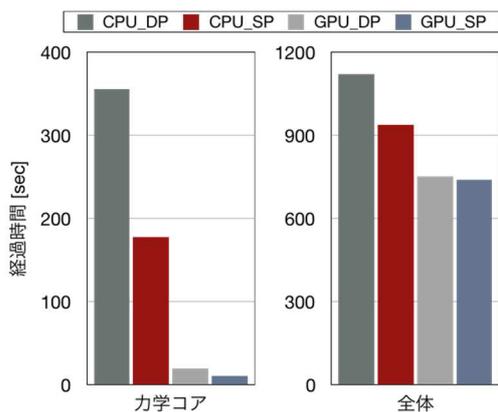


図3 Reedbush-Lを用いたNICAM計算時間評価、縦軸は112km解像度・78層で10ステップ計算した時の力学コア部分(左)と全体(右)の経過時間を表す、左からCPU1コアでの倍精度計算、単精度計算、CPU1コアとGPU1台を組み合わせて用いた場合の倍精度計算、単精度計算の結果

結果を図3に示す。単精度の利用によってCPUで

は1.87倍、GPUでは1.60倍の高速化が得られることがわかった。ただし、現在は力学コア区間のみの評価にとどまっており、全体性能では残りの計算部分に引っ張られる形で大幅な性能向上は得られていない。

次に単精度を用いた際のシミュレーション結果について、水平224kmメッシュ格子を用いた現実大気の1ヶ月シミュレーション結果を倍精度計算結果と比較した。この計算では上で述べたように、流体力学計算部分のみ浮動小数点精度が異なる。図4で示した降水量の月平均値において、精度の異なる二つの実験の降水量は大まかな分布に違いがほとんどなく、全球平均でも0.3%の違いに過ぎなかった。これは単精度化による流体力学計算の精度劣化が非常に小さいことを示すものである。加えて、丸め誤差から成長し徐々に一致なくなると考えられる雲と降水の分布についても、月平均値では十分に小さく、科学的な結果に影響を与えるものではないということが示唆された。

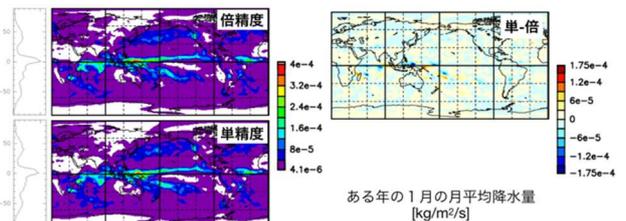


図4 NICAMを用いて224km, 78層で1ヶ月(3600ステップ)計算した時の全球降水量の分布、それぞれ倍精度(左上), 単精度(左下), 単精度-倍精度(右上)の結果を表す

地域的な降水分布の偏りは、より長期間のシミュレーションで増幅し、系統的かつ明瞭な違いをもたらす可能性がある。雲微物理や大気放射計算などの物理諸過程の単精度化とGPU最適化を含め、今後の研究においてさらなる評価が必要である。

③疎行列演算：熱伝導方程式

(1) 単精度・倍精度演算の比較

本節では、図5に示す差分格子によってメッシュ分割された三次元領域において、以下の定常熱伝導方程式を解くアプリケーションを対象とし、

導出された対称正定な疎行列を係数行列とする大規模連立一次方程式を ICCG 法によって解く。

熱伝導率 λ の分布は図 6 に示すように、一層のみ $\lambda=\lambda_2$ 、他の部分は $\lambda=\lambda_1$ とする。本節では $\lambda_2 \leq \lambda_1$ とし、 λ_1/λ_2 の比を様々に変化させた場合の計算を実施した。

$$\frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) = f$$

$$\phi = 0 @ z = z_{\max}$$

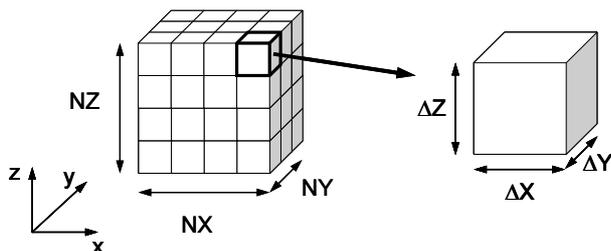


図 5 三次元ポアソン方程式ソルバーの解析対象差分格子の各メッシュは直方体 (辺長さは ΔX , ΔY , ΔZ), X, Y, Z 各方向のメッシュ数は NX, NY, NZ

形状は規則正しい差分格子であるが、プログラムの中では、一般性を持たせるために、有限体積法に基づき、非構造格子型のデータとして考慮する。

まず、 λ_1/λ_2 を変化させ、ICCG 法を倍精度、単精度で実施し、疎行列格納法として CRS (Compressed Row Storage) を適用した場合について、番号付けを Coalesced, Sequential とした場合の計算を実施した。問題サイズは $NX=NY=NZ=128$ とし、CM-RCM の色数=20 として東大 Reedbush-U (Intel Xeon/Broadwell-EP) の 1 ソケットを使用した。図 7 は、倍精度演算時の反復回数を 1 とした場合の単精度演算時の反復回数の比である。 λ_1/λ_2 (条件数に比例) が大きくなるに従って単精度演算時の反復回数が相対に大きくなっていることがわかる。

図 8 は倍精度・Sequential の計算時間を 1.00 とした場合の計算時間比である。色数が比較的少ないため Coalesced と Sequential の差は少ないが、Sequential の場合が概ね安定である。単精度演算の反復回数が安定している $\lambda_1/\lambda_2=10^4$ までは、計算時間の単精度/倍精度比は 0.60 程度であるが、それ

より大きい場合は、単精度演算の反復回数が増加して、倍精度演算の場合より計算時間が増えている。 $\lambda_1/\lambda_2=10^6$ では計算の誤差も大きく、正しい解から大きな差がある。

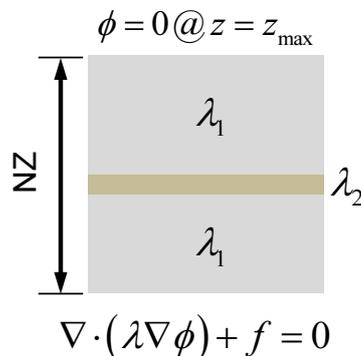


図 6 熱伝導率 λ の分布 ($\lambda_2 \leq \lambda_1$)

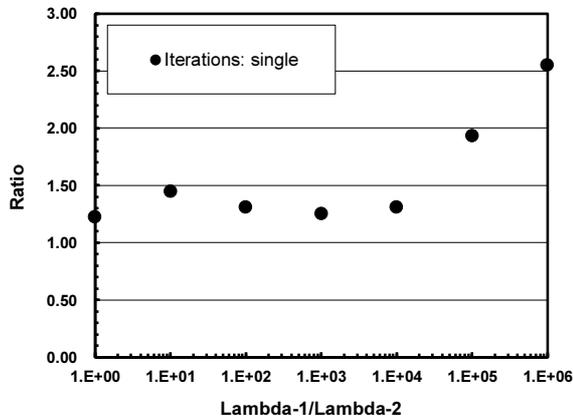


図 7 単精度/倍精度演算時の反復回数比と λ_1/λ_2 の関係 ($NX=NY=NZ=128$)

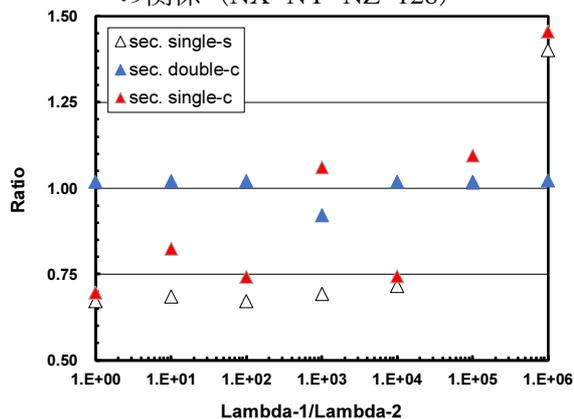


図 8 単精度/倍精度演算時の計算時間比と λ_1/λ_2 の関係 (倍精度・Sequential の場合を 1.00 とする) (Δ : single-s (単精度・Sequential), \blacktriangle : double-c (倍精度・Coalesced), \blacktriangle : double-c (倍精度・Coalesced)) ($NX=NY=NZ=128$)

(2) 半精度実数の適用

図 9 は図 6 に示す問題の右辺，Z 方向メッシュ数などを変更した場合について，NVIDIA V100 上で単精度演算（前処理のみ半精度）の収束解が得られるパラメータの範囲を示している。倍精度，単精度（前処理含む）の場合は全て収束解が得られているが，前処理のみ半精度となった場合には解を得られる範囲が極端に少なくなっていることがわかる。

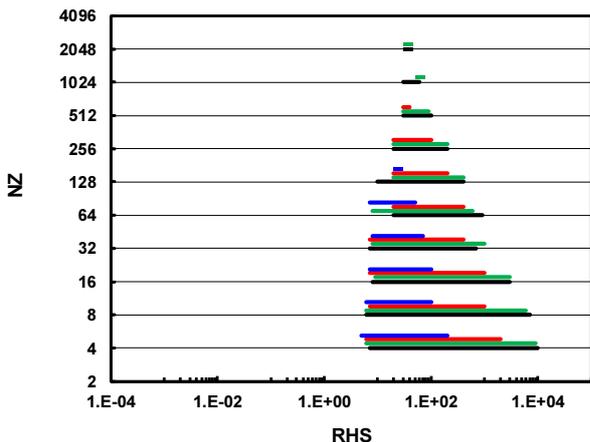


図 9 V100 上で単精度演算（前処理部分のみ半精度）を適用した場合の収束解が得られるパラメータの範囲（RHS：右辺，NZ：Z 方向メッシュ数， $\lambda_1/\lambda_2=10^0, 10^1, 10^2, 10^3$ ）

(3) SELL-C- σ の実装

続いて，(1) と同じ計算について，疎行列格納法として ELL 法，SELL-C- σ 法（図 10）を適用した場合の計算を Oakforest-PACS (OFP, Intel Xeon Phi (Knights Landing)) 1 ノードを使用

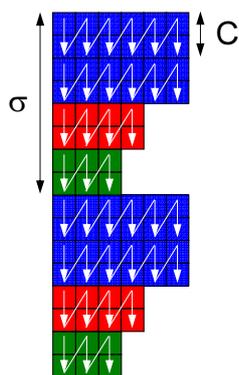


図 10 SELL-C- σ 形式， $C=2$ and $\sigma=8$

して実施した（64 コア，128 スレッド）。CM-RCMの色数は10と

した。図 11 は各ケースの ICCG 法部分の計算時間である。CRS \Rightarrow ELL で 2 倍以上，ELL \Rightarrow SELL-C- σ で 15-25%程度の計算時間短縮が得られている。更に倍精度 \Rightarrow 単精度で 30%程度の短縮が達成されている。本研究では， $\sigma=1$ とする SELL-C-1 を適用し

ている。C の最適値は倍精度で 8-16，単精度で 16 となっている。単精度での計算速度向上率は (1) で示した Reedbush-U (Intel Xeon/Broadwell-EP) と比べて低く，SIMD 化が十分でないと考えられる。

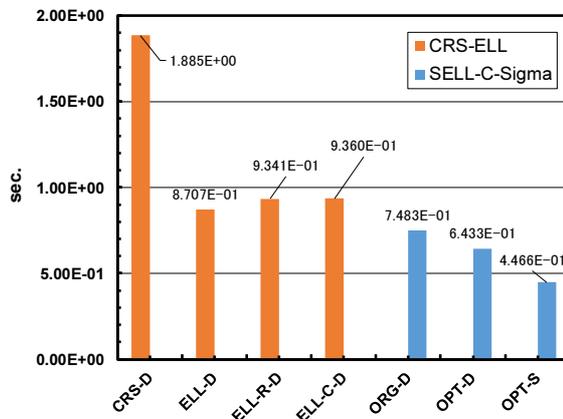


図 11 OFP 上での計算時間（64 コア，128 スレッド）（CRS-D：CRS・倍精度，ELL-D:ELL・倍精度，ELL-R-D：同行方向ブロック化・倍精度，ELL-C-D:同列方向ブロック化・倍精度，ORG-D：SELL-C- σ 最適ケース・倍精度，OPT-D：SELL-C- σ 最適ケース・倍精度（星野他，情報処理学会研究報告（2017-HPC-158-18）），OPT-S：同単精度）（ $NX=NY=NZ=128$ ）

(3) Pipeline 型手法の安定性に関する検討

大規模な並列計算機を使用する場合，ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが，通信をできる限り効率的に実施し，削減することは EFLOPS 級システムにおいて重要である。共役勾配法（Conjugate Gradient, CG）に代表されるクリロフ部分空間法が広く使用されている。同手法では内積の計算が現れるが，全 MPI プロセスによる通信を必要とするため，計算量に比べて通信のオーバーヘッドの割合が大きくなる。Pipelined CG 法では，CG 法の結果が変わらないように計算順序を変更し，MPI-3 でサポートされている非同期集団通信（MPI_Iallreduce）によって，集団通信のオーバーヘッドを隠蔽することに成功している（P. Ghysels et al., Parallel Computing 40-7, 224-238, 2014）。

本研究では，[P. Ghysels et al., 2014] で紹介されている以下の 4 つのアルゴリズムについて，(1) (2) で述べたのと同じ計算について，単精度・倍

精度演算を適用した：

- Alg.1：オリジナルの ICCG 法
- Alg.2：Chronopoulos/Gear アルゴリズムによる ICCG 法
- Alg.3：Pipelined CG 法
- Alg.4：Gropp アルゴリズムによる CG 法

4 つのアルゴリズムは漸化式によって計算順序が異なっているが、同じ計算を実施している。ただ、計算順序の変更により丸め誤差の伝播が異なるため、悪条件問題において収束履歴が変化する可能性がある。

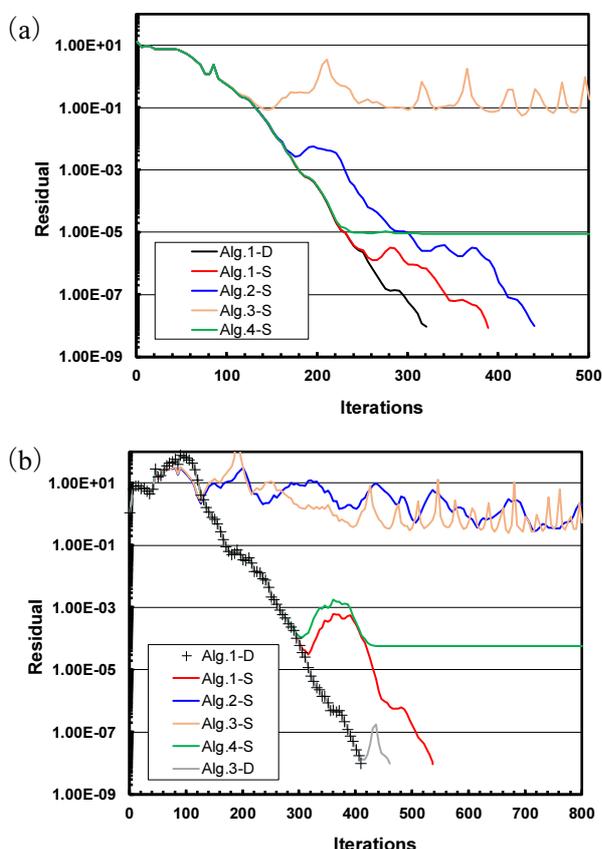


図 12 各アルゴリズムの収束履歴 (NX=NY=NZ=128), -D：倍精度演算, -S：単精度演算, (a) $\lambda_1/\lambda_2=10^0$, (b) $\lambda_1/\lambda_2=10^4$

本研究では、これらのアルゴリズムの単精度演算適用時の安定性を検討するため、(1)、(2)と同様の 1 ノードにおける計算を Reedbush-U 1 ノード (36 コア) を使用して実施した。図 12 は $\lambda_1/\lambda_2=10^0$, 10^4 の場合について各アルゴリズムを適用した場

合の収束履歴である。倍精度演算 (-D) の場合は各アルゴリズムで同じ収束履歴が得られているが、単精度演算 (-S) の場合は各種法において異なっている。Alg.1 (オリジナルのアルゴリズム) が最も安定であるが、Alg.2, Alg.3 は不安定である。Alg.4 (Gropp アルゴリズム) は Alg.2, Alg.3 と比較して安定している。

(4) まとめ

上記(1)～(3)では、ICCG 法に倍精度演算, 単精度演算を適用した場合の比較検討を実施した。単精度演算適用時は、反復回数が増加するものの計算時間は倍精度演算適用時の概ね 60%～70%となるが、係数行列が悪条件になると収束が悪化し、計算時間が増加する傾向にある。SELL-C- σ の有効性についても確認されたが、Intel Xeon Phi 上では SIMD の更なる有効な活用を目指して最適化を継続して実施する必要がある。

今後は、⑦で述べる精度保証, ⑧・⑨で述べる消費電力測定とも連携して、Iterative Refinement (反復改良法) の導入により、悪条件下でも高効率で安定な手法の検討を実施する。また、真の残差ベクトルを導入することにより、単精度演算時の安定性を向上させる。また精度を途中で切り替える変動精度的手法の導入についても検討する予定である。

④疎行列演算：量子科学計算

本研究では固有値ソルバーなどから導出されるような悪条件な問題を対象として、前処理付きクリロフ部分空間法に高精度演算を適用し、収束性の向上および高い要求精度への対応が可能かどうか検証を行った。

本年度には対角シフトおよびブロック化を適用した IC 前処理付き COCG 法に 4 倍精度演算 (IEEE754) を採用し、収束性および実現可能な相対残差の精度を評価した。結果、収束性の向上は確認できなかったが、 10^{-32} 程度の高い要求精度への対応は可能であることが分かった。

評価に際して、ブロック IC 前処理付き COCG

法で高精度演算が効果を持つと考えられる個所は複数存在するが、ここではすべての演算を 4 倍精度演算に置き換えた。また、対象のモデルは graphene および Topological insulator とした。なお、4 倍精度の採用により正則化の効果が変わる可能性があるため、予備実験として 4 倍精度を採用した IC-COCG でパラメータ探索を行い、最適な正則化パラメータの確認を行った。結果、精度を変えても正則化の最適なパラメータは変わらないことを確認した。したがって、倍精度の場合で最適であった、ブロックサイズ 8, 対角シフトは(0.0, 1.0)を本評価でも採用した。結果、倍精度の場合ではいずれの問題でも最終的な相対残差は 10^{-14} 程度であったが、4 倍精度の採用によって 10^{-32} と大きく向上することを確認した。一方で、4 倍精度演算を用いた場合でも収束性の改善は確認できなかった。これはいずれの精度でも収束履歴が非常に似通っていたためである(図 13)。ただし、4 倍精度の採用によって、一定反復以上の収束性は非常に良好になっていることも確認している。

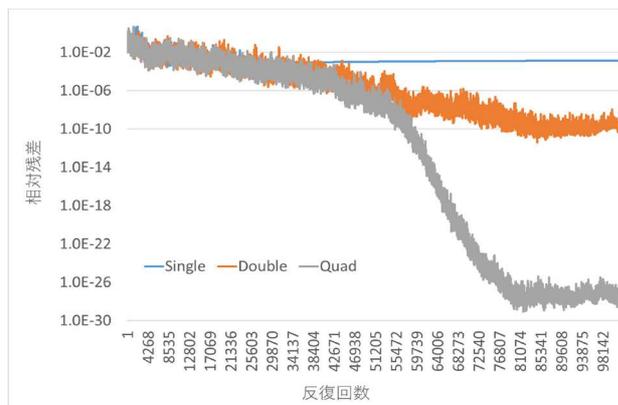


図 13 Topological insulator (1 万自由度) モデルでの各演算精度の収束履歴

今回の評価では収束性向上の観点からは 4 倍精度の優位性を見いだせなかったが、より高度な前処理との組み合わせによって収束性改善を見込めないか検証をおこなっていく。また、精度保障演算との組み合わせによって収束性がどう変化するかの評価も行う。加えて、実際の固有値ソルバーなどでは高い演算精度が要求されるため、これら実アプリケーションでの有効性の評価も行ってい

く。

⑤Adaptive CG

線形方程式の反復法ソルバーの一種である共役勾配法(CG法)の前処理において、固定された行列 M を使って探索方向 $z \leftarrow M^{-1}r$ を定めるのではなく、対象とする方程式 $Ax=f$ の A による前処理方程式 $r=Az$ を解いた z を前処理の探索方向に使うアルゴリズムを Adaptive CG 法と呼ぶ。本研究では、市村らによって SC14 で開発された三次元有限要素法による地震解析プログラムにおける Adaptive CG ソルバーを拡張することでさらなる高速化を実現した。非構造格子有限要素法においてランダムアクセスは避けられず、近年の計算機においては疎行列ベクトル積におけるランダムアクセスとメモリ転送が性能のボトルネックとなっている。そこで本研究では非構造格子であっても時間方向にはメッシュが不変であることに着目し、時間方向に複数ステップ同時に解くことでランダムアクセス・メモリ転送量を削減する時間並列アルゴリズムを開発して Adaptive CG 法の中に実装した。これにより、演算数をほぼ一定に保ちつつランダムアクセス量・メモリ転送量を減らすことが可能になった。なお、開発した Adaptive CG 法では前処理で単精度、CG 法本体で倍精度を使う精度混合演算を実装している。これを京, Oakforest-PACS (OFP) に最適化し、性能を計測した。開発したソルバーは SC14 ソルバー比で、京において 1.76 倍の高速化、OFP において 5.02 倍の高速化を達成した。SIMD 幅が広いために相対的にランダムアクセス時のペナルティが相対的に大きい Intel Xeon Phi (Knights Landing) ベースの OFP において性能が顕著に改善していることがわかる。さらに、時間並列計算により MPI_Allreduce 呼び出し回数を削減できるため、京 1152 ノードから 82944 ノードまでの weak scaling が SC14 ソルバーの 88.8% から 91.4% に改善した。また、OFP における単体性能の高さと高い weak scaling 効率 (128 ノードから 8192 ノードまでの weak scaling が 83.7%) により、OFP ほぼ全系にあたる 8192 ノードでピーク倍精

度 FLOPS 性能比 13.4% という高い性能を達成した。低減できる可能性がある。

開発した時間並列アルゴリズムを実装した Adaptive CG 法の研究は順調に進んでおり, 先行的に得られた成果の一部は IPDPS 2018 でも発表している [1]。また, IPDPS 論文 [1] を拡張する形で人工知能による前処理の高度化を実施した成果は SC18 において Gordon Bell Prize Finalist に選ばれている [2]。また, SIMD 幅の広い計算機上で並列時間積分法を高性能で実行できるアルゴリズムを開発し, Oakforest-PACS 上で性能を確認し, ICCS 2019 に採択されている [3,4]。

⑥ H 行列

(1) H 行列ライブラリの新たな実装

階層的低ランク近似法には H 行列だけでなく H^2 行列, HSS 行列, HOLDR 行列, BLR 行列など様々なものがあり, 並列性や演算効率を考慮するとどれも一長一短である。本実装では, これらのどの階層的低ランク近似にもなることができるような汎用的な実装を実現した。

C++ の演算子オーバーロードを用いて Matlab で密行列を操作するように H 行列を操作できるようなインターフェイスを提供している。内部カーネルは Batched BLAS を用いて CPU 上でも GPU 上でも高い演算効率を実現できる仕組みになっている。現在, Batched ACA を用いて圧縮の部分を高速化することと, Batched GEMV を用いて行列-ベクトル積を高速することに成功している。

今後は, LU 分解や QR 分解の内部も GPU 上で Batch 処理できるよう, StarPU, OmpSs などのランタイムを用いて処理の依存関係を考慮し非同期的に GPU 上で行列分解の内部カーネルが実行されるような実装を行う予定である。

(2) 階層的低ランク近似が低精度演算の誤差を低減できる可能性

単精度や半精度などの低精度演算を用いる場合, 行列演算の足し込みの部分で情報落ちによる演算精度の低下が起きる。これは演算量が増えるに従って誤差が増大することを意味しており, 演算量を低減する近似手法は逆に情報落ちによる誤差を



図 14 密行列, 低ランク行列, 階層的低ランク行列の非ゼロ要素の分布

図 14 に示すような 3 種類の行列との行列-ベクトル積を単精度で行ったときの倍精度の結果と比べてときの相対二乗ノルム誤差を図 15 に示す。行列の規模が大きくなるにつれ, 密行列では多数の余計な演算により倍精度と比べてときの精度が大きく低下していることが見て取れる。一方, 低ランク近似を行った場合, わずかに精度の低下が抑えられていることが分かる。さらに, 階層的低ランク近似を用いた場合では, 同じ行列の計算にも関わらず, 特に N が大きいときの精度の低下が大幅に抑えられているのが分かる。これは, 半精度の演算や Tensor Core を用いた演算においてはより顕著に現れることが予想されており, 今後の重要な課題と位置付けられる。

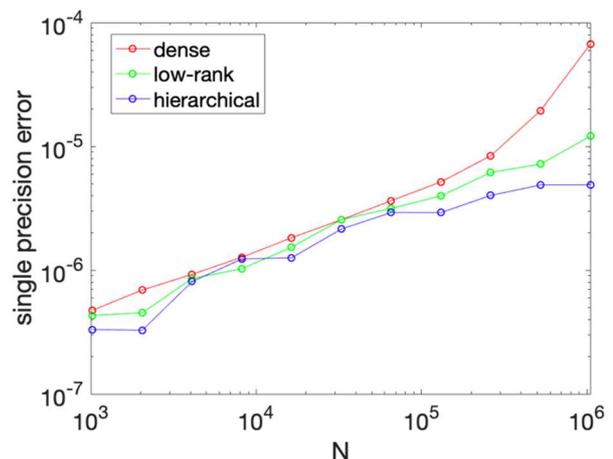


図 15 単精度での行列-ベクトル積をそれぞれ密行列, 低ランク行列, 階層的低ランク行列で行ったときの倍精度密行列積との相対二乗ノルム誤差

(3) Batched QR 分解の TensorCore 実装

本研究では, これまで H 行列, H^2 行列, HSS 行列, HOLDR 行列, BLR 行列など様々な階層的低ランク近似法 (以下ではこれらの手法を簡単なた

めにまとめて H 行列と表記する) に対応できるライブラリの開発を行ってきた。このライブラリの GPU 実装や MPI 対応などを現在進めており、StarPU などのランタイムを用いた実装により GPU 上でもノード間でも非同期的に H 行列の LU 分解などを行うことができるようになる。

特に、GPU 上で高い性能を出すためには H 行列の計算で最もコストのかかる randomized SVD を用いた低ランク近似の部分的高速化しなければならない。Randomized SVD の内部カーネルは複数の QR 分解からなっており、H 行列のそれぞれのブロックをまとめて低ランク近似するためには batched QR 分解が必要になる。Volta 世代の GPU には TensorCore が搭載されており近似計算である randomized SVD の内部の QR 分解には十分な精度であると考えられる。

そこで、今回は batched QR 分解の TensorCore 実装を行いその精度と性能を検証した[10]。GPU 上での実装のしやすさから QR 分解は Householder 変換を用いるアルゴリズムを採用した。また、その際に発生する行列積の部分を TensorCore を用いて実装した。精度の検証のために入出力もノルムの計算も単精度で行った場合(FF)、入出力は半精度でノルムの計算は単精度で行なった場合(HF)、両方を半精度で行なった場合と、それぞれの行列積を TensorCore で行なった場合(FF-TC, HF-TC, HH-TC)の計 6 通りの実験を行なった。また、バッチサイズは 2^5 から 2^{16} まで変化させて FLOPS 値を計測した。

図 16 にバッチサイズを変化させてときの FLOPS 値を示す。TensorCore を用いて入出力を半精度で行なった場合はノルムの計算の演算精度に関わらず最も高い性能が得られた。V100 を 1 台用いて 10TFLOPS の演算性能で QR 分解を行えることになる。入出力は単精度にすると演算性能は少し落ち、TensorCore を使わない場合は 5TFLOPS に満たない性能になっていることがわかる。また、バッチサイズは 2^{10} 程度あればかなり高い性能が出るが、それ以下では急激に性能が低下することが見てとれる。ただし、一つ一つの QR

分解は 16×16 の大きさであった。

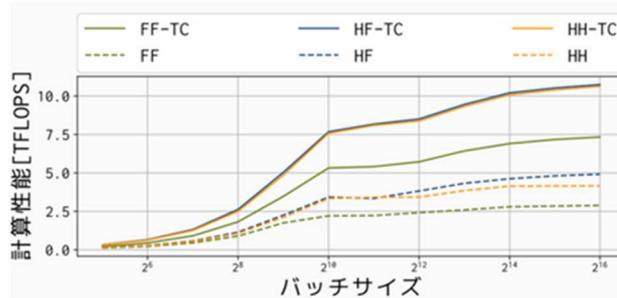


図 16 バッチサイズと計算性能の関係

(4) 混合精度演算による H 行列ベクトル積の高速化

本研究では、従来の倍精度演算に加えて単精度演算を利用することによる H 行列ベクトル積の高速化について検討を行った。具体的には 3 種類の混合精度演算 H 行列ベクトル積の実装法を提案し、その性能評価を行った。

実装方式 1 では、H 行列内の低ランク近似部分行列を、すべて単精度化することを行った。右辺ベクトルについては、単精度、倍精度の双方の方式を比較しており、他の実装方式でも同様である。実装方式 2 では、前述の方法に加えて、低ランク近似部分行列の表現を変更した。本研究の基礎となっている Hcapk ライブラリでは、低ランク近似部分行列を 2 つの長方形行列の積として表現している。提案手法では、これらの長方形行列を対角スケールリングし、その対角成分を対角行列として表現することで、表現力を強化した。これは、主に単精度化に伴う指数部の表現力低下を補償する目的で考案したものであるが、仮数部においてもなんらかの効果が期待できる。実装方式 3 では、各部分行列において単精度・倍精度の双方を用いることを導入した。実装方式 2 における対角行列の各成分の大きさ (絶対値) を比較し、相対的に大きいものを倍精度表現することとした。

提案した 3 種の実装方式を境界要素法による静電場解析上で評価した。同解析では、BiCGSTAB 法を用い、連立一次方程式の求解を行うが、その内部で H 行列ベクトル積を用いている。混合精度演算を用いた場合、倍精度演算と比べて、一般に

行列ベクトル積の結果は変化する。これらの精度の変化を H 行列ベクトル積単位で見積もることは困難であり、本研究では BiCGSTAB 法の反復数でその影響を評価する。今回の数値実験では、実装方式 1, 2 において若干の反復回数の増加がみられた。一方、実装方式 3 では、適切に単精度、倍精度演算導入の閾値を設定することで、反復回数の増加を抑えることができた。一方、計算時間の点では、実装方式 2 が最も高速で、倍精度演算を用いる従来の実装方式に比べて 1.47 倍の高速化を達成した。

⑦精度保証

③で示したような 7 点差分メッシュで離散化して得られた係数行列 A が M 行列性を持つ場合、高速な精度保証法 [Ogita et al., Computing, 2001] を適用可能である。具体的には、以下のようなアルゴリズムとなる：

- I. 離散化して得られた連立一次方程式 $Ax = b$ を解く。得られた近似解を \hat{x} とする。
- II. 連立一次方程式 $Ay = e$ (e はすべての要素が 1 のベクトル) を解く。得られた近似解を \hat{y} とする。
- III. 上記の \hat{y} を用いて係数行列 A の M 行列性の保証を行う。
- IV. 以下の誤差評価式の右辺について、浮動小数点演算の丸めモードの変更を適宜使用しながら、厳密な上限を求める：

$$\|x - \hat{x}\|_{\infty} \leq \frac{\|\hat{y}\|_{\infty} \|b - A\hat{x}\|_{\infty}}{1 - \|e - A\hat{y}\|_{\infty}}$$

(但し、右辺の分母が 0 以下になった場合は精度保証失敗)

上記のアルゴリズムを実装し、東京大学情報基盤センターの Reedbush-U (1 ノード) において数値実験を行った。連立一次方程式のソルバには ICCG 法 (CM-RCM : 20 色) を用いて、OpenMP による並列化 (36 スレッド) を行った。ICCG 法の反復の停止条件は残差ノルムを用いて：

$$\begin{aligned} \|b - A\hat{x}\|_2 / \|b\|_2 &< \varepsilon_1 \\ \|e - A\hat{y}\|_{\infty} &< \varepsilon_2 \end{aligned}$$

のように与えた。但し、それぞれの残差ベクトルの計算については ICCG 法の反復中に計算される残差ベクトルを用いた。また、浮動小数点演算は、すべて倍精度を用いた。ここでは、以下のような問題設定とする：

- $NX = NY = NZ = 128$ ($n = 2,097,152$)
- $\lambda_1 = 1.0, \lambda_2 = 1.0$ に固定
- $\varepsilon_1 = 10^{-12}, \varepsilon_2 = 10^{-2}$

離散化して得られた連立一次方程式の近似解 \hat{x} を計算し、精度保証法を用いて、最大相対誤差：

$$\max_{1 \leq i \leq n} \left| \frac{x_i - \hat{x}_i}{x_i} \right|$$

の上限及び相対残差ノルム：

$$\|b - A\hat{x}\|_2 / \|b\|_2$$

の上限をそれぞれ求めた。最大相対誤差は $3.38e-8$ 、相対残差ノルムは $3.66e-11$ であった。したがって、近似解は最低でも 10 進 7 桁は正しいことがわかる。また、計算時間及び ICCG 法の反復解法について、表 1 に示す。この結果から、この問題設定では、近似解の計算と比べても実用的な計算時間で精度保証ができていくことがわかる。

表 1 計算時間(s)及び反復回数

	近似解 \hat{x}	精度保証
計算時間(s)	3.26	1.25
反復回数	415	211

次に、数値例 1 と同様の問題設定で、 λ_1/λ_2 を 1.0 から 10^6 まで変化させたときの最大相対誤差の上限及び相対残差ノルムの上限を図 17 に示す(但

し、 $\lambda_1 = 1.0$ に固定)。

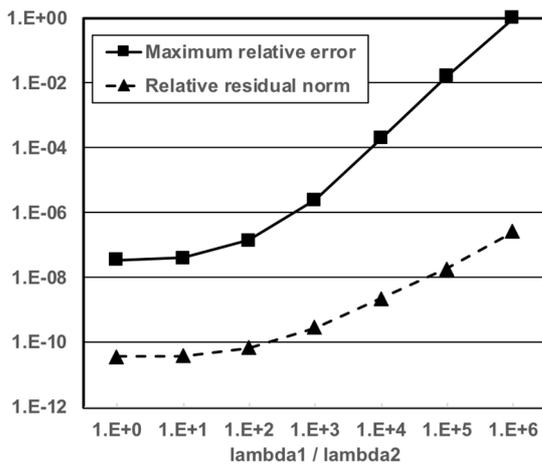


図 17 条件 (λ_1/λ_2) 毎の最大相対誤差 (実線) と相対残差ノルム (破線)

図 17 から、条件が悪くなる (λ_1/λ_2 が大きくなる) につれて、最大相対誤差の上限及び相対残差ノルムの上限が増大していることがわかる。また、相対残差ノルムのみから最大相対誤差を推定することは困難であることがわかる。特に、 $\lambda_1/\lambda_2 = 10^6$ のときは、相対残差ノルムは 1 より小さいが、最大相対誤差の上限は 1 を超えており、近似解の精度が 1 桁も保証できていないことがわかる。

精度保証については、三次元熱伝導方程式を離散化して得られた連立一次方程式に対しては、実用的な精度保証法の適用に成功した。今後は、より高精度な誤差評価式の適用、単精度演算等の低精度演算での数値実験、離散化誤差の評価方法の検討、階層型行列法における精度保証法の検討等が課題である。

⑧消費電力測定 (Reedbush-L)

(1) Reedbush 向け電力評価環境の構築

Reedbush 上で HPC アプリケーションの電力評価を行うために、下記の作業を行った。

A) Reedbush への電力評価ライブラリの導入 : 電力計測には当グループの研究プロジェクトで開発した電力評価ライブラリを用いた。本ライブラリは実行中のプログラムの消費電力や

パフォーマンスカウンタの値を詳細に記録する機能を有する。後述するポアソン方程式のソルバーに対し、本ライブラリを組み込み、評価を実施した。この際、ライブラリを利用するためソースコードに軽微な修正を加えている。また、Reedbush 環境向けに Makefile やジョブ実行スクリプト等を新たに整備した。

B) 電力制御のための特権設定 : 電力制御を行うためにはプロセッサが提供する MSR と呼ばれる特殊なレジスタに対してアクセスを行う必要がある。MSR はプロセッサの状態を制御するレジスタであり、通常 root 権限を持つユーザのみアクセス可能である。そこで、一般ユーザから電力評価・制御を行うための仕組みをベンダと協力して作成した。上記の MSR レジスタに対する読書許可をジョブ実行の直前に実行バイナリファイルに対し付加する仕組みを構築した。これにより、Reedbush 上でプログラムの消費電力計測と、プロセッサの最大消費電力を制限するパワーキャッピング機能が利用可能となった。現在は特定ノード・特定ユーザのみ利用可能であるが、設定により他ユーザも簡単に電力計測を行うことができる。

(2) 熱伝導方程式ソルバーの電力評価

③で述べた ICCG 法による熱伝導方程式ソルバーの電力評価を行った。本評価では低精度演算利用時における電力を確認するため、精度を変えて (double, single) 計算を行った場合の平均消費電力を評価した。また、データ配置方式 (coalesced, sequential), CM-RCM の色数 (c8, c32, c128) を変えて評価を行った。図 18 に結果を示す。倍精度を単精度化することにより CPU 電力(PKG_power)が増加した。これは、演算密度が上昇したためである。IPC も 2 倍程度上昇した。

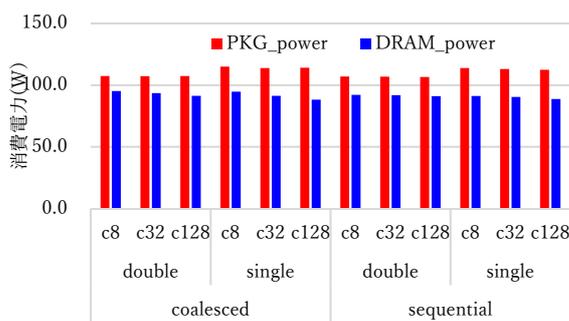


図 18 倍精度(double)を単精度(single)化した場合のICCG 関数の平均消費電力の比較

また、CM-RCM の色数を 8, 32, 128 と増やすことによって DRAM の電力が低下することが分かった。データ分割数を増やすと CPU キャッシュの利用効率が低下したためだと考えられる。次に消費エネルギーに関して評価を行った。この結果を図 19 に示す。

低精度化することにより実行時間が大きく短縮されることが確認でき、これに伴い消費エネルギーも大きく削減された。また、coalesced より sequential の方が削減率大きい。分割数を増やすぎると実行時間が長くなり、エネルギー効率が悪化した。

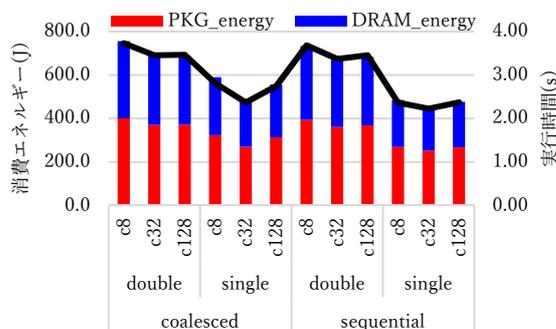


図 19 倍精度(double)を単精度(single)化した場合の消費エネルギーの比較

次に、プロセッサの動作周波数を変えた場合の消費エネルギーについて確認を行った。Reedbush では動作周波数を一般ユーザから制御できなかったため、本評価は手元の Haswell マシンを用いて行った。また、スレッド数を変化させた。これらの結果を図 20 に示す。

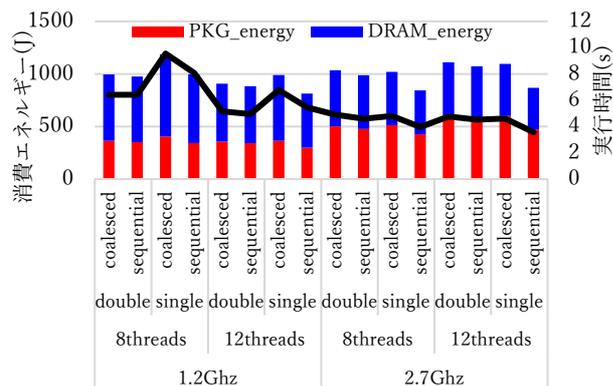


図 20 プロセッサの動作周波数を変えた場合のエネルギー評価

この結果、消費エネルギーを最小化するためにはプロセッサの動作周波数を 1.2GHz とし、さらに 12 スレッドを用いて計算を行ったほうが良いことが確認された。本ソルバーにおいて消費エネルギーを削減したい場合、低い周波数(1.2GHz)でスレッド数を増やして計算したほうが良い傾向にあることが分かった。

(3) AdaptiveCG の電力評価

⑤で述べた AdaptiveCG の電力評価を行った。AdaptiveCG は前処理行列を荒く解くことが可能である。そのため、この前処理部分に単精度演算を用いることが可能である。本地震シミュレーションの評価では前処理部分に着目し、倍精度を用いた場合と単精度を用いた場合との比較を図 21 に示す。また、計算に用いる OpenMP のスレッド数を 3, 6, 9, 12, 15, 18 スレッドと変化させた。

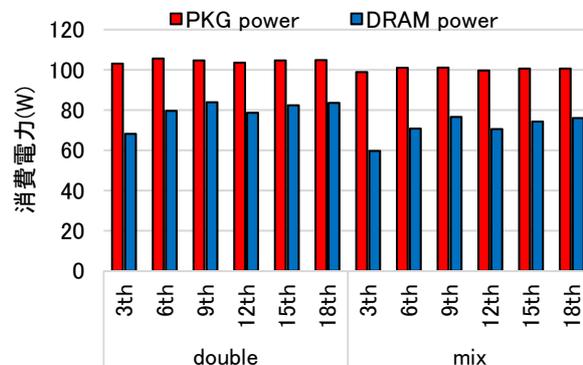


図 21 AdaptiveCG の平均消費電力の比較

ソルバーの全域を倍精度で解いたものを double、前処理部分のみを単精度化したものを mix とする。これらの結果、ICCG 法とは対比的に倍精度 (double) を単精度化 (mix) 化することにより、CPU 電力 (PKG power) と DRAM 電力 (DRAM power) がともに低下することが確認された。倍精度の CPU 電力の平均は 104.4[W] であったが、単精度の CPU 電力の平均は倍精度よりも 4[W] 程低い 100.3[W] であった。また、倍精度の DRAM 電力の平均は 79.4[W] であったが、単精度の場合、8[W] 程度低い 71.3[W] であった。詳細は確認中であるが、AdaptiveCG は非常に大きなデータを扱うため、LLC ミス率が 50% 近くと非常に高いことが関係していると考えられる。

次に実行時間と消費エネルギーの評価結果を図 22 に示す。

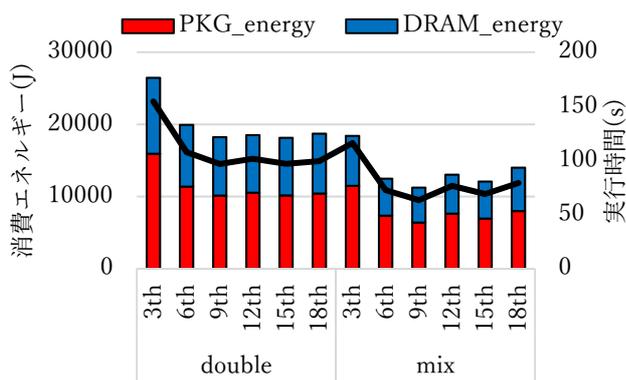


図 22 AdaptiveCG の実行時間と消費エネルギー

この結果、倍精度の場合も単精度の場合も OpenMP のスレッド数を 9 にしたほうが実行時間が短くな

ることがわかった。評価に用いた Reedbush の CPU はソケット当たり 18 コアの物理コアを有しているが、半数のコアを用いたほうが実行時間が短くなる結果となった。また、倍精度と単精度を比較した場合、ICCG 法と同様に単精度化することにより実行時間が大幅に短くなることが確認できた。その結果、単精度で 9 スレッドを用いた際に最もエネルギー効率が良くなることがわかった。倍精度で 9 スレッドを用いていたものを単精度にすることで、最大で 38.3% のエネルギーを削減することができた。また、平均で 32.4% のエネルギー削減となった。

(4) H 行列計算の電力評価

⑥で述べた H 行列計算の主要な計算は行列とベクトルを掛け合わせる行列ベクトル積である。そこで、行列データのみを単精度化し行列ベクトル積を行う mix と、行列データとベクトルデータの双方を単精度化し行列ベクトル積を行う single を用意し、電力計測を行った。これらの結果を図 23 に示す。PKGPOWER は CPU 電力、DRAMPWR は DRAM 電力を示している。また、異なる 8 つのデータセット (sample, 10ts, 216h, 100ts, human_5x5, human_1x30, 676th, 1188kp25) に対して精度を変えて電力計測を行った。double は行列データ、ベクトルデータ共に倍精度のデータを用いた場合を示している。これらの結果、sample 以外の 7 つのデータセットにおいて、精度を倍精度 (double)、行列部のみ単精度 (mix)、行列ベクトル共に単精

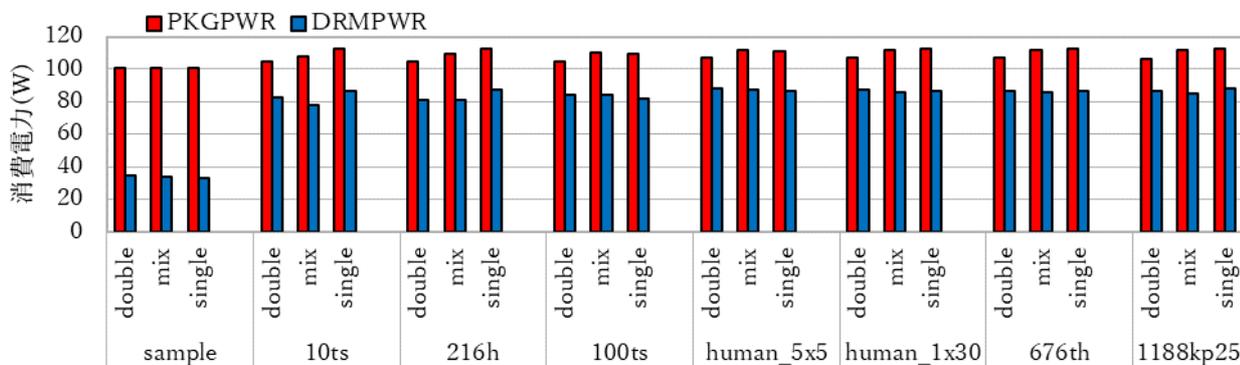


図 23 H 行列計算の平均消費電力

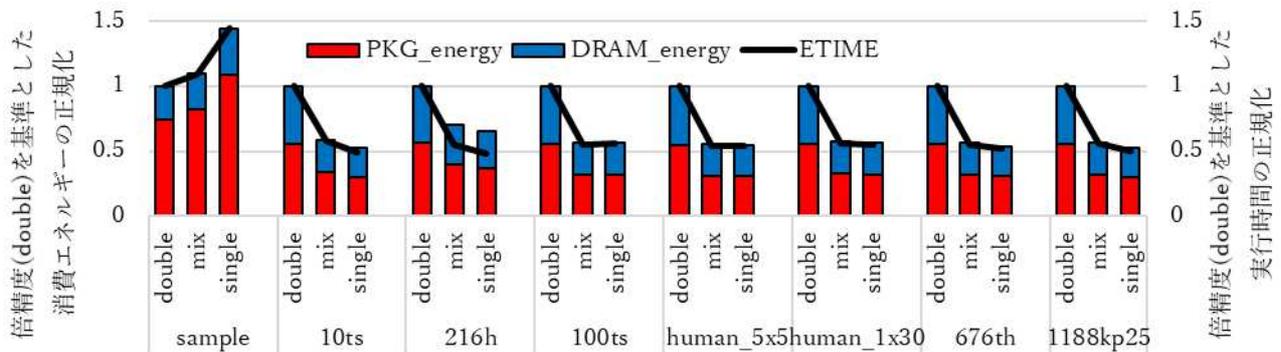


図 24 H 行列計算の実行時間と消費エネルギー

度(single)と変化させることによって CPU 電力が上昇することが分かった。これは、単精度化することにより演算密度が上昇したためである。100s, human_5x5 については、行列のみ単精度(mix)から行列ベクトル共に単精度(single)にした際の消費電力の電力増加が少なかった。sample は行列サイズが非常に小さく、演算時間が短かったため十分な電力計測精度が得られなかった。また、DRAM の消費電力についてはデータセットによって異なる傾向となった。

次に実行時間と消費エネルギーの計測結果を図 24 に示す。データセットによって実行時間が大きく異なるため、それぞれのデータセットにおいて倍精度(double)を基準として正規化を行った際の実行時間と消費エネルギーの結果を示している。PKG_energy は CPU が消費したエネルギーを示し、DRAM_energy は DRAM が消費したエネルギーを示し、ETIME は実行時間を示している。これらの結果より sample 以外の 7 つのデータセットにおいて、倍精度(double)を行列のみ単精度(mix)化することによって実行時間が 4 割程度短くなったことがわかる。10ts, 100ts, human_5x5, human_1x30, 676th, 1188kp25 については消費エネルギーも 4 割程度削減された。216h は 3 割程度の削減であった。また、行列のみ単精度(mix)を行列ベクトル共に単精度(single)化した場合もエネルギーの削減効果が見られたが、倍精度(double)を行列のみ単精度(mix)化することによるエネルギー削減効果と比較すると効果は限定的である。これは、行列部分を単精度化することで大幅なデータサイズ削減が達成で

き B/F 比が大きく改善するが、ベクトル部分は全体に占めるデータ量が限定的であり B/F 比の改善率が小さいためだと考えられる。

(5) NICAM の電力評価

気候・気象シミュレータである全球雲解像モデルである NICAM を単精度化することによる消費電力の変化について評価を行った。計測結果を図 25 に示す。

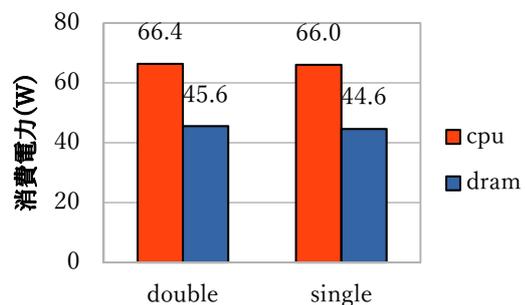


図 25 NICAM の平均消費電力

double は NICAM を倍精度で計算を行ったオリジナルコードでの消費電力を示し、single はオリジナルコードを単精度化したものである。この結果、倍精度(double)を単精度(single)化することにより、CPU 電力、メモリ電力ともに消費電力が多少低下する結果となった。ICCG 法と H 行列計算の場合倍精度を単精度化することによって CPU 電力が上昇する傾向にあったが、NICAM は CPU 電力が低下する結果となった。今回評価を行った NICAM は評価の都合上 MPI プロセス数が 4 であり、十分に並列化が行えていない状態で評価を行った。本評価に用いた Reedbush-L は 18 個の CPU コアを有し

ており、CPU の資源を有効に利用できなかったためであると考えられる。また、多数のカーネルから構成されており、単精度化の恩恵を受けにくいカーネルも含まれることが要因と考えられる。次に、実行時間と消費エネルギーの評価結果を図 26 に示す。

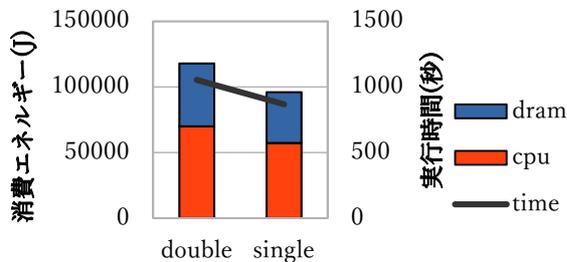


図 26 NICAM の実行時間と消費エネルギー

倍精度(double)を単精度(single)化することにより、2 割程度実行時間が短くなった。これは、演算密度が上昇したためである。消費エネルギーについても同様に 2 割程度削減することができた。他のアプリケーションと比較すると削減率が小さい結果となった。NICAM の実行時間の削減率が小さい理由は先にも述べたように多数のカーネルが含まれており、単精度化が寄与した部分が比較的に少ないためであると考えられる。

(6) アプリと電力特性

一般に低精度演算を導入することによって B/F 比が上昇する。これにより単位時間当たりの演算密度が上昇することにより、多少消費電力が増加する。演算密度の上昇は実行時間の短縮に大きく寄与する。したがって消費エネルギーが大きく削減される。ICCG ではこの結果が顕著であり、倍精度演算を単精度化することによって多少の CPU 電力増加がみられた。また、実行時間が大幅に短縮されたことにより最大 34%のエネルギー削減効果が得られた。H 行列においても単精度化を行うことにより CPU 電力が若干増加したが、実行時間の短縮化により大きくエネルギーを削減できた。一方で AdaptiveCG や NICAM は単精度化することにより CPU 電力が若干減少する結果となった。AdaptiveCG

や NICAM は多数のカーネルから構成されており、カーネルごとに特性が異なることが影響し、最終的な平均電力が下がる結果となったと考えられる。AdaptiveCG は実行時間が 3 割程度高速化されたため、消費エネルギーを最大 38%削減することができた。NICAM は単精度化することで実行時間が 18%程度短くなり、消費エネルギーも 18%削減できた。NICAM も多数のカーネルから構成されており、単精度化による高速化の効果が限定的であったためと考えられる。

(7) まとめ

4 つのアプリケーションに対し単精度演算を積極的に用いることによる効果を実験評価した。この結果、単精度化を行うことにより B/F 比が上昇し実行時間が短縮されることが確認できた。実行時間の短縮化は大きく消費エネルギーの削減に寄与した。しかしながらその削減率はアプリによって 38%~18%と大きく異なる結果となり、アプリごとに特性が異なることが分かった。今後は動的な消費電力変動の確認や、キャッシュ挙動の確認など、より細かな粒度に着目する。さらに、計算中に精度を切り替えることによるエネルギー削減を検討する。初めは単精度で計算し、ある程度収束したら倍精度に切り替えるといった方法が考えられる。

⑨消費電力測定 (Knights Landing)

③で述べた ICCG 法による熱伝導方程式ソルバーを用いた電力評価を Oakforest-PACS(OFP)のテストベッド基であり、OFP 同様に Knights Landing(KNL)ベースのシステムである OFP-Mini を用いて行った。ただし、電力制御にて root 権限が必要なため、OFP ではなく、本システムにて評価を行っている。特に、本評価の目的は、精度選択(単精度・倍精度)が性能・電力に与える影響を明らかにする事である。下記は評価環境の概要である：

- システム : OFP-Mini, 64 コア/ソケット, 1 ソケット/ノード, 16GiB MCDRAM, 96GiB DDR4, ノード内設定: quadrant, flat
- 電力評価環境 : インターフェイス: RAPL, 電力ドメイン 1: CPU+MCDRAM, TDP 215W, 電力ドメイン 2: DDR4, 最大実測電力 80W
- 電力制約設定 : 電力ドメイン 1 {120W, 140W, 160W, 180W, 200W, no limit}, 電力ドメイン 2 {no limit}
- アプリケーション設定 : スレッド数: 128, 色数: 10, メモリマップ: MCDRAM のみ, SIMD 幅: 16(単精度), 8(倍精度)

OFP のテストベッド基にて電力評価環境を構築し、3D Poisson コードを用いた電力評価を行った。今後は、本環境を用いて、その他のアプリケーションの電力評価を行い、精度選択が性能・電力に与える影響を評価する。今回は SIMD 幅を固定して評価を行っているが、SIMD 幅は演算器の電力に大きく影響を与えるため、これを変えた場合の評価も行う。

⑩自動チューニング

所望の精度で計算時間を最小化する最適演算精度を、アプリケーション・係数行列の性質、問題サイズ、ハードウェア環境等に基づき自動チューニング技術 (AT) によって動的に制御する手法を検討する。また、既存の ppOpen-AT (ppOpen-HPC の自動チューニング機構) への実装の検討を行う。

具体的には、NICAM の物理過程のカーネルにおいて、以下のような多変数による演算が中で行われている：

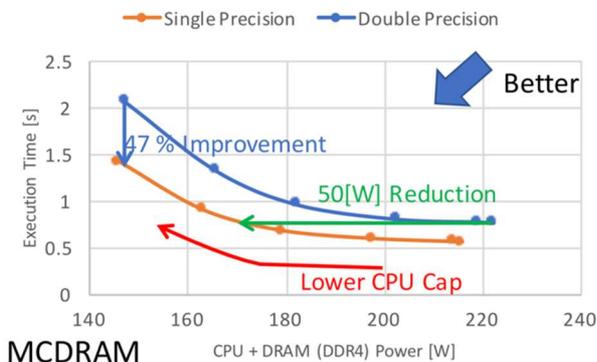


図 27 性能・電力比較

図 27 に評価結果の一部を示す。横軸は上記 2 つの電力ドメインにおける消費電力の和を表し、縦軸は実行時間を示している。各々 2 つの曲線は、単精度(Single Precision)及び、倍精度(Double Precision)の場合の評価結果である。各々の点は電力ドメイン 1 の設定電力制約と対応している(電力制約を厳しくすれば、消費電力は減少するものの、実行時間は増加する)。図 27 において、左下の領域に行けば行くほど、実行時間は減少し、電力も減少する。図 27 に示す通り、単精度の曲線は倍精度のそれよりも左下に存在し、性能・電力の観点から、より良い事が分かる。例えば、総電力の制約を 150[W] に設定した場合には、単精度を用いる事で、性能を 47% 向上させる事ができる。一方、実行時間制約を設定した場合には、単精度を用いる事で、最大 50[W] の電力削減効果が得られることが分かる。

!べき乗計算

$$RLMDr_dr = \text{sqrt}(RLMDr)$$

$$RLMDr_2 = RLMDr^{**}2$$

$$RLMDr_3 = RLMDr^{**}3$$

...

!指数計算

$$\text{loga}_- = \text{coef_at}(1); \text{b}_- = \text{coef_bt}(1)$$

$$\text{MOMs}_0 = (\text{sw_roh2014}) * \text{exp}(\ln 10 * \text{loga}_-) * \text{exp}(\text{log}(\text{Xs2} + \text{zerosw}) * \text{b}_-) * (1.0_RP - \text{zerosw}) \& + (1.0_RP - \text{sw_roh2014}) * \text{MOMs}_0$$

...

以上の内部で使われている変数を、単精度、半精度に変更し、計算結果の精度が保たれる変数精度の組み合わせを探すことで、高速化したいというアプリケーション上の特性からの要求がある。しかし現在、以上の要求を満たす ppOpen-AT の機能は現在存在しない。

そこで机上検討を行い、以下の既存機能、および新規機能を組み合わせることで、実現できる可

能性があることを検討した。

- コード選択のための指定子 select, および, 選択のための評価関数を指定する補助指定子 according の利用
- 各変数宣言部を指定する新ディレクティブの策定
- 精度評価を行う変数を指定する新ディレクティブの策定

現在までの進捗状況は机上検討のみであり, 今後は, 上述の内部変数の精度を変化させて精度を考慮して高速化が可能か, NICAM ベンチマークを用いて AT 機能のプロトタイピングを行い, 性能評価することを検討する。

6. 今年度の進捗状況と今後の展望

本研究は, 計算科学・計算機科学・数値アルゴリズム分野の研究者の協力のもと, 様々な数値アルゴリズムの最新アーキテクチャに向けた最適化, 低精度・変動精度導入による高速化, 消費電力節約, および精度保証に基づく最適精度選択のための自動チューニング選択手法の確立と実アプリケーションでの検証を目指したものである。個別の研究開発項目の進捗状況と今後の展望については 5. に示した通りである。

本研究では, ステンシル計算 (①構造格子, ②半構造格子), 疎行列演算 (③一般行列, ④悪条件問題, ⑤Adaptive CG (局所前処理による)), ⑥H 行列, ⑦精度保証, ⑧消費電力測定, ⑨自動チューニング手法, の各項目についての研究開発を実施する。本研究は 2018 年度から, 3 年計画として実施している。2018 年度は, 上記①~⑩の項目について, (1) 性能最適化, (2) 多様な計算精度, (3) 精度保証, (4) 実アプリケーションを対象とした消費電力測定, を中心として予備的な検討を実施し, 後掲のように学会等で発表した他, ISC18, SC18 等の国際会議の展示ブースでも紹介した。低精度・変動精度演算をアルゴリズム, 最適化, 精度保証, 消費電力まで含めて扱った研究事例はほ

とんどなく, 大きな反響があった。2018 年度に得られた知見は以下の通りである :

- 従来, 主として倍精度演算が適用されていた科学技術計算に単精度・半精度演算を適用することによって, 同等の精度の計算結果がより短時間で得られる場合がある。
- 低精度演算の適用により, 演算密度が増加し時間当たり消費電力 (Watt) が若干増加する場合があるものの, ①で述べた計算時間短縮に比例して消費エネルギー (Joule) は減少する]。
- 悪条件問題では, 低精度演算では正しい解が得られない場合がある。特に半精度演算は変数の範囲が限定されるため注意が必要であり, 精度のあまり要求されない反復法前処理等に適用するべきである。
- 前処理付き反復法による疎行列演算では, 反復改良法 (Iterative Refinement), 残差計算 ($r=b-Ax$) の高精度化により, 低精度・変動精度演算でも安定した収束が得られる [7]。
- 従来の M 疎行列向け精度保証手法 (T. Ogita 他, 2001) は, 相対誤差上限の見積もりが厳しめである。

当初計画では, 2018 年度 : 各アルゴリズム・アプリケーションに対する最適化, 多様な演算精度適用, 消費電力測定, 2019 年度 : 精度保証手法確立, 2020 年度 : 自動チューニング手法確立を目標としていた。

本課題は 2019 年度も継続して採択されている。2018 年度は, 中間報告時以降, Serial・Parallel 最適化, 各アプリケーションの消費電力測定が進み, ほぼ当初の目標を達成した。消費電力・エネルギー測定については, 2018 年度は低精度演算の効果の確認のための予備的評価が中心であったため, 各アルゴリズムについて, 演算精度, 最適化, アーキテクチャの消費電力・エネルギーへの影響について, 体系的に評価を実施する必要がある。半精度演算については, 2018 年度に得られた知見を

元に、ステンシル計算、反復解法前処理への適用を継続して検討する。①～⑨は密接に連携して研究開発を実施するが、2019 年度の各項目の概要は下記の通りである：

- ステンシル計算 (①, ②) : 差分法コード, 地震波動伝播コード **Seism3D** (①), 全球大気計算コード **NICAM**(②)を対象として, 倍精度, 単精度, 半精度, 混合精度における計算の安定性・実行時間の評価, 各 CPU, GPU での消費電力測定の他, ループタイリング等の最適化による消費電力への影響についても検討する。
- 疎行列演算 (③, ④, ⑤) :
 - ✓ **ppOpen-MATH**, **pK-Open-SOL**, **Adaptive CG** とともに, 反復改良法適用による低精度・変動精度演算の安定化, 反復法前処理に対する低精度演算 (特に半精度) の適用の検討, 様々な手法 (**Pipeline** 法, **Dynamic Loop Scheduling**, **hCGA** 法, **SELL-C- σ**) の各演算精度による影響について, 安定性・実行時間の評価, 各 CPU, GPU での消費電力測定を実施する。特に **Adaptive CG** では半精度演算の実用化に重点を置く。
 - ✓ 更に。半精度演算については [4] で提案された 21 ビット浮動小数点表現 (FP21) の MPI 通信への導入を検討する (5) を参照)。
 - ✓ 悪条件問題については, **Pipeline** 法における倍々精度演算導入, 演算精度自動切替手法を研究する。
 - ✓ また, 国際会議発表時 [8] の指摘に基づき, 大規模分散並列問題で局所的・動的に演算精度を変動, 動的に負荷分散を適用する手法を **ppOpen-HPC** に基づき開発し, 不均質場地下水流れ問題等により検証する。
- H 行列 (⑥) : 低精度・変動精度演算に関する検討を継続して実施し, 悪条件問題向けの前

処理手法の開発, 電磁気学, 地震発生サイクル等シミュレーションによる性能検証, 精度評価, 電力測定を実施する。また, 悪条件問題を中心とした精度保証手法について検討を実施する。

- 精度保証 (⑦) :
 - ✓ 疎行列演算 (反復解法) については, 2018 年度研究で問題となった誤差上限の見積もりについてより精密な判定を実施する手法の研究開発を実施するとともに, 悪条件問題に対して反復改良法と組み合わせた精度保証手法の研究開発を実施する。
 - ✓ H 行列については, 悪条件問題を中心とした精度保証の研究開発を実施する。
- 電力測定 (⑧), 自動チューニング (⑨) : ①～⑦の各項目と連携して, アルゴリズム, 実装手法, 問題規模, 諸パラメータ, 演算精度, アーキテクチャと消費電力, 計算時間, 計算結果の関係を体系化し, 所望の計算精度を得るための最適パラメータ設定を自動的に決定する手法の検討を実施し, 2020 年度に予定する **ppOpen-HPC/pK-Open-HPC** への実装の準備を行う。

7. 研究成果リスト

(1) 学術論文

なし

(2) 国際会議プロシーディングス

- [1] [T. Ichimura](#), [K. Fujita](#), [M. Horikoshi](#), L. Meadows, [K. Nakajima](#), T. Yamaguchi, K. Koyama, H. Inoue, [A. Naruse](#), K. Katsushima, M. Hori, L. Madgededara, A Fast Scalable Implicit Solver with Concentrated Computation for Nonlinear Time-evolution Problems on Low-order Unstructured Finite Elements, 32nd IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2018
- [2] [Tsuyoshi Ichimura](#), [Kohei Fujita](#), Takuma Yamaguchi, [Akira Naruse](#), Jack C. Wells, Thomas

- C. Schulthess, Tjerk P. Straatsma, Christopher J. Zimmer, Maxime Martinasso, Kengo Nakajima, Muneo Hori, Lalith Madgededara, A Fast Scalable Implicit Solver for Nonlinear Time-Evolution Earthquake City Problem on Low-Ordered Unstructured Finite Elements with Artificial Intelligence and Transprecision Computing, SC '18 Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, 49:1-49:11, 2018 (Gordon Bell Prize Finalist).
- [3] Kohei Fujita, Masashi Horikoshi, Tsuyoshi Ichimura, Larry Meadows, Kengo Nakajima, Muneo Hori and Lalith Madgededara, Development of Element-by-Element Kernel Algorithms in Unstructured Implicit Low-Order Finite-Element Earthquake Simulation for Many-Core Wide-SIMD CPUs, Proceedings of International Conference of Computational Science (ICCS-2019), Springer Lecture Notes in Computer Science (LNCS) series (in press).
- [4] Ryota Kusakabe, Kohei Fujita, Tsuyoshi Ichimura, Muneo Hori, and Lalith Wijerathne, A Fast 3D Finite-element Solver for Large-scale Seismic Soil Liquefaction Analysis, Proceedings of International Conference of Computational Science (ICCS-2019), Springer Lecture Notes in Computer Science (LNCS) series (in press).
- (3) 国際会議発表
- [5] K. Nakajima, Numerical Library with High-Performance/Adaptive-Precision/High-Reliability Extension of ppOpen-HPC towards the Post Moore Era, Future Accelerated Math Library Design, ISC High Performance 2018 (Frankfurt, Germany, June 25, 2018) (**Invited Talk**)
- [6] K. Nakajima, Innovative Method for Integration of Computational & Data Sciences in the Exa-Scale/Post-Moore Era, APSCIT 2018 Annual Meeting (Asia Pacific Society for Computing and Information Technology) (Sapporo, Hokkaido, Japan, July 21, 2018) (**Keynote Talk**)
- [7] K. Nakajima, Innovative Method for Integration of Computational & Data Sciences in the Exa-Scale/Post-Moore Era, Seminar of IBM T.J. Watson Research Center (Yorktown Heights, NY, USA, July 25, 2018) (**Invited Talk**)
- [8] K. Nakajima, Application Development Framework for Manycore Architectures -from Exascale to Post Moore Era-, Exa/Post-Peta Scale Computational Mechanics, The 13th World Congress on Computational Mechanics (WCCM XIII) (New York, NY, USA, July 27, 2018)
- [9] T. Ogita, K. Nakajima, Verified solutions of linear systems arising from three-dimensional Poisson equations, Exhibits at SC18 (Dallas, USA, Nov. 12-15, 2018)
- [10] K. Nakajima, T. Ogita, and R. Sakamoto, Numerical Library with High-Performance/Adaptive-Precision/High-Reliability Extension of ppOpen-HPC towards the Post Moore Era, 2019 Conference on Advanced Topics & Auto Tuning in High-Performance Scientific Computing (ATAT 19), National Sun Yat Sen University (Kaohsiung, Taiwan February 16, 2019)
- [11] K. Nakajima, T. Ogita, R. Sakamoto, Numerical Library with High-Performance/Adaptive-Precision/High-Reliability Extension of ppOpen-HPC/ESSEX-II towards the Post Moore Era, 4th SPPEXA Workshop on Parallel Programming Models - Productivity & Applications for Exascale and Beyond, University of Versailles (Versailles, France, March 21, 2019) (**Invited Talk**)
- [12] K. Nakajima, M. Horikoshi, T. Hanawa, H. Fujita, M. Shiryaev, Optimization of Parallel Multigrid Solvers on Manycore Clusters, 19th Copper Mountain Conference on Multigrid Methods

(Copper Mountain, CO, USA, March 24-28, 2019)

(4) 国内会議発表

- [13] 中島研吾, ポストムーア時代の並列反復法アルゴリズム, 2018年並列/分散/協調処理に関する『熊本』サマー・ワークショップ (SWoPP 熊本2018), 日本応用数理学会「行列・固有値問題の解法とその応用」研究部会 (MEPA) (熊本, 2018年7月31日)
- [14] 中島研吾, 荻田武史, 悪条件問題におけるパイプライン型共役勾配法, 情報処理学会研究報告 (2018-HPC-167), 日本情報処理学会第167回HPC研究会 (那覇, 沖縄, 2018年12月17-18日)
- [15] 坂本龍一, 近藤正章, 中島研吾, 藤田航平,

市村強, HPCアプリケーションにおける低精度演算の積極的利用による電力効率改善の検討, 日本情報処理学会第167回HPC研究会 (那覇, 沖縄, 2018年12月17-18日)

- [16] 中島研吾, 高性能・変動精度・高信頼性数値解析手法とその応用, 第 10 回 自動チューニング技術の現状と応用に関するシンポジウム (ATTA2018), 東京大学弥生講堂 (東京, 2018年12月25日)
- [17] 大友広幸, 横田理央. Tensorコアを用いた Batched QR分解, 第81回情報処理学会全国大会, (福岡, 2019年3月15日)

(5) その他 (特許, プレス発表, 著書等)
なし