jh190042-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（東大情報基盤センター）

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 大 （ Oakforest-PACS， Reedbush-U/H/L，Oakbridge-CX）
- 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（東大）（代表）全体統括・高性能アルゴリズム
- 市村 強（東大）（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（東工大）（副代表）高性能アルゴリズム・精度保証
- 岩下武史（北大）高性能アルゴリズム
- 深谷 猛（北大）高性能アルゴリズム
- 塙 敏博（東大）性能最適化・電力測定
- 伊田明弘（東大）高性能アルゴリズム
- 星野哲也（東大）高性能アルゴリズム・性能最適化
- 坂本龍一（東大）電力測定・自動チューニング（AT）
- 有間英志（東大）電力測定・AT
- 古村孝志（東大）地震シミュレーション
- 藤田航平（東大）高性能アルゴリズム・地震シミュレーション
- 近藤正章（東大）精度保証・電力測定・AT
- 奥田洋司（東大）工学シミュレーション・精度保証
- 森田直樹（東大）工学シミュレーション・精度保証
- 荻田武史（東女大）精度保証・AT
- 田中一成（早大）精度保証・AT
- 尾崎克久（芝工大）精度保証・AT
- 片桐孝洋（名大）高性能アルゴリズム・精度保証・AT
- 櫻井刀麻（名大）精度保証・AT
- 八代 尚（環境研）大気シミュレーション
- 河合直聡（理研⇒東大）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（九大）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・AT
- 堀越将司（インテル）精度保証・AT
- 大島聡史（名大）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

- Osni Marques （ Lawrence Berkeley National Laboratory, USA）高性能アルゴリズム・AT

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，メモリ・ネットワークの階層の深化に対応した通信最適化（Serial，Parallel）とともに省電力・省エネルギー（以下「省電力」）に向けた検討が必要である。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り，従来は画像認識等の計算精度の要求されない分野を対象としていたが，昨今は数値計算において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り，低精度・変動精度使用時，悪条件問題には重要であるが，実問題で現れる大規模疎行列・H 行列への応用例はほとんどない。本研究では，JHPCN システム群の中で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，各アプリケーション（地震，大気科学，量子科学，構造力学）について，Serial・Parallel 通信最適化に着目した高性能最適化手法を各システムにおいて実装し，低精度演算・変動精度演算について検討し，消費電力を測定する。
- 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。更に①の各アルゴリズム，アプリケーションについて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，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 （ JST-CREST 2011-2018 ，https://github.com/Post-Peta-Crest/ppOpenHPC）及び（計算＋データ＋学習）融合を実現する革新的ソフトウェア基盤 h3-Open-BDEC（科研費基盤 S 2019-2023，http://nkl.cc.u-tokyo.ac.jp/h3-Open-BDEC/）に実装し，東大 Oakforest-PACS（OFP），東大 Reedbush（RBH，RBL），東大 Oakbridge-CX（OBCX），東工大 Tsubame-3（TSB3）で公開する。将来的には「富岳」も含む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 OFP, RBH, RBL，OBCX，Tsubame-3 等の大規模システムを有し，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。RBL，OBCX 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，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〔23〕

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。
- 従来の M 疎行列向け精度保証手法〔Ogita, T. et al., Computing, 2001〕は，相対誤差上限の見積もりが厳しめである。

2018 年度研究では，図 1 に示すような不均質場における三次元熱伝導問題を有限体積法で離散化して得られる連立一次方程式を ICCG 法で解くケース（問題規模は 2,097,152=128^3）を共通問題として扱った。以下に，本アプリケーションを使用した 2018 年度研究の研究成果のうち，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に着目して紹介する。

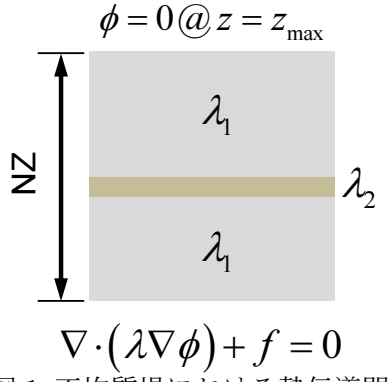


図 1 不均質場における熱伝導問題

1）性能最適化

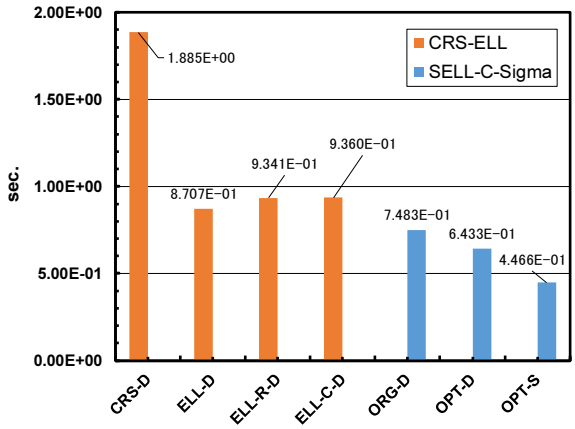


図 2 三次元熱伝導問題（図 1）の KNL での計算時間（64 コア，128 スレッド）（ICCG 法）(-D：倍精度，-S：単精度，OPT：（星野他，2017）の最適化手法適用）

図 2 は図 1 に示す問題を対象として様々な行列格納手法に対して Oakforest-PACS（OFP，Intel Xeon Phi（Knights Landing，KNL））1 ノードを使用して実施した（64 コア，128 スレッド）場合の ICCG 法の計算時間である。CRS⇒ELL で 2 倍以上, ELL⇒SELL-C-σ（図 3）で 15-25%程度の計算時間短縮が得られている。更に倍精度⇒単精度で 30%程度の短縮が達成されている。C の最適値は倍精度で 8-16，単精度で 16 となっている。

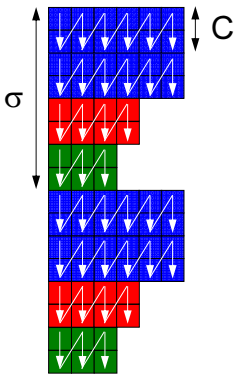


図 3　SELL-C-σ形式，C=2 and σ=8

2）多様な計算精度の適用

従来，科学技術計算には倍精度浮動小数点演算（FP64）を適用してきたが，2018 年度研究では，半精度（FP16）単精度（FP32），4 倍精度（FP128）等を上記②～⑥の各項目について適用し，様々な CPU，GPU（Intel Xeon/Broadwell（BDW），Intel Xeon Phi（KNL），NVIDIA Tesla P100（P100）・V100（V100））を使用して効果を検討した。単精度演算の適用により，倍精度と比較して反復回数等の増加は見られるものの，計算時間は 15～50%程度減少することが確認された。計算時間の減少率はアルゴリズム，アーキテクチャ，最適化の度合いによって異なっており，これらの関係について更なる検討が必要であることが明らかとなった。

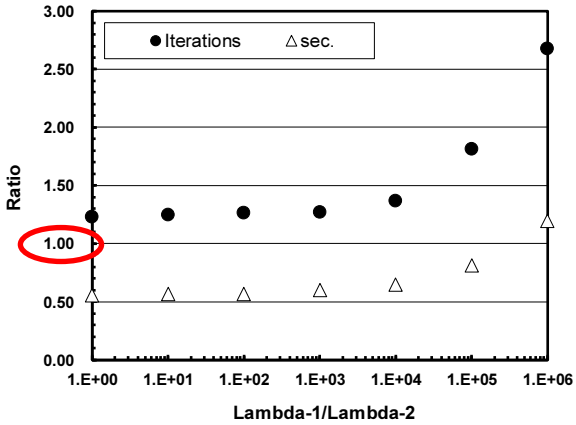


図 4 単精度／倍精度演算時の反復回数比とλ_1/λ_2の関係（BDW）

図 4 は，図 1 に示す問題を対象として，λ_2 を変化させた場合の Reedbush-U の 1 ノード（Intel Xeon Broadwell-EP（BDW），36 コア）を使用した場合の計算結果である。反復回数と ICCG 法の計算時間について，倍精度演算の場合を 1 として，単精度演算の値を示している。単精度演算の反復回数は若干増加するものの，計算時間は半分程度になっている。λ_1/λ_2 が増加し，係数行列の条件が悪化すると，単精度演算の反復回数は増加し，計算時間も倍精度の場合を上回るようなケースがある。また，λ_1/λ_2 が増加時は単精度演算による解の誤差も増加し，図 4 の$\lambda_1/\lambda_2=10^6$ のケースでは 10%にも達している。倍精度⇒単精度⇒半精度によって扱うことのできる値の範囲，有効桁数は減少するため，低精度演算は，変数値の変動幅が大きい問題，大規模問題は不適当である。

図 5 は図 1 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，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。

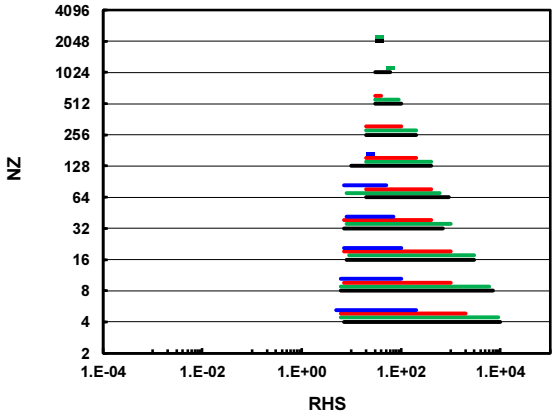


図 5　V100 上で単精度演算（前処理部分のみ半精度）を適用した場合の収束解が得られるパラメータの範囲（RHS：右辺，NZ：Z 方向メッシュ数，$\lambda_1/\lambda_2=10^0$, 10^1, 10^2, 10^3）

3）精度保証

2018 年度研究では，M 疎行列向けに提案された精度保証手法〔Ogita, T. et al., Computing, 2001〕を，図 1 の問題に適用した。求解と精度

保証を同時に実施するアルゴリズムを提案し，別々に計算する場合と比較して，20%程度計算時間を削減できることが示された。この手法では，$\lambda_1/\lambda_2=10^6$ の場合，倍精度演算でも最大相対誤差が 10^0 のオーダーになってしまうことから，やや厳しすぎる判定であり，より精密に誤差の最大値を推定する手法の研究開発が必要である（図 6）。

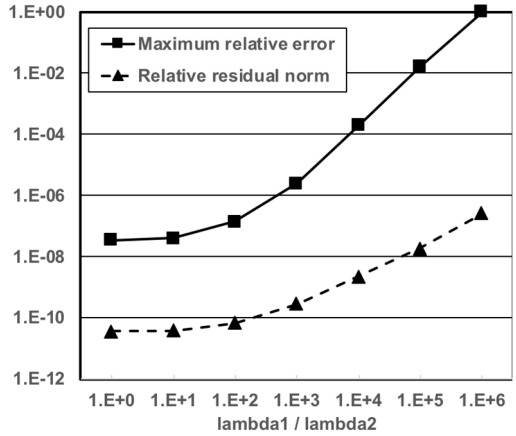


図 6　λ_1/λ_2 と最大相対誤差（実線）と相対残差ノルム（破線）の関係，図 1 の問題，倍精度

4）消費電力・エネルギー測定

図 1 の問題に対して，精度（倍精度・単精度），データ配置方式（Coalesced，Sequential），CM-RCM 法の色数（8，32，128）等を変化させて測定した場合の，計算時間と消費エネルギー（Joule）の値を図 7 に示す。

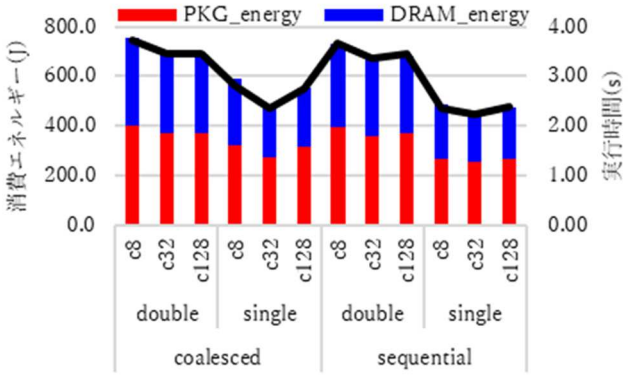


図 7　倍精度(double)を単精度(single)化した場合の消費エネルギーの比較

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，これに比例して消費エネルギー（Joule）も大きく削減された。また，Coalesced より Sequential の方が削減率が大きい。CM-RCM の色数を増やし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。

5.　今年度の研究成果の詳細

2019 年度の重点実施項目は：

- 反復改良法による低精度演算安定化（③）
- 大規模分散並列問題で局所的・動的に演算精度を変動させる手法の検討（③，④）
- 単精度より低い精度の演算の有効性の検討（⑤）
- 疎行列・H 行列を係数行列とする実問題向けの精度保証手法の開発（⑦）
- 様々な演算における消費電力の系統的測定（⑧）
- 最適精度自動選択へ向けての検討（⑦, ⑧, ⑨）

である。括弧内は，4. に示した①～⑨の研究開発項目である。予備的検討に留まった項目もあるが，当初予定の目標は順調に達成しており，研究成果，知見も多数得られている。以下①～⑨の各研究項目について順番に説明する。

①ステンシル演算：構造格子

変動精度を用いた効率的なステンシル計算の実現のためには，同じステンシル計算の問題で，計算精度が異なる場合における，時空間タイリングの効果を明らかにする必要がある。そこで，今年度は，計算精度が異なるステンシル計算のプログラムに，我々がこれまでに研究を行ってきた，異なるタイルレベルの並列性を持つ各種の時空間タイリングの手法を適用した，ベンチマークプログラムの整備を実施した。ベンチマークプログラムの第一段階として，3 次元空間における熱伝導方程式（拡散方程式）を差分法で離散化した場合に得られる，7 点ステンシル型の計算を対象とし，倍精度浮動小数点（FP64），単精度浮動小数点（FP32），半精度の浮動小数点を想定した short 整数型（INT16），の三種類の演算精度（データ型）と，時空間タイリングなし，および，1 次元，2 次元，

3 次元のタイルレベルの並列性を持つ時空間タイリングを適用した，計 4 パターンの実装とを組み合わせて，合計 12 パターンの計算カーネルを実装し，問題サイズや時空間タイリングにおけるタイルサイズ等のパラメータを自由に変化させて，統一的に性能を評価することができるベンチマークプログラムを開発した。

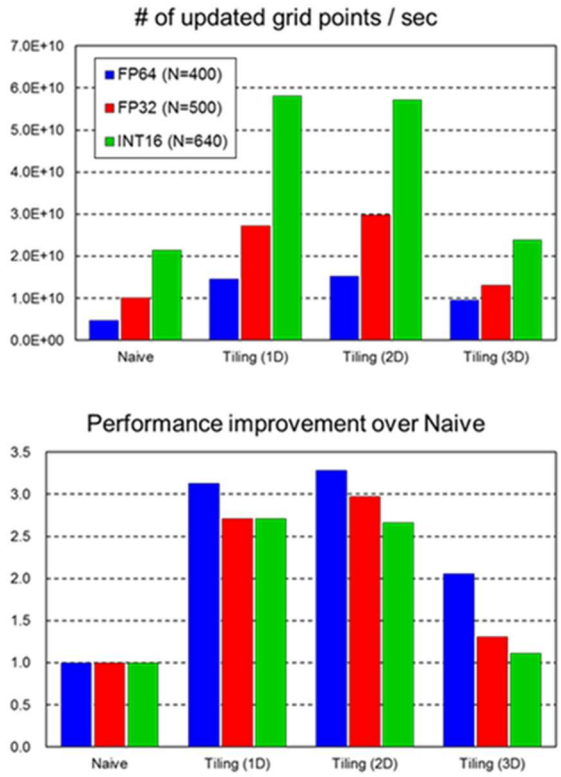


図 8　Oakbridge-CX（1 ノード）での結果
（上：絶対性能，下：タイリングの効果）

東大 Oakbridge-CX の 1 ノード（56 コア）上で，開発したベンチマークプログラムを用いて性能評価を実施した結果を図 8 に示す。データサイズが等しくなるように，各演算精度における問題サイズ N（格子点数：N^3）を設定し，性能（単位時間当たりに更新した格子点の数）と，各種時空間タイリング手法による性能向上の効果を検証した。なお，タイルサイズ等のパラメータは，過去の研究を参考にして設定している。図 8 から分かるように，演算精度に関わらず，時空間タイリングが効果的である。このことから，可能な範囲で低精度演算を利用し，同時に，適切な時空間タイリング手法を組み合わせることで，大きな性能向上が期待できる。一方で，演算精度が異なる場合に対する，タイリングパラメータのチューニング方法が確立できていないので，今後の課題である。

変動精度演算を想定した時空間タイリング付きのステンシル計算のベンチマークプログラムの開発の第一段階は完了した。5 節で提示した結果が示すように，本ベンチマークプログラムを利用することで，異なる演算精度間での性能評価を効率的に行うことが可能となるため，意義のある結果である。今後は，このプログラムを用いて，パラメータのチューニング方法の検討や，消費電力性能を含む詳細な性能評価を行う予定である。また，2018・2019 年年度に対象とした Seism3D や，我々が過去の研究で取り扱った 3D FDTD 法等の，実アプリケーションを想定したベンチマークプログラムの拡充も計画している。

②ステンシル演算：準構造格子

GPU 上での NICAM のフルコード単精度実行を目指し，流体力学計算部分に引き続いて単精度化された諸物理過程の OpenACC による GPU 最適化を進めた。NICAM には水の相変化と落下を解く雲微物理過程，短波・長波放射による加熱・冷却を解く大気放射過程，格子スケールよりも小さい大気運動による混合を解く乱流過程などの諸物理過程が導入されており，それぞれの過程について複数のスキームを実装している。このうち，最も頻繁に利用する主要スキーム 3 つについて，先行して単精度化が進められてきた。それぞれ，シングルモーメント雲微物理スキーム NSW6，ブロードバンド大気放射スキーム mstrnX，乱流境界層スキーム MYNN である。これらのスキームの呼び出し部分の前後でドライバモジュール内の倍精度配列から単精度配列へのコピーを行い，膨大なソースコードの全体を変更することなく段階的な単精度化が進められた。OpenACC の適用にはこの単-倍アダプタをそのまま利用し，デバイスとのコピーイン，コピーアウトを実現した。対象とした物理過程の計算区間（コピー部分を除く）をそれぞれ FX100，Oakforest-PACS，Reedbush-L で測定した結

果を図 9 に示す。３つのスキームはアルゴリズムの違いやコーディングの違いによって，それぞれ異なる結果の傾向を示した。NSW6 については，OFP での計算性能に劣化がみられた。これは主に巨大なループボディを持ち，多量のスカラー変数をループ内で利用する区間が原因であり，レジスタスピルが頻発したことによるものであった。FX100 ではレジスタ数が多いために起こらなかった。mstrnX については，Reedbush で顕著な性能劣化がみられる。これは巨大な配列の２次元目を切り出して，順にサブルーチンに渡して計算を行う区間が原因であった。１次元目の配列サイズは GPU で計算するには十分な大きさであり，データは GPU 上に保持されたまま計算が行われていたので，GPU 上での計算は想定された速度が得られていた。しかし，サブルーチンの外で２次元目のループが回転するたびに OpenACC のカーネル起動が頻発し，そのオーバヘッドが無視できないほど大きいことがわかった。OpenACC のカーネルはサブルーチンをまたぐことが出来ない。上記２つの性能劣化を解決するためには，コードリファクタリングを行う必要がある。

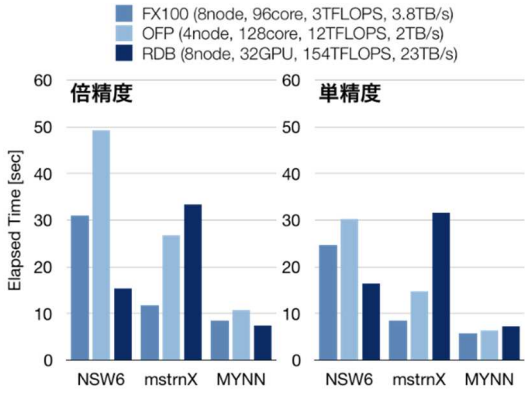


図 9　FX100, Oakforest-PACS, Reedbush-L を用いた NICAM 物理過程の計算時間評価，それぞれ倍精度計算（左），単精度計算の結果を示す FX100 ではスレッド数 12 でハイブリッド並列を用いており，OFP ではスレッド並列は用いていない

また，各スキームのアダプタ部分でのデータコピーは特に GPU 上で無視できないほど時間がかかっている。今後は諸物理過程ドライバ全体を GPU 上で計算するよう，OpenACC 適用領域を拡げていく必要がある。倍精度と単精度の比較においては，メモリによって計算速度が律速されている場合に高速化し，特に OFP で mstrnX は 1.8 倍高速化した。一方で Reedbush（GPU）では単精度化による高速化がほとんどみられなかった。これは前述のカーネル起動のオーバーヘッドと並列度の不足による性能頭打ちに起因すると考えられる。加えて，⑧消費電力測定と連携し，NICAM の電力測定を Reedbush 上で試行した。また，⑩自動チューニングと連携して，NSW6 を題材にサブルーチン内の任意の変数の浮動小数点精度型を変更し，計算結果と計算速度を同時に評価する手法についての具体的なテスト実験を進め，アプリケーション開発側の要望をフィードバックし

NICAM を用いたステンシル計算の主要部分に関する評価は 2018 年度の段階で一定の成果を得ることが出来たが，大規模シミュレーションを実現するためには，巨大な実アプリケーション全体に変動精度演算を適用する必要がある。今年度は諸物理過程の GPU 最適化，混合精度化を進めたが，その中でいくつかのプログラムコードの構造的問題が浮き彫りになった。今後は得られた知見をもとにコードの改良を進め，大規模シミュレーションと消費電力性能評価を進めていく予定である。また，自動チューニング手法の研究と連携し，ループ内に倍精度と単精度の計算を混在させざるを得ない時に，SIMD 等での高速化を両立できるコーディング等についても検討を進めていきたい。

③疎行列演算（混合精度）

1）機械学習と混合精度・反復改良法を利用した連立一次方程式解法〔21,25〕

メニーコアプロセッサを用いた HPC における近年の技術的課題のひとつは消費電力であり，省電力かつ省エネルギーなアルゴリズムの開発，実用化が期待されている。混合精度演算については既に多くの研究があるが，低精度利用時の精度保証の研究は実施されているものの，実問題での大規模疎行列への応用例はほとんどない。GPU のライブラリ cuSOLVER に Iterative Refinement のサブ

ルーチンが追加されるなど，混合精度を用いた Iterative Refinement の手法も近年注目されているが，Iterative Refinement は必ずしもすべての行列に有効ではなく，Iterative Refinement 内にある反復解法の収束判定値も適切な値を設定しないと効果が得られない。また，機械学習への期待から，疎行列を対象として最適な数値計算ライブラリ選択に関する研究が行われているが，Iterative Refinement への応用例はない。

本研究では解の精度を改善するための反復改良法(Iterative Refinement)に着目し，有限要素解析で生成される大規模疎行列連立一次方程式の求解のための，高精度な混合精度演算手法を開発することを目的とする。

Iterative Refinement のアルゴリズムではインナーループで単精度共役勾配法を実行するため，本研究の準備フェーズとして，単精度演算，倍精度演算それぞれで共役勾配法の計算を実行し，収束判定条件を変化させて反復回数・計算時間を計測した。SuiteSparse Matrix Collection より正定値対象の疎行列を選び，右辺ベクトルは厳密解がすべて 1 になるように定めた。また，プログラム中で疎行列は全て CRS 形式に変換して処理している。また，単精度演算を用いた共役勾配法の計算では，係数行列・右辺ベクトル共に単精度で計算している。

結果は次の 3 ケースの特徴に分けられることがわかった。

- **<ケース 1>** 反復回数は単精度も倍精度もほぼ等しく，計算時間が単精度の方が常に短い
- **<ケース 2>** 単精度の方が反復回数は多いが，収束判定値が緩いときは単精度の方が計算時間が短い
- **<ケース 3>** 倍精度が単精度よりも反復回数も少なく計算時間も短い

また，共役勾配法の 1 反復にかかる単精度，倍精度それぞれの計算時間をさまざまな行列に対して測定し，単精度に対する倍精度の時間増加率を調査した（図 10）。図 10 の横軸は行列の番号であり，便宜上，増加率の昇順にソートしている。1 反復あたりの時間増加率は，行列でいかにばらつきを生じるかがわかる。

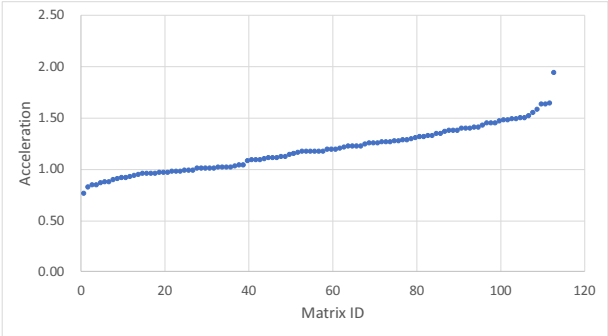


図 10　CG 法 1 反復あたりの時間増加率

次に，混合精度演算を用いた Iterative Refinement の数値実験を行った。混合精度演算を用いた Iterative Refinement のアルゴリズムを図 11 に示す。

Algorithm 2 Iterative Refinement

A を単精度行列に変換
$\boldsymbol{c}_0 = \boldsymbol{b}$ ($\boldsymbol{c}_i$ は単精度ベクトル)
for $i = 0...$ **do**
　単精度 CG ソルバ　▷ 収束判定値 η を設定し，
　　$A\boldsymbol{z}_i = \boldsymbol{c}_i$ の単精度の解 $\boldsymbol{z}_i$ を求める．
　$\boldsymbol{x}_0 = \boldsymbol{z}_0$, $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{z}_{i+1}$
　　▷ 倍精度 $\boldsymbol{x}$ のアップデート
　$\boldsymbol{r}_i = \boldsymbol{b}_i - A\boldsymbol{x}_i$
　if $\|\boldsymbol{r}_{i+1}\| < \epsilon$ **then**
　　Break　▷ 倍精度の解 $\boldsymbol{x}$ を得る．
　end if
　$\boldsymbol{c}_{i+1} = \boldsymbol{r}_{i+1}/\alpha_i$　▷ $\|\boldsymbol{r}\|$ を 1 にスケーリング
end for

図 11　Iterative Refinement 最終的な収束判定値を 10^{-8} に設定しインナーループの単精度 CG 法の収束判定値 η を変化させて計算時間を比較した（図 12）

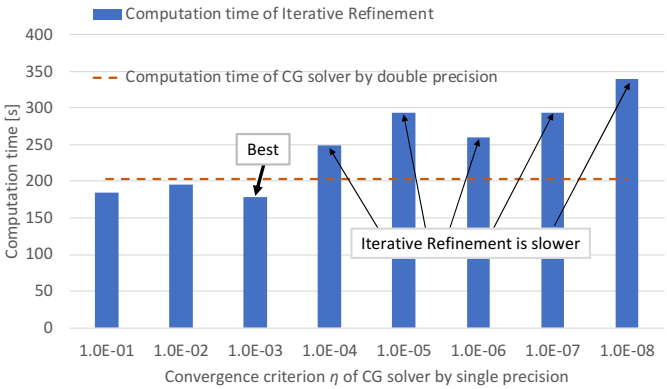


図 12 インナーループ収束判定値 η の影響

図 12 に Iterative Refinement による計算結果の例を示す。このとき用いた疎行列は行列サイズが 83,334，非ゼロ要素数が 6,010,480 の行列である。図 12 より，インナーループの単精度 CG 法の収束判定値 η の値を 10^{-1} に設定した時が最も計算が速く，η の設定値によっては倍精度 CG 法に比べて遅いケースがあることがわかる。

このように Iterative Refinement は適切なインナーループの単精度 CG 法の収束判定値 η を設定しないと倍精度 CG 法に比べて良いパフォーマンスが得られないことがわかった。

今後は，機械学習を用いて行列の情報から η の値を設定することを考える予定である。

2）局所的な変動精度適用

2018 年 7 月に出席した国際会議 WCCM では，問題の条件に応じて局所的に演算精度を変動させる可能性について指摘された。2019 年度は，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の開発を進めている。まず，領域分割に基づく並列有限要素法において，領域毎に異なる精度を適用した場合の検証を，並列有限要素法による三次元構造解析における加法シュワルツ法（Additive Schwartz Domain Decomposition（ASDD），図 13）に基づく ICCG 法

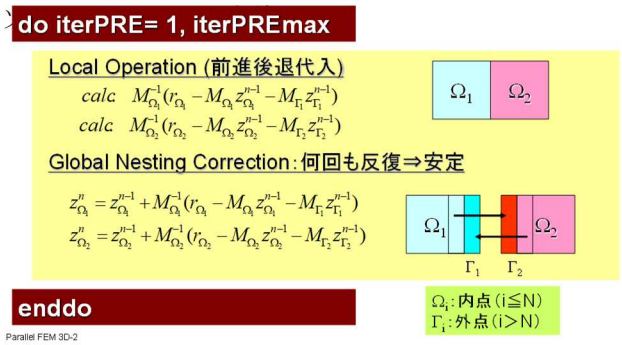


図 13　Additive Schwartz Domain Decomposition (加法シュワルツ法)による並列前処理安定化

図 14 に示すような規則形状を 8 分割した場合，6 領域については倍精度，2 領域については混合精度（前処理・加法シュワルツ法部分のみ単精度）の疎行列ソルバーを適用し，均質問題において，全 8 領域倍精度の場合と同じ反復回数で同じ解を得た。2020 年度以降は不均質問題，動的負荷分散に継続して取り組む予定である。

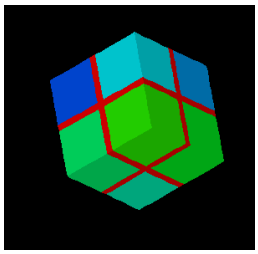


図 14　並列有限要素法（8 分割）

④疎行列演算：固有値分布を保存する行列縮退法について

有限要素法などから導かれる疎行列を係数行列とする連立一次方程式を反復法で解く場合，収束条件は係数行列の固有値分布に依存し，固有値分布を改善するための様々な前処理手法（Preconditioner）が研究されている。近年，機械学習等の手法を用いて問題，すなわち係数に応じた最適な前処理手法を選択する研究が盛んになっているが，係数行列の固有値分布は最適前処理手法選択にあたって重要な指標となる。大規模な係数行列の固有値分布の算出には連立一次方程式を解くのと同等のコストを要するため，効率的な算出手法が必要となる。本研究では，元の大規模行列と同様の固有値分布を保ちながら，行列を縮退（reduction）する方法について検討した。

本研究では，以下に示す 3 種類の行列縮退手法を提案している：

- Random Diagonal Projection Method（RDPM）
- Random Column Basis Projector Method（RCBPM）
- Mixed Projection (MP) of RDPM and RCBPM

第 1 の手法（RDPM，ランダムな対角投影法）は固有値分布のスペクトルを，保持するものの最大固有値の計算精度が低い（図 14）。逆に第 2 の方法（RCBPM，列単位投影法）は固有値分布の

スペクトルは不正確であるが，最大固有値をより正確に求めることが出来る（図 15）。

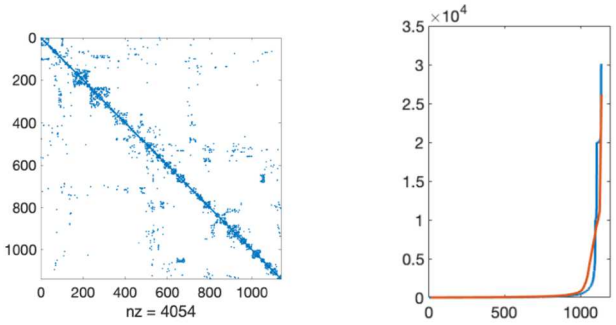


図 14　第 1 の手法（RDPM，ランダムな対角投影法）による固有値分布計算（—：固有値を実際に求めた場合，—：RDPM による計算）（1138_bus（SuiteSparse Matrix Collection），Power Network）

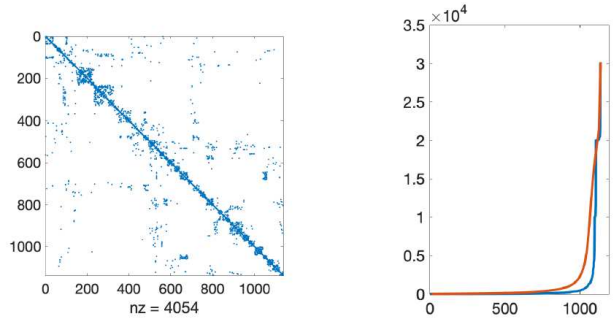


図 15　第 2 の手法（RCBPM，列単位投影法）による固有値分布計算（—：固有値を実際に求めた場合，—：RCBPM による計算）（1138_bus（SuiteSparse Matrix Collection），Power Network）

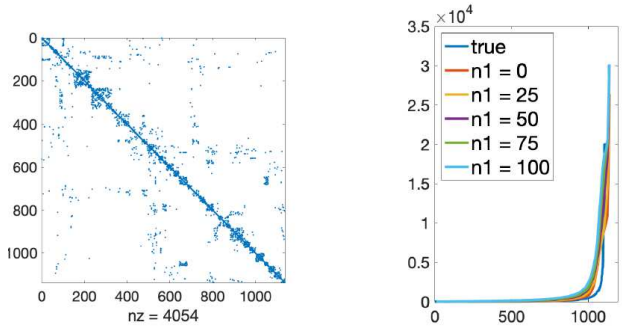


図 16　第 3 の手法（MP，混合投影法）による固有値分布計算（—：固有値を実際に求めた場合，—：MP による計算）（1138_bus（SuiteSparse Matrix Collection），Power Network）

第 3 の方法は第 1 の手法（RDPM，ランダムな対角投影法）は第 2 の方法（RCBPM，列単位プロジェクション）を組み合わせる混合投影法によって，最大固有値，固有値分布スペクトルの両者を正確に模擬できるような行列縮退手法を提供するものである。SuiteSparse Matrix Collection の行列群に対して，第 3 の手法を適用することによって，100×100 程度の行列で大規模な疎行列の固有値分布を再現できることを確認した（図 16）。

2019 年度は，固有値分布を保存する行列縮退について予備的な評価を実施した。今後は，縮退処理の並列化に取り組んで行く予定である。

⑤疎行列演算：Adaptive CG

1）精度変動演算用 FP21 の CUDA/OpenACC 実装〔3, 36〕

比較的条件の良い問題では，倍精度演算(FP64)のかわりに単精度(FP32）を使うことによって計算時間を短縮できる場合があるが，機械学習などで使われる半精度(FP16)は有効桁数が 3 桁程度のため，実問題に使用することは困難である。本研究では FP16 と FP32 の中間的な特性を持つデータ型である FP21 を定義し，有限要素解析における前処理において活用した。FP21 は符号 1 ビット・指数部 8 ビット・仮数部 12 ビットの合計 21 ビットからなり，FP32(符号 1 ビット・指数部 8 ビット・仮数部 23 ビットの合計 32 ビット)と指数部のビット数が同一となる。FP21 は FP32 と同じレンジを持つもののデータサイズが 1/1.5 となるため，メモリバンド幅ネックのカーネルにおいては実行時間が 1/1.5 になると期待される。なお，現在の主要な計算機においては FP21 の演算器はサポートされていないため，変数のメモリへの格納時のみに FP21 を使い，計算時には FP21 を FP32 に変換して演算する。上記を CUDA にて実装し V100 GPU で計測した結果，想定通りの性能が得られ，FP21 の有効性が示された。本研究ではさらに上記を OpenACC で実装し，ソースコードを公開することでより多くのアプリケーションで FP21 を活用できるようにした〔36〕。性能計測の結果，実際の地震アプリケーションに組み込んだ際に OpenACC 版において CUDA 版と同等の性能が得られることを確認した〔3〕。

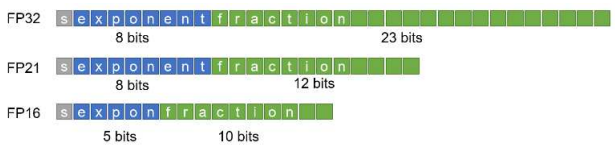


図 17　低精度データ型 FP21 の模式図

2）AI 用低精度演算器の Equation-based Modeling への活用〔4〕

2019 年 11 月現在で Top 500 リスト一位の米国 Summit は主に AI や data analytics 用に設計された Tensor Core を搭載する V100 GPU を多数搭載している。この Tensor Core では，FP16 において小規模な行列行列積を通常の FMA 演算器に比べて 4 倍のピーク性能で実行できるため，物理シミュレーションに活用することで性能改善の可能性がある。その一方で，演算パターンが行列行列積に限定されている点，演算精度が低い点，演算性能が高いためにデータアクセスがボトルネックになりやすい点，の 3 点が制約となり equation-based modeling において使用が普及するに至っていない。そこで本研究では，equation-based modeling の一例である非構造格子有限要素法を対象にこのような演算器を活用できるアルゴリズムの構築例を示す。

通常の非構造格子有限要素法においては各要素の形状と節点の接続状況に応じた非均一な基底関数が使われるため，ランダムアクセスなどの複雑な計算が生じ，結果として行列行列積に特化した Tensor Core の高い性能を発揮することが難しい。そこで本研究では，AI の学習において用いられる計算パターンと類似の計算が生じるよう，ソルバー内で局所的・規則的な基底関数展開を使う。ここでは，四面体二次要素メッシュを対象とした陰解法ソルバーにおいて adaptive conjugate gradient method を使い，前処理部に構造格子からなる共役勾配法ソルバーを使う。このようにすることで，ソルバー内の大部分の計算を構造格子における演算に移すことができ，その内部の主要演算カーネルとなる行列ベクトル積において，小規模な行列・行列積を用いることが可能となる。また，使用は前処理に限定されるため，低精度演算を用いても十分な精度の解を得ることができ，Tensor Core の FP16 演算を使うことができるようになる。また，レジスタ上でデータを使いまわす等の工夫をすることでデータアクセスコストを抑えた。開発手法により Summit のほぼ全系にあたる 4544 ノード (V100 GPU 27,264 枚)を使って 1.67×10^{12} 自由度の問題を解いたところ，ベースラインソルバー（3×3 ブロックヤコビ前処理による共役勾配法ソルバー）比で 17.6 倍速，V100 GPU 用にチューニングを施した SC14 ゴードンベルソルバーに比べて 3.89 倍の高速化を達成した。この際のピーク性能はソルバー全体で 416 PFLOPS，行列ベクトル積部分で 1.10 Exa FLOPS となった〔4〕。このような，アルゴリズム的に equation-based modeling と AI を融合させる新しい HPC の方向性は他のアプリケーションへも活用可能であると考えられ，今後の発展が期待される。

⑥ H 行列〔5,6,17,18,27,30〕

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のためにまとめて 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。混合精度演算を行う場合，H 行列の中で最も工夫を要するのは，密行列を低ランク行列に圧縮する部分である。Randomized SVD や Interpolatory Decomposition など様々な圧縮方法が提案されているが，いずれの手法も QR 分解を基礎としているため，2019 年度の JHPCN 課題では，多くの小さな QR 分解を GPU 上で並列に行う batched QR 分解を開発した。この際に，Volta 世代の GPU に搭載されている TensorCore を利用することで 10TFLOPS を超える性能を得ることができた〔27〕。ただし，行列の大きさは 16x16 に限定されており，このままでは H 行列の圧縮に用いることは難しい。今年度はこれを TSQR に拡張し，任意の大きさの行列の低ランク近似を高速に行うための QR 分解の枠組みを開発した〔6，30〕。Z 図 18 に行列サイズを変えたときの TSQR の V100 上での計算時間を示す。TSQR では列数 16 までの行列に対する QR 分解を行い，

行数が大きい場合に NVIDIA が開発している線形代数ライブラリ cuSOLVER と比較して高速に QR 分解を行うことができていることが確認された。TensorCore は積を計算する 2 つの行列を半精度として入力する必要があり，TensorCore を用いて単精度行列積を計算する場合は半精度への変換を行う必要がある。これにより行列積の計算精度が単精度演算器で計算した場合と比較して劣化するが，半精度への変換により失われる仮数部を別途保持し，これを用いて行列積の計算精度を補正することができる。この精度補正計算を用いた場合，TensorCore による計算量は 3 倍となるが，TSQR に利用した場合計算時間は 2 倍程度にとどまっており，cuSOLVER を用いた場合と比較しても高速に QR 分解が行われていることが確認された。一方で計算精度として相対残差（図 19）及び直交性（図 20）を見ると，精度補正を行うことで cuSOLVER を用いた単精度 QR 分解により近い精度で計算を行えていることが確認された。QR 分解を行うのに必要な作業用メモリ量では，本研究での TSQR は cuSOLVER を用いた QR 分解と比較し 40%～80% の作業用メモリを削減できており，より大きな行列に対する QR 分解が可能であることが確認された。

TSQR の実装では Householder 変換を用いた QR 分解を用いた。この QR 分解では通常陽的に Householder 行列を作らず，行列ベクトル積とベクトルの直積を用いて計算を行うことで理論計算量を削減する。TensorCore を用いてこの QR 分解を行う場合，陽的に Householder 行列を作りベクトルの直積と行列行列積を用いた方が高速に計算を行うことができることが確認された。このようにアルゴリズムを変更することが行列行列積を計算する回路である TensorCore をより効率的に利用する上で重要となる。

開発した TSQR では列数が 16 という制限があるため，この制限を解消し一般的な行列に対する QR 分解を行うために Block QR を用いる。Block QR では Q の直交性が劣化することが知られているが，再直交化を行うことで直交性の劣化を抑えることができる。しかし再直交化を行うことで計算量は増加するため，並列性の抽出・計算のオーバーラップを行うことで増加する計算時間を隠蔽することが重要となる。

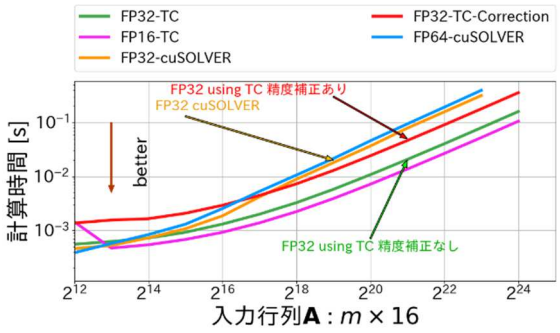


図 18　TSQR の計算時間

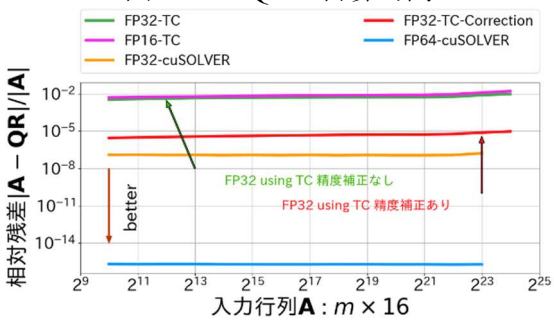


図 19　TSQR の相対残差

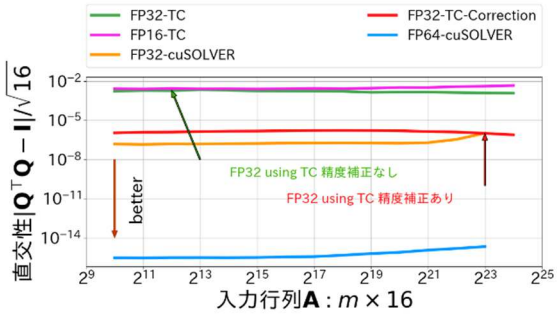


図 20　TSQR の直交性

一方で，開発した TSQR は入力行列がフルランクであることを前提としており，フルランクでない行列が入力された場合には計算が破綻する。Randomized SVD で行う QR 分解では，フルランクでない行列が入力される場合が十分に想定されるため，計算の破綻を防ぐ，または破綻を検知する仕組みを導入する必要がある。また，今年度の研究により TensorCore による精度補正単精度行列積を用いた TSQR では，cuSOLVER と比較して計算精度に 10 進数で 1 桁程度の精度の劣化が確認されている（図 19，20）。この精度劣化により

Randomized SVD の近似精度がどう影響を受けるかを調査する必要がある。

TensorCore を用いた精度補正単精度行列積の既存手法では TensorCore による計算量が 4 倍となる。本研究ではこの精度補正計算の一部を省略し計算量を 3 倍に削減することで高速に精度補正計算を行えており，QR 分解の精度の劣化も確認されていない。今後この精度補正計算の省略による計算精度への影響を注意深く調べるとともに，精度補正を効果的に行える行列の条件や逆に精度補正を行いにくい行列の調査，行いにくい行列に対するアプローチの調査を行っていく。

今後は H 行列の分散並列化における StarPU や ParSEC などのランタイムを用いた非同期実行〔17〕や H 行列の QR 分解〔18〕，GPU 並列化における独自のランタイムを用いた非同期バッチ実行〔5〕などへと TensorCore による混合精度の精度保証を拡張していく予定である。

⑦精度保証

1）実問題向け精度保証手法の開発（疎行列）〔10,11〕

実問題に精度保証を適用するための基盤として，三次元 Poisson 方程式の数値解の精度保証を行った。数値シミュレーションでは，図 21 に示すように，微分方程式を離散化し，得られた連立一次方程式を解くことが多いが，離散化誤差については，独立に考えることができるため，本研究では，連立一次方程式に対して精度保証を適用する。

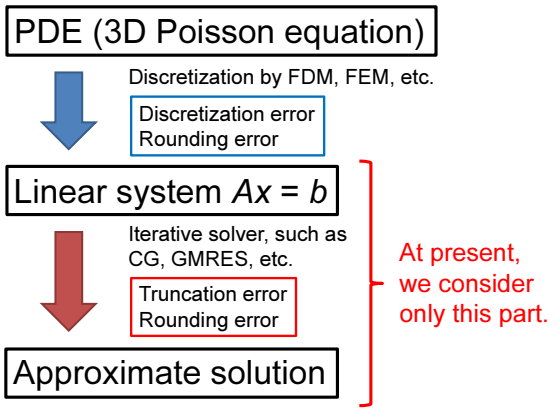
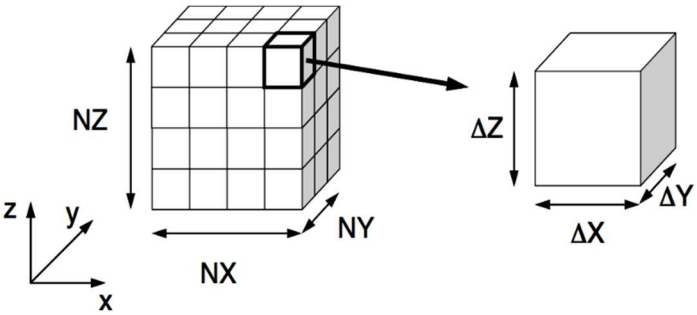


図 21　今回の精度保証の対象

本研究では，有限体積法（7 点差分メッシュ）によって離散化して得られた連立一次方程式に対する精度保証の改善に成功した。解析モデルを図 1，図 22 に示す。



$$-\nabla\cdot(\lambda\nabla\phi)=f,\quad f(x,y,z)=x+y+z$$

（境界は，すべて Dirichlet 境界条件）

図 22　解析モデル（三次元 Poisson 方程式）

係数行列Aが M 行列性を持つ場合，高速な精度保証法〔Ogita, T. et al., Computing, 2001〕を適用可能であるが，誤差限界が過大評価になっている可能性がある。そこで，それを抑制するための方法〔Ogita, T. et al., Computing, 2002〕を適用した。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. 残差$r=b-A\hat{x}$を精度保証付きで計算する。残差の近似値を$\hat{r}$，誤差限界をe_rとする。
v. 連立一次方程式$Az=\hat{r}$を解く。得られた近似解を$\hat{z}$とする。
vi. 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。

$$\|x-\hat{x}\|_\infty \le \|\hat{z}\|_\infty + \frac{\|\hat{y}\|_\infty \|b-A(\hat{x}+\hat{z})\|_\infty}{1-\|e-A\hat{y}\|_\infty}$$

（但し，右辺第 2 項の分母が 0 以下になった場合は精度保証失敗）

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(マルチカラー前処理: 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：

$$Ax = b,\ \ \|b - A\hat{x}\|_2/\|b\|_2 < \varepsilon_1$$
$$Ay = e,\ \ \|e - A\hat{y}\|_\infty < \varepsilon_2$$
$$Az = \hat{r},\ \ \|\hat{r} - A\hat{z}\|_2/\|\hat{r}\|_2 < \varepsilon_3$$

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)
- $\lambda_1 = 1.0$に固定，λ_2を変化させる。
- $\varepsilon_1 = 10^{-12}$，$\varepsilon_2 = 10^{-2}$，$\varepsilon_3 = 10^{-9}$

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：

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

の上限及び相対残差ノルム

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

の上限を求めた。結果を図 23 及び表 1 に示す。

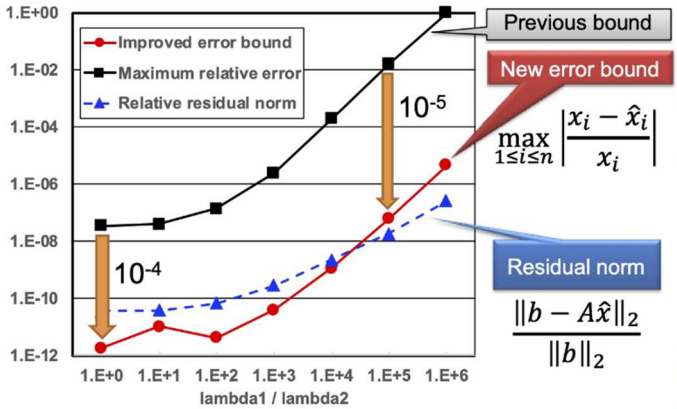


図 23 条件（λ_1/λ_2）毎の最大相対誤差（実線）と相対残差ノルム（破線）

表 1 計算時間（秒）

NX=NY=NZ=128 (n = 2,097,152)	近似解の計算	精度保証
$\lambda_1/\lambda_2 = 1$	3.75	4.47
$\lambda_1/\lambda_2 = 10^6$	6.83	7.85

図 23 から，提案方式によって誤差限界の過大評価が大幅に抑制できていることがわかる。また，表 1 から，近似解と精度保証の計算時間は同程度であることがわかる。

精度保証については，三次元 Poisson 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，実用的かつ高精度な精度保証法の適用に成功した。今後は，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。

2) 不連続係数ポアソン方程式に対する離散化誤差のL^2評価〔31,34〕

図 1，図 22 で説明した不均質場における温度分布モデルに対する離散化誤差の評価を実現した。本評価と前述の連立一次方程式に対する誤差評価を組み合わせることにより，対象モデルの真の解とコンピュータによって得られた数値解のL^2誤差を包括的に評価することができる。対象問題は同モデルを不連続係数ポアソン方程式

$$\begin{cases} -\nabla\cdot(\lambda\nabla\phi) = f & \text{in} \quad \Omega \\ \phi = 0 & \text{on} \quad \partial\Omega \end{cases}$$

として定式化した楕円型偏微分方程式である。ここで$f \in L^2(\Omega)$は任意に与えられた関数とし，λは図 1 で説明した各部分領域で定数値をとる不連続係数である。本境界値問題とその有限要素法で離散化された方程式の解ϕ_hに対して，まず以下の形式での誤差評価を得た。

$$\|\phi - \phi_h\|_\lambda \le C(h)\|f\|_{L^2}$$

ここで，誤差評価式中のノルム$\|\cdot\|_\lambda$は拡散係数の選び方に依存し$\|u\|_\lambda^2 = \sum_k \lambda_k \|\nabla u\|_{L^2(\Omega_k)}^2$で与えられる。また$C(h)$のとり方は離散化手法に依存する。例えば立方体(正方形)メッシュ上で各方向での線

形近似を採用した場合は，メッシュの一辺の長さをhとして

$$C(h) = \frac{h}{\pi}\max_k \lambda_k^{-\frac{1}{2}}$$

と設定すれば良いことを明らかにした。このとき，ノルム$\|\cdot\|_\lambda$の意味での誤差評価はhオーダーであり，またその誤差は領域中で最も小さい拡散係数λ_kに依存する。上記誤差評価式と Aubin-Niche の技巧による式変形により，以下のL^2誤差評価（h^2オーダー）が得られる。

$$\|\phi - \phi_h\|_{L^2} \le C(h)^2 \|f\|_{L^2}$$

例えば，図 1，図 22 の設定下で$\lambda_1 = 1, \lambda_2 = 10^{-3}, h = 10^{-7}$とした場合

$$C(h) \le 7.87 \times 10^{-2}$$
$$C(h)^2 \le 9.78 \times 10^{-3}$$

のように評価できる。これに$f(x,y,z) = x + y + z$のL^2ノルムである$\sqrt{2.5}$を乗じたものが，それぞれ$\|\phi - \phi_h\|_\lambda$と$\|\phi - \phi_h\|_{L^2}$の上界となる。

以上の結果と前述の連立一次方程式に対する誤差評価を組み合わせることにより，所望の真解ϕと数値解$\hat{\phi}_h$（有限要素解のコンピュータによる数値的近似ϕ_h）の誤差を包括的に評価することができる。最も簡単には，連立一次方程式のl^∞誤差評価

$$\|\phi_h - \hat{\phi}_h\|_\infty \le \varepsilon$$

が得られ，かつ先と同様に$\phi_h, \hat{\phi}_h$がメッシュ内で各方向に線形補間されてる場合を考える。この場合，真解ϕと数値解$\hat{\phi}_h$のL^2ノルムの意味での誤差は：

$$\begin{aligned}\|\phi - \hat{\phi}_h\|_{L^2} &\le \|\phi - \phi_h\|_{L^2} + \|\phi_h - \hat{\phi}_h\|_{L^2} \\ &\le C(h)^2\|f\|_{L^2} + \varepsilon\sqrt{|\Omega|}\end{aligned}$$

で与えられる。

2019 年度の研究により，図 1，図 22 にある不均質場における温度分布モデルに対する離散化誤差のL^2誤差評価を実現することができた。本成果を疎行列計算に対する精度保証法と組み合わせることにより，対象問題の離散化誤差を含めた包括的誤差評価を高速に達成する理論基盤が整った。

2020 年度には上記をベースとした手法を実装し，コンピュータ上で表現された実際の有限要素解のすべての誤差（離散化誤差＋丸め誤差）を定量的に評価する。特に，離散化パラメータと演算精度のバランス，例えば「倍精度など特定の演算精度に対してどの程度メッシュサイズhを小さくとることが妥当なのか」という問いに対してある種の指標を与えることが今後の目標である。

また，2019 年度はL^2ノルムの意味での離散化誤差評価を得たが，2020 年度はこれをベースとしL^∞ノルムの意味での誤差評価を目指す。これにより真解ϕと数値解$\hat{\phi}_h$の領域各点での差異を見積もることが可能となる。

⑧消費電力測定：疎行列格納法，問題規模による低精度演算の計算時間，消費電力・エネルギーへの影響〔7, 12, 14, 15, 19, 20, 23, 32, 33, 35〕

本研究では，図 1 のような形状に対して，有限体積法によるポアソン方程式ソルバーから導かれる対称正定な疎行列を係数とする連立一次方程式を不完全コレスキー分解前処理付き共役勾配法（ Preconditioned Conjugate Gradient Method by Incomplete Cholesky Factorization，ICCG 法）を対象として，倍精度・単精度演算の比較，消費電力，消費エネギーの測定を実施し，条件の比較的良い問題では，単精度演算により倍精度の場合と比較して，50～60%程度の計算時間，消費エネルギーで正しい計算を実施することができた〔23〕。

これまで実施してきた研究は，低精度演算の有効性を示すための予備的研究であるが，2019 年度は，問題サイズ，行列格納手法，対象アーキテクチャなどの様々な影響を考慮した検討を実施した。本研究では以下の 5 種類の計算機環境を使用した：

マルチコア・メニィコア CPU

- Reedbush-U（東京大学情報基盤センター）
- OFP-Mini（東京大学情報基盤センター）
- Oakbridge-CX（東京大学情報基盤センター）

GPU

- Reedbush-L（東京大学情報基盤センター）
- P9-V100 クラスタ（東京大学情報基盤センター）

本研究では，CPU については各 1 ノード，GPU については GPU1 基を使用して実施した。表 2 は CPU の各 1 ソケット，表 3 は各 GPU の概要である。OFP-Mini は，最先端共同 HPC 基盤施設（JCAHPC）の運用する Oakforest-PACS と同じ CPU を使用した 8 ノードのクラスタである。また，P9-V100 クラスタは，IBM Power 9 と NVIDIA Tesla V100（V100）から構成されるクラスタである。

表 2　計算機（CPU）各ソケット概要

システム名	Reedbush-U	OFP-mini	Oakbridge-CX
略称	RBU	OFP	OBCX
CPU 名称	Intel Xeon E5-2695 v4 (Broadwell-EP)	Intel Xeon Phi 7250 (Knights Landing, KNL)	Intel Xeon Platinum 8280 (Cascade Lake, CLX)
ソケット当たりコア数	18	68	28
ノード当たりソケット数	2	1	2
理論演算性（GFLOPS）	604.8	3,046.4	2,419.2
主記憶容量（GB）	128	MCDRAM: 16 DDR4: 96	96
メモリ性能（GB/sec）STREAM Triad	65.5	MCDRAM: 490 DDR4: 84.5	101.0
TDP (W)	120	215	205

表 3　GPU 概要

システム名	Reedbush-L	P9 Cluster
略称	P100	V100
GPU 名称	NVIDIA Tesla P100	NVIDIA Tesla V100
理論演算性能（GFLOPS）	5,300	7,500
主記憶容量（GB）	16	32
メモリ性能（GB/sec）STREAM Triad	557	855
TDP (W)	250	300

元のプログラムは Fortarn+OpenMP によって開発されているが，GPU 用にはこれに OpenACC を適用したバージョンを使用している。

電力評価には RBU，OBCX については Intel が提供する CPU の電力制御機能の RAPL を使用した。また OFP については，同じく Intel の提供する Intel pcm-power.x を使用し，GPU（P100，V100）では nvidia-smi を使用した。また，GPU 利用時は CPU の消費電力は測定していない。

本研究では，以下の 3 種類の行列格納形式について検討した（図 24）：

- Compressed Row Storage（CRS）
- Column-Wise な Sliced ELL（ELL）
- SELL-C-σ（C=8, σ =1）（SCS）

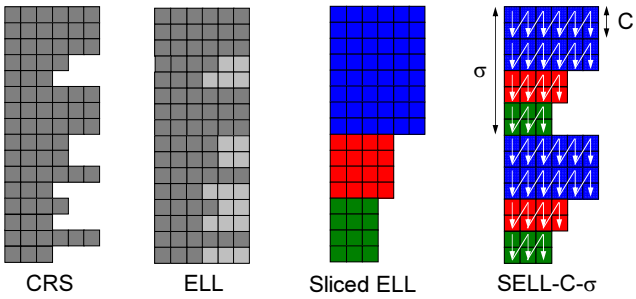


図 24　様々な疎行列格納形式

図 25 は SELL-C-σ（SCS），OPT における前処理計算・前進代入部分の実装方法である。

CRS⇒ELL/Sliced ELL⇒SELL-C-σに従って，計算密度は増加する。また，RBU，OFP，OBCX については SCS を更に最適化してスレッド並列化によるオーバーヘッドを取り除いた実装を適用した。

本稿ではこれを「OPT」と呼ぶ〔星野他，OpenACC を用いた ICCG 法ソルバーの Pascal GPU における性能評価，情報処理学会研究報告（2017-HPC-158-18），2017〕。

(a)

(b)

```
!$omp parallel (…)                                   SCS: SELL-C-σ
    do ic= 2, NCOLORtot-1
!$omp do
      do ip= 1, PEsmpTOT
        iq1= SMPindex((ip-1)*NCOLORtot + ic-1) + 1
        iq2= SMPindex((ip-1)*NCOLORtot + ic)
        ip0= (ip-1)*NCOLORtot + ic
        iq0= (iq1-1)*8
      do ib  = iq1, iq2
        ib0= (ib-iq1)*8
!$omp simd
      do is  = 1, 8
        i= iq0 + ib0 + is
        VAL= W(i,Z)
        do k= 1, 3
          VAL= VAL- AL(ib0+is,k,ip0)*W(itemL(ib0+is,k,ip0),Z)
        enddo
        W(i,Z)= VAL * W(i,DD)
      enddo
      enddo
      enddo
    enddo
!$omp end parallel
```

```
!$omp parallel (…)                                   OPT
(…)
    do ic= 2, NCOLORtot-1

      iq1= SMPindex((ip-1)*NCOLORtot + ic-1) + 1
      iq2= SMPindex((ip-1)*NCOLORtot + ic)
      ip0= (ip-1)*NCOLORtot + ic
      iq0= (iq1-1)*8
      do ib  = iq1, iq2
        ib0= (ib-iq1)*8
!$omp simd
      do is  = 1, 8
        i= iq0 + ib0 + is
        VAL= W(i,Z)
        do k= 1, 3
          VAL= VAL- AL(ib0+is,k,ip0)*W(itemL(ib0+is,k,ip0),Z)
        enddo
        W(i,Z)= VAL * W(i,DD)
      enddo
      enddo
!$omp barrier
    enddo
(…)
!$omp end parallel
```

図 25　前処理計算・前進代入部分の実装方法 (a) SELL-C-σ（SCS），(b) OPT〔星野他，2017〕

RBU, OBCX については 1 ノード (2 ソケット) を使用した。RBU では合計 36 コア (36 スレッド)，OBCX では 48 コア（48 スレッド）を使用した。また OFP では，先行研究における最適な設定である 64 コア (128 スレッド) を使用した。P100, V100 は GPU1 基を使用した。

OFP は表 2 に示すように，通常の DDR4 メモリに加えて，CPU パッケージに直接収められている高速の MCDRAM を使用することが可能である。Cache モードでは，MCDRAM を DDR4 のキャッシュとして使用ができる他，Flat モードでは，DDR4 と MCDRAM を使いわけることが可能である。本研究では，以下の 3 ケースについて検討を実施した：

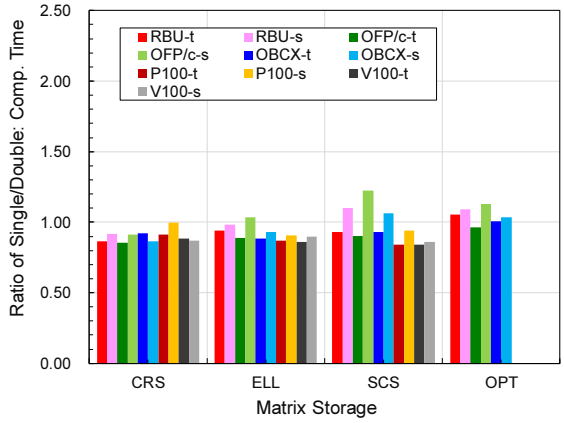
- a:　Cache モード
- b:　Flat モード，DDR4 のみ使用
- c:　Flat モード，MCDRAM のみ使用

問題規模としては：

- Tiny（t）：　32,768 DOF（=32^3）
- Small（s）：　262,144 DOF（=64^3）
- Medium（m）：　2,097,152 DOF（=128^3）
- Large（l）：　16,777,216 DOF（=256^3）

の 4 種類を実施した。Large の場合でも，P100, OFP (MCDRAM) の各メモリに充分収まっている。

(a)



(b)

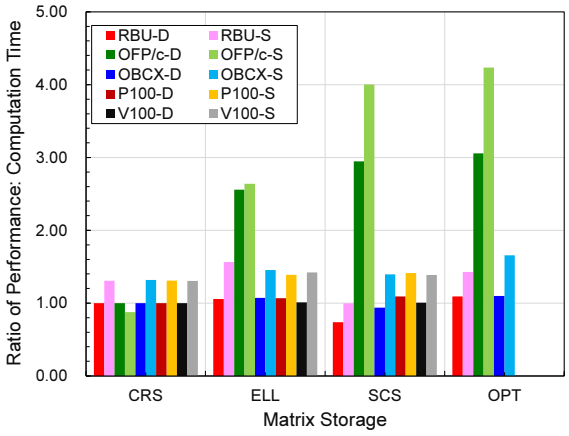


図 26　ICCG 法ソルバーの計算性能，「倍精度・CRS」の計算時間によって無次元化，(a) Medium，(b) Large，1.00 より大きい場合は「倍精度・CRS」よりも ICCG 法の計算時間が短い

本研究では，全計算を倍精度で実施した場合（Double）と全て単精度実施した場合（Single）の 2 種類の計算を実施した。単精度のみの場合は，倍精度の場合と比較して 20%程度反復回数が増加している。

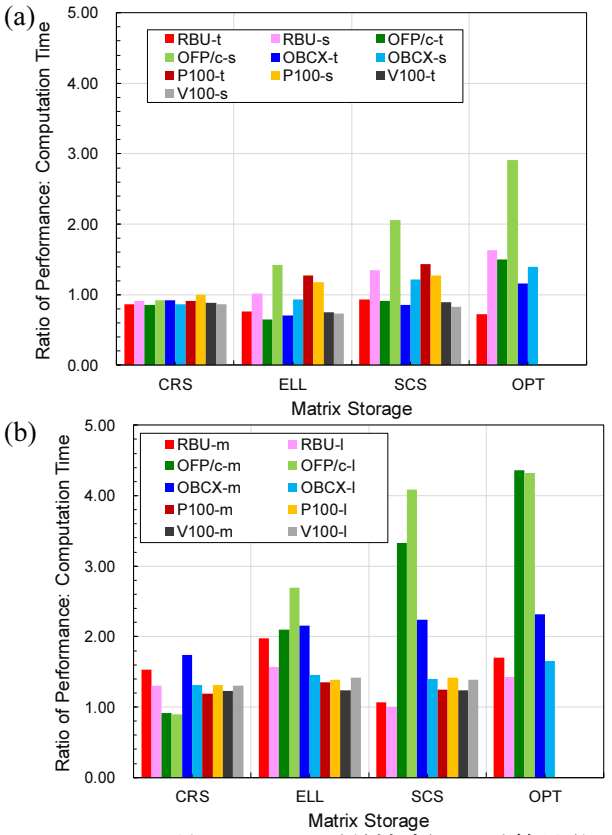


図 27　ICCG 法ソルバー（単精度）の計算性能，「倍精度・CRS」の計算時間によって無次元化，(a) Tiny，Small，(b) Medium，Large : 1.00 より大きい場合は「倍精度・CRS」よりも ICCG 法の計算時間が短い

図 26 は（a）Medium，（b）Large の場合に ICCG 法部分の計算時間について，各計算機環境，各問題サイズにおける，「倍精度・CRS」の計算時間によって無次元化したものである。値が 1.00 より大きい場合は，「倍精度・CRS」より ICCG 法部分の計算時間が短いことを示している。RBU，OBCX は CRS⇒ELL で倍精度，単精度ともに計算性能が向上しているが，SCS でやや遅くなり，OPT でまた速くなっている。RBU，OBCX（ともに Intel Xeon プロセッサ）では倍精度⇒単精度の速度向上は特に Medium サイズにおいて顕著である。一方 OFP では CRS⇒ELL⇒SCS⇒OPT と順調に性能向上が見られる（元々SCS，OPT は OFP 向け最適化であるが）。OFP においては倍精度⇒単精度の速度向上は RBU，OBCX と比較して少ない。特に CRS で速度向上が少ないのはベクトル化，SIMD 化が不十分なためと考えられる。GPU（P100，V100）の傾向は似ており，また行列格納方法による差異は少ないが，三者の中では ELL が比較的良い。また倍精度⇒単精度の速度向上は 20%程度である。

図 27 は単精度を使用して実施した ICCG 法部分の計算時間について，各計算機環境，各問題サイズにおける，「倍精度・CRS」の計算時間によって無次元化したものである。値が 1.00 より大きい場合は，「倍精度・CRS」より単精度演算による ICCG 法部分の計算時間が短いことを示している。

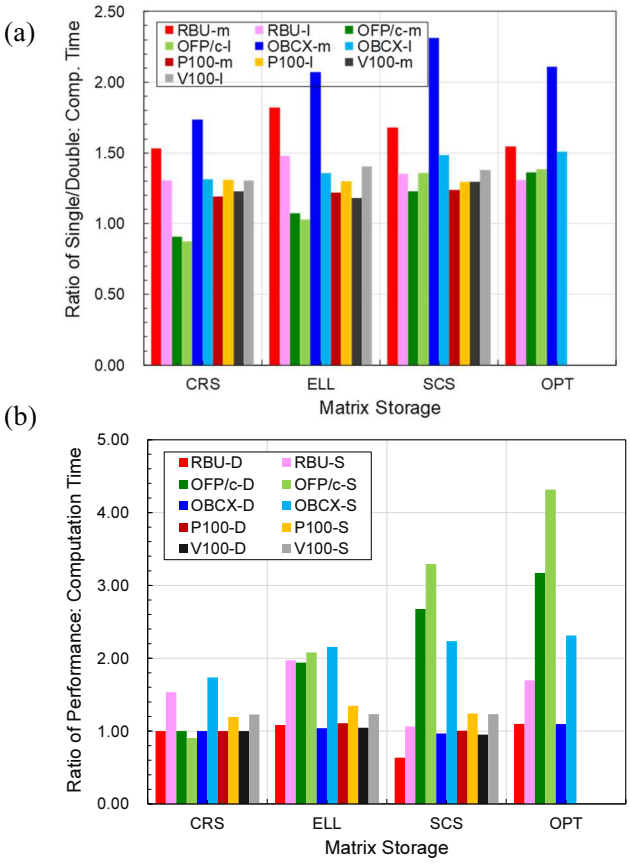


図 28　ICCG 法ソルバー（単精度）の計算性能，「倍精度」計算時間によって無次元化，(a) Tiny，Small，(b) Medium，1.00 より大きい場合は「単精度」の計算時間が短い

図 27（a）は Tiny，Small，図 27（b）は Medium，Large の場合である。OFP については，Flat モード，MCDRAM のみを使用した，ケース c（OFP/c）の場合の結果である。全般的に問題規模が小さい場合には，単精度適用による速度向上の効果は少なく，問題規模，データ移動量が大きくなるほど顕著となる傾向がある。また OFP については，CRS では単精度による速度向上が得られず，むしろ遅くなっているが，ELL，SCS，OPT では 2 倍以上の速度向上が得られており，図 26 に示した場合と同

様，CRS⇒ELL⇒SCS・OPT と演算密度上昇とともに順調に計算速度が増している。特に，「単精度・OPT」の場合には問題規模が Medium, Large では，AVX512 によるベクトル化の効果が顕著であり，反復回数が 20%増加しているにもかかわらず，「倍精度・CRS」の場合の 4 倍以上の速度向上が得られている。OFP では全般的に問題規模増加ともに速度向の効果が顕著であるが，OBCX の場合は，問題規模増加とともにむしろ速度向上は低下している。

図 28 は各ケースにおいて単純に単精度と倍精度の計算時間を比較しているが図 27 と同様の傾向が見られる。

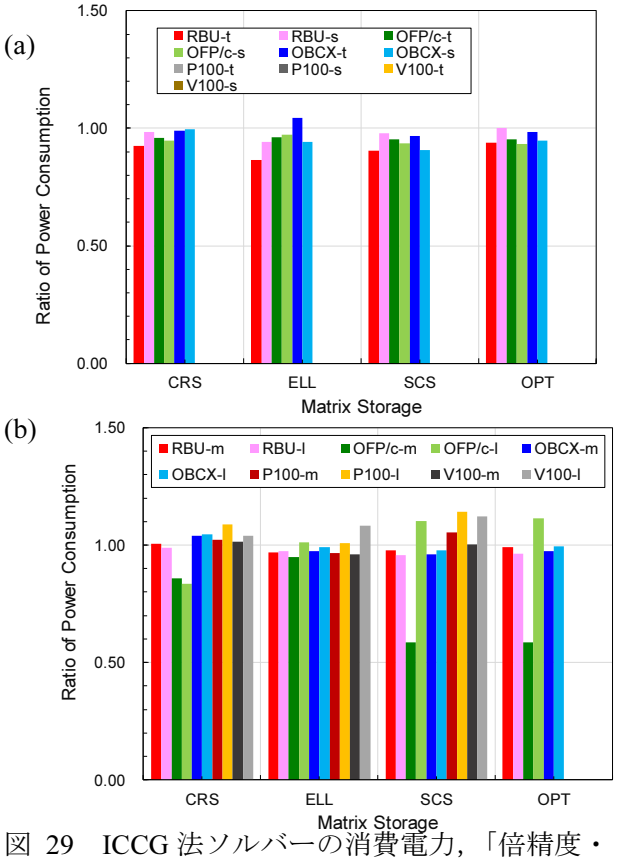


図 29 ICCG 法ソルバーの消費電力，「倍精度・CRS」の消費電力（W）によって無次元化，（a）Tiny，Small，（b）Medium，1.00 より大きい場合は「倍精度・CRS」よりも消費電力（W）が大

図 30 は (a) OFP/c (MCDRAM のみを使用した，ケース c)，（b）OBCX の各問題サイズ（Small，Medium，Large）における消費電力（W）の推移である。図 30 ではメモリ■，CPU/Package■の消費電力をそれぞれ表示しているが，OFP/c では CPU 上に直接装着された MCDRAM のみ使用しているためメモリ（DDR4）の消費電力は極めて少ない。また，問題サイズが大きくなると OBCX ではメモリ，CPU/Package ともに消費電力が増加していることがわかる。OFP/c では Medium の SCS, OPT では演算密度が CRS，ELL と比較して高いにもかかわらず消費電力（W）が低下している。

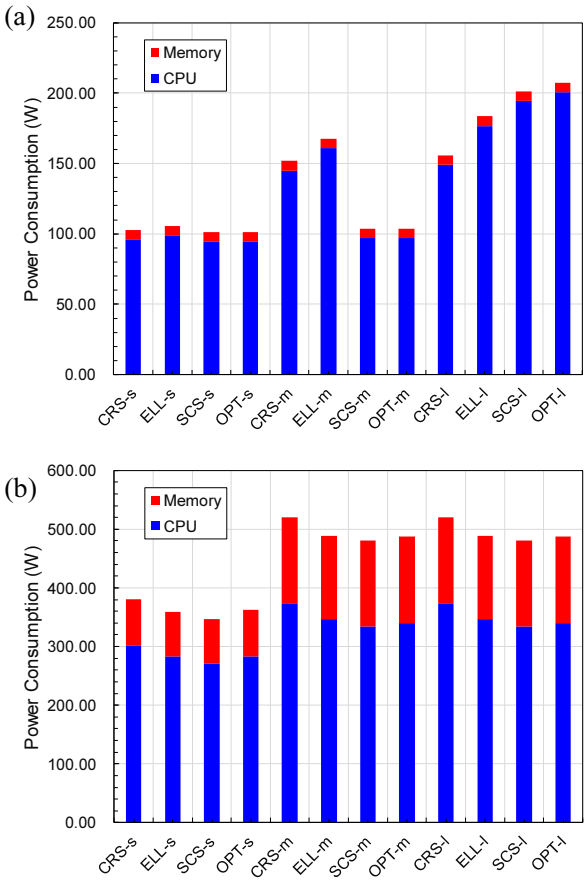


図 30 ICCG 法ソルバーにおける消費電力（W）（単精度）（a）OFP/c，（b）OBCX（s：Small，m：Medium，l：Large）

図 31，図 32 は RBU，OFP/c，OBCX，P100，V100 各計算機環境の最適ケースにおける問題サイズ「Medium，Large」の場合の ICCG 法の（a）消費電力（W），（b）消費エネルギー（J）と計算時間である。消費エネルギーと計算時間はほぼ正比例していることがわかる。計算時間としては，OFP/c, OBCX, P100, V100 はほぼ拮抗しているが，OBCX, V100 がやや速い。OBCX は消費電力（W），消費エネルギ（J）が OFP/c，P100，V100 の 3 倍以上である。OFP/c，P100，V100 は消費エネルギ（J）はほぼ等しいが，P100，V100 についてはホストの CPU の消費電力，消費エネルギー（アイドル時で

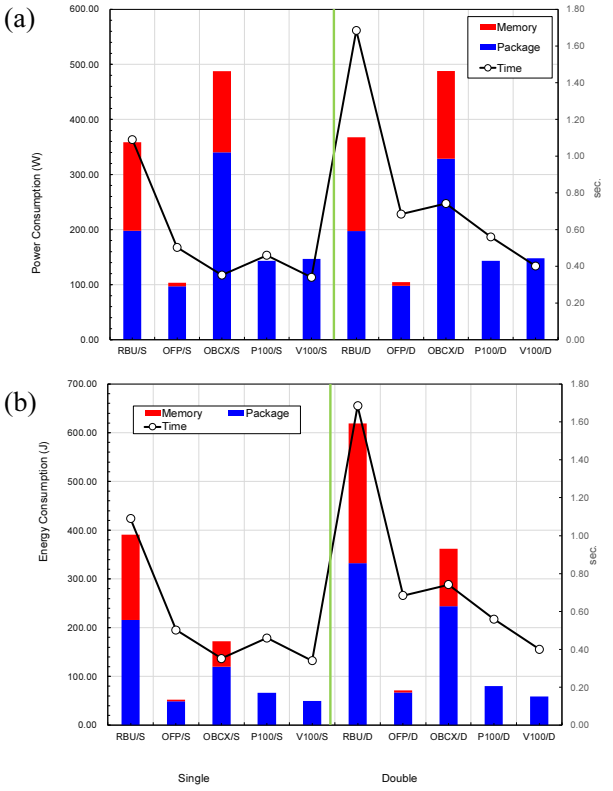


図 31 問題サイズ「Medium」の場合の ICCG 法の計算時間と（a）消費電力（W），（b）消費エネルギー（J），各計算機環境の最適ケース

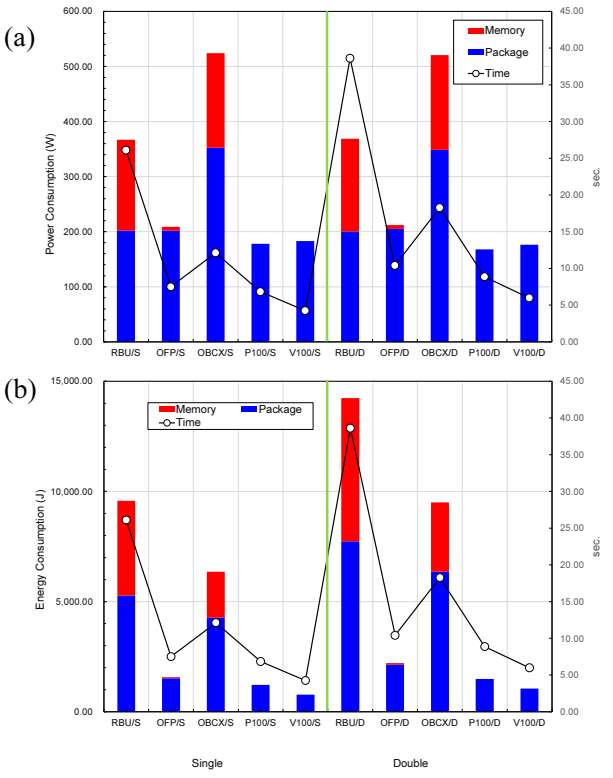


図 32 問題サイズ「Large」の場合の ICCG 法の計算時間と（a）消費電力（W），（b）消費エネルギー（J），各計算機環境の最適ケース

も 1 ノード 120W 以上）は考慮されていない。P100 ⇒V100 の計算速度向上は Medium で 35-40%, Large で 45-60%である。全般的に消費電力（W）は Medium⇒Large で増加しており，特に OFP（OFP/c）では 2 倍程度になっている。

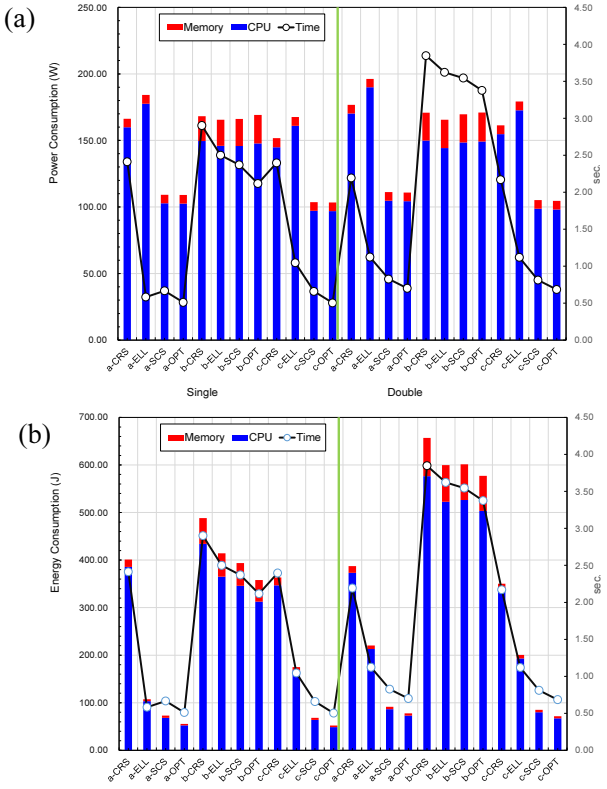


図 33　OFP：問題サイズ「Medium」の場合の ICCG 法の計算時間と（a）消費電力（W），（b）消費エネルギー（J）

図 33，図 34 は OFP において，

- a:　Cache モード
- b:　Flat モード，DDR4 のみ使用
- c:　Flat モード，MCDRAM のみ使用

を Medium（図 33），Large（図 34）の問題サイズを適用し，ICCG 法の計算時間，消費電力（W），消費エネルギー（J）を比較したものである。Cache モード（OFP/a）と Flat/MCDRAM（OFP/c）はほぼ同じ挙動である。これらのケースでは基本的に CPU 部分に収納されている MCDRAM が主として使用されるため，メモリ部分の電力消費はほとんどない。

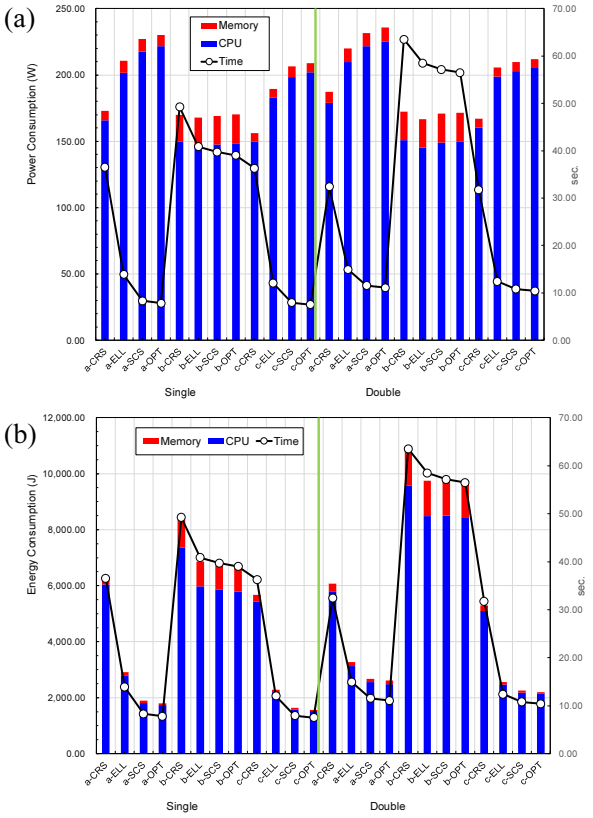


図 34　OFP：問題サイズ「Large」の場合の ICCG 法の計算時間と（a）消費電力（W），（b）消費エネルギー（J）

問題サイズが大きくなると消費電力（W）はむしろ OFP/b が他と比べて小さくなる。図 31，図 32 でも示したようにメモリ消費電力は RBU，OBCX と比較すると少ない。

表 2 に示すように MCDRAM と DDR のメモリバンド幅の比は 6:1 程度であるため，本研究で扱っている疎行列ソルバーのような memory-bound なアプリケーションでは，その影響は特に顕著であり，OFP/c-OPT と OFP/b-OPT の計算性能の比は倍精度の場合 5 倍程度である。

本研究では，著者等が先行研究において開発した有限体積法における ICCG ソルバーをターゲットとし，実装方法，問題規模と低精度演算による性能改善の関係に注目し，5 種類のハードウェア環境下での検討を実施した。実装方法（疎行列格納形式），問題規模，ハードウェア環境によって，低精度演算を使用することにより，計算時間，消費電力（W），消費エネルギー（J）への様々な効果があることがわかった。

全般的に，問題規模が大きく，SIMD 化・ベクトル化が進んでおり，演算密度が高い場合，単精度演算適用による計算速度の増加は顕著である。OFP, GPU（P100, V100）は Intel Xeon（RBU, OBCX）と比較して消費電力，消費エネルギー共に小さい。

また，SELL-C-σは，OFP を除くと必ずしも有効ではなく，むしろ〔星野他 2017〕によるスレッド並列化オーバーヘッドの除去（OPT）がより効果的であることがわかった。今後は ELL への OPT の実装を実施し，性能を評価する予定である。また，今回は GPU を使用する場合は CPU の消費電力・消費エネルギーを評価していないが，アプリケーション全体としての効率を評価するためには，CPU と合わせた消費電力・消費エネルギ評価が重要である。

⑨自動チューニング：ppOpen-AT における演算精度を考慮した実行時間最適化機能の設計〔13,22〕

自動チューニング（AT）言語の ppOpen-AT では，性能パラメータを設定し，プログラムをチューニングする際に性能パラメータを変化させて性能を測定することが AT 適用のシナリオとして定義されている。そのため，プログラムの実行時に測定結果を参照できる。

ppOpen-AT により生成される最適化候補において，どの候補を用いて実行するかを内部では性能パラメータにとる。対象箇所の実行時間の計測結果から，実際にどの候補を利用するかを決定する仕組みがプログラムとして自動生成される。

ここでは，演算対象の演算精度を変化させ，ユーザによりあらかじめ提示された演算精度以下となる条件で最高速となる演算精度の組み合わせを探索する最適化を，新たな AT 機能として実現する。

具体的には，以下である：

- **AT ソフトウェア開発者による要求指定**：
  基準となる演算に対する相対的な演算精度劣化の許容値（許容相対誤差）を事前に与える。

- **最適化対象**：
  基準となる演算精度に対して，より低い精度演算（もしくは，より高い精度演算）による混合精度演算により，(A)の要求を満たし，最高速となる組み合わせを選ぶ。

以上の新らしい AT 機能を実現する，ppOpen-AT における拡張機能（拡張した記述方式（ディレクティブ））を設計した。それを以下に示す。

```
!oat$ static changeString region start
(From 変換前 to 変換後)
＜対象の文＞
!oat$ static changeString region end
(From 変換前 to 変換後)
```

加えて，上記ディレクティブで囲まれた範囲内で，同じグループとして扱う変数を以下で囲む。

```
!oat$ changeString definition region start
＜対象の文＞
!oat$ changeString definition region end
```

2019 年度は，提案 AT 機能が有効かどうかを検証することを目的に予備実験を行なった。

使用するプログラムは，全球雲解像モデル NICAM (Non-hydrostatic ICosahedral Atmospheric Model)(非静力学 正二十面体 大気モデル)の，雲の微小な物理演算を行うカーネル physicskernel_microphysics に含まれるサブルーチン mp_snw6 を対象とした．mp_nsw6 の変数は変数を 139 個使用しており，もともとはすべて倍精度として宣言されている。

本予備実験では，倍精度（double precision）から，一部に単精度（real）を用いた混合精度演算を想定する。

以下に mp_snw6 を対象とした場合の，ppOpen-AT によるコード指定例を掲載する：

```
!oat$ static changeString region start
(From DP to SP)
!oat$ changeString definition region start
    real(DP) :: drhogqv(ijdim,kdim)
    real(DP) :: drhogqc(ijdim,kdim)
    real(DP) :: drhogqi(ijdim,kdim)
     …
!oat$ changeString definition region end
!oat$ changeString definition region start
    real(DP) :: wk       (wk_nmax)
    real(DP) :: ml_Pconv(ijdim,kdim)
    real(DP) :: ml_Pconw(ijdim,kdim)
     …
!oat$ changeString definition region end
!oat$ static changeString region end
(From DP to SP)
```

名古屋大学情報基盤センター設置の Fujitsu PRIMEHPC FX100 を用いて予備評価を行った結果，ある変数宣言グループをまとまりとした単精度演算（SP）で実行した場合，全て倍精度演算（DP）で実行した場合に対して 0.7%速度向上が得られる場合があることを確認した。この場合，DP による演算精度を基準とした場合の相対誤差は 8.50×10-13 であった。

以上の予備実験結果より，AT ソフトウェア開発者による許容相対誤差が 1.00×10^{-12} 以下である場合は混合精度演算で高速化の可能性があることが判明した。その場合，0.7%程度の速度向上を得る可能性があることを明らかにした。

なお，この単精度演算との混合精度演算を用いることによる速度向上の度合いは，DP に対する低精度演算の高速化率が大きく影響する。そのため，半精度演算を用いるとハードウェア的に DP に対してより高速化される計算機環境を用いて再評価をする予定である。

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，ISC19，SC19 等の国際会議の展示ブースでも紹介した。2018 年度の段階で低精度・混合／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かったが，SC19（2019 年 11 月）では，低精度・混合／変動精度演算を数値計算に取り入れた研究事例が多数見られた。2019 年度までに得られた知見をまとめると以下の通りである：

- 従来，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・混合演算を適用し，反復改良法（Iterative Refinement）併用により，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〔7,12,14,15,19,20,32,33,35〕。一般に，計算時間短縮に比例して消費エネルギー（Joule）は減少する〔7, 35〕。
- 悪条件問題では，低精度演算では正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，精度の要求されない反復法前処理等に適用するべきである〔12,14,15,19,20,32,33〕。
- 従来の M 疎行列向け精度保証手法〔Ogita, T. et al., Computing, 2001〕は，相対誤差上限の見積もりが厳しめであったため，より現実的な手法を開発し，悪条件問題への有効性が示された〔10,11〕。
- 演算精度の影響は，問題規模・疎行列格納手法，アーキテクチャによって多様である。
- 局所的に演算精度を変更する手法の開発に着手し，有効性が示された。

当初計画は，2018 年度：各アルゴリズム・アプリケーション最適化，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，であった。

2019 年度末時点で，ほぼ当初の予定通り目標を達成する見込みだが，2020 年度は各アルゴリズムの，演算精度，最適化，アーキテクチャの他，問題規模も考慮して消費電力・エネルギーへの体系的な影響評価を継続して実施する他，低精度・混合／変動精度演算に関する研究開発を，これまでの研究成果〔7,10,11,35〕を元に継続して実施し，自動チューニング手法確立を目指す。精度保証手法については，疎行列演算に加えて，H 行列，偏微分方程式解法向け手法の研究開発も実施する（後者については 2019 年度途中から検討を開始）。各センターの保有する NVIDIA V100，東大情報基盤センターに 2019 年度末に導入された富士通 FX700 クラスタ等を使用して，新アーキテクチャ向け検討も実施する。**特に富士通 FX700 クラスタを使用した，半精度演算（FP16）のフィージビリティースタディを様々な手法，アプリケーションについて重点的に実施する。**①～⑨は密接に連携しているが，2020 年度の各項目概要を以下に示す：

- **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，計算時間・消費エネルギー最小化のための演算精度選択手法の研究開発を継続して実施する。
- **疎行列演算（③，④，⑤）**：
  - ✓ 一般行列，悪条件行列，Adaptive CG ともに，反復改良法適用による低精度・混合／

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適用の検討，様々な手法（Pipeline 法，Dynamic Loop Scheduling，hCGA 法，SELL-C-σ），計算条件，問題規模等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継続して実施する。

- ✓ 計算時間・消費エネルギーを最小にするための演算精度選択手法の研究開発を継続して実施する。
- ✓ 2019 年度から新たに研究項目として追加した，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の研究開発を継続して実施する。

- **H 行列（⑥）**：
  - ✓ 低精度・混合／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（⑦参照）。

- **精度保証（⑦）：**
  - ✓ 疎行列演算（反復解法）については，2019 年度研究成果〔10,11〕を元に，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。階層的低ランク近似法により，密行列における行列計算の演算量を低減することができる。低精度演算を用いる場合には H 行列のランクを小さくすることができ，大幅に少ない演算量で同等の計算精度を実現できる。H 行列のランクが特異値の減衰率に依存することに着目し，様々なアプリケーションで用いられる行列について特異値の観点から、低精度演算と低ランク近似の適用範囲を明らかにし，密行列直接解法における効率的な精度保証付き計算法を開発する。
  - ✓ 三次元不均質場における楕円型偏微分方程式の解の精度保証付き数値計算を実現する。有限要素法等の離散化する際に生じる誤差も考慮し，2019 年度迄に開発した疎行列計算に対する精度保証法と組み合わせることにより，対象問題の離散化誤差を含めた包括的誤差評価を高速に達成する。

- **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，ppOpen-HPC，h3-Open-BDEC への実装を行う。

7. 研究成果（発表予定も含む）

(1) 学術論文

なし

(2) 国際会議プロシーディングス

[1] K. Fujita, M. Horikoshi, T. Ichimura, L. Meadows, K. Nakajima, M. Hori and L. Maddegedara, Development of Element-by-Element Kernel Algorithms in Unstructured Implicit Low-Order Finite-Element Earthquake Simulation for Many-Core Wide-SIMD CPUs, Proceedings of ICCS 2019, Lecture Notes in Computer Science 11536, 267-280, 2019

[2] Nakajima, K., Gerofi, B., Ishikawa, Y., Horikoshi, M., Parallel Multigrid Methods on Manycore Clusters with IHK/McKernel, IEEE Proceedings of 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems in conjunction with SC19 (The International Conference for High Performance Computing, Networking, Storage, and Analysis), 2019

[3] Yamaguchi, T., Fujita, K., Ichimura, T., Naruse, A., Lalith, M., Hori, M., GPU implementation of a

sophisticated implicit low-order finite element solver with FP21-32-64 computation using OpenACC, Proceedings of Sixth Workshop on Accelerator Programming Using Directives (WACCPD) in conjunction with SC19 (The International Conference for High Performance Computing, Networking, Storage, and Analysis), 2019

[4] Ichimura, T., Fujita, K., Yamaguchi, T., Naruse, A. et al., 416-PFLOPS fast scalable implicit solver on low-ordered unstructured finite elements accelerated by 1.10-ExaFLOPS kernel with reformulated AI-like algorithm: For equation-based earthquake modeling, Research Poster for SC19 (The International Conference for High Performance Computing, Networking, Storage, and Analysis), 2019

[5] Ma, Q., Yokota, R., Runtime System for GPU-based Hierarchical LU factorization, Research Poster for SC19 (The International Conference for High Performance Computing, Networking, Storage, and Analysis), 2019

[6] Ootomo, H., Yokota, R., TSQR on TensorCores, Research Poster for SC19 (The International Conference for High Performance Computing, Networking, Storage, and Analysis), 2019 (**Best Poster Candidate**)

[7] Sakamoto, R., Kondo, M., Fujita, K., Ichimura, T., Nakajima, K., The Effectiveness of Low-Precision Floating Arithmetic on Numerical Codes: A Case Study on Power Consumption, ACM Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, (HPC Asia 2020), 2020

[8] Nakajima, K., Parallel Multigrid Method on Multicore/Manycore Clusters, IXPUG (Intel Extreme Performance Users Group) HPC Asia 2020, 2020

(3) 国際会議発表

[9] Hosokawa, H., Okuda, H., Mixed Precision Conjugate Gradient Iterations Considering Hierarchical Memory Configurations, Proceedings of 15th U.S. National Congress on Computational Mechanics (USNCCM15) (Austin, Texas, July 28-August 1, Austin, 2019)

[10] Ogita, T., Nakajima, K., Accurate and verified solutions of large sparse linear systems arising from 3D Poisson equation, International Conference on Matrix Analysis and its Applications (MAT TRIAD 2019), Liblice, Czech Republic, September 10, 2019)

[11] Ogita, T., Nakajima, K., Verified solutions of large sparse linear systems arising from 3D Poisson equation in HPC environments, European Numerical Mathematics and Advanced Applications Conference 2019 (EnuMath 2019) (Egmond aan Zee, The Netherlands, October 2, 2019)

[12] Nakajima, K., Innovative Methods for Scientific Computing in the Exascale Era by Integrations of (Simulation+Data+ Learning) (S+D+L): Supercomputing in “Society 5.0”, Jornada Universritaria de Supercomputo 2019 (El Supercmoputo en la Tranformacion Digital) (Guadalajara, Mexico) (**Keynote Talk**)

[13] Katagiri, T., Towards Auto-tuning Technology in Exascale Era, CANDAR'19 (The Seventh International Symposium on Computing and Networking), in 7th International Workshop on Computer Systems and Architectures (CSA'19) (Nagasaki, November 26-29, 2019) (**Keynote Talk**)

[14] Nakajima, K., An Innovative Method for Integration of Simulation /Data/Learning in the Exascale/Post-Moore Era, APCOM 2019: Asian Pacific Congress on Computational Mechanics (December 18-21, 2020, Taipei, Taiwan) (**Semi-Plenary Talk**)

[15] Nakajima, K., Parallel Multigrid Methods with Adaptive Multilevel hCGA on Manycore Clusters, MS1603: Parallel Programming Models, Algorithms and Frameworks for Extreme Computing & Big Data, APCOM 2019: Asian Pacific Congress on Computational Mechanics (December 18-21, 2020, Taipei, Taiwan)

[16] Ozaki, K., Upper bound of maximum norm of inverse matrix in test set and its application, MS1603: Parallel Programming Models, Algorithms and Frameworks for Extreme Computing & Big Data, APCOM 2019: Asian Pacific Congress on Computational Mechanics (December 18-21, 2020, Taipei, Taiwan)

[17] Deshmukh, S., Yokota, R., Distributed Memory Task-Based Block Low Rank Direct Solver, HPC Asia 2020 (poster) (Fukuoka, January 2020)

[18] Apriansyah, M.R., Yokota, R., QR Decomposition of Block Low-Rank Matrices, HPC Asia 2020 (poster) (Fukuoka, January 2020)

[19] Iwashita, T, Nakajima, K., Shimokawabe, T., Nagao, H., Ogita, T., Katagiri, T., Yashiro, H., Matsuba, H., h3-Open-BDEC: Innovative Software Platform for Scientific Computing in the Exascale Era by Integrations of (Simulation + Data + Learning) , HPC Asia 2020 (poster) (Fukuoka, January 2020)

[20] Nakajima, K., Innovative Methods for Scientific Computing in the Exascale Era by Integrations of (Simulation+Data+ Learning), SIAM Conference on Parallel Processing for Scientific Computing (PP20), MS25/36: Progress and Challenges in Extreme Scale Computing and Big Data (Seattle, WA, USA, February 12-15, 2020)

(4) 国内会議発表

[21] 細川洋輝, 奥田洋司, 階層的メモリ構造を考慮した反復法ソルバーの混合精度計算，第24回日本計算工学会講演会（大宮，2019年5月29日）

[22] 片桐孝洋，櫻井刀麻，ポストムーア時代に向けた自動チューニングに向けて～スレッド数の動的最適化，第24回日本計算工学会講演会（大宮，2019年5月30日）

[23] 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24回日本計算工学会講演会（大宮，2019年5月30日）

[24] 伊田明弘，星野哲也，メニーコアクラスタにおける階層型行列法の高速化に向けた性能評価，第24回日本計算工学会講演会（大宮，2019年5月30日）

[25] 森田直樹，橋本学，奥田洋司，深層学習による反復法線形方程式ソルバの計算時間推定―学習モデルの提案と適用性評価―，第24回日本計算工学会講演会（大宮, 2019年5月31日）

[26] 深谷猛，グドール聖哉，張臨傑，岩下武史，倍精度と単精度を用いた混合精度GMRES(m)法の性能評価，第48回数値解析シンポジウム（福井，2019年6月12日）

[27] 大友広幸，横田理央，Tensorコアを用いたTSQRのGPU実装，情報処理学会研究報告（2019-HPC-170-38）（北見，2019年7月24日）

[28] 堀越将司，中島研吾，Balazs Gerofi，石川裕，Parallel Preconditioned Itertative Solvers on Oakforest-PACS，2019年並列／分散／協調処理に関する『北見』サマー・ワークショップ（SWoPP北見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）（北見，2019年7月25日）

[29] 深谷猛，グドール聖哉，張 臨傑，岩下武史，倍精度と単精度を用いた混合精度GMRES(m)法の収束性に関する実験的評価，日本応用数理学会2019年度年会（東京，2019年9月3日）

[30] 大友広幸，横田理央, Tensor コアを用いたTSQR，日本応用数理学会 2019 年度年会（東京，2019 年 9 月 5 日）

[31] 田中一成，楕円型方程式の弱解に対する正値性証明法，日本応用数理学会 2019 年度年会（東京，2019 年 9 月 5 日）

[32] 中島研吾，塙 敏博，伊田明弘，下川辺隆史，

三木洋平, 星野哲也, 有間英志, 田浦健次朗, 工藤知宏, 関谷勇司, 中村 遼, Society 5.0 実現に向けた（計算＋データ＋学習）融合, AXIES 2019 福岡（福岡, 2019 年 12 月 13 日）

[33] 中島研吾, 岩下武史, 八代尚, 下川辺隆史, 長尾大道, 荻田武史, 片桐孝洋, 松葉浩也,（計算＋データ＋学習）融合によるエクサスケール時代の革新的シミュレーション手法, 第11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（ATTA2019）（東京, 2019年12月22日）

[34] 田中一成, 中尾充宏, A priori error estimates for Poisson's equation with discontinuous coefficients, 日本応用数理学会 2019 年度連合発表会（東京, 2020 年 3 月 5 日）

[35] 中島研吾, 坂本龍一, 星野哲也, 有間英志, 塙敏博, 近藤正章, 低精度演算とアプリケーション性能, 情報処理学会研究報告（2020-HPC-174-5）（第 174 回 HPC 研究会）（オンライン, 2020 年 5 月 13 日）（in press）

（5）その他（特許, プレス発表, 著書等）

[36] FP21AXPY OpenACC source code, https://github.com/y-mag-chi/fp21axpy