Vol.2017-HPC-160 No.17
2017/7/27

HPC 分野における精度保証付き数値計算学の展開

荻田 武史[1,a)]　尾崎 克久[2]　柏木 雅英[3]　片桐 孝洋[4]

概要：本稿では，精度保証付き数値計算学の観点からの現状の HPC における問題点について議論する．まず，具体的な問題を提起し，本研究課題によってどのように改善し，解決していくかについて述べる．次に，我々が現在推進している研究プロジェクトであるポスト「京」萌芽的課題「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」について紹介する．本研究課題における最新の研究成果として，大規模な連立一次方程式に対する高精度計算方法と数値実験結果を中心に述べる．

Development of Verified Numerical Computations in HPC

Ogita Takeshi[1,a)]　Ozaki Katsuhisa[2]　Kashiwagi Masahide[3]　Katagiri Takahiro[4]

1. はじめに

本研究の目的は，HPC（ハイパフォーマンス・コンピューティング）における計算の品質を向上させ，その計算結果の信頼性を高める方法論やアルゴリズムを開発することである．近年の計算機における演算器のマルチコア・メニーコア化や汎用 GPU などの発展により，スーパーコンピュータのみならず，ワークステーションや PC においても，非常に大規模かつ複雑なコンピュータシミュレーションが可能となってきている．すなわち，計算機において計算結果を得るまでに，大量の演算・データ処理が必要となるが，このとき，計算結果の精度について，以下のような問題が発生する．

- 計算結果の精度劣化（数値計算における誤差の増大）
- 計算結果の不安定化（計算プログラムの最適化・並列化に伴う再現性の喪失）

そこで，これらの問題を解決するために，精度保証及び HPC における研究分野の研究者が協働して本研究を推進する．

1 東京女子大学
Tokyo Woman's Christian University, 2-6-1 Zempukuji, Suginami-ku, Tokyo 167–8585, Japan
2 芝浦工業大学
Shibaura Institute of Technology
3 早稲田大学
Waseda University
4 名古屋大学
Nagoya University
a) ogita@lab.twcu.ac.jp

2. HPC における問題点

様々なコンピュータシミュレーションにおいて，解くべき問題を微分方程式系として数理モデル化し，これを数値的に解くために，最終的に連立一次方程式や固有値問題などの線形問題に帰着する手法が用いられる．問題が大規模化すると，必然的に解くべき線形問題のサイズも巨大化し，現在では数万次元や数億次元の問題を数値計算によって計算機上で効率的に解くことが要求されてきている．前述のように，演算器の発展によって計算の処理速度は飛躍的に向上しており，大規模な問題であっても数値解（数値計算によって得られる近似解）を得ることは可能であるが，一方で，数値解の品質については議論されていないことが多い．しかしながら，問題の大規模化・複雑化に伴って，数値計算における誤差が増大していることも確かであり，誤差解析の観点からも，従来の方式のままでは信頼性の高い計算結果を得られなくなることが危惧される．

解くべき問題を，関数 $f : \mathbb{R}^n \to \mathbb{R}^m$ に対して $x \in \mathbb{R}^n$ を入力したときの $y = f(x)$ の計算として，これに数値計算（浮動小数点演算における単位丸め誤差を eps とする）を適用することを考える．ここで，n は問題サイズであり，$y^* \in \mathbb{R}^m$ を真の解，$\widehat{y} \in \mathbb{R}^m$ を数値解とする．このとき，多くの数値計算アルゴリズムは，後退安定 [1] となるように

表 1 条件数を変化させたときの連立一次方程式の数値解の最大相対誤差 ($n = 100$)

Table 1 Maximum relative errors for numerical solutions of linear systems with various condition numbers ($n = 100$)

cond(A)	HPL check: Eq. (1)	Max. relative error: Eq. (2)
1E+03	2.07E-16	5.08E-14
1E+06	1.00E-16	2.59E-11
1E+09	8.23E-17	1.98E-08
1E+12	9.40E-17	1.59E-05
1E+15	6.49E-17	1.02E-02

設計されており，その場合，以下のような誤差評価となる．

$$\frac{\|y^* - \widehat{y}\|}{\|y^*\|} \le p(n) \cdot \mathrm{cond}(f, x) \cdot \mathsf{eps}$$

ここで，左辺は数値解 $\widehat{y}$ の相対誤差を表す．$p(n)$ は n に関する増加関数であり，IEEE 754 の倍精度演算を用いた場合は $\mathsf{eps} \sim 10^{-16}$ である．また，$\mathrm{cond}(f, x)$ は

$$\mathrm{cond}(f, x) := \lim_{\epsilon \to 0} \sup_{\|\delta x\| \le \epsilon} \left[\frac{\|f(x + \delta x) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right]$$
$$= \frac{\|J(x)\| \cdot \|x\|}{\|f(x)\|} \quad (J(x)\text{: } f \text{ のヤコビ行列})$$

のように定義される条件数であり，$y = f(x)$ を計算する際の難しさの指標となる．よって，解くべき問題が大規模化することによって $p(n)$ が増大したり，悪条件で $\mathrm{cond}(f, x)$ が大きくなると，精度の良い数値解を得ることが困難になることが分かる．

たとえば，連立一次方程式 $Ax = b$ (A: $n \times n$ 行列, b: n 次元ベクトル) について，LU 分解を用いて数値解 $\widehat{x}$ を計算することを考える．スパコン TOP500 のベンチマークで使用されるのは，LINPACK ベンチマークの HPL [2] であるが，HPL では，精度のチェックを以下の評価式で行う．

$$\frac{\|b - A\widehat{x}\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty + \|b\|_\infty} \le C_1 n \mathsf{eps}, \ C_1 = \mathcal{O}(1) \qquad (1)$$

左辺は「相対残差」であり，後退安定なアルゴリズムを用いた場合，係数行列 A の条件数 $\mathrm{cond}_\infty(A) = \|A\|_\infty \cdot \|A^{-1}\|_\infty$ に関係なく，この評価式は（ほとんどの場合において）成立する．すなわち，これはプログラムが正しく実装されていることを確認しているだけであり，これは数値解の精度をチェックしているわけではない．実際，数値解 $\widehat{x}$ の「相対誤差」については，以下の評価式が成立する．

$$\frac{\|A^{-1}b - \widehat{x}\|_\infty}{\|A^{-1}b\|_\infty} \le C_2 n \mathsf{eps} \cdot \mathrm{cond}_\infty(A), \ C_2 = \mathcal{O}(1) \quad (2)$$

このような背景から，大規模問題に対して数値計算によって意味のある数値解を計算するためには，問題サイズや問題の条件数が大きくなった場合にも適用できるアルゴリズムを設計することであることが分かる．我々の目標は，与えられた許容誤差 $\varepsilon_{\mathrm{tol}}$ について，問題の条件数の大きさに関わらず

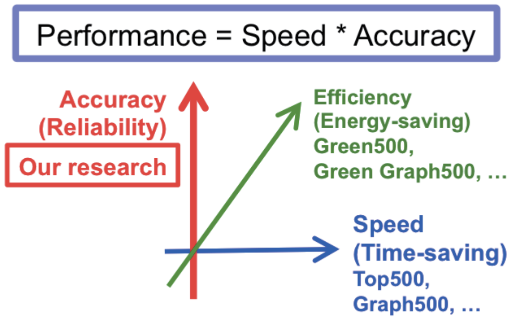

図 1 本研究課題の方針

Fig. 1 Direction of our research project

$$\frac{\|y^* - \widehat{y}\|}{\|y^*\|} \le \varepsilon_{\mathrm{tol}}$$

を満たすような $\widehat{y}$ を求めることが可能な精度保証付き数値計算アルゴリズムの開発である．このとき，既存の HPC 技術や数値計算ライブラリ（BLAS/LAPACK/ScaLAPACK 等）を資産として継承しながら，計算の高速性を可能な限り維持することが重要であると考える．

以下では，ポスト「京」萌芽的課題において，実際に我々が推進している研究について紹介する．

3. 研究課題の紹介

ここでは，我々が推進している研究プロジェクトであるポスト「京」萌芽的課題「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」について紹介する．

3.1 概要

本研究課題では，ハイパフォーマンス・コンピューティングに「精度」の軸を新たに導入し，

> 正しい計算結果を得るために最も高性能な計算環境を構築するべきである

というスーパーコンピュータの新しい指針を掲げ，そのために必要なベンチマークの設定を学問的に確立することを目標としている．**図 1** は，本研究課題の位置付けを表す．

我々は，解くべき問題に対する計算機の性能を以下のように定義する．

$$(\text{性能}) = (\text{速度}) \times (\text{精度})$$

ただし，「速度」は「計算量 / 計算時間」，「精度」は「数値解の正しい桁数」を意味する．すなわち，たとえ数値解の計算が高速であったとしても，その数値解の精度が非常に低かったとしたら，性能は低い（計算機の性能を引き出せていない）ということである．逆に，数値解の精度が高かったとしても，速度が非常に遅い場合は，やはり性能は低いということになるため，速度と精度のバランスを取ることが重要と考える．

著者らによって開発されてきた独自の数値計算法である

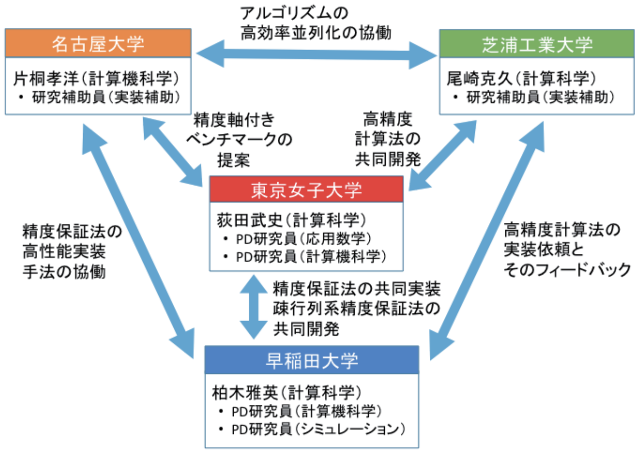

図 2 研究体制
Fig. 2 Research organization

数値線形代数における高速精度保証法 [3] やエラーフリー変換法 [4], [5] を用いると,「京」やポスト「京」において,数学的に正しい結果を数値計算によって実用的に得ることが可能となる．これによって,「スーパーコンピュータでしか取り扱うことのできない超大規模計算を必要としながら,計算の精度に起因して解くことが困難であった様々な難問」を解決することができるようになることが期待できる．

3.2 研究体制

本研究グループの研究体制を**図 2** に示す．東京女子大学では,研究代表者の荻田が『研究統括』を行い,荻田を中心として『超高性能計算環境向け精度保証付き数値計算法の開発』及び『アプリケーションソフトウェアの精度保証化・高精度化』に取り組んでいる．早稲田大学では,分担者の柏木を中心として『「京」及びポスト「京」における精度保証付き数値計算アルゴリズムの開発及び実装』に取り組んでいる．名古屋大学では,分担者の片桐を中心として『「京」及びポスト「京」におけるベンチマークの整備』に取り組んでいる．芝浦工業大学では,分担者の尾崎を中心として『超高性能計算環境向け高精度数値線形代数アルゴリズムの開発』に取り組んでいる．これらに加えて,それぞれ相互に協働可能な箇所は,積極的に共同研究を進めている．

3.3 目標

本研究課題では,以下の数値目標を設定している．

A. 密行列系では 100 万次元,スパース系では数億次元の問題（連立一次方程式,固有値問題等）について,所望の精度を持つ解を得られるようにする．

B. 問題の困難さに応じて,従来の近似計算の数倍から数十倍で精度保証を可能とする．

アウトプット成果として,本研究課題終了時には,ポスト「京」用の精度保証付きベンチマークが完成し,他の計算機向けに汎用化したものを世界に公開可能とすることを目標としている．そして,ポスト「京」の運用開始 5 年後までに,本研究成果のアウトリーチ活動を展開することによって,他に類を見ない「精度」の軸を持ったポスト「京」の超高性能性が世界中に認知されることを目指す．

アウトカム成果としては,ポスト「京」運用開始 5 年後までに,精度に関する問題でこれまでに解くことが困難であった基礎科学における各種の難問に本研究成果が適用され始め,運用開始 10 年後にはいくつかのそういった難問が解決されることが期待される．

本研究課題の成果を,計算機のトップであるポスト「京」で示すことにより,超高性能計算環境の概念はスーパーコンピュータからワークステーション,PC のレベルまでダウンサイジング化され,最終的には,あらゆる計算機上での様々な計算結果が精度保証化されるようになると予測している．これにより,数値計算を必要とする諸分野において計算結果の品質が劇的に向上することが期待される．

4. 研究成果

ここでは,本研究課題における最新の成果について述べる．

4.1 悪条件問題に対する反復改良アルゴリズムの開発

悪条件な連立一次方程式に対して,反復改良法によって高精度な数値解を得るためのアルゴリズムを開発した [6], [7]．本アルゴリズムでは,係数行列に対する前処理部分に高精度な行列積が必要となるが,最適化 BLAS を用いた高精度行列計算アルゴリズムを適用し,従来と比較して実行時間で 3 倍程度の高速化を達成した．さらに,そのような悪条件の問題に対して適応的な前処理方式を開発し,従来と比較して理論的に計算量を最悪のケースでも 1/3 程度に削減できることを示し,さらに数値実験によってその有効性を示した．

4.2 連立一次方程式に対する精度保証アルゴリズムの実装

大規模分散並列計算機における,係数行列を密とする連立一次方程式の精度保証付き数値計算アルゴリズムの開発・実装とその性能評価を行った．まず,数値線形代数で使用される関数の性能評価として,精度保証に必要な PBLAS や ScaLAPACK のルーチンの性能を調べ,実装に用いる関数の決定を行った．特に逆行列の計算においては,`pdgetrf` と `pdgetri` の組み合わせて用いるよりも,`pdgesv` を用いるほうが 20 % 程度高速であることがわかった．また,現状の数値計算の精度を把握するために,誤差が厳密に分かるテスト問題の生成法を開発した．精度保証のアルゴリズムとしては,文献 [8] の方式を用いた．まずはアルゴリズムの性質を調べるため,特定の問題から現れ

表 2 乱数行列を係数とする連立一次方程式の数値解の最大相対誤差 ($E_{\max}$)

Table 2 Maximum relative errors ($E_{\max}$) for numerical solutions of linear systems with pseudo-random matrices

Dimension	$E_{\max}$
80,000	3.80E-09
160,000	2.81E-08
320,000	1.37E-07
640,000	2.66E-07
1,280,000	5.25E-06

表 3 「京」における連立一次方程式の数値解を得るために要した時間 t_a と精度保証の計算時間 t_v の比 ($R_{\max} := t_v/t_a$)

Table 3 Ratios of computing times ($R_{\max} := t_v/t_a$) on K computer: computing numerical solutions (t_a) vs. verified numerical computations (t_v)

Dimension	Number of nodes	$R_{\max}$
80,000	16	5.57
160,000	64	5.59
320,000	256	5.58
640,000	1,024	5.55
1,280,000	4,096	4.87

る行列ではなく，疑似乱数行列を係数行列とした連立一次方程式 $Ax = b$ の数値解 $\widehat{x}$ の誤差を，京コンピュータを用いて確認した結果を**表 2** にまとめた．最大相対誤差は，真の解を $x = A^{-1}b$ に対して

$$E_{\max} := \max_{1 \le i \le n} \left| \frac{x_i - \widehat{x}_i}{x_i} \right|$$

を表す．表 2 から，128 万次元のときに数値解の相対精度が 10^{-6} 程度となっており，倍精度浮動小数点数を用いながら，数値解は単精度浮動小数点数以下の正確さしか持たないことが分かる．

表 3 では，上記の数値実験において，数値解 $\widehat{x}$ を得るために要した時間と精度保証の計算時間の比

$$R_{\max} := \frac{\text{精度保証の計算時間}}{\text{数値解の計算時間}}$$

を表している．今回用いた精度保証法では，理論的には，数値解を得るための計算（LU 分解，前進・後退代入をあわせたもの）の 6 倍の計算量を必要とする．表 3 が示すように，数値解の計算に対する精度保証の実際の計算時間の比としては，64 万次元までは 5.5 倍程度，128 万次元のときには約 4.9 倍となっている．このことから，理論的な計算量の比よりも，実際の計算時間の比は小さくなることが分かる．これは，数値解の計算に必要な LU 分解の計算速度よりも，精度保証に必要な行列積の計算速度のほうが高いためであると思われる．

また，高速精度保証法の精度面を強化するために，行列・ベクトル積と行列積の高精度計算カーネルが必要となるため，高精度内積計算 [4] の MPI 並列実装を開始した [*1]．行列・ベクトル積の関数を試作したところ，倍精度計算の約 28 倍の計算時間で倍々精度（約 4 倍精度）の計算が可能となった．行列・ベクトル積の計算量は $\mathcal{O}(n^2)$ であるため，これは十分に実用的な性能である．

4.3 高性能実装と自動チューニングの適用

本研究では，精度保証機能を有する数値計算ライブラリの開発のみならず，スーパーコンピュータの特性を考慮した高性能実装の開発も目的にしている．そのため，精度保証計算における単体性能や通信性能の高効率化も行わなくてはならない．特に近年の計算機においては，精度保証計算の問題レベルの並列性の利用，キャッシュブロッキング，階層メモリ最適化 (Non Uniform Memory Access, NUMA 最適化) を考慮しなくてはならない．また，通信時間削減のためには，通信処理の回数削減や，なるべく同期を行わない実装方式を開発する必要がある．

本研究で開発を行うベンチマークや数値計算ライブラリにおいては，精度保証計算特有のチューニングパラメタが存在することが判明している．例えば，倍精度演算の精度限界まで保証する高性能行列–行列積アルゴリズムでは，疎行列と見なす疎度，疎行列圧縮形式における疎行列–ベクトル積演算（Sparse Matrix–Vector Multiplication, SpMV）において同時に計算する複数右辺の数，および，実装方式の選択（数値計算ライブラリ BLAS における dgemm ルーチンを使う実装方式や疎行列圧縮方式を用いて SpMV する実装方式）など [9], [10] が，チューニングパラメタとなる．

先行研究 [10] では，精度保証のための無誤差変換を行う際，密行列から疎行列になるという特性を利用し，密行列から疎行列への変更を行い，かつ複数のスレッド並列化方式を SpMV 演算に実装することで，適切にチューニングパラメタの設定を行えば最大で 2.6 倍の高速化が得られることを示した．

以上のチューニングパラメタは，対象となるハードウェア構成，入力行列の数値特性に依存する．そのため，同一のパラメタでの実行では最適化が不十分となる．

以上の問題を解決するため，自動チューニング [11]–[14] の適用が必要である．我々は，精度保証計算特有の処理に自動チューニング技術を適用することで，新たな高性能実装技法を開発することも重要な研究対象であると考えている．

5. おわりに

本稿では，精度保証付き数値計算学の観点からの現状の HPC における問題点について議論した．その中で，我々が推進しているポスト「京」萌芽的課題について紹介した．

*1 実装については，本研究課題の参加者である芝浦工業大学大学院の落合涼太氏が実施した．

今後は，量子力学等，具体的なアプリケーションを設定し，本研究を様々なシミュレーションサイエンスの分野に展開していく予定である．

謝辞 本研究は，文部科学省 ポスト「京」萌芽的課題「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」の助成を受けたものである．

参考文献

[1] N. J. Higham: Accuracy and Stability of Numerical Algorithms, 2nd ed., SIAM, Philadelphia, PA, 2002.
[2] A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary: HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, Version 2.2, February 24, 2016. available from ⟨http://www.netlib.org/benchmark/hpl/⟩
[3] 荻田 武史, 大石 進一: 大規模連立一次方程式のための高速精度保証法, 情報処理学会論文誌: 数理モデル化と応用, 46:SIG10 (TOM12) (2005), 10–18.
[4] T. Ogita, S. M. Rump, S. Oishi: Accurate sum and dot product, SIAM J. Sci. Comput., 26:6 (2005), 1955–1988.
[5] S. M. Rump, T. Ogita, S. Oishi: Accurate floating-point summation part I: Faithful rounding, SIAM J. Sci. Comput., 31:1 (2008), 189–224.
[6] Y. Kobayashi, T. Ogita: Accurate and efficient algorithm for solving ill-conditioned linear systems by preconditioning methods, Nonlinear Theory and Its Applications, IEICE, 7:3 (2016), 374–385.
[7] Y. Kobayashi, T. Ogita, K. Ozaki: Acceleration of a preconditioning method for ill-conditioned dense linear systems by use of a BLAS-based method, Reliable Computing, 25 (2017), 15–23.
[8] T. Ogita, S. M. Rump, S. Oishi: Verified solution of linear systems without directed rounding, Technical Report 2005-04, Advanced Research Institute for Science and Engineering, Waseda University, Tokyo, Japan, 2005.
[9] 片桐 孝洋, 尾崎 克久, 荻田 武史, 大石 進一: 高精度行列–行列積アルゴリズムのスレッド並列化と ABCLibScript への機能実装，情報処理学会研究報告，2012-HPC-133 (26), pp. 1–8 (2012)
[10] 片桐 孝洋, 尾崎 克久, 荻田 武史, 大石 進一: 高精度行列–行列積アルゴリズムの疎行列演算化による高速化, 日本応用数理学会「行列・固有値問題の解法とその応用」研究部会第 15 回研究会, SWoPP2013, 2013.
[11] T. Katagiri, K. Kise, H. Honda, T. Yuba: ABCLibScript: a directive to support specification of an auto-tuning facility for numerical software, Parallel Computing, 32:1 (2006), 92–112.
[12] T. Katagiri, S. Ohshima, M. Matsumoto: Directive-based auto-tuning for the finite difference method on the Xeon Phi, Proceedings of IPDPSW2015, pp. 1221–1230, 2015.
[13] T. Katagiri, M. Matsumoto, S. Ohshima: Auto-tuning of Hybrid MPI/OpenMP Execution with Code Selection by ppOpen-AT, Proceedings of IPDPSW2016, pp. 1488–1495, 2016.
[14] T. Katagiri, S. Ohshima, M. Matsumoto: Auto-tuning on NUMA and Many-core Environments with an FDM Code, Proceedings of IPDPSW2017, 2017.