Vol.2020-HPC-177 No.16
2020/12/22

最近点丸めのみを用いた実対称行列に対する標準固有値問題の精度保証法

寺尾 剛史[1,a)]　尾崎 克久[2]　荻田 武史[3]　今村 俊幸[1]

概要：実対称行列に対する標準固有値問題の精度保証法は，これまでに様々な手法が提案されてきた．その実装には浮動小数点演算の丸めモードを変更した区間演算がよく用いられるが，この丸めモードの変更は計算機環境によっては難しい場合がある．本発表では，多くの計算機環境でデフォルトで用いられる最近点丸めによる浮動小数点演算のみを用いて，全近似固有値の精度を保証する新しい手法を提案する．また高精度行列積を用いて，大規模行列に対応可能かつ悪条件な行列においても誤差上限の過大評価を抑制する精度保証法を提案する．最後に，速度と精度に対する数値実験を紹介する．提案手法は LAPACK の固有値計算ソルバを用いた全近似固有対の計算時間と比較して，高速な手法の場合は 3 割，高精度な方法の場合 6 割程度の計算時間で誤差上限を計算できた．また，全近似固有値の計算時間と全近似固有対の計算を含む精度保証に要する計算時間は，行列サイズが 20000 の場合に高速な手法は 3.5 倍程度，高精度な手法は 4.5 倍程度で計算できた．また精度に関しては，高速な手法と高精度な手法で誤差の上限を比較し，高精度計算を用いて誤差上限の過大評価を抑制できていることを確認した．

1. はじめに

本論では，実対称行列 $A \in \mathbb{R}^{n\times n}$ に対する全固有値問題

$$Ax^{(i)} = \lambda_i x^{(i)}, \quad 1 \le i \le n$$

の最近点丸めによる浮動小数点演算のみを用いた精度保証付き数値計算法について述べる．全固有値に関する精度保証付き数値計算法は複数提案されている（[1], [2], [3], [4], [5], [6], [7] など）．本研究では，[7] で記載されている精度保証法を最近点丸めの浮動小数点演算のみを用いて実装する手法の提案とその評価を行う．

A の近似固有値を $d_i \in \mathbb{R}$，その近似固有ベクトルを $\hat{x}^{(i)} \in \mathbb{R}^n$ とし，全ての i について近似固有対が得られているとする．本研究では，全固有値に対して

$$d_i - \delta \le \lambda_i \le d_i + \delta, \quad 1 \le i \le n$$

を満たす $0 \le \delta \in \mathbb{R}$ を最近点丸めの浮動小数点演算のみを用いて計算する手法を提案する．先行研究の多くは有向丸め（上向き丸めや下向き丸め）を用いた実装がなされている．これは，有向丸めを用いた行列積の区間演算 [8] が有効な手法なためである．しかし，計算環境の多様化に伴い，有向丸めを用いた精度保証法の実装が難しい場合が多々発生する．実際に，丸めモードの変更に伴うコストが無視できないケースも報告されている [9]．効率的に丸めモードを変更できる場合でも，最適化された BLAS や PBLAS, cuBLAS 等のルーチンを活用することは難しい．実際に，丸めの変更がマルチスレッド計算に対応できるか？また PBLAS の場合はアシスタントコアの活用に対応できるか？などの確認事項が多い．このように，有向丸めを用いた精度保証付き数値計算法は，ハードウェアやソフトウェアの影響を受ける．この問題に対処するため，最近点丸めのみを用いた連立 1 次方程式の数値解に対する精度保証付き数値計算法が提案されている [10], [11]．また，固定された精度を用いた区間演算法として [9], [12] などが提案されている．これらの手法は，実数で与えられる数値解の誤差上限を，さらに誤差解析を行うことで最近点丸めのみでの実装を可能とする．これらの手法を固有値問題に応用することで，最近点丸めのみで固有値の保証が可能である．しかし，この場合の特徴として最終的な誤差解析の結果が複雑になる傾向がある．

本研究では，精度保証式の複雑化を抑える最近点丸めのみを用いた精度保証付き数値計算法の開発を目的とする．計算の多くを BLAS のルーチンを用いて実装可能な手法を提案し，共通メモリ型計算機等を用いた小中規模な行列に対する精度保証法の評価を行う．2 章では本稿で用いる表

[1] 理化学研究所計算科学研究センター
[2] 芝浦工業大学システム理工学部数理科学科
[3] 東京女子大学現代教養学部数理科学科
[a)] takeshi.terao@riken.jp

記と誤差評価を紹介する．3 章では，最近点丸めのみを用いた全固有値の精度保証法を 2 つ提案する．1 つめは通常の行列積を用いた高速な方法であり，2 つめは高精度行列積を用いて誤差の過大評価を抑制する方法である．4 章では，提案手法の精度，速度に対する数値実験結果を紹介する．最後に，本稿のまとめと今後の展開について述べる．

2. 準備

本章では，本稿で用いられる記号と基本的な丸め誤差解析について紹介する．

2.1 表記

ここでは，本稿で用いる記号について紹介する．$\mathbb{F}$ は IEEE 754 規格 [13] により定められたある固定精度の浮動小数点数の集合とする．u を単位相対丸め [14] とする（binary 64 の場合 $u = 2^{-53}$）．$fl(\cdot), fl_{\triangle}(\cdot), fl_{\triangledown}(\cdot)$ は，それぞれ括弧内の各演算を最近点丸め（roundTiesToEven），上向き丸め（roundTowardPositive），下向き丸め（roundTowardNegative）の丸めのモードによる浮動小数点演算で計算した結果とする．議論の簡略化のため，アンダーフローとオーバーフローは精度保証の計算中に発生しないと仮定する．計算途中でオーバーフローが発生した場合の計算結果は ±Inf や NaN になるため，オーバーフローが発生しなかったことは事後に検証可能である．また，アンダーフローが発生する場合は，比較的低コストで確認と修正が可能であるため，本稿では式を簡略化することを優先する．ベクトル $v, w \in \mathbb{R}^n$ に対する絶対値 $|v|$ はすべての要素で $|v|_i = |v_i|$ となるベクトルとする．不等号 $v < w$ はすべての要素で $v_i < w_i$ が成り立つものとする．ベクトルに対する絶対値記号と不等号については，行列についても同様に拡張して用いる．$I \in \mathbb{F}^{n\times n}$ は単位行列，$e \in \mathbb{F}^n$ はすべての要素が 1 のベクトルとする．

2.2 誤差解析

本節では，丸め誤差解析に用いる基本的な式を紹介する．まず，$a, b \in \mathbb{F}$ に対して

$$|a \circ b - fl(a \circ b)| \le u|a \circ b|, \quad \circ \in \{+, -, *, /\} \quad (1)$$

$$|a \circ b - fl(a \circ b)| \le u \cdot fl(|a \circ b|), \quad \circ \in \{+, -, *, /\} \quad (2)$$

が成り立つ．文献 [14] にて，$|a \circ b - fl(a \circ b)|$ の上限は $u/(1+u)$ で抑えられることが示されているが，本議論では u を用いて，丸め誤差解析の結果を簡潔にしている． 本稿では，(1)，(2) を変形した

$$a \circ b - u|a \circ b| \le fl(a \circ b) \le a \circ b + u|a \circ b| \quad (3)$$

と

$$fl(a \circ b) - ufl(|a \circ b|) \le a \circ b \le fl(a \circ b) + ufl(|a \circ b|) \quad (4)$$

を以後多く利用する． $a \ge 0$ に対する平方根に対しても

$$|\sqrt{a} - fl(\sqrt{a})| \le u\sqrt{a} \quad (5)$$

が成り立つ．

また $v, w \in \mathbb{F}^n$ に対して，[15] より，

$$\left|\sum_{i=1}^{n} v_i - fl\left(\sum_{i=1}^{n} v_i\right)\right| \le (n-1)u\sum_{i=1}^{n}|v_i| \quad (6)$$

$$|v^T w - fl(v^T w)| \le nu|v^T||w| \quad (7)$$

が成り立つ．ここで，(6), (7) は，Strassen のアルゴリズム [16] 等の特殊な計算順序を除き，内積計算における積と和を任意の順序で計算しても成立することが知られている．また，FMA（Fused Multiply Add）を用いた内積計算に対しても (7) が成り立つ．次に，有向丸めを用いた区間包括を紹介する．文献 [8] より，

$$fl_{\triangledown}(v^T w) \le v^T w \le fl_{\triangle}(v^T w) \quad (8)$$

が成り立つ．この式を用いると通常の計算の 2 倍の計算コストで，(7) と比較してタイトな区間包含が可能であることが知られている．

次に，誤差解析に用いる式変形をまとめる．文献 [10] から $1 + nu$ について，$nu < 1$ ならば，

$$1 + nu \le \frac{1}{1-nu} \le fl\left(\frac{1}{1-(n+1)u}\right) \quad (9)$$

が成り立つ．特に，$|a| \ge u_N$ ならば，

$$\frac{|a|}{1-nu} \le fl\left(\frac{|a|}{1-(n+1)u}\right) \quad (10)$$

が成り立つ．ここで，u_N は正の正規化数の最小値である．

3. 全固有値の精度保証法

冒頭に述べたように，固有値に対する精度保証付き数値計算法は数多く提案されている．本稿では，近似固有値と摂動に関する定理を用いた精度保証付き数値計算法に焦点を当てる．実対称行列 $A \in \mathbb{R}^{n\times n}$ の固有値を λ_i, 近似固有対を $(d_i, \hat{x}^{(i)})$ とする．また，D を (i,i) 成分に d_i を持つ対角行列，X を i 列目に $\hat{x}^{(i)}$ を持つ行列とする．このとき，

$$\tilde{S} := AX - XD, \quad \tilde{T} := X^T X - I \quad (11)$$

とおくと，$\|\tilde{T}\|_\infty < 1$ のとき

$$|\lambda_i - d_i| \le \sqrt{\frac{\|\tilde{S}\|_1 \|\tilde{S}\|_\infty}{1 - \|\tilde{T}\|_\infty}}, \quad 1 \le i \le n \quad (12)$$

が成り立つ [7]．本稿では，(12) に基づく最近点丸めによる浮動小数点演算のみを用いた固有値の精度保証付き数値計算法について述べる．連立 1 次方程式に対して，誤差評価式を最近点丸めのみで計算する方法は [10] で提案されて

おり，[11] では，unit in the first place (ufp) 関数 ([9], p. 364 など) を用いた誤差評価式の改良がおこなわれている．これらの先行研究を応用することで，δ の上限を評価することは可能である．しかし，これらの方法を用いた場合，誤差評価式が複雑化する傾向があり，実装にも計算順序を慎重に指定する必要がある．本提案手法では，計算順序に関する制約を緩和し，コードの多くを BLAS 等の線形計算ライブラリで記述できる精度保証法を提案する．

3.1 有向丸めを用いた方法

ここでは，有向丸めを用いた精度保証法を紹介する．まず (11) で定めた $\tilde{S}$ に対して (8) より

$$\underline{S} := fl_{\triangledown}(AX + (-XD)), \quad \overline{S} := fl_{\triangle}(AX + (-XD))$$

とおくと，$\underline{S} \le \tilde{S} \le \overline{S}$ が成り立つ．同様に (11) で定めた $\tilde{T}$ に対して

$$\underline{T} := fl_{\triangledown}(X^TX - I), \quad \overline{T} := fl_{\triangle}(X^TX - I)$$

とおくと，$\underline{T} \le \tilde{T} \le \overline{T}$ が成り立つ．よって

$$|\tilde{S}| \le \max(|\underline{S}|, |\overline{S}|) =: S', \quad |\tilde{T}| \le \max(|\underline{T}|, |\overline{T}|) =: T'$$

が成り立つ．上記にある行列と行列の max については成分ごとの最大値を選択した行列を返すものとする．従って，$\|\tilde{S}\|_\infty$ に対して

$$\|\tilde{S}\|_\infty = \||\tilde{S}|e\|_\infty \le \|S'e\|_\infty \le \|fl_{\triangle}(S'e)\|_\infty =: \alpha_2$$

と上限が得られる．同様に

$$\alpha_1 \ge \|\tilde{S}\|_1, \quad \beta \ge \|\tilde{T}\|_\infty \tag{13}$$

を満たす $\alpha_1,\ \beta \in \mathbb{F}$ が計算でき，$\beta < 1$ ならば

$$\sqrt{\frac{\|\tilde{S}\|_1\|\tilde{S}\|_\infty}{1 - \|\tilde{T}\|_\infty}} \le fl_{\triangle}\left(\sqrt{\frac{\alpha_1\alpha_2}{fl_{\triangledown}(1-\beta)}}\right)$$

が成り立つ．

3.2 最近点丸めのみを用いた精度保証法

本節では，最近点丸めのみを用いて (12) の上限を求める精度保証法を提案する．

$$\alpha_1 \ge \|\tilde{S}\|_1, \quad \alpha_2 \ge \|\tilde{S}\|_\infty, \quad 1 > \beta \ge \|\tilde{T}\|_\infty \tag{14}$$

を満たす $0 \le \alpha_1, \alpha_2, \beta \in \mathbb{F}$ があるとする．$0 \le a \in \mathbb{F}$ に対して，(5) により

$$fl(\sqrt{a}) \ge (1-u)\sqrt{a}$$

であり，$\beta < 1$ のとき (3)，(9) より，

$$\begin{aligned}
fl\left(\sqrt{\frac{\alpha_1\alpha_2}{1-\beta}}\right) &\ge (1-u)\sqrt{fl\left(\frac{\alpha_1\alpha_2}{1-\beta}\right)} \\
&\ge (1-u)\sqrt{(1-u)\frac{fl\,(\alpha_1\alpha_2)}{fl\,(1-\beta)}} \\
&\ge (1-u)\sqrt{(1-u)\frac{(1-u)\alpha_1\alpha_2}{(1+u)(1-\beta)}} \\
&\ge (1-u)\sqrt{(1-u)\frac{(1-u)\alpha_1\alpha_2}{\frac{1}{1-u}(1-\beta)}} \\
&= (1-u)\sqrt{(1-u)^3\frac{\alpha_1\alpha_2}{1-\beta}} \\
&\ge (1-3u)\sqrt{\frac{\alpha_1\alpha_2}{1-\beta}}
\end{aligned}$$

が成り立つ．また，(10) と (12) から

$$\begin{aligned}
|\lambda_i - d_i| &\le \sqrt{\frac{\alpha_1\alpha_2}{1-\beta}} \le \frac{1}{1-3u}\cdot fl\left(\sqrt{\frac{\alpha_1\alpha_2}{1-\beta}}\right) \\
&\le fl\left(\sqrt{\frac{\alpha_1\alpha_2}{1-\beta}}/(1-4u)\right)
\end{aligned}$$

を得る．以後，上式における $\alpha_1,\ \alpha_2,\ \beta$ の上限を最近点への丸めの演算のみを用いて求める方法を紹介する．

まず，

$$S := fl(AX - XD), \quad T := fl(X^TX - I) \tag{15}$$

とおく．行列のノルムの計算について，

$$\|S\|_1 = \|e^T|S|\|_\infty, \quad \|S\|_\infty = \||S|e\|_\infty, \quad \|S\|_\infty = \||S|e\|_\infty$$

であるから，全要素が非負の行列・ベクトル積の上限を計算する方法を考える．以下に，全要素が非負のベクトルに対する内積の上限の計算方法に関する定理を示す．

定理 1 全要素が非負のベクトル $v, w \in \mathbb{F}^n$ と正規化数であるスカラー $p \in \mathbb{F}$ に対して，

$$(n+2)u < 1, \quad \rho_{(p,k)} = \frac{p}{1-ku}$$

であるとき，

$$fl((\rho_{(p,n+2)}v^T)w),\ fl(\rho_{(p,n+2)}(v^Tw)),\ fl(v^T(\rho_{(p,n+2)}w))$$

はいずれも pv^Tw の上限となる．

証明.

まず，$fl((\rho_{(p,n+2)}v^T)w)$ について考える．(9)，(4) より，

$$\begin{aligned}
\rho_{(p,n)}v = \frac{p}{1-nu}v &= \frac{1+u}{1+u}\cdot\frac{p}{1-nu}v \\
&\le \frac{1}{1+u}\cdot\frac{1}{1-u}\cdot\frac{p}{1-nu}v \\
&\le \frac{1}{1+u}\cdot\frac{p}{1-(n+1)u}v \\
&\le \frac{1}{1+u}\cdot fl\left(\frac{p}{1-(n+2)u}\right)\cdot v \\
&\le fl\left(\frac{p}{1-(n+2)u}v\right) = fl\left(\rho_{(p,n+2)}v\right) =: \bar{v}
\end{aligned}$$

を得る．$\rho_{(p,n)}$ の定義より，

$$pv = (1-nu)\rho_{(p,n)}v$$

であり，また $\rho_{(p,n)}v$ と $\bar{v}$ の関係，(7) を用いれば

$$pv^Tw = (1-nu)\rho_{(p,n)}v^Tw \le (1-nu)\bar{v}^Tw \le fl(\bar{v}^Tw)$$

が成り立つ．よって

$$pv^Tw \le fl((\rho_{(p,n+2)}v^T)w)$$

を示した．$v^Tw = w^Tv$ であるから，$fl(v^T(\rho_{(p,n+2)}w))$ も同様に pv^Tw の上限となる．

次に $fl(\rho_{(p,n+2)}(v^Tw))$ について考える．式 (7) より

$$v^Tw \le \frac{1}{1-nu}\cdot fl(v^Tw)$$

であるから，$\bar{v}$ に関する不等式と同様の変形により

$$\begin{aligned} pv^Tw &\le \frac{p}{1-nu}\cdot fl(v^Tw) \\ &\le \frac{1}{1+u}\cdot fl\left(\frac{p}{1+(n+2)u}\right)\cdot fl(v^Tw) \\ &\le fl(\rho_{(p,n+2)}(v^Tw)) \end{aligned}$$

が成り立つ．

□

成分が浮動小数点数で表現されたベクトルと定数があり，このベクトルの内積の定数倍は定理 1 を用いれば最近点の丸めモードによる浮動小数点演算により上限が求まる（以後で用いる $fl(\rho_{(p,n)})$ はすべて正規化数である）．また，定理 1 は行列・ベクトル積の定数倍にも同様に拡張される．

例として，$|\tilde{S}|e$ の上限を計算する場合を考える．(7) から

$$\begin{aligned} |\tilde{S}| &\le |S| + (n+1)u(|A||X| + |X||D|) \\ |\tilde{T}| &\le |T| + (n+1)u(|X^T||X| + I) \end{aligned}$$

が成り立つ．上記については D と I が対角行列であるため，$AX - XD$ と $X^TX - I$ では長さが $n+1$ のベクトルの内積計算を行うことから $(n+1)u$ を用いている．これらの式の右辺のノルムの上限を求める．$|S|e$ の計算では要素毎の積で丸め誤差が発生しないため，定理 1 より

$$z^{(1)} := fl(\rho_{(1,n+1)}|S|e) \ge |S|e$$

が成り立つ．また $d = De$ とし，同様に定理 1 を用いて

$$\begin{aligned} y^{(2)} &:= fl(\rho_{(1,n+1)}|X|e) \ge |X|e \\ z^{(2)} &:= fl(\rho_{((n+1)u,n+2)}|A|y^{(2)}) \ge (n+1)u|A||X|e \\ z^{(3)} &:= fl(\rho_{((n+1)u,n+2)}|A||d|) \ge (n+1)u|X||D|e \end{aligned}$$

を得る．3 つの非負の浮動小数点数 $a, b, c \in \mathbb{F}$ に対して，(6) より

$$(1-2u)(a+b+c) \le fl(a+b+c)$$

であるため，

$$a+b+c \le \frac{1}{1-2u}fl(a+b+c) \tag{16}$$

となる．上記と (9) より

$$\begin{aligned} \|\tilde{S}\|_\infty &\le \max(z^{(1)} + z^{(2)} + z^{(3)}) \\ &\le \frac{1}{1-2u}\cdot\max(fl(z^{(1)} + z^{(2)} + z^{(3)})) \\ &\le fl\left(\frac{\max\left(z^{(1)} + z^{(2)} + z^{(3)}\right)}{1-3u}\right) =: \alpha_2 \end{aligned}$$

が得られる．このとき，$z^{(1)}, z^{(2)}, z^{(3)}$ の内部およびその和の計算順序は任意でよい．同様の方法で $\|\tilde{S}\|_1, \|\tilde{T}\|_\infty$ の上限 α_1, β を計算可能である．

定理 1 は計算順序の制約が弱いため，行列・ベクトル積の定数倍に対して BLAS level 2 の gemv 系列のルーチンをそのまま適用できる．$\mathrm{gemv}(\alpha, A, x, \beta, y)$ を

$$y \leftarrow \alpha Ax + \beta y \tag{17}$$

とする．ただし，ここでの引数は (14) で用いる記号との関連はないものとする．このとき，$\alpha \leftarrow \rho_{(1,n+1)}$, $A \leftarrow |S|$, $x \leftarrow e$, $\beta \leftarrow 0$, $y \leftarrow z^{(1)}$ として，$\mathrm{gemv}(\alpha, A, x, \beta, y)$ を最近点丸めで実行したとき，$y \ge |S|e$ が成り立つ．このように，提案手法は $\rho_{(p,n)}$ の p, n を適切に設定するのみでよい．また定理 1 の結果があれば，以後に誤差上限を計算するうえで複雑な誤差解析を必要とせず，コードの多くを BLAS のルーチンで記載することが可能である．

3.3 高精度行列積を用いた精度保証法

本節では，高精度行列積を応用した精度保証法を提案する．式 (12) に基づく精度保証法では，β と比較して α_1, α_2 の依存度が大きい．例えば，$\sqrt{\alpha_1\alpha_2} = 1$ で，β が 0.01 と 0.001 の場合を考える．このとき，

$$\sqrt{\frac{1}{1-0.01}} = 1.0050\ldots, \quad \sqrt{\frac{1}{1-0.001}} = 1.0005\ldots$$

であり，β が 10 倍悪化したとしても，その影響は小さい．しかし，$\sqrt{\alpha_1\alpha_2}$ が 10 倍悪化した場合は誤差上限も 10 倍悪化する．この性質を踏まえて，本稿では α_1, α_2 の過大評価を避けるために，$|AX - XD|$ に対してのみ高精度行列積を適用する．

3.2 節で紹介した手法では，$|AX - XD|$ の計算に対して 1 回の行列積とその誤差評価式を用いた精度保証付き数値計算法を提案した（ここでは D が対角行列であるため，計算コスト面を考慮して XD を 1 回の行列積とみなさない表現を用いた）．この手法は，計算コストが比較的小さく高速である一方，大規模行列に対して誤差の過大評価が発生しやすくなる．このような場合に，高精度行列積を用い

た精度保証法が提案されている [17]．固有値を倍精度で計算した場合，行列積を 4 倍精度や疑似多倍長精度で計算する必要があり，これらを用いて計算性能を出すことは容易ではない．

本稿では，行列を分割（以後，尾崎スプリット）して行列積を高精度に計算する手法 [18] に着目する．この手法の特徴として，既存の最適化された BLAS ルーチンを用いて実装が可能であり，高性能計算の分野で注目されている．実際に，OzBLAS [19] や Batched BLAS を用いた実装法 [20]，疎行列に対する計算法 [21] など，尾崎スプリットを用いた高精度行列積を計算するルーチンが活発に開発されている．

本稿では, 簡単のために尾崎スプリットにより行列を二分割する場合の高精度計算法を紹介する．行列 $A, X \in \mathbb{F}^{n\times n}$ に対して，

$$A^{(1)}X^{(1)} = fl(A^{(1)}X^{(1)}), \tag{18}$$
$$A = A^{(1)} + A^{(2)}, \quad X = X^{(1)} + X^{(2)} \tag{19}$$
$$A^{(1)},\ A^{(2)},\ X^{(1)},\ X^{(2)} \in \mathbb{F}^{n\times n}$$

を満たすように分割する．具体的な行列の分割方法は [18] に記載されている．このとき，

$$|a_{ij}^{(1)}| \geq |a_{ij}^{(2)}|, \quad (a_{ij}^{(1)} \neq 0)$$

という大小関係であり，実際には．$|a_{ij}^{(1)}|$ が $|a_{ij}^{(2)}|$ よりもはるかに大きいことが多い．(19) より，

$$\begin{aligned} AX &= (A^{(1)} + A^{(2)})(X^{(1)} + X^{(2)}) \\ &= A^{(1)}X^{(1)} + A^{(1)}X^{(2)} + A^{(2)}X \end{aligned}$$

となり，AX は 3 つの行列積の和で表現できる．

次に，$AX - XD$ に関する誤差解析結果を紹介する．まず，

$$S_1 := fl(A^{(1)}X^{(1)} - XD), \quad S_2 := fl(A^{(1)}X^{(2)} + A^{(2)}X)$$

とおき, $A^{(1)}X^{(1)} - XD$ を下限と上限で包含する．(4) より，

$$S_1 - u|S_1| \leq fl(A^{(1)}X^{(1)}) - fl(XD) \leq S_1 + u|S_1|$$

であるから，(18) より $A^{(1)}X^{(1)} - XD$ の上限は

$$\begin{aligned} A^{(1)}X^{(1)} - XD &\leq A^{(1)}X^{(1)} - fl(XD) + u|X||D| \\ &= fl(A^{(1)}X^{(1)}) - fl(XD) + u|X||D| \\ &\leq S_1 + u|S_1| + u|X||D| \end{aligned} \tag{20}$$

となる．同様に，$A^{(1)}X^{(1)} - XD$ の下限として

$$A^{(1)}X^{(1)} - XD \geq S_1 - u|S_1| - u|X||D| \tag{21}$$

を得る．次に，$A^{(1)}X^{(2)} + A^{(2)}X$ の包含を考える．(7) より

$$A^{(1)}X^{(2)} \geq fl(A^{(1)}X^{(2)}) - nu|A^{(1)}||X^{(2)}| \tag{22}$$
$$A^{(1)}X^{(2)} \leq fl(A^{(1)}X^{(2)}) + nu|A^{(1)}||X^{(2)}| \tag{23}$$

と

$$A^{(2)}X \geq fl(A^{(2)}X) - nu|A^{(2)}||X| \tag{24}$$
$$A^{(2)}X \leq fl(A^{(2)}X) + nu|A^{(2)}||X| \tag{25}$$

を得る．(4) より，

$$S_2 - u|S_2| \leq fl(A^{(1)}X^{(2)}) + fl(A^{(2)}X) \leq S_2 + u|S_2|$$

であるため，(22)，(23)，(24)，(25) から

$$\begin{aligned} &S_2 - u|S_2| - nu(|A^{(1)}||X^{(2)}| + |A^{(2)}||X|) && (26)\\ \leq &A^{(1)}X^{(2)} + A^{(2)}X \\ \leq &S_2 + u|S_2| + nu(|A^{(1)}||X^{(2)}| + |A^{(2)}||X|) && (27) \end{aligned}$$

を得る．(20)，(21)，(26)，(27) より，

$$\begin{aligned} |\tilde{S}| &= |A^{(1)}X^{(1)} - XD + A^{(1)}X^{(2)} + A^{(2)}X| \\ &\leq (1+u)|S_1| + (1+u)|S_2| + u|X||D| \\ &\quad + nu(|A^{(1)}||X^{(2)}| + |A^{(2)}||X|) \end{aligned}$$

と $|\tilde{S}|$ の上限を得る．上式において $1+u \notin \mathbb{F}$ であるため，$1+2u \in \mathbb{F}$ で置き換え，右から e をかければ

$$\begin{aligned} |\tilde{S}|e &\leq (1+2u)|S_1|e + (1+2u)|S_2|e + u|X||D|e \\ &\quad + nu(|A^{(1)}||X^{(2)}|e + |A^{(2)}||X|e) \end{aligned}$$

を得る．ここで，

$$u|X||D|e, \quad nu|A^{(1)}||X^{(2)}|e, \quad nuA^{(2)}||X|e$$

については，まず

$$|D|e, \quad |X^{(2)}|e, \quad |X|e$$

に対して，定理 1 を適用して，

$$d_1 \geq |D|e, \quad d_2 \geq |X^{(2)}|e, \quad d_3 \geq |X|e$$

となる $d_1,\ d_2,\ d_3 \in \mathbb{F}^n$ を得る．次に，

$$(1+2u)|S_1|e,\ (1+2u)|S_2|e,\ u|X|d_1,\ nu|A^{(1)}|d_2,\ nu|A^{(2)}|d_3$$

に対して，再び定理 1 を適用する．これらの結果を順に $f_1, f_2, f_3, f_4, f_5 \in \mathbb{F}^n$ とすると，(16) と同様の式変形と (9) より

$$\|\tilde{S}\|_\infty \leq fl\left(\frac{\max(f_1 + f_2 + f_3 + f_4 + f_5)}{1-5u}\right)$$

となり，$\|\tilde{S}\|_\infty$ の上限を最近点への丸めの浮動小数点演算のみを用いて得た．同様に $\|\tilde{S}\|_1$ と $\|\tilde{T}\|_\infty$ の上限も計算できる．

4. 数値実験

まず，3 章で示した全近似固有値の精度保証法の計算量の比較を行う．表 1 では，各節で示した手法の丸めモードの必要性と計算量を示す．ここで示した計算量は近似固有対に必要な計算量を含まない，精度保証に要する計算量である．

表 1 各手法の計算量と特徴

	3.1 節	3.2 節	3.3 節
丸めの変更	必要	不要	不要
計算量	$6n^3 + \mathcal{O}(n^2)$	$3n^3 + \mathcal{O}(n^2)$	$7n^3 + \mathcal{O}(n^2)$

3 章で紹介した手法は，有向丸めを用いた手法と比較して高速な手法と高精度な手法である．以降に倍精度計算を用いた数値実験結果を紹介する．

4.1 精度に関する実験結果

ここでは，精度に関する数値実験結果を紹介する．計算機環境は，XPS 13 9300, CPU：Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz, Memory：16 GB，BLAS：OpenBLAS v3.10, OS： Windows 10, コンパイラ：gcc version 9.3.0 (Ubuntu 9.3.0–10ubuntu2) を用いた．また，行列は最大固有値が約 1，最小固有値が約 10^{-5} の実対称行列を用いた．また，各固有値は幾何分布に従うように作成した．計算された誤差上限を以下のように表記する．

δ_1： 最近点丸めのみを用いた高速な手法（3.2 節）
δ_2： 有向丸めを用いた手法（3.1 節）
δ_3： 最近点丸めのみを用いた高精度な手法（3.3 節）

表 2 は，各誤差上限の値を示す．

表 2 各行列サイズに対する近似固有値の誤差上限

n	δ_1	δ_2	δ_3
2000	6.21e-11	5.09e-13	3.96e-14
4000	2.41e-10	9.27e-13	7.73e-14
6000	5.39e-10	1.34e-12	1.54e-13
8000	9.41e-10	1.72e-12	1.54e-13
10000	1.47e-09	2.08e-12	2.41e-13

表 1, 2 より，$\delta_1 > \delta_2 > \delta_3$ となっており，計算コストと誤差上限のトレードオフを確認した．また，表 2 で紹介した程度の行列サイズの場合は，高速な手法であっても実用的な誤差上限が得られる．次に行列の条件数を固定し，サイズの増加に伴う誤差上限の悪化率を紹介する．

図 1 より，δ_1 は行列サイズの増加に伴い，誤差上限の悪化率が δ_3 と比較して高い．n が 1000 から 10000 への増加で，δ_1 は 91 倍近く誤差上限が増加するが，δ_3 は 10 倍程度の増加である．つまり，δ_1 は小規模から中規模な行列に対して計算コストが低く有効な手法であるが，大規模行列に対応できない可能性がある．それに対して，δ_2, δ_3 の誤差

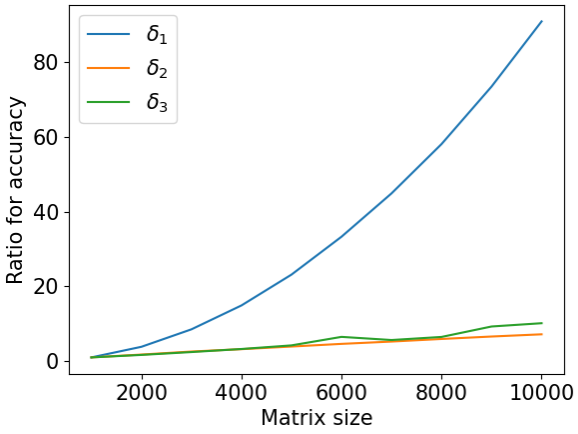

図 1 $n = 1000$ の場合の $\delta_1, \delta_2, \delta_3$ を 1 とした，サイズの増加に対する誤差上限の悪化率．(条件数は約 10^5，最大固有値は約 1 に設定)

上限の悪化率は低く，大規模な行列に対しても有向な可能性がある．しかし，δ_2 の悪化率は 7.2 倍程度で δ_3 より良い．つまり，問題が中小規模の場合は δ_3 は δ_2 よりタイトな誤差上限を計算できたが，超大規模な問題に対しては逆転する可能性がある．これは尾崎スプリットの特性上，行列サイズの増加に伴い，$A^{(1)}, X^{(1)}$ の情報量（仮数部における非ゼロビット数）が少なくなるためである．しかし，行列サイズが数 100 万次元程度ならば十分に高精度に計算が可能である．

次に、条件数と誤差上限の関係を紹介する．図 2 は行列サイズを 1000 に固定し，条件数を 10 から 10^{10} まで変化させた．

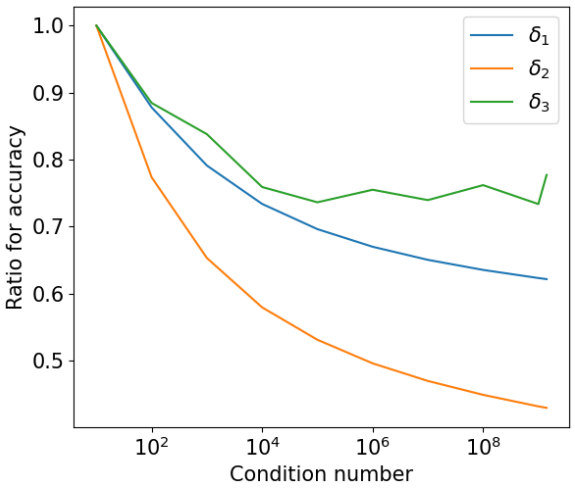

図 2 条件数が 10 の場合を 1 とした誤差上限の変化率．($n = 1000$，最大固有値は約 1 に設定)

この数値実験結果により，固有の精度保証法で得られる誤差上限は行列サイズと最大固有値に大きく依存すること

がわかる．また，精度保証を実行する際に，全近似固有値が得られているため，行列の条件数を近似的に求めることができる．よって，近似固有対の計算精度，条件数の近似値，行列サイズから，ユーザの目的にあう精度保証法を検討することが可能である．

4.2 速度

次に，計算速度に対する数値実験結果を紹介する．計算機環境は，スーパーコンピュータ「HOKUSAI BWMPC」，CPU：Intel Xeon Gold 6148 CPU @ 2.4GHz，Memory：96 GB/1 node, BLAS：Intel MKL，コンパイラ：icc version 19.0.5.281 を用いた．まず，「HOKUSAI BWMPC」上での固有値ソルバ（DSTEV，DSTEVD，DSTEVR）の性能比較を行う．また，（N）：全近似固有値のみの計算時間，（V）：全近似固有対の計算時間を示す．

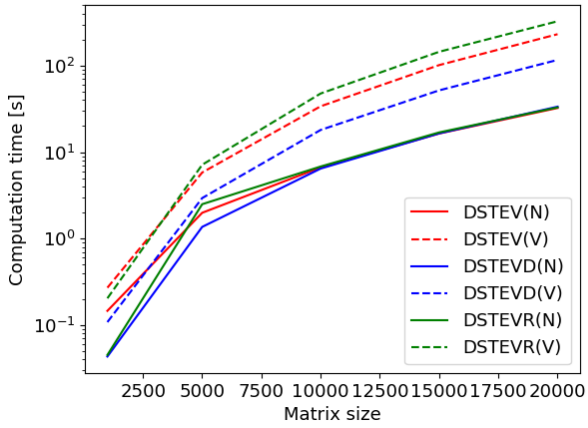

図 3 LAPACK の固有値ソルバに対する計算時間の比較．（N）：全固有値のみの計算時間，（V）：全固有対の計算時間．

図 3 より，DSTEVD が計算性能が優れているため，DSTEVD の計算時間と精度保証に要する計算時間の比較を行う．この環境では，有向丸めを用いた行列積の区間演算が機能しなかったため[*1]，最近点丸めのみで実行可能な提案手法のみを紹介する．また，比較する手法を以下のように表記する．

Veri： 最近点丸めのみを用いた高速な手法（3.2 節）

Veri_acc： 最近点丸めのみを用いた高精度な手法（3.3 節）

また，精度保証の計算時間には，全固有対を計算する時間は含まないものとする．

表 3 で，固有値計算ソルバ（DSYEVD）と精度保証の計算時間を示す．また，DSYEVD と精度保証の計算時間の比（精度保証の計算時間/固有値の計算時間）を図 4 に示す．次に，精度保証付きの固有値を計算する計算時間を以下のように表記する．

T1： Veri と DSYEVD(V) の計算時間

T2： Veri_acc と DSYEVD(V) の計算時間

このとき T1，T2 に対して全近似固有値のみの計算時間と全近似固有対の計算時間との比較を行う．

[*1] 丸め誤差が発生する行列積の区間包含に対して，有向丸めを用いた方法で計算した際に区間半径の値がすべて 0 となったため，有向丸めが作用していないと判断した．

表 3 固有値計算と精度保証の計算時間 [s]

n	DSYEVD(N)	DSYEVD(V)	Veri	Veri_acc
1000	4.31e-02	1.07e–01	1.61e–02	3.44e–02
5000	1.36e+00	2.66e+00	6.36e–01	1.43e+00
10000	6.45e+00	1.41e+01	3.81e+00	8.62e+00
15000	1.65e+01	4.23e+01	1.19e+01	2.65e+01
20000	3.36e+01	9.78e+01	2.68e+01	6.02e+01

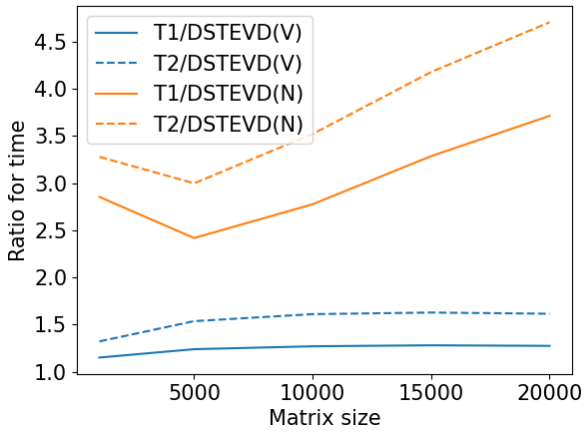

図 4 固有値計算ソルバ（DSYEVD）と全近似固有対の計算時間を含む精度保証法に対する計算時間の比較．

図 4 より，3.2 節で提案した手法は近似固有対を計算する時間と比較して 3 割弱で精度保証が可能である．また，3.3 節で提案した高精度計算を用いる手法でも，固有値計算の 6 割程度の計算時間で精度保証が完了した．しかし，近似固有値のみが必要な場合でも，精度保証法では全近似固有対を必要とする．そのため精度保証された全固有値を得るには，行列サイズが 20000 のときに全近似固有値のみを計算の 3.5〜4.5 倍程度の計算時間が必要となった．また，実装の観点からも，全近似固有ベクトルを保存する必要があるため，必要なメモリ量が相対的に増加する．

5. まとめと今後

手法の特徴は，最近点丸めのみを用いて全近似固有値に対する誤差上限を多くの計算機環境において高速に計算できる点である．1 ノードの計算機で，高速な手法は 3 割弱，高精度な手法は 6 割程度で精度保証に成功した．これは，精度に関する検算のために固有値問題を 2 回解くよりも，固有値問題を 1 回解いて精度保証を行った方が計算時間が短いことを意味する．

また，高精度計算を用いた提案手法は，大規模行列に対

してもタイトな誤差上限が得られることが期待できる．しかし，提案手法はノルム評価のために，絶対値最大の固有値と絶対値最小の固有値に対して同値の誤差上限を与える．そのため，絶対値最小付近の固有値に対して，過大評価になる可能性がある．よって，条件数の大きい問題に対しては，要素毎の誤差評価が可能な精度保証付き数値計算法の実装が必要である．

今後は，

- MPI 並列環境における大規模固有値問題への評価
- 固有ベクトルに対する精度保証法の実装と評価
- $\rho_{(p,n)}$ に対する評価式の改良
- ブロック計算を用いた丸め誤差解析の影響を低減した方法の実装と評価
- 一般化固有値問題への拡張

等を行う．

謝辞

本研究は，研究教育拠点（COE）形成推進事業「高性能・高信頼性数値計算手法の研究開発・展開と人材育成」の助成を受けた．

参考文献

[1] Hoshi, T., Ogita, T., Ozaki, K. and Terao, T.: An a posteriori verification method for generalized real-symmetric eigenvalue problems in large-scale electronic state calculations, *Journal of Computational and Applied Mathematics*, p. 112830 (2020).

[2] Miyajima, S.: Numerical enclosure for each eigenvalue in generalized eigenvalue problem, *Journal of Computational and Applied Mathematics*, Vol. 236, No. 9, pp. 2545–2552 (2012).

[3] Miyajima, S.: Fast enclosure for all eigenvalues and invariant subspaces in generalized eigenvalue problems, *SIAM Journal on Matrix Analysis and Applications*, Vol. 35, No. 3, pp. 1205–1225 (2014).

[4] Rump, S. M.: Computational error bounds for multiple or nearly multiple eigenvalues, *Linear Algebra and its Applications*, Vol. 324, No. 1-3, pp. 209–226 (2001).

[5] Oishi, S.: Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled computation, *Linear algebra and its Applications*, Vol. 324, No. 1-3, pp. 133–146 (2001).

[6] Yamamoto, T.: Error bounds for computed eigenvalues and eigenvectors, *Numerische Mathematik*, Vol. 34, No. 2, pp. 189–199 (1980).

[7] 大石進一編：精度保証付き数値計算の基礎，コロナ社 (2018).

[8] Rump, S. M.: Fast and parallel interval arithmetic, *BIT Numerical Mathematics*, Vol. 39, No. 3, pp. 534–554 (1999).

[9] Rump, S. M., Ogita, T., Morikura, Y. and Oishi, S.: Interval arithmetic with fixed rounding mode, *Nonlinear Theory and Its Applications, IEICE*, Vol. 7, No. 3, pp. 362–373 (2016).

[10] Ogita, T., Rump, S. M. and Oishi, S.: Verified solution of linear systems without directed rounding, *Advanced Research Institute for Science and Engineering, Waseda University* (2005).

[11] Morikura, Y., Ozaki, K. and Oishi, S.: Verification methods for linear systems using ufp estimation with rounding-to-nearest, *Nonlinear Theory and Its Applications, IEICE*, Vol. 4, No. 1, pp. 12–22 (2013).

[12] 尾崎克久，荻田武史，大石進一：有向丸めの変更を使用しないタイトな行列積の包含方法，応用数理，Vol. 21, No. 3, pp. 186–196 (2011).

[13] IEEE: *ANSI/IEEE 754-2008: IEEE Standard for Floating-Point Arithmetic*, New York (2008).

[14] Rump, S. M. and Lange, M.: On the definition of unit roundoff, *BIT Numerical Mathematics*, Vol. 56, No. 1, pp. 309–317 (2016).

[15] Jeannerod, C.-P. and Rump, S. M.: Improved error bounds for inner products in floating-point arithmetic, *SIAM Journal on Matrix Analysis and Applications*, Vol. 34, No. 2, pp. 338–344 (2013).

[16] Higham, N. J.: *Accuracy and stability of numerical algorithms*, SIAM (2002).

[17] 太田貴久，荻田武史，大石進一：悪条件連立一次方程式の精度保証付き数値計算法，日本応用数理学会論文誌，Vol. 15, No. 3, pp. 269–286 (2005).

[18] Ozaki, K., Ogita, T., Oishi, S. and Rump, S. M.: Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications, *Numerical Algorithms*, Vol. 59, No. 1, pp. 95–118 (2012).

[19] Mukunoki, D., Ogita, T. and Ozaki, K.: Reproducible BLAS routines with tunable accuracy using ozaki scheme for many-core architectures, *International Conference on Parallel Processing and Applied Mathematics*, Springer, pp. 516–527 (2020).

[20] 石黒史也，片桐孝洋，大島聡史，　永井亨，荻野正雄：GPGPU による高精度行列-行列積アルゴリズムのための Batched BLAS を用いた実装方式の提案, 研究報告ハイパフォーマンスコンピューティング (HPC)，Vol. 2018, No. 32, pp. 1–8 (2018).

[21] Ichimura, S., Katagiri, T., Ozaki, K., Ogita, T. and Nagai, T.: Threaded accurate matrix-matrix multiplications with sparse matrix-vector multiplications, *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, pp. 1093–1102 (2018).