

スパース行列に対する固有値の厳密計算手法の開発

柳澤 優香^{*}, 劉 雪峰[†], 大石 進一[‡]

1 はじめに

本論文の目的は、行列の一般化固有値問題

$$Ax = \lambda Bx \quad (1)$$

の特定の固有値 λ_i とその大きさの順位まで込めて、厳密に評価する方法の構築である。式 (1) において $x \in \mathbb{R}$ は非ゼロベクトルとし、 A を $n \times n$ の実対称行列、 B を $n \times n$ の実対称正定値行列とする。式 (1) の固有値は $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ の順に並んでいるとする。

我々は、Lehman-Goerisch の方法とブロック LDL 分解を用いたシルベスターの慣性律に基づく方法を用いて、大規模な疎行列に適用可能な高速かつ高精度な手法を構築する。

連立一次方程式、固有値問題などの数値線形代数の近似計算手法は多く研究されているが、本論文では得られた数値解により真の解の存在を考察し、真の解が存在するとしたら真の解と数値解の誤差を厳密に評価する計算方法を扱う。このような手法を精度保証付き数値計算と言う。特に近年、非線形偏微分方程式をはじめとする関数方程式の解に対する数値的検証法（例えば [1]）の過程で微分作用素の固有値問題が現れ、その離散化として行列の一般化固有値問題における固有値の厳密な評価が不可欠である。さらに固有値の存在と非存在の範囲を検討するとき、固有値の順位を確定するのも重要である。従って、一般化固有値問題の数値解に対する高品質な評価を実用的な計算時間で安定的に得られる方法が必要となる。

そこで、ある近似の固有値が与えられたとき、演算の結果を包含する区間演算を用いて、固有値の存在範囲を精度保証付き数値計算によって厳密に確定する方法を紹介する。例えば、Oishi らの方法 [2] はゲルシュゴリンの定理を利用して全ての固有値を精度保証することを目的としており、問題が小規模でかつ B が悪条件（計算途中に混入する微小な丸め誤差に対して計算の誤差が増大する問題）でない密行列であれば、十分実用的である。しかしながら、実際の応用上、全ての固有値や固有ベクトルに対する厳密な評価は必要なく、優越固有値（絶対値が最大の固有値）や、特定の固有値の厳密な評価とその大きさの順位を厳密に得られれば十分である。例えば、有限要素法を利用して非線形微分方程式の解を検証するとき、メッシュ分割が細くなるほど数万次元など大規模な行列が現れる場合が

¹早稲田大学 理工学術院総合研究所 (yuuka@aoni.waseda.jp).

²新潟大学 新潟大学理学部数学科

³早稲田大学 理工学術院

多く、Oishi らの方法のように全ての固有値の厳密な評価を行うには莫大な計算コストが必要である。実際の問題では特定の固有値のみを算出すれば十分であり、現れる問題のほとんどは疎行列（成分のほとんどが零である行列）である。Oishi らの方法は逆行列を経由するため疎行列には適用できない。ここで、疎行列にも対応可能で特定の固有値への評価方法として、Yamamoto の方法 [3] がある。原理としては、行列に関するシルベスターの慣性律（実際はブロック LDL 分解を用いる）、Wely の単調性の定理を用いて負の固有値の個数を調べ、近似固有値の近傍にある固有値の大きさとその順位を精度保証する。当手法は実装が容易なことから頻繁に使用されているが、精度の良い評価を得るために、二分法のように試行計算が必要となり、計算コストの増大と厳密解への遅い収束が懸念される。

本論文において、真の固有値が含まれたシャープな区間とその大きさの順位を厳密に評価する方法の構築を行う。アイデアとしては、Yamamoto の方法 [3] に使用されるシルベスターの慣性律 [3] を用いて、特定の近似固有値の順位と粗い評価を得て、さらに Lehman-Goerisch の定理 [4] を適用し、特定の固有値が含まれたシャープな区間を得るというものである。Lehman-Goerisch の定理は、Kato の固有値評価の理論 [6] を拡張したもので、 λ_i の粗い評価から少ない計算量でシャープな評価を得る定理である。我々は、Yamamoto の方法 [3] から Lehmann-Goerisch の定理に必要な λ_i の粗い評価を得て、ロバストかつ効率的な固有値の計算手法の確立する。具体的なステップは次の通りであり、次章から具体的な手法と手順を説明する。

1. B の最小固有値の下限を高速に評価
2. Yamamoto の方法 [3] を基に近似固有値 $\tilde{\lambda}$ の近傍にある真の固有値の粗い評価と大きさの順位を得る。
3. Step1 と 2 の事前情報を基に、Lehmann-Goerisch [4] の定理から真の固有値を含むシャープな区間を得る。

2 実対称行列の最小固有値に対する高速評価法

本章において我々は式 (1) の B の最小固有値の下限を評価（粗い評価を許容）する方法を提案する。シャープな評価でなくても、式 (1) の高精度な固有値評価（詳細は 4 章を参照のこと）を得ることに十分に役立つため、シャープな評価である必要はないことに注意する。従って、本章で提案する B の最小固有値を評価するアルゴリズムの特徴は、粗い評価を許容し高速かつロバストな（反復または試行錯誤計算なし）計算手法である。

準備 $\mathbb{F}$ は浮動小数点数の集合で、 $\mathbb{F} \subset \mathbb{R}$ である。IEEE754 規格の倍精度浮動小数点演算の場合、仮数部の相対精度は $2^{-53} = 1.1 \times 10^{-16}$ である [5]。これを $\mathbf{u}$ と定義する。 $f(\cdot)$ は括弧内の演算を浮動小数点演算で評価することを意味し、 $a, b \in \mathbb{F}$ について、 $\circ \in \{+, -, *, /\}$ とすると、

$$f(a \circ b) = (1 + \varepsilon)(a \circ b), \quad |\varepsilon| \leq \mathbf{u}.$$

$0 \leq a \in \mathbb{F}$ について,

$$fl(\sqrt{a}) = \sqrt{a}(1 + \varepsilon_1), \quad |\varepsilon_2| \leq \mathbf{u}.$$

コレスキー分解 浮動小数点演算によるコレスキー分解¹は、実対称行列の最小固有値を評価する上で重要な役割を果たす。コレスキー分解にはいくつかのバリエーションがあり、その大部分は以下のアルゴリズムである²。実対称行列 $B = (b_{ij}) \in \mathbb{R}^{n \times n}$ が正定値であれば $B = R^T R$ ($R \in \mathbb{R}^{n \times n}$ は上三角行列) の形に分解できる。具体的な算法は次のように与えられる。

アルゴリズム 2.1. コレスキー分解

```

for  $j = 1 : n$ 
  for  $i = 1 : j - 1$ 
     $r_{ij} = (b_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}) / r_{ii}$ 
  end
   $r_{jj} = (b_{jj} - \sum_{k=1}^{j-1} r_{kj}^2)^{1/2}$ 
end

```

先行研究 B の最小固有値の下限を評価する方法として、Rump の方法 [8] がある。 B の近似最小固有値 $\hat{\lambda}_B$ を計算し、ある微小な数 $0 < \tau \ll 1$ より $\hat{\lambda}_B := (1 - \tau)\bar{\lambda}_B$ のようにコレスキー分解が数値的に破綻することを防ぐために $\bar{\lambda}_B$ より少し小さく設定する。 $\bar{B} := B - \hat{\lambda}_B I$ のコレスキー分解を $\bar{R} := \text{chol}(\bar{B})$ とし、 $E := \bar{R}^T \bar{R} - \bar{B}$ 、つまりコレスキー分解の後退誤差を E_1 と定義し $\|E\|_2$ の上界を $e \in \mathbb{F}$ とすると、

$$\lambda_{\min}(B) \geq \hat{\lambda}_B - e =: \beta$$

β は B の最小固有値の下限である。適切な $\hat{\lambda}_B$ を選ぶ必要があるため、試行計算を繰り返すと結果的に計算量が増大してしまう。そこで、粗い評価を許容し高速でロバストな (適切なパラメータを探索して試行することを繰り返さない) 手法を提案する。具体的には、コレスキー分解の破綻を防ぐためのシフト [8] を適用し、Rump-Ogita の定理 [9] を用いる。

定理 2.1 (定理 1 [8]). $B \in \mathbb{F}^{n \times n}$, $B = B^T$ とし、 $(n+1)(n+3)\mathbf{u} < 1$ を満たすと仮定する。

$$\text{shift}(B) := c_n \mathbf{u} \cdot \text{Trace}(B), \quad c_n = \frac{n+2}{1 - (n+1)(n+3)\mathbf{u}}$$

と定義し、対角シフト量を $\delta := \text{shift}(B)$ と置く。 B が正定値であれば、 $\bar{B} := fl(B + \delta I)$ の Cholesky 分解は成功する。

¹疎行列に対してコレスキー分解を適用すると fill-in(非零要素)が発生するため、最小次数法などの手法を基に行列の行と列を並べ替えることで、分解で得られる行列の非零要素の量を減らし、それにより計算時間やストレージコストを減らすことが可能。

²例えば、Matlab では chol, LAPACK では xPOTRF でサポートされている。

定理 2.2 (定理 2.3 [9]). $B \in \mathbb{F}^{n \times n}$ は実対称行列で, b_{jj} は B の対角成分とする. φ_k を $\varphi_k := \frac{k\mathbf{u}}{1-2k\mathbf{u}}$ と定義し, オーバーフローやアンダフローが起きないと仮定すると,

1. $\lambda_{\min}(B) \geq \sum_{j=1}^n \varphi_{j+1} b_{jj}$ ならば, $\text{chol}(B)$ は成功する.

2. $\lambda_{\min}(B) < -\sum_{j=1}^n \varphi_{j+1} b_{jj}$ ならば, $\text{chol}(B)$ は負の数の平方根が現れ途中で失敗する.

2 の反対を考えると, $\text{chol}(B)$ が成功したら $\lambda_{\min}(B) \geq -\sum_{j=1}^n \varphi_{j+1} b_{jj}$ となることがわかる. これら 2 点を用いてロバストで高速な手法を提案する.

提案手法 B の近似最小固有値 $\tilde{\lambda}_B$ を計算し, $\tilde{B} := B - \tilde{\lambda}_B I$ とするが, $\tilde{B}$ はほとんど singular である. コレスキー分解の破綻を防ぐために定理 2.1 より $\tilde{B}_\delta := fl(\tilde{B} + \delta I)$, $\delta := \text{shift}(\tilde{B})$ とする. $\text{chol}(\tilde{B}_\delta)$ が成功すれば, 定理 2.2 より

$$\lambda_{\min}(\tilde{B}_\delta) \geq -\sum_{j=1}^n \varphi_{j+1} b_{\delta jj}$$

であるから,

$$\lambda_{\min}(B) \geq \tilde{\lambda}_B - \delta - \sum_{j=1}^n \varphi_{j+1} b_{\delta jj} - c_1 \mathbf{u} - c_2 \mathbf{u} =: \beta'$$

β' は最小固有値の下限. ただし, $c_1 := \max_{1 \leq i \leq n} (|\tilde{b}_{ii} + \delta|)$, $c_2 := \max_{1 \leq i \leq n} (|b_{ii} - \lambda|)$. 計算量は, コレスキー分解 1 回をのみのロバストな手法である³.

数値実験 提案手法と Rump の方法 [8] を精度の観点から比較するために, 次のテスト行列 $B \in \mathbb{F}^{n \times n}$ [10] を用いる.

$$\begin{aligned} B_{ij} &= \min(n-i+1, n-j+1), \quad i = 1, \dots, n \\ \lambda_{\min}(B) &= \frac{1}{2} \left[1 - \cos \frac{(2n-1)\pi}{2n+1} \right]^{-1}. \end{aligned}$$

n を大きくすると $2n-1 \approx 2n+1$ なので最小固有値は 0.25 に収束する. n のサイズを 64 から 8192 まで変化させ, それぞれの場合で得られた最小固有値の下限を表 1 に示す.

Rump の手法 [8] を適用するには, 適切な ε をを見つけるために多くの試行が必要であった. 例えば, $n = 8192$ の場合, 12 回の試行を行った結果である. 一方, 提案手法は 1 回で完了することができた. ただし, 表 1 の通り n が大きくなるにつれて Rump の手法より精度が少し悪くなる傾向がある.

³ $\tilde{B}$ が正定値でない場合でも定理 2.1 のシフト量は少し大きめに評価されているため, ほとんどの場合成功する. 実験的にこの方法でコレスキー分解が破綻した例は観察されていない.

表 1: テスト行列の $B \in \mathbb{F}^{n \times n}$ の最小固有値に対する下限の評価 [10]

n	$\tilde{\lambda}_{\min}(B)$	Rump's method [8]	Proposed method
64	0.2501483310.	<u>0.250148331023562</u>	<u>0.250148331030732</u>
256	0.250009375...	<u>0.250009375792001</u>	<u>0.250009374705642</u>
1024	0.250000587...	<u>0.250000581273825</u>	<u>0.250000507980726</u>
4096	0.250000036...	<u>0.249999915311914</u>	<u>0.249994946537337</u>
8192	0.250000009...	<u>0.249999510450566</u>	<u>0.24995930325251</u>

3 ブロック LDL 分解を用いた簡便な精度保証法

本章では、式 (1) の特定の固有値 λ_i とその大きさの順位まで込めて、厳密に評価する方法 [3] を解説する。本手法はブロック LDL 分解を用いたシルベスターの慣性律と誤差評価に基づく方法である。 $A \in \mathbb{R}^{n \times n}$, $A = A^T$ のブロック LDL 分解⁴ はブロック対角行列 $D \in \mathbb{R}^{n \times n}$ と対角成分が 1 の下三角行列 $L \in \mathbb{R}^{n \times n}$ に分解される。

定理 3.1. 任意の実対称行列 $A \in \mathbb{R}^{n \times n}$ がブロック対角行列 $D \in \mathbb{R}^{n \times n}$ と下三角行列 $L \in \mathbb{R}^{n \times n}$ によって、 $A = LDL^T$ と表される。この時、 A と D の固有値のうち負の個数は一致する。

本定理を計算機上で厳密に評価する（粗い評価を許容）ためには、LDL 分解の丸め誤差を考慮する必要があり、次の計算ステップとなる： $A \in \mathbb{F}^{n \times n}$ の i 番目の固有値 λ_i の近似固有値 $\tilde{\lambda}$ の近傍の 3 つの近似固有値を算出 (Matlab では `eigs(A, B, 3,  $\tilde{\lambda}$ )`)。隣の近似固有値までの距離の半分をそれぞれ δ_1, δ_2 ($\delta_1 < \delta_2$) とする。

1. 正の定数 δ_1, δ_2 によって、

$$Y_1 := A - \delta_1 B$$

$$Y_2 := A - \delta_2 B$$

のそれぞれにブロック LDL 分解を適用し、 $Y_1 \approx L_1 D_1 L_1^T$, $Y_2 \approx L_2 D_2 L_2^T$ とする。

2. ϵ_1, ϵ_2 を厳密な上限を計算する。ただし、 λ_B は B の最小固有値の下限とする（詳細は 2 章を参照のこと）。

$$\epsilon_1 := \frac{1}{\lambda_B} \|A - (\tilde{\lambda} - \delta_1)B - L_1 D_1 L_1^T\|_{\infty} \quad (2)$$

$$\epsilon_2 := \frac{1}{\lambda_B} \|A - (\tilde{\lambda} + \delta_2)B - L_2 D_2 L_2^T\|_{\infty} \quad (3)$$

⁴Matlab では `ldl` でサポートされている。コレスキー分解と同様に疎行列に適用すると fill-in が発生するが、行列の非零要素の量を減らす手法（例えば、[11]）を選択可能。

3. D_1 が $i-1$ 個, D_2 が $i+r$ の負の要素を持つとすると, λ_i から λ_{i+r} までが以下の区間に存在する ($i > 0, r \geq 0$).

$$[\tilde{\lambda} - \delta_1 - \epsilon_1, \tilde{\lambda} + \delta_2 + \epsilon_2]$$

4 高精度な固有値の厳密計算手法

本章では, 式 (1) のある特定の固有値が包含されたシャープな区間を得ることができる Lehman-Goerisch の定理を解説する. 式 (1) のある近似固有対を $(\tilde{\lambda}, \tilde{u})$ とし, 开区間 (ρ, σ) は i 番目の固有値 λ_i 以外含まないとする. 事前情報 $\rho, \sigma \in \mathbb{R}$ は 3 章の手法で得られているものとする. また, B の最小固有値の下限は 2 章の手法で得られているものとする.

定理 4.1 (Goerisch [4], 1990). $A \in \mathbb{R}^{n \times n}$ と $B \in \mathbb{R}^{n \times n}$ は $A = A^T, B = B^T, B$ とし, B は正定値とする. B の最小固有値の下限を $c > 0$ とする.

1. $\tilde{v} \approx B^{-1}A\tilde{u}$, つまり連立一次方程式 $Bv = A\tilde{u}$ の v について計算し, 近似解を $\tilde{v}$ とする.
2. $A_0, A_1, A_2, \hat{a}, \hat{b}$ をそれぞれ次のように定義する.

$$\begin{aligned} A_0 &:= (\tilde{u}^T B \tilde{u}) \\ A_1 &:= (\tilde{u}^T A \tilde{u}) \\ A_2 &:= (\tilde{u}^T A \tilde{v} - \tilde{v}^T (B \tilde{v} - A \tilde{u}) + \frac{1}{c} (B \tilde{v} - A \tilde{u})^T (B \tilde{v} - A \tilde{u})) \\ \hat{a} &:= A_1 - \sigma A_0 \\ \hat{b} &:= A_2 - 2\sigma A_1 + \sigma^2 A_0 \end{aligned}$$

3. $\mu := \frac{\hat{a}}{\hat{b}} < 0$ について, $\sigma + \frac{1}{\mu} < \lambda_i$.

定理 4.1 を $-Ax = Bx$, $\sigma = -\rho$ として適用すれば λ_i の上限が得られる.

4.1 数値実験

本章では, 提案手法をテスト行列に適用し, 式 (1) の固有値 $\lambda_i (i = 1 \dots n)$ に対して得られた数値解よりその大きさの順位を込めて, 厳密な評価を行い, 有用性を検証する. 計算環境は, 下記の通りである.

CPU: Intel Core i7, 2GHz, 2Core

OS : Mac OS X version 10.13

Software: MATLAB R2017a

表 2: テスト行列 (4), (5) の各固有値の厳密評価

i	$\tilde{\lambda}$	Step1&2	Step3
1	$5.896438502882 \cdot 10^{-01}$	$[-18.1887267, 19.71808020] \cdot 10^{-1}$	$5.89643850288_{78}^{97} \cdot 10^{-1}$
2	$3.35429531632924 \cdot 10^0$	$[1.972129897, 20.75002052] \cdot 10^0$	$3.35429531632_{794}^{941} \cdot 10^0$
3	$3.814923768266030 \cdot 10^{01}$	$[2.075350049, 5.692436800] \cdot 10^{01}$	$3.8149237682_{58368}^{68934} \cdot 10^{01}$
4	$6.81343664169345 \cdot 10^{02}$	$[3.597752815, 10.2.256139] \cdot 10^{02}$	$6.813436641_{46855}^{78389} \cdot 10^{02}$
5	$1.83926874347035 \cdot 10^{04}$	$[9.537792889, 2.756771861] \cdot 10^{04}$	$1.83926874_{211019}^{400365} \cdot 10^{04}$
6	$7.72599247352213 \cdot 10^{05}$	$[3.955283701, 11.58464329] \cdot 10^{05}$	$7.725992197_{77635}^{993412} \cdot 10^{05}$
7	$5.5590338.2373547 \cdot 10^{07}$	$[2.818381485, 8.336959127] \cdot 10^{07}$	$5.5590_{2212730313}^{3828128531} \cdot 10^{07}$
8	$8.99653531497995 \cdot 10^{09}$	$[4.526422152, 13.49135669] \cdot 10^{09}$	$8.99_{387781323449}^{753539747641} \cdot 10^{09}$

まず, 8×8 のテスト区間行列 [4] に対して検証を行う. B はヒルベルト行列であり, $\text{cond}(B) \approx 1.5257 \cdot 10^{10}$ である. 数値実験結果を表 2 に示す.

$$A = (a_{ij})_{i,j=1,\dots,8} = 1, \quad a_{ij} = 0 \quad (4)$$

$$B = [b_{ij}]_{i,j=1,\dots,8} = \left[\frac{1}{i+j-1} - 10^{-13}, \frac{1}{i+j-1} + 10^{-13} \right] \quad (5)$$

次に, 10×10 のテスト行列 [4] に対して検証を行う. B はヒルベルト行列であり, $\text{cond}(B) \approx 1.6027 \cdot 10^{13}$ である. 数値実験結果を表 3 に示す.

$$A = \begin{pmatrix} 5 & -4 & 1 & & & & & \\ -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & 1 & -4 & 6 & -4 \\ & & & & & 1 & -4 & 5 \end{pmatrix} \quad (6)$$

$$B = (b_{ij}) = \left(\frac{l}{i+j-1} \right)_{i,j=1,\dots,10}, \quad l := 232792560, \quad (7)$$

表 2.3 の通り, 固有値山本の定理から得た固有値の評価は非常に粗いが, Lehmann-Goerisch の定理に必要となる事前情報として有益であり, 実際に Lehmann-Goerisch の定理から得た固有値の評価は著しく改善されていることが明らかである.

表 3: テスト行列 (6), (7) の各固有値の厳密評価

i	$\tilde{\lambda}$	Step1&2	Step3
1	$2.697325009 \cdot 10^{-11}$	$[-4.27447210, 5.50079193] \cdot 10^{-11}$	$2.6973250090_{518}^{696} \cdot 10^{-11}$
2	$2.687554297 \cdot 10^{-09}$	$[2.14306018, 3.30160041] \cdot 10^{-09}$	$2.6875542970_{269}^{273} \cdot 10^{-09}$
3	$8.337070281 \cdot 10^{-08}$	$[8.20702589, 8.46557367] \cdot 10^{-08}$	$833707028114_{102}^{10} \cdot 10^{-08}$
4	$2.30613899 \cdot 10^{-06}$	$[2.27731640, 2.33301836] \cdot 10^{-06}$	$2.30613899574_{53}^{67} \cdot 10^{-06}$
5	$7.73369231 \cdot 10^{-05}$	$[7.62904207, 7.83301357] \cdot 10^{-05}$	$7.733692318_{3954}^{4593} \cdot 10^{-05}$
6	$3.33832070 \cdot 10^{-03}$	$[3.30426869, 3.37866516] \cdot 10^{-03}$	$3.33832070_{416}^{503} \cdot 10^{-03}$
7	$1.91995342 \cdot 10^{-01}$	$[1.89646576, 1.94233679] \cdot 10^{-01}$	$1.9199534_{170968}^{290704} \cdot 10^{-01}$
8	$1.56094806 \cdot 10^{01}$	$[1.54267114, 1.581426671] \cdot 10^{01}$	$1.56094_{7685126519}^{8164076014} \cdot 10^{01}$
9	$2.01464194 \cdot 10^{03}$	$[1.98817533, 2.041476890] \cdot 10^{03}$	$2.01460_{342643809}^{5175389956} \cdot 10^{03}$
10	$5.50593300 \cdot 10^{05}$	$[5.43580230, 5.581901414] \cdot 10^{05}$	$5_{48785833903136}^{51025716551162} \cdot 10^{05}$

5 まとめ

本論文において、ロバストかつ効率的な固有値の厳密計算手法を確立した。本手法は次のステップで実現する：

- 1. B の最小固有値の下限を高速に算出する方法を提案
- 2. 山本の定理 [3] より、近似固有値 $\tilde{\lambda}$ の近傍にある真の固有値の粗い評価と大きさの順位を得る。
- 3. 得られた粗い評価に対して、Lehmann-Goerisch の定理 [4] を適用。

数値実験例では、条件数大きい問題に対しても、真の固有値を含む区間を得ることができたため、十分に実用的な計算手法であると期待できる。今回はある一つの λ_i に対しての評価を実施したが、与えられた近似固有値 $\tilde{\lambda}$ に対してその近傍にある複数個のそれぞれの真の固有値を含む区間とその大きさの順位を厳密に評価する方法を確立させたいと考えている。

参考文献

[1] X. Liu, S. Oishi: On guaranteed eigenvalue estimation of compact differential operator with singularity, Proceedings of 2012 International Symposium on Nonlinear Theory and its Applications (NOLTA 2012), 2012, 812-815.

[2] K. Maruyama, T. Ogita, Y. Nakaya, and S. Oishi: Numerical inclusion method for all eigenvalues of real symmetric definite generalized eigenvalue problem, IEICE Trans., J87-A (2004), pp. 1111-1119, (in Japanese).

- [3] N. Yamamoto: A simple method for error bounds of eigenvalues of symmetric matrices, *Linear Algebra Appl.*, 324 (2001), pp. 227–234.
- [4] H. Behnke: The calculation of guaranteed bounds for eigenvalues using complementary variational principles, *Computing*, 47 (1991), pp. 11–27.
- [5] N. J. Higham: *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, USA, 2nd edition, 2002.
- [6] T. Kato, On the upper and lower bounds of eigenvalues, *J. Phys. Soc. Jpn.* 4 (4) (1949) 334–339.
- [7] S.M. Rump. Validated Solution of Large Linear Systems, *Validation numerics: theory and applications*, vol. 9, pp.191–212. Springer (1993).
- [8] Y. Yanagisawa, T. Ogita, S. Oishi: A modified algorithm for accurate inverse Cholesky factorization, *Nonlinear Theory and Its Applications*”, vol.5, no.1 (2014), pp.35–46.
- [9] S. M. Rump and T. Ogita: Super-fast validated solution of linear systems. *J. Comput. Appl. Math.*, 199(2):199–206, 2007.
- [10] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, 1967, Chapter 19.
- [11] I. S. Duff, MA57–A new code for the solution of sparse symmetric definite and indefinite systems.” Technical Report RAL-TR-2002-024, Rutherford Appleton Laboratory, 2002.