2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

Fast Error Estimation with Guaranteed Accuracy for Eigenvalues of Symmetric Matrix without Directed Rounding

Shinya Miyajima†, Takeshi Ogita†, Katsuhisa Ozaki† and Shin'ichi Oishi‡

†Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo Shinjuku-ku, Tokyo 169-8555 Japan
‡School of Science and Engineering, Waseda University
3-14-9 Okubo Shinjuku-ku, Tokyo 169-8555 Japan
Email: shinya_miyajima@fuji.waseda.jp, ogita@waseda.jp, k_ozaki@suou.waseda.jp, oishi@waseda.jp

Abstract—A new verification method is proposed for calculating the guaranteed error bounds of the approximate eigenvalues of real symmetric matrix without directed rounding. In this method, the guaranteed error bounds of the approximate eigenvalues are computed only in rounding-to-nearest mode. And this paper includes some numerical examples to show the property of the new method.

1. Introduction

In this paper, we consider calculating guaranteed error bounds of approximate eigenvalues for

$$Ax = \lambda x \tag{1}$$

where A is an $n \times n$ real symmetric matrix and λ is an eigenvalue and x is an eigenvector corresponding to λ. To calculate the guaranteed error bounds of the approximate eigenvalues, Oishi [1] modified the Weyl's theorem (e.g. [2][3]) suited for numerical inclusion of matrix eigenvalues. By this modification, for true eigenvalues λ_i $(i = 1, \cdots, n)$, it holds that

$$|\lambda_i - \tilde{\lambda_i}| \le |\tilde{\lambda_i}| ||PP^T - I||_2 + ||PDP^T - A||_2 \tag{2}$$

where $\tilde{\lambda_i}$ are computed (approximate) eigenvalues and P is an approximately orthogonal matrix whose column vectors are corresponding approximate eigenvectors and D is a diagonal matrix whose diagonal elements are approximate eigenvalues of A. Oishi proposed a fast verification method for calculating the guaranteed error bounds of the approximate eigenvalues with directed rounding introducing the rounding mode instructions defined by IEEE 754 standard for floating point arithmetic [4].

The purpose of this paper is to propose a new verification method for calculating the guaranteed error bounds of the approximate eigenvalues without directed rounding. In this method, the guaranteed error bounds of the approximate eigenvalues are computed only in rounding-to-nearest mode. By applying the new verification method, the guaranteed error bounds for the approximate eigenvalues of real symmetric matrix are calculated no matter though the working environment does not support the rounding mode instructions (e.g. Java and FORTRAN 77). Finally, this paper includes some numerical examples to show the property of the new method.

2. Floating Point Arithmetic

We assume that floating point arithmetic in this paper adheres to IEEE 754 standard. Let $\mathbb{F}$ be the set of floating point numbers. Let $\mathrm{fl}(\cdot)$ be the result of a floating point computation, where all operations inside parentheses are executed by ordinary floating point arithmetic. We assume that over/underflow do not occur. For $a, b \in \mathbb{F}$ and $\circ \in \{+, -, \cdot, /\}$, floating point operations according to IEEE 754 satisfy

$$|a \circ b| \le (1 + \mathbf{u}) \cdot |\mathrm{fl(a \circ b)}| \tag{3}$$

for $\mathbf{u}$ denoting the unit roundoff. Here, we cite the following definitions:

Definition 1 *Let $\mathbf{u}$ be the unit roundoff (especially, $\mathbf{u} = 2^{-53}$ in IEEE 754 double precision). Then, the constants $\gamma_{m,n}$ and $\tilde{\gamma}_{m,n}$ for $m, n \in \mathbb{N}$ are defined as follows:*

$$\gamma_{m,n} := \frac{m\mathbf{u}}{1 - n\mathbf{u}}, \quad \tilde{\gamma}_{m,n} := \mathrm{fl}(\frac{m\mathbf{u}}{1 - n\mathbf{u}}) \quad . \tag{4}$$

To simplify the description, when $m = n$, we define γ_n and $\tilde{\gamma}_n$ as follows:

$$\gamma_n := \gamma_{n,n}, \quad \tilde{\gamma}_n := \tilde{\gamma}_{n,n} \quad . \tag{5}$$

For $p \in \mathbb{N}$ and $a \in \mathbb{F}$, we present fundamental properties with $\mathbf{u}$ utilizing a priori error estimation (see [5]).

$$(1 + \mathbf{u})^n \le \frac{1}{1 - n\mathbf{u}} = 1 + \gamma_n \tag{6}$$

$$\gamma_{m,n}(1 + \mathbf{u})^p \le \gamma_{m,n+p} \tag{7}$$

$$(1 + \mathbf{u})^n |a| \le \frac{1}{1 - n\mathbf{u}}|a| \le \mathrm{fl}(\frac{|a|}{1 - (n+1)\mathbf{u}}) \tag{8}$$

$$\gamma_{m,n}|a| \le \mathrm{fl}(\tilde{\gamma}_{m,n+2}|a|) \tag{9}$$

Next, we cite the following definitions:

Definition 2 *For $x \in \mathbb{F}^n$ and $A \in \mathbb{F}^{n\times n}$, a norm $||\cdot||_\infty$ is defined as follows:*

$$||x||_\infty = \max_{1\le i\le n}|x_i|, \quad ||A||_\infty = \max_{1\le i\le n}(\sum_{j=1}^{n}|x_{ij}|) \quad . \tag{10}$$

We present properties of vector and matrix operations between real arithmetic and floating point arithmetic (see [5]). For $x, y \in \mathbb{F}^n$,

$$\begin{aligned} ||x||_\infty &= \mathrm{fl}(||x||_\infty) & (11)\\ \sum_{i=1}^{n}|x_i| &\le (1+\mathbf{u})^{n-1}\cdot \mathrm{fl}(\sum_{i=1}^{n}|x_i|) & (12)\\ |x^T||y| &\le (1+\mathbf{u})^{n}\cdot \mathrm{fl}(|x^T||y|) & (13)\\ |\mathrm{fl}(x^T y) - x^T y| &\le \gamma_n |x^T||y| \quad . & (14)\end{aligned}$$

For an vector $e = (1, \cdots, 1)^T$,

$$|x^T|e = \sum_{i=1}^{n}|x_i| \le (1+\mathbf{u})^{n-1}\cdot \mathrm{fl}(|x^T|e) \quad . \tag{15}$$

For $A \in \mathbb{F}^{n\times n}$,

$$||A||_\infty \le (1+\mathbf{u})^{n-1}\cdot \mathrm{fl}(||A||_\infty) \quad . \tag{16}$$

From Eq. (14), we obtain

$$|x^T y| \le |\mathrm{fl}(x^T y)| + \gamma_n |x^T||y| \quad . \tag{17}$$

Let p and s be natural numbers for which $\mathbf{u} = 2^{-p}$ and $s = \lceil p/2 \rceil$ hold (in IEEE 754 double precision, $p = 53$ and $s = 27$). Introducing s, Dekker [6] proposed the algorithm "Split" which splits a floating point number $a \in \mathbb{F}$ into two parts x and y where both parts have at most $s-1$ nonzero bits. Introducing this algorithm, Veltkamp (see [6]) proposed the algorithm "TwoProduct" which transform the product ab ($a, b \in \mathbb{F}$) into $x, y \in \mathbb{F}$. And this algorithm is able to be expanded into the following algorithm (throughout this paper, we express algorithms MATLAB-like [7]):

Algorithm 1 *Calculation of $x, y \in \mathbb{F}^n$ such that $\alpha v = x + y$, ($\alpha \in \mathbb{F}, v \in \mathbb{F}^n$) without error.*

```
function [x, y] = TwoProductSV(α, v)
x = fl(α * v);
[α1, α2] = Split(α);   % α = α1 + α2
[v1, v2] = Split(v);   % v = v1 + v2
y = fl(α2v2 − (((x − α1v1) − α2v1) − α1v2));
```

Introducing Algorithm 1, the following algorithm transform the product PD ($P, D \in \mathbb{F}^{n\times n}$) into two matrices $G, H \in \mathbb{F}^{n\times n}$ where D is a diagonal matrix.

Algorithm 2 *Calculation of $G, H \in \mathbb{F}^{n\times n}$ such that $PD = G + H$, ($P, D \in \mathbb{F}^{n\times n}$) where D is a diagonal matrix.*

```
function [G, H] = MatProduct(P, D)
for i=1:n
    [G(:, i), H(:, i)] = TwoProductSV(D(i, i), P(:, i));
end
```

3. New Verification Method

In this section, we propose a new verification method for calculating the guaranteed error bounds of the approximate eigenvalues without directed rounding. Here, inequality for vectors means that the inequality holds for all components. Inequality for matrices means similarly. At first, we note a well-known fact that for any $n \times n$ real symmetric matrix A,

$$||A||_2 \le ||A||_\infty \tag{18}$$

holds. Therefore we are able to modify (2) as follows:

$$|\lambda_i - \tilde{\lambda_i}| \le |\tilde{\lambda_i}|||PP^T - I||_\infty + ||PDP^T - A||_\infty \tag{19}$$

From (17), it holds that

$$|PP^T - I| \le \mathrm{fl}(|PP^T - I|) + \gamma_{n+1}(|P||P^T| + I) \quad . \tag{20}$$

From this, we obtain

$$\begin{aligned} ||PP^T - I||_\infty \le\ & ||\mathrm{fl}(PP^T - I)||_\infty \\ & + \gamma_{n+1}|||P||P^T|||_\infty + \gamma_{n+1} . \end{aligned} \tag{21}$$

And from (15),

$$|P^T|e \le (1+\mathbf{u})^{n-1}\cdot \mathrm{fl}(|P^T|e) \tag{22}$$

for $e = (1, \cdots, 1)^T$ and by (13),

$$\begin{aligned} |P||P^T|e &\le (1+\mathbf{u})^{n-1}(1+\mathbf{u})^{n}\cdot \mathrm{fl}(|P|(|P^T|e)) \\ &= (1+\mathbf{u})^{2n-1}\cdot \mathrm{fl}(|P|(|P^T|e)) \quad . \end{aligned} \tag{23}$$

From this, we obtain

$$\begin{aligned} |||P||P^T|||_\infty &= |||P|(|P^T|e)||_\infty \\ &\le (1+\mathbf{u})^{2n-1}||\mathrm{fl}(|P|(|P^T|e)))||_\infty . \end{aligned} \tag{24}$$

Therefore, from (11) we obtain

$$|||P||P^T|||_\infty \le (1+\mathbf{u})^{2n-1}\mathrm{fl}(|||P|(|P^T|e)||_\infty) \quad . \tag{25}$$

Moreover, from (16), we obtain

$$||\mathrm{fl}(PP^T - I)||_\infty \le (1+\mathbf{u})^{n-1}\mathrm{fl}(||PP^T - I||_\infty) \quad . \tag{26}$$

We define

$$\begin{aligned} \alpha_1 &:= \mathrm{fl}(||PP^T - I||_\infty) \\ \alpha_2 &:= \mathrm{fl}(|||P|(|P^T|e)||_\infty) \quad . \end{aligned} \tag{27}$$

Inserting (25) and (26) into (21) and utilizing (6) and (7), we obtain

$$||PP^T - I||_\infty \le (1+\gamma_{n-1})\alpha_1 + \gamma_{n+1,3n}\alpha_2 + \gamma_{n+1} \quad . \tag{28}$$

By applying Algorithm 2, we are able to modify $PDP^T - A$ as follows:

$$\begin{aligned} PDP^T - A &= (G+H)P^T - A \\ &= GP^T - A + HP^T \quad . \end{aligned} \tag{29}$$

Therefore,

$$||PDP^T - A||_\infty = ||GP^T - A + HP^T||_\infty \le ||GP^T - A||_\infty + ||HP^T||_\infty . \quad (30)$$

Similar to (24) and utilizing (6), it holds that

$$||HP^T||_\infty \le |||H||P^T|||_\infty \le (1+\gamma_{2n-1})\mathrm{fl}(|||H|(|P^T|e)||_\infty) . \quad (31)$$

We define

$$\alpha_3 := \mathrm{fl}(|||H|(|P^T|e)||_\infty) \quad . \quad (32)$$

Inserting α_3 into (31), we obtain

$$||HP^T||_\infty \le (1+\gamma_{2n-1})\alpha_3 \quad . \quad (33)$$

On the other hand, similar to (20), it holds that

$$|GP^T - A| \le \mathrm{fl}(|GP^T - A|) + \gamma_{n+1}(|G||P^T| + |A|). \quad (34)$$

Therefore, similar to (24) and utilizing (6), (7) and (11), we obtain

$$\begin{aligned} ||GP^T - A||_\infty &\le ||\mathrm{fl}(|GP^T - A|)||_\infty \\ &\quad + \gamma_{n+1}|||G||P^T| + |A|||_\infty \\ &\le (1+\gamma_{n-1})\mathrm{fl}(||GP^T - A||_\infty) \\ &\quad + \gamma_{n+1,3n}\mathrm{fl}(|||G|(|P^T|e)||_\infty) \\ &\quad + \gamma_{n+1,2n}\mathrm{fl}(||A||_\infty) \quad . \end{aligned} \quad (35)$$

We define

$$\begin{aligned} \alpha_4 &:= \mathrm{fl}(||GP^T - A||_\infty) \\ \alpha_5 &:= \mathrm{fl}(|||G|(|P^T|e)||_\infty) \\ \alpha_6 &:= \mathrm{fl}(||A||_\infty) \quad . \end{aligned} \quad (36)$$

Inserting α_4, α_5 and α_6 into the (35), we obtain

$$||GP^T - A||_\infty \le (1+\gamma_{n-1})\alpha_4 + \gamma_{n+1,3n}\alpha_5 + \gamma_{n+1,2n}\alpha_6 \ . \quad (37)$$

Inserting (33) and (37) into the Eq, (30), we obtain

$$||PDP^T - A||_\infty \le (1+\gamma_{2n-1})\alpha_3 + (1+\gamma_{n-1})\alpha_4 + \gamma_{n+1,3n}\alpha_5 + \gamma_{n+1,2n}\alpha_6 \ . \quad (38)$$

Inserting (28) and (38) into (19) and utilizing (3) and (7), we obtain

$$\begin{aligned} |\lambda_i - \tilde{\lambda}_i| &\le (1+\mathbf{u})(\mathrm{fl}(|\tilde{\lambda}_i|\alpha_1) + \mathrm{fl}(\alpha_3+\alpha_4)) \\ &\quad + \gamma_{n-1}|\tilde{\lambda}_i|\alpha_1 + \gamma_{n+1,3n}|\tilde{\lambda}_i|\alpha_2 + \gamma_{n+1}|\tilde{\lambda}_i| \\ &\quad + \gamma_{2n-1,2n}\mathrm{fl}(\alpha_3+\alpha_4) \\ &\quad + \gamma_{n+1,3n}\alpha_5 + \gamma_{n+1,2n}\alpha_6 \quad . \end{aligned} \quad (39)$$

We define

$$\alpha_7 := \mathrm{fl}(\alpha_3 + \alpha_4) \quad . \quad (40)$$

Inserting α_7 into (39) and utilizing (7), we obtain

$$\begin{aligned} |\lambda_i - \tilde{\lambda}_i| &\le (1+\mathbf{u})^2\mathrm{fl}((|\tilde{\lambda}_i|\alpha_1) + \alpha_7) + \gamma_{n-1}|\tilde{\lambda}_i|\alpha_1 \\ &\quad + \gamma_{n+1,3n}|\tilde{\lambda}_i|\alpha_2 + \gamma_{n+1}|\tilde{\lambda}_i| + \gamma_{n+1,3n}\alpha_5 \\ &\quad + \gamma_{2n-1,2n+1}\mathrm{fl}(\alpha_6+\alpha_7) \quad . \end{aligned} \quad (41)$$

We define

$$\alpha_8 := \mathrm{fl}(\alpha_6 + \alpha_7) \quad . \quad (42)$$

Inserting α_8 into (41) and utilizing (3), (7), (8) and (9), we obtain

$$\begin{aligned} |\lambda_i - \tilde{\lambda}_i| &\le \mathrm{fl}((((|\tilde{\lambda}_i|\alpha_1) + \alpha_7) \\ &\quad + \tilde{\gamma}_{2n-1,3n+6}((|\tilde{\lambda}_i|(\alpha_1+\alpha_2+1)) \\ &\quad + (\alpha_5+\alpha_8)))/(1-4\mathbf{u})) \quad . \end{aligned} \quad (43)$$

From these, we obtain the following theorem:

Theorem 1

$$\begin{aligned} |\lambda_i - \tilde{\lambda}_i| &\le \mathrm{fl}((((|\tilde{\lambda}_i|\alpha_1) + \alpha_7) \\ &\quad + \tilde{\gamma}_{2n-1,3n+6}((|\tilde{\lambda}_i|(\alpha_1+\alpha_2+1)) \\ &\quad + \alpha_9))/(1-4\mathbf{u})) \end{aligned} \quad (44)$$

where $\alpha_1, \alpha_2, \alpha_7$ and α_9 are defined by

$$\begin{aligned} \alpha_1 &:= \mathrm{fl}(||PP^T - I||_\infty) \\ \alpha_2 &:= \mathrm{fl}(|||P|(|P^T|e)||_\infty) \\ \alpha_7 &:= \mathrm{fl}(|||H|(|P^T|e)||_\infty + ||GP^T - A||_\infty) \\ \alpha_9 &:= \mathrm{fl}(|||G|(|P^T|e)||_\infty + (||A||_\infty + \alpha_7)) \ . \end{aligned} \quad (45)$$

Utilizing Theorem 1, we are able to present an algorithm for calculating upper bounds on $|\lambda_i - \tilde{\lambda}_i|$ applying only ordinary floating point arithmetic with rounding-to-nearest.

Algorithm 3 *Calculatin of vectors $d = (\tilde{\lambda}_1, \cdots, \tilde{\lambda}_n)^T$ and $r = (r_1, \cdots, r_n)^T$ such that $|\lambda_i - \tilde{\lambda}_i| \le r_i \quad (i = 1, \cdots, n)$.*

function $[d, r] = \mathrm{VeigNear}(A)$
$[P, D] = \mathrm{eig}(A);$
$[G, H] = \mathrm{MatProduct}(P, D); \quad \% G + H = PD$
$\alpha_1 = \mathrm{fl}(||PP^T - I||_\infty);$
$\alpha_2 = \mathrm{fl}(|||P|(|P^T|e)||_\infty);$
$\alpha_7 = \mathrm{fl}(|||H|(|P^T|e)||_\infty + ||GP^T - A||_\infty);$
$\alpha_9 = \mathrm{fl}(|||G|(|P^T|e)||_\infty + (||A||_\infty + \alpha_7));$
$d = \mathrm{diag}(D); \quad \% \ d_i = D_{ii}$
$\hat{d} = \mathrm{abs}(d); \quad \% \ \hat{d}_i = |d_i|$
$p = \mathrm{fl}(\alpha_7 \cdot \mathrm{ones}(n,1)); \quad \% \ p = (\alpha_7, \cdots, \alpha_7)$
$q = \mathrm{fl}(\alpha_9 \cdot \mathrm{ones}(n,1)); \quad \% \ q = (\alpha_9, \cdots, \alpha_9)$
$\tilde{\gamma} = \mathrm{fl}(\dfrac{(2n-1)\mathbf{u}}{1-(3n+6)\mathbf{u}});$
$r = \mathrm{fl}(\dfrac{((\alpha_1\hat{d}) + p) + \tilde{\gamma}(((\alpha_1+\alpha_2+1)\hat{d}) + q)}{1-4\mathbf{u}});$

This algorithm requires $4n^3$ flops while the rounding mode controlled algorithm proposed in [1] requires $10n^3$ flops.

4. Numerical Example

In this section, we show some numerical examples. Here, we apply IEEE 754 double presicion and our computer environment is Pentium IV 2.53GHz CPU. We use MATLAB 6.5.1 for all computations. In this environment, $\mathbf{u}$ is able to be defined as $\mathbf{u} = 2^{-53}$. Let the vector r_1 and r_2 be guaranteed componentwise error bounds of computed (approximate) eigenvalues applying Algorithm 3 and the rounding mode controlled algorithm proposed in [1]. In order to show a typical order of guaranteed bounds of errors, $||r_1||_\infty$ and $||r_2||_\infty$ are calculated. And let t_{eig}, t_1 and t_2 be the calculating costs (s) when we calculate approximate eigenvalues, $||r_1||_\infty$ and $||r_2||_\infty$.

Table 1 shows t_{eig}, t_1, t_2, $||r_1||_\infty$ and $||r_2||_\infty$ for a $n \times n$ matrix

$$A = \frac{B + B^T}{2} \tag{46}$$

where B is an $n \times n$ matrix whose entries are pseudo-random numbers uniformly distributed in $[-1, 1]$.

Table 1: Comparison of the both methods for various n.

n	t_{eig}	t_1	t_2	$\|\|r_1\|\|_\infty$	$\|\|r_2\|\|_\infty$
100	0.016	0.016	0.016	1.05×10^{-12}	3.32×10^{-11}
250	0.176	0.127	0.074	3.65×10^{-12}	3.12×10^{-10}
500	1.903	0.851	0.363	1.12×10^{-11}	1.71×10^{-9}
1000	14.38	5.570	1.914	2.68×10^{-11}	9.45×10^{-9}
1500	34.89	13.14	4.102	3.29×10^{-11}	1.94×10^{-8}
2000	107.8	39.55	12.06	6.08×10^{-11}	5.31×10^{-8}

Let cond(A) be the condition number of A. The Table 2 shows cond(A), t_{eig}, t_1, t_2, $||r_1||_\infty$ and $||r_2||_\infty$ for a 1000×1000 matrix $A = B^T B$ where B is an 1000×1000 matrix whose entries are pseudo-random numbers obtained by `randsvd` utilizing Higham's test matrices [8].

Table 2: Comparison of the both methods for various cond(A).

cond(A)	t_{eig}	t_1	t_2	$\|\|r_1\|\|_\infty$	$\|\|r_2\|\|_\infty$
1.0×10^{0}	7.406	5.438	1.906	5.35×10^{-13}	2.88×10^{-10}
1.0×10^{2}	12.72	5.531	1.922	4.46×10^{-13}	1.82×10^{-10}
1.0×10^{4}	10.69	5.594	1.875	3.81×10^{-13}	1.67×10^{-10}
1.0×10^{6}	9.656	5.516	1.906	3.43×10^{-13}	1.60×10^{-10}
1.0×10^{8}	8.688	5.546	1.875	3.26×10^{-13}	1.56×10^{-10}
1.0×10^{10}	7.969	5.516	1.890	3.09×10^{-13}	1.55×10^{-10}
1.0×10^{12}	6.922	5.563	1.937	2.84×10^{-13}	1.54×10^{-10}
1.0×10^{14}	6.547	5.578	1.891	2.84×10^{-13}	1.54×10^{-10}

By Table1, we are able to confirm that t_2 is smaller than t_1 and much smaller than t_{eig}. Therefore, Algorithm 3 is faster than the rounding mode controlled algorithm and much faster than the computation of the approximate eigenvalues. Moreover, $||r_2||_\infty$ is larger than $||r_1||_\infty$ but comparable. And by Table 2, we are able to confirm that $||r_1||_\infty$, $||r_2||_\infty$, t_1 and t_2 do not depend on the condition number.

5. Conclusion

In this paper, a new verification method was proposed for calculating the guaranteed error bounds of the approximate eigenvalues of real symmetric matrix without directed rounding. Finally, some numerical examples were implemented to show the property of the new method.

Acknowledgments

This work is partially supported by 21st Century COE Program (Productive ICT Academia Program, Waseda University) form the Ministry of Education, Science, Sports and Culture, Japan.

References

[1] Shin'ichi Oishi. , "Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled," *Linear Algebra and its Applications*, 324, 133-146, 2001

[2] G. W. Stewart, J. Sun. , "Matrix Perturbation Theory," *Academic Press*, New York, p.203.,1990

[3] G. Golub, C. Van Loan. , "Matrix Computations," *Johns Hopkins University Press*, Baltimore, MD, 1983

[4] "ANSI / IEEE 754-1985, Standard for Binary Floating-Point Arithmetic", 1985.

[5] Takeshi Ogita, S.M. Rump, Shin'ichi Oishi, "Verified Solutions of Linear Systems without Directed Rounding,"*SCAN2004*,Fukuoka, Japan, Oct. 2004.

[6] T. J. Dekker , "A Floating-Point Technique for Extending the Available Precision,"*Numerische Mathematik*, 18:224-242,1971 .

[7] "MATLAB User's Guide,Version 6"The Math Works Inc., 2000.

[8] N. J. Higham, "Accuracy and Stability of Numerical Algorithms, 2nd ed. ,"*SIAM Publications*, Philadelphia, 2002.