

Accuracy and Stability of Numerical Algorithms

Nick Higham
School of Mathematics
The University of Manchester

`higham@ma.man.ac.uk`
`http://www.ma.man.ac.uk/~higham/`

**3rd Many-core and Reconfigurable
Supercomputing Network Workshop**
Queen's University, Belfast, January 15-16, 2009

Floating Point Number System

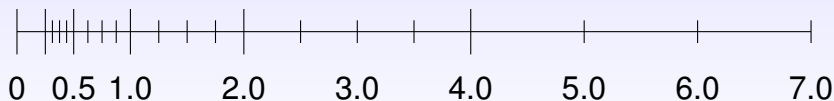
Floating point number system $F \subset \mathbb{R}$:

$$y = \pm \beta^e \times .d_1 d_2 \dots d_t, \quad 0 \leq d_i \leq \beta - 1.$$

- *Base* β ,
- *precision* t ,
- *exponent range* $e_{\min} \leq e \leq e_{\max}$.

Floating point numbers are **not** equally spaced.

If $\beta = 2$, $t = 3$, $e_{\min} = -1$, and $e_{\max} = 3$:



Relative Error

If $\hat{x} \approx x \in \mathbb{R}^n$ the **relative error** is

$$\frac{\|x - \hat{x}\|}{\|x\|}.$$

The **absolute error** $\|x - \hat{x}\|$ is scale dependent.

Common error not to normalize errors and residuals.

Rounding

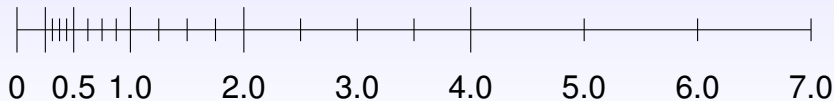
For $x \in \mathbb{R}$, $fl(x)$ is an element of F nearest to x , and the transformation $x \rightarrow fl(x)$ is called **rounding**.

Theorem

If $x \in \mathbb{R}$ lies in the range of F then

$$fl(x) = x(1 + \delta), \quad |\delta| \leq u := \frac{1}{2}\beta^{1-t}.$$

u is the **unit roundoff**, or machine precision.



IEEE Floating Point Arithmetic

IEEE Standard 754 (1985); $\beta = 2$.

- Arithmetic ops (+, -, *, /, $\sqrt{\quad}$) performed *as if* first calculated to infinite precision, then rounded.
- Default: round to nearest, round to even in case of tie.

Type	Size	Range	$u = 2^{-t}$
single	32 bits	$10^{\pm 38}$	$2^{-24} \approx 6.0 \times 10^{-8}$
double	64 bits	$10^{\pm 308}$	$2^{-53} \approx 1.1 \times 10^{-16}$

Model of Arithmetic

$$fl(x \text{ op } y) = (x \text{ op } y)(1 + \delta), \quad |\delta| \leq u, \quad \text{op} = +, -, *, /.$$

Precision versus Accuracy

$$\begin{aligned} fl(abc) &= ab(1 + \delta_1) \cdot c(1 + \delta_2) & |\delta_i| \leq u, \\ &= abc(1 + \delta_1)(1 + \delta_2) \\ &\approx abc(1 + \delta_1 + \delta_2). \end{aligned}$$

- Precision = u .
- Accuracy $\approx 2u$.

Precision versus Accuracy

$$\begin{aligned} fl(abc) &= ab(1 + \delta_1) \cdot c(1 + \delta_2) & |\delta_i| \leq u, \\ &= abc(1 + \delta_1)(1 + \delta_2) \\ &\approx abc(1 + \delta_1 + \delta_2). \end{aligned}$$

- Precision = u .
- Accuracy $\approx 2u$.

Accuracy is not limited by precision



Exceptional Results

IEEE arithmetic is **closed**: every operation produces a result. Default results:

Exception type	Default result
Invalid operation	NaN (Not a Number)
Overflow	$\pm\infty$
Divide by zero	$\pm\infty$
Underflow	Subnormal numbers
Inexact	Correctly rounded result

NaN is generated by operations such as $0/0$, $0 \times \infty$, ∞/∞ , $(+\infty) + (-\infty)$ and $\sqrt{-1}$.

Infinity symbol satisfies $\infty + \infty = \infty$, $(-1) \times \infty = -\infty$ and $(\text{finite})/\infty = 0$.

Advantages of IEEE—1

Problem: find biggest element of an array $a(1:n)$.

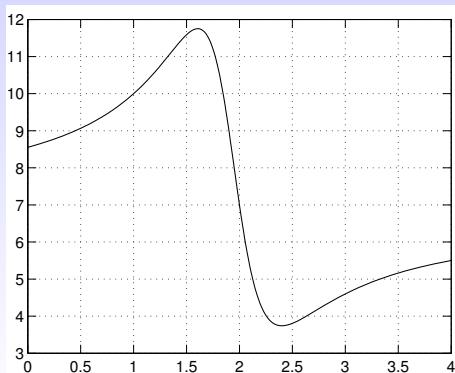
```
1 max = -inf
2 for j = 1:n
3     if a_j > max
4         max = a_j
5     end
6 end
```

- Unknown or missing elements of the array a could be assigned NaNs: $a(j) = \text{NaN}$.
- A NaN compares as unordered with everything.

Advantages of IEEE—2

Rational function

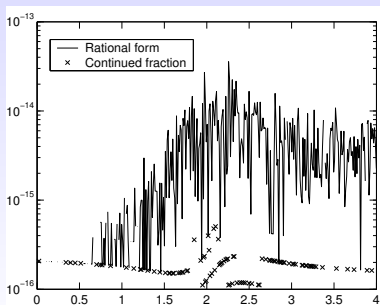
$$r(x) = \frac{(((7x - 101)x + 540)x - 1204)x + 958}{(((x - 14)x + 72)x - 151)x + 112}$$



Cont. Frac. versus Rational

$$r(x) = 7 - \frac{3}{x - 2 - \frac{1}{x - 7 + \frac{10}{x - 2 - \frac{2}{x - 3}}}}$$

Division by zero at $x = 1, 2, 3, 4$, but r evaluates correctly in IEEE arithmetic!



GPU Floating Point

E.g., Nvidia GeForce 8800 (G80), SSE, IBM Altivec, Cell SPE.

- May not support
 - double precision (only single),
 - all rounding modes,
 - NaN,
 - subnormal numbers (gradual underflow).
- Transcendental functions may not provide full accuracy.
- Square root, division may be software only.

Fixed Point Arithmetic

- Little in the numerical analysis literature since Wilkinson's 1963 book .
- No standards.
- Most of the same issues (see below) apply.

Cancellation Example

$f(x) = (1 - \cos x)/x^2 \Rightarrow 0 \leq f(x) < 1/2$ for all $x \neq 0$.
With $x = 1.2 \times 10^{-5}$, $\cos x$ rounded to 10 sig figs is

$$c = 0.9999\ 9999\ 99,$$

so

$$1 - c = 0.0000\ 0000\ 01.$$

Then $(1 - c)/x^2 = 10^{-10}/1.44 \times 10^{-10} = 0.6944\dots!$

Yet the subtraction $1 - c$ is **exact**.

Easy to rewrite $f(x)$ to avoid cancellation:

$$f(x) = \frac{1}{2} \left(\frac{\sin(x/2)}{x/2} \right)^2.$$

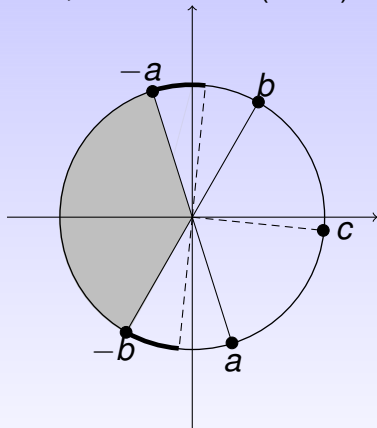
Cancellation

Cancellation **brings earlier errors into prominence** but is not *always* a bad thing.

- Numbers being subtracted may be error free.
- Cancellation may be a symptom of intrinsic ill conditioning of problem.

Midpoint of Arc

Guo, H & Tisseur (2008):



- **Problem:** Find midpoint c of an arc (a, b) .
- $z = x + iy$ with $x^2 + y^2 = 1$.
- Obvious formula $c = (a + b)/|a + b|$ is unstable when $a \approx -b$.
- **Solution:** If $a = e^{i\theta_1}$, $b = e^{i\theta_2}$ then $c = e^{i(\theta_1 + \theta_2)/2}$.

Subtraction

Theorem (Sterbenz)

Let x and y be floating point numbers with $y/2 \leq x \leq 2y$. Then $x - y$ is computed exactly (assuming $x - y$ does not underflow).

Area of a Triangle

Heron's formula for area A of triangle with sides of length a , b , and c :

$$A = \sqrt{s(s-a)(s-b)(s-c)}, \quad s = (a+b+c)/2.$$

Inaccurate for needle-shaped triangles, e.g., $a \approx b+c$.

Kahan (1983): let $a \geq b \geq c$, then evaluate $4A$ as

$$\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}.$$

Gives an accurate answer for all data.

Examples

In IEEE double:

<i>a</i>	10	1	1
<i>b</i>	11	1	$\sin(\text{eps})$
<i>c</i>	12	$1\text{e-}13$	$\cos(\text{eps})$
Exact	$5.152123\text{e+}1$	$5.000000\text{e-}14$	$1.110223\text{e-}16$
Heron	$5.152123\text{e+}1$	$4.996004\text{e-}14$	$0.000000\text{e+}0$
Kahan	$5.152123\text{e+}1$	$5.000000\text{e-}14$	$1.110223\text{e-}16$

Aberrant Arithmetics

—those that do not conform to the IEEE standard.

- Lack of guard digit is dangerous.

E.g., Kahan's triangle formula no longer works.

- Rounding may be biased.

Cray Y-MP subtraction produced biased results:

$fl(x - y)$ too big if $x > y > 0$. Caused large errors for Carter (1991, NASA Ames Lab.) using Cholesky factorization on Cray Y-MP to solve linear system of order 16146.

Fused Multiply-Add Instruction

Intel IA-64 architecture has a fused multiply-add instruction with just one rounding error:

$$fl(x + y * z) = (x + y * z)(1 + \delta), \quad |\delta| \leq u.$$

With an FMA:

- Inner product $x^T y$ can be computed with half the rounding errors.
- The algorithm

$$1 \quad w = b * c$$

$$2 \quad e = w - b * c$$

$$3 \quad x = (a * d - w) + e$$

computes $x = \det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$ with high relative accuracy (Kahan).

Fused Multiply-Add Instruction (cont.)

But

- What does $a*d + c*b$ mean?
- The product

$$(x + iy)^*(x + iy) = x^2 + y^2 + i(xy - yx)$$

may evaluate to non-real with an FMA.

- $b^2 - 4ac$ can evaluate negative even when $b^2 \geq 4ac$.

Three Misconceptions of Floating Point Arithmetic

1. An Innocuous Calculation?

For any $x \geq 0$ consider

```
1 for  $i = 1:60$ 
2    $x = \sqrt{x}$ 
3 end
4 for  $i = 1:60$ 
5    $x = x^2$ 
6 end
```

For any x the 12-digit HP 48G calculator computes, in place of $f(x) = x$,

$$\hat{f}(x) = \begin{cases} 0, & 0 \leq x < 1, \\ 1, & x \geq 1. \end{cases}$$

Explanation

The positive numbers x representable on the HP 48G satisfy $10^{-499} \leq x \leq 10^{500}$.

Defining $r(x) = x^{1/2^{60}}$, for any machine number $x \geq 1$,

$$\begin{aligned} 1 &\leq r(x) < r(10^{500}) = 10^{500/2^{60}} \\ &= e^{500 \cdot 2^{-60} \cdot \log 10} < e^{10^{-15}} \\ &= 1 + 10^{-15} + \frac{1}{2} \cdot 10^{-30} + \dots, \end{aligned}$$

which rounds to 1. Similar argument for $0 < x < 1$.

Myth 1

A short computation free from cancellation, underflow and overflow must be accurate.

2. Increasing the Precision

$y = e^{\pi\sqrt{163}}$ evaluated at t digit precision:

t	y
20	262537412640768744.00
25	262537412640768744.0000000
30	262537412640768743.999999999999

Is the last digit before the decimal point 4?

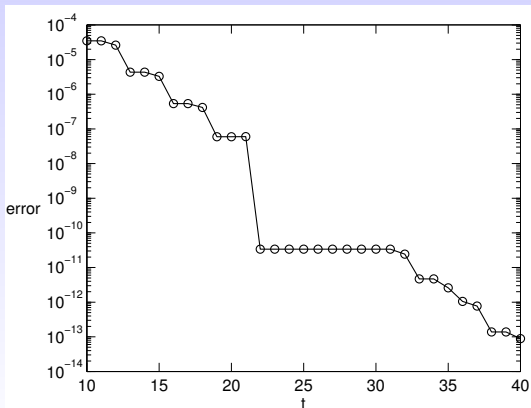
t	y
35	262537412640768743.99999999999925007
40	262537412640768743.9999999999992500725972

So no, it's 3!

Another Example

Consider the evaluation in precision $u = 2^{-t}$ of

$$y = x + a \sin(bx), \quad x = 1/7, \quad a = 10^{-8}, \quad b = 2^{24}.$$



Myth 2

Increasing the precision at which a computation is performed increases the accuracy of the answer.

3. Cancellation of Rounding Errors

$$f(x) = \frac{e^x - 1}{x} = \sum_{i=0}^{\infty} \frac{x^i}{(i+1)!}$$

% Algorithm 1.

```
1 if x = 0
2   f = 1
3 else
4   f = (ex - 1)/x
5 end
```

% Algorithm 2.

```
1 y = ex
2 if y = 1
3   f = 1
4 else
5   f = (y - 1)/log y
6 end
```

Some Results: $f(x) = (e^x - 1)/x$

x	Algorithm 1	Algorithm 2
10^{-5}	1.0000050000 <u>06965</u>	1.000005000016667
10^{-6}	1.000000 <u>499962184</u>	1.000000500000167
10^{-7}	1.0000000 <u>49433680</u>	1.000000050000002
10^{-8}	<u>0.999999993922529</u>	1.000000005000000
10^{-9}	1.0000000 <u>82740371</u>	1.000000000500000
10^{-10}	1.0000000 <u>82740371</u>	1.000000000050000
10^{-11}	1.0000000 <u>82740371</u>	1.000000000005000
10^{-12}	1.0000 <u>88900582341</u>	1.000000000000500
10^{-13}	<u>0.999200722162641</u>	1.000000000000050
10^{-14}	<u>0.999200722162641</u>	1.000000000000005
10^{-15}	<u>1.110223024625156</u>	1.000000000000000 <u>0</u>
10^{-16}	<u>0</u>	1

A Closer Look

Consider $x = 9 \times 10^{-8}$ and $u \approx 6 \times 10^{-8}$, for which $f(x) = 1.00000005$ to the sig digits shown.

Algorithm 1 gives

$$fl\left(\frac{e^x - 1}{x}\right) \equiv fl\left(\frac{1.19209290 \times 10^{-7}}{9.00000000 \times 10^{-8}}\right) = 1.32454766.$$

Algorithm 2:

$$fl\left(\frac{e^x - 1}{\log e^x}\right) \equiv fl\left(\frac{1.19209290 \times 10^{-7}}{1.19209282 \times 10^{-7}}\right) = 1.00000006.$$

Algorithm 2 in exact arithmetic would give

$$\frac{e^x - 1}{\log e^x} \equiv \frac{9.00000041 \times 10^{-8}}{9.00000001 \times 10^{-8}} = 1.00000005.$$

Divided Difference Connection

$$f(x) = \frac{e^x - 1}{x} = \frac{e^x - e^0}{x - 0}.$$

More generally:

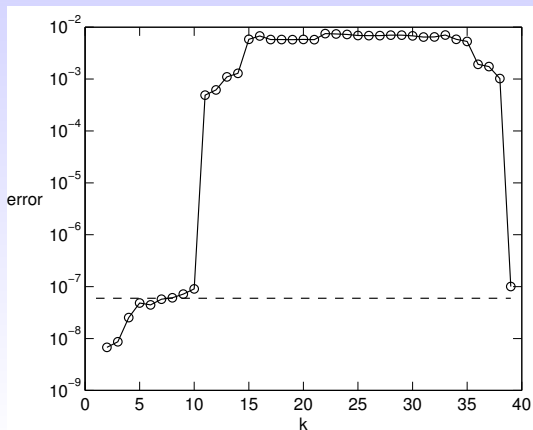
$$\begin{aligned} \frac{e^{\lambda_2} - e^{\lambda_1}}{\lambda_2 - \lambda_1} &= e^{(\lambda_1 + \lambda_2)/2} \frac{e^{(\lambda_2 - \lambda_1)/2} - e^{(\lambda_1 - \lambda_2)/2}}{\lambda_2 - \lambda_1} \\ &= e^{(\lambda_1 + \lambda_2)/2} \frac{\sinh((\lambda_2 - \lambda_1)/2)}{(\lambda_2 - \lambda_1)/2}. \end{aligned}$$

Givens QR Factorization

$$G_p G_{p-1} \dots G_1 A = R.$$

$A \in \mathbb{R}^{10 \times 6}$, single precision ($u \approx 6 \times 10^{-8}$).

$\|A\|_2 = 1$, $\kappa_2(A) \approx 24$. Look at $\|A_k - \hat{A}_k\|_2 / \|A\|_2$:



Myth 3

The final computed answer from an algorithm cannot be more accurate than any of the intermediate quantities, that is, errors cannot cancel.

Vector 2-Norm

Classic example: $\|x\|_2 = (\sum_i |x_i|^2)^{1/2}$.

For about half of all machine numbers x , x^2 either **underflows** or **overflows**!

Overflow is avoided by the following algorithm:

```
1  $t = \|x\|_\infty$ 
2  $s = 0$ 
3 for  $i = 1:n$ 
4    $s = s + (x(i)/t)^2$ 
5 end
6  $\|x\|_2 = t\sqrt{s}$ 
```

- Requires two passes over the data, so is slow.

One-Pass 2-Norm Algorithm

Level-1 BLAS `xNRM2` distrib. with LAPACK is *one-pass* alg of Hammarling avoiding overflow:

```
1  t = 0
2  s = 1
3  for i = 1:n
4      if xi ≈ 0
5          if |xi| > t
6              s = 1 + s(t/xi)2
7              t = |xi|
8          else
9              s = s + (xi/t)2
10         end
11     end
12 end
13 ||x||2 = t√s
```

Complex Division

$$\frac{a + ib}{c + id} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2},$$

overflows when c or d exceeds $\sqrt{\text{realmax}}$.

Cray and NEC machines both implemented complex division in this way.

Smith '67 suggests how to avoid overflow: if $|c| \geq |d|$ use

$$\frac{a + ib}{c + id} = \frac{a + b(dc^{-1})}{c + d(dc^{-1})} + i \frac{b - a(dc^{-1})}{c + d(dc^{-1})}.$$

If $|d| \geq |c|$ use similar formula involving d^{-1} .

- ▶ Stewart '85 shows how to reduce error due to underflow.
- ▶ Priest '04 how to efficiently avoid underflow and overflow.

Ariane 5 Rocket Failure

Paris, 19 July 1996
ARIANE 5
Flight 501 Failure
Report by the Inquiry Board

...

The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error.



Old Mutual's new chief weighs rescue options

JUDGING by the empty state of his spacious South African office, it is quite clear that Julian Roberts has yet to settle into his role as the new chief executive of Old Mutual.

While his secretary bustles around, tidying away his few possessions – a 5p piece and a penny coin left lying on his desk – the four books on his vacant shelves stand out. The titles *Blown to Bits* and *On the Brink of Failure* could almost sum up the state of the blue-chip company Mr Roberts has just taken over. Old Mutual was the worst-performing European

PROFILE

Julian Roberts

*Chief executive,
Old Mutual*

The economic turmoil revealed cracks in Old Mutual's model when it emerged that its \$2.8bn (£1.9bn) variable annuity business in the US could not meet guarantees due to adverse movements in the Asian markets. It has been forced to inject

going to be immune. South Africa lags the rest of the world by six months to a year.”

Political tensions are also playing on his mind. Old Mutual is listed not only in the UK and Johannesburg but also on the Zimbabwe Stock Exchange. **Due to technical difficulties of transferring a figure with so many noughts on the end of it, Old Mutual struggled to pay shareholders an interim dividend of Z\$453 trillion per share – which in November equated to just 2.45p.**

“It is absolutely tragic. We have a significant business with a large

Extended and Mixed Precision BLAS

- Part of new 2002 standard developed by BLAS Technical Forum.
- Use extended precision internally: a precision at least 1.5 times as accurate as double precision and wider than 80 bits.
- Input and output arguments remain single or double.
- Provide extended/mixed precision counterparts of selected level 1, 2, and 3 BLAS routines. Extra input argument specifies precision of internal computations.
- Reference implementation employs the double-double format, giving an extended precision of about 106 bits.

SECOND EDITION

Nicholas J. Higham

Accuracy and Stability of Numerical Algorithms

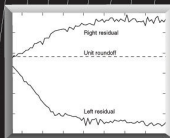
Nicholas J. Higham

siam

OFBO

Accuracy and Stability of Numerical Algorithms

SECOND EDITION



siam

Accuracy and Stability of Numerical Algorithms gives a thorough, up-to-date treatment of the behavior of numerical algorithms in finite precision arithmetic. It combines algorithmic, arithmetic, perturbation theory, and rounding error analysis, all enhanced by historical perspective and informative anecdotes.

This second edition appends and updates the coverage of the first edition (1996) and includes numerous improvements to the original material. Two new chapters treat consensus algorithms, systems and slow systems systems, and nonlinear systems and Newton's method. An expanded treatment of Gaussian elimination encompasses rank pivoting and additional error bounds. Other new topics include rank-revealing LU factorizations, weighted and constrained least squares problems, and the fast multiple multiplications based on some modern computer architectures.

Although not designed specifically as a textbook, this new edition is a valuable reference for an advanced course. The title has been changed to reflect the new content and to distinguish it from the first which draws examples, historical perspective, statements of results, and exercises.

From reviews of the first edition:

"The author writes on the accuracy and stability of numerical algorithms in quite a length and it is worth the while to the library of any statistician heavily involved in computing."
—Robert S. Strimling, *Journal of the American Statistical Association*, March 1995.

"This text may become the new Bible about accuracy and stability for the solution of linear equations. It covers 100 pages carefully selected, investigated and written. One will find that this book is a very valuable and comprehensive reference for research in numerical linear algebra, software usage and development, and for numerical linear algebra courses."
—Jo Kiefer, *Zentralblatt für Mathematik*, Band 84759.

"Nick Higham has assembled an enormous amount of important and useful material in a coherent, readable form. His book belongs on the shelf of anyone who has ever been a casual student in teaching, error and matrix computation."
—Cliff Smith, *SIAM News*, March 1997.

Nicholas J. Higham is Richardson Professor of Applied Mathematics at the University of Manchester, England. He is the author of more than 80 publications and is a member of the editorial boards of *Foundations of Computational Mathematics*, the *SIAM Journal of Numerical Analysis*, *Linear Algebra and its Applications*, and the *SIAM Journal on Matrix Analysis and Applications*.



For more information about SIAM books, journals,

memberships, or activities, contact:

siam

Society for Industrial and Applied Mathematics
3605 University City Science Center
Philadelphia, PA 19104-1228
215-382-1800 • Fax 215-382-7999
siam@siam.org • www.siam.org



8K070000

Time to L^AT_EX

DX2-33	7.5 mins	Pentium 2.8Ghz	5 secs
Pentium 120Mhz	1.3 mins	Athlon X2 4400	4 secs
Pentium 500Mhz	20 secs	Pentium E6850	4 secs
Pentium 1Ghz	10 secs		

Developing an HPC/NA Roadmap

Anne Trefethen, OeRC, University of Oxford

Peter Coveney, University College London

Nick Higham, University of Manchester


Iain Duff, STFC, Rutherford-Appleton Laboratory

Reports from first two workshops available at


<http://www.oerc.ox.ac.uk/research/hpc-na>

Workshop 3: January 26–27, 2009, Royal Society, London.

References I

 Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard.

Int. J. High Performance Applications and Supercomputing, 16(1), 2002.

 C.-H. Guo, N. J. Higham, and F. Tisseur.
An improved arc algorithm for detecting definite Hermitian pairs.
MIMS EPrint 2008.115, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, Nov. 2008.
22 pp.

References II



N. J. Higham.

Accuracy and Stability of Numerical Algorithms.

Society for Industrial and Applied Mathematics,
Philadelphia, PA, USA, second edition, 2002.

ISBN 0-89871-521-0.

xxx+680 pp.




*IEEE Standard for Binary Floating-Point Arithmetic,
ANSI/IEEE Standard 754-1985.*

Institute of Electrical and Electronics Engineers, New
York, 1985.

Reprinted in SIGPLAN Notices, 22(2):9–25, 1987.

References III

-  X. S. Li, J. W. Demmel, D. H. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, A. Kapur, M. C. Martin, T. Tung, and D. J. Yoo.

Design, implementation and testing of Extended and Mixed Precision BLAS.

Technical Report CS-00-451, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, Oct. 2000.

61 pp.

LAPACK Working Note 149.

-  D. M. Priest.

Efficient scaling for complex division.

ACM Trans. Math. Software, 30(4):389–401, 2004.

References IV



R. L. Smith.

Algorithm 116: Complex division.

Comm. ACM, 5(8):435, 1962.



G. W. Stewart.

A note on complex division.

ACM Trans. Math. Software, 11(3):238–241, 1985.

References V



J. H. Wilkinson.

Rounding Errors in Algebraic Processes.

Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963.

ISBN 0-486-67999-3.

vi+161 pp.

Also published by Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New York, 1994.