Rounding Errors in Complex Floating-Point Multiplication

Colin Percival

cperciva@irmacs.sfu.ca

IRMACS, Simon Fraser University

Review of Floating-Point Arithmetic

- Floating-point values are expressed as a sign, significand, and exponent, e.g.,

$$(-1)^s \cdot \beta^{e-B} \cdot m$$

where $\{s, e, m\} \subset \mathbb{N}$, $0 < e < E$, $\beta^{t-1} \leq m < \beta^t$, and β, t, B, E are parameters of the floating-point system being used.

Review of Floating-Point Arithmetic

- Floating-point values are expressed as a sign, significand, and exponent, e.g.,

$$(-1)^s \cdot \beta^{e-B} \cdot m$$

  where $\{s, e, m\} \subset \mathbb{N}$, $0 < e < E$, $\beta^{t-1} \leq m < \beta^t$, and β, t, B, E are parameters of the floating-point system being used.

- As a special case, $e = 0$ and $m = \beta^{t-1}$ represents ± 0.

Review of Floating-Point Arithmetic

- Floating-point values are expressed as a sign, significand, and exponent, e.g.,

$$(-1)^s \cdot \beta^{e-B} \cdot m$$

  where $\{s, e, m\} \subset \mathbb{N}$, $0 < e < E$, $\beta^{t-1} \leq m < \beta^t$, and β, t, B, E are parameters of the floating-point system being used.
- As a special case, $e = 0$ and $m = \beta^{t-1}$ represents ± 0.
- In IEEE 754 "double precision" arithmetic, $\beta = 2$, $t = 53$, $B = 1075$ and $E = 2047$.

Review of Floating-Point Arithmetic

- Floating-point values are expressed as a sign, significand, and exponent, e.g.,

$$(-1)^s \cdot \beta^{e-B} \cdot m$$

  where $\{s, e, m\} \subset \mathbb{N}$, $0 < e < E$, $\beta^{t-1} \leq m < \beta^t$, and β, t, B, E are parameters of the floating-point system being used.
- As a special case, $e = 0$ and $m = \beta^{t-1}$ represents ± 0.
- In IEEE 754 “double precision” arithmetic, $\beta = 2$, $t = 53$, $B = 1075$ and $E = 2047$.
- There are also denormals, infinities, and NaNs, but in numerical code they usually never occur.

Terminology

- Denote by $\oplus$, $\ominus$, and $\otimes$ the results of rounded floating-point addition, subtraction, and multiplication, and define the "unit in the last place" $\mathrm{ulp}(x)$ for $x \neq 0$ as the (unique) power of β such that

$$\beta^{t-1} \leq |x| / \mathrm{ulp}(x) < \beta^t$$

and $\mathrm{ulp}(0) = 0$.

Terminology

- Denote by $\oplus$, $\ominus$, and $\otimes$ the results of rounded floating-point addition, subtraction, and multiplication, and define the "unit in the last place" $\mathrm{ulp}(x)$ for $x \neq 0$ as the (unique) power of β such that

$$\beta^{t-1} \leq |x| \,/ \mathrm{ulp}(x) < \beta^{t}$$

and $\mathrm{ulp}(0) = 0$.

- Note that for $x \neq 0$,

$$\mathrm{ulp}((-1)^s \cdot \beta^{e-B} \cdot m) = \beta^{e-B}$$

and $\mathrm{ulp}(x) \leq x \cdot \beta^{1-t}$.

Terminology

- Denote by $\oplus$, $\ominus$, and $\otimes$ the results of rounded floating-point addition, subtraction, and multiplication, and define the “unit in the last place” $\mathrm{ulp}(x)$ for $x \neq 0$ as the (unique) power of β such that

$$\beta^{t-1} \leq |x| / \mathrm{ulp}(x) < \beta^{t}$$

  and $\mathrm{ulp}(0) = 0$.

- Note that for $x \neq 0$,

$$\mathrm{ulp}((-1)^{s} \cdot \beta^{e-B} \cdot m) = \beta^{e-B}$$

  and $\mathrm{ulp}(x) \leq x \cdot \beta^{1-t}$.

- Also define $\epsilon = \frac{1}{2}\mathrm{ulp}(1) = \frac{1}{2}\beta^{1-t}$.

Floating-Point Rounding Errors

- IEEE 754 requires that arithmetic operations produce results which are *exactly rounded*, i.e., the same as if the values were computed to infinite precision prior to rounding.

Floating-Point Rounding Errors

- IEEE 754 requires that arithmetic operations produce results which are *exactly rounded*, i.e., the same as if the values were computed to infinite precision prior to rounding.
- In round-to-nearest mode on IEEE 754 systems,

$$|(x+y)-(x\oplus y)| \leq \frac{1}{2}\mathrm{ulp}(x+y)$$

$$|(x-y)-(x\ominus y)| \leq \frac{1}{2}\mathrm{ulp}(x-y)$$

$$|(xy)-(x\otimes y)| \leq \frac{1}{2}\mathrm{ulp}(xy)$$

Floating-Point Rounding Errors

- IEEE 754 requires that arithmetic operations produce results which are *exactly rounded*, i.e., the same as if the values were computed to infinite precision prior to rounding.
- In round-to-nearest mode on IEEE 754 systems,

$$|(x+y)-(x\oplus y)| \le \frac{1}{2}\mathrm{ulp}(x+y) < \epsilon(x+y)$$

$$|(x-y)-(x\ominus y)| \le \frac{1}{2}\mathrm{ulp}(x-y) < \epsilon(x-y)$$

$$|(xy)-(x\otimes y)| \le \frac{1}{2}\mathrm{ulp}(xy) < \epsilon(xy)$$

Complex Rounding Errors

- For complex $z_0 = a_0 + ib_0$, $z_1 = a_1 + ib_1$, if we compute $z_2 = a_2 + ib_2 = (a_0 \oplus a_1) + i(b_0 \oplus b_1)$, then

$$\left|(z_0 + z_1) - z_2\right| = \sqrt{((a_0 + a_1) - a_2)^2 + ((b_0 + b_1) - b_2)^2}$$
$$< \sqrt{(\epsilon\,|a_0 + a_1|)^2 + (\epsilon\,|b_0 + b_1|)^2} = \epsilon\,|z_0 + z_1|$$

Complex Rounding Errors

- For complex $z_0 = a_0 + ib_0$, $z_1 = a_1 + ib_1$, if we compute $z_2 = a_2 + ib_2 = (a_0 \oplus a_1) + i(b_0 \oplus b_1)$, then

$$
\begin{aligned}
\left|(z_0 + z_1) - z_2\right| &= \sqrt{((a_0 + a_1) - a_2)^2 + ((b_0 + b_1) - b_2)^2} \\
&< \sqrt{(\epsilon \left|a_0 + a_1\right|)^2 + (\epsilon \left|b_0 + b_1\right|)^2} = \epsilon \left|z_0 + z_1\right|
\end{aligned}
$$

- Problem: If we compute

$$
\begin{aligned}
x_2 &= (a_0 \otimes a_1) \ominus (b_0 \otimes b_1) \\
y_2 &= (a_0 \otimes b_1) \oplus (b_0 \otimes a_1),
\end{aligned}
$$

what is the smallest α such that $\left|z_0 z_1 - z_2\right| < \epsilon\alpha \left|z_0 z_1\right|$?

Motivation

- The Fast Fourier Transform makes large polynomial (and integer) arithmetic practical.

Motivation

- The Fast Fourier Transform makes large polynomial (and integer) arithmetic practical.
  - Finite Field arithmetic is slow on most CPUs.

Motivation

- The Fast Fourier Transform makes large polynomial (and integer) arithmetic practical.
  - Finite Field arithmetic is slow on most CPUs.
  - Floating-Point arithmetic has rounding errors.

Motivation

- The Fast Fourier Transform makes large polynomial (and integer) arithmetic practical.
  - Finite Field arithmetic is slow on most CPUs.
  - Floating-Point arithmetic has rounding errors.
- Theorem [Percival, 2002]: The FFT allows accurate computation of the cyclic convolution $z = x * y$ of two vectors of length $N = 2^n$ of Gaussian integers if

$$|x| \cdot |y| \cdot ((1+\epsilon)^{3n}(1+\epsilon\alpha)^{3n+1}(1+\beta)^{3n} - 1) < \frac{1}{2}$$

where $\epsilon\alpha$ is the maximum relative error of complex multiplication, and β is the maximum error in the precomputed complex roots of unity used.

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].
- We can take $\alpha = \sqrt{16/3}$ [Olver, 1986].

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].
- We can take $\alpha = \sqrt{16/3}$ [Olver, 1986].
- We can take $\alpha = \sqrt{5}$ [Percival, 2002].

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].
- We can take $\alpha = \sqrt{16/3}$ [Olver, 1986].
- We can take $\alpha = \sqrt{5}$ [Percival, 2002].
  - Conjectured based on comparing the results of single-precision and double-precision complex multiplication of several million randomly chosen inputs.

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].
- We can take $\alpha = \sqrt{16/3}$ [Olver, 1986].
- We can take $\alpha = \sqrt{5}$ [Percival, 2002].
  - Conjectured based on comparing the results of single-precision and double-precision complex multiplication of several million randomly chosen inputs.
  - Unfortunately the proof was wrong...

Previous Bounds

- We can take $\alpha = \sqrt{8}$ [Higham, *Accuracy and Stability of Numerical Algorithms*].
- We can take $\alpha = \sqrt{16/3}$ [Olver, 1986].
- We can take $\alpha = \sqrt{5}$ [Percival, 2002].
  - Conjectured based on comparing the results of single-precision and double-precision complex multiplication of several million randomly chosen inputs.
  - Unfortunately the proof was wrong...
  - ... and it took five years before anyone noticed!

Error Bound

Theorem 1. *[Brent, Percival, Zimmermann, 2006]*
Let $z_0 = a_0 + b_0 i$ and $z_1 = a_1 + b_1 i$, with a_0, b_0, a_1, b_1 floating-point values with t-digit base-β significands, and

$$z_2 = ((a_0 \otimes a_1) \ominus (b_0 \otimes b_1)) + ((a_0 \otimes b_1) \oplus (b_0 \otimes a_1))i.$$

Providing that no overflow or underflow occur, no denormal values are produced, arithmetic results are correctly rounded to a nearest representable value, $z_0 z_1 \neq 0$, and $\beta^t \geq 2^5$,

$$|z_0 z_1 - z_2| < \frac{1}{2}\beta^{1-t} |z_0 z_1| = \epsilon\sqrt{5} |z_0 z_1| .$$

Equivalent Inputs

Without loss of generality, we can assume the greatest possible relative error occurs when

- $0 \leq a_0, b_0, a_1, b_1$, by multiplying by powers of i,

Equivalent Inputs

Without loss of generality, we can assume the greatest possible relative error occurs when

- $0 \leq a_0, b_0, a_1, b_1$, by multiplying by powers of i,
- $b_0 b_1 \leq a_0 a_1$, by taking complex congugates and multiplying z_0, z_1 by i,

Equivalent Inputs

Without loss of generality, we can assume the greatest possible relative error occurs when

- $0 \leq a_0, b_0, a_1, b_1$, by multiplying by powers of i,
- $b_0 b_1 \leq a_0 a_1$, by taking complex congugates and multiplying z_0, z_1 by i,
- $b_0 a_1 \leq a_0 b_1$, by swapping z_0 and z_1,

Equivalent Inputs

Without loss of generality, we can assume the greatest possible relative error occurs when

- $0 \leq a_0, b_0, a_1, b_1$, by multiplying by powers of i,
- $b_0 b_1 \leq a_0 a_1$, by taking complex congugates and multiplying z_0, z_1 by i,
- $b_0 a_1 \leq a_0 b_1$, by swapping z_0 and z_1,
- $\frac{1}{2} \leq a_0 < 1$, by multiplying z_0 by powers of 2, and

Equivalent Inputs

Without loss of generality, we can assume the greatest possible relative error occurs when

- $0 \le a_0, b_0, a_1, b_1$, by multiplying by powers of i,
- $b_0 b_1 \le a_0 a_1$, by taking complex congugates and multiplying z_0, z_1 by i,
- $b_0 a_1 \le a_0 b_1$, by swapping z_0 and z_1,
- $\frac{1}{2} \le a_0 < 1$, by multiplying z_0 by powers of 2, and
- $\frac{1}{2} \le a_0 a_1 < 1$, by multiplying z_1 by powers of 2,

none of which affect the resulting relative error.

Imaginary Error

- To bound the imaginary error $|\Im(z_0 z_1 - z_2)|$, we consider two cases:

Imaginary Error

- To bound the imaginary error $|\Im(z_0 z_1 - z_2)|$, we consider two cases:
  - Case I1: $\mathrm{ulp}(a_0 b_1 + b_0 a_1) < \mathrm{ulp}(a_0 \otimes b_1 + b_0 \otimes a_1)$

Imaginary Error

- To bound the imaginary error $|\Im(z_0 z_1 - z_2)|$, we consider two cases:
  - Case I1: $\mathrm{ulp}(a_0 b_1 + b_0 a_1) < \mathrm{ulp}(a_0 \otimes b_1 + b_0 \otimes a_1)$
  - Case I2: $\mathrm{ulp}(a_0 \otimes b_1 + b_0 \otimes a_1) \leq \mathrm{ulp}(a_0 b_1 + b_0 a_1)$

Imaginary Error

- To bound the imaginary error $|\Im(z_0 z_1 - z_2)|$, we consider two cases:
  - Case I1: $\mathrm{ulp}(a_0 b_1 + b_0 a_1) < \mathrm{ulp}(a_0 \otimes b_1 + b_0 \otimes a_1)$
  - Case I2: $\mathrm{ulp}(a_0 \otimes b_1 + b_0 \otimes a_1) \leq \mathrm{ulp}(a_0 b_1 + b_0 a_1)$
- In each case, we find that

$$|(a_0 \otimes b_1 + b_0 \otimes a_1) - ((a_0 \otimes b_1) \oplus (b_0 \otimes a_1))| < \epsilon \cdot (a_0 b_1 + b_0 a_1)$$

and thus

$$|\Im(z_0 z_1 - z_2)| < \epsilon \cdot (2a_0 b_1 + 2b_0 a_1).$$

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$
  - Case R2: $\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$
  - Case R2: $\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R3: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$
  - Case R2: $\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R3: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R4: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$
  - Case R2: $\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R3: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R4: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$
- Since we have assumed that $0 \leq b_0 b_1 \leq a_0 a_1$, these four cases cover all possible inputs.

Real Error

- To bound the real error $|\Re(z_0 z_1 - z_2)|$, we will consider four cases:
  - Case R1: $\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$
  - Case R2: $\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R3: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$
  - Case R4: $\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$
- Since we have assumed that $0 \leq b_0 b_1 \leq a_0 a_1$, these four cases cover all possible inputs.
- Once we have bounds on the real error for each of these cases, we can combine them with the imaginary error bound to obtain a bound on the complex error.

Case R1

$$\mathrm{ulp}(b_0 b_1) \le \mathrm{ulp}(a_0 a_1) \le \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$$

Case R1

$$\mathrm{ulp}(b_0 b_1) \le \mathrm{ulp}(a_0 a_1) \le \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$$

Note that

$$\frac{1}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \epsilon \cdot (a_0 a_1 - b_0 b_1 + \epsilon(a_0 a_1 + b_0 b_1))$$

Case R1

$$\mathrm{ulp}(b_0 b_1) \leq \mathrm{ulp}(a_0 a_1) \leq \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$$

Note that

$$\frac{1}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \epsilon \cdot (a_0 a_1 - b_0 b_1 + \epsilon(a_0 a_1 + b_0 b_1))$$

Consequently,

$$\begin{aligned} |\Re(z_0 z_1 - z_2)| &\leq \frac{1}{2}\mathrm{ulp}(b_0 b_1) + \frac{1}{2}\mathrm{ulp}(a_0 a_1) + \frac{1}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \\ &\leq \frac{1}{2}\mathrm{ulp}(b_0 b_1) + \frac{2}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \\ &\leq \epsilon \cdot (2a_0 a_1 - b_0 b_1) + 2\epsilon^2 \left|z_0 z_1\right| \end{aligned}$$

Case R2

$$\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$$

Case R2

$$\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$$

Note that $\mathrm{ulp}(x) < \mathrm{ulp}(y)$ implies $\mathrm{ulp}(x) \leq \frac{1}{2}\mathrm{ulp}(y)$, i.e.,

$$\mathrm{ulp}(b_0 b_1) \leq \frac{1}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1)$$

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \frac{1}{2}\mathrm{ulp}(a_0 a_1)$$

Case R2

$$\mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(a_0 a_1)$$

Note that $\mathrm{ulp}(x) < \mathrm{ulp}(y)$ implies $\mathrm{ulp}(x) \leq \frac{1}{2}\mathrm{ulp}(y)$, i.e.,

$$\begin{aligned} \mathrm{ulp}(b_0 b_1) &\leq \frac{1}{2}\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \\ \mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) &\leq \frac{1}{2}\mathrm{ulp}(a_0 a_1) \end{aligned}$$

Consequently,

$$\begin{aligned} |\Re(z_0 z_1 - z_2)| &\leq \left(\frac{1}{8} + \frac{1}{4} + \frac{1}{2}\right) \cdot \mathrm{ulp}(a_0 a_1) \\ &\leq \epsilon \cdot \left(\frac{7}{4} a_0 a_1\right) \end{aligned}$$

Case R3

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$$

Case R3

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$$

Note that

$$(a_0 \otimes a_1) \ominus (b_0 \otimes b_1) = a_0 \otimes a_1 - b_0 \otimes b_1$$

$$\mathrm{ulp}(b_0 b_1) \leq \frac{1}{2}\mathrm{ulp}(a_0 a_1)$$

Case R3

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) \leq \mathrm{ulp}(b_0 b_1) < \mathrm{ulp}(a_0 a_1)$$

Note that

$$(a_0 \otimes a_1) \ominus (b_0 \otimes b_1) = a_0 \otimes a_1 - b_0 \otimes b_1$$

$$\mathrm{ulp}(b_0 b_1) \leq \frac{1}{2}\mathrm{ulp}(a_0 a_1)$$

Consequently,

$$\begin{aligned} |\Re(z_0 z_1 - z_2)| &\leq \left(\frac{1}{4} + \frac{1}{2}\right) \mathrm{ulp}(a_0 a_1) \\ &\leq \epsilon \cdot \left(\frac{3}{2} a_0 a_1\right) \end{aligned}$$

Case R4

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$$

Case R4

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$$

Note that

$$(a_0 \otimes a_1) \ominus (b_0 \otimes b_1) = a_0 \otimes a_1 - b_0 \otimes b_1$$

Case R4

$$\mathrm{ulp}(a_0 \otimes a_1 - b_0 \otimes b_1) < \mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1)$$

Note that

$$(a_0 \otimes a_1) \ominus (b_0 \otimes b_1) = a_0 \otimes a_1 - b_0 \otimes b_1$$

Consequently,

$$\begin{aligned} |\Re(z_0 z_1 - z_2)| &\leq \frac{1}{2}\mathrm{ulp}(a_0 a_1) + \frac{1}{2}\mathrm{ulp}(b_0 b_1) \\ &\leq \epsilon \cdot (a_0 a_1 + b_0 b_1) \end{aligned}$$

Absolute Complex Error

- Imaginary error: $|\Im(z_0 z_1 - z_2)| < \epsilon \cdot (2a_0 b_1 + 2b_0 a_1)$

Absolute Complex Error

- Imaginary error: $|\Im(z_0 z_1 - z_2)| < \epsilon \cdot (2a_0 b_1 + 2b_0 a_1)$
- Case R1: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (2a_0 a_1 - b_0 b_1) + 2\epsilon^2 |z_0 z_1|$
  $\implies |z_0 z_1 - z_2| < \epsilon \left(\sqrt{32/7} + 2\epsilon \right) |z_0 z_1|$

Absolute Complex Error

- Imaginary error: $|\Im(z_0 z_1 - z_2)| < \epsilon \cdot (2a_0 b_1 + 2b_0 a_1)$
- Case R1: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (2a_0 a_1 - b_0 b_1) + 2\epsilon^2 |z_0 z_1|$
  $\implies |z_0 z_1 - z_2| < \epsilon \left(\sqrt{32/7} + 2\epsilon \right) |z_0 z_1|$
- Case R2: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot \left(\frac{7}{4} a_0 a_1\right)$
  $\implies |z_0 z_1 - z_2| < \epsilon \sqrt{1024/207}\, |z_0 z_1|$

Absolute Complex Error

- Imaginary error: $|\Im(z_0z_1 - z_2)| < \epsilon \cdot (2a_0b_1 + 2b_0a_1)$
- Case R1: $|\Re(z_0z_1 - z_2)| \leq \epsilon \cdot (2a_0a_1 - b_0b_1) + 2\epsilon^2 |z_0z_1|$
  $\implies |z_0z_1 - z_2| < \epsilon\left(\sqrt{32/7} + 2\epsilon\right)|z_0z_1|$
- Case R2: $|\Re(z_0z_1 - z_2)| \leq \epsilon \cdot (\frac{7}{4}a_0a_1)$
  $\implies |z_0z_1 - z_2| < \epsilon\sqrt{1024/207}\,|z_0z_1|$
- Case R3: $|\Re(z_0z_1 - z_2)| \leq \epsilon \cdot (\frac{3}{2}a_0a_1)$
  $\implies |z_0z_1 - z_2| < \epsilon\sqrt{256/55}\,|z_0z_1|$

Absolute Complex Error

- Imaginary error: $|\Im(z_0 z_1 - z_2)| < \epsilon \cdot (2a_0 b_1 + 2b_0 a_1)$
- Case R1: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (2a_0 a_1 - b_0 b_1) + 2\epsilon^2 |z_0 z_1|$
  $\implies |z_0 z_1 - z_2| < \epsilon \left(\sqrt{32/7} + 2\epsilon \right) |z_0 z_1|$
- Case R2: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (\frac{7}{4} a_0 a_1)$
  $\implies |z_0 z_1 - z_2| < \epsilon \sqrt{1024/207} |z_0 z_1|$
- Case R3: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (\frac{3}{2} a_0 a_1)$
  $\implies |z_0 z_1 - z_2| < \epsilon \sqrt{256/55} |z_0 z_1|$
- Case R4: $|\Re(z_0 z_1 - z_2)| \leq \epsilon \cdot (a_0 a_1 + b_0 b_1)$
  $\implies |z_0 z_1 - z_2| < \epsilon \sqrt{5} |z_0 z_1|$

Worst-Case Multiplicands for $\beta = 2$

Theorem 2. *Assume that*

$$\frac{|z_0 z_1 - z_2|}{|z_0 z_1|} > \epsilon\sqrt{5 - n\epsilon} > \epsilon \cdot \max\left(\sqrt{1024/207}, \sqrt{32/7} + 2\epsilon\right)$$

for some positive integer n. Then $a_0 \neq b_0$, $a_1 \neq b_1$, and

$$a_0 a_1 = 1/2 + (j_{aa} + 1/2)\epsilon + k_{aa}\epsilon^2 \quad a_0 b_1 = 1/2 + (j_{ab} + 1/2)\epsilon + k_{ab}\epsilon^2$$
$$b_0 a_1 = 1/2 + (j_{ba} + 1/2)\epsilon + k_{ba}\epsilon^2 \quad b_0 b_1 = 1/2 + (j_{bb} + 1/2)\epsilon + k_{bb}\epsilon^2$$

for some integers j_{xy}, k_{xy} satisfying

$$0 \leq j_{aa}, j_{ab}, j_{ba}, j_{bb} < \frac{n}{4}, \qquad |k_{aa}|, |k_{bb}| < n, \qquad |k_{ab}|, |k_{ba}| < \frac{n}{2}$$

Sketch of Proof

- From the argument in Theorem 1, case R4 must hold: $\mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1) = \epsilon$, and there is no rounding error introduced in the subtraction.

Sketch of Proof

- From the argument in Theorem 1, case R4 must hold: $\mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1) = \epsilon$, and there is no rounding error introduced in the subtraction.
- Juggling of inequalities leads to

$$1 \leq |z_0 z_1|^2 \leq \frac{5}{5 - n\epsilon}$$

$$\epsilon^2 (5 - n\epsilon) < |z_0 z_1 - z_2|^2 < 5\epsilon^2$$

Sketch of Proof

- From the argument in Theorem 1, case R4 must hold: $\mathrm{ulp}(b_0 b_1) = \mathrm{ulp}(a_0 a_1) = \epsilon$, and there is no rounding error introduced in the subtraction.
- Juggling of inequalities leads to

$$1 \leq |z_0 z_1|^2 \leq \frac{5}{5 - n\epsilon}$$

$$\epsilon^2 (5 - n\epsilon) < |z_0 z_1 - z_2|^2 < 5\epsilon^2$$

- Considering what these bounds imply about $a_0 a_1$, $|a_0 a_1 - a_0 \otimes a_1|$, *et cetera*, provides the result desired.

Computation is Useful!

- At this point, I turned to computation.

Computation is Useful!

- At this point, I turned to computation.
    - Pruned exhaustive search of IEEE 754 single-precision inputs, using the results of Theorem 2 to eliminate most of the search space.

Computation is Useful!

- At this point, I turned to computation.
  - Pruned exhaustive search of IEEE 754 single-precision inputs, using the results of Theorem 2 to eliminate most of the search space.
  - Searching took about 5 CPU-hours.

Computation is Useful!

- At this point, I turned to computation.
  - Pruned exhaustive search of IEEE 754 single-precision inputs, using the results of Theorem 2 to eliminate most of the search space.
  - Searching took about 5 CPU-hours.
  - Worst case inputs:

$$a_0 = \frac{3}{4} \quad b_0 = \frac{3}{4}(1 - 4\epsilon) \quad a_1 = \frac{2}{3}(1 + 11\epsilon) \quad b_1 = \frac{2}{3}(1 + 5\epsilon)$$

Computation is Useful!

- At this point, I turned to computation.
  - Pruned exhaustive search of IEEE 754 single-precision inputs, using the results of Theorem 2 to eliminate most of the search space.
  - Searching took about 5 CPU-hours.
  - Worst case inputs:

$$a_0 = \frac{3}{4} \quad b_0 = \frac{3}{4}(1 - 4\epsilon) \quad a_1 = \frac{2}{3}(1 + 11\epsilon) \quad b_1 = \frac{2}{3}(1 + 5\epsilon)$$

- This suggests a more general form...

Worst-Case Multiplicands for $\beta = 2$

Theorem 3. *Assume that*

$$\frac{|z_0 z_1 - z_2|}{|z_0 z_1|} > \epsilon\sqrt{5 - n\epsilon} > \epsilon \cdot \max\left(\sqrt{1024/207}, \sqrt{32/7} + 2\epsilon\right)$$

for some $n < \frac{1}{4}\epsilon^{-1/2}$ *and* $\epsilon \leq 2^{-6}$*. Then there exist integers* c_0*,* d_0*,* α_0*,* β_0*,* c_1*,* d_1*,* α_1*,* β_1 *satisfying*

$$z_0 = \frac{c_0}{d_0}(1 + i + (\alpha_0 + \beta_0 i)\epsilon) \qquad z_1 = \frac{c_1}{d_1}(1 + i + (\alpha_1 + \beta_1 i)\epsilon)$$

$$\min(\alpha_0, \beta_0) + \min(\alpha_1, \beta_1) \geq 0 \qquad 2c_0 c_1 = d_0 d_1 < 3n$$

$$|\alpha_0\alpha_1|, |\alpha_0\beta_1|, |\beta_0\alpha_1|, |\beta_0\beta_1| < n \qquad \frac{1}{2} < a_0, b_0, a_1, b_1 < 1.$$

Worst-Case Multiplicands for IEEE 754

- Now an “exhaustive” search is far less exhausting.

Worst-Case Multiplicands for IEEE 754

- Now an “exhaustive” search is far less exhausting.
- The IEEE 754 single-precision worst-case inputs are

$$a_0 = \frac{3}{4} \quad b_0 = \frac{3}{4}(1 - 4\epsilon) \quad a_1 = \frac{2}{3}(1 + 11\epsilon) \quad b_1 = \frac{2}{3}(1 + 5\epsilon)$$

and have a relative error of $\epsilon \cdot \sqrt{4.9999899864}$.

Worst-Case Multiplicands for IEEE 754

- Now an "exhaustive" search is far less exhausting.
- The IEEE 754 single-precision worst-case inputs are

$$a_0 = \frac{3}{4} \quad b_0 = \frac{3}{4}(1-4\epsilon) \quad a_1 = \frac{2}{3}(1+11\epsilon) \quad b_1 = \frac{2}{3}(1+5\epsilon)$$

  and have a relative error of $\epsilon \cdot \sqrt{4.9999899864}$.
- The IEEE 754 double-precision worst-case inputs are

$$a_0 = \frac{3}{4}(1+4\epsilon) \quad b_0 = \frac{3}{4} \quad a_1 = \frac{2}{3}(1+7\epsilon) \quad b_1 = \frac{2}{3}(1+\epsilon)$$

  and have a relative error of $\epsilon \cdot \sqrt{4.999999999999893}$.

Worst-Case Multiplicands for IEEE 754

- Now an “exhaustive” search is far less exhausting.
- The IEEE 754 single-precision worst-case inputs are

$$a_0 = \frac{3}{4} \quad b_0 = \frac{3}{4}(1-4\epsilon) \quad a_1 = \frac{2}{3}(1+11\epsilon) \quad b_1 = \frac{2}{3}(1+5\epsilon)$$

  and have a relative error of $\epsilon \cdot \sqrt{4.9999899864}$.
- The IEEE 754 double-precision worst-case inputs are

$$a_0 = \frac{3}{4}(1+4\epsilon) \quad b_0 = \frac{3}{4} \quad a_1 = \frac{2}{3}(1+7\epsilon) \quad b_1 = \frac{2}{3}(1+\epsilon)$$

  and have a relative error of $\epsilon \cdot \sqrt{4.999999999999893}$.
- Clearly $\epsilon\sqrt{5}$ is the best (practical) bound possible.

Roots of Unity

- We have the best possible error bound on multiplication, but for FFTs we still need a bound on the errors in the precomputed roots of unity.

Roots of Unity

- We have the best possible error bound on multiplication, but for FFTs we still need a bound on the errors in the precomputed roots of unity.
- Even if your CPU provides exactly rounded transcendental functions, $\cos(2\pi k/2^n) + i\sin(2\pi k/2^n)$ still suffers from rounding in the value of π used and the multiplication $k \otimes \pi$, in addition to the two trigonometric evaluations.

Roots of Unity

- We have the best possible error bound on multiplication, but for FFTs we still need a bound on the errors in the precomputed roots of unity.
- Even if your CPU provides exactly rounded transcendental functions, $\cos(2\pi k/2^n) + i\sin(2\pi k/2^n)$ still suffers from rounding in the value of π used and the multiplication $k \otimes \pi$, in addition to the two trigonometric evaluations.
- Many FFT implementations use shockingly inaccurate iterations to compute roots of unity.

Roots of Unity

- We have the best possible error bound on multiplication, but for FFTs we still need a bound on the errors in the precomputed roots of unity.
- Even if your CPU provides exactly rounded transcendental functions, $\cos(2\pi k/2^n) + i\sin(2\pi k/2^n)$ still suffers from rounding in the value of π used and the multiplication $k \otimes \pi$, in addition to the two trigonometric evaluations.
- Many FFT implementations use shockingly inaccurate iterations to compute roots of unity.
- It is possible to compute the 2^nth roots of unity in $\frac{27}{32} \cdot 2^n + \mathrm{O}(n)$ FLOPS with a maximum error $< 2\epsilon$.

Roots of Unity

- We have the best possible error bound on multiplication, but for FFTs we still need a bound on the errors in the precomputed roots of unity.
- Even if your CPU provides exactly rounded transcendental functions, $\cos(2\pi k/2^n) + i\sin(2\pi k/2^n)$ still suffers from rounding in the value of π used and the multiplication $k \otimes \pi$, in addition to the two trigonometric evaluations.
- Many FFT implementations use shockingly inaccurate iterations to compute roots of unity.
- It is possible to compute the 2^nth roots of unity in $\frac{27}{32} \cdot 2^n + \mathrm{O}(n)$ FLOPS with a maximum error $< 2\epsilon$.
  - ... I need to find time to write this paper some day.

A Historical Note

"Indeed, in unpublished work R.P. Brent has demonstrated that in base 2, for example, [the error term] can be reduced to $\sqrt{5}\ldots$"

— F.W.J. Olver, 1986

References

- N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, Second Edition, SIAM, 2002.
- C. Percival, *Rapid multiplication modulo the sum and difference of highly composite numbers*, Math. Comp. **72** (2002), 387–395.
- R.P. Brent, C. Percival, P. Zimmermann, *Error bounds on complex floating-point multiplication*, Math. Comp., to appear.