Endpoint and Midpoint Interval Representations Theoretical and Computational Comparison

Tomáš Dzetkulič

Institute of Computer Science
Academy of Sciences of the Czech Republic

5th of June 2012

Example

Task: Compute an interval enclosure for $x = 1/15$

Example

Task: Compute an interval enclosure for $x = 1/15$

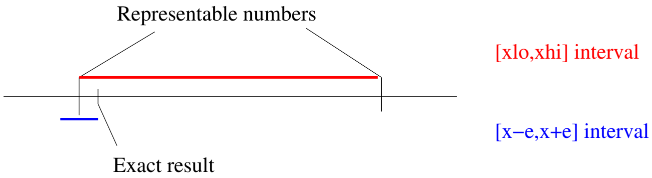
Classical interval analysis:
$x \in [6.6666666666666657415 \times 10^{-2}, 6.6666666666666796193 \times 10^{-2}]$
Width: 1.387779×10^{-17}

Example

Task: Compute an interval enclosure for $x = 1/15$

Classical interval analysis:
$x \in [6.6666666666666657415 \times 10^{-2}, 6.6666666666666796193 \times 10^{-2}]$
Width: 1.387779×10^{-17}

Another possible representations:
$6.6666666666666657415 \times 10^{-2} \pm 9.252 \times 10^{-19}$
$6.6666666666666657415 \times 10^{-2} + [0, 9.252 \times 10^{-19}]$

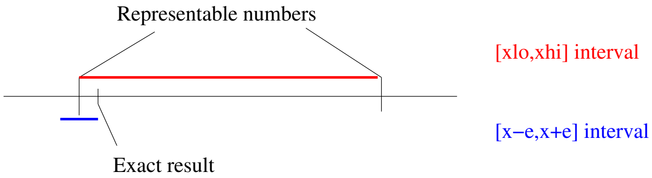


Example

Task: Compute an interval enclosure for $x = 1/15$

Classical interval analysis:
$x \in [6.666666666666666657415 \times 10^{-2}, 6.666666666666666796193 \times 10^{-2}]$
Width: 1.387779×10^{-17}

Another possible representations:
$6.666666666666666657415 \times 10^{-2} \pm 9.252 \times 10^{-19}$
$6.666666666666666657415 \times 10^{-2} + [0, 9.252 \times 10^{-19}]$



$6.666666666666666657415 \times 10^{-2} + [9.251 \times 10^{-19}, 9.252 \times 10^{-19}]$
Width: $< 10^{-30}$

Interval Types

Let Ω be the set of numbers representable on digital computer

We consider four kinds of intervals:

1. $[x_{lo}, x_{hi}]$ such that $x_{lo}, x_{hi} \in \Omega$
2. $[x - e, x + e]$ such that $x, e \in \Omega$
3. $[x - e_{lo}, x + e_{hi}]$ such that $x, e_{lo}, e_{hi} \in \Omega; e_{lo}, e_{hi} >= 0$
4. $[x - e_{lo}, x + e_{hi}]$ such that $x, e_{lo}, e_{hi} \in \Omega$

Interval Types

Let Ω be the set of numbers representable on digital computer

We consider four kinds of intervals:

1. $[x_{lo}, x_{hi}]$ such that $x_{lo}, x_{hi} \in \Omega$
2. $[x - e, x + e]$ such that $x, e \in \Omega$
3. $[x - e_{lo}, x + e_{hi}]$ such that $x, e_{lo}, e_{hi} \in \Omega; e_{lo}, e_{hi} >= 0$
4. $[x - e_{lo}, x + e_{hi}]$ such that $x, e_{lo}, e_{hi} \in \Omega$

Assumptions: intervals are narrow and the entire mantissa is used

$\oplus, \otimes$ are *round to nearest, ties to even* addition and multiplication

$\overline{+}, \underline{+}$ denote operations rounded up/down

Computing With Midpoint Intervals

In [1], Dekker showed that given $a, b \in \Omega$
$(a \oplus b) - (a + b) \in \Omega$ and $(a \otimes b) - (a \times b) \in \Omega$
(if there was not overflow or underflow in multiplication)

i.e., the exact result of the arithmetic operation can be given as an unevaluated sum of two floating point numbers

Computing With Midpoint Intervals

In [1], Dekker showed that given $a, b \in \Omega$
$(a \oplus b) - (a + b) \in \Omega$ and $(a \otimes b) - (a \times b) \in \Omega$
(if there was not overflow or underflow in multiplication)

i.e., the exact result of the arithmetic operation can be given as an unevaluated sum of two floating point numbers

Let $\texttt{add}(a, b) = (x, y)$ be a function that computes x, y such that $x = a \oplus b$ and $a + b = x + y$

Computing With Midpoint Intervals

In [1], Dekker showed that given $a, b \in \Omega$
$(a \oplus b) - (a + b) \in \Omega$ and $(a \otimes b) - (a \times b) \in \Omega$
(if there was not overflow or underflow in multiplication)

i.e., the exact result of the arithmetic operation can be given as an unevaluated sum of two floating point numbers

Let $\texttt{add}(a, b) = (x, y)$ be a function that computes x, y such that $x = a \oplus b$ and $a + b = x + y$

We can then compute $[x_1 - e_1, x_1 + e_1] + [x_2 - e_2, x_2 + e_2]$:
1. $(x, e_3) := \texttt{add}(x_1, x_2)$
2. $e := e_1 \overline{+} e_2 \overline{+} |e_3|$
3. $\textbf{return}[x - e, x + e]$

Addition With Huge Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-50})$
Exact result mantissa is long (ones in the beginning and in the end)

In classical interval analysis one of the interval bounds changes to next floating point number
The error introduced is ϵ (2^{-52})

Addition With Huge Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-50})$
Exact result mantissa is long (ones in the beginning and in the end)

In classical interval analysis one of the interval bounds changes to next floating point number
The error introduced is ϵ (2^{-52})

In intervals of the second kind, only the smaller term is added to error
The error introduced is thus of the magnitude of the smaller term

Addition With Huge Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-50})$
Exact result mantissa is long (ones in the beginning and in the end)

In classical interval analysis one of the interval bounds changes to next floating point number
The error introduced is ϵ (2^{-52})

In intervals of the second kind, only the smaller term is added to error
The error introduced is thus of the magnitude of the smaller term

$\rightarrow$ In case small intervals are often added to our interval, the use of second interval kind has a huge advantage over the classical interval

Addition With Medium Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-10})$

Exact result mantissa is longer than allowed by the standard
$\rightarrow$ rounding occurs

In classical interval analysis the bounds of exact result can lie anywhere in between of two representable numbers

The expected error introduced is ϵ

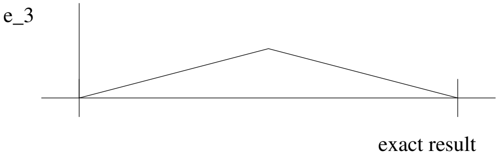
Addition With Medium Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-10})$

Exact result mantissa is longer than allowed by the standard
$\rightarrow$ rounding occurs

In classical interval analysis the bounds of exact result can lie anywhere in between of two representable numbers
The expected error introduced is ϵ

In intervals of the second kind, the expected magnitude of e_3 is $\epsilon/4$



In intervals of the second kind, the expected error introduced is $\epsilon/2$

Addition With Medium Magnitude Difference

Example: $(1.3) + (1.4 \times 10^{-10})$

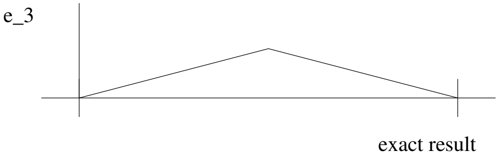
Exact result mantissa is longer than allowed by the standard

$\rightarrow$ rounding occurs

In classical interval analysis the bounds of exact result can lie anywhere in between of two representable numbers

The expected error introduced is ϵ

In intervals of the second kind, the expected magnitude of e_3 is $\epsilon/4$



In intervals of the second kind, the expected error introduced is $\epsilon/2$

For multiplication, the same observations are valid

Addition With No Magnitude Difference

Example: $(1.3) + (1.4)$
Exact result mantissa is one bit longer than allowed

In classical interval analysis the bounds of exact result are representable with the probability 0.5
The expected error introduced is $\epsilon/2$

Addition With No Magnitude Difference

Example: $(1.3) + (1.4)$
Exact result mantissa is one bit longer than allowed

In classical interval analysis the bounds of exact result are representable with the probability 0.5
The expected error introduced is $\epsilon/2$

In intervals of the second kind, e_3 is $\epsilon/2$ with the probability 0.5

In intervals of the second kind, the expected error introduced is $\epsilon/2$

The Effect of Rounding To Nearest

In previous example, result was one bit longer than allowed

Either:

1. last bit was zero and there was no rounding
2. there was a tie in rounding

The Effect of Rounding To Nearest

In previous example, result was one bit longer than allowed

Either:

1. last bit was zero and there was no rounding
2. there was a tie in rounding

The result of *round to nearest, ties to even* rounding mode has zero last mantissa bit

The Effect of Rounding To Nearest

In previous example, result was one bit longer than allowed

Either:

1. last bit was zero and there was no rounding
2. there was a tie in rounding

The result of *round to nearest, ties to even* rounding mode has zero last mantissa bit

It is more likely that the result of a succesive operation with such a number is in Ω

Example: (a+b)+(c+d)

The Effect of Rounding To Nearest

In previous example, result was one bit longer than allowed

Either:

1. last bit was zero and there was no rounding
2. there was a tie in rounding

The result of *round to nearest, ties to even* rounding mode has zero last mantissa bit

It is more likely that the result of a succesive operation with such a number is in Ω

Example: (a+b)+(c+d)

This effect does not affect intervals of the first kind, since there is never a tie in a directed rounding

Addition Of Opposite Numbers

Example: $(1.3) + (-1.4)$
Result mantissa is shorter than allowed by the standard

There is no error introduced in classical interval analysis

There is a minor error introduced in the directed rounding of $e_1 \overline{+} e_2$ in intervals of the second kind

Implementation Pitfalls and Wide Intervals

Special care has to be taken for underflowing multiplication

Dekker algorithm does not work in that case

Implementation Pitfalls and Wide Intervals

Special care has to be taken for underflowing multiplication

Dekker algorithm does not work in that case

$\rightarrow$ Underflowing results can be enclosed by 0 ± 10^{-200}

Implementation Pitfalls and Wide Intervals

Special care has to be taken for underflowing multiplication

Dekker algorithm does not work in that case

$\rightarrow$ Underflowing results can be enclosed by 0 ± 10^{-200}

In wide intervals as e, e_{lo} and e_{hi} gain magnitude, additional error is introduced in directed rounding of error

Implementation Pitfalls and Wide Intervals

Special care has to be taken for underflowing multiplication

Dekker algorithm does not work in that case

$\rightarrow$ Underflowing results can be enclosed by 0 ± 10^{-200}

In wide intervals as e, e_{lo} and e_{hi} gain magnitude, additional error is introduced in directed rounding of error

Multiplication of wide intervals $[1-1, 1+1] \times [1-1, 1+1]$ yields suboptimal results ($[1-3, 1+3]$)

$\rightarrow$ shift of the interval center is required in intervals of the second kind

Rigorous Polynomial

1. $\sum_i [a_i, b_i] x^i$
2. $(\sum_i a_i x^i) + [-e, e]$
3. $(\sum_i a_i x^i) + [-e_{lo}, e_{hi}]$

Rigorous Polynomial

1. $\sum_i [a_i, b_i] x^i$
2. $(\sum_i a_i x^i) + [-e, e]$
3. $(\sum_i a_i x^i) + [-e_{lo}, e_{hi}]$

In [2], Neumaier says:
"I have not seen any convincing evidence that the use of floating point numbers as coefficients is an essential improvement over using narrow interval coefficients."

Rigorous Polynomial

1. $\sum_i [a_i, b_i] x^i$
2. $(\sum_i a_i x^i) + [-e, e]$
3. $(\sum_i a_i x^i) + [-e_{lo}, e_{hi}]$

In [2], Neumaier says:
"I have not seen any convincing evidence that the use of floating point numbers as coefficients is an essential improvement over using narrow interval coefficients."

In case an operation rounds a polynomial coefficient, the error introduced depends also on the value of the monomial
If $x \in [-1, 1]$ then $x^i \in [-1, 1] \rightarrow$ the sign of the error does not matter

Rigorous Polynomial

1. $\sum_i [a_i, b_i] x^i$
2. $(\sum_i a_i x^i) + [-e, e]$
3. $(\sum_i a_i x^i) + [-e_{lo}, e_{hi}]$

In [2], Neumaier says:
"I have not seen any convincing evidence that the use of floating point numbers as coefficients is an essential improvement over using narrow interval coefficients."

In case an operation rounds a polynomial coefficient, the error introduced depends also on the value of the monomial
If $x \in [-1, 1]$ then $x^i \in [-1, 1] \rightarrow$ the sign of the error does not matter

In second and third case we need less memory to store polynomial

Computational Experiments

Test 1: Add 10000 random numbers from interval $[-1.0, 1.0]$
Test 2: Add 10000 random numbers from interval $[0.5, 1.5]$
Test 3: Multiply 10000 random numbers from distribution $e^{[-1.0,1.0]}$

Test versions: Sequential and Divide&Conquer

Computational Experiments

Test 1: Add 10000 random numbers from interval $[-1.0, 1.0]$
Test 2: Add 10000 random numbers from interval $[0.5, 1.5]$
Test 3: Multiply 10000 random numbers from distribution $e^{[-1.0,1.0]}$

Test versions: Sequential and Divide&Conquer

	$[a, b]$	$[a - e, a + e]$	$[a - e_{lo}, a + e_{hi}]$
Test	Error		
1 Sequential	8.7×10^{-11}	4.4×10^{-11}	2.2×10^{-11}
1 D&C	1.1×10^{-12}	7.6×10^{-13}	3.8×10^{-13}
2 Sequential	8.3×10^{-9}	4.1×10^{-9}	2.1×10^{-9}
2 D&C	1.1×10^{-11}	8.9×10^{-12}	4.5×10^{-12}
3	1.6×10^{-12}	1.0×10^{-12}	5.0×10^{-13}

Arithmetic Operations Count

	$[a, b]$	$[a-e, a+e]$	$[a-e_{lo}, a+e_{hi}]$
Addition			
Rounding mode change	2	2	2
Add	2	8	9
Time(10^9 operations)	$40s$	$48s$	$51s$
Multiplication			
Rounding mode change	2	2	2
Add	0	14	17
Mul	8	9	18
Min/Max/Abs	6	3	7
Time(10^9 operations)	$57s$	$63s$	$86s$

Conclusion

We have compared three kinds of intervals

Intervals of second and third kind provide tighter enclosures for narrow intervals

Conclusion

We have compared three kinds of intervals

Intervals of second and third kind provide tighter enclosures for narrow intervals

Computational experiments confirm the advantage of midpoint intervals

Conclusion

We have compared three kinds of intervals

Intervals of second and third kind provide tighter enclosures for narrow intervals

Computational experiments confirm the advantage of midpoint intervals

Thank you for you attention.

[1] T. Dekker. A floating-point technique for extending the available precision. *Numerische Mathematik*, 18:224—-242, 1971/72.

[2] A. Neumaier. Taylor forms–use and limits. *Reliable Computing*, pages 43–79, 2003.