Numerical error in Floating Point arithmetic, what you should look out for

For CITI's PhDDays

Orégane Desrentes

14th May 2024

Motivation

Why doesn't my code work ?

What are Floating Point Numbers

Floating Point numbers ?

Computer representation for real numbers
$\approx$ scientific representation

Floating Point numbers ?

Computer representation for real numbers
$\approx$ scientific representation
$(-1)^S \times 2^E + 1.F$
S, E, F are stored in binary, in a finite format.
For floats :
$w_S = 1, w_E = 8, w_F = 23$

Floating Point numbers ?

Computer representation for real numbers
$\approx$ scientific representation
$(-1)^S \times 2^E + 1.F$
S, E, F are stored in binary, in a finite format.
For floats :
$w_S = 1, w_E = 8, w_F = 23$
Avoid numbers being flushed to 0 abruptly : gradual underflow.
Guarantees if $x \neq y$ then $x - y \neq 0$ and other useful properties.
Those numbers are called subnormal and have less precision than w_F.

Floating Point numbers ?

Computer representation for real numbers
$\approx$ scientific representation
$(-1)^S \times 2^E + 1.F$
S, E, F are stored in binary, in a finite format.
For floats :
$w_S = 1, w_E = 8, w_F = 23$
Avoid numbers being flushed to 0 abruptly : gradual underflow.
Guarantees if $x \neq y$ then $x - y \neq 0$ and other useful properties.
Those numbers are called subnormal and have less precision than w_F.

Other special values :
$+0, -0, +\infty, -\infty, NaN$

Floating Point numbers ?

Computer representation for real numbers
$\approx$ scientific representation
$(-1)^S \times 2^E + 1.F$
S, E, F are stored in binary, in a finite format.
For floats :
$w_S = 1, w_E = 8, w_F = 23$
Avoid numbers being flushed to 0 abruptly : gradual underflow.
Guarantees if $x \neq y$ then $x - y \neq 0$ and other useful properties.
Those numbers are called subnormal and have less precision than w_F.

Other special values :
$+0, -0, +\infty, -\infty, NaN$

MPFR : Floating Point numbers with arbitrary precision

C library that allow $w_E = 32$ and w_F of arbitrary size.
In my demos, comparing precision will be done with MPFR as the reference.

Examples

Range issues : $e^{x+y} \neq e^x \times e^y$

The biggest positive floating point number is $\simeq 3.410^{38}$
The smallest positive floating point number is $\simeq 1.7 \times 10^{-45}$
If your computation comes out of this range, **anything** can happen.

Range issues : $e^{x+y} \neq e^x \times e^y$

The biggest positive floating point number is $\simeq 3.410^{38}$
The smallest positive floating point number is $\simeq 1.7 \times 10^{-45}$
If your computation comes out of this range, **anything** can happen.

- Overflow issues : $e^{600} \times e^{-600} = +\infty \times 0 = NaN$ while $e^{600-600} = e^0 = 1$

Range issues : $e^{x+y} \neq e^x \times e^y$

The biggest positive floating point number is $\simeq 3.410^{38}$
The smallest positive floating point number is $\simeq 1.7 \times 10^{-45}$
If your computation comes out of this range, **anything** can happen.

- Overflow issues : $e^{600} \times e^{-600} = +\infty \times 0 = NaN$ while $e^{600-600} = e^0 = 1$
- Underflow issues : $e^{40} \times e^{-130} = 2.35 \times 10^{17} \times 0 = 0$ while $e^{40-130} = e^{-90} \simeq 8.19 \times 10^{-40}$

Range issues : $e^{x+y} \neq e^x \times e^y$

The biggest positive floating point number is $\simeq 3.410^{38}$
The smallest positive floating point number is $\simeq 1.7 \times 10^{-45}$
If your computation comes out of this range, **anything** can happen.

- Overflow issues : $e^{600} \times e^{-600} = +\infty \times 0 = NaN$ while $e^{600-600} = e^0 = 1$
- Underflow issues : $e^{40} \times e^{-130} = 2.35 \times 10^{17} \times 0 = 0$ while $e^{40-130} = e^{-90} \simeq 8.19 \times 10^{-40}$
- Underflow issues (hard to debug version) : $e^{-102} \times e^{75} = 2.092554 \times 10^{-12}$ while $e^{-102+75} = e^{-27} \simeq 1.879529 \times 10^{-12}$ (11% error !!)
  This one is due to a floating point quirk : gradual underflow.

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

```
>>> 0.1 + 0.2
0.30000000000000004
>>>
```

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$\begin{array}{l|l} \overbrace{1.00000000} & 1010 \\ \hline & 0. \end{array}$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$\begin{array}{r|l} \overbrace{1.00000000} & 1010 \\ \hline & 0.0 \end{array}$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

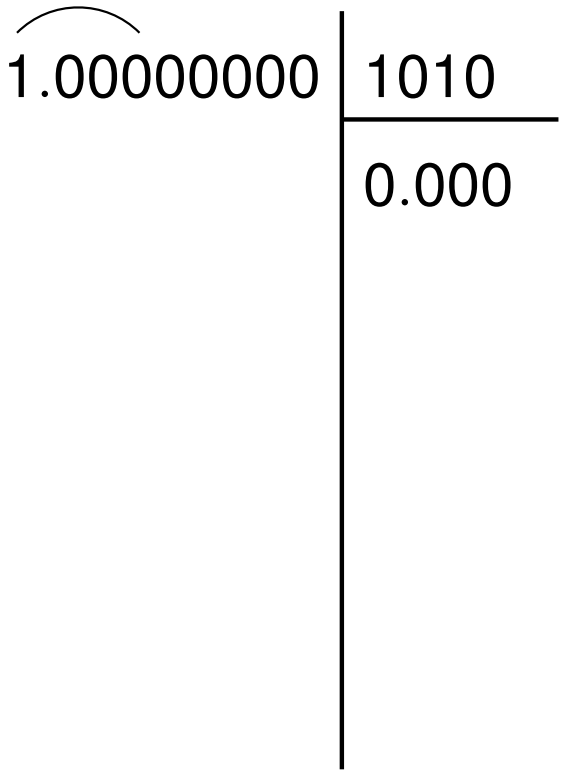
$$\begin{array}{r|l} \overbrace{1.00000000} & 1010 \\ \hline & 0.00 \end{array}$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$



Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

1.00000000	1010
− 1010	0.0001
00110	

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)
0.1 cannot be represented exactly in binary
In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$
\begin{array}{rl|l}
 & 1.00000000 & 1010 \\
\hline
- & 1010 & 0.00011 \\
\hline
 & 001100 & \\
- & 1010 & \\
\hline
 & 0010 & \\
\end{array}
$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)
0.1 cannot be represented exactly in binary
In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$
\begin{array}{rl|l}
 & 1.00000000 & 1010 \\
\hline
- & 1010 & 0.000110 \\
\cline{2-2}
 & 001100 & \\
- & 1010 & \\
\cline{2-2}
 & 00100 &
\end{array}
$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)
0.1 cannot be represented exactly in binary
In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$
\begin{array}{rr|l}
 & \overset{\frown}{1.00000000} & 1010 \\ \cline{3-3}
- & \underline{1010} & 0.0001100 \\
 & 001100 & \\
- & \underline{1010} & \\
 & 001000 &
\end{array}
$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$
\begin{array}{rl|l}
 & 1.00000000 & 1010 \\
\hline
- & 1010 & 0.00011001 \\
\cline{1-2}
 & 001100 & \\
- & 1010 & \\
\cline{1-2}
 & 0010000 & \\
- & 1010 & \\
\cline{1-2}
 & 0110 &
\end{array}
$$

Representation issues : counting time by 100ms increments

Know as the Patriot bug (28 dead)

0.1 cannot be represented exactly in binary

In floating point 32, it is stored as $2^{-4} \times 1.10011001100110011001101$

$$
\begin{array}{rl|l}
 & 1.00000000 & 1010 \\
\cline{3-3}
- & \underline{1010} & 0.00011001 \\
 & 001100 & \\
 & \underline{1010} & \\
 & 0010000 & \\
 & \underline{1010} & \\
 & 0110 & \\
 & \vdots &
\end{array}
$$

Rounding issues : For loop with floating point

What happens when you execute this :

```
for (float i = 0.0; i<16777217.0; i+=1.0) {
        if (i >= 16777215 || (int) i % 1000000 == 0) {
                printf("%2.1f ", i);
        }
}
```

Rounding issues : For loop with floating point

What happens when you execute this :

```
for (float i = 0.0; i<16777217.0; i+=1.0) {
        if (i >= 16777215 || (int) i % 1000000 == 0) {
                printf("%2.1f ", i);
        }
}
```

When i is too big, $i + 1.0 = i$

Rounding issues : For loop with floating point

What happens when you execute this :

```
for (float i = 0.0; i<16777217.0; i+=1.0) {
        if (i >= 16777215 || (int) i % 1000000 == 0) {
                printf("%2.1f ", i);
        }
}
```

When i is too big, $i + 1.0 = i$

Numerical example in decimal, with 3 digits precision :

$1.23 \times 10^3 + 1.00 \times 10^0 = 1230 + 1 = 1.23 \times 10^3$

because 1231 is closer to 1230 than 1240

Cancellation issues : Heron's formula for the area of a triangle

For a triangle with sides of size a, b, c, you can compute the area of the triangle with Heron's formula :

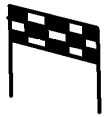
$$s = \frac{a+b+c}{2}$$

$$\mathscr{A} = \sqrt{s(s-a)(s-b)(s-c)}$$

This is numerically unstable. Kahan's [1] version of this formula, for $a \geq b \geq c$ is :

$$\mathscr{A} = \frac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}$$

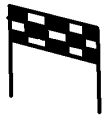
Numerical example



Let's use :

- $a = 100000$
- $b = 50024$
- $c = 50000$

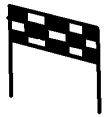
Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

Numerical example

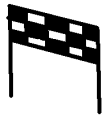


Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = \frac{a+b+c}{2}$

Numerical example

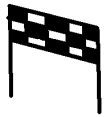


Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$s = \frac{1.0000 \times 10^5 + 5.0024 \times 10^4 + 5.0000 \times 10^4}{2}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = \frac{1.5002 \times 10^5 + 5.0000 \times 10^4}{2}$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = \frac{2.0002 \times 10^5}{2}$

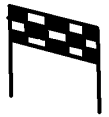
Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

Numerical example



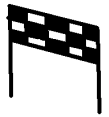
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = \sqrt{s(s-a)(s-b)(s-c)}$

Numerical example



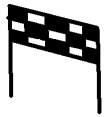
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{s(s-a)(s-b)(1.0001 \times 10^5 - 5.0000 \times 10^4)}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = \sqrt{s(s-a)(s-b)(5.0010 \times 10^4)}$

Numerical example

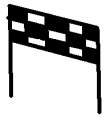
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{s(s-a)(1.0001 \times 10^5 - 5.0024 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{s(s-a)(4.9986 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{s(1.0001 \times 10^5 - 1.0000 \times 10^5)(4.9986 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

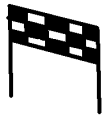
$s = 1.0001 \times 10^5$

$\mathscr{A} = \sqrt{s(1.0000 \times 10^1)(4.9986 \times 10^4)(5.0010 \times 10^4)}$

In the exact computation $s - a = 100012 - 100000 = 12 \neq 10$.

This is a huge relative error even if the absolute error *shouldn't* be too bad . . .

Numerical example



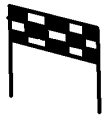
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{(1.0001 \times 10^5)(1.0000 \times 10^1)(4.9986 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{(1.0001 \times 10^6)(4.9986 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example

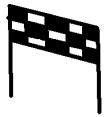
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$$\mathscr{A} = \sqrt{(1.0001 \times 10^6)(4.9986 \times 10^4)(5.0010 \times 10^4)}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = \sqrt{(4.9991 \times 10^{10})(5.0010 \times 10^4)}$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = \sqrt{2.5000 \times 10^{15}}$

Numerical example

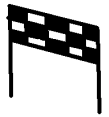
Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = 5.0000 \times 10^7$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$s = 1.0001 \times 10^5$

$\mathscr{A} = 5.0000 \times 10^7$

$\mathscr{A} = \frac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

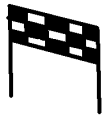
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(a + 1.0002 \times 10^5)(c - (a-b))(c + (a-b))(a + (b-c))}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

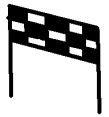
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(1.0000 \times 10^5 + 1.0002 \times 10^5)(c-(a-b))(c+(a-b))(a+(b-c))}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

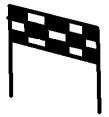
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(c-(a-b))(c+(a-b))(a+(b-c))}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

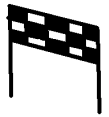
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(c - 4.9976 \times 10^4)(c + 4.9976 \times 10^4)(a + (b - c))}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(5.0000 \times 10^4 - 4.9976 \times 10^4)(c + 4.9976 \times 10^4)(a + (b - c))}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

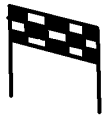
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(c + 4.9976 \times 10^4)(a + (b - c))}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(5.0000 \times 10^4 + 4.9976 \times 10^4)(a + (b - c))}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(9.9976 \times 10^4)(a + (b - c))}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(9.9976 \times 10^4)(a + 2.4000 \times 10^1)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(9.9976 \times 10^4)(1.0000 \times 10^5 + 2.4000 \times 10^1)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \tfrac{1}{4}\sqrt{(2.0002 \times 10^5)(2.4000 \times 10^1)(9.9976 \times 10^4)(1.0002 \times 10^5)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(4.8005 \times 10^6)(9.9976 \times 10^4)(1.0002 \times 10^5)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{(4.7993 \times 10^{11})(1.0002 \times 10^5)}$$

Numerical example

Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

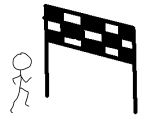
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \frac{1}{4}\sqrt{4.8003 \times 10^{16}}$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

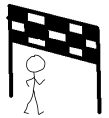
$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = \tfrac{1}{4} \times 2.1910 \times 10^8$$

Numerical example



Let's use :

- $a = 1.0000 \times 10^5$
- $b = 5.0024 \times 10^4$
- $c = 5.0000 \times 10^4$

$$b + c = 5.0024 \times 10^4 + 5.0000 \times 10^4 = 1.0002 \times 10^5$$
$$a - b = 1.0000 \times 10^5 - 5.0024 \times 10^4 = 4.9976 \times 10^4$$
$$b - c = 5.0024 \times 10^4 - 5.0000 \times 10^4 = 2.4000 \times 10^1$$

$$s = 1.0001 \times 10^5$$

$$\mathscr{A} = 5.0000 \times 10^7$$

$$\mathscr{A} = 5.4774 \times 10^7$$

Numerical example - error

First formula gives result 5.0000×10^7
Second formula gives result 5.4774×10^7
What is the real value ?

```
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> s = (100000+50024+50000) // 2 # exact
>>> s
100012
>>> tmp = s * (s-100000) * ( s - 50024) * (s - 50000) # exact (int)
>>> tmp
3000359827179264
>>> math.sqrt(tmp) # this is approached
54775540.40974186
>>>
```

Relative error of formula 1 : 8.7%
Relative error of formula 2 : $2.8 \times 10^{-3}\%$

Area of a triangle : Why would anyone need this ?

Example picture [2]

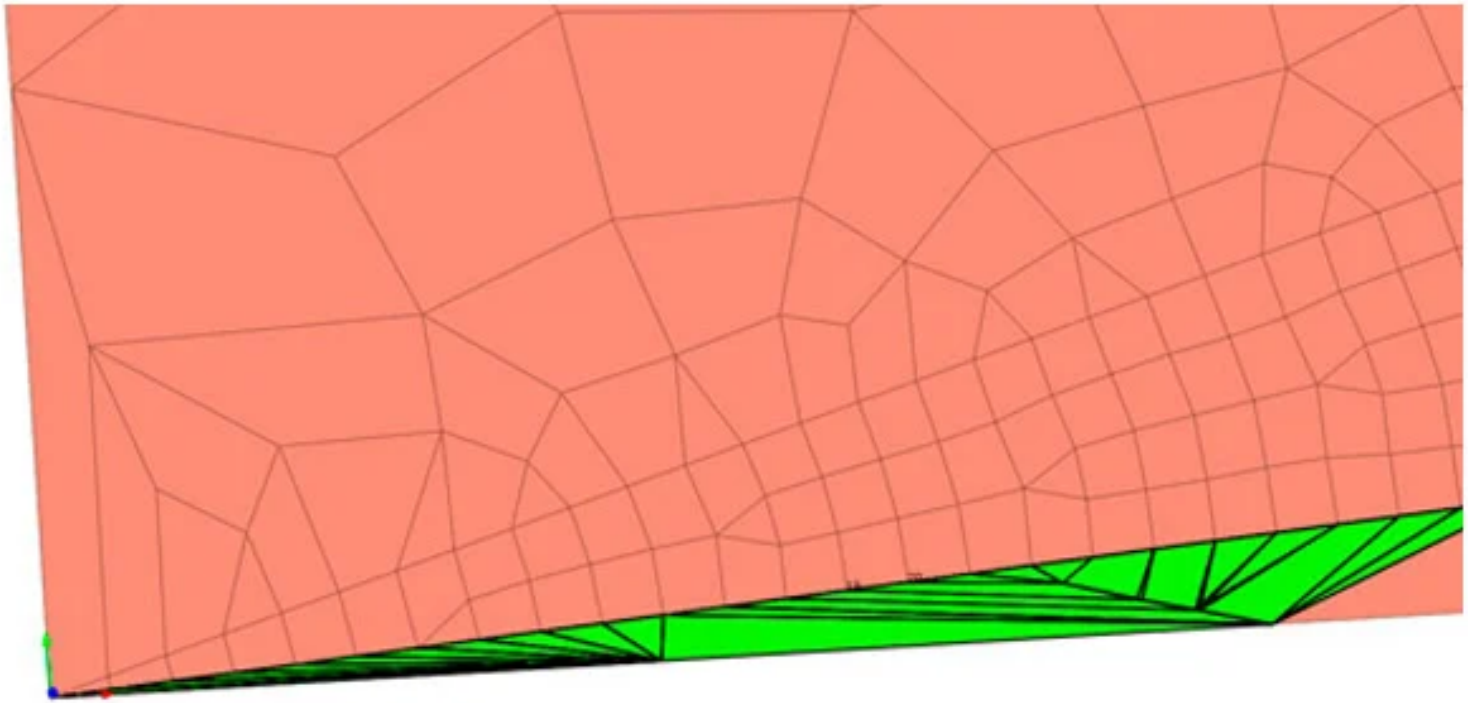


Figure 1_Distorted elements due to the transition from large elements to small ones

Conclusion

- Floating-Point numbers are not associative or distributive.

Conclusion

- Floating-Point numbers are not associative or distributive.
- Loops are particularly sensitive to errors.

Conclusion

- Floating-Point numbers are not associative or distributive.
- Loops are particularly sensitive to errors.
- Compilers (like GCC) assume you know what you're doing (and will use the order of operation you gave them)

Conclusion

- Floating-Point numbers are not associative or distributive.
- Loops are particularly sensitive to errors.
- Compilers (like GCC) assume you know what you're doing (and will use the order of operation you gave them)
- If the formula you're using is bad, there's a better one to be found.

References

W. Kahan

Miscalculating Area and Angles of a Needle-like Triangle

https://people.eecs.berkeley.edu/~wkahan/Triangle.pdf

Midas Geotech

Blog | Verification And Quality Assurance Of Finite Element Meshes And Errors In The Model

https://www.midasgeotech.com/blog/finite-element-mesh-and-error