Elementary Numerical Mathematics

Supporting material for

Round-off Errors and Computer Arithmetic

Winter Term 2019/20

Gerhard Wellein
HPC Services - Regionales Rechenzentrum Erlangen (RRZE)
Department for Computer Science

Round-off Errors and Computer Arithmetic

- **`1 ; 999,999 ; 1,000,000`** : These numbers can be represented *exactly* in IEEE 754 standard (32-Bit)

- IEEE 754 32-Bit representation of **`0.999999`**

Tools & Thoughts

IEEE-754 Floating Point Converter

Translations: de

This page allows you to convert between the decimal representation of numbers (like "1.02") and the binary format used by all modern CPUs (IEEE 754 floating point).

IEEE 754 Converter (JavaScript), V0.22

	Sign	Exponent	Mantissa
Value:	+1	2^{-1}	1.9999979734420776
Encoded as:	0	126	8388591
Binary:	☐	☐☑☑☑☑☑☑☐	☑☑☑☑☑☑☑☑☑☑☑☑☑☑☑☑☑☑☐☑☑☑☑

You entered	0.999999
Value actually stored in float:	0.999998986721038818359375
Error due to conversion:	-1.3278961181640625E-8
Binary Representation	00111111011111111111111111101111
Hexadecimal Representation	0x3f7fffef

+1   -1

http://www.h-schmidt.net/FloatConverter/IEEE754de.html

`x1=0.999999 → x1_32= 0.999998986721038…`

Round-off Errors and Computer Arithmetic

- Compute `(x0-x1) * 1 000 000`
- Assume: `float x0,x1 // 32 bit values`

- **`x0= 1,000,000.0;      x1= 999,999.0`**

→ **`(x0-x1)*1,000,000 = 1,000,000 //exact arithmetic`**

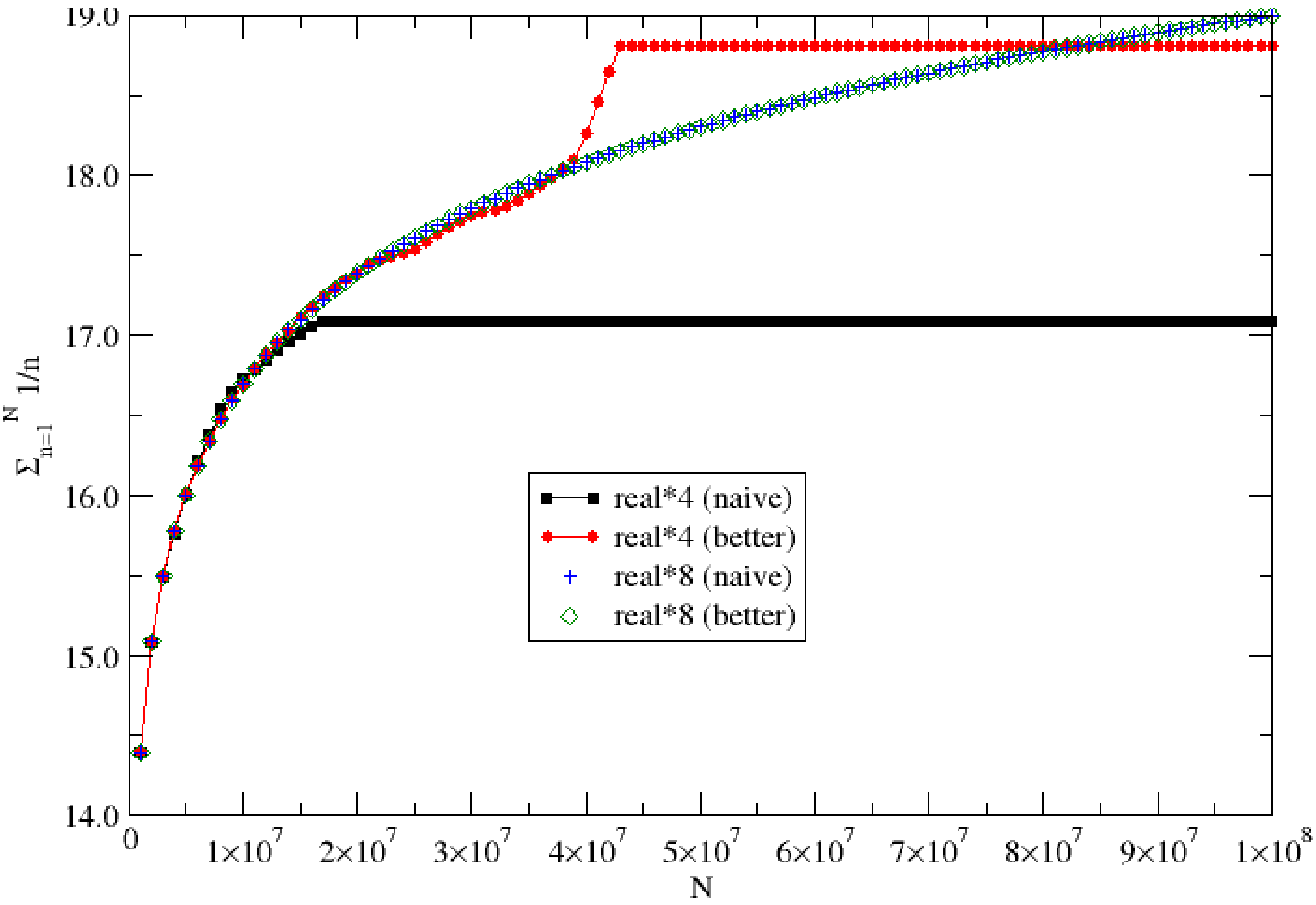
```
========================================
 x0,x1 - 32 Bit float
 x0                  =    1000000.
 x1                  =    999999.0
(x0-x1) *1 000 000 =    1000000.
========================================
```

- **`x0= 1.000000;   x1= 0.999999`**

→ **`(x0-x1)*1,000,000 = 1.000000 //exact arithmetic`**

```
========================================
 x0,x1 - 32 Bit float
 x0                  =    1.000000
 x1                  =   0.9999990
(x0-x1) *1 000 000 =    1.013279
========================================
```

Round-off Errors and Computer Arithmetic
Harmonic series: Divergence ?!



Accumulated relative error after a large series of FP (floating point) operations assuming a simple random walk theory for error propagation.

After N successive arithmetic operations (each having a relative error ε_m), the relative error becomes:

$$\varepsilon_{RE} \sim N^{1/2} \varepsilon_m$$

		Accumulated relative error ε_{RE}	
	#FP operations	32-Bit	64-Bit
MFlop	10^6	10^{-3}	10^{-12}
TFlop	10^{12}	1	10^{-9}
10 PFlop	10^{16}	100	10^{-7}

Stability of Computations

- A computation $M(x_1,x_2,\ldots,x_n)$ with input data $(x_1,x_2,\ldots,x_n)$ is called stable if small errors with respect to the input data ($< \varepsilon$) lead to small errors of the output data ($< c^* \varepsilon$)

- Let $E_0 > 0$ denote an initial error and E_n represent the magnitude of an error after n subsequent operations. Let C>1 be a constant independent of n. Then,

  1) If $E_n \approx C\, n\, E_0$ the error growth is called linear
  2) If $E_n \approx C^n\, E_0$ the error growth is called exponential