Universidade Federal de Alagoas
Instituto de Computação

Métodos Numéricos

Erros numéricos

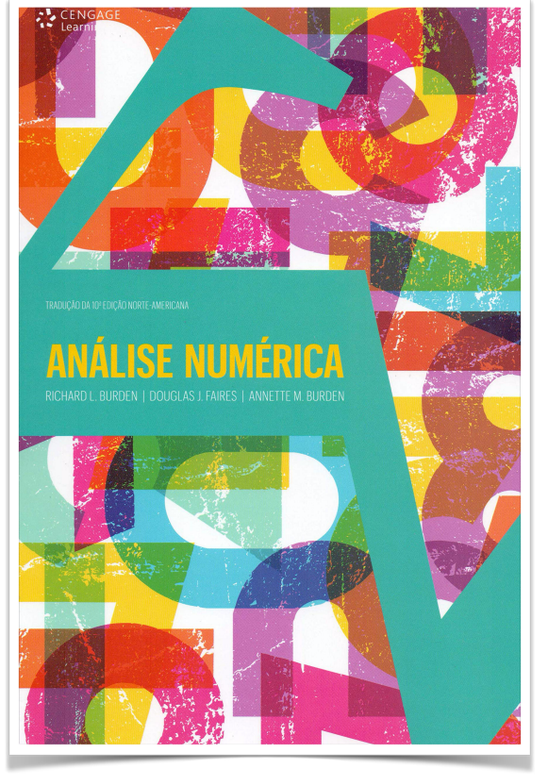
Prof. Thales Vieira

thales@ic.ufal.br

https://ic.ufal.br/professor/thales/

Referência

- Burden, R.L. and Faires, J.D., 2008. Análise numérica. Cengage Learning.



Seção 1.2

IEEE Floating Point Numbers

Long real (double precision) format

- Widely adopted standard
- Default data type in MATLAB, "`double`" in C
- Base 2, 1 sign bit, 11 exponent bits, 52 significand bits:

x	xxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
s	c	f

- Represented number:

$$(-1)^s 2^{c-1023}(1+f)$$

Long real (double precision) format

- Widely adopted standard
- Default data type in MATLAB, "`double`" in C
- Base 2, 1 sign bit, 11 exponent bits, 52 significand bits:

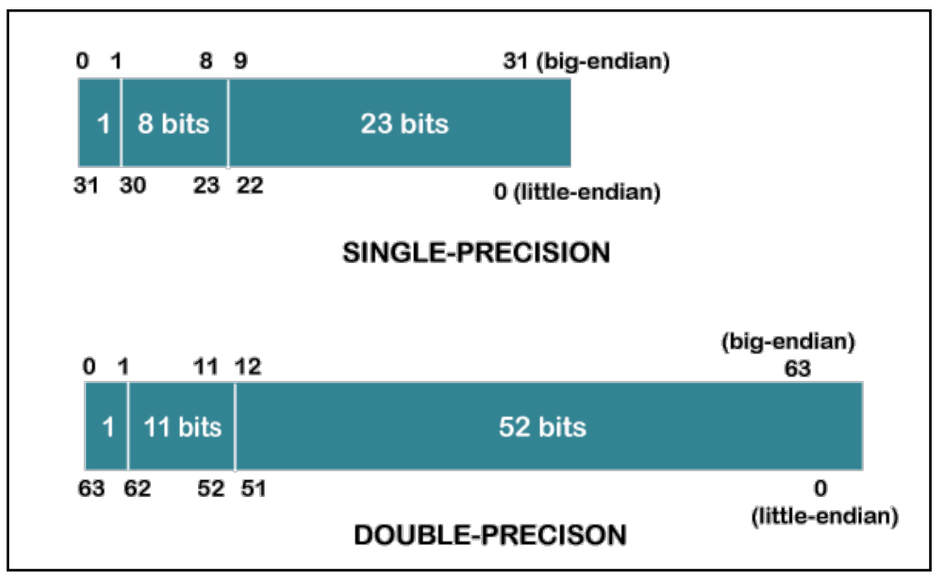
x	xxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
s	c	f

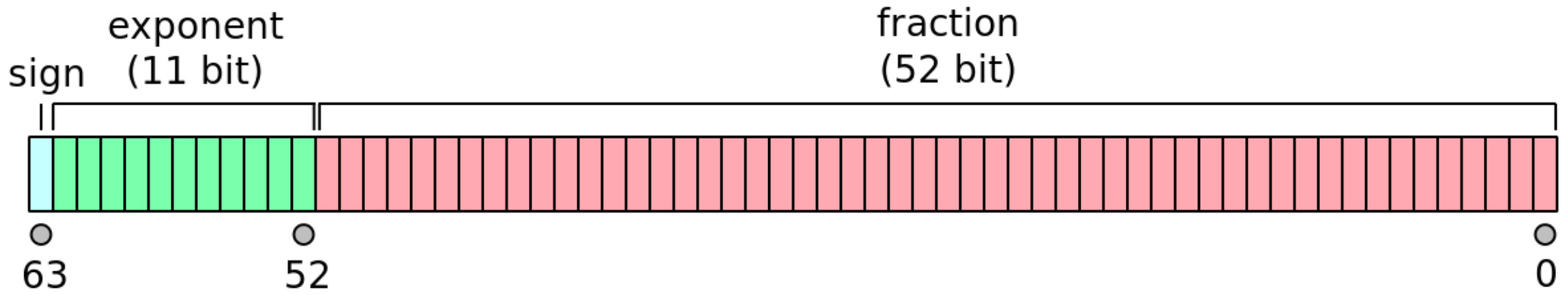
- Represented number:

$$(-1)^s 2^{c-1023}(1+f)$$

- A IEEE define três formatos.

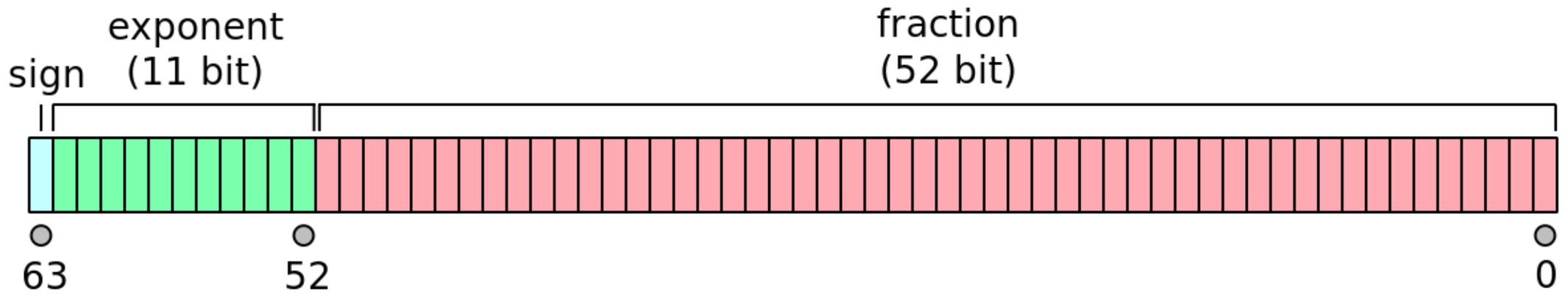
	Simples (32 bits)	Duplo (64 bits)	Quádruplo (128 bits)
Sinal (S)	1 bit	1 bit	1 bit
Expoente (E)	8 bits	11 bits	15 bits
Mantissa (M)	23 bits	52 bits	112 bits





$$(-1)^{\text{sign}}(1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

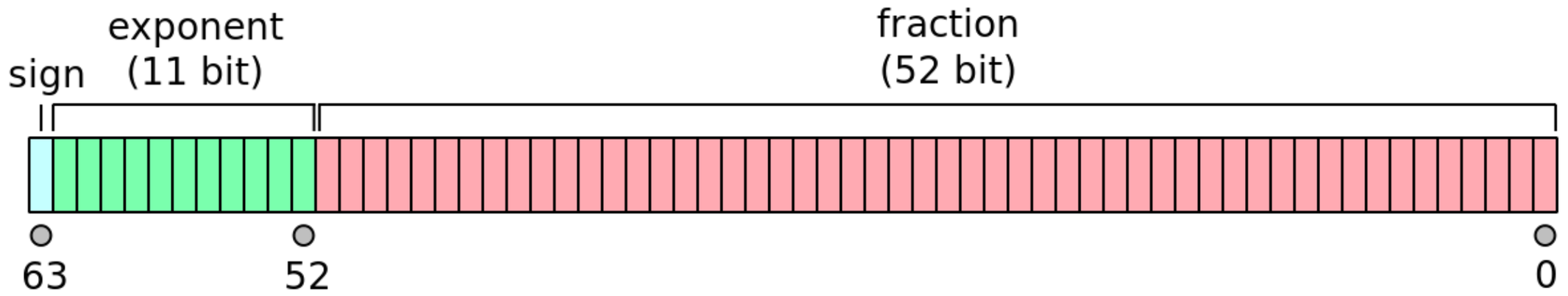


$$(-1)^{\text{sign}}(1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

Considere o seguinte número, representado em 64 bits:

0 10000000011 1011100100010000000000000000000000000000000000000000.



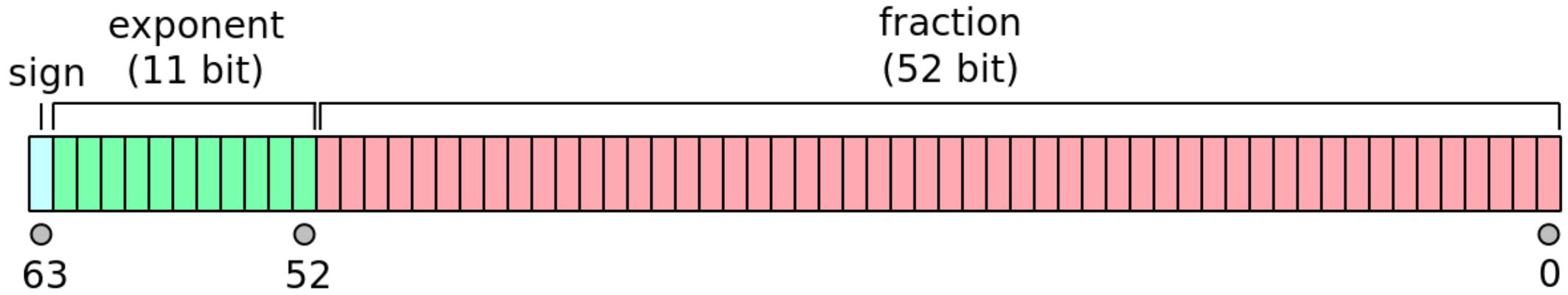
$$(-1)^{\text{sign}}(1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

Considere o seguinte número, representado em 64 bits:

$$0\ 10000000011\ 1011100100010000000000000000000000000000000000000000.$$

$$e = 1\cdot 2^{10} + 0\cdot 2^{9} + \cdots + 0\cdot 2^{2} + 1\cdot 2^{1} + 1\cdot 2^{0} = 1024 + 2 + 1 = 1027.$$



$$(-1)^{\text{sign}}(1.b_{51}b_{50}\ldots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

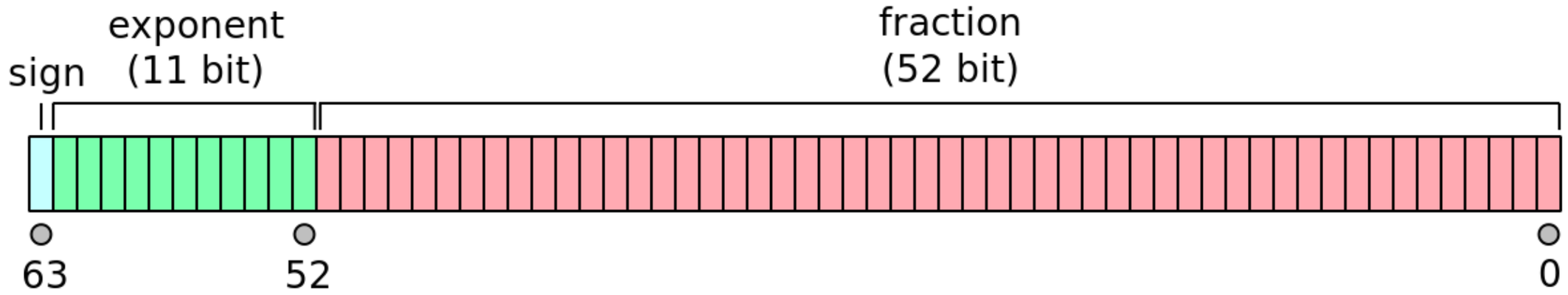
f

Considere o seguinte número, representado em 64 bits:

0 10000000011 1011100100010000000000000000000000000000000000000000.

$$e = 1\cdot 2^{10} + 0\cdot 2^{9} + \cdots + 0\cdot 2^{2} + 1\cdot 2^{1} + 1\cdot 2^{0} = 1024 + 2 + 1 = 1027.$$

$$f = 1\cdot\left(\frac{1}{2}\right)^{1} + 1\cdot\left(\frac{1}{2}\right)^{3} + 1\cdot\left(\frac{1}{2}\right)^{4} + 1\cdot\left(\frac{1}{2}\right)^{5} + 1\cdot\left(\frac{1}{2}\right)^{8} + 1\cdot\left(\frac{1}{2}\right)^{12}$$



$$(-1)^{\text{sign}}(1.b_{51}b_{50}\ldots b_0)_2 \times 2^{e-1023}$$

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

f

Considere o seguinte número, representado em 64 bits:

0 10000000011 1011100100010000000000000000000000000000000000000000.

$$e = 1\cdot 2^{10} + 0\cdot 2^9 + \cdots + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 1024 + 2 + 1 = 1027.$$

$$f = 1\cdot\left(\frac{1}{2}\right)^1 + 1\cdot\left(\frac{1}{2}\right)^3 + 1\cdot\left(\frac{1}{2}\right)^4 + 1\cdot\left(\frac{1}{2}\right)^5 + 1\cdot\left(\frac{1}{2}\right)^8 + 1\cdot\left(\frac{1}{2}\right)^{12}$$

$$(-1)^s 2^{c-1023}(1+f) = (-1)^0\cdot 2^{1027-1023}\left(1+\left(\frac{1}{2}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{256}+\frac{1}{4096}\right)\right) = 27.56640625$$

Decimal Floating-Point Numbers

Base-10 Floating-Point

- For simplicity we study k-digit *decimal machine numbers*:

$$\pm 0.d_1 d_2 \ldots d_k \times 10^n, \quad 1 \leq d_1 \leq 9, \quad 0 \leq d_i \leq 9$$

- Any positive number within the range can be written:

$$y = 0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \ldots \times 10^n$$

- Two ways to represent y with k digits:
  - *Chopping*: Chop off after k digits:

$$fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n$$

  - *Rounding*: Add $5 \times 10^{n-(k+1)}$ and chop:

$$fl(y) = 0.\delta_1 \delta_2 \ldots \delta_k \times 10^n$$

Definition

If p^* is an approximation to p, the *absolute error* is $|p - p^*|$, and the *relative error* is $|p - p^*|/|p|$, provided that $p \neq 0$.

Finite-Digit Arithmetic

- Machine addition, subtraction, multiplication, and division:

$$x \oplus y = fl(fl(x) + fl(y)), \quad x \otimes y = fl(fl(x) \times fl(y))$$
$$x \ominus y = fl(fl(x) - fl(y)), \quad x \oslash y = fl(fl(x) \div fl(y))$$

- "Round input, perform exact arithmetic, round the result"

Problema da subtração

Cancelation

- Common problem: Subtraction of nearly equal numbers:

$$fl(x) = 0.d_1 d_2 \dots d_p \alpha_{p+1} \alpha_{p+2} \dots \alpha_k \times 10^n$$
$$fl(y) = 0.d_1 d_2 \dots d_p \beta_{p+1} \beta_{p+2} \dots \beta_k \times 10^n$$

gives fewer digits of significance:

$$fl(fl(x) - fl(y)) = 0.\sigma_{p+1} \sigma_{p+2} \dots \sigma_k \times 10^{n-p}$$

Problema da divisão

If a finite-digit representation or calculation introduces an error, further enlargement of the error occurs when dividing by a number with small magnitude (or, equivalently, when multiplying by a number with large magnitude)

Suppose, for example, that the number z has the finite-digit approximation $z + \delta$, where the error δ is introduced by representation or by previous calculation.

Now divide by $\varepsilon = 10^{-n}$, where $n > 0$. Then

$$\frac{z}{\varepsilon} \approx fl\left(\frac{fl(z)}{fl(\varepsilon)}\right) = (z + \delta) \times 10^n.$$

The absolute error in this approximation, $|\delta| \times 10^n$, is the original absolute error, $|\delta|$, multiplied by the factor 10^n.

Problema da divisão

If a finite-digit representation or calculation introduces an error, further enlargement of the error occurs when dividing by a number with small magnitude (or, equivalently, when multiplying by a number with large magnitude)

Suppose, for example, that the number z has the finite-digit approximation $z + \delta$, where the error δ is introduced by representation or by previous calculation.

Now divide by $\varepsilon = 10^{-n}$, where $n > 0$. Then

Problema da divisão

If a finite-digit representation or calculation introduces an error, further enlargement of the error occurs when dividing by a number with small magnitude (or, equivalently, when multiplying by a number with large magnitude)

Suppose, for example, that the number z has the finite-digit approximation $z + \delta$, where the error δ is introduced by representation or by previous calculation.

Now divide by $\varepsilon = 10^{-n}$, where $n > 0$. Then

$$\frac{z}{\varepsilon} \approx fl\left(\frac{fl(z)}{fl(\varepsilon)}\right) = (z + \delta) \times 10^n.$$

The absolute error in this approximation, $|\delta| \times 10^n$, is the original absolute error, $|\delta|$, multiplied by the factor 10^n.