

# Aproximações e Erros

Andréa M. P. Valli e Lucia Catabriga

Laboratório de Otimização e Modelagem Computacional  
Departamento de Informática  
Universidade Federal do Espírito Santo - UFES, Vitória, ES, Brasil

# Aproximações e Erros

- 1 Introdução
- 2 Tipos de Erros
- 3 Erros de Ponto Flutuante
- 4 Formato proposto pela IEEE
- 5 Bibliografia

- **Programação e Software:** neste curso podem ser utilizadas as linguagens estruturadas **C** ou **Fortran** para a implementação dos métodos estudados ou software **octave** (ou **MATLAB**) para a implementação e resolução de problemas em engenharia.

- **Programação e Software:** neste curso podem ser utilizadas as linguagens estruturadas **C** ou **Fortran** para a implementação dos métodos estudados ou software **octave** (ou **MATLAB**) para a implementação e resolução de problemas em engenharia.
- **Custo Computacional:** **tempo computacional** e **memória**. O objetivo de uma implementação eficiente de um algoritmo numérico é tentar **reduzir**, sempre que for possível, o tempo computacional (número de operações de ponto flutuante) e a utilização da memória. Em geral, isto define a escolha do algoritmo numérico a ser implementado.

- **Algarismos Significativos** [2]: O conceito de um algarismo significativo foi desenvolvido para designar formalmente a confiabilidade de um valor numérico. Os algarismos significativos de um número são aqueles que podem ser usados com confiança. Eles correspondem ao número de algarismos corretos mais um algarismo estimado.

### Exemplos:

- 1 51.5 tem três alg. signif.;

- **Algarismos Significativos** [2]: O conceito de um algarismo significativo foi desenvolvido para designar formalmente a confiabilidade de um valor numérico. Os algarismos significativos de um número são aqueles que podem ser usados com confiança. Eles correspondem ao número de algarismos corretos mais um algarismo estimado.

### Exemplos:

- 1 51.5 tem três alg. signif.;
- 2 0.0001163, 0.001163 e 0.01163 têm quatro alg. signif.;

- **Algarismos Significativos** [2]: O conceito de um algarismo significativo foi desenvolvido para designar formalmente a confiabilidade de um valor numérico. Os algarismos significativos de um número são aqueles que podem ser usados com confiança. Eles correspondem ao número de algarismos corretos mais um algarismo estimado.

### Exemplos:

- 1 51.5 tem três alg. signif.;
- 2 0.0001163, 0.001163 e 0.01163 têm quatro alg. signif.;
- 3 3.100 pode ter dois, três ou quatro alg. signif., dependendo se os zeros serem conhecidos com confiança;

- **Algarismos Significativos** [2]: O conceito de um algarismo significativo foi desenvolvido para designar formalmente a confiabilidade de um valor numérico. Os algarismos significativos de um número são aqueles que podem ser usados com confiança. Eles correspondem ao número de algarismos corretos mais um algarismo estimado.

### Exemplos:

- 1 51.5 tem três alg. signif.;
- 2 0.0001163, 0.001163 e 0.01163 têm quatro alg. signif.;
- 3 3.100 pode ter dois, três ou quatro alg. signif., dependendo se os zeros serem conhecidos com confiança;
- 4  $4.69 \times 10^4$ ,  $4.690 \times 10^4$ ,  $4.6900 \times 10^4$  designam que o número é conhecido com três, quatro ou cinco algarismos significativos, respectivamente.



O conceito de **algoritmos significativos** tem duas implicações importantes [2]:

- 1 Como os métodos numéricos fornecem **resultados aproximados**, é necessário especificar quantos algoritmos significativos a aproximação é considerada aceitável.

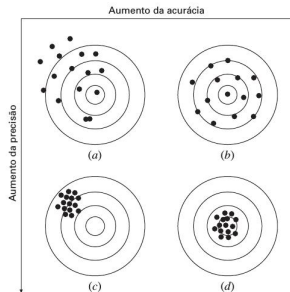
O conceito de **algarismos significativos** tem duas implicações importantes [2]:

- 1 Como os métodos numéricos fornecem **resultados aproximados**, é necessário especificar quantos algarismos significativos a aproximação é considerada aceitável.
- 2 Como os computadores mantêm apenas um **número finito** de algarismos significativos, números como  $\pi$  ou  $\sqrt{7}$  jamais podem ser representados exatamente. A omissão dos algarismos significativos remanescentes é chamada de **erro de arredondamento**.

- **Acurácia e Precisão** [2]: Os erros associados tanto aos cálculos quanto às medidas podem ser caracterizados com relação a sua acurácia e precisão. A **acurácia** se refere a quão próximo o valor calculado ou medido está do valor verdadeiro. A **precisão** se refere a quão próximos os valores individuais calculados ou medidos estão uns dos outros.

**FIGURA 3.2**

Um exemplo do tiro ao alvo ilustrando os conceitos de acurácia e precisão. (a) Inacurado e impreciso; (b) acurados e impreciso; (c) inacurado e preciso; (d) acurado e preciso.



Tipos de Erros que aparecem na modelagem numérica:

- 1 Erros na modelagem: erros obtidos pelo uso de dados experimentais errados ou pela própria representação matemática errada de um modelo físico.

Tipos de Erros que aparecem na modelagem numérica:

- 1 **Erros na modelagem:** erros obtidos pelo uso de dados experimentais errados ou pela própria representação matemática errada de um modelo físico.
- 2 **Erros de truncamento:** é o erro devido à aproximação de uma fórmula por outra, ou seja, quando são feitas aproximações para representar procedimentos matemáticos exatos.

Exemplo:  $\text{sen}(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}$

Tipos de Erros que aparecem na modelagem numérica:

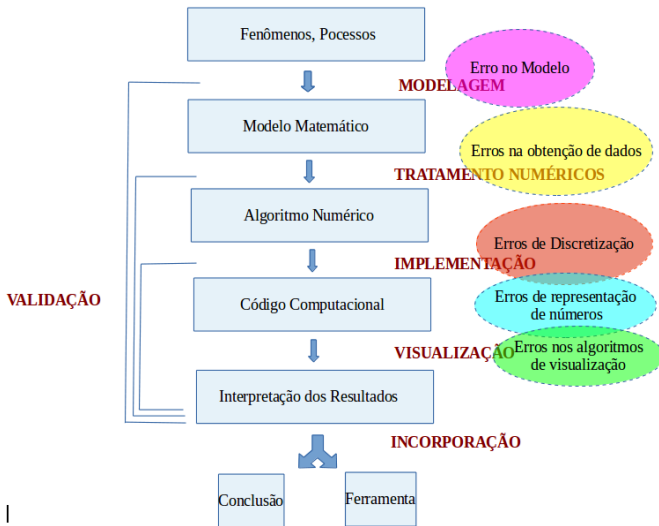
- 1 **Erros na modelagem**: erros obtidos pelo uso de dados experimentais errados ou pela própria representação matemática errada de um modelo físico.
- 2 **Erros de truncamento**: é o erro devido à aproximação de uma fórmula por outra, ou seja, quando são feitas aproximações para representar procedimentos matemáticos exatos.

Exemplo:  $\text{sen}(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}$

$\hat{u}_h$  : solução aproximada,  $u$  : solução exata

$$|u - \hat{u}_h| \leq \epsilon \text{ (Erro Absoluto)} \quad \text{e} \quad \frac{|u - \hat{u}_h|}{|u|} \leq \epsilon \text{ (Erro Relativo)}$$

- 3 **Erros de arredondamento** (ou de **ponto flutuante**): é o erro causado pela imperfeição na representação de um número, ou seja, quando uma quantidade limitada de algarismos significativos são usados para representar números.



Aritmética de ponto flutuante é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).



**Aritmética de ponto flutuante** é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).

**Definição:** um número  $x \in \mathfrak{R}$  é um **número de ponto flutuante** se

$$x = \pm .d_1 d_2 \cdots d_p \times B^e$$

where

$B$  = valor da **base** (geralmente 2,8,10 ou 16)

**Aritmética de ponto flutuante** é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).

**Definição:** um número  $x \in \mathfrak{R}$  é um **número de ponto flutuante** se

$$x = \pm .d_1 d_2 \cdots d_p \times B^e$$

where

$B$  = valor da **base** (geralmente 2,8,10 ou 16)

$d_i$ 's = **dígitos** da parte fracionária (ou **mantissa**)

Aritmética de ponto flutuante é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).

**Definição:** um número  $x \in \mathfrak{R}$  é um número de ponto flutuante se

$$x = \pm .d_1 d_2 \cdots d_p \times B^e$$

where

- $B$  = valor da base (geralmente 2,8,10 ou 16)
- $d_i$ 's = dígitos da parte fracionária (ou mantissa)
- $p$  = número de dígitos na mantissa,  
 $d_1 \neq 0, 0 \leq d_i \leq B - 1, i = 2, \cdots, p$

Aritmética de ponto flutuante é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).

**Definição:** um número  $x \in \mathfrak{R}$  é um número de ponto flutuante se

$$x = \pm .d_1 d_2 \cdots d_p \times B^e$$

where

$B$  = valor da base (geralmente 2,8,10 ou 16)

$d_i$ 's = dígitos da parte fracionária (ou mantissa)

$p$  = número de dígitos na mantissa,

$$d_1 \neq 0, 0 \leq d_i \leq B - 1, i = 2, \cdots, p$$

$e$  = expoente inteiro

Aritmética de ponto flutuante é a aritmética usada nos computadores, ou seja, como os números são representados, armazenados e operados em um computador (ou um sistema de ponto flutuante).

**Definição:** um número  $x \in \mathfrak{R}$  é um número de ponto flutuante se

$$x = \pm .d_1 d_2 \cdots d_p \times B^e$$

where

$B$  = valor da base (geralmente 2,8,10 ou 16)

$d_i$ 's = dígitos da parte fracionária (ou mantissa)

$p$  = número de dígitos na mantissa,

$$d_1 \neq 0, 0 \leq d_i \leq B - 1, i = 2, \cdots, p$$

$e$  = expoente inteiro

$\pm$  = sinal do número

Um **sistema de ponto flutuante** pode ser representado por

$$F = F(B, p, e_1, e_2),$$

onde  $e_1, e_2 =$  menor e maior expoente.

Um sistema de ponto flutuante pode ser representado por

$$F = F(B, p, e_1, e_2),$$

onde  $e_1, e_2$  = menor e maior expoente.

A quantidade de elementos no sistema de ponto flutuante  $F(B, p, e_1, e_2)$  pode ser calculada e é dada por:

$$\#F = 2(B - 1)(B^{p-1})(e_2 - e_1 + 1) + 1$$

Um sistema de ponto flutuante pode ser representado por

$$F = F(B, p, e_1, e_2),$$

onde  $e_1, e_2$  = menor e maior expoente.

A quantidade de elementos no sistema de ponto flutuante  $F(B, p, e_1, e_2)$  pode ser calculada e é dada por:

$$\#F = 2(B - 1)(B^{p-1})(e_2 - e_1 + 1) + 1$$

Observação: a quantidade de números que um sistema de ponto flutuante (ou um computador) consegue representar é sempre finita.



Exemplo:  $F(10, 5, -2, 3)$ ,

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

+ .45327  $\times 10^2$       representação de ponto flutuante

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

+ .45327  $\times 10^2$       representação de ponto flutuante

178.235      representação de ponto fixo

+ .17824  $\times 10^3$       representação de ponto flutuante, arredondamento

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

+ .45327  $\times 10^2$       representação de ponto flutuante

178.235      representação de ponto fixo

+ .17824  $\times 10^3$       representação de ponto flutuante, arredondamento

+ .17823  $\times 10^3$       representação de ponto flutuante, truncamento

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

+ .45327  $\times 10^2$       representação de ponto flutuante

178.235      representação de ponto fixo

+ .17824  $\times 10^3$       representação de ponto flutuante, arredondamento

+ .17823  $\times 10^3$       representação de ponto flutuante, truncamento

$$|\text{menor número}| = +.10000 \times 10^{-2} = 0.001 = 10^{-3}$$

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327      representação de ponto fixo

+ .45327  $\times 10^2$       representação de ponto flutuante

178.235      representação de ponto fixo

+ .17824  $\times 10^3$       representação de ponto flutuante, arredondamento

+ .17823  $\times 10^3$       representação de ponto flutuante, truncamento

$$|\text{menor número}| = +.10000 \times 10^{-2} = 0.001 = 10^{-3}$$

$$|\text{maior número}| = +.99999 \times 10^3 = 999.99$$



Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327	representação de ponto fixo
+ .45327 $\times 10^2$	representação de ponto flutuante
178.235	representação de ponto fixo
+ .17824 $\times 10^3$	representação de ponto flutuante, arredondamento
+ .17823 $\times 10^3$	representação de ponto flutuante, truncamento

$$|\text{menor número}| = +.10000 \times 10^{-2} = 0.001 = 10^{-3}$$

$$|\text{maior número}| = +.99999 \times 10^3 = 999.99$$

$$\text{região de overflow} = (-\infty, -999.99) \cup (999.99, +\infty)$$

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327	representação de ponto fixo
+ .45327 $\times 10^2$	representação de ponto flutuante
178.235	representação de ponto fixo
+ .17824 $\times 10^3$	representação de ponto flutuante, arredondamento
+ .17823 $\times 10^3$	representação de ponto flutuante, truncamento

$$|\text{menor número}| = +.10000 \times 10^{-2} = 0.001 = 10^{-3}$$

$$|\text{maior número}| = +.99999 \times 10^3 = 999.99$$

$$\text{região de overflow} = (-\infty, -999.99) \cup (999.99, +\infty)$$

$$\text{região de underflow} = (-10^{-3}, 0) \cup (0, +10^{-3})$$

Exemplo:  $F(10, 5, -2, 3)$ ,

$\Rightarrow B = 10, p = 5, e_1 = -2, e_2 = 3, \Rightarrow e = -2, -1, 0, 1, 2, 3$

45.327	representação de ponto fixo
+ .45327 $\times 10^2$	representação de ponto flutuante
178.235	representação de ponto fixo
+ .17824 $\times 10^3$	representação de ponto flutuante, arredondamento
+ .17823 $\times 10^3$	representação de ponto flutuante, truncamento

$$|\text{menor número}| = +.10000 \times 10^{-2} = 0.001 = 10^{-3}$$

$$|\text{maior número}| = +.99999 \times 10^3 = 999.99$$

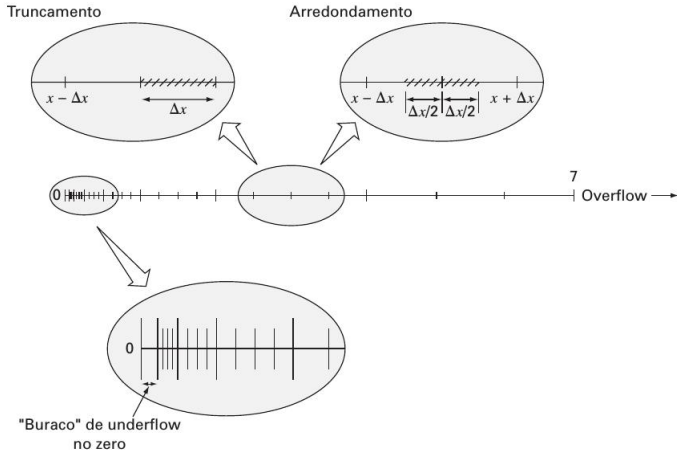
$$\text{região de overflow} = (-\infty, -999.99) \cup (999.99, +\infty)$$

$$\text{região de underflow} = (-10^{-3}, 0) \cup (0, +10^{-3})$$

$$\#F = 2(10 - 1)(10^{5-1})(3 - (-2) + 1) + 1$$

$$= 1,080,001$$

Sistema de Ponto Flutuante [2]: apenas os números positivos estão mostrados; existe um conjunto idêntico na direção negativa.



Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

- **Intervalo limitado**:  $(-maior, -menor) \cup \{0\} \cup (menor, maior)$ .

Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

- **Intervalo limitado**:  $(-maior, -menor) \cup \{0\} \cup (menor, maior)$ .
- Região de **underflow** =  $(-menor, 0) \cup (0, +menor)$  e região de **overflow** =  $(-\infty, -maior) \cup (maior, +\infty)$ . Mensagem de erro = **NAN** (*not a number*).

Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

- **Intervalo limitado**:  $(-\text{maior}, -\text{menor}) \cup \{0\} \cup (\text{menor}, \text{maior})$ .
- Região de **underflow** =  $(-\text{menor}, 0) \cup (0, +\text{menor})$  e região de **overflow** =  $(-\infty, -\text{maior}) \cup (\text{maior}, +\infty)$ . Mensagem de erro = **NAN** (*not a number*).
- *Existe apenas um número finito de valores que podem ser representados dentro do intervalo.*



Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

- **Intervalo limitado**:  $(-\text{maior}, -\text{menor}) \cup \{0\} \cup (\text{menor}, \text{maior})$ .
- Região de **underflow** =  $(-\text{menor}, 0) \cup (0, +\text{menor})$  e região de **overflow** =  $(-\infty, -\text{maior}) \cup (\text{maior}, +\infty)$ . Mensagem de erro = **NAN** (*not a number*).
- *Existe apenas um número finito de valores que podem ser representados dentro do intervalo.*
- **Fontes de erros**: arredondamento e truncamento, operações de ponto flutuante. No curso, usaremos **arredondamento**.

Seja **maior** = maior valor positivo e **menor** = menor valor positivo de um sistema de ponto flutuante  $F(B, p, e_1, e_2)$ .

- **Intervalo limitado**:  $(-\text{maior}, -\text{menor}) \cup \{0\} \cup (\text{menor}, \text{maior})$ .
- Região de **underflow** =  $(-\text{menor}, 0) \cup (0, +\text{menor})$  e região de **overflow** =  $(-\infty, -\text{maior}) \cup (\text{maior}, +\infty)$ . Mensagem de erro = **NAN** (*not a number*).
- *Existe apenas um **número finito** de valores que podem ser representados dentro do intervalo.*
- **Fontes de erros**: arredondamento e truncamento, operações de ponto flutuante. No curso, usaremos **arredondamento**.
- *Para melhorar a **precisão** precisamos aumentar o número de algarismos significativos, ou seja, o número de algarismos na matissa.*

## Mudança de Base:

- 2 → 10:

$$(11.001)_2 =$$

## Mudança de Base:

- 2 → 10:

$$(11.001)_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3}$$
$$=$$

## Mudança de Base:

- 2 → 10:

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8} \\ (0.1101)_2 &= \end{aligned}$$

## Mudança de Base:

- 2 → 10:

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \end{aligned}$$

## Mudança de Base:

- 2 → 10:

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}\end{aligned}$$

$$(0.1 \times 2^{-3})_2 =$$

## Mudança de Base:

- 2  $\rightarrow$  10:

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}\end{aligned}$$

$$(0.1 \times 2^{-3})_2 = (0.0001)_2 =$$



## Mudança de Base:

- $2 \rightarrow 10$ :

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}\end{aligned}$$

$$(0.1 \times 2^{-3})_2 = (0.0001)_2 = 1 \times 2^{-4} = \frac{1}{16}$$

$$(1.11 \times 2^2)_2 =$$

## Mudança de Base:

- 2  $\rightarrow$  10:

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}\end{aligned}$$

$$(0.1 \times 2^{-3})_2 = (0.0001)_2 = 1 \times 2^{-4} = \frac{1}{16}$$

$$(1.11 \times 2^2)_2 = (111.)_2 =$$

## Mudança de Base:

- $2 \rightarrow 10$ :

$$\begin{aligned}(11.001)_2 &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^{-3} \\ &= 1 + 2 + \frac{1}{8} = \frac{25}{8}\end{aligned}$$

$$\begin{aligned}(0.1101)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}\end{aligned}$$

$$(0.1 \times 2^{-3})_2 = (0.0001)_2 = 1 \times 2^{-4} = \frac{1}{16}$$

$$\begin{aligned}(1.11 \times 2^2)_2 &= (111.)_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 \\ &= 1 + 2 + 4 = 7\end{aligned}$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 =$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 =$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

- 10  $\rightarrow$  2:

Exemplo: 21.78125 = (10101.11001)<sub>2</sub>

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 =$$



- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 =$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 = 1 \times 2 + 0$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 = 1 \times 2 + 0$$

$$\rightarrow (10101.)_2$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 = 1 \times 2 + 0$$

$$\rightarrow (10101.)_2$$

verificando

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 = 1 \times 2 + 0$$

$$\rightarrow (10101.)_2$$

verificando

$$= 1 \times 2^0 + 1 \times 2^2 + 1 \times 2^4$$

- $10 \rightarrow 2$ :

Exemplo:  $21.78125 = (10101.11001)_2$

$$21/2 = 10 \times 2 + 1$$

$$10/2 = 5 \times 2 + 0$$

$$5/2 = 2 \times 2 + 1$$

$$2/2 = 1 \times 2 + 0$$

$$\rightarrow (10101.)_2$$

verificando

$$= 1 \times 2^0 + 1 \times 2^2 + 1 \times 2^4$$

$$= 1 + 4 + 16 = 21$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 =$$



- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 =$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 =$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 =$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 =$$



- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

verificando

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

verificando

$$= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5}$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

verificando

$$= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5}$$

$$= \frac{1}{2} + \frac{1}{4} + \frac{1}{32} = \frac{25}{32} = .78125$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

verificando

$$= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5}$$

$$= \frac{1}{2} + \frac{1}{4} + \frac{1}{32} = \frac{25}{32} = .78125$$

$$0.6 =$$

- $10 \rightarrow 2$ :

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

$$0.78125 \times 2 = 1.56250$$

$$0.56250 \times 2 = 1.12500$$

$$0.12500 \times 2 = 0.25000$$

$$0.25000 \times 2 = 0.50000$$

$$0.50000 \times 2 = 1.00000$$

$$\rightarrow (.11001)_2$$

verificando

$$= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5}$$

$$= \frac{1}{2} + \frac{1}{4} + \frac{1}{32} = \frac{25}{32} = .78125$$

$$0.6 = (0.1001100110011 \dots)_2$$

Exemplo de um computador (sistema de ponto flutuante):

$F(2, 2, -1, 2)$ ,



Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$
$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$
$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$
$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

$$= \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3$$

$$|\text{menor número}| = (.10 \times 2^{-1})_2 = (0.01)_2 = 1 \times 2^{-2} = \frac{1}{4}$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

$$= \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3$$

$$|\text{menor número}| = (.10 \times 2^{-1})_2 = (0.01)_2 = 1 \times 2^{-2} = \frac{1}{4}$$

$$|\text{maior número}| = (.11 \times 2^2)_2 = (11.)_2 = 1 \times 2^0 + 1 \times 2^1 = 3$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

$$= \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3$$

$$|\text{menor número}| = (.10 \times 2^{-1})_2 = (0.01)_2 = 1 \times 2^{-2} = \frac{1}{4}$$

$$|\text{maior número}| = (.11 \times 2^2)_2 = (11.)_2 = 1 \times 2^0 + 1 \times 2^1 = 3$$

$$\text{região de overflow} = (-\infty, -3) \cup (3, +\infty)$$

Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

$$= \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3$$

$$|\text{menor número}| = (.10 \times 2^{-1})_2 = (0.01)_2 = 1 \times 2^{-2} = \frac{1}{4}$$

$$|\text{maior número}| = (.11 \times 2^2)_2 = (11.)_2 = 1 \times 2^0 + 1 \times 2^1 = 3$$

$$\text{região de overflow} = (-\infty, -3) \cup (3, +\infty)$$

$$\text{região de underflow} = \left(-\frac{1}{4}, 0\right) \cup \left(0, \frac{1}{4}\right)$$



Exemplo de um computador (sistema de ponto flutuante):

$$F(2, 2, -1, 2), \Rightarrow B = 2, p = 2, e_1 = -1, e_2 = 2 \Rightarrow e = -1, 0, 1, 2$$

$$(.10 \times 2^{-1})_2, (.10 \times 2^0)_2, (.10 \times 2^1)_2, (.10 \times 2^2)_2$$

$$(.11 \times 2^{-1})_2, (.11 \times 2^0)_2, (.11 \times 2^1)_2, (.11 \times 2^2)_2$$

$$= \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3$$

$$|\text{menor número}| = (.10 \times 2^{-1})_2 = (0.01)_2 = 1 \times 2^{-2} = \frac{1}{4}$$

$$|\text{maior número}| = (.11 \times 2^2)_2 = (11.)_2 = 1 \times 2^0 + 1 \times 2^1 = 3$$

$$\text{região de overflow} = (-\infty, -3) \cup (3, +\infty)$$

$$\text{região de underflow} = \left(-\frac{1}{4}, 0\right) \cup \left(0, \frac{1}{4}\right)$$

$$\begin{aligned} \#F &= 2(2-1)(2^{2-1})(2-(-1)+1)+1 \\ &= 2(1)(2)(4)+1=17 \end{aligned}$$

Erros de arredondamento (ou ponto flutuante):

$$0.6 = (0.1001100110011 \dots)_2$$

Erros de arredondamento (ou ponto flutuante):

$$0.6 = (0.1001100110011 \dots)_2$$
$$\frac{1}{4} + \frac{3}{2} = \frac{7}{4} = 1.75$$

Erros de arredondamento (ou ponto flutuante):

$$\begin{aligned}0.6 &= (0.1001100110011 \dots)_2 \\ \frac{1}{4} + \frac{3}{2} &= \frac{7}{4} = 1.75 \\ &= (0.10 \times 2^{-1})_2 + (0.11 \times 2^1)_2\end{aligned}$$

Erros de arredondamento (ou ponto flutuante):

$$\begin{aligned}0.6 &= (0.1001100110011 \dots)_2 \\ \frac{1}{4} + \frac{3}{2} &= \frac{7}{4} = 1.75 \\ &= (0.10 \times 2^{-1})_2 + (0.11 \times 2^1)_2 \\ &= (0.001 \times 2^1)_2 + (0.11 \times 2^1)_2\end{aligned}$$

Erros de arredondamento (ou ponto flutuante):

$$\begin{aligned}0.6 &= (0.1001100110011 \dots)_2 \\ \frac{1}{4} + \frac{3}{2} &= \frac{7}{4} = 1.75 \\ &= (0.10 \times 2^{-1})_2 + (0.11 \times 2^1)_2 \\ &= (0.001 \times 2^1)_2 + (0.11 \times 2^1)_2 \\ &= (0.111 \times 2^1)_2\end{aligned}$$

Erros de arredondamento (ou ponto flutuante):

$$\begin{aligned}0.6 &= (0.1001100110011 \dots)_2 \\ \frac{1}{4} + \frac{3}{2} &= \frac{7}{4} = 1.75 \\ &= (0.10 \times 2^{-1})_2 + (0.11 \times 2^1)_2 \\ &= (0.001 \times 2^1)_2 + (0.11 \times 2^1)_2 \\ &= (0.111 \times 2^1)_2 \\ &\rightarrow \text{número entre } (0.11 \times 2^1) = 1.5 \text{ e } (0.10 \times 2^2)_2 = 2 \\ &\rightarrow \text{Erro} = 0.25\end{aligned}$$

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 *bits* (*precisão simples*)



**Formato proposto pela IEEE** ( Institute of Electrical and Electronics Engineers) para um computador de 32 *bits* (*precisão simples*)

*Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real*

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

*Definição: a palavra ( $b_1, b_2, \dots, b_{32}$ ) pode ser interpretada como o número real*

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

*Observações:*

- Um (1) bit é reservado para o *signal*.

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Observações:

- Um (1) bit é reservado para o *signal*.
- Oito (8) bits são reservados para o *expoente*.

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Observações:

- Um (1) bit é reservado para o *signal*.
- Oito (8) bits são reservados para o *expoente*.  
→ como  $(11111111)_2 = 255, \Rightarrow 0 \leq e \leq 255$

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Observações:

- Um (1) bit é reservado para o *signal*.
- Oito (8) bits são reservados para o *expoente*.  
→ como  $(11111111)_2 = 255, \Rightarrow 0 \leq e \leq 255$   
→  $-127 \leq e - 127 \leq 128$

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Observações:

- Um (1) bit é reservado para o *signal*.
- Oito (8) bits são reservados para o *expoente*.
  - como  $(11111111)_2 = 255$ ,  $\Rightarrow 0 \leq e \leq 255$
  - $-127 \leq e - 127 \leq 128$
  - $-127, 128$  são reservados  $(0, \infty)$

Formato proposto pela IEEE ( Institute of Electrical and Electronics Engineers) para um computador de 32 bits (*precisão simples*)

Definição: a palavra  $(b_1, b_2, \dots, b_{32})$  pode ser interpretada como o número real

$$(-1)^{b_1} \times 2^{(b_2, b_3, \dots, b_9)} \times 2^{-127} \times (1.b_{10}b_{11} \dots b_{32})$$

Observações:

- Um (1) bit é reservado para o  *sinal*.
- Oito (8) bits são reservados para o  *expoente*.
  - como  $(11111111)_2 = 255$ ,  $\Rightarrow 0 \leq e \leq 255$
  - $-127 \leq e - 127 \leq 128$
  - -127, 128 são reservados  $(0, \infty)$
  - expoente máximo: 127
  - expoente mínimo: -126

- Vinte e três (23) *bits* são reservados para a *mantissa*. Como o primeiro bit não precisa ser armazenado porque é sempre 1



- Vinte e três (23) *bits* são reservados para a *mantissa*. Como o primeiro bit não precisa ser armazenado porque é sempre 1 ⇒ temos 24 dígitos na mantissa

- Vinte e três (23) *bits* são reservados para a *mantissa*. Como o primeiro bit não precisa ser armazenado porque é sempre 1  
⇒ temos 24 dígitos na mantissa  
→  $2^{-24} = 0.596 \times 10^{-7}$

- Vinte e três (23) *bits* são reservados para a *mantissa*. Como o primeiro bit não precisa ser armazenado porque é sempre 1  
⇒ temos 24 dígitos na mantissa  
→  $2^{-24} = 0.596 \times 10^{-7}$   
→ sete (7) dígitos decimais

- Vinte e três (23) *bits* são reservados para a *mantissa*. Como o primeiro bit não precisa ser armazenado porque é sempre 1
  - ⇒ temos 24 dígitos na mantissa
  - $2^{-24} = 0.596 \times 10^{-7}$
  - sete (7) dígitos decimais
  - no máximo 7 casas de precisão
- Precisão:
  - ⇒ simples (float): 7 dígitos significativos
  - ⇒ dupla (double): 16 dígitos significativos
  - ⇒ estendida (long double): 19 dígitos significativos

## Épsilon da máquina

O **épsilon de máquina**,  $\epsilon_{mach} = \epsilon$ , é definido de forma que  $1 + \epsilon$  seja o menor número representável maior que 1, isto é,  $1 + \epsilon$  é representável, mas não existem números representáveis em  $(1, 1 + \epsilon)$ .

$$1 \rightarrow (1.0000\ 0000 \dots 0000)_2 \times 2^0$$

$$\epsilon \rightarrow +(0.0000\ 0000 \dots 0001)_2 \times 2^0 = 2^{-52}$$

$$1 + \epsilon = (1.0000\ 0000 \dots 0001)_2 \times 2^0 \neq 1$$

$$\epsilon = 2^{-52}$$

No GNU Octave:

```
eps = 2.2204e-16
```

```
1 + 1e-16 == 1; ans = 1
```

```
1 + eps == 1; ans = 0
```

Exemplo: indicar como o número 21.78125 é armazenado em um computador de 32 *bits*

$$21.78125 = (10101.11001)_2 =$$

Exemplo: indicar como o número 21.78125 é armazenado em um computador de 32 *bits*

$$21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$$

Exemplo: indicar como o número 21.78125 é armazenado em um computador de 32 *bits*

$$\begin{aligned}21.78125 &= (10101.11001)_2 = +.1010111001 \times 2^5 \\ &= +1.010111001 \times 2^4\end{aligned}$$



Exemplo: indicar como o número 21.78125 é armazenado em um computador de 32 *bits*

$$\begin{aligned}21.78125 &= (10101.11001)_2 = +.1010111001 \times 2^5 \\ &= +1.010111001 \times 2^4 \\ e - 127 = 4 &\Rightarrow e = 131 = (10000011)_2\end{aligned}$$

Exemplo: indicar como o número 21.78125 é armazenado em um computador de 32 *bits*

$$\begin{aligned}21.78125 &= (10101.11001)_2 = +.1010111001 \times 2^5 \\ &= +1.010111001 \times 2^4 \\ e - 127 = 4 &\Rightarrow e = 131 = (10000011)_2 \\ &\Rightarrow [010000011010111001 \dots]\end{aligned}$$

Tabela: Formatos da IEEE 754-1985.

tipo	bits	intervalo	precisão
single precision	32	$\pm 1.18 \times 10^{-38}$ a $\pm 3.4 \times 10^{38}$	$\simeq 7$
double precision	64	$\pm 2.23 \times 10^{-308}$ a $\pm 1.80 \times 10^{308}$	$\simeq 16$



- Números com representação finita:
  - $2^2 = 4$  (comando no GNU Octave: `2 * 2 == 4`)
- Números sem representação finita:
  - $(\sqrt{3})^2 \neq 3$  (comando no GNU Octave `sqrt(3) * 2 == 3`)
- Maior e menor DOUBLE positivo:
  - `realmax; ans = 1.7977e+308`
  - `realmin; ans = 2.2251e-308`

## Indeterminações no GNU Octave

- $0/0$ ;  
warning: division by zero;  
ans = NaN
- $1/0$ ;  
warning: division by zero;  
ans = Inf
- $1/-0$ ;  
warning: division by zero;  
ans = -Inf
- $2*\text{realmax}$   
????
- $1/2^{9999}$   
????

## Bibliografia Básica

[1] Algoritmos Numéricos, Frederico F. Campos, Filho - 2ª Ed., Rio de Janeiro, LTC, 2007.

[2] Métodos Numéricos para Engenharia, Steven C. Chapa e Raymond P. Canale, Ed. McGraw-Hill, 5ª Ed., 2008.

[3] Cálculo Numérico - Aspectos Teóricos e Computacionais, Márcia A. G. Ruggiero e Vera Lúcia da Rocha Lopes, Ed. Pearson Education, 2ª Ed., 1996.

[4] [https://www.ufrgs.br/reatmat/CalculoNumerico/livro-oct/rdneadm-representacao\\_de\\_numeros\\_em\\_maquina.html](https://www.ufrgs.br/reatmat/CalculoNumerico/livro-oct/rdneadm-representacao_de_numeros_em_maquina.html)