UNIVERSIDADE FEDERAL DE ALAGOAS

UNIVERSIDADE FEDERAL DE ALAGOAS
CAMPUS SERTÃO
EIXO TECNOLOGIA

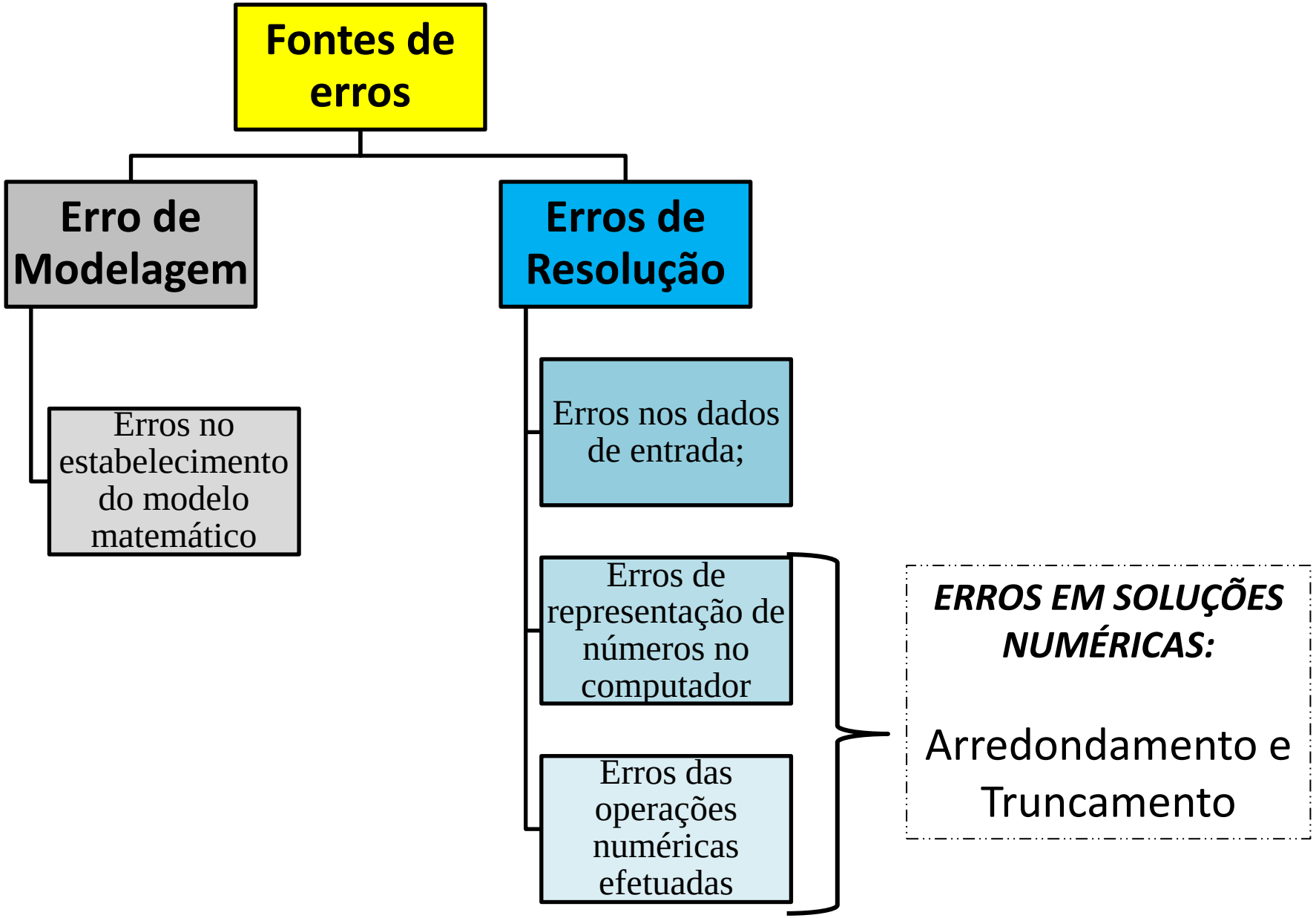
Cálculo Numérico

Prof. Dr. Alverlando Ricardo

Aula 3: PARTE I: ERROS EM SOLUÇÕES NUMÉRICAS

ERROS EM SOLUÇÕES NUMÉRICAS

Fontes de erros



REPRESENTAÇÃO NUMÉRICA

REPRESENTAÇÃO NUMÉRICA

- Cada computador possui uma precisão numérica diferente (***número máximo de dígitos***).
- Essa precisão depende do hardware, do sistema operacional, etc...

- Os erros ocorridos em soluções numéricas dependem da representação dos números na maquina utilizada – ***base de máquina.***

REPRESENTAÇÃO NUMÉRICA

- **Qual a base utilizada no nosso dia-a-dia?**

- A base decimal, onde se utiliza os algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.
- Um computador normalmente opera no sistema ou base binária (***zeros e uns***);
- Interação entre usuário e computador:

REPRESENTAÇÃO NUMÉRICA

- **Exemplos:**

- $(100110)_2 = (38)_{10}$
- $(25)_{10} = (11001)_2$

Como realizar essa conversão?

CONVERSÃO DE NÚMEROS: DECIMAL E BINÁRIO

- **Representação de um número na base β:**

Qualquer computador trabalha internamente com uma **base fixa β**, onde β é um inteiro ≥ 2.

Dessa forma, dado um número real ($x \neq 0$), este possui uma única representação, dada, de forma genérica, por:

$$x = \pm(d_n d_{n-1} \ldots d_2 d_1 d_0) = \pm(d_n \beta^n + d_{n-1}\beta^{n-1} + \ldots + d_1 \beta^1 + d_0 \beta^0)$$

$$x_f = \pm(b_n b_{n-1} \ldots b_2 b_1 b_0) = \pm(b_1 \beta^{-1} + b_2 \beta^{-2} + \ldots + d_{n-1}\beta^{-(n-1)} + d_n \beta^n)$$

CONVERSÃO DE NÚMEROS: DECIMAL E BINÁRIO

- **Representação de um número na base decimal:**

$$\begin{array}{cccccccccc} 10^4 & 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} & 10^{-3} & 10^{-4} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 6 & 0 & 7 & 2 & 4 & . & 3 & 1 & 2 & 5 \end{array}$$

$$6\times10^4+0\times10^3+7\times10^2+2\times10^1+4\times10^0+3\times10^{-1}+1\times10^{-2}+2\times10^{-3}+5\times10^{-4}=60.724,3125$$

Como seria a representação do número 1997 em uma base $\beta = 10$?

E do número 19,625 em uma base $\beta = 10$?

CONVERSÃO DE NÚMEROS: DECIMAL E BINÁRIO

- **Representação de um número na base binária:**

$$\begin{array}{cccccccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} & 2^{-3} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 0 & 1 & 1 & . & 1 & 0 & 1 \end{array}$$

$$1\times 2^4+0\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$$

$$1\times 16+0\times 8+0\times 4+1\times 2+1\times 1+1\times 0{,}5+0\times 0{,}25+1\times 0{,}125 = 19{,}625$$

CONVERSÃO DE NÚMEROS: DECIMAL E BINÁRIO

- **Representação de um número na base binária:**

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	. 0	1	0	1

$$1\times2^{15}+1\times2^{14}+1\times2^{13}+0\times2^{12}+1\times2^{11}+1\times2^{10}+0\times2^{9}+1\times2^{8}+0\times2^{7}+0\times2^{6}+1\times2^{5}$$
$$+1\times2^{4}+0\times2^{3}+1\times2^{2}+0\times2^{1}+0\times2^{0}+0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}=60.724,3125$$

- Cada algarismo binário (1 ou 0) é chamado de **bit** (*binary digit*);
- A **aritmética binária** é usada por computadores porque **transistores** modernos podem ser usados como **chaves extremamente rápidas**.
- "**1**" se referindo à chave na posição "**ligada**", e o "**0**" se referindo à posição "**desligada**".

MAS, COMO FAZER AS CONVERSÕES?

Usuário

Dados de entrada sistema decimal

Decimal para binário

Computador

Operações no sistema binário

Binário para decimal

Usuário

Dados de saída sistema decimal

Base 10	Base 2			
	2^3	2^2	2^1	2^0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

BINÁRIA PARA DECIMAL

- **Exemplos:**

- $(1101)_2 = (??)_{10}$
- $(11001)_2 = (??)_{10}$

BINÁRIA PARA DECIMAL

- **Exemplos:**

- **$(1101)_2 = (??)_{10}$**

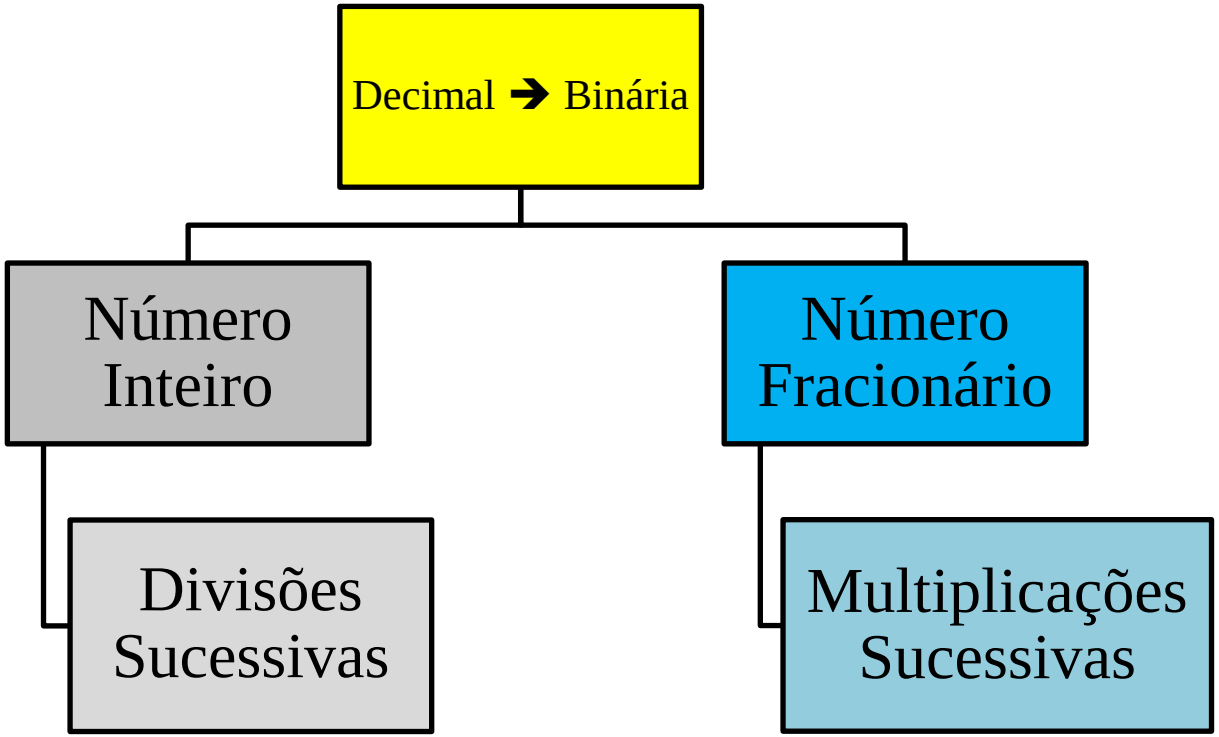
$$(1101)_2 = 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = 8+4+0+1 = (13)_{10}$$

- **$(11001)_2 = (??)_{10}$**

$$(11001)_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 = 16+8+0+0+1 = (25)_{10}$$

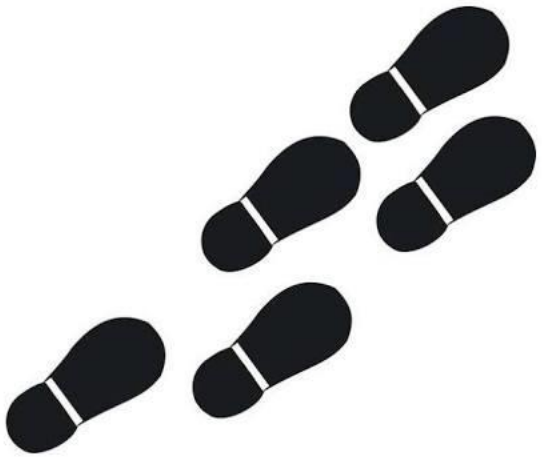
DECIMAL PARA BINÁRIA

- Na conversão de um número escrito na **base decimal** para a **base binária**, são utilizados: o **método das divisões sucessivas**, para a parte inteira, e o **método das multiplicações sucessivas**, para conversão da parte fracionária do número em questão.



Método das divisões sucessivas

- **Passos:**
  - a) Divide-se o número (parte inteira) por 2;
  - b) Em seguida, divide-se novamente por 2, o quociente da divisão anterior;
  - c) E se repete este processo até o último quociente ser igual a 1.



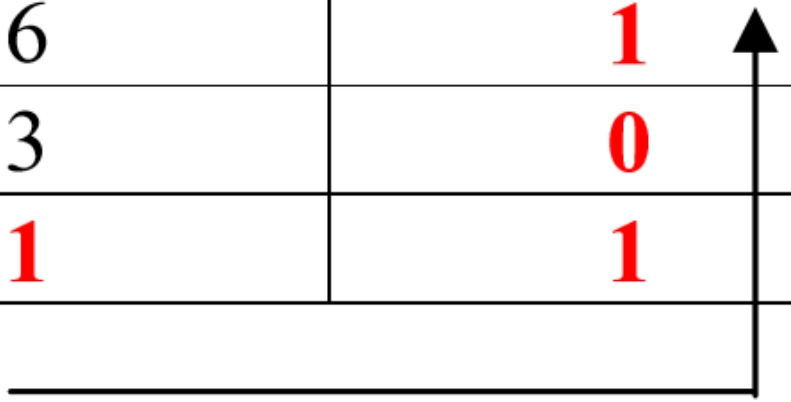
Método das multiplicações sucessivas

- **Passos:**
  - a) Multiplica-se o número (parte fracionária) por 2;
  - b) o resultado, a parte inteira será o primeiro dígito do número na base binária e a parte fracionária é novamente multiplicada por 2;
  - c) E se repete este processo até o último quociente ser igual a 1.

EXEMPLOS

- a) $(13)_{10} = (??)_2$

	Quociente	Resto
13/2	6	**1**
6/2	3	**0**
3/2	**1**	**1**



- $(13)_{10} = (1101)_2$

EXEMPLOS

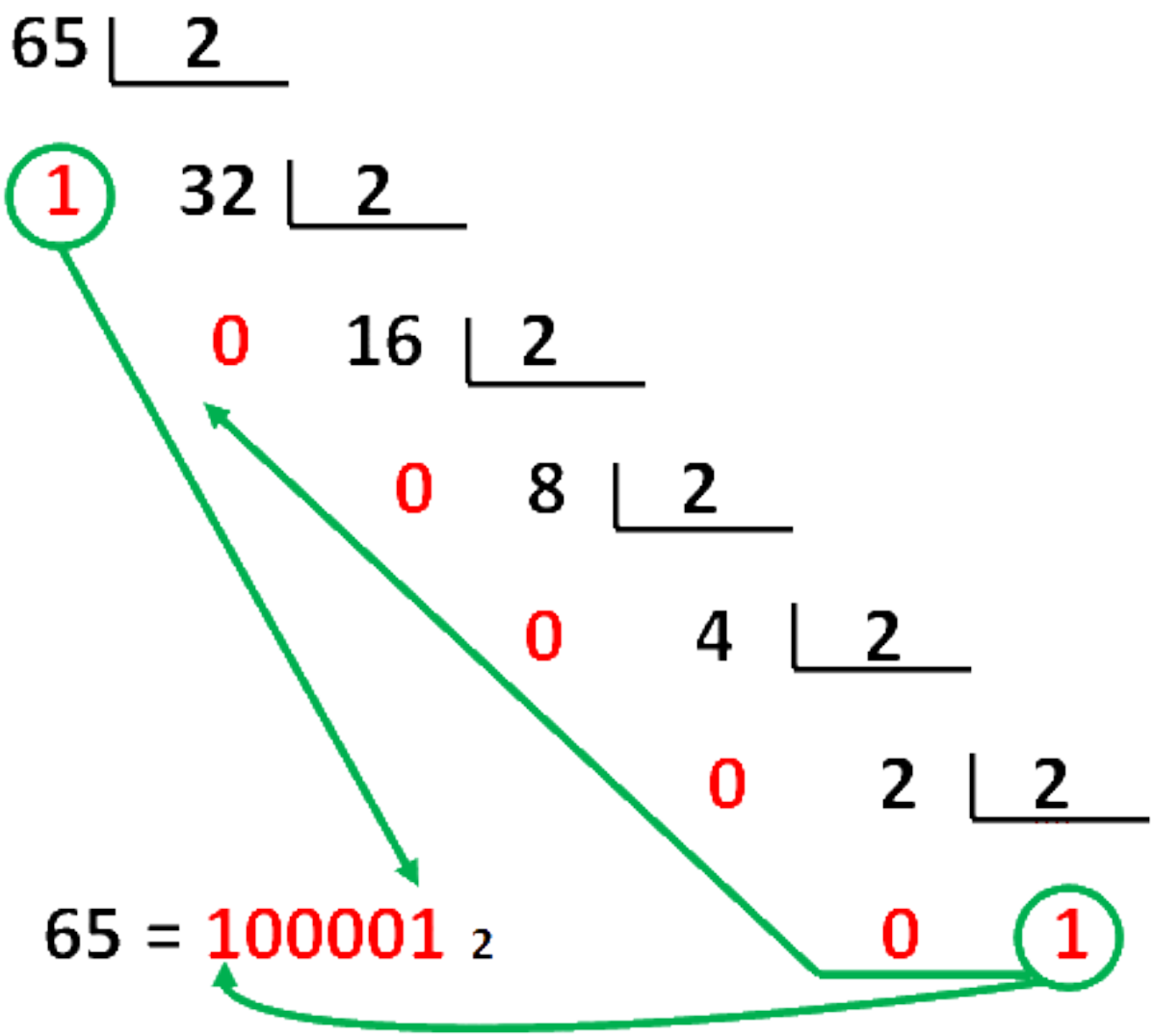
- b) $(25)_{10} = (??)_2$

	Quociente	Resto
25/2	12	**1**
12/2	6	**0**
6/2	3	**0**
3/2	**1**	**1**

- $(25)_{10} = (11001)_2$

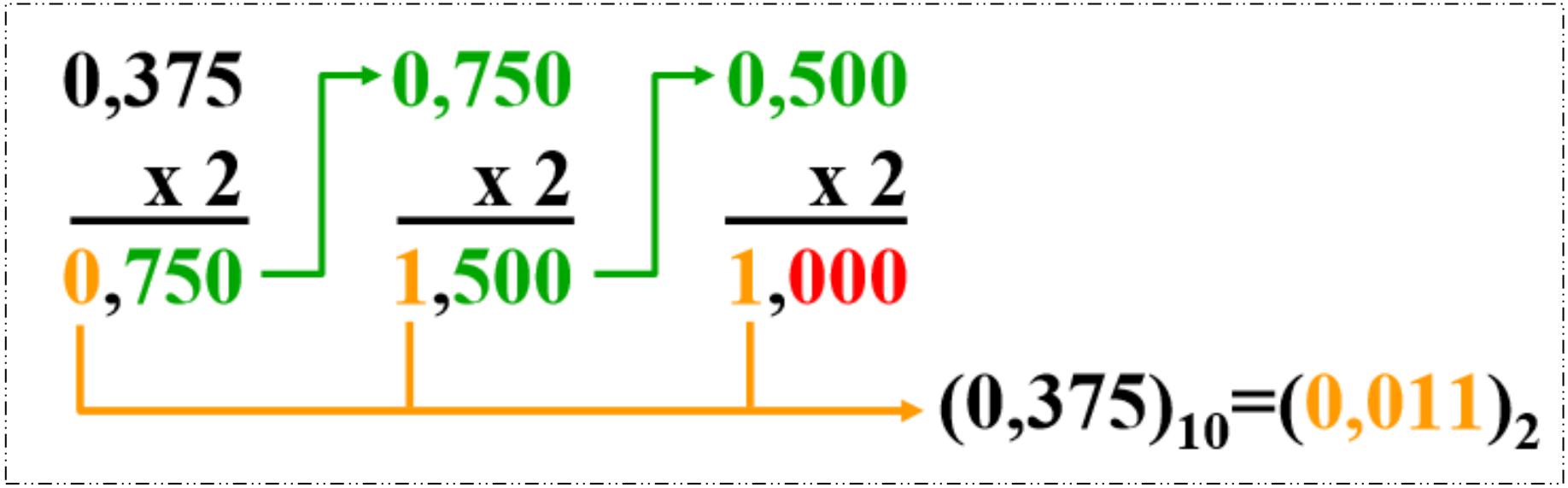
EXEMPLOS

- c) $(65)_{10} = (??)_2$



EXEMPLOS

- c) $(0{,}375)_{10} = (??)_2$



- $(0{,}375)_{10} = (0{,}011)_2$

EXERCÍCIO

- d) $(13,25)_{10} = (??)_2$

1. Converte-se inicialmente a parte inteira do número:

2. Em seguida, converte-se a parte fracionária:

EXERCÍCIO

Converte-se inicialmente a parte inteira do número:

	Quociente	Resto
13/2	6	1
7/2	3	0
3/2	1	1

... Em seguida, converte-se a parte fracionária:

$$(0{,}25)_{10} = (0{,}01)_2$$

0,25	0,50
x 2	x 2
0,50	1,0

Resultado: $(13{,}25)_{10} = (1101{,}01)_2$

ATENÇÃO!

Nem todo número real na base decimal possui uma representação finita na base binária.

Tente fazer a conversão de $\mathbf{(0,1)_{10}}$. Esta situação ilustra bem o caso de erro de arredondamento nos dados.

Aritmética em Ponto Flutuante

Aritmética em Ponto Flutuante

- BASE DECIMAL $(\#\#)_{10}$

- Números REAIS grandes e pequenos, são escritos na representação em ponto flutuante.
- A representação decimal em ponto flutuante (também chamada de notação científica) tem a forma:

$$0,ddddddd \times 10^{p}$$

- Exemplos:

6519,23 é escrito como $0{,}651923 \times 10^{4}$

0,00000391 é escrito como $0{,}391 \times 10^{-5}$

Aritmética em Ponto Flutuante

- BASE BINÁRIA $(\#\#)_2$

- A representação binária em ponto flutuante tem a forma:

$$1,bbbbbb \times 2^{bbb}$$

- O número $0,bbbbbb$ é chamado de **MANTISSA**, e pode ser representado genericamente por:

$$\pm \left[\frac{b_1}{\beta} + \frac{b_2}{\beta^2} + \frac{b_3}{\beta^3} + \cdots + \frac{b_t}{\beta t} \right] . \beta^e$$

Aritmética em Ponto Flutuante

- Como representar um número real na forma binária em ponto flutuante?

Aritmética em Ponto Flutuante

- Como representar um número real na forma binária em ponto flutuante?
- Através da normalização do número em relação à maior potência de 2 (menor que o próprio número)
  - Exemplos:

$$50 = \frac{50}{2^5} \times 2^5 = 1{,}5625 \times 2^5, \text{ que na forma binária é } 1{,}1001 \times 2^{101}$$

$$1344 = \frac{1344}{2^{10}} \times 2^{10} = 1{,}3125 \times 2^{10}, \text{ que na forma binária é } 1{,}0101 \times 2^{1001}$$

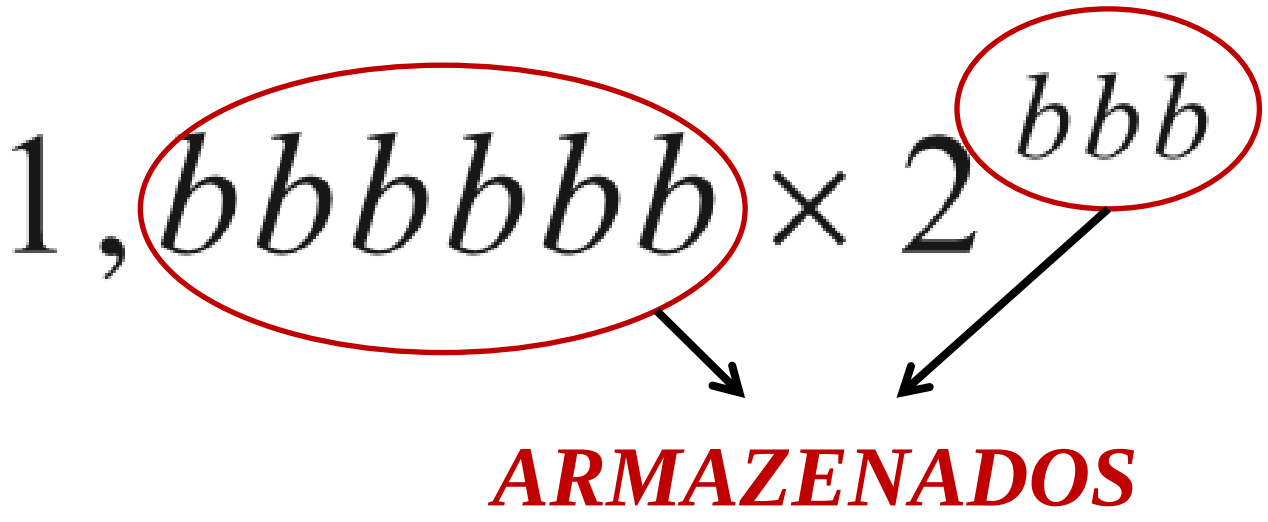
$$0{,}3125 = \frac{0{,}3125}{2^{-2}} \times 2^{-5} = 1{,}25 \times 2^{-2}, \text{ que na forma binária é } 1{,}01 \times 2^{-10}$$

Armazenando um número na memória do computador

Armazenando um número no computador

- Uma vez colocado na representação binária em ponto flutuante, o número é armazenado no computador;

- O computador armazena os valores do **expoente** e da **mantissa** SEPARADAMENTE, não sendo armazenado o primeiro 1 à frente da vírgula decimal.



Cálculo no Computador

Em precisão Simples

- Os números são armazenados em uma cadeia **32 bits** (4 bytes)
- O **1º bit** armazena o **sinal** do número (0 [+] e 1 [−])
- **8 bits** são usados para armazenar o **expoente**
- **23 bits** são usados para armazenar a **mantissa**

Em precisão Dupla

- Os números são armazenados em uma cadeia **64 bits** (8 bytes)
- O **1º bit** armazena o **sinal** do número (0 [+] e 1 [−])
- **11 bits** são usados para armazenar o **expoente**
- **52 bits** são usados para armazenar **a mantissa**

Armazenando um número no computador

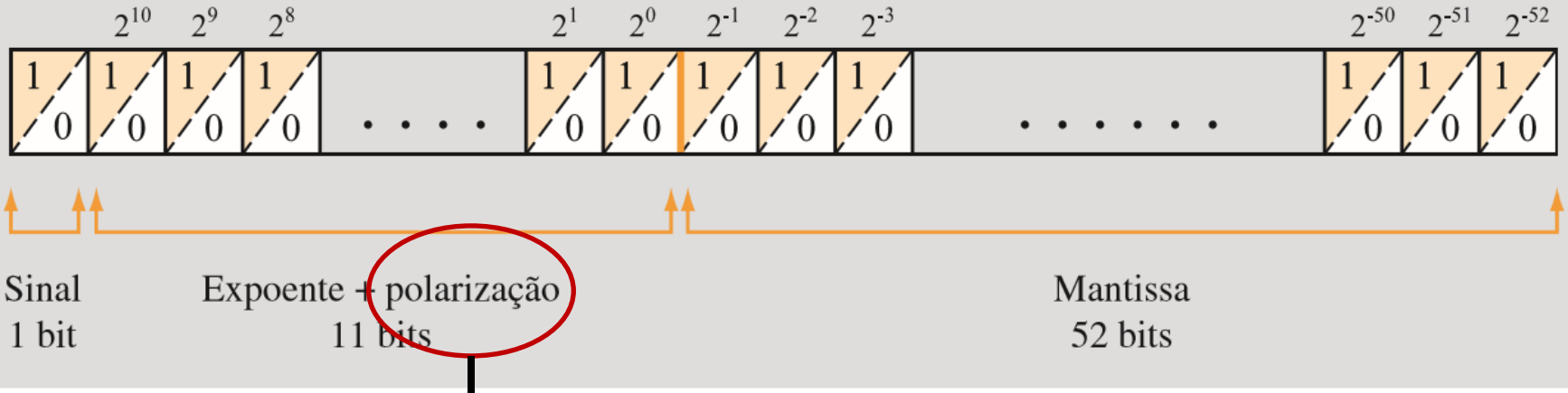


Figura Armazenando em precisão dupla um número escrito na representação binária em ponto flutuante.

- A polarização corresponde à adição de uma constante ao valor do expoente;
- A polarização é introduzida para se evitar o uso de um dos bits para representar o sinal do expoente (+ ou -).

Armazenando um número no computador

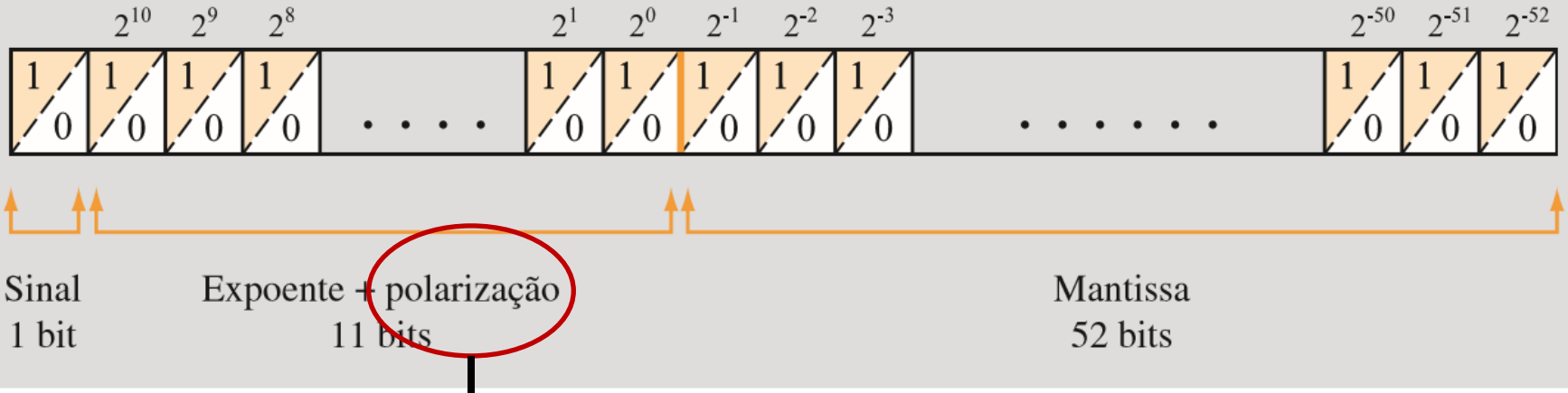


Figura Armazenando em precisão dupla um número escrito na representação binária em ponto flutuante.

- Na notação binária. o maior número que pode ser escrito com 11 bits é 2047; **{-1023 à 1023}**

- A polarização utilizada é 1023. Isso significa que, se o expoente for 4, então o valor armazenado é 4 + 1023 = 1027.

Armazenando um número no computador

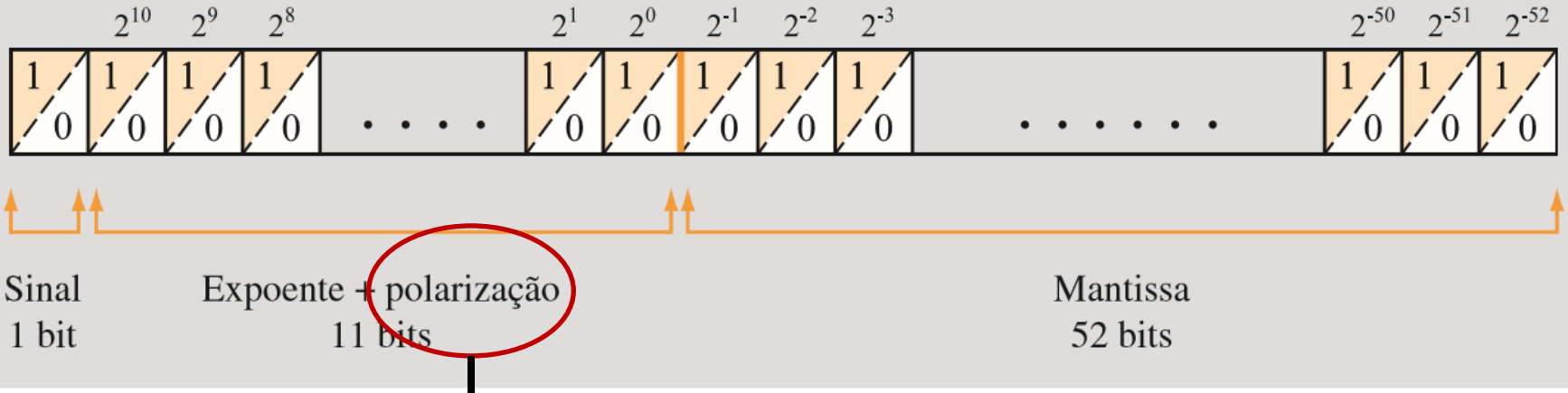


Figura Armazenando em precisão dupla um número escrito na representação binária em ponto flutuante.

- O menor expoente que pode ser armazenado pelo computador é –1023 (que é armazenado como 0);
- O maior é 1024 (que é armazenado como 2047).

- Na precisão simples, 8 bits são alocados para o valor do expoente, e a polarização é 127. (-127 à 127)

Armazenando um número no computador

- **EXEMPLO: Armazenar o número 22,5 em precisão dupla:**

- **1º Passo:** Normalizar o Número:

$$\frac{22{,}5}{2^4} . 2^4 = 1{,}40625 \times 2^4$$

Armazenando um número no computador

- **EXEMPLO: Armazenar o número 22,5 em precisão dupla:**

- **1º Passo:** Normalizar o Número: $1{,}40625 \times 2^{4}$

- **2º Passo:** Cálculo do expoente Polarizado em precisão Dupla:

$$4 + 1023 = 1027$$

Armazenando um número no computador

- **EXEMPLO: Armazenar o número 22,5 em precisão dupla:**

- **1º Passo:** Normalizar o Número: $1{,}40625 \times 2^4$

- **2º Passo:** Polarização do expoente em precisão Dupla:

$$4 + 1023 = 1027$$

- **3º Passo:** Converter a mantissa e o expoente polarizado para a base binária:
  - Expoente + Polarização : 10000000011
  - mantissa é 0,40625 : 0,01101000….000

Armazenando um número no computador

- **3º Passo:** Converter a mantissa e o expoente polarizado para a base binária:
  - Expoente: 10000000011
  - mantissa é 0,40625 : 0,01101000….000
- **4º Passo:** Diagrama do número armazenado no computador:

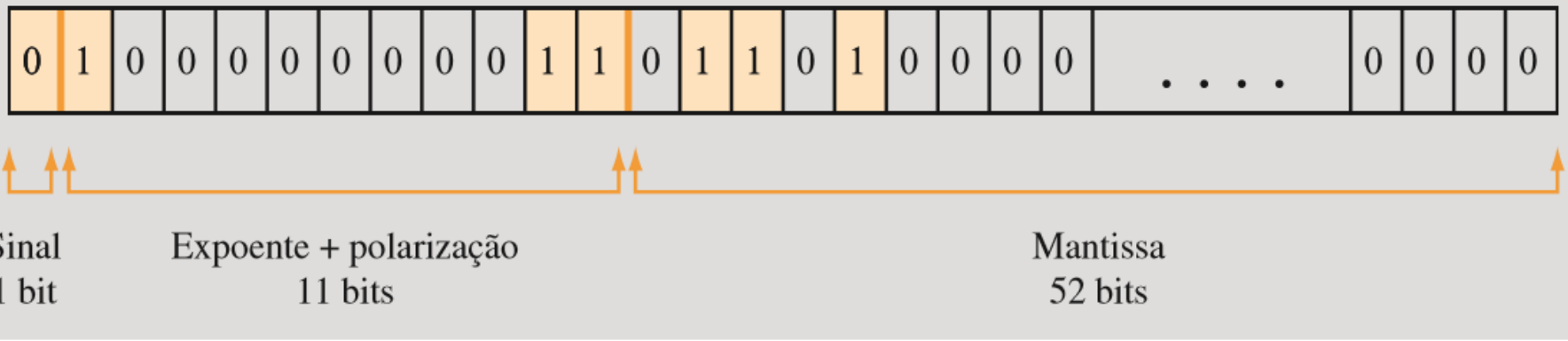


Figura Armazenamento do número 22,5 em precisão dupla

Armazenando um número no computador

Notas adicionais:

- O menor número positivo que pode ser expressado em precisão dupla é: $2^{-1023} \cong 1{,}1 \times 10^{-308}$

- O maior número positivo que pode ser expressado em precisão dupla é aproximadamente: $2^{1024} \cong 1{,}8 \times 10^{308}$

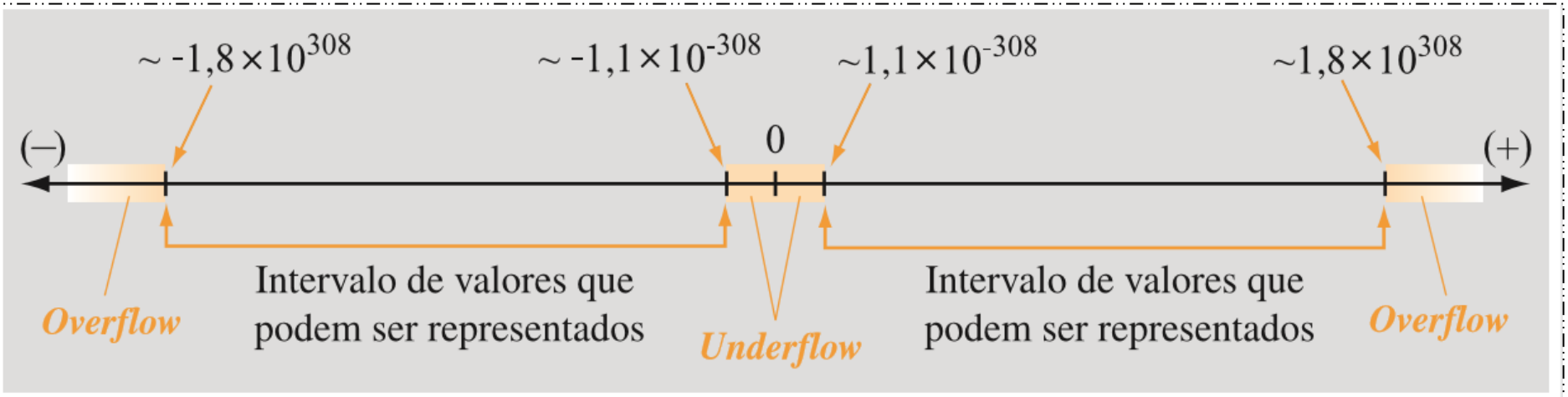


Figura Intervalo de números que podem ser representados em precisão dupla.

Exemplo

- Escrever os números reais: a) $x_1 = 0.35$; b) $x_2 = -5.172$; a) $x_3 = 0.0123$; $x_4 = 0.0003$; $x_5 = 5391.3$ que estão todos na base $\beta = 10$, em notação científica.

a) $0.35 = (3\times10^{-1} + 5\times10^{-2})x10^{0} = 0.35\times10^{0}$

b) $-5.172 = -(5\times10^{-1} + 1\times10^{-2} + 7\times10^{-3} + 2\times10^{-4})\times10^{1} = -0.5172\times10^{1}$

c) $0.0123 = (1\times10^{-1} + 2\times10^{-2} + 3\times10^{-3})\times10^{-1} = 0.123\times10^{-1}$

d) $5391.3 = (5\times10^{-1} + 3\times10^{-2} + 9\times10^{-3} + 1\times10^{-4} + 3\times10^{-5})\times10^{4} = 0.53913\times10^{4}$

e) $0.0003 = (3\times10^{-1})\times10^{-3} = 0.3\times10^{-3}$

Exemplo

- Considerando agora que estamos diante de uma máquina que utilize apenas três dígitos significativos e que tenha como limite inferior e superior para o expoente, respectivamente, -2 e 2, como seriam representados nesta máquina os números da questão anterior?

a) $0.35 = 0.350 \times 10^{0}$

b) $-5.172 = -0.517 \times 10^{1}$

c) $0.0123 = 0.123 \times 10^{-1}$

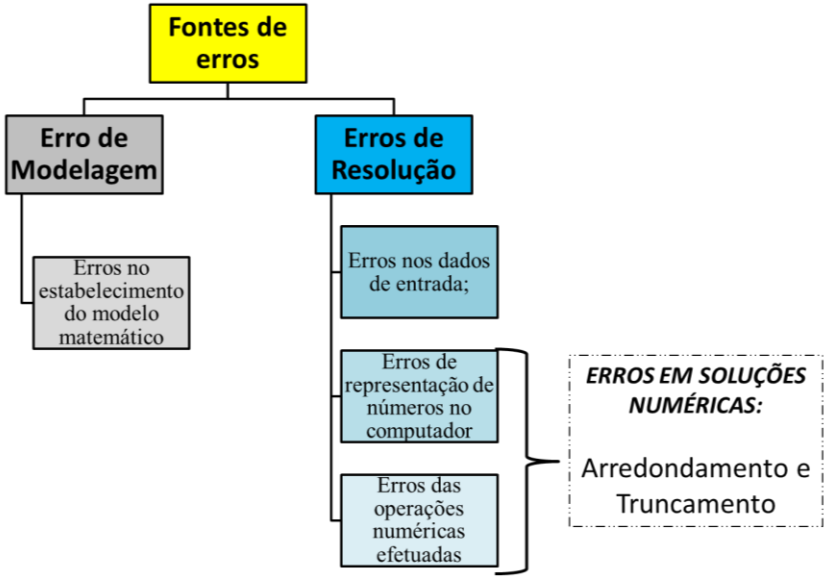
d) $5391.3 = 0.53913 \times 10^{4}$. Não pode ser representado por esta máquina. Erro de ***overflow***.

e) $0.0003 == 0.3 \times 10^{-3}$. Não pode ser representado por esta máquina. Erro de ***underflow***.

ERROS EM SOLUÇÕES NUMÉRICAS

Erro de Arredondamento e Truncamento

Números reais que têm uma **mantissa mais longa** do que o **número de bits disponíveis** para representá-los têm que ser **encurtados.**



- Arredondamento;
- Truncamento.

Erro de Arredondamento e Truncamento

Exemplo: Dar a representação dos números a seguir num sistema de aritmética de ponto flutuante de três dígitos para uma base decimal e expoente variando de -4 a 4.

x	**Representação por arredondamento**	**Representação por truncamento**
1,25	0,125x10	0,125x10
10,053	$0,101x10^2$	$0,100x10^2$
-238,15	$-0,238x10^3$	$-0,238x10^3$
2,71828	0,272x10	0,271.10
0,000007	Exp< -4 (underflow)	Exp < -4 (underflow)
718235,82	Exp > 4 (overflow)	Exp > 4 (overflow)

- Quando se utiliza o **arredondamento** os erros cometidos são menores que no **truncamento**, no entanto, o **arredondamento** requer um maior tempo de execução.

Erro de Arredondamento e Truncamento

- Exemplo erro Truncamento:

Avaliação numérica de *sen(x)*, feita a partir da expansão em série de Taylor.

$$\operatorname{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \ldots$$

Erros absoluto, relativo e percentual

Erros absoluto, relativo e percentual

- Erro absoluto: Diferença entre o valor exato de um número ***x*** e seu valor aproximado $\bar{x}$ obtido a partir de um procedimento numérico.

$$E_{ax} = |x - \bar{x}|$$

- Em geral, apenas $\bar{x}$ é conhecido, e o que se faz é assumir um limitante superior e inferior ou uma estimativa para o módulo do erro absoluto.

Seja y representado por $\bar{y} = 5{,}3$ de forma que $|E_{a_y}| < 0{,}1$, podemos dizer que $y \in (5{,}2;\ 5{,}4)$

Erros absoluto, relativo e percentual

Erro Relativo: Erro absoluto dividido pelo valor real.

$$E_{rx} = \left| \frac{E_{ax}}{x} \right| = \left| \frac{x - \bar{x}}{x} \right|$$

Erro Percentual: é o Erro relativo em termos percentuais.

$$E_{px} = 100 \times E_{rx}$$

Erros absoluto, relativo e percentual

Exemplo: Escreva o número 0,1 no formato binário usando um número suficiente de algarismos de forma que o erro relativo real seja menor que 0,07.

Erros absoluto, relativo e percentual

Exemplo: Escreva o número 0,1 no formato binário usando um número suficiente de algarismos de forma que o erro relativo real seja menor que 0,07.

Passo 1: Cálculo do intervalo em que o valor aproximado e válido para o Erro Relativo considerado:

$$\left|\frac{0.1-\bar{x}}{0.1}\right| < 0.07$$

$$-0.07 < \frac{0.1-\bar{x}}{0.1} < 0.07$$

$$-0.007 < 0.1-\bar{x} < 0.007$$

$$0.093 < \bar{x} < 0.107$$

Erros absoluto, relativo e percentual

Exemplo: Escreva o número 0,1 no formato binário usando um número suficiente de algarismos de forma que o erro relativo real seja menor que 0,07.

Passo 2: Escrever o número na forma binária e verificar se o mesmo obedece o intervalo calculado no passo 1:

$$(0.1)_{10} = (0.000110...)_2 = (0 \times 2^{-1}) + (0 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4}) + (1 \times 2^{-5}) + (...)$$

$$(0.00011)_2 = (0.09375)_{10}$$

$$0.093 < \bar{x} < 0.107$$

Propagação de erros

Propagação de erros

- Suponhamos que as operações indicadas nos itens a) e b) sejam processadas numa máquina com 4 dígitos significativos.

a) $(x_2 + x_1) - x_1$

b) $x_2 + (x_1 - x_1)$

Fazendo $x_1 = 0.3491 \times 10^4$ e $x_2 = 0{,}2345 \times 10^0$ temos:

$$a)\ (x_2 + x_1) - x_1 = (0{,}2345.10^0 + 0{,}3491.10^4) - 0{,}3491.10^4$$
$$= 0.3491.10^4 - 0{,}3491.10^4$$
$$= 0{,}0000$$

$$b)\ x_2 + (x_1 - x_1) = 0{,}2345.10^0 + (0{,}3491.10^4 - 0{,}3491.10^4)$$
$$= 0.2345.10^0 + 0{,}0000$$
$$= 0{,}2345$$

...CONTINUA