UFPR

UNIVERSIDADE FEDERAL DO PARANÁ

Métodos Numéricos

Introdução

Professor Volmir Eugênio Wilhelm
Professora Mariana Kleina

TP062-Métodos Numéricos para Engenharia de Produção

Ementa

Matrizes. Sistemas lineares. Zeros de funções algébricas e transcendentes. Soluções de sistemas lineares. Interpolação. Integração.

Programa

1. **Introdução ao Octave. Matrizes e sistemas lineares.**
2. **Representação de números reais e erros**: representação de número no sistema binário, aritmética do ponto flutuante, erros absolutos e relativos
3. **Zero de equações**: método da bisseção, método da posição falsa, método do ponto fixo, método de Newton-Raphson, método da secante.
4. **Sistemas de equações lineares**: métodos da eliminação de Gauss, fatoração LU, fatoração Cholseky, método iterativo de Gauss-Jacobi, método iterativo de Gauss-Seidel.
5. **Interpolação**: interpolação polinomial, forma de Lagrange, forma de Newton, ajuste de curvas pelo método dos quadrados mínimos
6. **Integração numérica**: regra dos trapézios, regra de Simpson

Referências

Ruggiero, M.A.G. e Lopes, V.L.R., **Cálculo Numérico Aspectos Teóricos e Computacionais**, 2a. ed., Pearson, 1996 (impressão de 2012).

Software Livre

Octave - http://www.gnu.org/software/octave/doc/interpreter/index.html

TP062-Métodos Numéricos para Engenharia de Produção

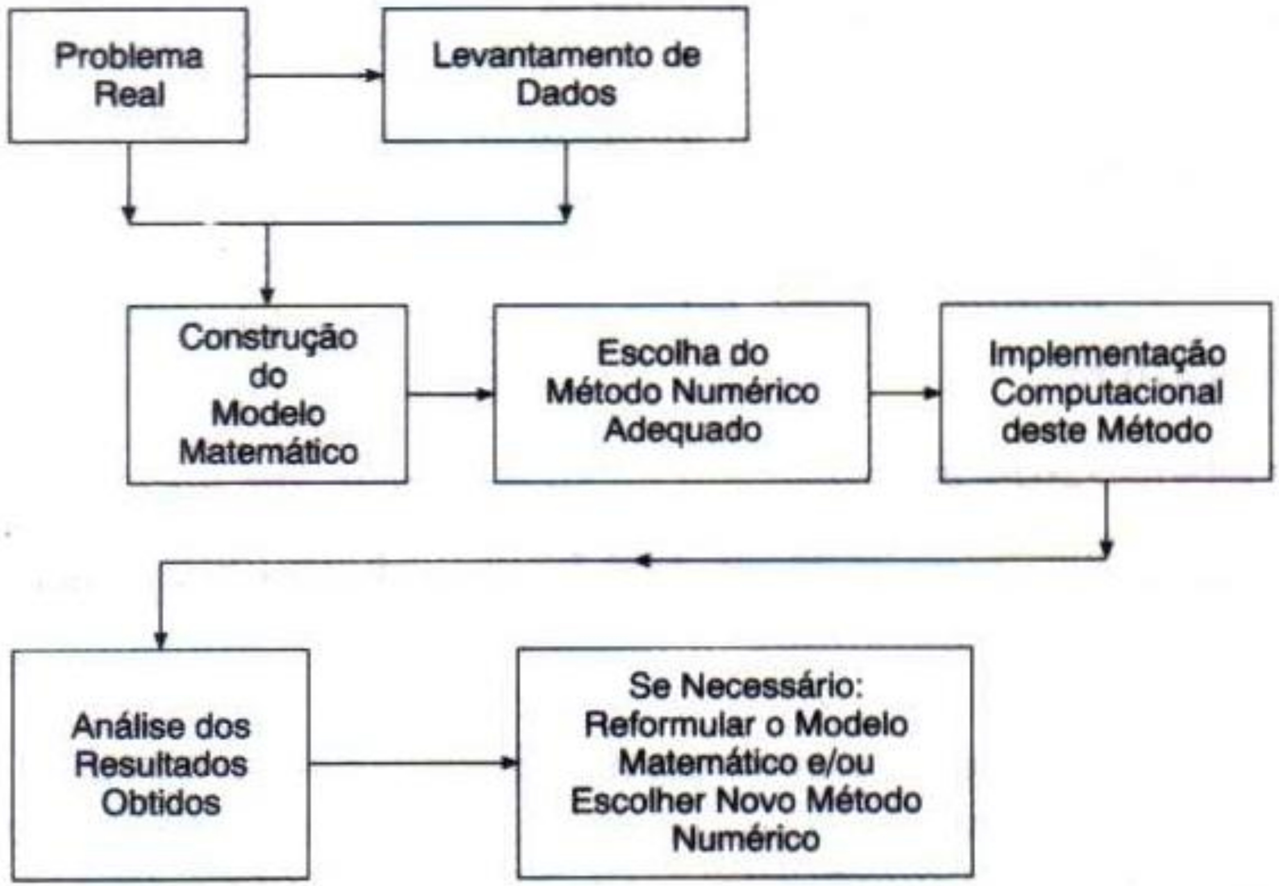
Nota

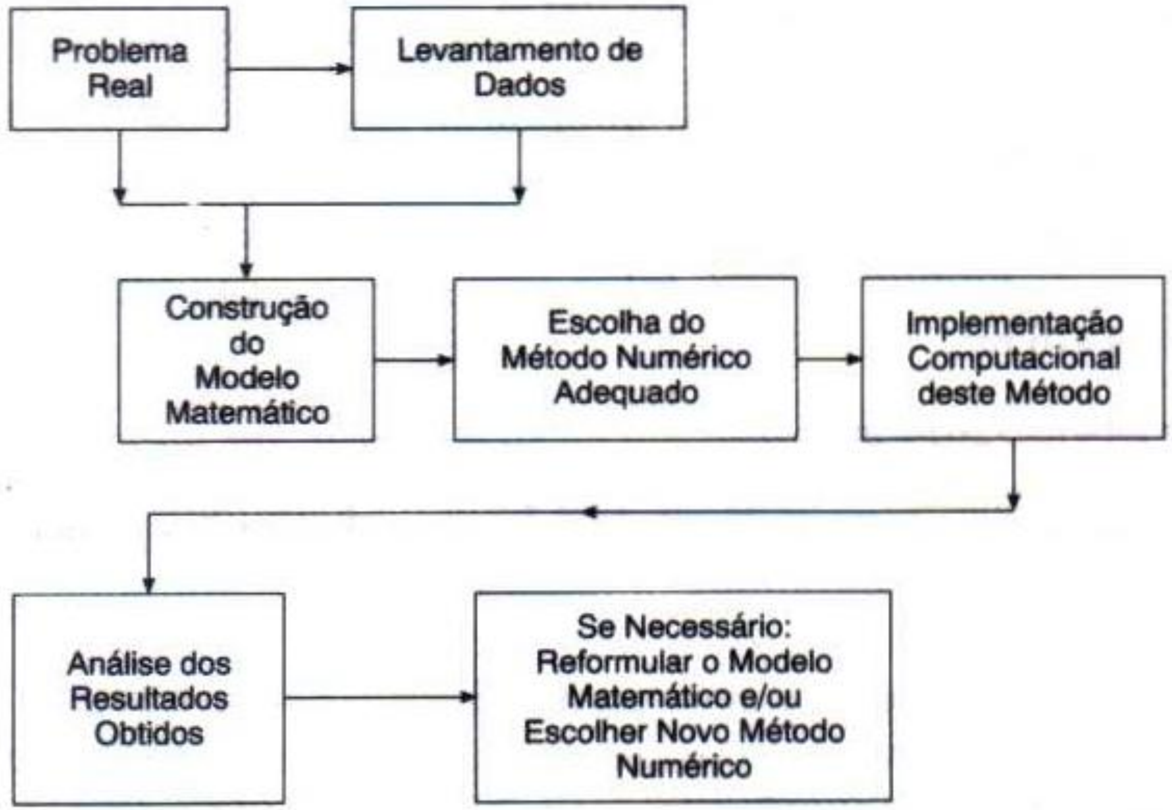
Este material de aula foi elaborado para compilar o conteúdo de várias referências bibliográficas tendo em vista o conteúdo programático da disciplina de Cálculo Numérico. Em particular, elas não substituem a consulta a livros textos e artigos. Seu principal objetivo é dispensar a necessidade dos alunos terem que copiar as aulas e, deste modo, poderem se concentrar em entender o conteúdo das mesmas.

Tem materiais incluídos neste documento de outros autores e fontes bibliográficas quase todos devidamente identificados. Caso o usuário conheça alguma fonte não identificada no texto, por favor, comunique para efetuar os devidos créditos.

Para informações mais completas, deve-se recorrer aos livros ou Web sites citados durante o semestre letivo.

TP062-Métodos Numéricos para Engenharia de Produção





Os resultados obtidos dependem também:

1. da precisão dos dados de entrada;
2. da forma de representação no computador;
3. das operações numéricas.

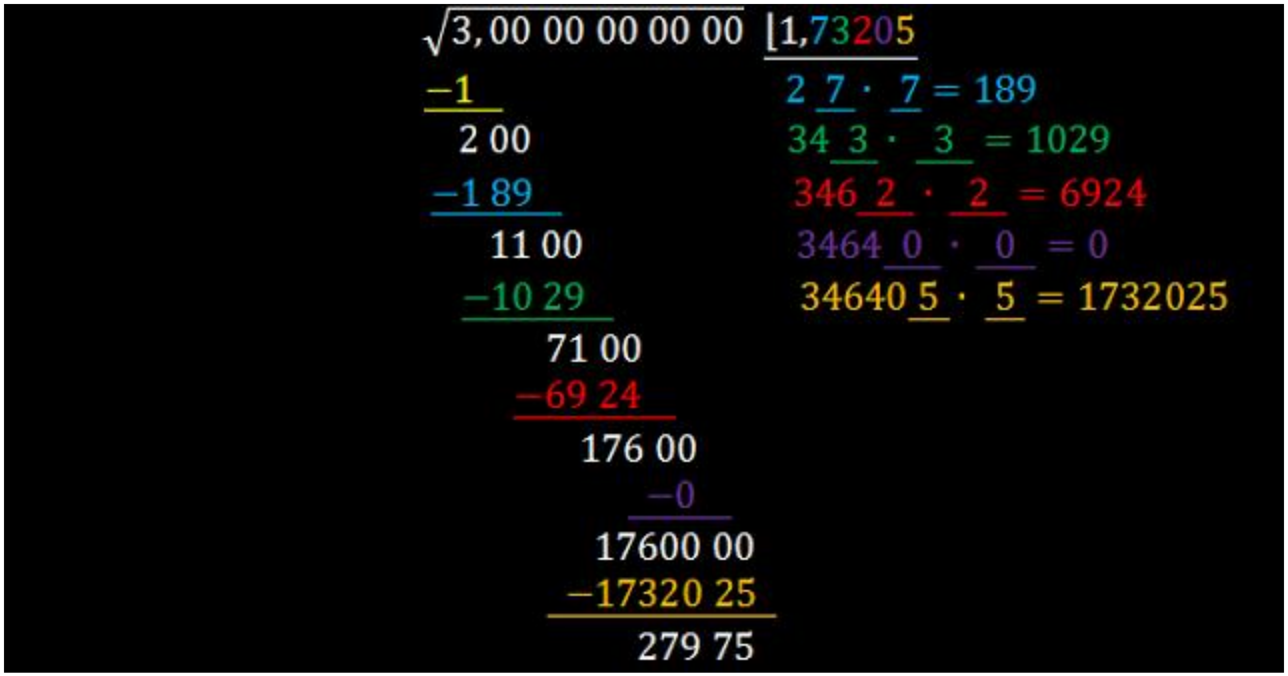
O que é Cálculo Numérico?

"O Cálculo Numérico corresponde a um conjunto de **ferramentas** ou **métodos** utilizados para se obter a solução de problemas matemáticos de forma **aproximada**. Esses métodos se aplicam principalmente a problemas que não apresentam uma solução exata, portanto precisam ser resolvidos **numericamente"**.

Rodrigo Cristiano Silva - rodrigo@facens.br

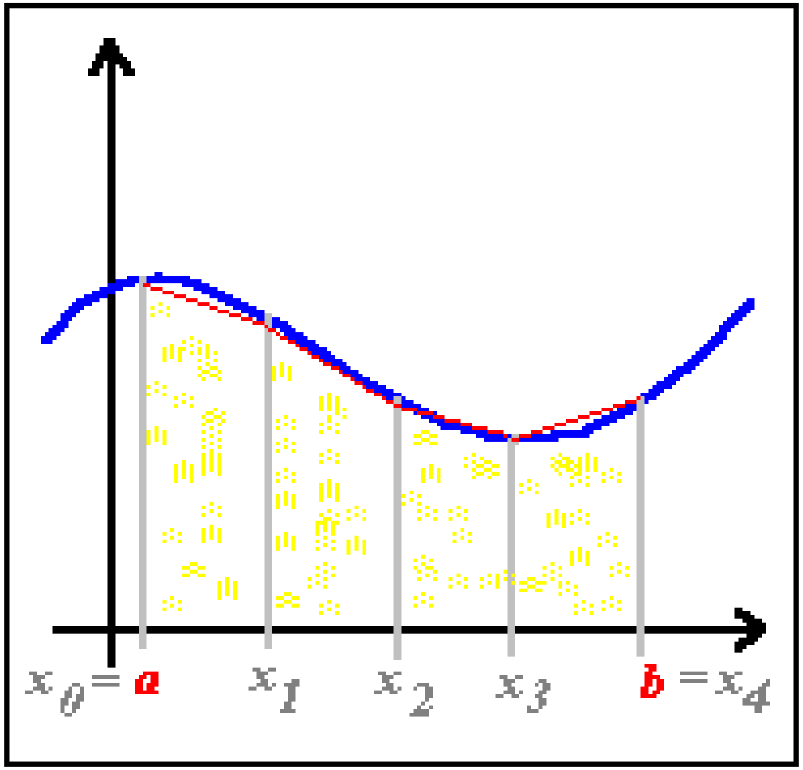
Curiosidades

a) Calcular $\sqrt{3}$



Qual é a precisão utilizada pelo computador ou calculadora, tendo em vista que raiz de 3 é um número irracional?

b) Calcular $\int e^{x^2}\,dx$



Soma da área de ***n*** trapézios, cada qual definido pelo seu subintervalo.

Qual é a precisão utilizada pelo computador ou calculadora, tendo em vista que é um número irracional? Qual o valor de ***n*** para que a precisão desejada seja alcançada?

c) Sabe-se que o diâmetro aproximado da terra é 12.750km. Calcular aproximadamente o volume da terra

pi = 3,14159265358979

volume=(4/3)*(3,1)*(12750/2)^3 %volume = 1070880468750,00

volume=(4/3)*(3,1415)*(12750/2)^3 %volume = 1085216449218,75
%diferença = 14335980468,7500 (≈1,33%)

volume=(4/3)*(pi)*(12750/2)^3 %volume = 1085248455967,03
%diferença = 32006748,2803955 (≈0,0029%)

Como explicar essa diferença no valor final do volume?

d) Calcular $\text{soma} = \sum_{i=1}^{30000} x_i$ (usando OCTAVE)

- **Para x_i = 0,5**

```
soma = 0; for i=1:3000, soma = soma + 0.5; end; soma
```

soma = 1500

correto = 3000*0.5   %correto = 1500
erro = soma – correto   %erro = 0

- **Para x_i = 0,11**

```
soma = 0; for i=1:3000, soma = soma + 0.11; end; soma
```

soma = 330.000000000024

correto = 3000*0.11   %correto = 330
erro = soma – correto   %erro = 2,44426701101474e-011

Como explicar esse erro?

e) Calcular $\text{soma} = \sum_{i=1}^{30000} x_i$ para x_i = 0,11 e x_i = 0,10

Para x_i = 0,11

Na mão: 330

No Octave: 330,000000000024 [erro: 2,44426701^(-11)]

No Matlab: 330,0000000000244 [erro: 2,44426701^(-11)]

Para x_i = 0,10

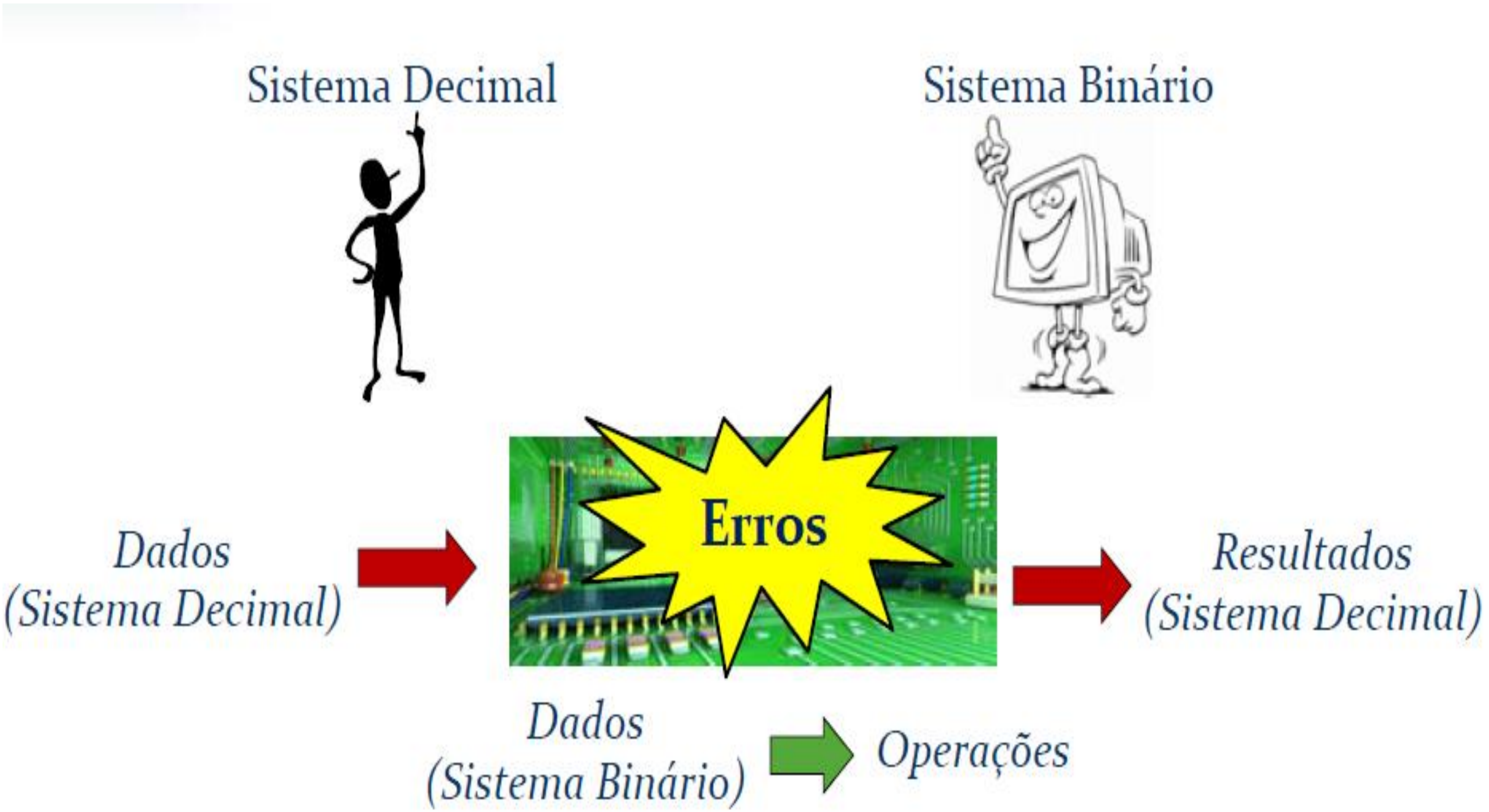
Na mão: 330

No Octave: 330,000000000000000 [erro: -2,8421709^(-13)]

No Matlab: 299,999999999999700 [erro: -2,8421709^(-13)]

Por que essa diferença?

Representação Numérica



Representação Numérica

Representação posicional – Números Inteiros

– Base decimal (10)
  - 10 dígitos disponíveis {0,1,2, ... ,9}
  - “Posição” indica potência positiva de 10
  - $5432_{(10)} = 5\text{x}10^3 + 4\text{x}10^2 + 3\text{x}10^1 + 2\text{x}10^0$

– Base binária (2)
  - 2 dígitos disponíveis {0,1}
  - “Posição” indica potência positiva de 2
  - $1011_{(2)}$ (na base 2) =

    $1\text{x}2^3 + 0\text{x}2^2 + 1\text{x}2^1 + 1\text{x}2^0 = 8+0+2+1 = 11_{(10)}$

Representação Numérica

Representação posicional – Números Fracionários

- Base decimal (10)
  - "Posição" da parte inteira indica potência positiva de 10
  - Potência negativa de 10 para parte fracionária
  - $547{,}32_{(10)} = 5x10^2 + 4x10^1 + 7x10^0 + 3x10^{-1} + 2x10^{-2}$

- Base binária (2)
  - "Posição" da parte inteira indica potência positiva de 2
  - Potência negativa de 2 para parte fracionária
  - $10{,}11_{(2)} = 1x2^1 + 0x2^0 + 1x2^{-1} + 1x2^{-2} = 2+0+1/2+1/4 = 2{,}75_{(10)}$

Representação Numérica

- **Maior interesse em decimal (10)**
  Anatomia e cultura humanas

- **e binário (2)**
  Uso em sistemas computacionais

- **Outros sistemas**
  - ✓ Octal (8), {0,1,2, ... , 7}
  - ✓ Hexadecimal (16), {0,1,2, ... , 9, A,B,C,D,E,F}
  - ✓ Duodecimal (12)

Conversão da Base Decimal para Binária

Inteiro de decimal para binário

- Operação: Divisão inteira (do quociente) sucessiva por 2, até que o resto seja 0 ou 1
- Composição do Binário : composição do **último quociente** (Bit Mais Significativo – **BMS**) com **restos** (primeiro resto é bit menos significativo – **bms**)

Em inglês, *Most Significant Bit – MSB e least significat bit – lsb,* respectivamente.

Conversão da Base Decimal para Binária

Inteiro de decimal (10) para binário (2) – Exemplos A

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

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

bms

BMS

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

Binário = BMS ... bms

(Binário = último quociente e todos os restos das divisões)

Binário = **MSB ... bms** = **1 1 0 1**$_{(2)}$ = $1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$ = 8 + 4 + 0 + 0 + 1 = $13_{(10)}$

---

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

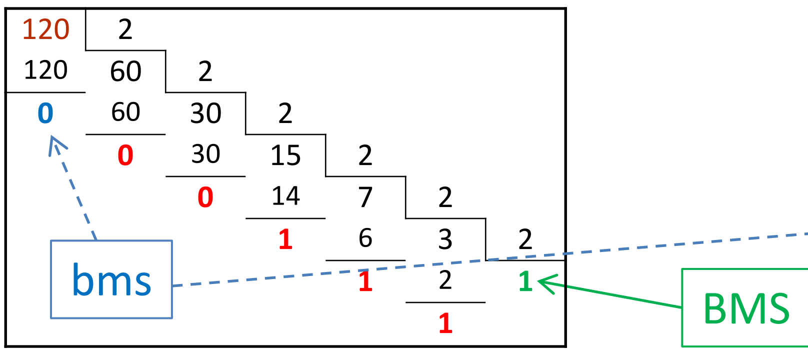
	Quociente	Resto
25/2	12	**1**
12/2	6	**0**
6/2	3	**0**
3/2	**1**	**1**

Resultado: $(25)_{10} = (11001)_2$ **1 1 0 0 1**$_{(2)}$ = $1x2^4 + 1x2^4 + 0x2^2 + 0x2^1 + 1x2^0$ = 16 + 8 + 0 + 0 + 1 = $25_{(10)}$

Conversão da Base Decimal para Binária

Inteiro de decimal (10) para binário (2) – Exemplos B

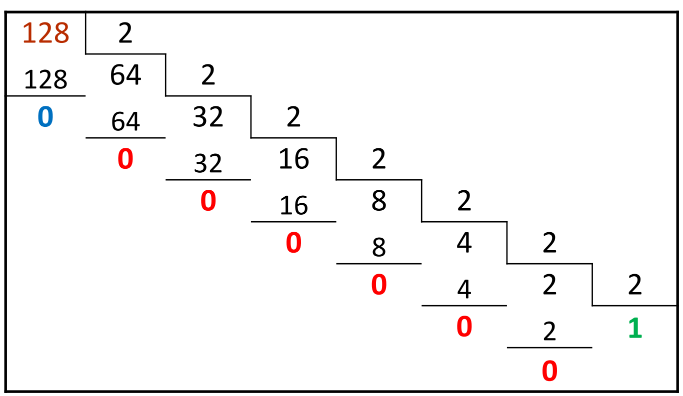
a)



$120_{(10)} = 1111000_{(2)}$

$1111000_{(2)} = 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 0x2^0 = 64 + 32 + 16 + 8 + 0 + 0 + 0 = 120_{(10)}$

b)



$128_{(10)} = 10000000_{(2)}$

$1000000_{(2)} = 1x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 0x2^2 + 0x21^1 + 0x2^0 = 128 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 128_{(10)}$

Conversão da Base Decimal para Binária

Fracionário de decimal para binário

- Operação inversa: multiplicar parte fracionária por 2 até que parte fracionária do resultado seja 0 (zero)

- Composição: Bits da parte fracionária derivados das partes inteiras das multiplicações. Bit imediatamente à direita da vírgula é a parte inteira da primeira multiplicação

Conversão da Base Decimal para Binária

Fracionário decimal para binário - Exemplo

a) $(0{,}375)_{(10)} = (?)_{(2)}$

$$\begin{array}{r} 0{,}375 \\ \times\ 2 \\ \hline 0{,}750 \end{array} \quad \begin{array}{r} 0{,}750 \\ \times\ 2 \\ \hline 1{,}500 \end{array} \quad \begin{array}{r} 0{,}500 \\ \times\ 2 \\ \hline 1{,}000 \end{array}$$

$(0{,}375)_{(10)} = (0{,}011)_{(2)}$

$(0{,}011)_{(2)} = 0 \text{x} 2^{-1} + 1 \text{x} 2^{-2} + 1 \text{x} 2^{-3} = 1/4 + 1/8 = 0{,}250 + 0{,}125 = 0{,}375_{(10)}$

---

b) $(0{,}25)_{(10)} = (?)_{(2)}$

$$\begin{array}{r} 0{,}25 \\ \times\ 2 \\ \hline 0{,}50 \end{array} \quad \begin{array}{r} 0{,}50 \\ \times\ 2 \\ \hline 1{,}00 \end{array}$$

$(0{,}25)_{(10)} = (0{,}01)_{(2)}$

$(0{,}01)_{(2)} = 0 \text{x} 2^{-1} + 1 \text{x} 2^{-2} = 1/4 = 0{,}25_{(10)}$

Conversão da Base Decimal para Binária

Fracionário decimal para binário - Exemplo

c) $(13{,}25)_{(10)} = (?)_{(2)}$

i) $13_{(2)} = (?)_{(2)}$

$$
\begin{array}{r|l}
13 & 2 \\
\underline{12} & 6 \;|\; 2 \\
\mathbf{1} & \underline{6} \quad 3 \;|\; 2 \\
 & \mathbf{0} \quad \underline{2} \quad \mathbf{1} \\
 & \quad\;\; \mathbf{1}
\end{array}
$$

$13_{(10)} = 1101_{(2)}$

ii) $0{,}25_{(10)} = (?)_{(2)}$

$$
\begin{array}{r}
0{,}25 \\
\times\, 2 \\
\hline
\mathbf{0}{,}50
\end{array}
\qquad
\begin{array}{r}
0{,}50 \\
\times\, 2 \\
\hline
\mathbf{1}{,}00
\end{array}
$$

$(0{,}25)_{(10)} = (0{,}01)_{(2)}$

$(13{,}25)_{(10)} = (1101{,}01)_{(2)}$

Conversão da Base Decimal para Binária

Limitação

- É possível representar exatamente apenas números racionais que tenham parte fracionária da forma $\sum_{k=-j}^{0} b_k 2^k$. Por exemplo:

Número	Representação Exata
1/8 (= 0,125)	$0{,}001_{(2)}$
1/16 (= 0,0625)	$0{,}0011_{(2)}$
5,625	$101{,}101_{(2)}$

- Outros números possuem sequências de bits repetidas indefinidamente - a representação binária não é precisa. Por exemplo:

Número	Representação Aproximada
1/3	$0{,}0101010101[01]\ldots_{(2)}$
1/5	$0{,}001100110011[0011]\ldots_{(2)}$
1/10	$0{,}0001100110011[0011]\ldots_{(2)}$

- Um problema desta representação: números muito grandes ou muito pequenos necessitariam de uma sequência muito longa de bits...

Conversão da Base Decimal para Binária

Limitação

$(0{,}11)_{(10)} \approx 0{,}00011100001010001111010111000010100011110 1..._{(2)}$

Suponha que sejam admitidos somente 6 dígitos

$(0{,}11)_{(10)} \approx 0{,}000111_{(2)}$

Porém este número binário na base decimal não é igual a 0,11

$(0{,}11)_{(10)} \approx 0{,}000111_{(2)} = 0{,}109375_{(10)}$