CCI-22

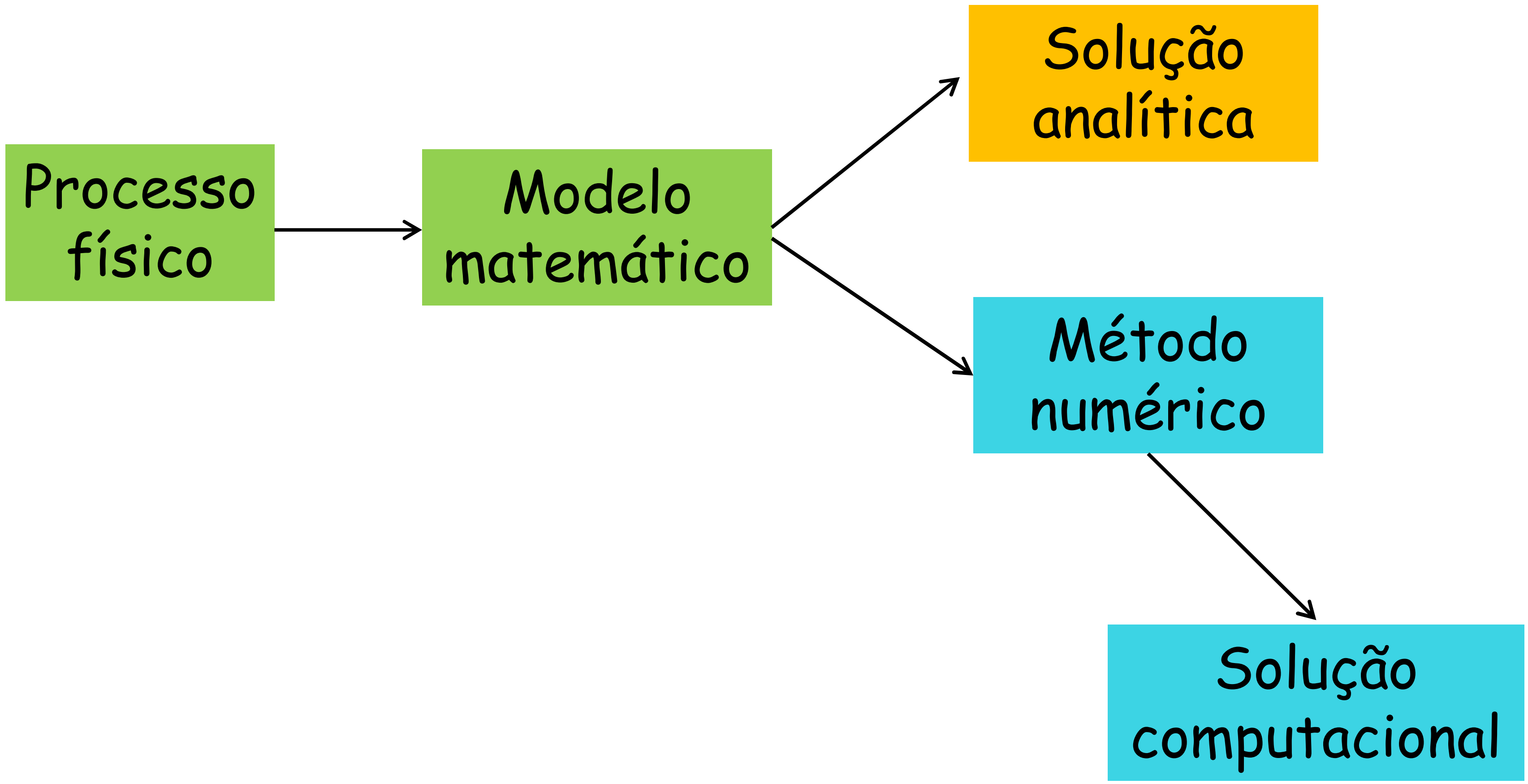
Matemática Computacional

Prof. Paulo André Castro
http://www.comp.ita.br/~pauloac
pauloac@ita.br
Sala 110 – Prédio da Computação

Objetivo – Matemática Computacional

- Fornecer ao aluno um conjunto de ferramentas ou métodos para a obtenção de uma *solução aproximada* de problemas matemáticos
- Exemplos: raízes de equações, interpolação de valores coletados, integração numérica, etc.
- Podem ser aplicados a um grande número de problemas numéricos que não possuem uma solução analítica (ou ela é muito difícil de obter) nas mais diversas áreas da engenharia
- Em muitas universidades, este curso costuma ser chamado de *Cálculo Numérico*

Resolução de Problemas com Métodos Numéricos



Justificativas

- Em alguns problemas, a resolução analítica é impraticável
  - Exemplo: sistemas lineares com muitas variáveis
- Há problemas que não podem ser resolvidos analiticamente
  - Exemplo: determinadas integrais e equações diferenciais
- Nos problemas reais, os dados são medidas físicas não exatas, com erros inerentes
  - É preciso considerar suas aproximações

Um caso real

- Em 04/06/1996, na Guiana Francesa, o lançamento do foguete Ariane 5 falhou por uma limitação da representação numérica (quantidade insuficiente de *bits*)
- Houve um erro na trajetória, 36,7 segundos após o lançamento, seguido de explosão
- Prejuízo: US$ 7,5 bilhões

Plano do curso

- Primeiro bimestre:
  - Representação numérica, erros e arredondamento
  - Ferramentas de suporte
  - Raízes de sistemas de equações (lineares e não lineares)
- Segundo bimestre:
  - Interpolação polinomial e ajuste de curvas
  - Integração e diferenciação numéricas

Avaliação

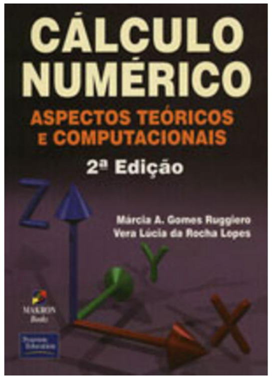
- Em cada bimestre:
  - 1 prova
  - 2 ou 3 exercícios de laboratório (trabalho individual)
- Pesos:
  - Prova: 50%
  - Média dos exercícios: 50%

Premissas éticas nos laboratórios

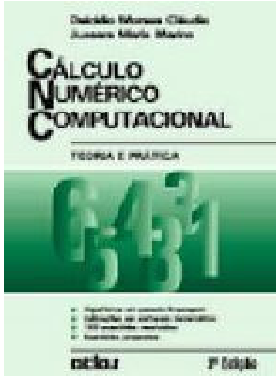
- É permitido:
  - Consultar material didático (*slides*, apostilas, códigos) de outros professores do ITA ou disponível na internet (neste último caso, se for código, sem fornecê-lo a outros colegas)
  - Pensar na solução junto com um colega, antes de programarem
  - Trocar ideias com outro colega, mas sem olhar o código que ele escreveu
  - Ajudar um colega a encontrar erros de codificação, desde que já tenha terminado o próprio laboratório
- Não é permitido:
  - Utilizar código pronto encontrado na internet
  - Olhar ou copiar soluções de outro aluno (da mesma turma ou de anteriores)
  - Fazer o exercício (mesmo parcialmente) de um colega com dificuldades
  - Escrever o código junto com outro colega

Bibliografia

- M.A.G. Ruggiero e V.L.R. Lopes
  Cálculo Numérico
  Aspectos Teóricos e Computacionais
  Pearson Makron Books

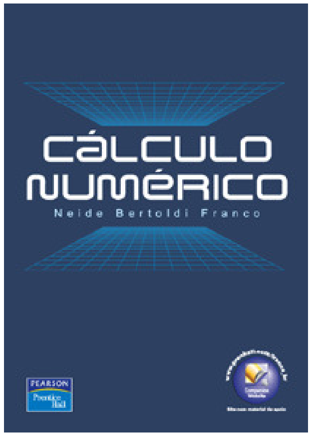


- D.M. Cláudio e J.M. Marins
  Cálculo Numérico Computacional
  Teoria e Prática
  Atlas



Bibliografia complementar

- N.B. Franco
  Cálculo Numérico
  Prentice-Hall



- S.C. Chapra e R.P. Canale
  Métodos Numéricos para Engenharia
  McGraw-Hill

CCI-22

1) Representações numéricas

Sistemas de Numeração, Mudanças de Base, Representações

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

Sistemas de numeração

- Base decimal
  - 10 dígitos disponíveis: 0, 1, 2, ..., 9
  - "Posição" indica a potência positiva de 10
  - Exemplo:
    - $5432 = 5.10^3 + 4.10^2 + 3.10^1 + 2.10^0$
- Base binária: é análogo
  - 2 dígitos (*<u>bi</u>nary digi<u>ts</u>*): 0, 1
  - "Posição" indica potência positiva de 2
  - Exemplo:
    - $1011_2 = 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 8+0+2+1 = 11_{10}$

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

Números Fracionários (deslocando a vírgula...)

Deslocamento em base 10 e em base 2

Base 10

15000,00	↑ × 10
1500,0	
150,0	
15,0	
1,5	
0,15	↓ ÷ 10
0,015	
0,0015	
0,00015	

Base 2

Base 2		Base 10
10000,0	↑ × 2	16
1000,0		8
100,0		4
10,0		2
1,0		1
0,1	↓ ÷ 2	0,5
0,01		0,25
0,001		0,125
0,0001		0,0625

Números fracionários

- Base decimal
  - Potência negativa de 10 para parte fracionária
  - Exemplo:
    - $54,32 = 5.10^1 + 4.10^0 + \mathbf{3.10^{-1}} + \mathbf{2.10^{-2}}$
- Base binária: também é análogo
  - Potência negativa de 2 para parte fracionária
  - Exemplo:
    - $(10,11)_2 = 1.2^1 + 0.2^0 + \mathbf{1.2^{-1}} + \mathbf{1.2^{-2}}$
    - $(10,11)_2 = 2 + 0 + \frac{1}{2} + \frac{1}{4} = (2,75)_{10}$
- Idem para outras bases: octal, hexadecimal, etc.

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

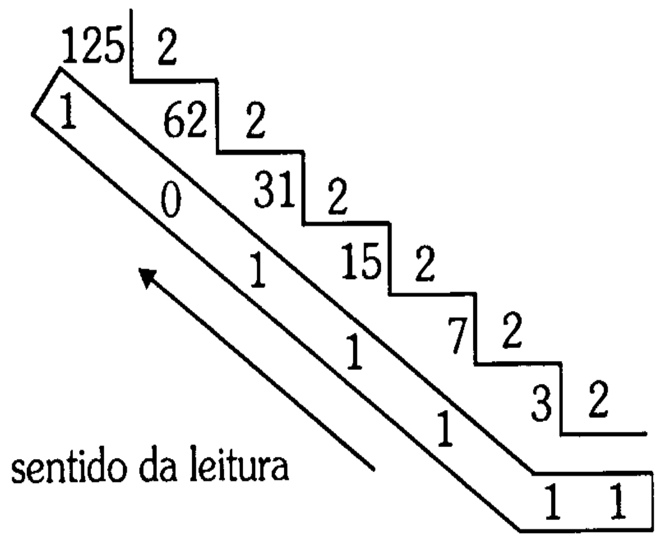
Conversão ou mudança de base

- Uma caixa alienígena com o número 25 gravado na tampa foi entregue a um grupo de cientistas. Ao abrirem a caixa, encontraram 17 objetos. Considerando que o alienígena tem um formato humanóide, quantos dedos deverá ter nas duas mãos?
- Solução:
  - $(17)_{10} = (25)_b$
  - $17 = 2.b^1 + 5.b^0$
  - $17 = 2b + 5$
  - $b = 6$

Outro exemplo

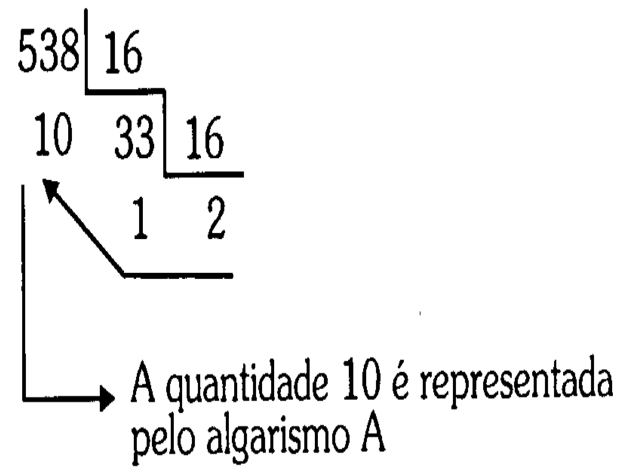
- Um *sistema de numeração ternário* tem três *trits*, que podem ter valor 0, 1 ou 2. Quantos *trits* são necessários para representar um número de seis *bits*?
- Solução:
  - $2^6 - 1 \leq 3^y - 1$
  - $6.\log_2 2 \leq y.\log_2 3$
  - $y = \lceil 6/\log_2 3 \rceil$
  - $y = 4$
  - Comprovando: $3^3=27 < 64 < 3^4=81$

Da base decimal para outra



$(125)_{10} = (1111101)_2$

$(538)_{10} = (\ ?\ )_{16}$



$(538)_{10} = (21A)_{16}$

Entre a base 2 e uma base 2^n

$(1011110010100111)_2 = (\ ?\ )_{16}$

$$\begin{array}{cccc} 1011 & 1100 & 1010 & 0111 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ B & C & A & 7 \end{array}$$

$$(1011110010100111)_2 = (BCA7)_{16}$$

$(A79E)_{16} = (\ ?\ )_2$

$$\begin{array}{cccc} A & 7 & 9 & E \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1010 & 0111 & 1001 & 1110 \end{array}$$

$$(A79E)_{16} = (1010011110011110)_2$$

Conversão de números fracionários

- Operação inversa: multiplicar por 2 a parte fracionária do número até que a parte fracionária do resultado seja zero
- Exemplo: converter $(0,625)_{10}$ para binário
  - 0,625 . 2 = 1,25: a primeira casa fracionária será 1, e a nova fração será 0,25
  - 0,25 . 2 = 0,5: a segunda casa fracionária será 0, e a nova fração será 0,5
  - 0,5 . 2 = 1,0: a terceira casa fracionária será 1, e a nova fração será zero
  - Resultado: $(0,625)_{10} = (0,101)_2$

Outro exemplo

$$(8{,}375)_{10} = (\ ?\ )_2$$

- parte inteira: $(8)_{10} = (1000)_2$
- parte fracionária:

0,375	0,750	0,500	0,000 → Final
x 2	x 2	x 2	
0,750	1,500	1,000	
↓	↓	↓	
0	1	1	

$$(8{,}375)_{10} = (1000{,}011)_2$$

Outros Exemplos...

$(0,625)_{10}$ para base 2:

0,625 * 2 = 1,25 -> 0,25
1 x 2
0,50 - > 0,5
0 x 2
1,0
1

Então $(0,625)_{10} = (0,101)_2$

$(0,3)_{10}$ para base 2:

0,3 * 2 = 0,6 -> 0,6
0 x 2
1,2 -> 0,2
1 x 2
0,4 -> 0,4
0 x 2
0,8 - > 0,8
0 x 2
1,6 -> 0,6
1

Então
$(0,3)_{10} = (0,0\ 1001\ 1001\ 1001\ 1001\ 1001\ 1001....)_2$

Exercícios

- Verificar:
  - $(5{,}8)_{10} = (101{,}11001100...)_2$, ou seja, é uma dízima
  - $(11{,}6)_{10} = (1011{,}10011001100...)_2$
    - Repare que a vírgula foi deslocada uma casa para a direita, pois 11,6 = 2 . 5,8
- Portanto, todo computador que trabalha com a base 2, como possui uma quantidade limitada de *bits*, armazenará uma aproximação para números como 5,8 ou 11,6
- Não se pode esperar resultados exatos em seus cálculos...

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

Representação de números inteiros

- No armazenamento de um número inteiro, os computadores utilizam geralmente uma quantidade fixa de m *bits*, chamada *palavra*
- O primeiro *bit* à esquerda representa o sinal, e os demais, o módulo do número
- Dentro desse esquema, há duas maneiras de representar os números inteiros:
  - Pelo módulo
  - Pelo complemento de 2

Representação pelo módulo

- O primeiro *bit* é o sinal, e os demais m-1 *bits* representam o módulo do número
- Exemplo para palavras com m = 4 *bits* :

$\boxed{(0\ 000)_2 = +0}$	$\boxed{(1\ 000)_2 = -0}$	$(0\ 100)_2 = +4$	$(1\ 100)_2 = -4$
$(0\ 001)_2 = +1$	$(1\ 001)_2 = -1$	$(0\ 101)_2 = +5$	$(1\ 101)_2 = -5$
$(0\ 010)_2 = +2$	$(1\ 010)_2 = -2$	$(0\ 110)_2 = +6$	$(1\ 110)_2 = -6$
$(0\ 011)_2 = +3$	$(1\ 011)_2 = -3$	$(0\ 111)_2 = +7$	$(1\ 111)_2 = -7$

- Problemas:
  - Duas representações para o zero
  - Incoerência nos cálculos
    $5 - 2 = 5 + (-2) = (0101)_2 + (1010)_2 = (1111)_2 = -7$

Representação pelo complemento de 2

- O primeiro *bit* continua sendo o sinal
- Os demais *bits* obedecem à seguinte regra:
  - Se o número for positivo, representarão o seu módulo
  - Exemplo: $(5)_{10} = (0101)_2$
  - Se o número for negativo, representarão seu módulo complementado e acrescido de 1
  - Exemplo: $(-5)_{10}$
    - Módulo: 101
    - Complemento: 010
    - Acréscimo de 1: 011
    - Portanto, $(-5)_{10} = (1011)_2$

<u>Ideia de fundo</u>:
ao serem somados, resultado final será $(0000)_2$

Representação pelo complemento de 2

- Exemplo para palavras com m = 4 *bits* :

$(0\ 000)_2 = +0$	$(0\ 100)_2 = +4$	$(1\ 000)_2 = -8$	$(1\ 100)_2 = -4$
$(0\ 001)_2 = +1$	$(0\ 101)_2 = +5$	$(1\ 001)_2 = -7$	$(1\ 101)_2 = -3$
$(0\ 010)_2 = +2$	$(0\ 110)_2 = +6$	$(1\ 010)_2 = -6$	$(1\ 110)_2 = -2$
$(0\ 011)_2 = +3$	$(0\ 111)_2 = +7$	$(1\ 011)_2 = -5$	$(1\ 111)_2 = -1$

- Valor de $(1xx...x)_2$: $(0xx...x)_2 - 2^{m-1}$
- Intervalo de representação: $[-2^{m-1}, 2^{m-1}-1]$
  - Zero e positivos: $[0, 2^{m-1}-1]$
  - Negativos: $[-2^{m-1}, -1]$

Exemplo: Complemento de 2 para 5 bits (com sinal)

N.o	Comp_2	N.o	Comp_2
+15	01111	-1	11111
+14	01110	-2	11110
+13	01101	-3	11101
+12	01100	-4	11100
+11	01011	-5	11011
+10	01010	-6	11010
+9	01001	-7	11001
+8	01000	-8	11000
+7	00111	-9	10111
+6	00110	-10	10110
+5	00101	-11	10101
+4	00100	-12	10100
+3	00011	-13	10011
+2	00010	-14	10010
+1	00001	-15	10001
0	00000	-16	10000

Conversão de Base:

- Se Positivo (isto é, bit de sinal igual a zero): Conversão Normal
- Se Negativo: Complementa número sem sinal, adiciona 1 e faz a Conversão Normal

CCI-22

- Sistemas de numeração
  - Bases: decimal, binária, etc.
  - Números fracionários
  - Mudanças de base
- Representação de números
  - Inteiros
  - Reais

Representação de números reais

- A representação de números reais é chamada de ponto flutuante (*float*), porque o ponto (a vírgula, em português) pode variar (ou flutuar) de posição conforme a potência da base
- Exemplo:
  - $54{,}32 = 54{,}32 \cdot 10^{0} = 5{,}432 \cdot 10^{1} = 0{,}5432 \cdot 10^{2} = 5432{,}0 \cdot 10^{-2}$

Representação em ponto flutuante

- Considere, por exemplo, o número $0,10111.b^{101}$:
  - $(0,10111)_2$: mantissa (ou significando)
  - $(101)_2$: expoente
- Representação genérica: $\pm 0,d_1d_2...d_n.b^{exp}$
  - n é o número de dígitos da mantissa
  - $d_1d_2...d_n$: mantissa, com $0 \leq d_i < b$ e $d_1 \neq 0$
  - exp: expoente (inteiro com sinal)
  - b: base numérica (geralmente é 2 nos computadores), que não precisa ser armazenada, pois é padrão em cada arquitetura

Armazenamento de *floats*

- Na arquitetura de cada computador, está definido:
  - A quantidade de *bits* da mantissa (é a sua precisão)
  - A quantidade de *bits* do expoente
  - Um *bit* de sinal
    - Geralmente, é o primeiro à esquerda
    - 0 é positivo e 1 é negativo
- Um exemplo com 8 *bits*:

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Sinal	Expoente (+/-)			Mantissa			