Métodos Numéricos - Notas de Aula

Prof^{a} Olga Regina Bellon

Junho 2007

1. Representação de números reais

1.1. Introdução

Cálculo Numérico X Método Numérico

1. Representação de números reais

1.1. Introdução

- ***Cálculo Numérico:***
  - Obtenção da solução de um problema pela aplicação de um método numérico.
  - A solução é caracterizada por um conjunto de números exatos ou aproximados.

1.1. Introdução

- ***Método Numérico***
  - Algoritmo composto por um número finito de operações.
  - Envolve apenas números:
    - Operações aritméticas elementares.
    - Cálculo de funções.
    - Consulta a tabela de valores.
    - Consulta a um gráfico.

1.1. Introdução

Problema Físico

↓ Modelagem

Modelo Matemático

↓ Resolução

Solução

1.1. Introdução

- ***Modelagem:***
  - Fase de obtenção do modelo matemático que descreve o comportamento do sistema.
- ***Resolução:***
  - Fase de obtenção da solução através da aplicação de métodos numéricos.

ERROS DE MODELAGEM

- Não se tem um descrição correta na representação de um fenômeno do mundo físico em um modelo matemático.
- Simplificações do mundo físico para obter o modelo matemático.

ERROS DE MODELAGEM

- **Exemplo:** Estudo do movimento de um corpo sujeito a a uma aceleração constante.

$$d = d_0 + v_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2$$

d = distância percorrida
d_0 = distância inicial
v_0 = velocidade inicial
t = tempo
a = aceleração

ERROS DE MODELAGEM

- Determinar a altura de um edifício com uma bolinha de metal e um cronômetro: 3s.

$$d = d_0 + v_0 * t + 1/2 * a * t^2$$

  - d = distância percorrida
  - d0 = distância inicial
  - v0 = velocidade inicial
  - t = tempo
  - a = aceleração
- Este resultado é confiável?

$$d = 0 + 0 \cdot 3 + \frac{1}{2} \cdot 9{,}8 \cdot 3^2 = 44{,}1\,m$$

ERROS DE MODELAGEM

- Fatores não considerados:
  - Resistência do ar.
  - Velocidade do vento.
  - ...
- Precisão dos dados de entrada:
  - Se o tempo fosse 3,5s --> d = 60,025m.
  - Variação de 16,7% no cronômetro --> 36% na altura.

ERROS DE REPRESENTAÇÃO

- Para resolução de modelos matemáticos muitas vezes é necessária a utilização de instrumentos de cálculo.
- Estes instrumentos necessitam de certas ***aproximações*** para o seu funcionamento.

ERROS DE REPRESENTAÇÃO

- Essas aproximações podem gerar erros:
  - Conversão de bases.
  - Erros de arredondamento.
  - Erros de truncamento.
  - ...

1.2. Representação: Números Reais

- Números
  - Complexos ($2+3\sqrt{-1}$)
  - Reais
    - Irracionais (π ; $\sqrt{2}$)
    - Racionais
      - Inteiros
      - Fracionários
        - Ordinários (32/7 ; 1/3)
        - Decimais (-3,15 ; 0,33...)

1.2. Representação: Números Reais

- Propriedades básicas da aritmética real não valem quando executadas no computador.
  - Matemática:
    - Alguns números são representados por infinitos dígitos.
  - Computador:
    - Palavra de memória finita, bem como a própria memória.
  - Exemplos: ($\sqrt{2}$; Π ; 1/3 ; $\sqrt{3}$).

1.2. Representação: Números Reais

- Jornada nas Estrelas – Cálculo de Π
- Cálculo da área de uma circunferência de raio 100m.
  - (a) $A = 31400\ m^2$
  - (b) $A = 31416\ m^2$
  - (c) $A = 31415,92654\ m^2$
    - Qual é o correto?

1.2. Representação: Números Reais

- Os erros ocorridos dependem da representação dos números na máquina utilizada.
- A representação depende:
  - Base disponível na máquina em uso.
  - Número máximo de dígitos usados na sua representação.
    - Π não pode ser representado por um número finito de dígitos decimais.

1.2. Representação: Números Reais

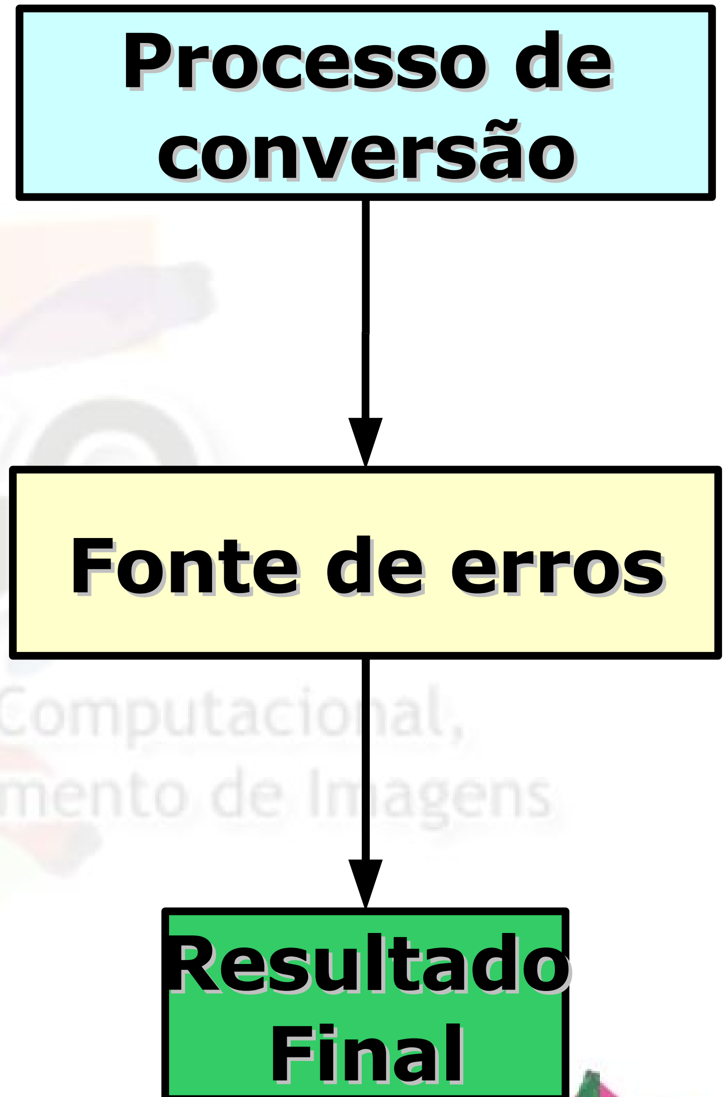
- Nos exemplos anteriores, Π foi escrito como:
  - 3,14 – caso (a)
  - 3,1416 – caso (b)
  - 3,141592654 – caso (c)
    - Resultados diferentes.
    - Erro depende exclusivamente da aproximação escolhida para Π.
    - A área nunca será exata.
      - Π é um número irracional.

1.2. Representação: Números Reais

- Cálculos com números infinitos não fornecem resultado exato.
- Quanto maior o número de dígitos, maior a precisão obtida.
  - Melhores aproximações para os cálculos das áreas – casos (c).
- Um número pode ter representação finita em uma base e não-finita em outra.
  - Base decimal: mais empregada.
  - Base binária: computadores.

Interação entre usuário e computador

- Dados de entrada são enviados ao computador pelo usuário no sistema decimal.
- Informação é convertida para o sistema binário onde são efetuadas as operações.
- Resultados finais são convertidos para o sistema decimal e então apresentados.



1.3. Sistema Decimal e Binário

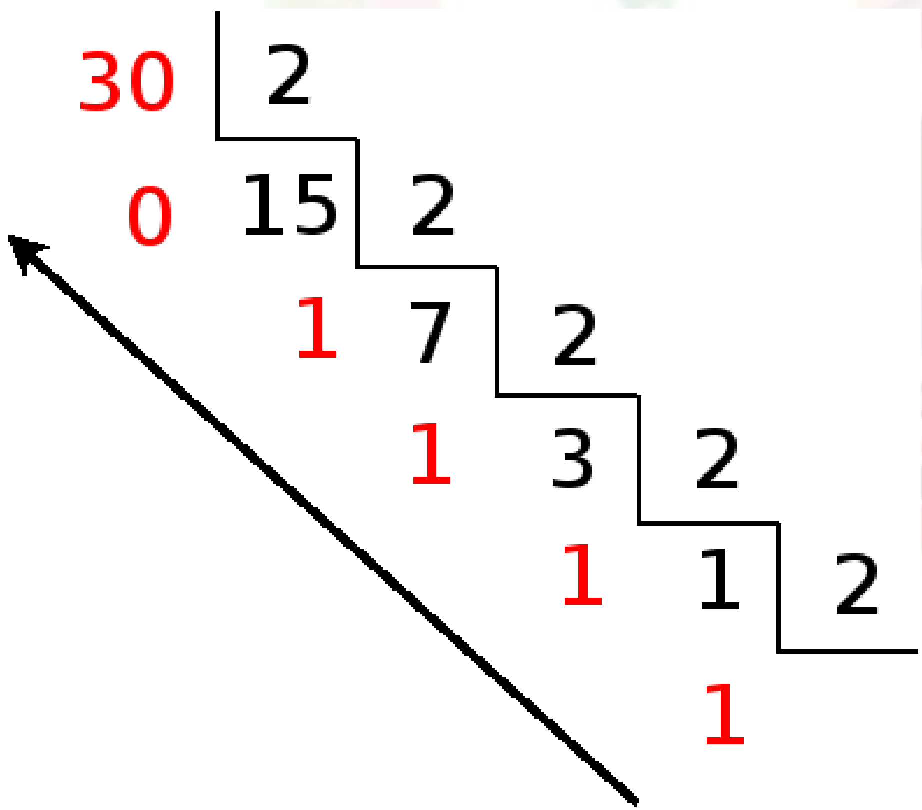
- **Sistema decimal.**
  - Dez dígitos – 0 , 1 , 2 , ... , 9.
  - 10 é a base do sistema.
  - Conveção atribui significado à posição ou lugar ocupado por um dígito.
    - $43087 = 4*10^4 + 3*10^3 + 0*10^2 + 8*10^1 + 7*10^0 = 40000 + 3000 + 0 + 80 + 7 = 43087$
    - Número expresso por uma soma de potências de 10 multiplicadas por coeficientes apropriados.

1.3. Sistema Decimal e Binário

- **Sistema binário**
  - Dois dígitos – 0 e 1.
  - 2 é a base do sistema.
  - Dígitos individuais representam os coeficientes de potências de 2.
    - 23 (decimal) = 10111 (binário)
    - $10111 = 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0$
      $= 16 + 0 + 4 + 2 + 1 = 23$
    - Número expresso por uma soma de potências de 2 multiplicadas por coeficientes apropriados.

1.3. Sistema Decimal e Binário

- Conversão de decimal para binário:
- $30_{(10)} = 11110_{(2)}$



Exemplo:

$212_{(10)} = 1322_{(5)}$

1.3. Sistema Decimal e Binário

- Conversão de um número fracionário da base 2 para a base 10.
  - $0{,}625_{10} = 6*10^{-1} + 2*10^{-2} + 5*10^{-3}$
    $= 0{,}6 + 0{,}02 + 0{,}005 = 0{,}625$
  - $0{,}101_{2} = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} = 0{,}5 + 0{,}125 = 0{,}625_{10}$

1.3. Sistema Decimal e Binário

- Conversão de um número fracionário da base 10 para a base 2.
  - Um número real entre 0 e 1 pode ter representação finita no sistema decimal, mas representação infinita no sistema binário.
- Seja r um número entre 0 e 1 no sistema decimal, e $(0,d_1d_2...d_j...)_2$ sua representação no sistema binário.
- Os dígitos binários $d_1d_2...d_j...$ são obtidos através do algoritmo a seguir.

1.3. Sistema Decimal e Binário

- Passo 0: $r_1 = r$; $k = 1$
- Passo 1: Calcule $2r_k$, Se $2r_k \geq 1$, $d_k = 1$, caso contrário, $d_k = 0$
- Passo 2: Faça $r_{k+1} = 2r_k - d_k$, Se $r_{k+1} = 0$, pare, caso contrário, Passo 3
- Passo 3: $k = k + 1$, volte ao Passo 1.

- O algoritmo pode ou não terminar após um número finito de passos $\Rightarrow$ número sem representação finita.

1.3. Sistema Decimal e Binário

- Exemplo: $r = (0{,}125)_{10} \Rightarrow (0{,}001)_2 \Rightarrow$ finita
  - Passo 0: $r_1 = 0{,}125$ e $k = 1$
  - Passo 1: $2r_1 = 0{,}25$; $d_1 = 0$
  - Passo 2: $r_2 = 0{,}25 - 0 = 0{,}25$
  - Passo 3: $k = 2$ Voltar ao passo 1
  - Passo 1: $2r_2 = 0{,}5$; $d_2 = 0$
  - Passo 2: $r_3 = 0{,}5 - 0 = 0{,}5$
  - Passo 3: $k = 3$ Voltar ao passo 1
  - Passo 1: $2r_3 = 1{,}0$; $d_3 = 1$
  - Passo 2: $r_4 = 1{,}0 - 1{,}0 = 0$ Pare

1.3. Sistema Decimal e Binário

- Observações:
  - Números sem representação finita no sistema binário podem gerar erros aparentemente inexplicáveis em cálculos efetuados em sistemas computacionais binários.

1.3. Sistema Decimal e Binário

- Observações:
  - Um computador no sistema binário armazena uma aproximação para $(0,1)_{10}$, pois possui um número fixo de posições para guardar os dígitos de mantissa de um número.
  - Portanto, não se pode esperar um resultado exato.

Números em máquina

- Formas de armazenamento dos números em máquina.
  - Ponto fixo: **valores inteiros**.
  - Ponto flutuante: **valores reais**.

1.4. Aritmética de ponto flutuante

- Ponto flutuante

$$x = \pm\left[\frac{d_1}{b_1} + \frac{d_2}{b_2} + \frac{d_3}{b_3} + \ldots + \frac{d_t}{b_t}\right] * b^e$$

1.4. Aritmética de ponto flutuante

Onde:

- b = base onde a máquina opera.
- d_i = números inteiros contidos no intervalo. $0 \leq d_i \leq (b-1)$ i = 1, 2, ... , t $d_1 \neq 0$
- e = expoente de b, assume valores entre $I \leq e \leq S$
- I e S = limite inferior e superior do expoente
- $\left[\dfrac{d_1}{b_1} + \dfrac{d_2}{b_2} + \dfrac{d_3}{b_3} + \ldots + \dfrac{d_t}{b_t}\right]$ = mantissa.
- t = número de dígitos significativos (precisão da máquina).

1.4. Aritmética de ponto flutuante

Exemplos:

- No sistema decimal tem-se:

$$0,625_{10} = \left(\frac{6}{10^1} + \frac{2}{10^2} + \frac{5}{10^3} \right) * 10^0$$

$$3,1415_{10} = 0,31415 * 10^1 = \left(\frac{3}{10^1} + \frac{1}{10^2} + \frac{4}{10^3} + \frac{1}{10^4} + \frac{5}{10^5} \right) * 10^1$$

- Os números acima estão normalizados $\Rightarrow$ mantissa é um valor entre 0 e 1.

- Forma normalizada $\Rightarrow$ operações em ponto flutuante em computadores digitais.

1.4. Aritmética de ponto flutuante

Exemplos:

- No sistema binário tem-se:

$$21_{10} = 10101_2 = 0{,}10101 * 2^5 = \left(\frac{1}{2^1} + \frac{0}{2^2} + \frac{1}{2^3} + \frac{0}{2^4} + \frac{1}{2^5}\right) * 2^5$$

$$4_{10} = 100_2 = 0{,}1 * 2^3 = \frac{1}{2} * 2^3$$

- Os números acima estão normalizados $\Rightarrow$ mantissa é um valor entre 0 e 1.

1.4. Aritmética de ponto flutuante

Exemplos:

- Numa máquina de calcular cujo sistema de representação utilizado tenha:
  - $b = 2$
  - $t = 10$
  - $I = -15$
  - $S = 15$
- Determinar a representação do número 41_{10}, sem sinal, mantissa e expoente.

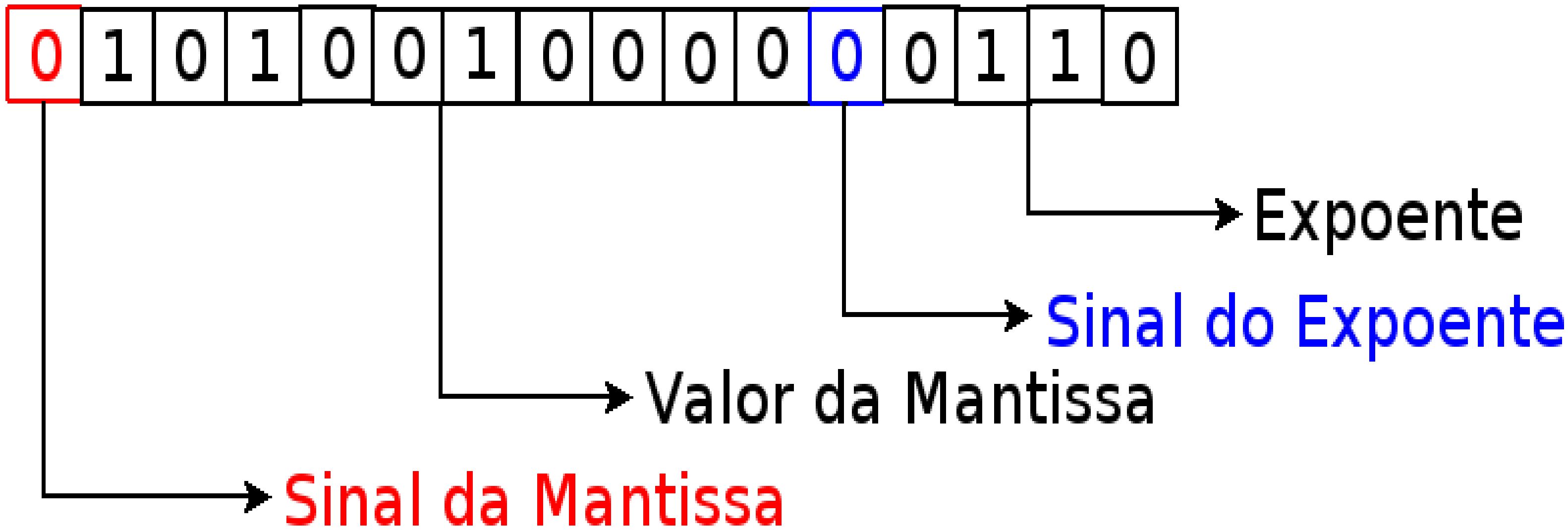
1.4. Aritmética de ponto flutuante

$$41_{10} = 101001_2 = 0{,}101001 * 2^6 = 0{,}101001 * 2^{110}$$

$$\left(\frac{1}{2^1}+\frac{0}{2^2}+\frac{1}{2^3}+\frac{0}{2^4}+\frac{0}{2^5}+\frac{1}{2^6}+\frac{0}{2^7}+\frac{0}{2^8}+\frac{0}{2^9}+\frac{0}{2^{10}}\right)*2^{110}$$

- Ou de forma mais compacta:
  - 1010010000 $\Rightarrow$ mantissa
  - 0110 $\Rightarrow$ expoente

1.4. Aritmética de ponto flutuante



- Palavras de 16 bits
  - 10 bits para a mantissa
  - 4 bits para o expoente
  - 1 bit para sinal da mantissa
    - 0 = positivo, 1 = negativo
  - 1 bit para sinal do expoente

1.4. Aritmética de ponto flutuante

- Maior valor representado por uma palavra de 16 bits

0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

  - Base decimal: $0,1111111111 * 2^{1111} = 32736$

- Menor valor

1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

  - Base decimal: $-0,1111111111 * 2^{1111} = -32736$

- Intervalo: $\left[-32736_{10} ; 32736_{10}\right]$

1.4.Aritmética de ponto flutuante

- Valor de zero

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Próximo número positivo

0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1

  - Base decimal: $0{,}1 * 2^{-15} = 0{,}000015259$

- O subseqüente

0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1

  - Base decimal: $0{,}1000000001 * 2^{-15} = 0{,}000015289$

1.4. Aritmética de ponto flutuante

- Exemplo:
  - Máquina opera no sistema: b = 10; t = 4; e $\in$ [-5 , 5]
- Menor número (m), em valor absoluto:
  - $m = 0,1000 * 10^{-5}$
- Maior número (M), em valor absoluto:
  - $M = 0,9999 * 10^{5}$

1.4. Aritmética de ponto flutuante

- Considere o conjunto de números reais *R* e o seguinte conjunto:

$$G = x \in R | m \leqslant |x| \leqslant M$$

- Dado um número real *x*, várias situações podem ocorrer:

1.4. Aritmética de ponto flutuante

- 1) $x \in G \rightarrow \quad x = 2007{,}99 = 0{,}200799 * 10^4$
- Estão representados nesta máquina os números:
  - $0{,}2007 * 10^4$
  - $0{,}2008 * 10^4$
- Com truncamento:
  - $0{,}2007 * 10^4$
- Com arredondamento:
  - $0{,}2008 * 10^4$

1.4. Aritmética de ponto flutuante

- 2) | *x* | < m: por exemplo, x = 0,7389 * 10^{-6}
  - Não há representação deste número nesta máquina.
    - O expoente *e* é menor que -5.
    - Nesta situação a máquina acusa a ocorrência de underflow.

1.4. Aritmética de ponto flutuante

- 3) | *x* | > M: por exemplo, $x = 0{,}1010 * 10^9$
  - Não há representação deste número nesta máquina.
    - O expoente *e* é maior que 5.
    - Nesta situação a máquina acusa a ocorrência de overflow.

1.4. Aritmética de ponto flutuante

- Algumas linguagens de programação permitem a declaração de variáveis em precisão dupla.
  - Aritmética de ponto flutuante da máquina.
  - Com o dobro de dígitos disponíveis na mantissa.
  - Precisão dupla aumenta.
    - tempo de execução.
    - requerimento de memória.