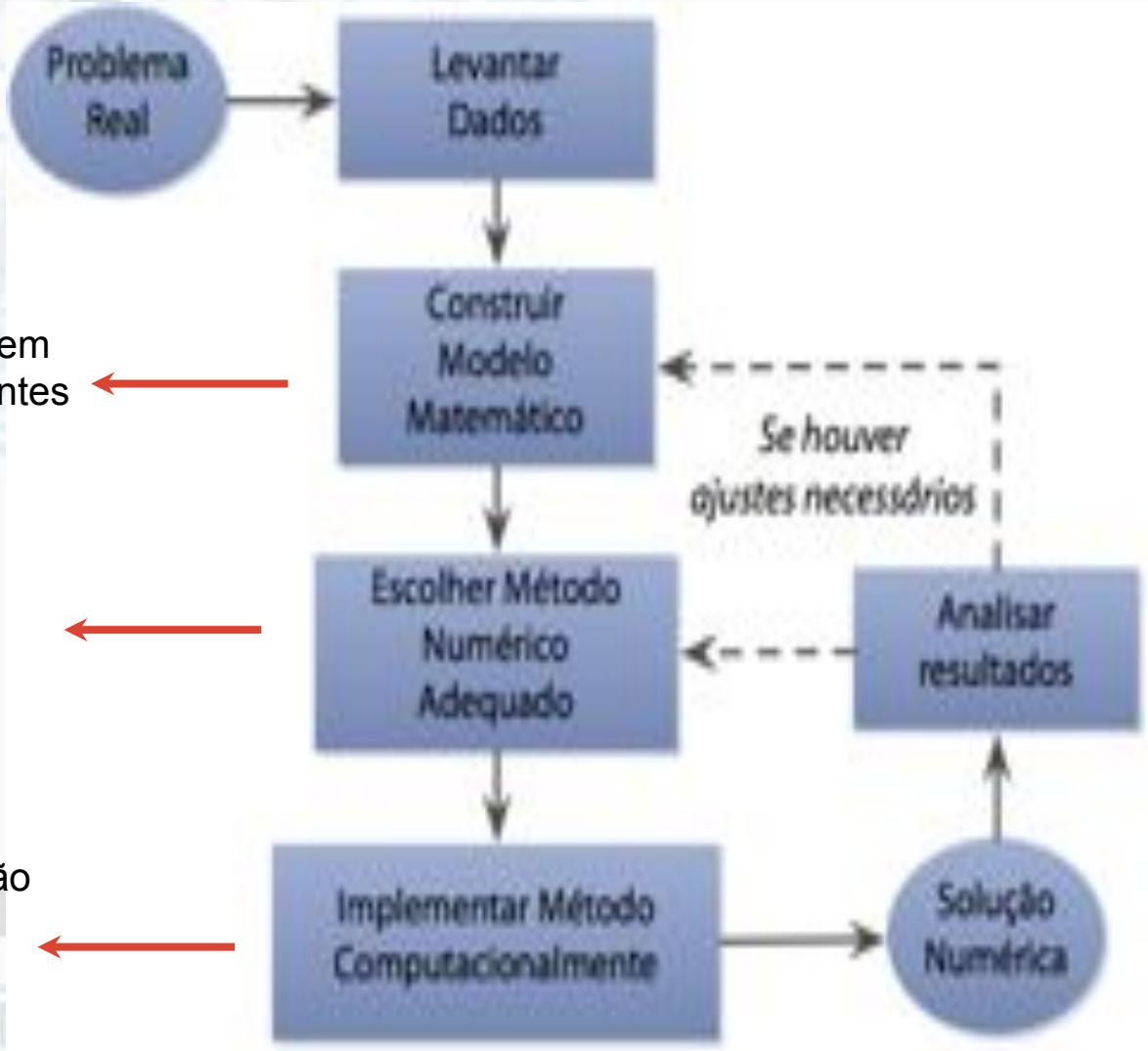
Métodos Numéricos 1 (MN1)

Unidade 1: Teoria dos Erros
Parte 1: Introdução a Erros Numéricos

Joaquim Bento Cavalcante Neto
joaquimb@lia.ufc.br

Grupo de Computação Gráfica, Realidade Virtual e Animação (CRAb)
Departamento de Computação (DC)
Universidade Federal do Ceará (UFC)

Erros em Métodos Numéricos



Erros de modelagem (ex. erros decorrentes de simplificações)

Erros de aproximação

Erros de resolução (ex. limitação do equipamento)

Importância de Erros

- Consequências catastróficas da ignorância a erros ao usar métodos numéricos:
  - Falha no lançamento de mísseis *Patriot:*
    - Guerra do Golfo, 25-fevereiro-1991
    - 28 soldados americanos morreram
    - Outras 100 pessoas ficaram feridas
    - Causa: cálculo errôneo do tempo devido à representação numérica
      - **Uso de 24 bits no cálculo**
      - **Erro de 0.34 segundos**

http://www.ima.umn.edu/~arnold/disasters/patriot.html

Importância de Erros

- Consequências catastróficas da ignorância a erros ao usar métodos numéricos:
  - Explosão do foguete Ariane 5:
    - Guiana Francesa, 04-junho-1996
    - Foguete não-tripulado explodiu aproximadamente 37 segundos após o seu primeiro lançamento
    - Causa: limitação na representação numérica: 64 bits => 16 bits
    - Prejuízo de 7,5 bilhões de dólares

  (mais de 10 anos de desenvolvimento)

http://www.ima.umn.edu/~arnold/disasters/ariane.html

Importância de Erros

- Consequências catastróficas da ignorância a erros ao usar métodos numéricos:
  - Afundamento da plataforma marítima Sleipner A:
    - Noruega, 23-agosto-1991
    - Uma das células que compunham a parede se rompeu, resultando num grande vazamento na plataforma
    - Causa: combinação de erro em análise de elementos finitos e ancoragem insuficiente em zona crítica
    - Prejuízo de quase $1 bilhão

http://www.ima.umn.edu/~arnold/disasters/sleipner.html

Representação de Números

- Soluções de problemas são dependentes da representação numérica utilizada na solução
- Exemplos:
  - Cálculo da área *A* de uma circunferência de raio 100m
    - Resultados dependentes do valor de π utilizado
      - **A = 31400 m^2 ou A = 31416 m^2 ou A = 31415.92654 m^2**
    - É possível calcular a área exata?
    - Qual é o resultado mais preciso?

Representação de Números

- Exemplos
  - Somatório no computador: $S = \sum_{i=1}^{30000} x_i$, para $x_i = 0.11$:

Solução real:

S = 3300

```cpp
#include <iostream>
#include <iomanip>

using namespace std;
int main(int argc, const char * argv[])
{
    float S = 0;
    for (int i=0; i < 30000; i++) {
        S += 0.11;
    }
    cout << setiosflags(ios::fixed) << setprecision(6) << S;
    return 0;
}
```

Em C++
precisão simples:
S = 3300.985107

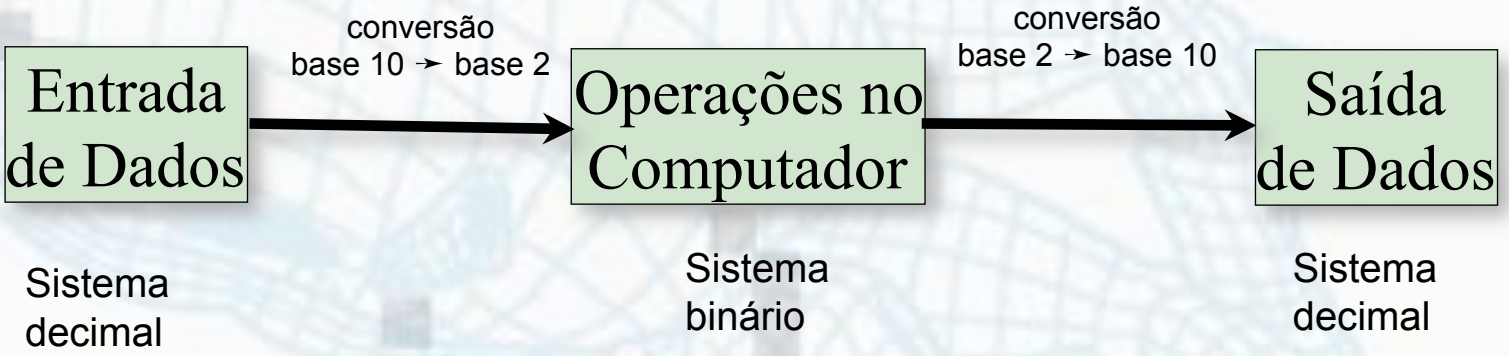
```
>>> S = 0>>> for x in
range(0,30000):...     S
+=0.11... >>>
S3300.0000000006285
```

Em Python
precisão dupla
S = 3300.0000000006285

Representação de Números

- Cálculo envolvendo números que não podem ser representados por número finito de dígitos não dá resultado exato
  - ⬆ o número de dígitos utilizados ⇒ ⬆ precisão
- Um número pode ter representação finita em uma base e não-finita em outras bases
  - Nós usamos o sistema decimal
  - Computador opera no sistema binário

Interação com o Computador



- Mudanças de bases podem introduzir erros que afetam os resultados finais dos cálculos

Definição de bases

- Pode-se representar um número real N, em qualquer base **b**, da seguinte forma:

$$N_b = \sum_{i=n}^{m} a_i \times b^i$$

- onde $a_i \in \{0, 1, 2, 3, ..., (b-1)\}$, com n e m inteiros

Base Binária

$$N_b = \sum_{i=n}^{m} a_i \times b^i$$

$$N_2 = \sum_{i=n}^{m} a_i \times 2^i, a_i \in \{0,1\}$$

- Exemplos:

$$(1011)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$$

$$(1011.01)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}$$

Base Decimal

$$N_b = \sum_{i=n}^{m} a_i \times b^i$$

$$N_{10} = \sum_{i=n}^{m} a_i \times 10^i, a_i \in \{0, 1, \ldots, 9\}$$

- Exemplos:

$$(231)_{10} = 2 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$$

$$(231.35)_{10} = 2 \times 10^2 + 3 \times 10^1 + 1 \times 10^0 + 3 \times 10^{-1} + 5 \times 10^{-2}$$

Mudanças de Base

- Base binária para base decimal (parte inteira):
  - Procedimento: multiplicar o dígito binário por uma potência adequada de 2 e depois somar números obtidos, ou seja, calcular o somatório da fórmula

$$\sum_{i=n}^{m} a_i \times 2^i$$

$$(1011)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = (11)_{10}$$

$$(1011)_2 = 1\times 8 + 0\times 4 + 1\times 2 + 1\times 1 = (11)_{10}$$

$$(1011)_2 = 8 + 0 + 2 + 1 = (11)_{10}$$

Mudanças de Base

- Base binária para base decimal (parte fracionária):
  - Procedimento: multiplicar o dígito binário por uma potência adequada de 2 e depois somar números obtidos, ou seja, calcular o somatório da fórmula

$$\sum_{i=n}^{m} a_i \times 2^i$$

$$(.01)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} = (0.25)_{10}$$

$$(.01)_2 = 0 \times 0.5 + 1 \times 0.25 = (0.25)_{10}$$

$$(.01)_2 = 0 + 0.25 = (0.25)_{10}$$

Mudanças de Base

- Base binária para base decimal (parte inteira):
  - Alternativa: usar parênteses (usar 2 em evidência)

$$(1011)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = (11)_{10}$$

$$(1011)_2 = 2\times(2\times(2\times 1+0)+1)+1 = (11)_{10}$$

$$b_3 = a_3 = 1 \Rightarrow b_j = a_j$$

$$b_2 = 2\times 1+0 = 2\times b_3 + a_2 = 2 \Rightarrow b_{j-1} = 2\times b_j + a_{j-1}$$

$$b_1 = 2\times 2+1 = 2\times b_2 + a_1 = 5 \Rightarrow b_1 = 2\times b_2 + a_1$$

$$b_0 = 2\times 5+1 = 2\times b_1 + a_0 = 11 \Rightarrow b_0 = 2\times b_1 + a_0$$

Mudanças de Base

- Base binária para base decimal (parte fracionária):
  - Alternativa: usar multiplicações sucessivas

1) Multiplicamos o número fracionário dado por $(10)_{10} = (1010)_2$
2) Do resultado do passo 1), a parte inteira é o primeiro dígito decimal, ou seja, o primeiro dígito considerado
3) Do resultado do passo 2), a parte fracionária é novamente multiplicada por 10, obtendo-se o próximo dígito decimal
4) O processo continua até que a parte fracionária seja nula

Mudanças de Base

- Base binária para base decimal (parte fracionária):
  - Alternativa: usar multiplicações sucessivas

$$r_1 = (.01)_2 = (0.25)_{10}$$

$$w_1 = (1010)_2 \times r_1 = 10.10 \Rightarrow b_1 = (10)_2 = (2)_{10} \Rightarrow r_2 = (.10)_2$$

$$w_2 = (1010)_2 \times r_2 = 101.0 \Rightarrow b_2 = (101)_2 = (5)_{10} \Rightarrow r_2 = (0)_2$$

1010	1010
0.01	0.10
1 0 . 1 0	0 0 . 0 0
0 0 0 . 0	1 0 1 . 0
0 0 0 0 .	0 0 0 0 .
0 0 1 0 . 1 0	0 1 0 1 . 0 0

Mudanças de Base

- Base decimal para base binária (parte inteira):
  - Procedimento: dividir o número decimal sucessivamente por 2 (divisão sucessiva)

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

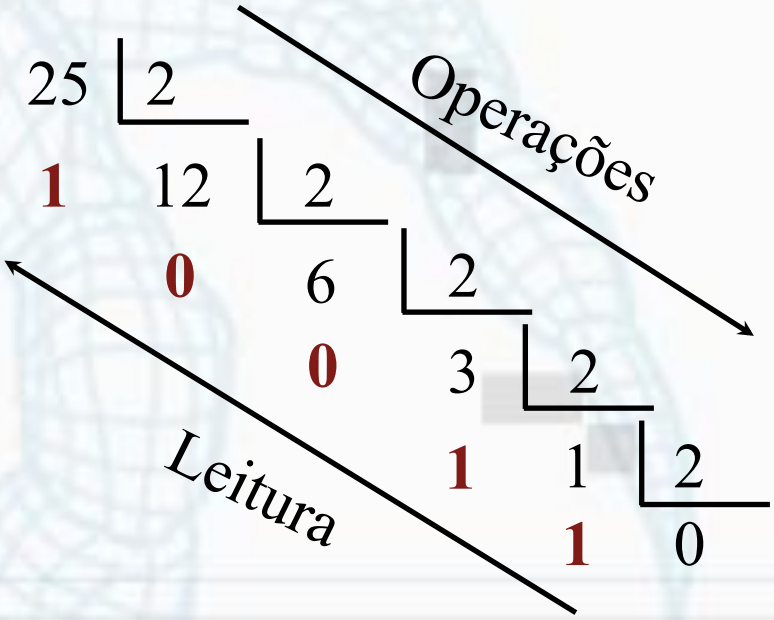
$$25 = 2 \times 12 + 1 \Rightarrow a_0 = 1 \Rightarrow a_0 = 1$$

$$12 = 2 \times 6 + 0 \Rightarrow a_1 = 0 \Rightarrow a_1 = 0$$

$$6 = 2 \times 3 + 0 \Rightarrow a_2 = 0 \Rightarrow a_{j-2} = 0$$

$$3 = 2 \times 1 + 1 \Rightarrow a_3 = 1 \Rightarrow a_{j-1} = 1$$

$$1 = 2 \times 0 + 1 \Rightarrow a_4 = 1 \Rightarrow a_j = 1$$



Mudanças de Base

- Base decimal para base binária (parte fracionária):
  - Procedimento: usar multiplicações sucessivas

1) Multiplicamos o número fracionário considerado por 2
2) Do resultado do passo 1), a parte inteira é o primeiro dígito binário, ou seja, o primeiro dígito considerado
3) Do resultado do passo 2), a parte fracionária é novamente multiplicada por 2, obtendo-se o próximo dígito binário
4) O processo continua até que a parte fracionária seja nula

Mudanças de Base

- Base decimal para base binária (parte fracionária):

$$(0.1875)_{10} = (0.0011)_2$$

(0.1875) x 2 = 0.375 ➛ parte inteira = 0 e parte fracionária 0.375
(0.375) x 2 = 0.75 ➛ parte inteira = 0 e parte fracionária 0.75
(0.75) x 2 = 1.5 ➛ parte inteira = 1 e parte fracionária 0.5
(0.5) x 2 = 1.0 ➛ parte inteira = 1 e parte fracionária 0

$$(13.25)_{10} = (13)_{10} + (0.25)_{10} = (1101)_2 + (0.01)_2 = (1101.01)_2$$

$$(0.11)_{10} = (0.00011100001\ldots)_2$$