UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Campus de São José dos Campos
Instituto de Ciência e Tecnologia

Análise de Erros em Ponto Flutuante

Prof. Dr. Rogério Galante Negri

Erros na representação numérica

- Nos computadores não é possível representar todos números de $[a, b] \subseteq \mathbb{R}$

- Logo, resultados de operações aritméticas ou cálculo de funções <u>podem conter erros</u>

- Imprecisões nos cálculos (modo geral):
  - Simplificação do modelo matemático
  - Erro de truncamento ($\pi = 3.1415926$)
  - Erros de arredondamento ($\pi = 3.141593$)
  - Erro durante coleta de dados (campo, laboratório, etc...)

Podem destruir a precisão dos resultados!

Erros na representação finita

- Consideremos a área da circunferência de raio 100 m

Aproximação para π	**Área (m^2)**
3.14	31400
3.1416	31416
3.141592654	31415.92654

- Todos os cálculos acima apresentam erros
- Tais erros dependem da aproximação de π
- Em vista dos resultados, a última aproximação é a melhor

Cálculos que envolvem números com representação infinita (e.g., $\pi, e, \sqrt{2}, \frac{7}{3}, \ldots$) não fornecem resultados exatos

- Existem ainda números com representação finita em uma determinada base, porém infinita em outra
  - A base decimal é a mais adotada, porém o computador usa a base binária

$\mathbb{Z}$ e conversão binário→decimal

- Não há dificuldades na representação dos inteiros
- Os computadores trabalham em uma base $\beta = 2$
- Seja $n \in \mathbb{Z}$ e $n \neq 0$, sua representação é:

$$n = \pm\left(n_j n_{j-1} \cdots n_1 n_0\right)_\beta = \pm(n_j \beta^j + n_{j-1}\beta^{j-1} + \cdots + n_0\beta^0)$$

sendo $0 \leq n_i < \beta$ e $n_j \neq 0$

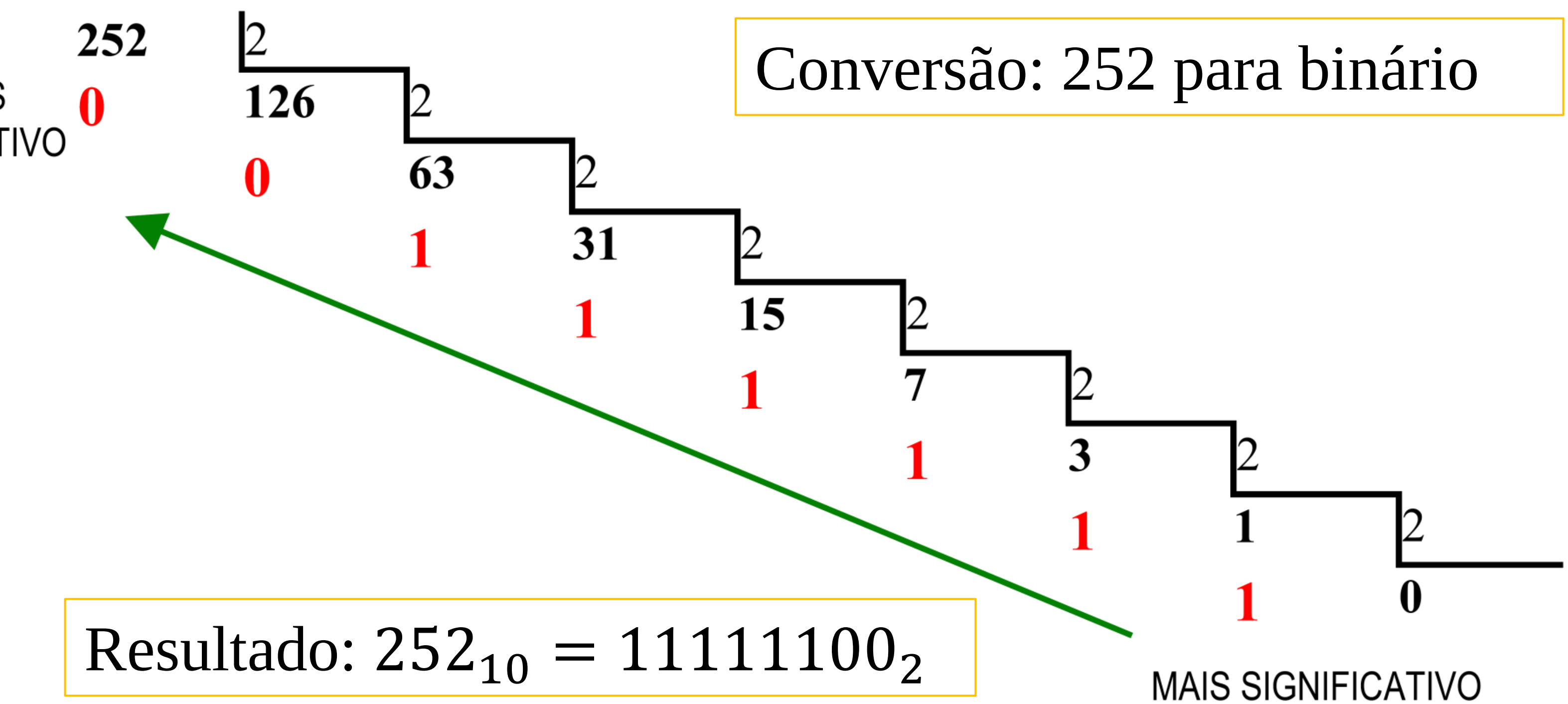
- Exemplos:
  - $(1234)_{10} = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$
  - $(100101)_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 37$

A representação "polinomial" proporciona a conversão entre qualquer base β para a base decimal

Conversão decimal→binário

- A conversão de um número na base decimal para a base binária é obtida por sucessivas divisões por 2, com quocientes inteiros
- O valor convertido corresponde aos restos das divisões

Conversão: 252 para binário



Resultado: $252_{10} = 11111100_2$

Conversão decimal→binário

```
%Conversão de números inteiros em base 10 para base binária

nbase10 = input('Informe um número inteiro pos. na base 10: ');

nbase2 = [-1]; %sera desconsiderado
while (nbase10 > 1)
    aux = mod(nbase10,2);
    nbase10 = fix(nbase10/2);
    nbase2 = [aux,nbase2];
end

if (nbase10 == 1)
    nbase2 = [1,nbase2];
else
    nbase2 = 0;
end

disp(['Representação binária: ',num2str(nbase2(1:end - 1))]);
```

Conversão racional → binário

- Seja um número n escrito na base decimal
- Suponhamos que n tenha <u>representação finita</u>

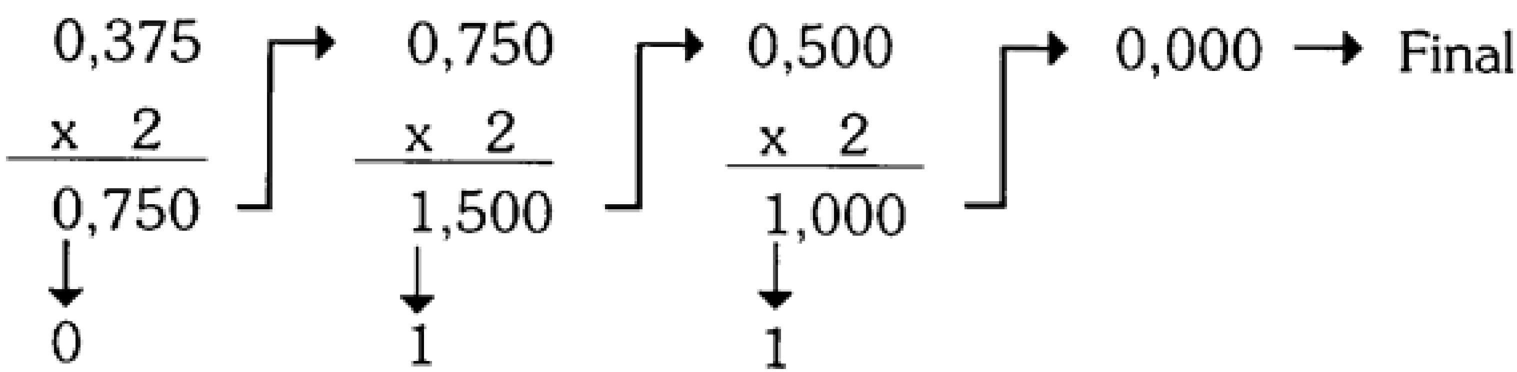
$$n = \underbrace{n_j n_{j-1} \cdots n_0}_{\text{parte inteira}}, \underbrace{n_{-1} n_{-2} \cdots n_{-i}}_{\text{parte fracionária}}$$

- A conversão da parte inteira é conhecida
- Outra estratégia é adotada na conversão da parte fracionária: <u>multiplicações sucessivas</u>

Conversão racional→binária

- A conversão da parte fracionária, da base decimal para a base binária, consiste em multiplicar a parte fracionária da base decimal por 2, até que a parte fracionária seja nula (ou quase).
- Vejamos o esquema, para conversão binária do número $8{,}375_{10}$

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



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

Exercício: Converta $13{,}5625_{10}$ para a base binária.

Conversão racional → binário

```
%Conversão de números fracionários da base decimal para binária

numReal = input('Informe um número real positivo na base 10: ');
precisao = input('Precisão da parte fracionária na conversão:' );

intePart = fix(numReal);    fracPart = numReal - intePart;

%Conversão da parte inteira
nbase2 = [-1]; %será desconsiderado
while (intePart > 1)
    aux = mod(intePart,2);
    intePart = fix(intePart/2);   nbase2 = [aux , nbase2];
end
if (intePart == 1) nbase2 = [1,nbase2];   else   nbase2 = 0;   end

%Conversão da parte fracionária
fra2 = [-1]; %será desconsiderado
for it = 1:1:precisao
    if (fracPart == 0); %caso a conversão seja finita
        fra2 = '0';         break;
    end
    conv = fix(fracPart * 2);    fracPart = (fracPart * 2) - conv;
    fra2 = [fra2 , conv];
end

disp(['Representação binária: ', num2str(nbase2(1:end-1)), ' . ' , ...
num2str(fra2(2:end))]);
```

Erros de conversão

- Considerando $(0.1)_{10}$, sua representação binária é:

$$(0.1)_{10} = (0.000110011001100110011\ldots)_2$$

- Logo, $(0.1)_{10}$ não possui representação binária finita
- No exemplo dado, o computador armazenará uma aproximação da representação
- Qualquer cálculo envolvendo tal número pode deixar de ser exato

Representação do $\mathbb{R}$

- Existem duas formas de representação
  - Ponto Fixo (computadores antigos)

$$x = \pm \sum_{i=k}^{t} x_i \beta^{-i}$$

$$k, t \in \mathbb{Z}$$
$$k \leq 0 < t$$
$$0 \leq x_i < \beta$$

  - Ponto Flutuante (computadores atuais)

$$x = \pm(0.d_1 d_2 \cdots d_t) \times \beta^e$$

β: base do sistema

d: mantissa

e: expoente $\mathbb{Z}$

t: *dígitos da mantissa*

Se $x \neq 0$, então $d_1 \neq 0$, com $0 \leq d_i < (\beta - 1)$, $i = 1,2,\ldots,t$

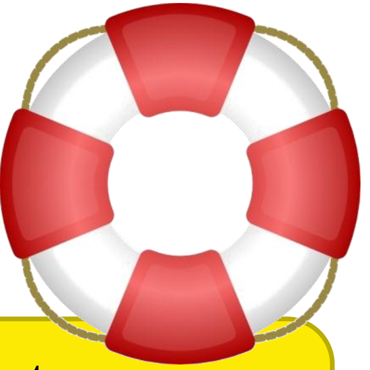
t define a "precisão"

$-m \leq e \leq M$

Notação: $\boldsymbol{F(\beta, t, m, M)}$

Importância da representação em ponto flutuante

- Em determinados cálculos, a faixa de variação dos números é grande
  - Imagine um problema que envolva a massa do elétron ($9 \times 10^{-28}\ g$) e a massa do Sol ($2 \times 10^{33}\ g$)
  - Neste caso, a faixa de variação excede 10^{60}! (34 à esquerda do ponto e 28 à direita)
- É possível projetar um computador para fazer cálculos nesta precisão
- Na verdade, poucas medidas podem ser feitas nessa precisão
- Logo, considerando essa precisão
  - No final são descartados 50 ~ 60 desses dígitos de precisão
  - Além do desperdiço de memória e tempo de processamento



Diante de problemas como este torne-se necessário um sistema para representação dos números que independa da quantidade de dígitos significativos

Exemplo: ponto flutuante

- Considerando um sistema que:

$$\beta = 10 \quad ; \quad t = 3 \quad ; \quad e \in \{-5, -4, \ldots, 4{,}5\}$$

temos $G = \{x \in \mathbb{R} : menor \leq |x| \leq maior\}$, onde:

  - $menor = 0.100 \times 10^{-5}$
  - $maior = 0.999 \times 10^{5}$

- Seja $x = 456{,}789 = 0{,}456789 \times 10^{3} \in G$

6 dígitos na mantissa

As representações possíveis são:

$0{,}456 \times 10^{3}$ se usado truncamento

$0{,}457 \times 10^{3}$ se usado arredondamento

- Seja $x = 0{,}123 \times 10^{-6} < m$, ocorre *underflow*

Não há representação

- Seja $x = 0{,}123 \times 10^{6} > M$, ocorre *overflow*

Erros absoluto e relativo

- Erro absoluto é a diferença entre o valor exato de um número n e sua aproximação $\bar{n}$, denotada por:

$$EA_n = n - \bar{n}$$

...porém, n é geralmente desconhecido

- Para isso, é considerado um limitante superior para o módulo do Erro Absoluto

- Seja $\pi \in (3.14, 3.15)$, temos $|EA_\pi| = |\pi - \bar{\pi}| < 0.01$

É uma característica do computador!
Está diretamente ligada a sua precisão

Erro absoluto e relativo

- Sejam $\bar{x} = 2112.9$ e $\bar{y} = 5.3$, considerando como limitantes superiores $|EA_x| = |EA_y| < 0.1$
- Logo, $\bar{x}$ é uma aprox. de $x \in (2112.8, 2113)$, e $\bar{y}$ é uma aprox. de $y \in (5.2, 5.4)$

- A precisão nas representações não são as mesmas, pois <u>depende da ordem de grandeza dos números</u> considerados

- Para isso é adotado o erro relativo:

$$ER_n = \frac{EA_n}{\bar{n}}$$

- Nos exemplos: $|ER_x| = \frac{|EA_x|}{|\bar{x}|} < \frac{0.1}{2112.9} \approx 4.7 \times 10^{-5}$ $\quad |ER_y| < \frac{0.1}{5.3} \approx 0.02$

logo, a aproximação de x é melhor que a de y

Arredondamento e Truncamento

- A representação do número no computador depende da base numérica, dígitos na mantissa e da faixa do expoente

- Considerando ponto flutuante e mantissa de t dígitos:

Podemos escrever qualquer número como:

$$x = f \times 10^{e} + g \times 10^{e-t}$$

com $0.1 \leq f < 1$ e $0 \leq g < 1$

- Por exemplo, para $x = 1234567.89$, considerando $t = 4$:

$$x = 0.1234 \times 10^{7} + \underline{0.56789 \times 10^{3}}$$

  - Como considerar esta parte na mantissa?
  - Quais efeitos sobre os erros relativo e absoluto?

Truncamento

- A parcela $g \times 10^{p-t}$ é desprezada, logo:

$$\bar{x} = f \times 10^{e}$$

- Com isso, temos:

$0 \leq g < 1$

$$|EA_x| = |x - \bar{x}| = \boxed{|g|} \times 10^{e-t} < 10^{e-t}$$

$$|ER_x| = \frac{|EA_x|}{|\bar{x}|} = \frac{|g| \times 10^{e-t}}{\boxed{|f|} \times 10^{e}} < \frac{10^{e-t}}{0.1 \times 10^{e}} = 10^{-t+1}$$

$0.1 \leq f < 1$

Considere $g = 1$ e $f = 0.1$, o que fornece um limite superior (pior caso!)

Arredondamento

- O valor de f é modificado para considerar g:

$$\bar{x} = \begin{cases} f \times 10^{e} & \text{se} |g| < 1/2 \\ f \times 10^{e} + 10^{e-t} & \text{se } |g| \geq 1/2 \end{cases}$$

- Se $|g| < 0.5$:

$$|EA_x| = |x - \bar{x}| = |g| \times 10^{e-t} < 0.5 \times 10^{e-t}$$

$$|ER_x| = \frac{|EA_x|}{|\bar{x}|} < \frac{0.5 \times 10^{e-t}}{0.1 \times 10^{e}} = 0.5 \times 10^{-t+1}$$

- Se $|g| \geq 0.5$:

$$|EA_x| = |x - \bar{x}| = |(f \times 10^{e} + g \times 10^{e-t}) - (f \times 10^{e} + 10^{e-t})| =$$
$$= |g - 1| \times 10^{e-t} \leq 0.5 \times 10^{e-t}$$

$$|ER_x| = \frac{|EA_x|}{|\bar{x}|} \leq \frac{0.5 \times 10^{e-t}}{|f \times 10^{e} + 10^{e-t}|} < \frac{0.5 \times 10^{e-t}}{|f| \times 10^{e}} < \frac{0.5 \times 10^{e-t}}{0.1 \times 10^{e}} =$$
$$= 0.5 \times 10^{-t+1}$$

Erros nas operações aritméticas

- Erros são inerentes nas operações aritméticas
- Os erros se propagam/acumulam ao longo das operações

Exemplo: $x = 0.567 \times 10^4$, $y = 0.153 \times 10^2$ e $t = 3$ (precisão)

Adição: $x + y = (0.567 + 0.00153) \times 10^4 = 0.56853 \times 10^4$

Arredon.: $\overline{x+y} = 0.569 \times 10^4$ Truncado: $\overline{x+y} = 0.568 \times 10^4$

Multiplicação: $xy = 0.86751 \times 10^5$

Arredon.: $\overline{xy} = 0.868 \times 10^5$ Truncado: $\overline{xy} = 0.867 \times 10^6$

Os resultados (OP) das operações são normalizados e arredondados/truncados em t dígitos

É obtido o resultado aproximado $(\overline{OP})$

- Os erros relativos serão:
  - $|ER_{\overline{OP}}| < 10^{-t+1}$ quando realizado truncamento
  - $|ER_{\overline{OP}}| < 0.5 \times 10^{-t+1}$ quando realizado arredondamento

Erros Op. Aritm. – Adição

- Consideremos a adição:

$$x + y = (\bar{x} + EA_x) + (\bar{y} + EA_y) =$$
$$= (\bar{x} + \bar{y}) + (EA_x + EA_y)$$

- Logo, o Erro Absoluto é $EA_{x+y} = EA_x + EA_y$

- Por sua vez:

$$ER_{x+y} = \frac{EA_{x+y}}{\bar{x} + \bar{y}} = \frac{EA_x + EA_y}{\bar{x} + \bar{y}} =$$
$$= \frac{EA_x}{\bar{x} + \bar{y}} + \frac{EA_y}{\bar{x} + \bar{y}} = \frac{EA_x}{\bar{x} + \bar{y}}\left(\frac{\bar{x}}{\bar{x}}\right) + \frac{EA_y}{\bar{x} + \bar{y}}\left(\frac{\bar{y}}{\bar{y}}\right) = \frac{EA_x}{\bar{x}}\left(\frac{\bar{x}}{\bar{x} + \bar{y}}\right) + \frac{EA_y}{\bar{y}}\left(\frac{\bar{y}}{\bar{x} + \bar{y}}\right) =$$
$$= ER_x\left(\frac{\bar{x}}{\bar{x} + \bar{y}}\right) + ER_y\left(\frac{\bar{y}}{\bar{x} + \bar{y}}\right)$$

- Logo, o Erro Absoluto é $ER_{x+y} = ER_x\left(\frac{\bar{x}}{\bar{x}+\bar{y}}\right) + ER_y\left(\frac{\bar{y}}{\bar{x}+\bar{y}}\right)$

Erros Op. Aritm. – Subtração

- Consideremos a subtração:

$$x - y = (\bar{x} + EA_x) - \left(\bar{y} + EA_y\right) =$$
$$= (\bar{x} - \bar{y}) + (EA_x - EA_y)$$

- Logo, o Erro Absoluto é $EA_{x-y} = EA_x - EA_y$

- Desenvolva o Erro Relativo...

Erros Op. Aritm. – Multiplicação

- Consideremos a multiplicação:

$$xy = (\bar{x} + EA_x)(\bar{y} + EA_y) =$$
$$= \overline{xy} + \bar{x}EA_y + \bar{y}EA_x + \boxed{EA_xEA_y}$$

Descartado, por ser muito pequeno

- Logo, o Erro Absoluto é $EA_{xy} \approx \bar{x}EA_y + \bar{y}EA_x$

- Desenvolva o Erro Relativo...

Erros Op. Aritm. – Divisão

- Consideremos a divisão:

$$\frac{x}{y} = \frac{(\bar{x} + EA_x)}{(\bar{y} + EA_y)} = \frac{(\bar{x} + EA_x)}{\bar{y}} \cdot \left(\frac{1}{1 + \frac{EA_y}{\bar{y}}} \right)$$

- Representando: $\left(\frac{1}{1+\frac{EA_y}{\bar{y}}}\right) = \sum_{i=0}^{n} \left(-\frac{EA_y}{\bar{y}}\right)^i = 1 - \left(\frac{EA_y}{\bar{y}}\right) + \cancel{\left(\frac{EA_y}{\bar{y}}\right)^2 - \cdots}$

temos:

$$\frac{x}{y} \approx \frac{(\bar{x} + EA_x)}{\bar{y}} \cdot \left(1 - \left(\frac{EA_y}{\bar{y}}\right)\right) = \frac{\bar{x}}{\bar{y}} + \frac{EA_x}{\bar{y}} - \frac{\bar{x}EA_y}{\bar{y}^2} - \cancel{\frac{EA_xEA_y}{\bar{y}^2}}$$

- Logo, o Erro Absoluto é $EA_{x/y} = \frac{EA_x}{\bar{y}} - \frac{\bar{x}EA_y}{\bar{y}^2} = \frac{\bar{y}EA_x - \bar{x}EA_y}{\bar{y}^2}$

- Desenvolva o Erro Relativo...

Exercício (da lista)

- Seja um sistema que adota aritmética de ponto flutuante de quatro dígitos de precisão, base decimal e acumulador de precisão dupla (faça os cálculos com o dobro de dígitos, mas arredonde o resultado).

  Calcule as <u>operações</u> e obtenha o <u>erro relativo</u> no resultado:

  a) $x + y + z$

  b) $\frac{x}{y}$

  Considerando,

  - $x = 0.7237 \times 10^4$
  - $y = 0.2145 \times 10^{-3}$
  - $z = 0.2585 \times 10^1$

  representados exatamente no sistema (não há erro na representação).

Exercícios (da lista)

- Suponha que x é representado no computador por $\bar{x}$, onde $\bar{x}$ é obtido por arredondamento. Obtenha os limitantes superiores para os erros relativos de $u = 2\bar{x}$ e $w = \bar{x} + \bar{x}$.

Bibliografia da aula

- RUGGIERO, M. A. G.; LOPES, V. L. R. **Cálculo Numérico - Aspectos Teóricos e Computacionais**, 2ª Ed. Pearson, 1996.
- FRANCO, N. B. **Cálculo Numérico**. Pearson, 2007.