# 1. Noções básicas de erros

Professor: Wemerson D. Parreira.

parreira@univali.br

Universidade do Vale do Itajaí
Escola Politécnica

2023

# Introdução ao Cálculo Numérico

## O que é?

➪ O Cálculo Numérico corresponde a um conjunto de **ferramentas ou métodos** usados para se obter a solução de problemas matemáticos de forma **aproximada**.

➪ **Por que produzir resultados numéricos?**

# Introdução ao Cálculo Numérico

## O que é?

➪ O Cálculo Numérico corresponde a um conjunto de **ferramentas ou métodos** usados para se obter a solução de problemas matemáticos de forma **aproximada**.

➪ **Por que produzir resultados numéricos?**

# Introdução ao Cálculo Numérico

## O que é?

➪ O Cálculo Numérico corresponde a um conjunto de **ferramentas ou métodos** usados para se obter a solução de problemas matemáticos de forma **aproximada**.

**➪ Por que produzir resultados numéricos?**

## Podemos recorrer a soluções numéricas quando?

- Problemas que possuem soluções analíticas simples mas com o aumento da dimensão do problema a solução analítica fica impraticável.

  Ex.: *solução de sistemas de equações lineares.*

- Não existem métodos matemáticos para solução analítica do problema.

  Ex.:

  a. $\int e^{x^2} dx$ (função sem primitiva na forma simples),
  b. *equações diferenciais parciais não lineares que podem ser resolvidas analiticamente apenas em casos particulares.*

## Podemos recorrer a soluções numéricas quando?

- Problemas que possuem soluções analíticas simples mas com o aumento da dimensão do problema a solução analítica fica impraticável.

  Ex.: *solução de sistemas de equações lineares.*

- Não existem métodos matemáticos para solução analítica do problema.

  Ex.:

  a. $\int e^{x^2} dx$ (função sem primitiva na forma simples),
  b. *equações diferenciais parciais não lineares que podem ser resolvidas analiticamente apenas em casos particulares.*

## Podemos recorrer a soluções numéricas quando?

- Problemas que possuem soluções analíticas simples mas com o aumento da dimensão do problema a solução analítica fica impraticável.

  Ex.: *solução de sistemas de equações lineares.*

- Não existem métodos matemáticos para solução analítica do problema.

  Ex.:

  a. $\int e^{x^2} dx$ (função sem primitiva na forma simples),
  b. *equações diferenciais parciais não lineares que podem ser resolvidas analiticamente apenas em casos particulares.*

I. O Cálculo Numérico tem por objetivo o estudo de esquemas numéricos (algoritmos numéricos) para resolução de problemas que podem ser representados por um modelo matemático.

II. Os esquemas numéricos nos fornecem **aproximações** para o que seria a solução exata do problema.

III. Um esquema é **eficiente** quando esse apresenta soluções dentro de uma **precisão desejada** com **esforço computacional (tempo de execução + memória) baixo**.

IV. Os **erros cometidos** nesta aproximação são decorrentes da **discretização** do problema, ou seja passar do modelo matemático para o esquema numérico, e da forma como as máquinas representam os dados numéricos.

# Fatores que podem influenciar os resultados

1. A representação de números em máquinas digitais (calculadoras, computadores, etc) é feita na forma de ponto flutuante com um número finito de dígito. Logo os números que tem representação infinita, por exemplo,

$$1.3333...,\ 2/11,\ \pi, e...$$

são representados de **forma truncada**.

2. Devido a representação finita algumas das propriedades da aritmética real não valem na aritmética computacional. Como exemplo, na aritmética computacional temos:

$$\sum_{k=1}^{n} \frac{a_0}{N} \neq \frac{1}{N} \sum_{k=0}^{n} a_k.$$

➪ Do ponto de vista analítico, as duas expressões são equivalentes, mas a segunda forma apresenta melhor resultado do ponto de vista computacional, pois realiza menos operações e comete menos erro de truncamento.

3. Tipo de máquina em que estamos trabalhando.

   Ex.: Numa calculadora simples a representação numérica e as operações são feitas usando 7 dígitos, enquanto que calculadoras mais avançadas, internamente estas calculadoras trabalham com mais dígitos do que e apresentado no visor e antes do resultado ser apresentado este é arredondado.

4. Os métodos iterativos e o critério de parada nos métodos iterativos. No geral, algoritmos diferentes apresentarão resultados diferentes. Em um dado algoritmo, critérios de parada distintos produzirão resultados com precisão distintas.

   ➪ Discutiremos mais detalhadamente no decorrer do curso.

5. Linguagem de programação usada na implementação dos algoritmos (Pascal, Fortran, C++, MatLab , etc). Diferentes linguagens podem apresentar diferentes resultados.

6. Diferença de compiladores. Mesmo quando usamos uma mesma linguagem, mas compiladores diferentes (Ex. C++ da Borland e C++ da Microsoft), os resultados podem apresentar diferenças. Existem várias bibliotecas de rotinas numéricas em diversas linguagens e algumas disponíveis na Internet.

# Etapas para resolução de um problema

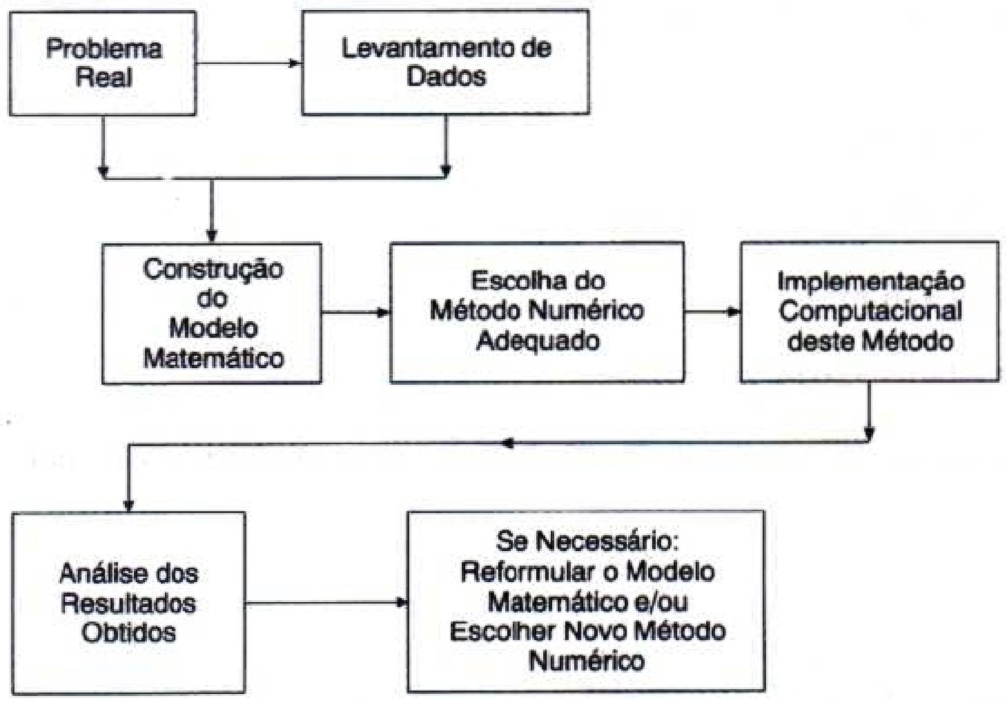


Figura: Etapas para resolução de um problema, por M. Rudiero e V. A. Lopes. *"Calculo Numérico: Aspectos Teóricos e Computacionais"*.

- **Definição do problema:** Nesta etapa, define-se qual é o problema real a ser resolvido.

  Ex.: Calcular $\sqrt{a}$, $a > 0$ usando apenas as 4 operações aritméticas.

- **Modelagem matemática:** O problema real é transformado no problema original por meio de uma formulação matemática.

  Ex.:

$$x = \sqrt{a} \rightarrow x^2 = a \rightarrow f(x) = x^2 - a = 0.$$

  O problema real foi transformado no problema original que é determinar a raiz de uma equação do segundo grau.

- **Solução numérica:** Nesta etapa, escolhe-se o método numérico para resolver o problema original. Pode-se dividir esta etapa em outras 3.

- **Etapas da solução numérica:**

  1. **Elaboração do algoritmo:** Um algoritmo é a descrição de um conjunto de comandos que, quando ativados, resultam em uma sucessão finita de acontecimentos. Apenas os detalhes matemáticos são levados em consideração.

  2. **Codificação:** Esta é a fase de implementação do algoritmo. Nesta fase deve-se preocupar com os aspectos da linguagem adotada.
     Ex.: MATLAB, Pascal, FORTRAN, C e etc.

  3. **Processamento do programa:** O código do programa é editado em um arquivo que possa ser executado pelo computador. Se detectado algum erro de sintaxe ocorrido durante a fase de codificação o programa deverá ser corrigido.

- **Avaliação dos resultados:** Nesta fase deve-se avaliar se os resultados encontrados estão de acordo com o esperado. Caso negativo, deve-se avaliar se existe algum erro de lógica, então deve-se voltar a fase de elaboração do algoritmo e corrigi-lo ou mudar o algoritmo escolhido.

# Representação Numérica

- Considere o seguinte problema: Calcule a área de um região circular com raio $r = 100$ m. Lembrando que a fórmula para o cálculo de área de uma região circular é: $A_c = \pi r^2$ m$^2$.
  - $A_c = 31400$ m$^2$
  - $A_c = 31416$ m$^2$
  - $A_c = 31415,92654$ m$^2$

➪ Por que foram encontrados valores diferentes se o problema é o mesmo?

➪ Qual é o valor exato?

# Erros

- **Erro Absoluto:** é a medida da diferença absoluta entre o valor exato $x$ e o valor aproximado $\tilde{x}$.

$$\boxed{EA_x = |x - \tilde{x}|}$$

Como em geral apenas $\tilde{x}$ é conhecido, o que obtemos é um limite superior ou uma estimativa.

**Ex.1:** $\pi \in (3,14;\ 3,15)$ assim $|EA_\pi| = |\pi - \tilde{\pi}| < 0,1$.

**Ex.2:** Sejam $\tilde{x} = 2344,9$ tal que $|EA_x| = 0,1$ e $\tilde{y} = 0,234$ tal que $|EA_y| = 0,1$.

➪ Qual desses valores está representado com maior precisão?

- **Erro Relativo:** é a razão entre o Erro Absoluto e o valor aproximado $\tilde{x}$.

$$\boxed{ER_x = \frac{EA_x}{\tilde{x}} = \frac{|x - \tilde{x}|}{\tilde{x}}}$$

**Ex.3:** Para o Ex. 2 então fazemos:

$$ER_x = \frac{0,1}{2344,9} = 4,26 \times 10^{-5}$$

e

$$ER_y = \frac{0,1}{0,234} = 4,27 \times 10^{-1}.$$

Portanto $x$ está representado com maior precisão por $\tilde{x}$.

- Qualquer cálculo que envolva números que não podem ser representados a partir de um número finito de dígitos não fornecerá como resultado um valor exato.

- Quanto maior o número de dígitos utilizados melhor será a precisão obtida.

- O número de dígitos necessários para representar uma quantidade (valor numérico qualquer) dependerá da base usada. Um valor número pode ter uma representação finita em uma base e infinita em outra. A base comumente usada atualmente é a decimal.

- A relação homem-máquina pode gerar erros pois:

usuário

$\downarrow$

$\boxed{\text{decimal}}$

$\downarrow$

$\boxed{\text{decimal} \rightarrow \text{binário}} \rightarrow \boxed{\text{processamento binário}} \rightarrow \boxed{\text{binário} \rightarrow \text{decimal}}$

$\downarrow$

$\boxed{\text{decimal}}$

$\downarrow$

usuário

- Cada um dos símbolos usados pelo computador é denominado *bit*, assim como no sistema decimal esse símbolo é denominado *dígito*.

- A álgebra usada pelo computador é denominada *álgebra de Boole* ou *Álgebra Binária*.

# Revisão: Conversão Binário $\rightarrow$ Decimal

- **Decimal:**

  O dígito menos significativo (da unidade) possui o seu valor (de $0$ a $9$) multiplicado por $10^0\,(1)$, o da casa das dezenas possui seu valor multiplicado por $10^1\,(10)$, o da casa das centenas multiplicado por $10^2\,(100)$ e assim por diante, cada vez aumentando o expoente da base $10$.

  Ex.: $98735_{10} = 9 \times 10^4 + 8 \times 10^3 + 7 \times 10^2 + 3 \times 10^1 + 5 \times 10^0$.

- **Binário:**
  O bit menos significativo possui o seu valor ($0$ ou $1$) multiplicado por $2^0$ ($1$), o próximo possui seu valor multiplicado por $2^2$ ($2$) e assim por diante, cada vez aumentando o expoente da base $2$.
  Ex.:
  $1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8_{10} + 2_{10} + 1_{10} = (11)_{10}$.
  $1,01_2 = 1 \times 2^0 + 0 \times 2^{(-1)} + 1 \times 2^{(-2)} = 1_{10} + 0,25_{10} = (1,25)_{10}$.

  Exercício: Converta os seguintes números para a base 10:
  (a) $(1001,1)_2$ (*binário*)
  (b) $(125,76)_8$ (*octal*)
  (c) $(123AF,A)_{16}$ (*hexadecimal*)

| | | |
|---|---|---|
| $(A)_{16} = 10_{10}$ | $(B)_{16} = 11_{10}$ | $(C)_{16} = 12_{10}$ |
| $(D)_{16} = 13_{10}$ | $(E)_{16} = 14_{10}$ | $(F)_{16} = 15_{10}$ |

# Revisão: Conversão Binário $\leftarrow$ Decimal

Qualquer número da forma $N = (a_n a_{n-1} \ldots a_2 a_1 a_0, b_{-1} b_{-2} \ldots b_{-k})_{10}$
pode ser separado em uma parte inteira e uma parte decimal
$N = (a_n a_{n-1} \ldots a_2 a_1 a_0)_{10} + (0, b_{-1} b_{-2} \ldots b_{-k})_{10}$

- Parte inteira: (Divisões sucessivas)
  Ex.: $(23)_{10}$

$$23/2 = 11 \text{ com resto } 1$$

$$11/2 = 5 \text{ com resto } 1$$

$$5/2 = 2 \text{ com resto } 1$$

$$2/2 = 1 \text{ com resto } 0$$

Portanto, $(23)_{10} = (10111)_2$.

- Parte fracionária: (Multiplicações sucessivas)
  Ex.:$(0,375)_{10}$

$$0,375 \times 2 = 0,750$$

$$0,750 \times 2 = 1,5$$

$$0,5 \times 2 = 1,0$$

  Critério de parada: a multiplicação por 2 resulta em um número inteiro (1) ou o limite de casas decimais atingido.
  Portanto, $(0,375)_{10} = (0,011)_2$

➪ Para converter um número $N$ qualquer para outra base $\beta$ qualquer basta substituir 2 por $\beta$.

**Exercício:** Converta os seguintes valores decimais para binários

a. $(7,5)_{10} =$

b. $(18,125)_{10} =$

c. $(0,1)_{10} =$

d. $(0,11)_{10} =$

e. $(3,7)_{10} =$

# Aritmética de ponto flutuante

A Representação pode variar ("flutuar") a posição da vírgula, ajustando potência da base.

**Exemplo [Sistema Decimal]:**
$54,32 = 54{,}32 \times 10^{0} = 5{,}432 \times 10^{1} = 0{,}5432 \times 10^{2} = 5432{,}0 \times 10^{-2}$
Forma normalizada usa um único dígito antes da vírgula, diferente de zero: $5,432 \times 10^{1}$.

➪ O Sistema computacional de aritmética de ponto flutuante é o sistema utilizado por calculadoras e computadores para representação e execução das operações.

**Exemplo [Sistema Binário]:**
$110101 = 110{,}101 \times 2^{3} = 1{,}10101 \times 2^{5} = 0{,}0110101 \times 2^{7}$

➪ No caso dos números serem armazenados em um computador, os expoentes serão também gravados na base dois, porém não será usada simplesmente a conversão decimal – binário.

➪ A principal vantagem da representação em ponto flutuante é que ela pode representar uma grande faixa de números se comparada a representação de ponto fixo.

➪ A representação em ponto flutuante permite representar uma faixa muito maior de números. O preço a ser pago é que esta representação tem quatro dígitos de precisão, em oposição à representação por ponto fixo que possui 6 dígitos de precisão.

**Definições:**

No número $1,10101 \times (10)^{101}$ a sequência de dígitos $(0,10101)$ é denominada significando (ou mantissa) e a sequência $(101)$ é denominada expoente.

**Observação:** Na base binária, o "1" antes da vírgula, na representação normalizada – se esta for adotada, também pode ficar implícito, economizando um bit (*bit escondido*).

## Representação genérica:

$$\pm\, d_0,\, d_1\, d_2\, \ldots\, d_t \times (\beta)^q$$

em que
$t$ é o número de dígitos da mantissa;
$d_1 d_2 \ldots d_t$ é a mantissa, com $0 \leq d_i \leq (\beta - 1)$;
$q$ é o expoente (inteiro com sinal).

➪ Adota-se para a base binária que uma sequência está normalizada quando $d_0 = 1$. Para o sistema decimal, ou qualquer outra base, consideraremos uma sequência normalizada quando $d_0 = 0$ e $d_1 \neq 0$.

# Exercícios Sugeridos:

Bibliografia: Chapra, S.C., "Métodos Numéricos Aplicados com Matlab para Engenheiros e Cientistas". Mc Graw Hill, 3a. ed.

➪ Disponível na Biblioteca Online

Problemas: **4.1, 4.2, 4.3, 4.9,**

Página: 121.