Matemática Computacional

Introdução

Felipe Augusto Lima Reis
felipe.reis@ifmg.edu.br

INSTITUTO FEDERAL
Minas Gerais

Sumário

Conceitos

Definição

Cálculo numérico é a disciplina que estuda as técnicas, de natureza analítica e computacional, para solução aproximada de problemas matemáticos [Justo et al., 2020].

Aplicações

- O cálculo numérico é utilizado para:
  - Solução numérica de problemas matemáticos para diversas áreas científicas e tecnológicas;
  - Criação de modelos que reproduzem fenômenos reais que possuem solução analítica difícil (ou impossível) mesmo quando foi provada a existência de solução
    - Nesses casos podem ser utilizadas aproximações numéricas que, apesar de diferentes da solução analítica, podem resultar em resultados próximos o suficiente do desejado.

Aplicações

- Solução de problemas que analiticamente são complexos de serem resolvidos
  - Exemplo 1:

$$x^7 + 2x^5 + 3x^4 - 2x^3 - x^2 + x + 8 = 0$$

  - Exemplo 2:

$$\int e^{-x^2} dx$$

  - Exemplo 3:

$$xe^x = 10$$

Aplicações

- O que pode ser solucionado numericamente?
    1. Equações de uma variável;
    2. Sistemas lineares;
    3. Sistemas de equações não lineares;
    4. Ajuste de curvas;
    5. Interpolação;
    6. Equações diferenciais;
    7. Integrais.

Características

- O cálculo numérico pode ser utilizado mesmo quando não há solução analítica;
- O resultado é sempre numérico;
- Segundo [Scott, 2011], o processo de cálculo pode ser dividido em duas fases:
  - Desenvolvimento de algoritmos;
  - Análise de algoritmos.

Problema Numérico

- **Tipo de problema que pode ser resolvido por meio de cálculo numérico.**
- Importante salientar que nem todo problema matemático é numérico;
  - O problema matemático deve ser convertido em um problema numérico[1];

$$\sqrt{a} = ?$$

$$x = \sqrt{a}$$

$$f(x) = x^2 - a = 0$$

[1] Exemplo da Seção 1.1 de [Campos Filho, 2007]

Método Numérico

- **Métodos numéricos são ferramentas matemáticas para solução de problemas numéricos.**
- Os procedimentos necessários podem incluir:
  1. Transformação de problema matemático em problema numérico;
  2. Solução do problema numérico.

Algoritmos

- Métodos numéricos estão diretamente relacionados a algoritmos;
- Desse modo, a análise de algoritmos é indicada para avaliação dos melhores métodos;
- A escolha do algoritmo mais adequado a uma tarefa está relacionada aos seguintes aspectos:
  - Acurácia (precisão) do algoritmo;
  - Velocidade de convergência;
  - Esforço computacional, em termos de tempo computacional e uso de memória;

Algoritmos

- Importante também observar características como:'
  - Estabilidade: comportamento contínuo independentemente dos parâmetros de entrada;
  - Efeitos de arredondamento (precisão).
- Os seguintes conceitos relacionados a algoritmos devem ser lembrados:
  - Complexidade de algoritmos;
  - Adaptabilidade (alguns algoritmos podem se adaptar a dados do problema para melhorar a eficiência ou a estabilidade).

Iteração - Aproximação Sucessiva

- **Iteração corresponde à execução repetida de um conjunto de instruções [Wentworth et al., 2012].**
- Métodos iterativos são constituídos pelas seguintes etapas:
  1. Tentativa inicial: primeira tentativa para solução de um problema numérico;
  2. Equação de recorrência: equação utilizada para que o método se aproxime do objetivo (solução do problema), por meio de iterações (ou aproximações sucessivas);
  3. Condição de parada: condição pelo qual o método é finalizado (erro menor que um determinado valor, número de iterações, etc.).

Complexidade Computacional

Complexidade Computacional

"A teoria de complexidade computacional busca quantificar a quantidade de recursos computacionais necessários para solução de uma determinada tarefa" [Arora and Barak, 2009]

- A complexidade pode ser definida em termos de:
  - Tempo: número de operações realizadas;
  - Espaço: quantidade de memória utilizada.

Complexidade computacional

- Em relação ao tempo, conta-se o número de funções elementares executadas, como adição, subtração, multiplicação e divisão;
- O número de operações deve ser contato com base em uma única operação *(adição/substração ou mult/divisão)*;
  - Complexidade adição e substração: $\Theta(n)$
  - Complexidade multiplicação: $\mathcal{O}(n \log n \log \log n)$ [2]

[2] Segundo [Harvey and Van Der Hoeven, 2019], multiplicações podem ser executadas em tempo $\mathcal{O}(n \log n)$

Ordem de Grandeza

- Definição: Sejam funções $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ e $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$. A função f possui a mesma ordem de grandeza de g, definida por $f = \Theta(g)$, se existirem constantes positivas x_0, c_1 e c_2 tal que $x \geq x_0$ e $c_1 \cdot g(x) \leq f(x) \leq c_2 \cdot g(x)$ [Gersting, 2014]
    - Essa definição indica que a função $f(x)$ é dominada assintoticamente pelas funções $c_1 \cdot g(x)$ e $c_2 \cdot g(x)$.

Ordem de Grandeza

INSTITUTO FEDERAL Minas Gerais

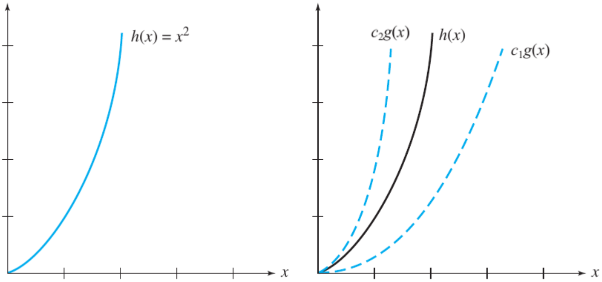
- Para a equação abaixo, com constantes positivas c_1 e c_2, podemos estabelecer as seguintes conclusões:

$$\boxed{c_1 \cdot g(x) \leq f(x) \leq c_2 \cdot g(x)}$$

  - A função $f(x)$ irá se manter sempre dentro de um "envelope" das demais;
  - A expressão $c_1 \cdot g(x)$ será um limite inferior, enquanto a expressão $c_2 \cdot g(x)$ será um limite superior de $f(x)$;
  - A alteração do valor das constantes altera a largura do envelope, porém não altera sua forma;
  - Se f está contida, a partir de n_0, em um envelope definido por g, então f e g tem a mesma ordem de grandeza [**?**].

Ordem de Grandeza

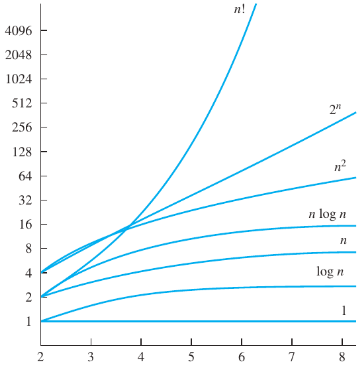
- A figura abaixo exibe o comportamento de uma função $h(x)$, dominada assintoticamente por $c_1 \cdot g(x)$ e $c_2 \cdot g(x)$.
    - Utilizam-se como limite, em geral, funções já conhecidas, para que seja possível inferir um comportamento da função avaliada.



Fonte: Adaptado de [Gersting, 2014]

Ordem de Grandeza

- Algumas funções genéricas com comportamento conhecido na literatura podem ser vistas na figura abaixo.



Fonte: [Rosen, 2019]

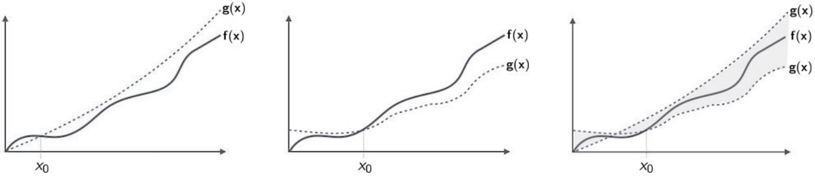
As funções são utilizadas como estimativa de comportamento. Gráfico em escala logarítmica.

Notações Big-O, Big-Ω e Big-Θ

- A notação Big-O é utilizada como uma estimativa teórica do limite superior de execução de um algoritmo
  - Está associada à execução do algoritmo no pior caso, ou seja, ao tempo máximo (ou tamanho máximo em memória) para finalização da execução do algoritmo;
  - A notação é criada com base no crescimento de funções;
- Além da notação Big-O, mais utilizada, também existem as notações Big-Ω e Big-Θ
  - Big-Ω: expressa o limite inferior do algoritmo - associada ao melhor caso em complexidade de tempo ou espaço;
  - Big-Θ: expressa limites inferiores e superiores do algoritmo.

Notações Big-O, Big-Ω e Big-Θ

- O comportamento das notações Big-O, Big-Ω e Big-Θ podem ser vistas na figura abaixo.



(a) Big-O (b) Big-Ω (c) Big-Θ

Fonte: Adaptado de [Tutorials Point, 2021]

Etapas na Solução de Problemas Numéricos

Etapas

- Segundo [Campos Filho, 2007], a solução de problemas numéricos envolve as seguintes fases:

1. Definição do problema;
2. Modelagem matemática;
3. Solução numérica
   a. Elaboração do algoritmo;
   b. Codificação do programa;
   c. Processamento do programa;
4. Análise dos resultados.

Etapas

1. Definição do problema: Corresponde a definição do problema real que será resolvido[3]
   - Ex.: Calcular, usando somente $+, -, \times, /$, a expressão abaixo:

$$\sqrt{a} \quad \text{, onde } a > 0$$

2. Modelagem matemática: Transformação do problema real em problema numérico, por meio de expressões matemáticas

$$x = \sqrt{a} \quad \rightarrow \quad x^2 = a \quad \rightarrow \quad x^2 - a = 0$$

$$f(x) = x^2 - a = 0$$

[3] Exemplos da Seção 1.1 de [Campos Filho, 2007]

Etapas

3. Solução numérica: Etapa na qual é feita a escolha do método numérico e construção do algoritmo para solução do problema. A construção do algoritmo pode ser dividida em 3 fases;
   - Elaboração do algoritmo;
   - Codificação do programa;
   - Processamento do programa;
4. Análise dos resultados: Etapa na qual é avaliado se o resultado da solução numérica corresponde à solução do problema real. Caso não seja satisfatória, o problema deve ser novamente modelado.

Referências I

Arora, S. and Barak, B. (2009).
Computational Complexity: A Modern Approach.
Cambridge University Press.

Campos Filho, F. F. (2007).
ALGORITMOS NUMERICOS.
LTC, 2 edition.

da Silva, D. M. (2020).
Cálculo Numérico - Slides de Aula.
IFMG - Instituto Federal de Minas Gerais, Campus Formiga.

Gersting, J. L. (2014).
Mathematical Structures for Computer Science.
W. H. Freeman and Company, 7 edition.

Harvey, D. and Van Der Hoeven, J. (2019).
Integer multiplication in time O(n log n).
working paper or preprint.

Justo, D., Sauter, E., Azevedo, F., Guidi, L., and Konzen, P. H. (2020).
Cálculo Numérico, Um Livro Colaborativo - Versão Python.
UFRGS - Universidade Federal do Rio Grande do Sul.
https://www.ufrgs.br/reamat/CalculoNumerico/livro-py/livro-py.pdf.

Referências II

INSTITUTO FEDERAL
Minas Gerais

Prof. Evgeni Burovski (2020).
Introduction to numerical analysis.
National Research University Higher School of Economics / Cousera.
https://www.coursera.org/learn/intro-to-numerical-analysis/home/welcome [Online]; acessado em 13 de Julho de 2020.

Rosen, K. H. (2019).
Discrete Mathematics and Its Applications.
McGraw-Hill, 8 edition.

Scott, L. (2011).
Numerical Analysis.
Princeton University Press.

Tutorials Point (2021).
Data structures - asymptotic analysis.
[Online]; acessado em 17 de Março de 2021. Disponível em:
https://www.tutorialspoint.com/data_structures_algorithms/asymptotic_analysis.htm.

Wentworth, P., Elkner, J., Downey, A. B., and Meyers, C. (2012).
How to Think Like a Computer Scientist: Learning with Python 3.
2nd edition.
[Online]; acessado em 13 de Julho de 2020.