Universidade Federal de Alagoas – UFAL
Centro de Tecnologia – CTEC

Disciplina: Cálculo Numérico (EAMB018/A-EPET019/A)
Professores: Adeildo S. Ramos Jr. e Luciana C. L. M. Vieira
Lista 05: Integração Numérica e Diferenciação Numérica
Data de divulgação: 03-04/05/2021. Data de entrega: 10-11/05/2021.

QUESTÃO 1

a) Usando sua linguagem de programação favorita, implementar duas funções para estimar o valor de integrais numericamente usando: (i) a regra dos trapézios composta; e (ii) a regra de 1/3 de Simpson composta, respectivamente (ver Figura 1).
- As funções devem receber os seguintes argumentos de entrada:
  - f: função que recebe o valor x e retorna o valor $y = f(x)$;
  - a, b: intervalo de integração;
  - n: número de intervalos considerado na regra composta.
- As funções devem retornar o seguinte argumento de saída:
  - i: valor aproximado de $\int_a^b f(x)\,\mathrm{d}x$.

b) Aplicar as funções implementadas a pelo menos duas funções $f(x)$ de sua escolha, com integral analítica finita e conhecida, adotando intervalos de integração para cada uma (ver Figura 2).

c) Comparar os valores numéricos obtidos com as regras supracitadas aos valores analíticos das respectivas integrais nos intervalos considerados. Justificar as diferenças obtidas em relação ao valor analítico e entre cada regra utilizada.

QUESTÃO 2

a) Usando sua linguagem de programação favorita, implementar três funções para estimar o valor da derivada em um dado valor de x usando o método das diferenças finitas: (i) anteriores; (ii) posteriores; e (iii) centradas, respectivamente (ver Figura 3).
- As funções devem receber os seguintes argumentos de entrada:
  - f: função que recebe o valor x e retorna o valor $y = f(x)$;
  - a: valor de x em que a derivada da função f será estimada;
  - h: valor do espaçamento entre dois pontos no método das diferenças finitas.
- As funções devem receber os seguintes argumentos de entrada:
  - d: valor aproximado de $\left.{}^{\partial f}\!/_{\partial x}\right|_{x=a}$.

b) Aplicar as funções implementadas a pelo menos duas funções $f(x)$ de sua escolha, que possuam derivada analítica finita e conhecida em um valor adotado de x para cada uma (ver Figura 4).

c) Comparar os valores numéricos obtidos com os métodos supracitados aos valores analíticos das respectivas derivadas nos pontos considerados. Justificar as diferenças obtidas em relação ao valor analítico e entre cada método utilizado.

```julia
# Regra do trapézio composta
function cn_trap(f, a, b, n)
    # <Desenvolva seu código aqui>
    return i
end;

# Regra de 1/3 de Simpson composta
function cn_simpson(f, a, b, n)
    # <Desenvolva seu código aqui>
    return i
end;
```

Figura 1: Exemplo de código em linguagem Julia: Implementação de integração numérica.

```julia
# Avaliação da integral de uma função f(x)
# em 0 ≤ x ≤ 0.8 e usando 10 intervalos
f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5
# [Em sua verificação, considere outras funções!]

i_trap = cn_trap(f, 0, 0.8, 10)
i_simpson = cn_simpson(f, 0, 0.8, 10)
```

Figura 2: Exemplo de código em linguagem Julia: Avaliação de integração numérica.

```julia
# Método das diferenças finitas anteriores
function cn_backward_diff(f, a, h)
    # <Desenvolva seu código aqui>
    return d
end;

# Método das diferenças finitas posteriores
function cn_forward_diff(f, a, h)
    # <Desenvolva seu código aqui>
    return d
end;

# Método das diferenças finitas centradas
function cn_centered_diff(f, a, h)
    # <Desenvolva seu código aqui>
    return d
end;
```

Figura 3: Exemplo de código em linguagem Julia: Implementação de diferenciação numérica.

```julia
# Avaliação da derivada de uma função f(x)
# em x = 0.5 e usando h=0.25
f(x) = -0.1x^4 -0.15x^3 - 0.5x^2 -0.25x + 1.2
# [Em sua verificação, considere outras funções!]

df_b = backward_diff(f, 0.5, 0.25)
df_c = centered_diff(f, 0.5, 0.25)
df_f =  forward_diff(f, 0.5, 0.25)
```

Figura 4: Exemplo de código em linguagem Julia: Avaliação de diferenciação numérica.