Métodos Numéricos — Lista 01

André Paladini 14182390
Tiago F. Oliva Costa 8004408

10 de outubro de 2023

1 Questão 01

Classifique as EDPs abaixo quanto à ordem, a linearidade / não-linearidade, a homogeneidade e ao tipo.

A 2a ordem; Linear; Homogênea.

B 2a ordem; Linear; Não-Homogênea.

C 1a ordem; Linear; Homogênea.

D 2a ordem; Linear; Homogênea.

E 2a ordem; Não-Linear; Homogênea.

F 2a ordem; Não-Linear; Não-Homogênea.

2 Questão 02

Qual a diferença entre as condições de contorno de Dirichlet, Neumann e Robin?

A condição de contorno de Dirichlet (ou primeiro tipo) especifica valores que a variável dependente $y(x)$ toma ao longo da fronteira do domínio. Ou seja

$$y(a) = \alpha, \quad y(b) = \beta.$$

A condição de contorno de Neumann (ou segundo tipo) especifica valores que a derivada $y'(x)$ da variável dependente toma ao longo da fronteira do domínio. Ou seja

$$y'(a) = \alpha, \quad y'(b) = \beta.$$

A condição de contorno de Robin (ou terceiro tipo) especifica valores que tanto a variável dependente $y(x)$, como a sua derivada $y'(x)$, tomam ao longo da fronteira do domínio. Ou seja, para um domínio Ω e sua fronteira representada por $\partial\Omega$, têm-se

$$ay + b\frac{\partial y}{\partial x} = g \qquad \text{em} \qquad \partial\Omega.$$

3 Suplemento

Para as questões 03 e 04, considere como *forward difference*

$$D_+f(x) = f(x+h) - f(x),$$

central difference

$$D_0f(x) = f(x+\frac{h}{2}) - f(x-\frac{x}{2}),$$

e *backward difference*

$$D_-f(x) = f(x) - f(x-h).$$

A metodologia exposta na Seção 1.5 de LeVeque[1], é adotada para calcular os coeficientes gerais para diferenças finitas

$$\frac{1}{(i-1)!}\sum_{j=1}^{n} c_j(x_j-\bar{x})^{(i-1)} \begin{cases} 1 & \text{se } i-1=k \\ 0 & \text{caso contrário} \end{cases}$$

implementada segundo o seguinte excerto de código em Python

```
"""
Find finite difference coefficients

k    - kth derivative order
xbar - target point to approximate around
x    - vector of N stencil points
"""
def fdcoeffV(k, xbar, x):
    if isinstance(x, list):
        x = np.array(x)

    n = len(x)
    A = np.ones((n, n))
    xrow = np.transpose(x - xbar)  # displacements as a row vector.

    for i in range(1, n+1):
        A[i-1, :] = np.divide(np.power(xrow, (i - 1)), math.factorial(i - 1))

    b = np.zeros((n, 1))  # b is right hand side,
    b[k] = 1  # so kth derivative term remains
    c = np.linalg.solve(A, b)  # solve system for coefficients
    return np.transpose(c)  # row vector
```

4 Questão 03

Pede-se $\frac{\mathrm{d}J_0(x)}{\mathrm{d}x}$ em $x = 3$, onde

$$J_\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,(m+\alpha)!} \left(\frac{x}{2}\right)^{2m+\alpha},$$

e por sua vez

$$J_0(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,m!} \left(\frac{x}{2}\right)^{2m}.$$

Considerando que a função de Bessel converge, podemos aplicar a derivada da série infinita obtendo

$$J_0'(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,(m-1)!} \left(\frac{x}{2}\right)^{2m-1} = J_{(-1)}(x) = -J_1(x).$$

Dessa forma temos a solução exata

$$f'(x) = -0.339059$$

e as aproximações

(a) Backward com 2 pontos

$$D_{-2} = -0.3836, \quad error = 4.46\text{e}{-2}$$

(b) Backward com 3 pontos

$$D_{-3} = -0.3439, \quad error = 4.89\text{e}{-2}$$

(c) Forward com 2 pontos

$$D_{+2} = -0.2908, \quad error = -4.83\text{e}{-2}$$

(d) Forward com 3 pontos

$$D_{+3} = -0.3414, \quad error = 2.38\text{e}{-3}$$

(e) Central com 2 pontos

$$D_{0,2} = -0.3372, \quad error = -1.84\text{e}{-3}$$

(f) Central com 4 pontos

$$D_{0,4} = -0.3390, \quad error = -1.53\text{e}{-5}$$

5 Questão 04

Considere a função

$$f(x) = e^x \sin(x).$$

Temos, pela regra do produto,

$$f'(x) = e^x \sin(x) + e^x \cos(x),$$

e aplicando a regra do produto novamente

$$f''(x) = 2e^x \cos(x).$$

(i) Central com 2 pontos (ii) Central com 4 pontos

Referências

[1] Randall J. LeVeque (2007) *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, Society for Industrial and Applied Mathematics