

Lista de Exercícios 3 de Análise Numérica

Prof.: Fabrício Murai

Informações importantes:

- Data de entrega: até 23:59 do dia 26/04/2018.
- Questões podem ser discutidas entre até três alunos. Nomes dos colegas precisam ser listados. Contudo, a escrita das soluções e submissão deve ser feita individualmente.
- Submissão deve ser feita em formato PDF através do Moodle, mesmo que tenham sido resolvidas a mão e escaneadas.
- Todas as soluções devem ser justificadas.
- Se puder, peço por favor que marque o tempo gasto para resolver a lista, para que o tamanho da lista de exercícios seja ajustado em semestres futuros.

1. Considere os pontos $\{(0.0, 0.0), (0.63, 0.59), (1.26, 0.95), (1.88, 0.95)\}$.
 - (a) Seja o polinômio de Lagrange $L_n(x) = c_0P_0(x) + c_1P_1(x) + \dots + c_nP_n(x)$. Determine $P_0(x)$ e $P_2(x)$.
 - (b) Com a ajuda de uma calculadora, calcule o valor da interpolação em $x = \sqrt{2}/2$ a partir de $L_n(x)$.
 - (c) Sem fazer contas, calcule $P_0(0.63)$.
 - (d) Sem fazer contas, calcule $L_n(0.0)$.
2. Implemente a função InterpolacaoLagrange abaixo usando a matriz G vista no final da aula do dia 10/Abril. A função recebe como entrada dois arrays de mesmo tamanho - x contém as abscissas, y , as ordenadas - e uma abscissa z a ser interpolada.

```
def InterpolacaoPolinomial(x, y, z):  
    m = length(x)          # numero de pontos m=n+1  
    G = np.zeros((m,m))  
    for i in range(m):  
        for j in range(m):  
            # Preencher G aqui  
  
    Gd =  
    Gi = np.zeros(m)  
    for i in range(m):  
        Gi =  
  
    somatorio = 0.0  
    for i in range(m):  
        somatorio +=  
  
    z = Gd * somatorio  
    return z
```

3. Nesta questão vamos usar sua implementação para verificar o que acontece ao escolhermos alguns dos pontos dados na Questão 1 para obter uma interpolação de grau menor.

- Usando todos os pontos dados, obtenha interpolações para $z \in \{0.01, 0.02, \dots, 3.14\}$. Dica: `z = np.arange(0.01, 3.15, 0.01)`.
- Plote um gráfico com duas curvas: a primeira é formada pelas interpolações obtidas e a segunda é obtida pela função $f(z) = \sin(z)$. Você pode usar o código no notebook da aula para gerar o gráfico. Dica: `f = np.sin(z)`.
- O que acontece se você trocar a ordem em que os pontos (x_i, y_i) aparecem na entrada? Por exemplo: `x = np.array([0.63, 0.0, 1.26, 1.88]); y = np.array([0.59, 0.0, 0.95, 0.95])`?
- O que acontece se você usar apenas três pontos para calcular a interpolação? Por exemplo, qual a diferença para a aproximação obtida para $z = 1.2$ quando você usa `x = np.array([0.0, 0.63, 1.26, 1.88])` e `x = np.array([0.0, 0.63, 1.88])`?