

Universidade Federal de São João
GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

PRÁTICAS DA DISCIPLINA DE CÁLCULO NUMÉRICO

Professor: Ramon Dornelas Soares.
e-mail: soares.ramon@ufsj.edu.br

Ouro Branco
Setembro 2016

Índice

1	Introdução ao Scilab	3
1.1	Diretivas Gerais	3
1.2	Diretivas Específicas	3
1.2.1	Programa Operações com Matrizes	3
1.2.2	Programa Construção de Gráficos	4
1.2.3	Programa Estrutura Condicional	5
1.2.4	Programa Estrutura de Repetição	6
1.2.5	Programa Estrutura de RepetiçãoII	6
2	Sistemas Lineares	8
2.1	Diretivas Gerais	8
2.2	Diretivas Específicas	8
2.2.1	Resolução de Sistemas com Duas Variáveis	9
2.2.2	Resolução de Sistemas com Quatro Variáveis, usando o Método de Gauss e o Método Iterativo.	9
3	Interpolação Linear e Polinomial com pol. interp. de grau ‘n’.	11
3.1	Diretivas Gerais	11
3.2	Diretivas Específicas	12
3.2.1	Explique a função FuncaoX	12
3.2.2	Interpolação Linear, método equação da reta	12
3.2.3	Interpolação, usando o método Polinomial	13

Resumo

Esta apostila contém todas as atividades que devem ser executadas em laboratório pelos alunos, as quais correspondem a uma parcela dos exercícios avaliados durante curso da disciplina de Cálculo Numérico. Todas as atividades presentes neste material serão pontuadas. Todas as atividades deverão ser executadas e entregues no portal da ufsj durante a aula prática de 1 hora 50 minutos. O ementário do curso de cálculo Numérico versa sobre os seguintes tópicos: Sistemas Lineares, Ajuste De curvas, Interpolação, Integração Numérica e Zeros de Funções, Resolução Numérica de Equações Diferenciais Ordinárias.

Palavras-chave: Sistemas Lineares, Ajuste De curvas, Interpolação, Integração Numérica, Zeros, Equações Diferenciais Ordinárias.

Prática 1 Introdução ao Scilab

Esta atividade visa a construção de funções diversas no Scilab, objetivando fornecer aos alunos conhecimentos da linguagem de programação e, também, da criação de uma funções. Especificamente serão criadas funções envolvendo matrizes, estruturas de repetição e condicionais.

1.1 Diretivas Gerais

As seguintes diretivas gerais deverão ser seguidas:

- Deverá ser construída individualmente ou em dupla. Os nomes dos integrantes do grupo deverão estar comentados (use “%”) no início de cada função, a qual receberá o nome de NomeDaFuncaoAB, sendo A e B as iniciais da dupla;
- Cada dupla deverá criar sua própria atividade, sendo vetada a comunicação entre as duplas.
- A comunicação interna entre a dupla deverá ser realizada de modo que somente os integrantes do dupla participem.
- A dupla que não respeitar as condições estabelecidas terá sua atividade anulada.

1.2 Diretivas Específicas

Os seguintes programas deverão ser construídos pelos alunos:

1.2.1 Programa Operações com Matrizes

Crie uma função no Scilab com o nome OperacaoMatrizesAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Escreva na primeira linha do programa de modo comentado (use “%”) “Comentários Iniciais”
2. Escreva o nome da Função (OperacaoMatrizesAB) de modo comentado em uma linha e os nomes dos alunos nas linhas subsequentes.
3. Use uma linha “%- - -” para separar a parte de Comentários Iniciais das demais partes do programa.
4. Defina uma segunda parte do programa com “Parâmetros de Entrada” (use “%”).
5. Crie duas matrizes (A e B) de dimensões 2×2 e diferentes entre si (as quais poderão ser alteradas, propiciando, posteriormente obtermos respostas de operações numéricas para matrizes diferentes).
6. Separe a parte de “Parâmetros de Entrada” das demais partes do programa usando uma linha do tipo “%- - -”.
7. Defina uma segunda parte do programa com “Corpo do Programa” (use “%”).
8. Crie uma variável “soma” use-a para armazenar a soma das duas matrizes (use o operador ‘+’). Para exibir no console o valor desta variável use, na linha posterior ao cálculo a função “disp”, ou seja, “disp(soma)”.
9. Crie uma variável “subtracao” use-a para armazenar a subtração das duas matrizes (use o operador ‘-’). Exiba no console o resultado.
10. Crie uma variável “multiplicacao” use-a para armazenar a multiplicação individual dos elementos das duas matrizes (use o operador ‘.*’). Exiba no console o resultado.
11. Crie uma variável “Divisao” use-a para armazenar a divisão dos elementos das duas matrizes (use o operador ‘./’). Exiba no console o resultado.

1.2.2 Programa Construção de Gráficos

Crie uma função no Scilab com o nome ConstrucãoGraficosAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item [1.2.1](#)
2. Na parte parâmetros de entrada (veja como fazer item [1.2.1](#)) defina um vetor x que varie de 0 até 10 passo 0.3, faça isso usando a seguinte diretiva: “x=[0:0.3:10]”.

3. Na parte corpo do programa (veja como fazer item 1.2.1) crie um vetor y1 que possua os valores de seno correspondentes aos valores de x criados, use “y1=sin(x)”.
4. Apresente um gráfico que possua os valores de x versus y. Use a função plot, “plot(x,y1,'b-’)”, na qual “b” corresponde a cor azul e “-” corresponde a linha contínua.
5. Construa, também usando o vetor x, um gráfico cosseno, crie a função y2 e use a função “cos(x)”. Construa o gráfico usando a cor rosa, substituindo “b” por “m”.
6. Construa, também usando o vetor x, um gráfico exponencial, crie a função y3 e use a função “exp(x)”. Construa o gráfico usando a cor verde, substituindo “b” por “g”.
7. Construa, também usando o vetor x, um gráfico para a função $y4 = 2x + 10$. Construa o gráfico usando a cor preta, substituindo “b” por “k”.
8. Construa, também usando o vetor x, um gráfico para a função $y5 = x^2 + 3$. Construa o gráfico

1.2.3 Programa Estrutura Condicional

Crie uma função no Scilab com o nome Estrutura CondicionalAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item 1.2.1
2. Na parte parâmetros de entrada (veja como fazer item 1.2.1) defina um variável x (dimensão 1×1) que possua o valor inicial zero, o qual posteriormente será alterado.
3. Na parte corpo do programa (veja como fazer item 1.2.1), deve-se implementar uma estrutura que retorne um valor y seguindo a função abaixo:

$$y = \begin{cases} 2x + 2, & \text{se } x < -2 \\ 3, & \text{se } -2 \leq x \leq 3 \\ -x, & \text{se } x > 3. \end{cases} \quad (1.1)$$

dica: use estruturas com if, elseif ou else. Para isto procure no help do scilab como usar estas estruturas e também operadores de comparação (digite operadores).

1.2.4 Programa Estrutura de Repetição

Crie uma função no Scilab com o nome EstuturaRepeticaoAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item 1.2.1
2. Na parte parâmetros de entrada (veja como fazer item 1.2.1), não defina nenhum parâmetro, pois este programa não possuirá parâmetros de entrada.
3. Na parte corpo do programa (veja como fazer item 1.2.1), deve-se implementar uma estrutura que retorne os seguintes valores abaixo:

$$y1 = \sum_{x=1}^{10} \frac{x}{3} \quad (1.2)$$

$$y2 = \prod_{m=1}^{10} (x + 0.5) \quad (1.3)$$

1.2.5 Programa Estrutura de RepetiçãoII

Crie uma função no Scilab com o nome mediaAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item 1.2.1
2. Na parte parâmetros de entrada (veja como fazer item 1.2.1), crie um vetor x de dimensão não definida ($1 \times n$), ou seja um conjunto de valores.
3. Na parte corpo do programa (veja como fazer item 1.2.1), deve-se implementar uma estrutura que retorne a média deste vetor de valores.

Dica: Como o vetor não possui dimensão definida (cada vez que a função é executada pode-se ter um vetor com tamanho diferente), deve-se usar a função **size** para determinar o tamanho deste vetor primeiro. A diretiva deve ser “[nl,nc]=size(x)”, a qual retornará o número de linhas e o número de colunas de x, respectivamente, nl e nc.

dica: aguarde o professor que ministrará, no quadro, as instruções.

Prática 2 Sistemas Lineares

Esta atividade visa a criação de três funções no Scilab, duas para solucionar sistemas lineares usando Gauss e a última usando o Método Iterativo. A primeira deve ser capaz de solucionar um sistema com duas variáveis (a matriz A possui uma dimensão 2×2) e os outros dois com 4 variáveis (cuja matriz A possui uma dimensão 4×4).

2.1 Diretivas Gerais

As seguintes diretivas gerais deverão ser seguidas:

- Deverá ser realizada durante o tempo de 1 hora e cinquenta minutos;
- Deverá ser construída individualmente ou em dupla. Os nomes dos integrantes do grupo deverão estar comentados (use “%”) no início de cada função, a qual receberá o nome de NomeDaFuncaoAB, sendo A e B as iniciais da dupla;
- Cada dupla deverá criar sua própria atividade, sendo vetada a comunicação entre as duplas.
- A comunicação interna entre a dupla deverá ser realizada de modo que somente os integrantes do dupla participem;
- A dupla que não respeitar as condições estabelecidas terá sua atividade anulada.

2.2 Diretivas Específicas

As seguintes funções deverão ser construídas pelos alunos:

2.2.1 Resolução de Sistemas com Duas Variáveis

Crie uma função no Scilab com o nome Resolucao2VariaveisAB (sendo A e B as iniciais da dupla), as quais deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item 1.2.1
2. Na parte parâmetros de entrada (veja como fazer item 1.2.1) defina a matriz A e o vetor b de um sistema linear com duas variáveis (x_1 e x_2).
3. Na parte corpo do programa (veja como fazer item 1.2.1), deve-se implementar uma estrutura que retorne a solução do sistema seguindo o método de **Gauss** com Pivotação, ou seja, os valores de x_1 e x_2 .
4. A função criada deverá apresentar como parâmetro de saída um vetor coluna X, que contenha os valores de x_1 e x_2 .
5. Use o sistema abaixo conferir a função:

$$\begin{cases} 2x_1 - x_2 &= 1 \\ x_1 + 2x_2 &= 3 \end{cases} \quad (2.1)$$

Solução:

$$X = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (2.2)$$

2.2.2 Resolução de Sistemas com Quatro Variáveis, usando o Método de Gauss e o Método Iterativo.

Crie duas funções no Scilab, a primeira o nome Resolucao4VariaveisGuassAB (sendo A e B as iniciais da dupla) e a segunda com o nome Resolucao4VariaveisIterativoAB (sendo A e B as iniciais da dupla), a qual deve contemplar os itens abaixo:

1. Crie os Comentários Iniciais de modo semelhante ao item 1.2.1
2. Na parte parâmetros de entrada (veja como fazer item 1.2.1) defina a matriz A e o vetor b de um sistema linear com quatro variáveis (x_1, x_2, x_3 e x_4).

3. Na parte corpo do programa (veja como fazer item 1.2.1), deve-se implementar uma estrutura que retorne a solução do sistema seguindo o método de **Gauss**, ou seja os valores de x_1 , x_2 , x_3 e x_4 .
4. A função criada deverá apresentar como parâmetro de saída um vetor coluna X , que contenha os valores de x_1 , x_2 , x_3 e x_4 .
5. Para a função do programa iterativo adote como critério de parada um erro absoluto máximo para interção k (E^k) menor que 10^{-3} , sendo este erro calculado por:

$$E^k = |x_i^{k+1} - x_i^k|_m \quad ax \quad (2.3)$$

6. Use o sistema abaixo conferir a função:

$$\begin{cases} -1x_1 + 2x_2 + 3x_3 + 1x_4 &= 1 \\ +2x_1 - 4x_2 - 5x_3 - 1x_4 &= 0 \\ -3x_1 + 8x_2 + 8x_3 + 1x_4 &= 2 \\ +1x_1 + 2x_2 - 6x_3 + 4x_4 &= -1 \end{cases} \quad (2.4)$$

Solução:

$$X = \begin{bmatrix} 5.6 \\ 0.7 \\ 1.6 \\ 0.4 \end{bmatrix} \quad (2.5)$$

Dica1: Use o programa construído em sala de aula pelo professor como referência.

Prática 3 Interpolação Linear e Polinomial com pol. interp. de grau ‘n’.

Esta atividade visa o cálculo de interpolações para os exemplos feitos em sala (Vide Notas de Aula). Primeiramente, os alunos deverão fazer um resumo descrevendo o funcionamento de uma função postada no Moodle. Em seguida, usando parte da função acima, deverão criar uma função para fazer uma interpolação Linear, **encontrando a reta que passa pelos dois pontos, Polinômio Interpolador $p_1(x)$ e valor para a interpolação no ponto X .**

3.1 Diretivas Gerais

As seguintes diretivas gerais deverão ser seguidas:

- Deverá ser realizada durante o tempo de 1 hora e quarenta minutos;
- Deverá ser construída individualmente ou em dupla. Os nomes dos integrantes do grupo deverão estar comentados (use `\\`) no início de cada função, a qual receberá o nome de NomeDaFuncaoAB, sendo A e B as iniciais da dupla;
- Cada dupla deverá criar sua própria atividade, sendo vetada a comunicação entre as duplas, cópias de atividades serão punidas com nota zero. **Sugiro que cada dupla ocupe computadores separados.**
- A comunicação interna entre a dupla deverá ser realizada de modo que somente os integrantes do dupla participem;
- A dupla que não respeitar as condições estabelecidas terá sua atividade anulada.

- O alunos presentes devem colocar no Moodle as funções criadas.

3.2 Diretivas Específicas

3.2.1 Explique a função FuncaoX

Cada Dupla deverá explicar o funcionamento da FuncaoX (Postada no moodle). A explicação deverá ser feita em documento word, com o nome **explicacaoAB.doc** (sendo A e B as iniciais da dupla).

- Crie os Comentários Iniciais “no Word” de modo semelhante ao item [1.2.1](#).
- Descreva quais são os parâmetros de entrada e quais são os dois parâmetros de saída desta função. Explique qual a funcionalidade da FuncaoX (apenas uma linha).
- Explique o que faz a função “gsort”, resalte a sua explicação descrevendo seus três parâmetros de entrada e seus dois parâmetros de saída.

A seguinte função deverá ser construída pelos alunos:

3.2.2 Interpolação Linear, método equação da reta

As seguintes diretivas específicas deverão ser seguidas:

- Crie os Comentários Iniciais de modo semelhante ao item [1.2.1](#).
- Deve-se criar uma função que execute o método da **determinação da reta, calculando a inclinação ‘a’ e o ponto onde a reta intercepta o eixo y, incógnita ‘b’** (nome da função deverá ser **LinearAB**, sendo A e B as iniciais da dupla), obtendo o valor para uma interpolação Linear feita em sala, vide notas de aula. Esta função deverá apresentar os seguintes parâmetros de saída: o valor das constantes (a e b) do polinômio interpolador $p_1(x)$ e valor da interpolação para o ponto X , ou seja, o valor de $f(X) \approx p_1(X)$.
- A função criada deverá possuir três parâmetros de entrada (veja como fazer item [1.2.1](#)), são eles:
 1. Vetor x dois valores da tabela de cada exercício (Nota de aula).
 2. Vetor f dois valores da tabela de cada exercício (Nota de aula).
 3. Vetor X ponto que se deseja conhecer o valor interpolado, veja exercício (Nota de aula).

- A função deverá selecionar os pontos necessários (2 pontos) de modo automático, esta funcionalidade somente poderá ser realizada se o tamanho do vetor x ou y for maior que $n + 1$.

3.2.3 Interpolação, usando o método Polinomial

As seguintes diretivas específicas deverão ser seguidas:

- Crie os Comentários Iniciais de modo semelhante ao item 1.2.1.
- Deve-se criar uma função que execute o método da **Interpolação Plinomial** (nome da função deverá ser **PolinomialAB**, sendo A e B as iniciais da dupla), obtendo o valor para a interpolação Polinomial feita em sala, vide notas de aula. Esta função deverá apresentar apenas o valor da interpolação para o ponto X , ou seja, o valor de $f(X) \approx p_n(X)$ como parâmetro de saída.
- A função deve ser capaz de construir interpolações com polinômio interpoladores de grau 'n' (onde o valor 'n' é um parâmetro de entrada).
- A função criada deverá possuir quatro parâmetros de entrada (veja como fazer item 1.2.1), são eles:
 1. O vetor x dos valores da tabela de cada exercício(Nota de aula).
 2. O vetor f dos valores da tabela de cada exercício (Nota de aula).
 3. O escalar X ponto que se deseja conhecer o valor interpolado, veja exercício (Nota de aula).
 4. O escalar n , grau do polinômio interpolador ($n=2$, significa segundo grau).
- A função deverá selecionar os pontos necessários ($n+1$ pontos) de modo automático, esta funcionalidade somente poderá ser realizada se o tamanho dos vetores x e y forem maior que $n + 1$ (caso contrário uma mensagem de erro deve ser exibida na Command Window usando a função `error-error('Erro!- Não existem n+1 pontos necessários para esta interpolação!')`)-)