

Octave/Matlab - Lista de Exercícios 07

Módulo 1 -2021

Coltec - UFMG

Márcio Fantini Miranda

17 de junho de 2021

1 Formas Geométricas

Essa lista contém exercícios com figuras geométricas diferentes, traçadas com for e while e suas soluções.

1. Faça um script para imprimir um triângulo de base n , n ímpar e altura $2n - 1$. Por exemplo, para $n = 7$, a figura deve ficar como abaixo:

```
      *
     ***
    *****
   *********
  ***********
 *************
*****
```

Solução

```
% 2o trimestre 2015
% repeticoes
% faz triangulo
% entrada: Numero inteiro impar
% saida: desenha figura geometrica no formato piramide (ou triangulo)

r = 1;
while(r) % poderia ser feito com do-while
    n = input("entre com o número de linhas do triangulo (inteiro IMPAR): ");
    n1=n-1;
    r = mod(n1,2);
endwhile

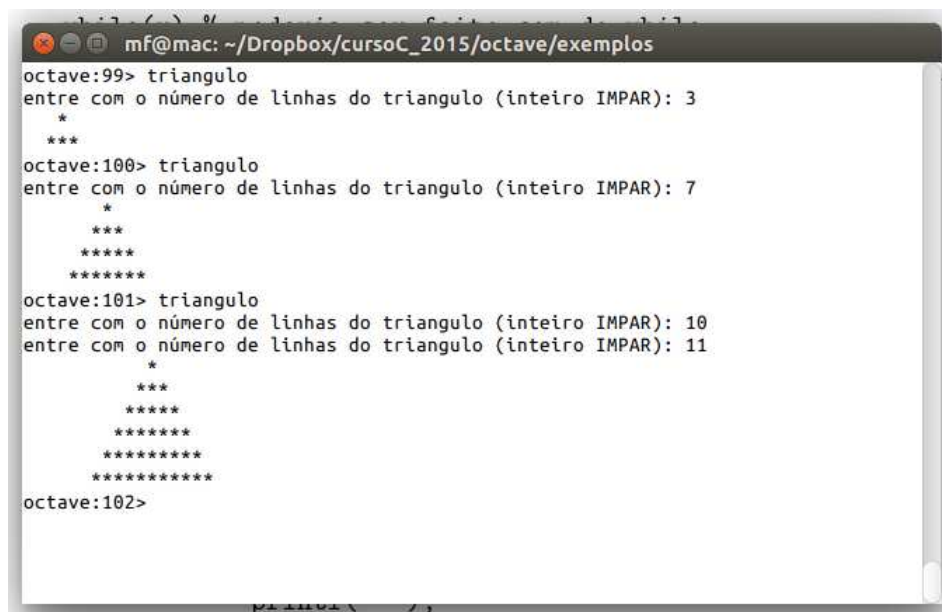
n1 = (n-1)/2;
espacos = n1*2;
k = 1;
for i=0:n1      % numero de linhas (= altura)

    % gera as colunas (largura)
    for m=0:espacos
        printf(" ");
    endfor
    for j=1:k
        printf("*");
    endfor

    espacos=espacos-1;
    k=k+2;
    printf("\n");
endfor
```

A figura 1 apresenta um exemplo de saída esperada, para $n = 4, 5$ e 9 . É possível alterar o programa

para que os espaços laterais sejam reduzidos, trazendo a figura para o canto esquerdo da tela. Veja o próximo exercício. Compare a solução de ambos



```
mf@mac: ~/Dropbox/cursoC_2015/octave/exemplos
octave:99> triangulo
entre com o número de linhas do triangulo (inteiro IMPAR): 3
  *
 ***

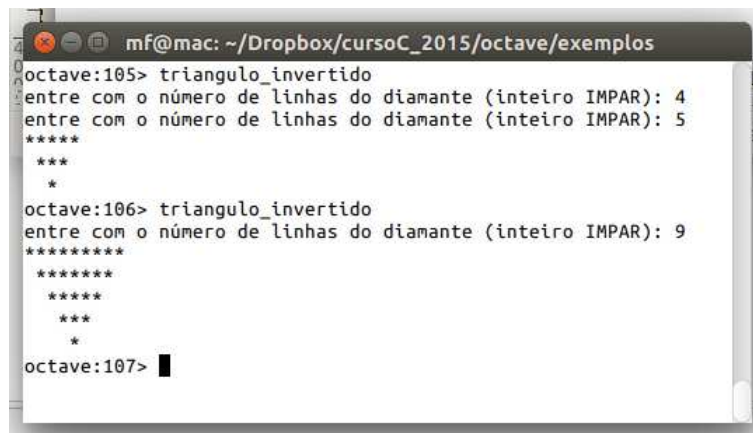
octave:100> triangulo
entre com o número de linhas do triangulo (inteiro IMPAR): 7
  *
 ***
*****
*****

octave:101> triangulo
entre com o número de linhas do triangulo (inteiro IMPAR): 10
entre com o número de linhas do triangulo (inteiro IMPAR): 11
  *
 ***
*****
*****
*****
*****

octave:102>
```

Figura 1: Tela do Octave para o programa triangulo

2. Repita o exercício anterior, agora fazendo um programa para gerar o "triângulo invertido", ou seja, com a base (de valor n) no início, e a última linha sendo um asterisco. A figura 2 apresenta um exemplo de saída esperada, para $n = 3, 7, 10$ e 11 . Nesse programa a figura deve estar colocada o máximo possível à esquerda.



```
mf@mac: ~/Dropbox/cursoC_2015/octave/exemplos
octave:105> triangulo_invertido
entre com o número de linhas do diamante (inteiro IMPAR): 4
entre com o número de linhas do diamante (inteiro IMPAR): 5
*****
***
*
octave:106> triangulo_invertido
entre com o número de linhas do diamante (inteiro IMPAR): 9
*****
*****
***
*
octave:107> 
```

Figura 2: Tela do Octave para o programa triangulo invertido

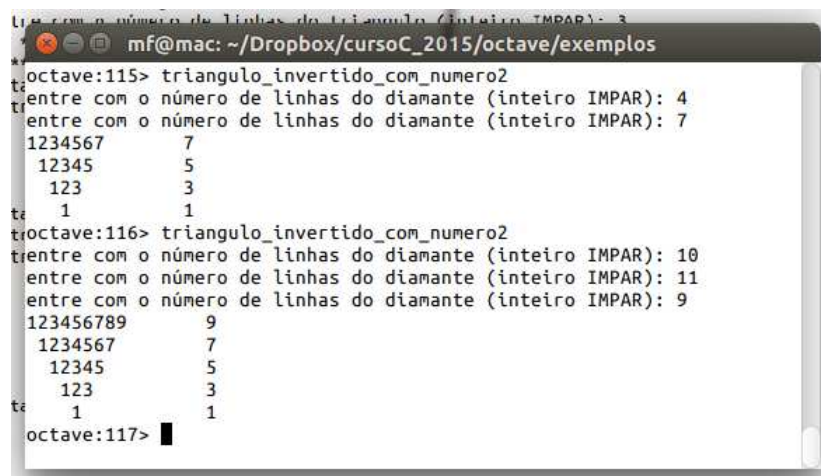
Veja o trecho principal da solução logo abaixo. A variável de entrada é o n .

```
n1 = (n-1)/2;
espacos = 0;
k = n;
for i=0:n1
    for m=1:espacos
        printf(" ");
    endfor
    for j=1:k
        printf("*");
    endfor
    espacos=espacos+1;
    k=k-2;
    printf("\n");
endfor
```

3. Incremente o exercício anterior para que seja mostrado, ao lado de cada linha (mantendo-se sempre a mesma distância) os números de asteriscos da referida linha. A figura abaixo mostra o resultado esperado para um triângulo com base $n = 9$. A solução está no final dessa lista. Antes de ver a solução, tente fazer o programa

```
*****      9
*****      7
*****      5
***         3
*           1
```

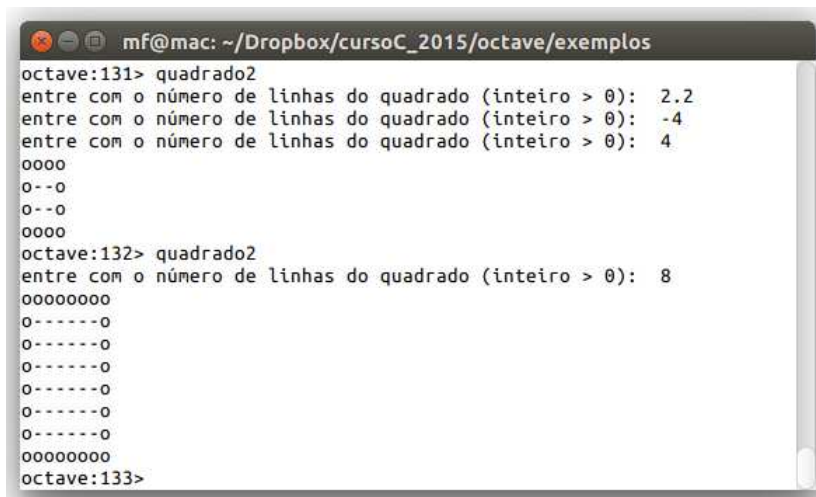
4. Altere o programa anterior, para que seja mostrado os números e não mais os asteriscos. Para que a figura mantenha sua simetria, o valor máximo para n deve ser $n = 9$. Veja a figura 3 para o programa sendo executado com $n = 4, 7, 10, 11$ e 9 .



```
mf@mac: ~/Dropbox/cursoC_2015/octave/exemplos
octave:115> triangulo_invertido_com_numero2
entre com o número de linhas do diamante (inteiro IMPAR): 4
entre com o número de linhas do diamante (inteiro IMPAR): 7
1234567      7
12345        5
123          3
1            1
octave:116> triangulo_invertido_com_numero2
entre com o número de linhas do diamante (inteiro IMPAR): 10
entre com o número de linhas do diamante (inteiro IMPAR): 11
entre com o número de linhas do diamante (inteiro IMPAR): 9
123456789    9
1234567      7
12345        5
123          3
1            1
octave:117>
```

Figura 3: Tela do Octave para o programa triangulo invertido

5. Faça um script para gerar um quadrado de lado n , sendo n inteiro positivo. O quadrado dever ter seus lados desenhados com o caracter "o" e seu interior com o caracter "-". Veja figura 4 para exemplos do programa sendo executado com diferentes entradas.



```
mf@mac: ~/Dropbox/cursoC_2015/octave/exemplos
octave:131> quadrado2
entre com o número de linhas do quadrado (inteiro > 0): 2.2
entre com o número de linhas do quadrado (inteiro > 0): -4
entre com o número de linhas do quadrado (inteiro > 0): 4
oooo
o--o
o--o
o--o
oooo
octave:132> quadrado2
entre com o número de linhas do quadrado (inteiro > 0): 8
oooooooo
o-----o
o-----o
o-----o
o-----o
o-----o
o-----o
o-----o
oooooooo
octave:133>
```

Figura 4: Quadrado recheado

2 Soluções

2.1 Triângulo com Números de Asteriscos das Linhas

```
n1 = (n-1)/2;
espacos = n1*2;
espacos = 0;
k = n;
el=3;
for i=0:n1
    for m=1:espacos
        printf(" ");
    endfor
    for j=1:k
        printf("*");
        if j==k
            for p = 1:el
                printf("%c", " ");
            end
            el = el+1;
            printf("    %i",j);
        end
    endfor
    espacos=espacos+1;
    k=k-2;
    printf("\n");
endfor
```

2.2 Quadrado Recheado

(apenas trecho do programa)

```
for i=1:n
    if (i==1 || i == n)
        for j=1:n
            printf("%c","o");
        end
    else
        for j=1:n
            if (j==1 || j ==n)
                printf("%c","o");
            else
                printf("%c","-");
            end
        end
    end

    printf("\n");
end
```

3 Mais Problemas com Loops

6. Faça um script para imprimir o "diamante de números". O script deve ler um número ÍMPAR (não deve aceitar números pares) e imprimir a "árvore" com o número lido, seguindo a sequência 1, 3, etc. Veja a lógica na figura abaixo. Por exemplo para o número 7 o desenho será:

```

      7
    7 7 7
  7 7 7 7 7
7 7 7 7 7 7 7
  7 7 7 7 7
    7 7 7
      7
```


3.1 Total de Cada Linha de uma Matriz

7. Faça um script para preencher, com números aleatórios, uma matriz de inteiros, M , de dimensão $5 \times m$, sendo que m deve ser informado pelo usuário. Os números aleatórios devem estar entre 1 e 9. m é um número inteiro, tal que $m \leq 5$. Uma vez preenchida a matriz, o script deve mostrá-la, no formato matricial e mostrar, ao lado de cada linha, o valor da soma da linha e ao lado do valor da soma, gerar um histograma dos respectivos valores. O histograma desses valores deve se mostrado com asteriscos, sendo o número de asteriscos correspondente ao valor da soma da linha. Veja o exemplo abaixo.

Considere o vetor M , de tamanho $m = 4$:

$$V = \begin{bmatrix} 5 & 1 & 2 & 7 \\ 1 & 0 & 3 & 4 \\ 9 & 1 & 5 & 2 \\ 4 & 2 & 3 & 1 \\ 1 & 1 & 2 & 4 \end{bmatrix}$$

Nesse caso, o seu histograma ficaria:

5	1	2	7	15	* * * * *
1	0	3	4	8	* * * * *
9	1	5	2	17	* * * * *
4	2	3	1	10	* * * * *
1	1	2	4	8	* * * * *

4 Figuras com Asteriscos

8. Faça um script para imprimir um triângulo de altura $2n - 1$ e largura n . O triângulo deve ser formado por asteriscos e traços. Os traços deve aparecer apenas na linha que contém n caracteres (na linha maior). Por exemplo, para $n = 4$ a figura deve ficar como abaixo:

```
*  
**  
***  
----  
***  
**  
*
```

9. Faça um script para gerar o "diamante de números". O diamante deve ter a largura máxima igual a n e altura (número de linhas) também igual a n . O Diamante deve ser formado de números, sendo que no meio (maior linha) a contagem deve ir até n . O Diamante deve ser preenchido externamente pelo caractere "-". Veja a figura abaixo, gerado com os números $n = 7$ e $n = 9$. O valor de n lido deve ser $1 \leq n \leq 9$. O número n deve ser ímpar.

```
Entre com o número de linhas do diamante (inteiro IMPAR): 7
---1---
--123--
-12345-
1234567
-12345-
--123--
---1---
```

```
entre com o número de linhas do diamante (inteiro IMPAR): 9
----1----
---123---
--12345--
-1234567-
123456789
-1234567-
--12345--
---123---
----1----
```

10. Repita o exercício anterior para gerar o Diamante de asteriscos, com o exterior preenchido com o caracter "-". O Diamante deve ter n linhas e a largura máxima n , n ímpar. Veja as figuras abaixo, para $n = 9$ e $n = 11$.

```
entre com o número de linhas do diamante (inteiro IMPAR): 11
-----*-----
----***-----
---*****---
--*****--
-*****-
*****
*****
-*****-
--*****--
---*****---
----***-----
-----*-----
octave:19> diamante_asteriscos1
entre com o número de linhas do diamante (inteiro IMPAR): 9
-----*-----
----***-----
---*****---
--*****--
-*****-
*****
*****
--*****--
---*****---
----*-----
octave:20>
```

11. Faça um script para gerar o "diamante de vazio", com o caracter "o" no exterior e espaços no interior. O diamante deve ter a largura máxima igual a n e altura (número de linhas) também igual a n . O meio (maior linha) do diamante deve ter n espaços. O Diamante deve ser preenchido externamente pelo caractere "o", como mostra a figura abaixo, gerado com os números $n = 7$ e $n = 11$. O valor de n lido deve ser $3 \leq n \leq 15$. O número n deve ser ímpar.

```
octave:11> diamante_vazio
entre com o número de linhas do diamante (inteiro IMPAR): 7
ooo ooo
oo  oo
o   o

o   o
oo  oo
ooo ooo
octave:12> diamante_vazio
entre com o número de linhas do diamante (inteiro IMPAR): 11
ooooo ooooo
oooo  oooo
ooo   ooo
oo    oo
o     o

o     o
oo    oo
ooo   ooo
oooo  oooo
ooooo ooooo
```

5 Histograma

12. Faça um script para preencher um vetor, V , de dimensão $N \times 1$ com números aleatórios entre 1 e 15. N é um número inteiro dado pelo usuário, tal que $1 < N \leq 10$. Uma vez preenchido o vetor, o script deve gerar o histograma do valores de cada linha. Deve-se mostrar a posição do valor (considere 1 a primeira linha), o valor e o desenho (com asteriscos) com o número de asteriscos correspondente ao valor da linha. Veja o exemplo abaixo.

Considere o vetor V , de tamanho $N = 5$:

$$V = \begin{bmatrix} 5 & 9 & 3 & 10 & 6 \end{bmatrix}$$

Nesse caso, o seu "histograma"ficaria:

```
1   5   * * * * *
2   9   * * * * * * * * *
3   3   * * *
4  10   * * * * * * * * *
5   6   * * * * * *
```

6 Funções

13. Faça um script contenha um menu de 5 opções. Quatro opções para escolher uma das funções e a quinta opção para sair do laço (loop). O script deve apresentar o menu até que o usuário escolha sair. As funções desse programa são:
- (a) Uma função que recebe um número inteiro n e retorna 0 se o inteiro for par e 1 se o inteiro lido for ímpar. O script principal deve receber o valor retornado pela função e escrever uma mensagem, mostrando o número n dizendo se ele é par ou ímpar.
 - (b) Uma função que recebe dois números inteiros: n e x e retorna 0 se n for divisível por x e 1 se n NÃO for divisível por x . O script principal (main) deve escrever a mensagem dizendo se n é ou não divisível por x .
 - (c) Uma função que recebe um inteiro n e retorna seu fatorial. O script principal deve imprimir uma frase com n e seu fatorial.
 - (d) Uma função que recebe um inteiro n e um float x e retorna o valor de $x^{\frac{1}{n}}$.

14. Faça um script que contenha um menu para chamar 3 funções diferentes e uma opção para sair do script. O script deve ficar rodando o menu até o usuário optar por sair. O menu deve ficar sempre "ativo", enquanto o script estiver rodando. Além de ter o menu o script deve ter 3 funções:

- (a) Uma função para ordenar um vetor de 3 posições. A função deve receber dois vetores. Um vetor com 3 números inteiros, M e um vetor vazio N . A função deve ordenar os números que estão em M , colocando-os em N da seguinte forma:

$$N[1] = \text{menor valor de } M \quad (1)$$

$$N[3] = \text{maior valor de } M \quad (2)$$

$$N[2] = \text{o outro valor de } M \quad (3)$$

O script principal deve carregar o vetor M , lendo os dados do usuário. Depois deve chamar a função. A função ordena os 3 valores de M em N . O script principal (após chamar a função) deve imprimir os valores de M e N .

- (b) Uma função para calcular o somatório pedido abaixo. A função deve receber os limites do somatório (N_0 e N_f) e retornar o resultado da soma.

$$S = \sum_{i=N_0}^{N_f} \frac{i}{i-1}$$

Antes de chamar a função, no script principal, deve-se verificar se $N_f \geq N_0$. Caso não seja, o script deve continuar pedindo os valores até que a condição $N_0 \leq N_f$ seja satisfeita.

- (c) Uma função que recebe um vetor M (de tamanho máximo 6) e um inteiro positivo N (N pode assumir valores entre 1 e 6), correspondente ao tamanho de um vetor. Com esse valor a função deve criar preencher o vetor M de tamanho N , preenchido com a regra

$$M[i] = 2i,$$

O script principal deve imprimir o vetor gerado. A função deve apenas preencher o vetor. O script principal deve verificar, antes de chamar a função se o N escolhido é maior que zero e possui valor máximo 6. Caso N seja maior que 6, o script principal deve definir $N = 6$. Caso N seja negativo, deve-se fazer $N = -N$.

15. Faça um script que contenha um menu para chamar 2 funções diferentes e uma opção para sair do script. O script deve ficar rodando o menu até o usuário optar por sair. O menu deve ficar sempre "ativo", enquanto o script estiver rodando. Além de ter o menu o script deve ter 2 funções:

- (a) Uma função para calcular as raízes reais de uma equação de segundo grau do tipo

$$ax^2 + bx + c = 0$$

A função deve receber os valores a , b e c além de um vetor (vazio) com duas posições. Esse vetor irá receber as duas raízes calculadas. A função deve retornar o valor inteiro 1 se houver soluções reais e um valor inteiro zero se não houver soluções reais. Além disso a função deve colocar no vetor os valores das raízes x_1 e x_2 , calculadas. No script principal, após chamada a função deve-se mostrar os valores encontrados para as raízes. Caso não tenha solução real, deve-se apresentar uma mensagem dizendo que não há soluções reais.

- (b) Uma função para calcular o somatório pedido abaixo. A função deve receber os limites do somatório (N_0 e N_f) e retornar o resultado da soma.

$$S = \sum_{i=N_0}^{N_f} \frac{2i + 5}{i + 1}$$

Antes de chamar a função, no script principal, deve-se verificar se $N_f \geq N_0$. Caso não seja, o script deve continuar pedindo os valores até que a condição $N_0 \leq N_f$ seja satisfeita.

7 Somatórios

Questão 1

Fazer um algoritmo que calcule e escreva o valor de $S = S_1 + S_2$, com S_1 dado por

$$S_1 = \frac{99}{50} + \frac{97}{49} + \frac{95}{48} + \dots + \frac{7}{4} + \frac{5}{3} + \frac{3}{2} + \frac{1}{1}$$

e S_2 dado por

$$S_2 = \frac{2^{\frac{1}{10}}}{50} + \frac{2^{\frac{2}{10}}}{49} + \frac{2^{\frac{3}{10}}}{48} + \dots + \frac{2^{\frac{50}{10}}}{1}$$

Questão 2 Fazer um algoritmo que calcule e escreva a soma dos N primeiros termos da série abaixo:

$$\frac{1000}{1} - \frac{997}{2} + \frac{994}{3} - \frac{991}{4} + \dots$$

N é um número inteiro positivo informado pelo usuário e pode variar de 2 a 50, isto é $2 \leq N \leq 50$.

Questão 3

Fazer um algoritmo para calcular e escrever o valor do número π , com precisão de δ (sendo δ um número dado pelo usuário), usando a série:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots \quad (4)$$

- Para definir a precisão do π calculado por você, efetue a diferença dele com o π dado pelo Octave.

Questão 4 Faça um script para somar os 20 primeiros números da série

$$R = \frac{\cos(\pi)}{\pi} - \frac{3 \cos(2\pi)}{2\pi} + \frac{5 \cos(3\pi)}{3\pi} - \frac{7 \cos(4\pi)}{4\pi} + \dots$$

Resultado: R =

Questão 5

Faça um script para somar os 30 primeiros números da série

$$R = \frac{\cos(\pi)}{3\pi} + \frac{2 \cos(2\pi)}{4\pi} + \frac{3 \cos(3\pi)}{5\pi} + \dots$$

Resultado: R =

Questão 6

Faça um script para efetuar o somatório dado abaixo. Ao final, o script deve imprimir na tela: "O somatório vale: xxxxx", onde xxxx é o valor calculado.

$$S = \sum_{i=-10}^{10} \frac{2i+1}{i+1} + \sum_{i=-10}^{10} \frac{i+1}{i-1}$$

Resultado: S =

Questão 7 Faça um script para somar os 20 primeiros números da série

$$R = \frac{\cos(1)}{\pi} + \frac{\sin(1/2)}{2\pi} + \frac{\cos(3)}{3\pi} + \frac{\sin(1/4)}{4\pi} \dots$$

Resultado: S =