07/03/07 – 2ª aula de Matlab 15:30 – 17:00

Exercícios: loops, gráficos

Funções: “if”,”for”,”plot”,”hold”, “xlabel”, “ylabel”, “figure”, “title”, “warning”, “length”, “normrnd” , “repmat”.

Explicar o que são “loops” e pra que servem:

- iterações de cálculos com variáveis aleatórias
- acumulação de valores

Loops

Estude a função **for**.

Dica: digite os exercícios primeiramente no editor de textos (o do matlab é melhor para isso (para abri-lo, basta clicar no ícone de “folha em branco”, à esquerda, logo abaixo do campo “arquivo). Depois de pronto e revisado, você pode copiar e colar para a janela *command window* do matlab. Isso faz você ganhar tempo, pois, se ao escrever diretamente na janela de comando você errar algo, o matlab começa a enviar mensagens de erro e o seu texto fica todo fragmentado.

Dica: nos exercícios que pedem muita digitação você pode ganhar tempo copiando o que for possível deste guia e colando diretamente no editor de texto do matlab. Lá você faz os ajustes e complementos necessários e depois copia e cola o algoritmo na *command window* para que ele corra.

Digite:

```
for x = 1:5
a(x) = 5 * x ;
end
a
clear
```

Digite:

```
a = 0
for x = 1:5
a = a + x;
end
a
clear
```

Repita agora esse exercício retirando o símbolo ";" da linha `a = a + x`.

Digite: `clear`

Estude a função **if**.

Imagine um viveiro de mudas com capacidade de produzir até 10000 mudas de diversas espécies arbóreas tropicais, por mês. Um funcionário é capaz de cuidar sozinho de até 5000 mudas/mês e seu salário é de R$1000,00/mês. Se a demanda for maior do que 5000 mudas /mês, será necessário contratar um ajudante ao custo de R$ 800,00/mês. O custo unitário médio de cada muda é de R$ 0,10/mês.

Digite:

```
mudas = 3000;
salario1 = 1000;
salario2 = 800;
if mudas > 5000
custo = .10*mudas +salario1 + salario2
else
custo = .10*mudas +salario1
end
```

Agora, repita o loop if – end após cada uma das seguintes alterações:

mudas = 5000

mudas = 5010

mudas = 7000

mudas = 10000

Digite: `clear`

Estude a função **warning**.

Digite:

```
adults = 10;
juveniles = 50;
seedlings = 100;
Et = 70; % Limite arbitrariamente escolhido para o tamanho da população, abaixo do
                qual a mesma pode ser considerada em rota de extinção.
Rt = 40; % Limite arbitrariamente escolhido abaixo do qual deve-se assumir que a
                população está com dificuldades de se regenerar.

population = [adults juveniles seedlings];
if sum(population) <= Et & seedlings < Rt
warning('population undergoing extinction risk and probable recruitment
                problems')
elseif sum(population) <= Et
warning('population under extinction risk')
elseif seedlings < Rt
warning('probable recruitment problems')
else
warning('apparently healthy population')
end
population'
```

Observação: Se você copiar este algoritmo do word para o matlab provavelmente terá de refazer os sinais " ' " nas mensagens *warning* e na última linha. Pode haver um problema de conversão desse sinal.

Observação: o símbolo " % " permite que você insira comentários ao longo do algoritmo. Nada do que vier no parágrafo iniciado com % será considerado pelo matlab nos cálculos. Insira vários desses comentários quando estiver programando, pois eles serão muito úteis em futuras leituras do algoritmo, tanto quando você quiser corrig-lo como quando você quiser aperfeiçoá-lo.

´Para cada um dos seguintes ajustes, corra o algoritmo novamente (de population a population’):

seedlings = 30
seedlings = 40 e juveniles = 10
seedlings = 0

Estude a função **length**.

Dica: desta vez, ao invés de copiar e colar, digite o algoritmo no editor de textos do matlab. Assim você poderá ver como o matlab “indenta” naturalmente as linhas de comando pertencentes a loops. Essa “indentação” é muito útil para que você consiga enxergar mais facilmente loops dentro de outros loops e ajuda você a não esquecer de fechar os loops com a função **end**. Em uma próxima oportunidade, se você achar mais conveniente copiar e colar, você pode reproduzir essa indentação com a tecla “tab” do seu teclado.

Digite:

```
b = zeros(3)
for i = 1:length(b)
for j = 1:length(b)
if i == j
b(i,j) = 1;
else
b(i,j) = 0;
end
end
end
b
clear
```

A partir da tabela abaixo, construa uma matriz quadrada de transição A cujas entradas aij representem as taxas de transições de indivíduos das classes j para as classes i.

Construa um algoritmo genérico, ou seja, um que possa ser utilizado para transformar qualquer tabela que traga o mesmo tipo de informações da tabela dada, independentemente da dimensão dessa tabela e dos valores que ela trás.

Dica: você precisará do loop **for** e das funções **length** e **sum**, pelo menos.

tabela = [50 0 0 0 0;15 25 3 0 0;0 4 11 2 0;0 0 2 6 1; 0 0 0 1 3;20 10 5 2 1]

A tabela abaixo representa uma população fictícia. As colunas representam as classes de origem dos indivíduos em *t* enquanto que as linhas representam seus destinos em *t+1*. A última linha da tabela traz o número de indivíduos que morreram ente *t* e *t+1*.

Resultado esperado:
A=[0.5882,0,0,0,0;0.1765,0.6410,0.1429,0,0;0,0.1026,0.5238,0.1818,0;0,0,0.0952,0.5455,0.2000;0,0,0,.0909,0.6000]

Gráficos

Estude as funções **plot**, **title**, **ylabel**, **xlabel**,e **normrnd**.

Digite:

```
for x = 1:100
a = normrnd (50,5);
b = 50;
y(x) = a*x + b;
end
x = (1:100);
```

Crie um gráfico cartesiano com y no eixo das ordenadas e x no eixo das abscissas. Dê ao gráfico o título “teste”, às abscissas o título “x” e às ordenadas o título “y”.

Estude a função **figure**.

Digite: figure

Estude agora a função **hold**.
Crie agora um loop de 50 repetições para o algoritmo que produziu y no exercício anterior e plote essas 50 novas curvas em um mesmo gráfico.

Exercício extra:

Crie um algoritmo genérico (literal) que estenda os valores numéricos de uma matriz em vetores de tamanhos equivalentes. Parta da matriz a = [8 4 1;0 3 5;0 0 10]. Após inseri-la, digite:

```
K = length(a); < enter>
b(1:k) = {repmat({[0]},1,k)} <enter>
```

Estude a função **repmat**.

Agora, crie o loop que irá gerar os vetores.

Ao final, confira os seus resultados, limpe sua *worspace* e finalize sua sessão de matlab.