Introdução à programação em MATLAB

Interacção com o matlab

- Através do interpretador (linha de comando)
- Com ficheiros batch (ou M-files)
  - Ficheiros de texto com sequências de comandos
- Com funções
  - Têm parâmetros de input/output
- Com GUI (Graphical User Interface)
- Com um pouco de todos…

Ficheiros Batch (M-Files)

- Ficheiros de texto com comandos (*.m)
  - Os ficheiros são pré-compilados a primeira vez
  - Usar o editor do MATLAB para manter o syc,
- Partilham o *workspace* com o interpretador
  - Variáveis são comuns
- Utilizar cut&paste
  - Editor/interpretador
- Utilizar “cells”
  - Pedaços de código limitados por “%%”

Exemplo de M-File

- 1º - Gerar um seno, e desenhá-lo
- 2º - Mudar a frequência e re-desenhar
- 3º - Gerar a soma de senos de frequência diferente

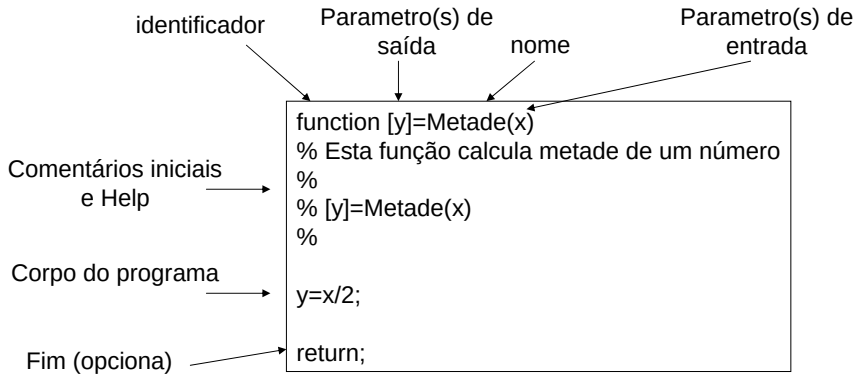
- 1º Representar um sistema de equações lineares com matrizes
- 2º Resolver o sistema em Matlab, e alterar os dados

Funções

- Ficheiros de texto
  - extensão “.m”
  - Começam com a palavra chave “function”
  - Têm parâmetros de entrada e saída
  - As variáveis internas são locais
  - Permitem incorporar “help”
  - Chamadas a funções

Exemplo de uma função

- Calcular metade de um número



Escrever no ecrã / ler o teclado

- Escrever no display
  - disp(x)
  - Exemplos:

```
x=5;
disp( x );
disp('ola');
```

- Ler o teclado
  - x=input()
  - Exemplos

```
X=input('qual o valor de x ?');
```

- Fazer uma pausa
  - Pause

Escrever / ler ficheiros

- Similar ao C
  - fopen – Abre o ficheiro
  - fprintf – Escreve no ficheiro
  - fscanf – Lê do ficheiro
  - flcose – Fecha o ficheiro
- CUIDADO:
  - Todas as variáveis são matrizes.
  - Um *printf* escreve toda a matriz.
- Pode-se sempre usar o *wizard* !

```
fid=fopen('exemplo.txt','r');
fid2=fopen('saida.txt','w');

A=fscanf(fid,'%d');

fprintf(fid2,'%d %d\n',A);

fclose(fid);
fclose(fid2);
```

Estruturas de programação

- Ciclos FOR
  - for *variável=expressão*
    - *instruções*
  - *end;*

```
for i=1:10
    a=a+i;
end;
```

```
for i=[ 3 1 9 5]
    b=i*2;
    a=a+b;
end;
```

- CUIDADO:
  - For são *lentos*
  - Deve-se usar paralelismo
  - Exemplo $S = \sum_{n=1}^{5000} \frac{1}{n^2}$

```
tic;
s=0;
for n=1:5000
    s=s+1/n^2;
end;
tempo=toc;
```

```
tic;
n=1:5000;
s=sum(1./n.^2);
tempo=toc;
```

Estruturas de programação (2)

- Ciclos WHILE
  - while *expressão*
    - *instrução*
  - end;

```
while a>2
    disp(a);
    a=a/2;
end;
```

- Blocos IF
  - if *expressão*
    - Instrução
  - else
    - *instrução*
  - end

```
if x>10
    disp('ola');
else
    disp('adeus');
end;
```

Estruturas de programação (3)

- Blocos switch (case)
- switch(*expressão*)
  - case valor1
    - *Instrução*
  - case valor2
    - *Instrução*
  - otherwise
    - *Instrução*
- end;

```
switch(x)
   case 1
      disp('É um');
   case {2,3,4}
      disp('É um pouco');
   otherwise
      disp(('nã sei');
end;
```

Estruturas de dados (1)

- Estruturas
  - Variáveis com campos
    - Cada campo tem um nome
    - Cada campo pode ter seu tipo
  - Pode ter sub-estruturas
  - Pode ter ou ser uma matriz

```
ficha.nome='Manel';
ficha.telef=311525;
ficha.nota=15;

ficha =

    nome: 'Manel'
   telef: 311525
    nota: 15
```

- Utilizadas por ex.para manipular gráficos

Estruturas de dados (1)

- Células
  - Variáveis que são de qualquer tipo
  - Matriz de células pode misturar tipos
  - Acedidas com {…} em vez de (…)
  - Células/matrizes
    - Num2cell
    - Mat2cell
    - cell2mat

```
a={ 'ola', 3 , 4 }

a =

    'ola'    [3]    [4]
```

Utilização do debugger

- Correr o programa passo a passo
  - F10/F11
- Breakpoints
- Acesso às variáveis
  - Com o cursor
  - Na linha de comandos

Problemas

- Fazer uma rotina para resolver equações do 2º grau
- Fazer uma rotina para calcular uma distância de hamming entre dois vectores, e depois entre matrizes com vectores
- Fazer uma rotina para calcular uma distância de minkowski
- Fazer uma rotina para implementar o algoritmo de k-medias

Rotinas úteis para a disciplina de SAD

Curso de introdução ao MATLAB

Toolboxes da Mathworks

- Statistics toolbox
  - Classificação hierarquica (árvores), PCA, gráficos, etc…
- Neural Networks toolbox
  - Redes neuronais MLP, perceptrões, RBF, etc…
- Fuzzy Logic toolbox
- Excel link
- Signal processing toolbox
- Optimization toolbox
- Symbolic Math toolbox

Toolboxs úteis (freeware)

- SOMTOOLBOX (versão 2.0)
  - http://www.cis.hut.fi/projects/somtoolbox/download
  - SOM, k-médias, sammon mapping, etc…
- NetLab (versão 3.3)
  - http://www.ncrg.aston.ac.uk/netlab
  - Redes neuronais, PCA, K-médias, Estimadores Bayesianos, etc.

Instalação de toolboxes

- Copiar os ficheiros para uma directoria
- Acrescentar essa directoria ao path
  - Com o comando path
  - Com o pathtool

SOMTOOLBOX

- Suporte para SOM e…
  - K-médias
  - Sammon mapping
  - U-Mat
  - Visualização
- Estruturas base
  - Dados (Têm que ser importados para uma estrutura própria)
  - Mapas (Registam o treino feito)

Utilização da somtoolbox

```
load iris                                   % Load Original Data
sD=som_data_struct(iris);                   % Convert data to SOM
sD=som_normalize(sD,'var')                  % Normalize by Z-score
                                            % Initialize a map with 50 units (10x5)
sM = som_randinit( sD, 'msize',[10 5],'rect','sheet');
sM.neigh = 'bubble';
                    % Establish training parameters (1st phase)
niterations_1 = 1000; radius_ini_1 = 4; alpha_ini_1 = 0.6;
                    % Establish training parameters (2nd phase)
niterations_2 = 2000; radius_ini_2 = 2; alpha_ini_2 = 0.1;
                    % Train the SOM
sM1 = som_seqtrain(sM,sD,'radius_ini',radius_ini_1, 'radius_fin',0,...
      'alpha_ini',alpha_ini_1,'trainlen', niterations_1, 'epochs');
sM2 = som_seqtrain(sM1,sD,'radius_ini',radius_ini_2,'radius_fin',0,...
      'alpha_ini',alpha_ini_2,'trainlen', niterations_2, 'epochs');

som_show(sM2)                               % Mostra o resultado do SOM
```

Visualização de dados

- Gráficos de dispersão
- Coordenadas paralelas
- Radar plots
- Caras de Chernoff

Radar Plots

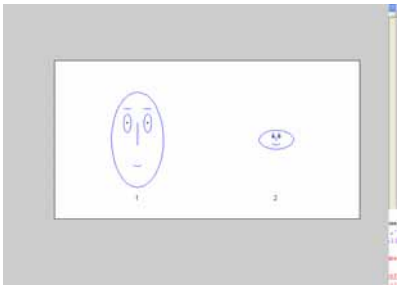
- glyphplot(x)
  - Faz um radarplot de x, normalizando as variáveis para o intervalo [0,1]
  - Exemplo
    - X=[1 2 3; 0 2 0; 2 1 1];
    - glyphplot(X);

(sem normalização)

- X=[0.3 0.6 0.95; 0.6 0.6 0.6; 0.6 0.3 0.3];
- glyphplot(X,'Standardize','off');

Caras de chernoff

- glyphplot(X, 'Glyph','face')



Column	Facial Feature
1	Size of face
2	Forehead/jaw relative arc length
3	Shape of forehead
4	Shape of jaw
5	Width between eyes
6	Vertical position of eyes
7	Height of eyes
8	Width of eyes (this also affects eyebrow width)
9	Angle of eyes (this also affects eyebrow angle)
10	Vertical position of eyebrows
11	Width of eyebrows (relative to eyes)
12	Angle of eyebrows (relative to eyes)
13	Direction of pupils
14	Length of nose
15	Vertical position of mouth
16	Shape of mouth
17	Mouth arc length

Análise de componentes principais

- Com o statistics toolbox
  - pcacov
    - PC -> vectores proprios
    - Variances -> Valores próprios

```
load iris
covx = cov(iris);
[pc,variances,explained] = pcacov(covx)
```

Utilização de clustering hierárquico e árvores de decisão

- Clustering hierarquico
  - linkage – Gera clusters hierárquicos

```
Y=pdist(T(:,1:3));
Z=linkage(Y);
dendrogram(Z);
```

- Árvores de decisão
  - classregtree - Gera uma árvore de decisão

```
t = classregtree(X,y)
view(t)
```

- GUI

Outros algoritmos úteis, com interface GUI

- Algoritmos genéticos
  - gatool
- Redes neuronais
  - Nntool
- Sistemas difusos
  - fuzzy