

Introdução ao Matlab

Objectivos:

- 1- Ficar com uma ideia do que é
- 2- Ficar com uma ideia das suas potencialidades
- 3- Ser capaz de fazer manipulação simples de matrizes
- 4- Ser capaz de facilmente aprender MATLAB

V.Lobo, Escola Naval

1

História

- Bibliotecas de análise numérica (para manipulação de matrizes), nomeadamente a LINPACK de álgebra linear
- Fins dos anos 70: primeira versão de MATLAB (MATrix LABoratory), com um interpretador
- Muito sucesso em “computação científica”
 - Investigação
 - Cursos de engenharia
 - Empresas que necessitam de simulação numérica, ciclos de desenvolvimento curtos, e boa visualização de dados
 - Livros, Programas de simulação, Utilitários diversos

2

Situação actual

- Disponível em MS-Windows, UNIX, e Macintosh
 - AIX, Digital UNIX, HP-UX 10, HP-UX 11, IRIX/IRIX64, Linux, Macintosh 68000, Open VMS, Power Macintosh, Solaris, SunOS 4, MS-Windows.
- Versão 6 (Release 12)
 - Melhor interface gráfico, melhor debugger, mais funções, melhor interface com hardware de aquisição
- MATLAB vem com o programa “principal” (o motor) e o resto da funcionalidade é dada por rotinas escritas em MATLAB, as TOOLBOXES:
 - Signal processing, Financial analysis, Neural Networks, Wavelet, Fuzzy Logic, Optimization, Non Linear Control, Data Acquisition, Statistics, Filter Design, etc., etc, etc.
- Versão para estudantes (p/MS-Windows) é barata e tem as toolboxes mais importantes

3

Motivação

- Razões de preferência
 - Rotinas numéricas muito boas (rápidas, fiáveis, fáceis de usar)
 - Sistema de visualização de dados excelente (potente, fácil de usar)
 - Interpretador interactivo
 - Permite fazer o processamento sem escrever nenhum programa
 - Blocos para programação visual
 - Simulink, DSP blockset, etc
 - Compilador
 - Permite linkar com programas em C, Labview, etc
- Áreas de aplicação
 - Processamento de sinal, Controlo, Visualização de dados, etc...

4

Princípios básicos

- Modo de utilização
 - Linha de comandos do interpretador (comandos “isolados”)
 - Pode ser também usado como linguagem de programação (com ciclos for, while, blocos if, etc)
- As variáveis são todas MATRIZES
 - Um escalar é uma matriz de 1 x 1
 - Um vector é uma matriz de 1 x N
 - A partir da versão 5, há matrizes multi-dimensionais (N x N x N x...)
- As variáveis são criadas / redimensionadas em run-time
 - Não é necessário declarar variáveis (embora possa acelerar o prog.)
- Todas as operações são matriciais
 - A * B ⇒ Multiplicação matricial
 - A .* B ⇒ Multiplicação ponto a ponto das matrizes A e B

5

Operações básicas

- Operadores aritméticos

- Soma	+	A+B
- Subtração	-	A-B
- Multiplicação	*	A*B
- Divisão à direita	/	A/B
- Divisão à esq.	\	A\B
- Transposição	'	A'
- Potenciação	^	A^B
- Operadores relacionais

- Igualdade	==	A==B
- Desigualdade	~=	A~=B
- Menor, maior, etc	<	A < B (tb. <=, >, >=)
- Outros

- Um elemento de uma matriz	()	A(3)
- Uma sub-matriz	:	A(3:7) - tb. A(3:end)
- Construção de uma matriz	[]	A = [1 2 3]
- Mudança de linha	;	A = [1 2 3 ; 4 5 6]

Todas admitem a versão escalar (ponto qq coisa) ./, .*, .^, etc

6

Comandos que “dão jeito”

Introdução ao MATLAB

- **WHOS**
 - Mostra que variáveis existem
- *Nome_da_variável*
 - Mostra o valor da variável
- **PLOT(variável)**
 - Grafica a variável
- **[parametros_de_retorno]=Função(parametros_de_entrada)**
 - Chama uma função. Por ex: [espectro, F]=psd(sinal,npontos)
- **HELP um_comando**
 - Mostra a ajuda para esse comando
- *um_comando ;*
 - Impede o resultado do comando de ser impresso no ecrã

7

Exemplos

Introdução ao MATLAB

- **Visualizar a transformada de um sinal**
 - Gerar um sinal com duas sinusoides
 - $t=1:0.1:100$
 - $x=\sin(\pi/10*t)+0.5*(\pi/5*t)$
 - Calcular o módulo da transformada de Fourier
 - $X=\text{abs}(fft(x))$
 - Visualizar o resultado
 - $\text{plot}(X)$
- **Processamento de um sinal de voz**
 - Ler um ficheiro WAV
 - Função WAVREAD(...)
 - Ouvi-lo, e ver o espectrograma
 - Função WAVPLAY(...), SPECGRAM(...)
 - Calcular a transformada de fourier, e deslocá-la
 - Voltar ao domínio do tempo e ouvir o resultado

8

Command files & funções

Introdução ao MATLAB

- **Command files**
 - São ficheiros de texto com sequências de comandos
 - Não têm parâmetros nem variáveis locais
- **Funções**
 - Têm parâmetros de entrada, e parâmetros de saída (os parâmetros de entrada são sempre passados por valor)
 - As variáveis usadas numa função são locais
 - Conceito de Workspace
 - As primeiras linhas de comentário da função funcionam como o seu “help”
 - Deve-se evitar sempre que possível os ciclos FOR, pois estragam o processamento vectorial

9

Utilização de ficheiros de comandos

Introdução ao MATLAB

- **É possível ir fazendo “cut & paste” entre o editor e a janela de comandos**
 - Testar todas as linhas de código antes de as executar
 - Utilizar extensivamente o HELP
 - Ficar com um registo dos comandos dados (NOTA: pode-se usar também a janela de históricos para obter essa informação)
 - Exemplo:
- **É possível usar o DEBUGGER para verificar interactivamente o programa**

10

Funções úteis para visualizar sinais

Introdução ao MATLAB

- **Para importar/exportar sinais**
 - Em formato matlab: load/save
 - Em formato texto: load/save -ascii
 - Em formato excel: usando o “excel link wizard”
 - Em formato WAV: wavread/wavwrite
 - Parâmetros N,FS
- **PLOT com argumentos**
 - Plot(x,'r-')
 - Utilizar HOLD ON para mostrar vários gráficos
- **WAVPLAY e SOUND**
 - Parâmetro FS

11