MATLAB

Workshop introdutório

Gonçalo Pena

Departamento de Matemática da UC

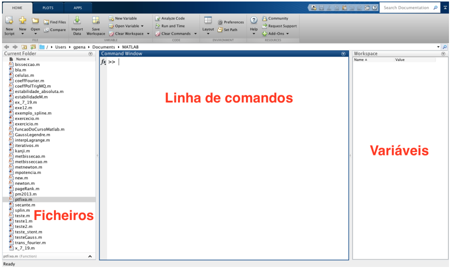
2022

Resumo

Introdução

MATLAB: o que é e para que serve?

- O nome origina de ≪MATrix≫ e ≪LABoratory≫
- Sistema de cálculo (numérico) científico
- Tem quatro componentes principais:
  - a linguagem
  - o ambiente de trabalho
  - gráficos
  - sistema de ≪toolboxes≫



Linha de comandos

- Permite a criação de variáveis e introdução de instruções a executar

Exemplo

```
N = 10
```

inicializa uma variável `N` com o valor `10`, que pode ser utilizada em instruções seguintes.

Exemplo

```
help sin
```

imprime na linha de comandos informações sobre a função `sin`.

Editor

- Editor de texto com coloração para as palavras chave usadas pela linguagem do MATLAB
- Boa ferramenta para encontrar erros no código
- É utilizado para editar ficheiros com extensão ≪.m≫ (extensão usada pelo MATLAB)

ptfixo.m

```
function x1=ptfixo(phi,x0,epsilon)

erro=1+epsilon;
x1=x0;

while (erro >= epsilon)
    x1 = phi(x0);
    erro=abs(x1-x0);
    x0=x1;
end
```

Exercício

Escreva as instruções do slide anterior no editor, grave o ficheiro com o nome exercicio1.m e execute-o na linha de comandos.

Definir variáveis

Números reais/complexos

- Números representados em precisão dupla
- A *soma*, *subtração*, *multiplicação*, *divisão* e *potência* estão disponíveis com a notação usual:

Exemplo

```
r = 2;   s = r + r;
t = r/4; w = s*t^2
```

Lista de (algumas) operações:

sqrt	exp	sin	cos
log	sinh	cosh	tan
factorial	atan		

Exercício

Defina uma variável `x = 2`. Calcule

1. `factorial(x+6)`
2. `(x+1)^4`
3. `log(x-1)`
4. `exp(x)`
5. `sin(x), cos(x)`
6. `(sin(x))^2 + (cos(x))^2`
7. `sin(sin(sin(x)))`

Vetores

- Dois tipos: linha ($1 \times N$) ou coluna ($N \times 1$)
- Como definir?

Exemplo

```
X = [ 1 2 3 4 5 6 7 ]
Y = [ 1;2;3;4;5;6;7 ]
```

X - vetor linha (1×7)
Y - vetor coluna (7×1)

NOTA: ≪;≫ faz a mudança de linha na introdução de valores

- Como definir vetores de maiores dimensões, por exemplo, um vetor linha com os 100 (ou 1000) primeiros inteiros positivos?

Vetores

- Alternativas:

Exemplo

```
T = 1:2:10
U = 0:2:10
```

elementos de T e U em progressão aritmética ($r = 2$)

Exemplo

```
Z = zeros(1,10); W = ones(5,1)
```

Exemplo

```
x = linspace(0, 1, 100)
```

Aceder a componentes de vetores

Exemplo

```
x = [ 5 2 -4 0 10 -20 9 ]
```

- `x(i)` - componente *i* de `x`
- `x([2 5])` - componentes 2 e 5 de `x`
- `x(2:5)` - componentes 2 a 5 de `x`
- `x(2:1:5)` - equivalente ao anterior
- `x(2:end)` - componentes 2 a 7 de `x`

Exercício

Defina um vetor `Y` com os 100 primeiros inteiros positivos ímpares. Extraia para um vetor auxiliar todas as componentes nesse vetor correspondentes às posições 1, 2, 4, 8, 16, 32 e 64.

Operações com vetores

Exemplo

```
V = [ 5 2 -4 0 10 -20 9 ]
```

- `length(V)`: número de componentes
- `sum(V)`: soma das componentes
- `mean(V)`: média aritmética das componentes
- `size(V)`: número de linhas e colunas
- `norm(V)`: norma euclideana
- `diag(V)`: matriz diagonal

Operações com vetores: soma, subtracção, multiplicação, transposição, etc (ver secção sobre matrizes)

Matrizes: como definir?

$$A = \begin{bmatrix} -1 & 7 \\ 9 & 5 \\ 2 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 10 & 1 \\ 5 & -4 & -2 \end{bmatrix}$$

Exemplo

```
A = [ -1 7; 9 5; 2 1 ]
B = [ 0 10 1; 5 -4 -2 ]
```

define matrizes 3×2 e 2×3, respetivamente.

- As funções `zeros` e `ones` também se podem usar
- `eye(n,m)` gera uma matriz $n \times m$ de zeros, com 1 nas entradas da diagonal principal
- `A(i,j)`: componente (i, j)

Matrizes: operadores algébricos

Exemplo

```
A = [ 3 -1; 2 0 ]
B = [ 9 5; 1 2 ]
```

- A+B: soma
- A−B: subtração
- A∗B: multiplicação
- 2∗B: multiplicação por um escalar
- A^2: potência

No caso da multiplicação, potência e divisão, podem definir-se:

- A.∗B: multiplicação termo a termo
- A./B: divisão termo a termo
- A.^2: potência termo a termo

Matrizes: outros operadores

- `A'`: transposta
- `inv(A)`: inversa
- `det(A)`: determinante
- `rank(A)`: característica
- `norm(A,p)`: norma-p (eg, p=1, p=2, p='Inf')
- `eig(A)`: valores e vetores próprios
- `diag(A)`: extrai a diagonal principal
- `abs(A)`: valor absoluto

Exercício

Considere `A=[1 2; 3 4]`. Calcule

1. A^4 e `A.^4` (compare os resultados)
2. os valores próprios de A
3. A^{-1} e `1./A` (compare os resultados)

Matrizes: outros operadores

Todas as funções trigonométricas, logarítmicas, exponencial, podem ter como argumento uma matriz. O resultado é a aplicação da função, componente a componente, à matriz dada.

Exercício

Dada `A = [ 1 2; 3 4; 5 6 ]`, calcule `sin(A)`, `cos(A)`, `tan(A)`, `atan(A)`.

Exercício

Calcule uma aproximação para $\int_0^\pi \cos(x)\sin(x)dx$ usando a regra do retângulo.

Regra do retângulo:

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=0}^{N-1} f(x_i),$$

onde $x_i = a + \frac{b-a}{N}i$, $i = 0, \ldots, N-1$.

Matrizes: o operador $\backslash$

Dada uma matriz $N \times N$, `A`, e um vetor $N \times 1$, `b`, o comando `A \ b` calcula a solução do sistema linear $Ax = b$.

Exercício

Considere `A=[1 2; 3 4]` e `b=[1;2]`. Calcule a solução de $Ax = b$

1. usando a inversa de A
2. usando o operador $\backslash$

NOTA: O operador $\backslash$ analisa a estrutura da matriz A e escolhe o método (directo) que considera «mais indicado» para resolver o sistema.

Alternativas:

- `[L,U,P] = lu(A)` fatorização LU
- PCG/GMRES

Controlo de fluxo

Controlo de fluxo

Existem duas instruções de controlo de fluxo em MATLAB:

- `for`
- `while`

Exemplo

Soma de todos os ímpares até 10:

```
x = 0;
for i=1:2:10
       x = x + i;
end
```

Controlo de fluxo: ciclo while

Exemplo

Usar a instrução `while` no exemplo anterior:

```
x = 0; i = 1;
while ( i < 10 )
        x = x + i;
        i = i + 2;
end
```

Representação verdadeiro/falso:

- `0` - `false`
- `1` - `true`

Lógica

Como construir condições para o ciclo while:

- operadores de relação: `<`, `>`, `<=`, `>=`, `==`, `~=`
- operadores lógicos
  - `A & B` ou `and(A,B)`
  - `A | B` ou `or(A,B)`
  - `~A` ou `not(A)`

Exemplo

Verifique se são verdadeiras ou falsas as seguintes proposições:

- `0 < 1`
- `0 == 1`
- `0 ~= 1`

Algumas sugestões de otimização

Preferir ciclos `for` a ciclos `while` sempre que o número de iterações seja conhecido à partida

Exemplo

Exercício: Executar na linha de comandos as seguintes instruções:

```
x=1:1000000;
tic; sum(x); toc
soma=0; tic; for i=1:1e6;
   soma=soma+x(i); end; toc
soma=0; i=1; tic; while (i<=1e6);
   soma=soma+x(i); i=i+1; end; toc
```

Instrução condicional

Instrução condicional: if

A instrução `if` tem uma sintaxe muito simples:

Exemplo

```
x = 1;
if (x == 1)
    disp ('um');
else
    disp ('diferente de um');
end
```

Funções

Funções

Podemos definir dois tipos de funções: de forma *inline* (mais simples) ou através de um ficheiro de extensão .m.

Exemplo

Função definida como *inline*

```
f = @(x)(x.*sin(x))
```

O primeiro argumento define qual a variável (ou variáveis) independente da função; o segundo argumento é uma sequência de carateres que define a expressão da função.

Alerta: entre variáveis que dependem de x, devem usar-se operações ≪componente a componente≫.

Funções

Exemplo

Cálculo de `f` nos valores de `abcissa`

```
x = 0:0.1:10;
y = f(x);
```

Exercício

Repita as instruções do exemplo anterior com `f` definida por

```
f = @(x)(x*sin(x))
```

Porque não funciona?

Exercício

Calcule o valor de $f(x) = \sin(x^2) + 1/x$, para todos os pontos da forma $x_i = ih,\ h = 0.1,\ i = 1, \ldots, 10$.

Funções

Uma função definida num ficheiro .m pode ser bastante mais complexa que uma função do tipo *inline*.

Exemplo

Implementação de minhaFuncao.m

```
function [y1,y2] = f(x1,x2)
    y1 = x1+x2;
    y2 = x1-x2;
return
```

Podemos chamar esta função da linha de comandos comando

```
 [soma,dif] = minhaFuncao(10,30);
```

As variáveis `x1`, `x2`, `y1` e `y2` podem ser escalares, vetores, matrizes, outras funções...

Produção de gráficos

Gráficos: comando fplot

O comando `fplot` permite traçar facilmente o gráfico de funções (inline) reais definidas num intervalo $I = [a, b]$

Exemplo

Trace o gráfico de

$$f(x) = \frac{\sin(x)}{x}$$

para $x \in [-\pi, \pi]$

```
f = @(x)(sin(x)./x);
fplot(f, [-pi,pi]);
```

Exercício

Trace o gráfico de $f(x) = \log(x)$ para $x \in [1, 10]$.

Gráficos: comando plot

Dados dois vetores x e y, plot(x,y) desenha os pontos de coordenadas (x(i),y(i)) no plano e une pontos consecutivos com um segmento de reta.

Exemplo

Desenhe um ≪quadrado incompleto≫

```
x = [-1 1 1 -1];
y = [-1 -1 1 1];
plot(x,y)
axis([ -2 2 -2 2])
```

ou um quadrado sem arestas e apenas com os pontos a vermelho (`o` é o símbolo usado no gráfico, `r` representa a cor)

```
plot(x,y,'or')
```

Gráficos: comando plot

Podemos usar o comando `plot` para traçar o gráfico de funções ou de conjuntos de dados discretos.

Exemplo

```
x = [1 2.2 3 4.3 5 5.2 6.5];
y = log(x);
plot(x,y,'-g',x,y,'or')
```

Exercício

Trace o gráfico do polinómio $p(x) = x^5 - 6x^4 + 7x^2 + 1$ no intervalo $I = [-1.5, 1.5]$ e identifique os zeros da sua derivada nesse intervalo.

Gráficos: figure e hold

A instrução `figure` permite criar novas janelas de gráficos.

Exemplo

```
x = -pi:pi/20:pi;
plot(x,cos(x),'-ro')
plot(x,sin(x),'-.b')
```

Neste exemplo, embora usemos dois comandos `plot`, a janela do gráfico apenas retem o último.

Uma alternativa é criar duas janelas distintas para os gráficos:

```
x = -pi:pi/20:pi;
figure(1); plot(x,cos(x),'-ro');
figure(2); plot(x,sin(x),'-.b');
```

Gráficos: hold

No entanto, se quisermos ter ambos os gráficos na mesma figura, podemos usar a instrução `hold`.

Exemplo

```
x = -pi:pi/20:pi;
plot(x,cos(x),'-ro')
hold on;
plot(x,sin(x),'-.b')
hold off;
```

A instrução `hold on` activa a colagem de gráficos na mesma figura. A instrução `hold off` desactiva essa função.

Gráficos: animações

Podem ser feitas animações usando a instrução `pause`.

Exemplo

```
pontos = [-1 1 1 -1 -1; -1 -1 1 1 -1];
angulo = 2*pi/100;
rotacao = [cos(angulo) sin(angulo);
           -sin(angulo) cos(angulo)];

for i=0:angulo:2*pi
    pontos = rotacao*pontos;
    plot(pontos(1,:),pontos(2,:));
    pause(0.01);
end
```

Alguns exercícios

Alguns exercícios

1. Dados $x = [4, 1, 6]$ e $y = [6, 2, 7]$, calcule a matriz definida por $a_{ij} = \frac{x_i}{2+x_i+y_j}$.
2. Calcule os primeiros 10 números da sucessão de Fibonacci.
3. Aproxime π usando uma série numérica.
4. Calcule um valor aproximado de $\int_0^1 \cos(x)\, dx$ usando a regra dos trapézios:

$$\int_a^b f(x)\, dx \approx \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right),$$

onde $h = (b-a)/n$ e $x_i = a + ih$.