Introducción a Matlab

Departament Física Aplicada, Universitat Politecnica de Catalunya

September 13, 2017

Contenido

1. ¿Qué es Matlab?

Abreviatura de **Mat**rix **Lab**oratory. Es un software matemático para cálculo cinetífico basado en matrices. Ofrece un entorno de desarrollo integrado (IDE) muy versátil con un lenguaje de programación propio. Permite:

- Realizar cálculos aritméticos (como una calculadora)
- Manipular fácilmente matrices
- Realizar gráficas en 2D y 3D
- Programar en un **lenguaje interpretado** (no compilado)

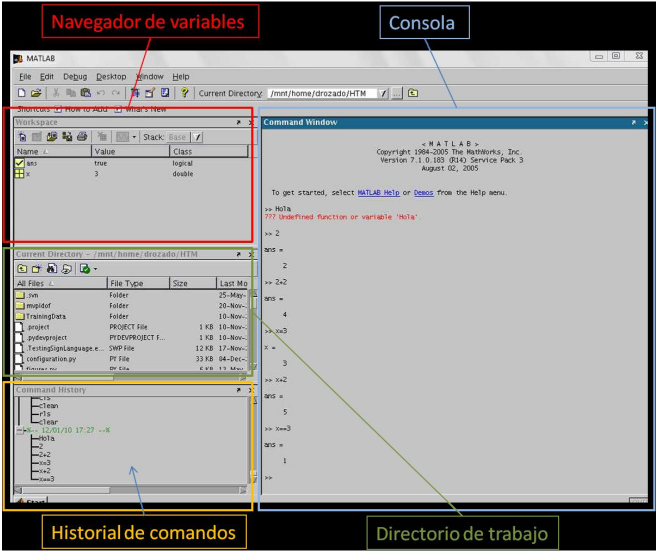
Escribiremos **scripts**: ficheros con una serie de órdenes que se pasan a un **intérprete** para que las ejecute.

Intérprete: capa de software que transforma estas órdenes en instrucciones en lenguaje máquina que se envían al hardware para ser ejecutadas.

El script se ejecuta **secuencialmente**: el intérprete lee línea por línea el código y no ejecuta la siguiente orden hasta que no haya terminado la anterior.

2. Entorno Matlab

Inicio sesión: `matlab (-nodesktop -nojvm)`



2. Entorno Matlab

Dos métodos de trabajo en Matlab:

- **Trabajo interactivo**
  El usuario escribe una operación en la consola (command window) y el progrma la ejecuta
- **Trabajo programado** (archivos.m)
  El usuario genera uno o varios ficheros con conjuntos de instrucciones matlab que se pueden ejecutar repetidas veces

Podemos hacer:

- **Cálculos directamente**. Operaciones: + - * / ^
- **Asignación de variables**. Estas son usadas más adelante.
- Uso de **funciones** definidas por matlab o por el usuario.

Salir sesión: `quit/exit`

2. Entorno Matlab

Ej.1

```
>> 2+3
>> x=2+3
>> y=5+4;
>> y
```

Si el resultado se asigna a una variable x, queda almacenado en ella.

Con ; no se muestra la respuesta, pero el cálculo y la asignación del resultado a la variable se han hecho.

Ej.2 Calcular πx^4 para $x = 0.2$. Constantes: `i`, `1i`, `pi`

```
>> x=0.2;
>> pi*x^4
>> pi*x^4 % Hola
>> y='posicion';
>> who; whos
>> clear; clear all
```

% Lo escrito a la derecha el intérprete lo ignora. Útil en programas largos poner comentarios del algoritmo.

'' Asignación de variables caracter

Enseña las variables usadas

Elimina variables de la memoria

Ej.3 Hacer script para multiplicar dos números guardándolo en un archivo *multiplicar.m*. Ejecutarlo en línea de comandos (*sin .m!*)

Funciones matemáticas: *sqrt, log, log10, exp, sin, cos, tan, asin, acos, atan, abs*

Otras funciones: *dir, cd, disp('una parrafada'), num2str(x)*

Ayuda: *Help instruccion*

2. Entorno Matlab

Más funciones intrínsecas y matemáticas:

who	Muestra las variables que hay en memoria
whos	Muestra las variables que hay en memoria indicando su tamaño y tipos de contenido
what	Lista los ficheros ".m" del directorio de trabajo
dir	Muestra el contenido del directorio de trabajo
!	Ejecuta un comando del sistema operativo

Trigonométricas

sin	Sine
sinh	Hyperbolic sine
asin	Inverse sine.
asinh	Inverse hyperbolic sine.
cos	Cosine.
cosh	Hyperbolic cosine
acos	Inverse cosine
acosh	Inverse hyperbolic cosine
tan	Tangent
tanh	Hyperbolic tangen
atan	Inverse tangent
atan2	Four quadrant inverse tangent
atanh	Inverse hyperbolic tangent
sec	Secant
sech	Hyperbolic secant
asec	Inverse secant
asech	Inverse hyperbolic secant
csc	Cosecant
csch	Hyperbolic cosecant
acsc	Inverse cosecant
acsch	Inverse hyperbolic cosecant
cot	Cotangent
coth	Hyperbolic cotangent
acot	Inverse cotangent
acoth	Inverse hyperbolic cotangent

Exponenciales

exp	Exponential
log	Natural logarithm
log10	Common logarithm
sqrt	Square root

Complejas

abs	Absolute value
angle	Phase angle
conj	Complex conjugate
imag	Complex imaginary part
real	Complex real part

Numéricas

fix	Round towards zero
floor	Round towards minus infinity
ceil	Round towards plus infinity
round	Round towards nearest integer
rem	Remainder after division
sign	Signum function

3. Representación en coma flotante de $\mathbb{R}$

- Recursos limitados de las máquinas $\Rightarrow$ sólo se puede representar subconjunto $\mathbb{F}$ de dim. finita de R: **números de coma flotante**. Cq. $x \in \mathbb{R}$ es truncado y da lugar a un número $fl(x)$ no necesariamente igual a x.
- Matlab trabaja con 16 cifras: **aritmética de 16 dígitos**. Los números reales son redondeados por el ordenador.

  Pero si escribimos $x = 1/7 = 0.\overline{142857}$ obtenemos 0.1429!

  Número decimales en pantalla depende del **formato de salida** seleccionado, pero no afecta a la **precisión** de los cálculos.

  Formatos de salida: **punto fijo (g)** y **coma flotante (e)**.

`format long`	0.142857142857143
`format short e`	1.4286e-01
`format long e`	1.428571428571428e-01
`format short g`	0.14286
`format long g`	0.142857142857143
`format rat`	$1/7$

3. Representación en coma flotante de $\mathbb{R}$

Un computador almacena número de $\mathbb{R}$ de la forma:

$$\boxed{x = (-1)^s \cdot (0.a_1 a_2 \dots a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t}, \quad a_1 \neq 0}$$

- $s = 0$ ó 1
- β (entero ≥ 2) $\Rightarrow$ **Base** adoptada por computador.
- m (entero) $\Rightarrow$ **Mantisa**. Su longitud t es el máximo de cifras a_i que se almacenan ($0 \leq a_i \leq \beta - 1$).
- $t \Rightarrow$ **Número de cifras significativas**.
- e (entero) $\Rightarrow$ **Exponente**. $e_{min} = L < 0 \qquad e_{max} = U > 0$

$\mathbb{F} = \mathbb{F}(\beta, t, L, U)$. **Matlab**: $\mathbb{F} = \mathbb{F}(2, 53, -1021, 1024)$

53 cifras significativas en base 2 corresponden a 15 cifras significativas en base 10

- $x_{min} = \beta^{L-1} \rightarrow$ `realmin` $\rightarrow$
  $x_{min} = 2.225073858507201 \cdot 10^{-308} \Rightarrow$ `underflow`
- $x_{max} = \beta^U (1 - \beta^{-t}) \rightarrow$ `realmax` $\rightarrow$
  $x_{max} = 1.797693134862316 \cdot 10^{+308} \Rightarrow$ `overflow, Inf`

3. Representación en coma flotante de $\mathbb{R}$

- $0/0, \infty/\infty \Rightarrow$ `Nan` (not a number)
- $\varepsilon_M = \beta^{1-t} \Rightarrow$ **Resolución de la máquina**: Distancia entre 1 y el n^0 > 1 en coma flotante más cercano a 1.
  `eps` $\rightarrow \varepsilon_M = 2^{-52} = 2.22 \cdot 10^{-16}$
- **Error de redondeo** generado al sustituir un $x \in \mathbb{R}$ por su representante en $\mathbb{F}$:

$$\boxed{\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\epsilon_M}$$

Ej.1

```
>> a=1; b=1;
>> while a+b~=a;
>> b=b/2;
>> end
```

Si operáramos con números $\mathbb{R}$ este programa nunca acabaría. En Matlab termina después de un número finito de pasos: $\exists b \neq 0$ tq $a + b = a$!!

`ans:` `b=1.1102e-16=`$\varepsilon_M/2$

Ej.2 Cuidado con pérdida/cancelación de cifras significativas. Peligro cuando operamos cerca de x_{min}, x_{max}

```
>> x=1.e-15;
>> ((1+x)-1)/x
```

`ans=1.1102` !!

4. Estructuras de datos

VECTORES

- **Vector fila**. `>>x=[2 3 4 5]` `>>x=[2,3,4,5]` `>>x=[2:1:5]` `>>x=linspace(2,5,4)`
- **Vector columna**. `>>y=[6;7;8;9]`
- **Trasponer**. `>>xtras=x.'` `>>xtras=x'`
- `sum(x)`, `mean(x)`, `diff(x)`,`prod(x)`
- Acceso componentes vector. **Indexing**. `>>x(2)` ans=3
- Extraer subvectores.
  `>>x(2:4)` ans=3 4 5
  `>>x(:)` ans=2 3 4 5
  `>>x(3:end)` ans=4 5
- Ordenación. **Sorting**.
  `>>y=[5 3 2 4 8 7]`
  `>>max(y)` ans=8 `>>min(y)` ans=2
  `>>[sy ii]=sort(y)` ii $\rightarrow$ Vector índices con posiciones de n^{os} ordenados
  `>>ii` ans=3 2 4 1 6 5
  `>>y(ii)` ans=2 3 4 5 7 8
  `>>jj=[3 1 5]; y(jj)` ans=2 5 8
  `>>fliplr(y)` ans=7 8 4 2 3 5
  `>>flipud(y.')` ans=7;8;4;2;3;5
- **Extracción componentes condicionadas**. `j=find(c(x))`
  `>>x=[0.82 0.69 0.31 0.45]`
  Operadores: `==` `>` `<` `~=`
  `>>j=find(abs(x)<0.5)` ans=3 4
  `>>x(j)` ans=0.31 0.45

4. Estructuras de datos

- **Vectorización de funciones**. `f(x)=[f(x1) f(x2) f(x3)...]`
  `>>x=[2 3 4 5]; x.^2` ans=4 9 16 35
  `>>sin(x)` ans= 0.9093 0.1411 -0.7568 -0.9589
- **Operaciones escalar-vector**. `>>a=2; x+a` ans=4 5 6 7
- **Operaciones vector-vector**. `x=[x1 x2...], y = [y1 y2...]`
  Suma/diferencia: `x+y=[x1+y1 x2+y2...]`
  Producto/cociente componente a componente: `x.*y=y.*x=[x1*y1 x2*y2...]; x./y=[x1/y1 x2/y2...]`
  Producto escalar: `x*y.'=sum(x.*y)=sum(y.*x)=dot(x,y)`
  Módulo: `norm(x,2)=norm(x)=sqrt(x*x')`
  Concatenación, actualización como variable:
  `>>v=[1 3 7]; u=[8 pi]`
  `>>v=[v u]` ans=1 3 7 8 pi

4. Estructuras de datos

MATRICES

- **Matriz**. `>>A=[1 2 3;4 5 6]`
- **Trasponer**.
  `>>B=A.'` ans=1 4;2 5;3 6
- **Inicialización. Matrices especiales**.
  `>>A=zeros(N)=zeros(N,N)`
  `>>B=zeros(M,N)`
  M filas, N columnas
  `>>eye(N)` Matriz identidad
  `>>ones(M,N)` Matriz MxN de 1
  `>>rand(M,N)` Matriz MxN aleatoria
- **Extracción filas**.
  `>>u=A(1,:)` ans=1 2 3
  `>>size(u)` ans=1 3
  `>>v=A(2,:)` ans=4 5 6
  `>>mat=[u;v;0 0 1]`
- **Extracción columnas**.
  `>>w=A(:,2)` ans=2;5
- Acceder a la matriz como si fuera un vector columna. **Matlab almacena por columnas**
  `>>mat(2:7)` ans=4 0 2 5 0 3
- **Acceder a submatrices**.
  `>>mat(2:3,[1 3])` ans=4 6;0 1
  `>>A=[1 2 3 4;5 6 7 8; 9 10 11 12;13 14 15 16]`
  `>>idx=[2;4]; A(idx,2)` ans=6;14
  `>>A(2,idx)` ans=6 8
  `>>A(2,end)` ans=8
- **Concatenación, actualización como variable**.
  `>>b=[1;0;-1;0]; A=[A b]; A=[A;b.']`