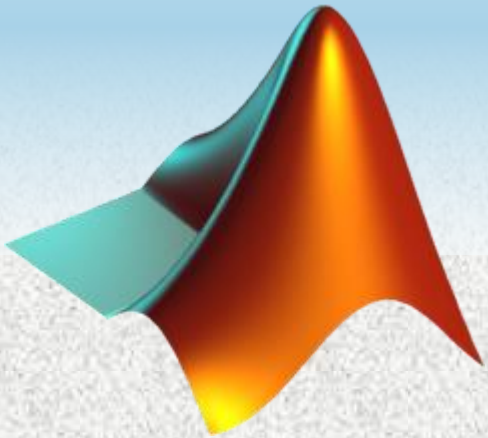
UNIVERSIDAD NACIONAL DE INGENIERÍA

MATLAB

R2017a



Maria Pimentel Herrera

uni.kernel@gmail.com

Cálculos básicos

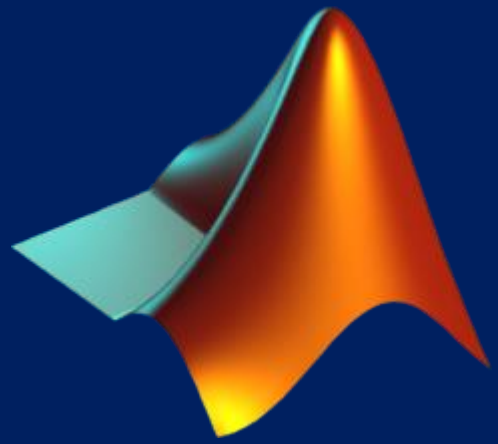
Operaciones aritméticas con MATLAB

- ^ Potenciación
- *, / Multiplicación y división
- +, - Suma y resta

Orden de prioridad de las operaciones: las expresiones se evalúan de izquierda a derecha; la operación potencia tiene el orden de prioridad más alto, seguida por multiplicación y división que tienen ambas igual prioridad y seguidas, finalmente, por suma y resta con igual prioridad.

Operaciones aritméticas con MATLAB

Se pueden emplear paréntesis para alterar esta ordenación, en cuyo caso la evaluación se inicia dentro del paréntesis más interno y se prosigue hacia afuera.



Aplicación

$3^2 - 5 - \frac{6}{3} \cdot 2$ >> 3^2-5-6/3*2

$3^2 - 5 - \frac{6}{3 \cdot 2}$ >> 3^2-5-6/(3*2)

$4 \cdot 3^2 + 1$ >> 4*3^2+1

$(4 \cdot 3)^2 + 1$ >> (4*3)^2+1

Variables

- La introducción de variables ofrece nuevas posibilidades. Supongamos que queremos calcular el área de un triángulo de base 21.3 m y altura 12.6 m.

```
>> 1/2*21.3*12.6
```

- Otra posibilidad es utilizar las variables base, altura, area y realizar los cálculos de la siguiente forma:

```
>> base=21.3
```

(Asigna a la variable base el valor 21.3)

```
>> altura=12.6
```

(Asigna a la variable altura el valor 12.6)

```
>> area=1/2*base*altura
```

(Asigna a la variable area el valor correspondiente)

- Las variables base, altura, area permanecen en el espacio de trabajo con los últimos valores asignados y pueden ser llamadas o modificadas cuantas veces se desee.
- **Observación:** La modificación de la variable altura no cambia el valor de la variable area si ésta no se vuelve a calcular.

```
>> altura=9.7
>> area
>> area=1/2*base*altura
```

Reglas para nombrar variables

- Las letras mayúsculas y minúsculas son distintas a efectos de nombrar variables. Por ejemplo, son diferentes las variables base, Base, BASE.
- El nombre de una variable puede tener hasta 31 caracteres; si hubiese más serían ignorados.
- El nombre de una variable debe comenzar obligatoriamente por una letra. Puede contener letras, números y el guión de subrayado (_); no se permiten espacios en blanco.

Reglas para nombrar variables

- No es conveniente nombrar variables mediante expresiones que tengan un significado específico en MATLAB: Por ejemplo, no es aconsejable utilizar log como nombre de variable ya que ésta es la designación de la función logarítmica en MATLAB.
- Como regla general es aconsejable que el nombre de una variable sea indicativo de su contenido.

Algunas variables predefinidas en MATLAB

- Hay algunas variables que, por defecto, tienen un valor asignado. Podemos citar:
- ans Es la variable que, por defecto, contiene los resultados.
- pi Contiene el valor del número real π.
- eps Es el número positivo más pequeño que sumado a 1 genera un número mayor que 1 en el ordenador.

Algunas variables predefinidas en MATLAB

- Inf o inf Representa el valor infinito. Se obtiene, por ejemplo, en caso de overflow o división por cero.
- NaN o nan (Not a Number) Representa una expresión indeterminada, por ejemplo: 0/0 .
- i o j Representa la unidad imaginaria i = j = $\sqrt{-1}$.

Signos de puntuación

- En una misma línea pueden definirse varias variables separadas por coma (,) o por punto y coma (;). La diferencia consiste en que el punto y coma inhibe la visualización en pantalla.

```
>> a=2,b=3;c=5
>> d=a*b+c;
>> d
```

Signos de puntuación

- Utilizando tres puntos se puede cambiar de línea sin ejecutar las órdenes escritas.

```
>> x=25,...
y=x/5
```

- Esto puede resultar útil cuando una expresión no cabe en una sola línea de la ventana de comandos.

Comentarios

- MATLAB ignora lo que se encuentra a la derecha del símbolo %. Esto permite introducir comentarios.

```
>> a=2;b=3;          %a es la base, b es la altura

>>area=a*b/2 %area del triangulo
```

Movimientos del cursor

↑ “Flecha arriba” (Recupera la línea anterior)

↓ “Flecha abajo” (Recupera la línea siguiente)

← “Flecha izquierda” (Mueve el cursor hacia la izquierda un carácter)

→ “Flecha derecha” (Mueve el cursor hacia la derecha un carácter)

- *Se recomienda hacer uso de esta utilidad siempre que sea posible ya que permite ahorrar mucho tiempo.*

Aplicación

- Calcular la suma a=1+2+3+4+5+6+7+8+9+10. Utilizando los movimientos del cursor, calcular b=1-2+3-4+5-6+7-8+9-10.

- >> a=1+2+3+4+5+6+7+8+9+10
- >> ↑ “pulse Flecha arriba” (editando la línea)
- >> b=1-2+3-4+5-6+7-8+9-10

Funciones predefinidas

exp	sin	asin	sinh	asinh
log	cos	acos	cosh	acosh
log10	tan	atan	tanh	atanh
log2	cot	acot	coth	acoth
sqrt	sec	asec	sech	asech
	csc	acsc	csch	acsch

Notación que se usa habitualmente en matemáticas

Matemática	Matlab	Ejemplo
e^x	exp(x)	exp(1)
$\ln(x)$	log(x)	log(3)
$\log_{10}(x)$	log10(x)	log10(3)
$\log_2(x)$	log2(x)	log2(3)
$\sqrt{x}$	sqrt(x)	sqrt(64)
$\text{sen}(x)$	sin(x)	sin(pi/3)
…	…	…

Funciones predefinidas en MATLAB de uso frecuente

Matlab	Definición
abs(x)	Valor absoluto
fix(x)	Redondea hacia cero
floor(x)	Redondea hacia menos infinito
ceil(x)	Redondea hacia más infinito
round(x)	Redondea hacia el entero más próximo
rem(m,n)	Resto al dividir m entre n
rand	Número aleatorio entre 0 y 1

Aplicación

```
>> sqrt(16)

>> 16^(1/2)

>> (-27)^(1/3)

>> sin(pi/3)

>> x=cos(pi);y=log(25);z=exp(y);
```

Aplicación

```
>> t=25*x+z

>> abs(-3)

>> fix(5.7),floor(5.7),ceil(5.7),round(5.7)

>> fix(-5.7),floor(-5.7),ceil(-5.7),round(-5.7)

>> rem(20,3)
```

Vectores y matrices

Vectores

- Los vectores se introducen escribiendo, entre corchetes, cada una de sus coordenadas separadas por un espacio en blanco o una coma. Por ejemplo, para introducir el vector (1, 3, 5, -7, 0.33, π, e) se escribe

>> [1 3 5 -7 0.33 pi exp(1)]

- o bien,

>> [1, 3, 5, -7, 0.33, pi, exp(1)]

- Un vector puede asignarse a una variable, por ejemplo:

>> v = [1 3 5 -7 0.33 pi exp(1)]

x=[x1:h:xn]

- En algunos casos se puede generar vectores sin necesidad de introducir explícitamente sus coordenadas:

>> [1:0.1:2]

x=[x1:xn]

- Hace lo mismo que x=[x1:1:xn].

>> [1:7]

>>[1:1:7]

linspace(a,b,n)

- Genera un vector de n coordenadas espaciadas uniformemente, comenzando por a y terminando por b.

- El vector x = (-3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8) se puede generar de las siguientes formas:

```
>> x=[-3 -2 -1 0 1 2 3 4 5 6 7 8]
```

linspace(a,b,n)

```
>> x=[-3:1:8]
>> x=[-3:8]
>> x=(-3:8)
>> x=-3:8
>> x=-3:8.7
>> x=linspace(-3,8,12)
```

- Apreciar la diferencia entre los dos modos descritos para generar vectores, observemos la diferencia entre los vectores x e y:

```
>> x=0:0.1:1.07

>> y=linspace(0,1.07,11)
```

Acceso a las coordenadas de un vector

- Una vez definido un vector se puede acceder a sus coordenadas para conocer su valor, utilizarlo o modificarlo.
- v(i)

Devuelve la coordenada i-ésima del vector v.

- v(i1:h:i2)

Devuelve las coordenadas de v cuyos índices van desde i1 hasta i2, con incremento h.

Acceso a las coordenadas de un vector

- v(i1:i2)

Hace lo mismo que v(i1:1:i2).

- v([i,j,k])

Devuelve las coordenadas i, j y k de v.

- v(end)

Devuelve la última coordenada de v.

Aplicación

```
>> v=[-1 3 2 4 -8 3]

>> v(2)
```

(Muestra la segunda coordenada de v)

```
>> v(end)
```

(Muestra la última coordenada de v)

```
>> v(2)+5
```

(Se utiliza la segunda coordenada de v en una operación)

Aplicación

>> v(2)=6

(Asigna el valor 6 a la segunda coordenada del vector v)

>> v(2:4)

(Muestra las coordenadas del vector v desde la segunda a la cuarta)

>> w=v(2:4)

(Define el vector w con las coordenadas segunda, tercera y cuarta de v)

>> u=v([1,5,6])

(Define u con las coordenadas primera, quinta y sexta de v)

Matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & 8 \\ 7 & 2 & 1 \end{pmatrix}$$

- se escribe

>> A=[1 2 3; -1 4 8; 7 2 1];

- Para visualizar la matriz A en pantalla escribimos:

>> A

Definición de matrices por cajas

- Dadas dos matrices A y B con igual número de filas se puede definir una matriz C formada por todas las columnas de A y B:

```
>> A=[1 2 3; -1 4 8]

>> B=[5 5; 6 6]

>> C=[A B]
```

Definición de matrices por cajas

- Dadas dos matrices A y B con igual número de columnas se puede definir una matriz C formada por todas las filas de A y B:

```
>> A=[1 2 3; -1 4 8]

>> B=[5 5 5; 6 6 6; 7 7 7]

>> C=[A;B]
```

Matrices especiales

- Para introducir algunas matrices de uso frecuente (matrices de ceros, matrices de unos, matrices unidad, …) MATLAB dispone de órdenes específicas:
- ones(n)

Genera una matriz cuadrada n x n formada por unos.

- ones(m,n)

Genera una matriz m x n formada por unos.

```
>> A=ones(5)
>> B=ones(3,4)
```

Matrices especiales

- zeros(n)

Genera una matriz cuadrada n x n formada por ceros.

- zeros(m,n)

Genera una matriz m x n formada por ceros.

```
>> C=zeros(5)

>> D=zeros(3,4)
```

Matrices especiales

- eye(n)

Genera una matriz unidad n x n.

- eye(m,n)

Genera una matriz m x n con unos en la diagonal principal y ceros en el resto.

>> E=eye(4)

>> F=eye(3,4), G=eye(4,3)

Acceso a los elementos de una matriz

- Una vez definida una matriz se puede acceder a sus elementos para conocer sus valores, utilizarlos o modificarlos.
- A(i,j)

  Devuelve el elemento (i, j) de la matriz A.

- A(i1:h:i2,j1:k:j2)

  Devuelve la submatriz de A formada por las filas de la i1 a la i2 con incremento h y las columnas j1 a la j2 con incremento k.

Acceso a los elementos de una matriz

- A(i1:i2,j1:j2)

  Devuelve la submatriz de A formada por las filas de la i1 a la i2 y las columnas j1 a la j2.

- A(i1:h:i2,:)

  Devuelve la submatriz de A formada por las filas de la i1 a la i2 con incremento h.

- A(i1:i2,j)

  Devuelve el elemento j de las filas comprendidas entre la i1 y la i2.

Acceso a los elementos de una matriz

- A(i,j1:j2)

Devuelve el elemento i de las columnas comprendidas entre la j1 y la j2.

- A(:,j)

Devuelve el elemento j de todas las filas, o sea, la columna j.

- A(i,:)

Devuelve el elemento i de todas las columnas, o sea, la fila i.

- A(i,[j k])

Devuelve la submatriz (aij, aik)

Aplicación

```
>> A=[1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15;
16 17 18 19 20]
>> A(2,5)
>> A(1:3,2:5)     % filas 1 a 3, columnas 2 a 5
>> A(1:2:4,1:3:5) % filas 1, 3, columnas 1, 4
>> A(:,1:2:5) % todas las filas, columnas 1, 3 y 5
```

Aplicación

```
>> A(2:4,3)      % filas 2 a 4, columna 3

>> A(2,1:4)   % filas 1 a 3, columnas 2 a 5

>> A(:,3)

>> B=[A(1:3,:) A(2:4,:)]

>> C=[A(1:3,1:3); A(:,[1 3 5])]

>> D=A([2 4], [3 5])
```

Información sobre las dimensiones

- Las órdenes length y size ofrecen información sobre el tamaño de vectores y matrices. Conviene recordar que en la versión 5.0 de MATLAB el tamaño máximo de una matriz es de 16384 (= 128 x 128) elementos.

- size(A)

Genera un vector con las dimensiones de la matriz A.

- size(A,1)

Devuelve el número de filas de la matriz A.

Información sobre las dimensiones

- size(A,2)

Devuelve el número de columnas de la matriz A.

- [m n]=size(A)

Asigna a m el número de filas de A y a n el de columnas.

- length(v)

Devuelve el número de coordenadas del vector v.

- length(A)

Devuelve el valor de max(size(A)).

Aplicación

```
>> A=ones(2,5), v=[1 2 3 4 5]
>> size(A)
>> size(A,1)
>> size(A,2)
>> [m n]=size(A)
>> length(v)
>> length(A)
```

Operaciones con vectores y matrices

Un vector de n coordenadas se puede considerar como una matriz 1 x n. Por esta razón vamos a describir simultáneamente las operaciones con vectores y matrices.

Operaciones algebraicas

- Si A y B son matrices con dimensiones adecuadas y λ un escalar, las operaciones algebraicas en MATLAB se efectúan con las siguientes órdenes:
- A(m,n)+B(m,n)

Suma A y B.

- A(m,n)-B(m,n)

Resta A y B.

- A(m,n)*B(n,k)

Multiplica A por B.

*Estas matrices no son conforme para el producto, las columnas de A deben ser iguales a las filas de B A(m,n)*B(m,n) error*

Operaciones algebraicas

- λ*A

Multiplica por λ todos los elementos de A.

- A^n

Eleva la matriz A al número entero n.

- A.'

Genera la traspuesta de A.

- A'

Genera la conjugada traspuesta de A.

- *Obsérvese que si A es una matriz real, A' y A.' coinciden.*

Operaciones algebraicas

- inv(A)

Calcula la inversa de A.

- *Si u y v son dos vectores con el mismo número de coordenadas*
- dot(u,v)

Calcula el producto escalar.

- cross(u,v)

Calcula el producto vectorial; es necesario que u y v tengan 3 coordenadas.

Aplicación

```
>> A=[1 2; 3 4]

>> B=[-1 4; 1 -2]

>> C=[1 2 3; 4 5 6]

>> A+B

>> 3*A

>> A*C
```

Aplicación

```
>> C’

>> inv(B), B^(-1)

>> C’*(A-2*B)

>> u=[-1 2 3]; v=[5 -7 9];

>> dot(u,v)

>> cross(u,v)
```

Manual Matlab

- Página 38