

Begoña Cantó Colomina
Macarena Boix García

Pràctiques d'Àlgebra amb Matlab 2013b

EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Revisió lingüística: Servei de Promoció i Normalització Lingüística de la UPV
La publicació d'aquest llibre ha rebut una ajuda del Servei de Promoció i Normalització Lingüística de la UPV de la Universitat Politècnica de València per a l'elaboració de publicacions docents en valencià a l'Editorial UPV

Els continguts d'aquesta publicació han sigut aprovats pel Comitè Editorial del Departament.

Colecció Acadèmica

Per a referenciar esta publicació utilitze la següent cita: CANTÓ COLOMINA, B.; BOIX GARCÍA, M. (2015) *Pràctiques d'Àlgebra amb Matlab 2013b*. València : Universitat Politècnica de València

© Begoña Cantó Colomina
Macarena Boix García

© 2015, de la present edició: Editorial Universitat Politècnica de València
distribució: Tel. 96 387 70 12 / www.lalibreria.upv.es / Ref : 0392_08_01_01

Imprimeix: Byprint Percom, sl

ISBN: 978-84-9048-368-8
Imprés sota demanda

Imprés en paper Creator Silk



Queda prohibida la reproducció, distribució, comercialització, transformació i, en general, qualsevol altra forma d'explotació, per qualsevol procediment, de tot o part dels continguts d'aquesta obra sense l'autorització expressa i per escrit dels autors.

Imprés a Espanya

Pròleg

L'objecte d'aquesta publicació és servir d'ajuda en matèria de matemàtiques als estudiants de primer curs d'escoles tècniques, especialment: en enginyeria informàtica, en administració i direcció d'empreses, en enginyeria elèctrica, en enginyeria química i en enginyeria mecànica.

Volem expressar el nostre agraïment personal als doctors David R. Hill i David E. Zitarelli, que ens han donat permís per incloure els arxius `gschmidt.m` i `reduce.m`, així com suggeriments sobre altres arxius. Ens hem permès una petita modificació en l'arxiu `reduce.m` que ha donat lloc a `reducesym.m` per a utilitzar-lo no sols en matrius numèriques, sinó també en matrius simbòliques.

Donem les gràcies igualment a The MathWorks, pel permís donat a l'hora d'utilitzar les imatges de Matlab, així com a Salvador Juan Torres i Juan Manuel Rodríguez per les seues gestions.

També volem agrair als nostres companys, especialment a Jesús Rodríguez López, les notes i els aclariments als dubtes que ens han sorgit.

Finalment, volem mostrar també el nostre agraïment al Servei de Normalització Lingüística i al Vicerectorat de Normalització i Promoció Lingüística l'oportunitat que ens han donat en finançar aquest projecte.

Les autores.

Índex

Pràctica 1. INTRODUCCIÓ A MATLAB	1
1.1. Consideracions.....	1
1.2. Introducció a Matlab 2013b.....	2
1.3. Operacions amb escalars	7
1.4. Generació de matrius	8
1.4.1. Comandaments de generació de matrius.....	11
1.4.2. Qüestions sobre la generació de matrius.....	16
1.5. Comandaments de visualització	19
1.6. Matrius definides per blocs.....	21
1.6.1. Funció repmat.....	22
1.6.2. Funció cat.....	23
1.7. Funcions especials	24
1.7.1. Funció clock.....	24
1.7.2. Funcions tic/toc.....	25

1.7.3. Funció display	25
1.7.4. Funció sprintf	26
1.8. Problemes	26
Pràctica 2. OPERACIONS AMB MATRIUS	29
2.1. Suma i diferència de matrius	29
2.2. Producte de matrius	31
2.2.1. Producte d'escalar per matriu	32
2.3. Potències de matrius	33
2.4. Transposició de matrius	34
2.5. Inversa d'una matriu	34
2.6. Altres funcions	38
2.6.1. La funció reshape	38
2.6.2. Les funcions min, max	40
2.6.3. La funció mean	41
2.6.4. La funció sum	41
2.7. Rang i factorització L U d'una matriu	41
2.7.1. Invariància del rang	42
2.7.2. Descomposició LU d'una matriu	44
2.8. Matrius disperses	49
2.8.1. Funció sparse	50
2.8.2. Funció spy	52
2.8.3. Funció gplot	53
2.8.4. Funció spdiags	55
2.9. Problemes	57

Pràctica 3. DETERMINANTS	59
3.1. Introducció.....	59
3.2. Polinomi d'interpolació	65
3.2.1. Funció interp1	68
3.3. Problemes	71
Pràctica 4. SISTEMES D'EQUACIONS LINEALS	73
4.1. Introducció.....	73
4.2. Mètode de Gauss	73
4.3. Ordres per resoldre S.E.L	76
4.3.1. Ordre \ inv.....	76
4.3.2. Ordre null.....	77
4.3.3. Ordre rref.....	79
4.4. Altres ordres per resoldre S.E.L	80
4.4.1. Ordre solve.....	81
4.4.2. Ordre linsolve	83
4.4.3. Ordre reduce	86
4.5. Aplicació: resoldre S.E.L. amb la descomposició LU.....	93
4.6. Càlcul simbòlic	96
4.7. Aplicacions: espai vectorial.....	106
4.7.1. Combinació lineal.....	106
4.7.2. Embolcall lineal	108
4.7.3. Independència lineal	108
4.7.4. Base.....	110
4.7.5. Coordenades d'un vector.....	112
4.8. Operacions de subespais vectorials	113
4.9. Problemes	116

Pràctica 5. ESPAI EUCLIDIÀ	119
5.1. Introducció.....	119
5.2. Producte escalar i norma.....	119
5.3. Conjunts ortonormals.....	121
5.3.1. Algorisme de Gram-Schmidt.....	124
5.3.2. Factorització QR.....	126
5.4. Subespai ortogonal i projecció ortogonal.....	131
5.4.1. Subespai ortogonal.....	131
5.4.2. Projecció ortogonal.....	133
5.5. Aplicació: mínims quadrats.....	135
5.5.1. Regressió.....	136
5.6. Problemes.....	140
Pràctica 6. DIAGONALITZACIÓ DE Matrius	143
6.1. Introducció.....	143
6.2. Valors propis i vectors propis.....	143
6.3. Diagonalització de matrius.....	147
6.4. Diagonalització de matrius simètriques.....	150
6.5. Aplicació a les cadenes de Markov.....	156
6.6. Problemes.....	158
Arxius.....	161
Bibliografia.....	179

Pràctica 1

Introducció a *MATLAB*

1.1. Consideracions.

En aquest treball estudiarem l'àlgebra amb l'ajuda del paquet matemàtic *MATLAB*. Aquest programa s'ha implementat sota els sistemes operatius més estesos (DOS, UNIX...); i en aquest treball s'utilitzarà la versió Matlab 2013b per a Windows.

En cada un dels capítols s'inclourà un breu comentari teòric (en cursiva) sobre la matèria que s'està tractant, així com les ordres i les funcions del programa necessàries per a plantejar els problemes i per a fer els càlculs que es proposen en cada moment. També apareixen exemples que serveixen per a familiaritzar l'usuari tant en l'ús de les ordres i funcions del programa, com dels mètodes i tècniques que s'estudien. A més a més, al final de cada capítol es proposen una sèrie de problemes amb la finalitat que l'usuari consolide els conceptes que s'han estudiat. Cal assenyalar que les observacions que es consideren importants es remarcaran amb un requadre. L'últim capítol d'aquest llibre recull tots els arxius que hem utilitzat i que no estan predefinits en el programa.

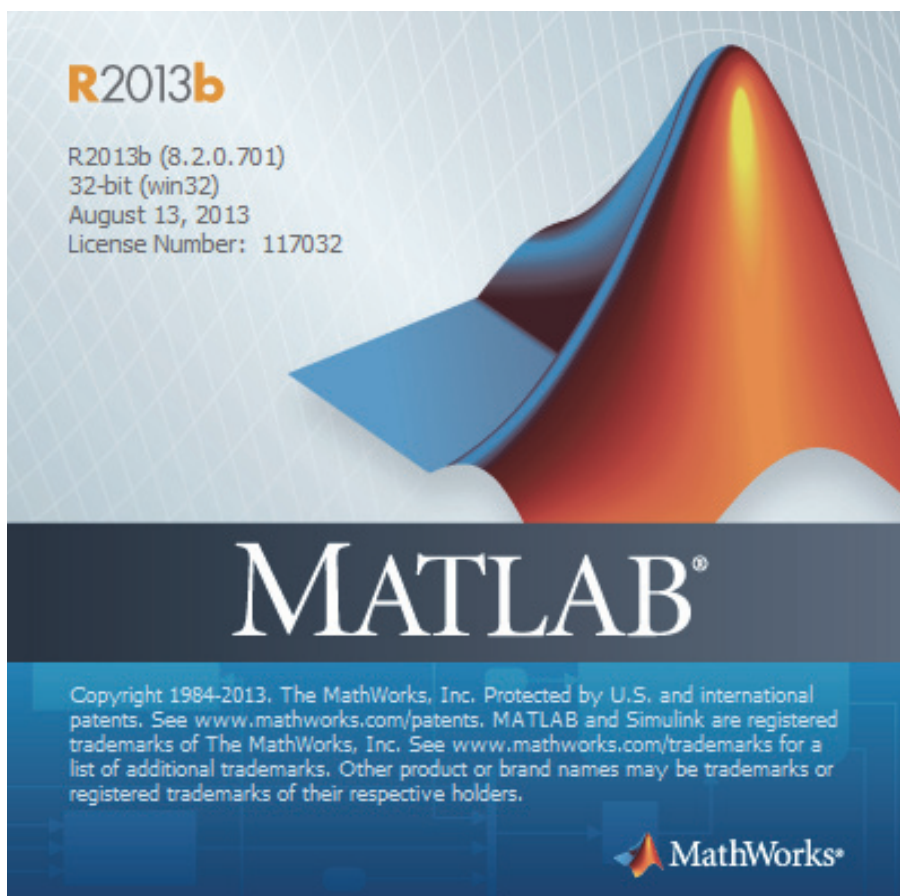
Aquest primer capítol d'introducció descriu el programa tant des del punt de vista conceptual com de funcionament bàsic. A més a més, es descriu quina és la metodologia a utilitzar per l'entrada de dades Matlab, com s'opera amb els escalars i com es generen les matrius. Finalment hi ha un resum sobre les ordres de visualització d'aquestes matrius.

1.2. Introducció a MATLAB 2013b.

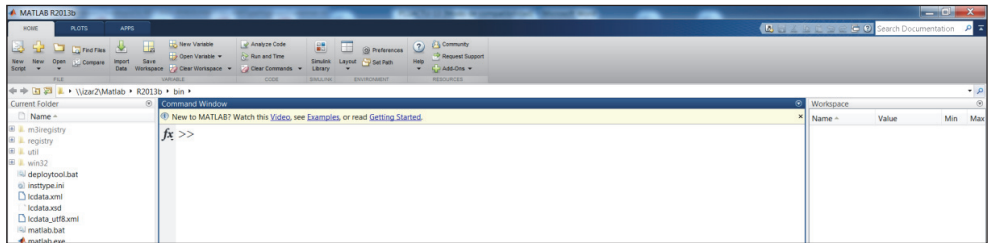
El paquet matemàtic Matlab integra la computació matemàtica amb funcions de visualització i un llenguatge de gran eficàcia amb la finalitat d'oferir un entorn flexible per a la realització de computació tècnica. L'arquitectura oberta facilita l'ús de Matlab per a explorar dades, crear algorismes i eines a mida que permeten obtenir d'una manera ràpida la informació. Matlab abasta una àmplia gamma de tasques de càlcul tècnic i científic, des de l'adquisició i anàlisi de dades fins al desenvolupament d'aplicacions.

En el cas que ens ocupa tractarem la part més bàsica de Matlab, ja que la missió del programa és fer més entenedors i manejables els conceptes teòrics i pràctics adquirits durant les lliçons magistrals.

Accedim al programa fins que aparega la pantalla següent:

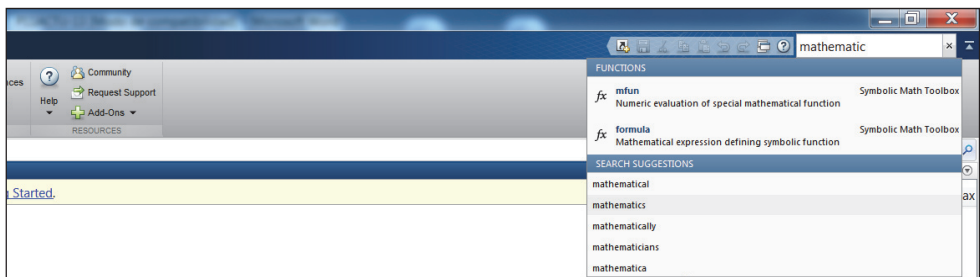


Imatge 1.1. Pantalla principal del programa

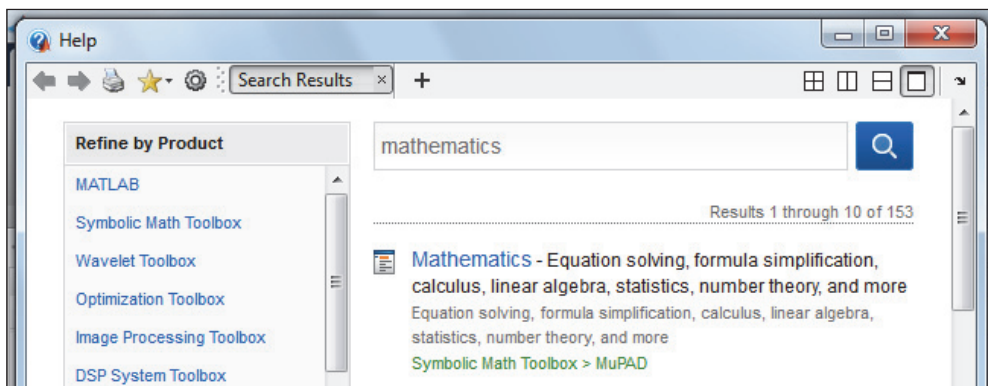


Imatge 1.2. Línea de comandaments.

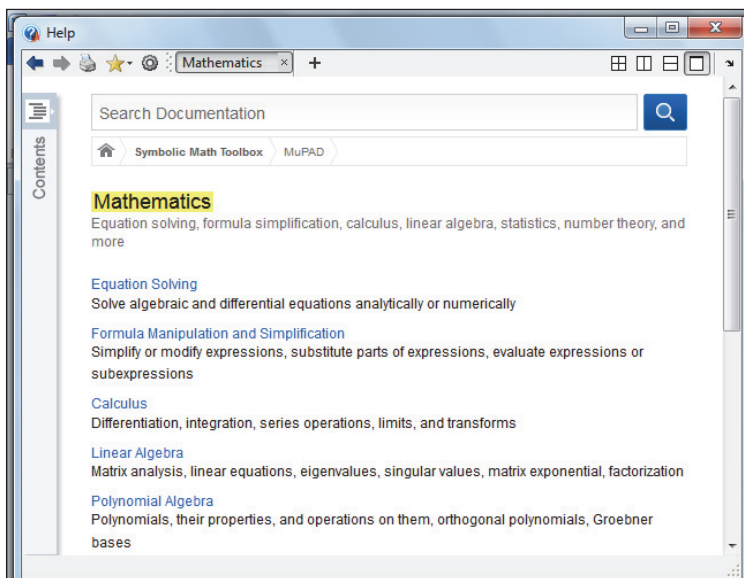
Aquest programa consta d'una ajuda a què s'accedeix amb la icona situat en la barra del programa (o F1). Aquesta ajuda conté totes les ordres o funcions del programa, on l'usuari sempre pot consultar qualsevol dubte que li sorgisca sobre l'ús de Matlab.



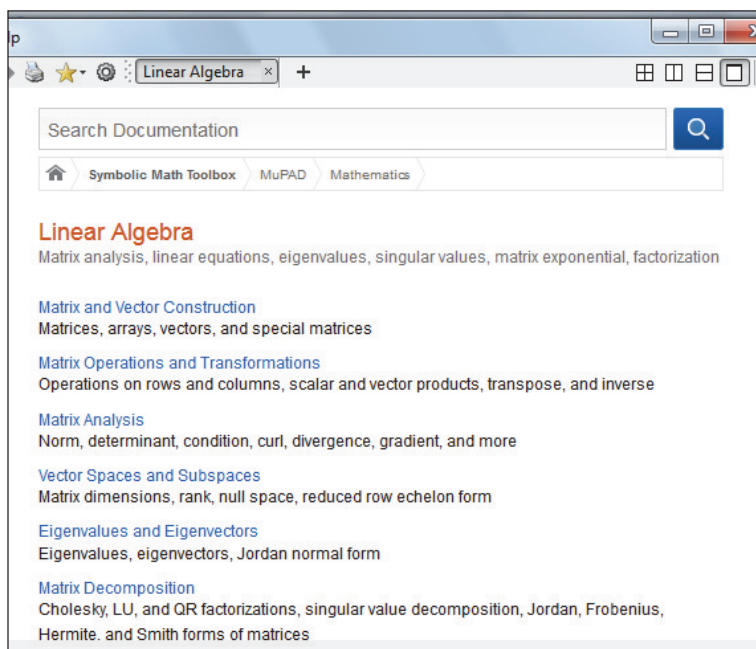
Imatge 1.3. Ajuda del programa.



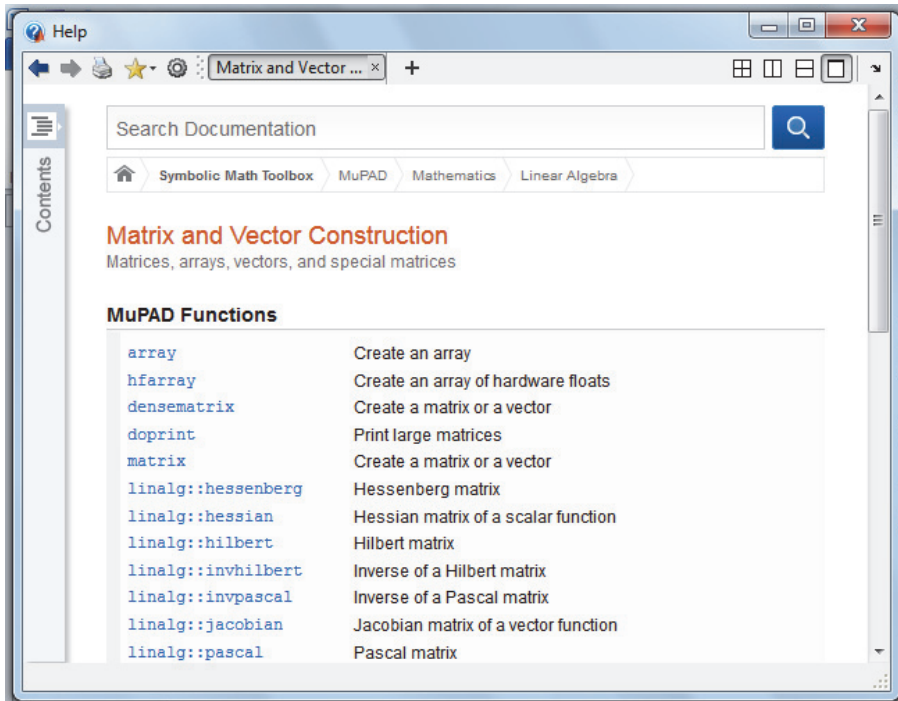
Imatge 1.4. Ajuda del programa.



Imatge 1.5. Ajuda del programa.



Imatge 1.6. Ajuda del programa.



Imatge 1.7. Ajuda del programa.

Un altre aspecte que cal tenir en compte és que Matlab té diversos formats de presentació dels resultats, de forma que podem elegir el que necessitem en cada moment. Aquests formats s'exposen a continuació:

- *Format curt* (format short): és el que presenta Matlab per defecte; aquest tipus de format ofereix 4 xifres decimals on l'última queda arrodonida.

```
>> x=2/3
x =
    0.6667
```

- Format científic (format short e): es presenta en pantalla la notació científica.

```
>> format short e;x  
x =  
6.6667e-01
```

- Format bank: format fix de dòlars i centaus.

```
>> format bank;x  
x =  
0.67
```

- Format llarg (format long): Matlab presenta en pantalla 15 xifres decimals.

```
>> format long;x  
x =  
0.666666666666667
```

- Format extrallarg (format long e): es presenta en pantalla la notació científica en 15 xifres decimals.

```
>> format long e;x  
x =  
6.66666666666666e-01
```

Ara podem tornar al format curt

```
>> format short
```

- **Format racional:** Matlab també permet la representació en format racional d'una expressió:

```
>> format rat
>> .3333

ans =

    3333/10000
```

Una altra forma és utilitzar el comandament **rats**:

```
>> 2/3

ans =

    0.6667

>> rats(ans)

ans =

    2/3
```

Nota:

Una expressió que inclogui `rats` no és operativa. Com a exemple incloem què obtindríem operant amb una expressió usant el comandament `rats`.

```
>> 2/3

ans =

    0.6667

>> rats(ans)+1

ans =

    33    33    33    33    33    33    51    48    52    33    33    33
```

1.3. Operacions amb escalars.

Matlab treballa amb escalars reals i complexos. Les operacions amb escalars que tractarem són les següents:

Tabla 1. 1. Alguns operadors de Matlab

Operacions	Operadors
suma	+
resta	-
producte	*
potència	a^p
divisió a entre b	a/b
divisió b entre a	a\b
arrel quadrada de nombre	sqrt(nombre)
arrel cúbica de nombre	(nombre)^(1/3)
arrel n-èsima de nombre	(nombre)^(1/n)

Per a utilitzar de forma adequada aquests operadors hem de tenir en compte l'ús dels parèntesis.

Exemple:

Per a comprovar el valor de l'expressió següent $2x^3y + \sqrt{y^5z}$, sent $x=1.5$, $y=3.7$, $z=9$, es procedeix de la manera següent:

```
>> x=1.5;y=3.7;z=9;
>> 2*x^3*y+sqrt(y^5*z)

ans =

    103.9747
```

Nota:

Per a utilitzar la potència, de vegades, cal prémer dues vegades $^{\wedge}$.

1.4. Generació de matrius.

Una matriu de dimensions $m \times n$ és un conjunt de $m \cdot n$ nombres escrits que formen un rectangle de m files i n columnes. Cada nombre és un element de la matriu.

Una matriu quadrada és aquella que té el mateix nombre de files que de columnes, és a dir, les dimensions són $m \times m$.

Per a utilitzar una matriu en el programa, els elements s'introdueixen sempre entre claudàtors, i s'escriuen ordenadament els elements de cada fila separats per espais en blanc o per comes i separant les columnes amb punt i coma o amb el caràcter enter (↵) al final de cada fila.

Exemple:

Si volem introduir la matriu següent:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

ho farem de les formes següents:

```
>> A=[1,2,3;4,5,6;7,8,9]
```

```
A =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> A=[1 2 3
```

```
4 5 6
```

```
7 8 9]
```

```
A =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> A=[1 2 3;4 5 6;  
7,8,9]  
  
A =  
  
     1     2     3  
     4     5     6  
     7     8     9
```

Per a recuperar una variable ja introduïda només cal prémer repetides vegades la tecla del cursor ascendent.

Les sentències en Matlab acaben sempre amb ↵, si les acabem amb punt i coma, no es fa ressò del resultat però Matlab les segueix tenint en memòria.

Per a comprovar l'observació, recuperem l'expressió de la matriu A, ja introduïda, afegint un punt i coma al final de l'expressió.

```
>> A=[1,2,3;4,5,6;7,8,9];  
>>
```

Hem de tenir en compte que si a una expressió li assignem una variable, aquesta queda emmagatzemada durant tota la sessió. En l'exemple hem assignat la lletra A a la matriu, i això ens permetrà recuperar-la amb posterioritat solament teclejant A.

```
>> A=[1,2,3;4,5,6;7,8,9];  
>> A  
  
A =  
  
     1     2     3  
     4     5     6  
     7     8     9
```

Si a una expressió no se li assigna cap variable, el programa li assigna la variable *ans*.

```
>> [1,2,3;4,5,6;7,8,9]

ans =

     1     2     3
     4     5     6
     7     8     9
```

Cal dir que el comandament *who* indica en cada moment quines són les variables emmagatzemades durant la sessió.

El programa distingeix entre majúscules i minúscules.

Si en l'exemple que ens ocupa, demanem a Matlab el valor de **a** en minúscules en compte de en majúscules, la seua resposta donarà un missatge d'error.

```
>> a
Undefined function or variable 'a'.

Did you mean:
>> A
```

Les dimensions d'una matriu s'obtenen amb la instrucció size. En el resultat, el primer element és el nombre de files i el segon, el de columnes.

```
>> size(A)

ans =

     3     3
```

1.4.1. Comandaments de generació de matrius.

Una altra manera de generar matrius en Matlab és mitjançant els comandaments que detallarem a continuació:

Una matriu fila és una matriu de dimensions $1 \times n$, és a dir, només té una fila.

- Dos punts (:): s'obtenen matrius (vectors fila). La sentència s'introdueix amb la seqüència següent:

Primer element : salt : últim element

Si volem generar un vector fila que vaja de l'1 al 10 amb un salt d'1 entre els elements, escriurem:

```
>> 1:1:10  
ans =  
     1     2     3     4     5     6     7     8     9    10
```

que és equivalent a:

```
>> 1:10  
ans =  
     1     2     3     4     5     6     7     8     9    10
```

Si volem generar un vector columna que vaja de l'1 al 10 amb un salt d'1 entre els elements, escriurem: $(1:1:10)'$ o $(1:10)'$.

- ones: crea una matriu en què tots els elements són 1.

```
>> help ones  
ones Ones array.  
ones(N) is an N-by-N matrix of ones.  
  
ones(M,N) or ones([M,N]) is an M-by-N matrix of ones.  
  
ones(M,N,P,...) or ones([M N P ...]) is an M-by-N-by-P-by-...  
ones.  
  
ones(SIZE(A)) is the same size as A and all ones.
```

Si volem generar una matriu d'uns amb 8 files i 4 columnes, escriurem:

```
>> ones(8,4)

ans =

     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

Una matriu identitat és una matriu quadrada on els elements de la diagonal són uns i la resta dels elements són zero.

- **eye**: crea una matriu amb 1 en la diagonal i 0 fora de la diagonal.

```
>> help eye
eye Identity matrix.
eye(N) is the N-by-N identity matrix.

eye(M,N) or eye([M,N]) is an M-by-N matrix with 1's on
the diagonal and zeros elsewhere.

eye(SIZE(A)) is the same size as A.

eye with no arguments is the scalar 1.

eye(..., CLASSNAME) is a matrix with ones of class specified
CLASSNAME on the diagonal and zeros elsewhere.

eye(..., 'like', P) is an identity matrix with the same data
and complexity (real or complex) as the numeric variable P.
```

Para seguir leyendo haga click aquí