AINA GARCIA ESPRIU
I MARTI RAMON ROS

Table of Contents

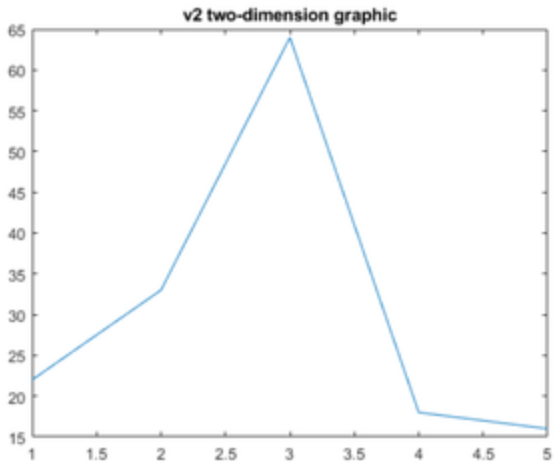
Tutorials about the enviroment

```
% 1.3 It's a list of the previously typed commands ordered by
 timestamp. The two ways to access it are by pressing Up-arrow or by
 typing "commandhistory" in the command window.

% 1.4 The current folder is a reference location that MATLAB uses
 to find files. This folder is sometimes referred to as the current
 directory, current working folder, or present working directory. You
 can order or group files by type, size, modified date, etc. You can
 also display this information.
```

Basic matrix operations. Matrix manipulation

```
% 2.1
v1 = [12 23 54 8 6] % v1

% 2.2
v2 = v1 + 10 % v1 + 10

% 2.3
plot(v2) % v2 plot
title('v2 two-dimension graphic')
```

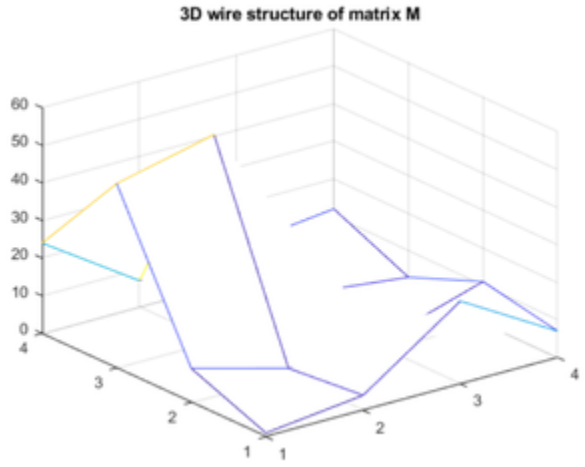
```
v1 =

    12    23    54     8     6


v2 =

    22    33    64    18    16
```



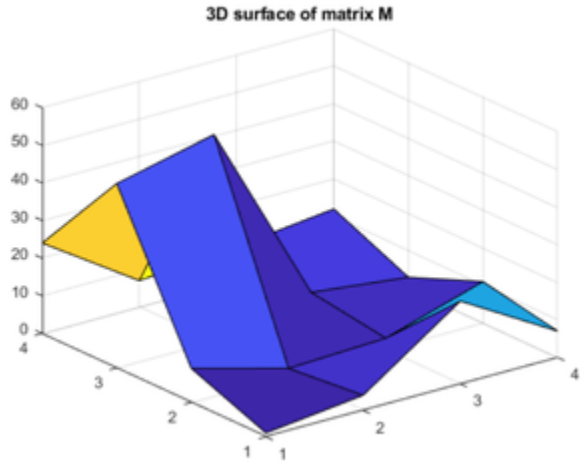

```
% 2.4
M = [1 4 22 7; 9 2 3 11; 49 55 6 3; 24 7 9 12] % matrix

% 2.5
Mt = transpose(M) % M transposed

% 2.6
Mi = inv(M) % M inverted

% 2.7 2.8
Mident = M * Mi % Demo

% 2.9
mesh(M) % wire structure
title('3D wire structure of matrix M')
```

```
M =

     1     4    22     7
     9     2     3    11
    49    55     6     3
    24     7     9    12


Mt =

     1     9    49    24
     4     2    55     7
    22     3     6     9
     7    11     3    12


Mi =

   -0.0223   -0.0755   -0.0063    0.0838
    0.0150    0.0651    0.0242   -0.0744
```

```
    0.0425   -0.0583   -0.0048    0.0298
    0.0039    0.1567    0.0021   -0.0631


Mident =

    1.0000         0   -0.0000         0
   -0.0000    1.0000   -0.0000    0.0000
    0.0000    0.0000    1.0000   -0.0000
   -0.0000    0.0000    0.0000    1.0000
```



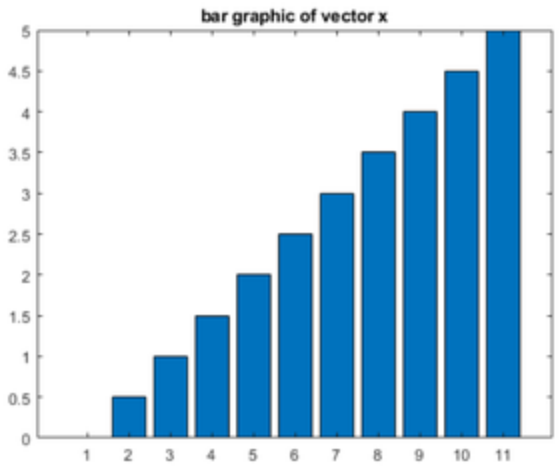

```
surf(M) % Surface
title('3D surface of matrix M')
```



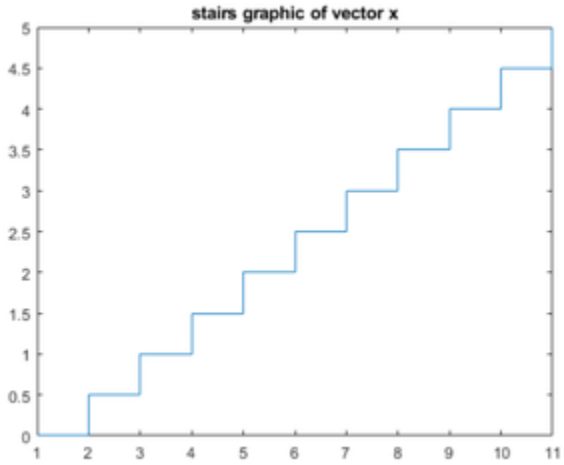
Graphics

```
x = 0:0.5:5; % 3.1 Creates a vector starting at 0 increasing 0.5 each
 position until reaching 5
```

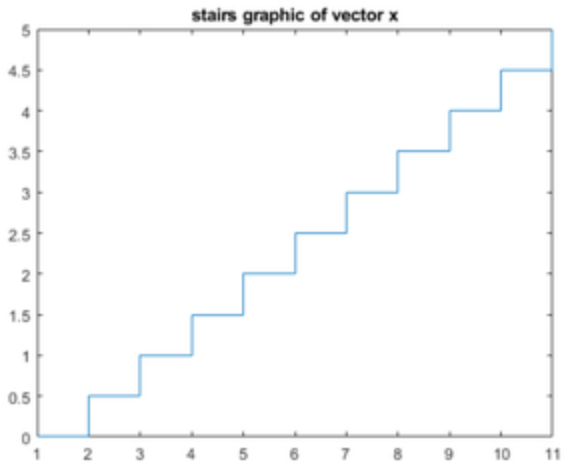
```
bar(x) % 3.2 Draws a vector as a bar's graphic
title('bar graphic of vector x')
```

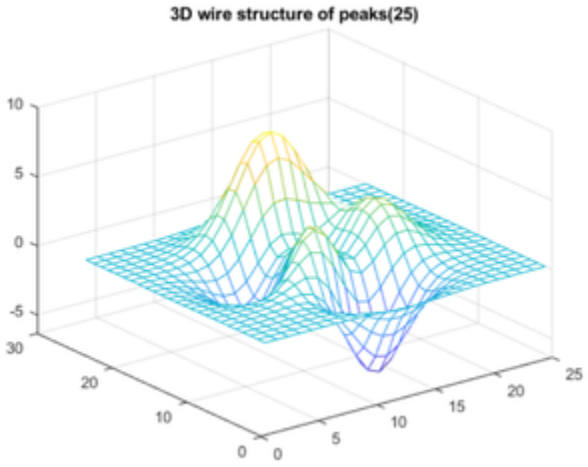



```
stairs(x) % 3.3 Draws a vector as a discretized line ressembling a
 staircase
title('stairs graphic of vector x')
```

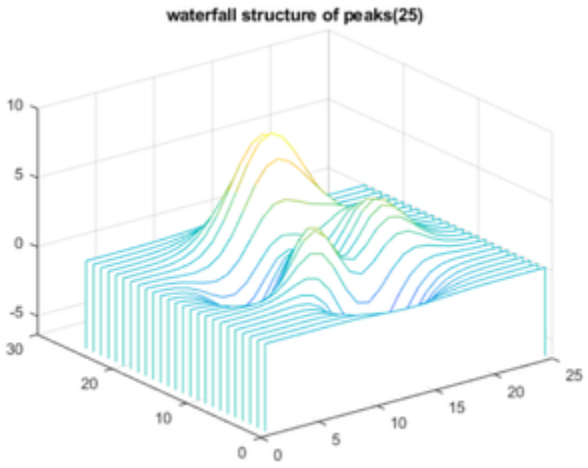



```
z = peaks(25); % 4.1 peaks is a function of two variables, obtained by
 translating and scaling Gaussian distributions, which is useful for
 demonstrating MESH, SURF, PCOLOR, CONTOUR, etc. #REFERENCE: matlab
 docu.

% 4.2 Waterfall plots only in one dimension, mesh in both of them
% MESH
figure,mesh(z)
title('3D wire structure of peaks(25)')
```

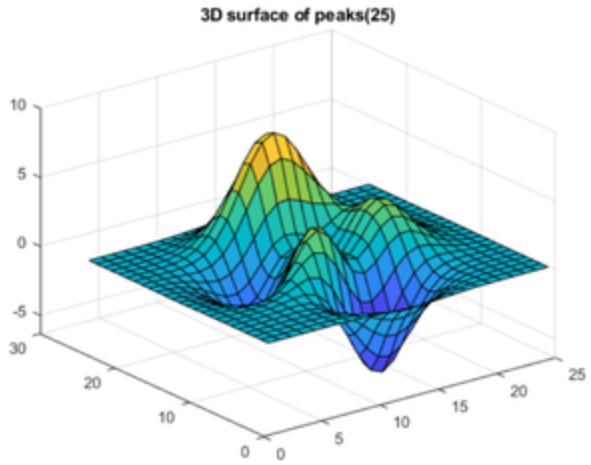





```
% WATERFALL
figure,waterfall(z)
title('waterfall structure of peaks(25)')
```

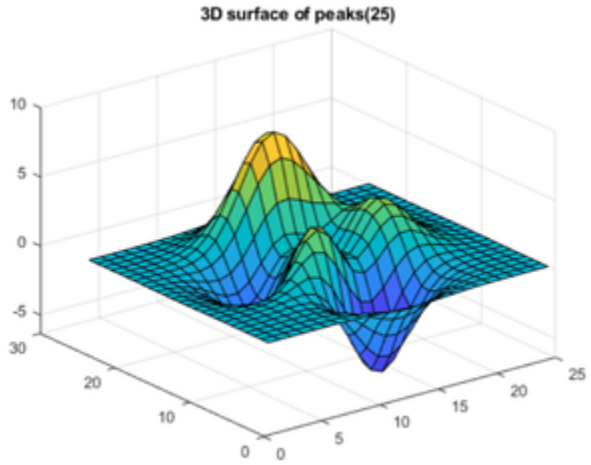


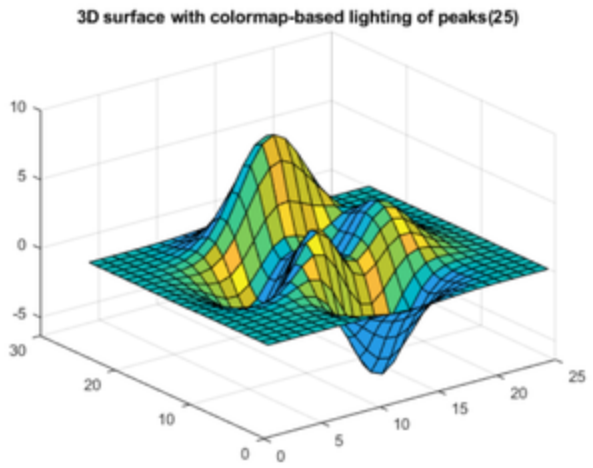
```
% 4.3 Surf paints de graph according to the heigh of the values, surf
 l
% plots it according to ambient, diffuse and specular lightning modes.
% SURF
figure,surf(z)
title('3D surface of peaks(25)')
```



SURFL

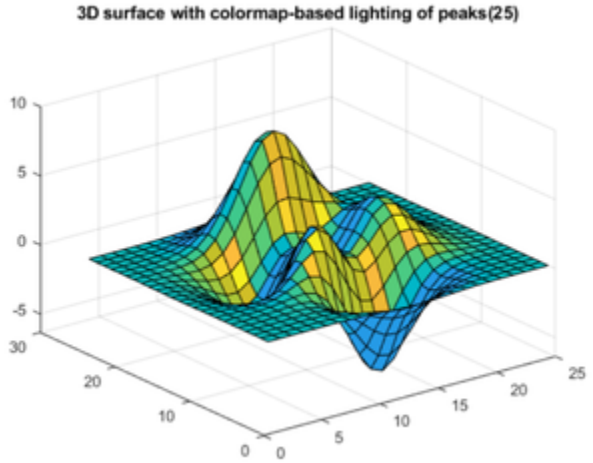
```
figure, surfl(z)
title('3D surface with colormap-based lighting of peaks(25)')
```

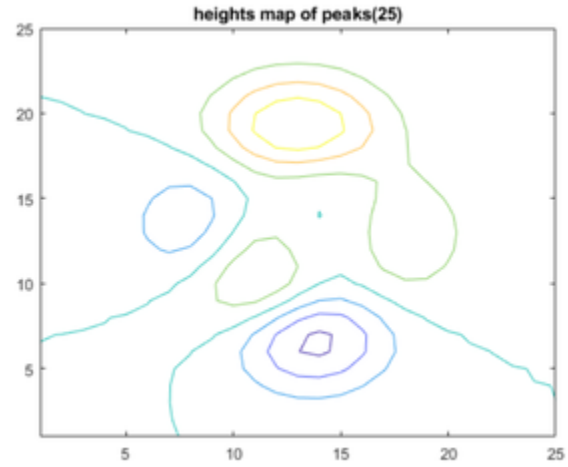




4.4

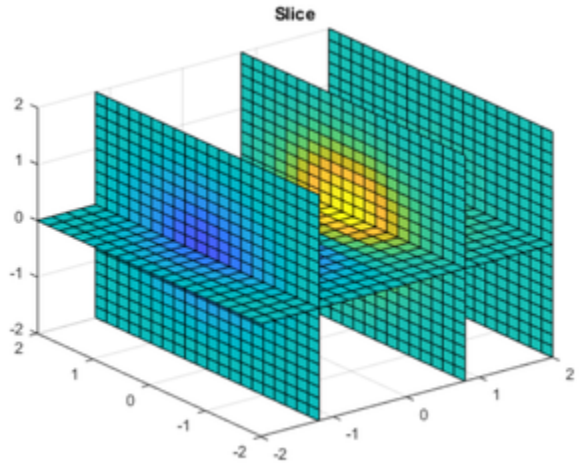
```
figure, contour(z) % Shows z as a heights map
title('heights map of peaks(25)')
```



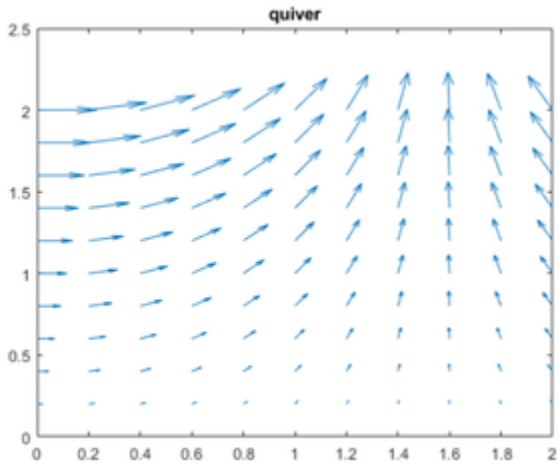


slice draws slices along the x,y,z directions at the points in the vectors Sx,Sy,Sz.

```
[X,Y,Z] = meshgrid(-2:.2:2);
V = X.*exp(-X.^2-Y.^2-Z.^2);

xslice = [-1.2,0.8,2];
yslice = [];
zslice = 0;
figure, slice(X,Y,Z,V,xslice,yslice,zslice)
title('Slice')
```

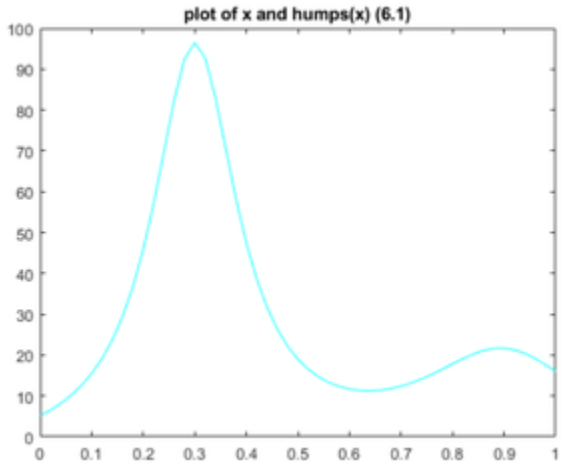



```
% quiver represents velocity vectors as arrows with components
[x,y] = meshgrid(0:0.2:2,0:0.2:2);
u = cos(x).*y;
v = sin(x).*y;

figure, quiver(x,y,u,v)
title('quiver')
```

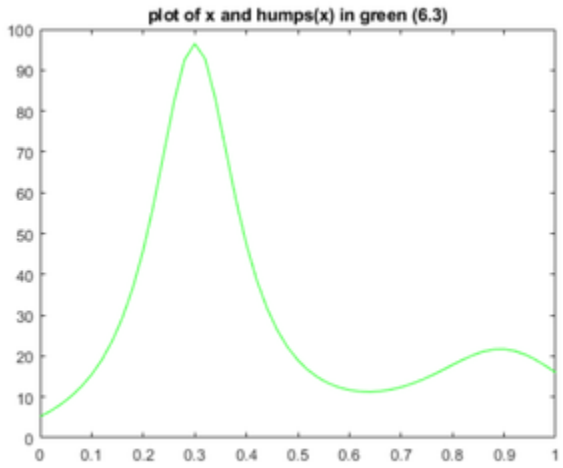



```
% 5.1
```

```
% winter uses cold and blueish colors
% summer uses greenish colors
% spring uses pink colors

% 6.1
x = 0:0.02:1; %creates a vector starting at 0 and ending at 10 with
 spacing of 0.02
hndl = plot(x, humps(x)); %humps is a function with maxim points at .3
 and .9
% Plot x, y plots vector X versus vector Y
set(hndl, 'Color', 'Cyan'); % Changes plot line color
title('plot of x and humps(x) (6.1)')
```

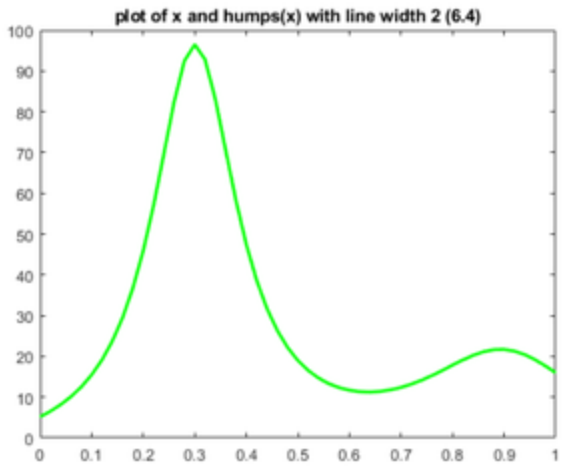



```
% 6.2 The graphic plotted

% 6.3 set(hndl, 'Color', 'Green');
set(hndl, 'Color', 'Green');
title('plot of x and humps(x) in green (6.3)')
```

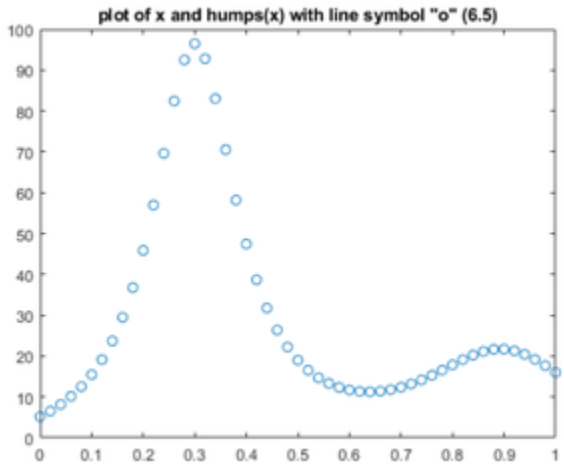



```
% 6.4 figure, set(hndl, 'LineWidth', 2);
```

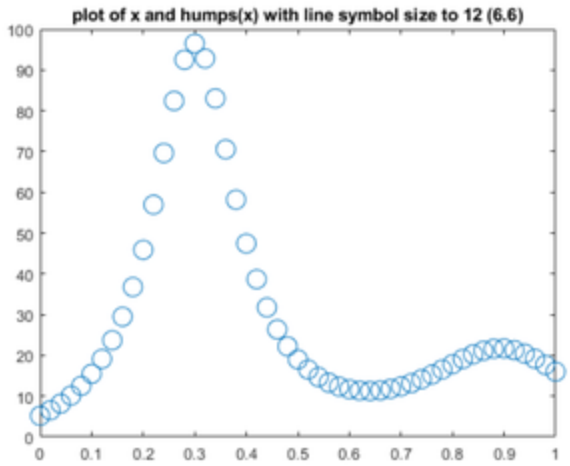
```
set(hndl, 'LineWidth', 2);
title('plot of x and humps(x) with line width 2 (6.4)')
```




```
% 6.5 plot(x, humps(x), 'o');
plot(x, humps(x), 'o');
title('plot of x and humps(x) with line symbol "o" (6.5)')
```




```
% 6.6 plot(x, humps(x), 'o', 'MarkerSize', 12)
plot(x, humps(x), 'o', 'MarkerSize', 12)
title('plot of x and humps(x) with line symbol size to 12 (6.6)')
```

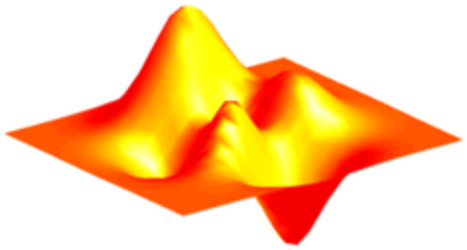


7.1 z = peak(25); creates a 25x25 matrix initialized following a Gaussian distribution surfl(z); plots z according to lightning parameters (4.3) shading interp; Matlab ref: Interpolated shading, which is also known as Gouraud shading, is piecewise bilinear; the Color in each segment or patch varies linearly and interpolates the end or corner values. colormap(hot); changes colors to hot palette axis off; disables axes

```
z = peaks(25);
x = surfl(z)
shading interp
colormap(hot)
axis off
title('3D surface with colormap-based lighting of peaks(25) (7.1)')
```

```
x =

  Surface with properties:

      EdgeColor: [0 0 0]
      LineStyle: '-'
      FaceColor: 'flat'
  FaceLighting: 'flat'
      FaceAlpha: 1
          XData: [25×25 double]
          YData: [25×25 double]
          ZData: [25×25 double]
          CData: [25×25 double]

  Use GET to show all properties
```

3D surface with colormap-based lighting of peaks(25) (7.1)



Programming

```
% 1.3 Usage example
ans = calculate_rectangle_area(2,3)

% 1.4
fib = fibonacci(5)
% 1.5 Yes
fib2 = fibonacci_rec(5)


% 2.2
x = .01;
y = zeros(1000);
tStart = tic;
for k = 1:1001
    y(k)=log10(x);
    x = x + 0.1;
end
t1 = toc(tStart) % 2.2
tStart = tic;

x = .01:.01:10;
y = log10(x);

t2 = toc(tStart) % 2.2

% 3.2
tStart = tic;
for n = 1:32
    r(n) = rank(magic(n));
end
t1 = toc(tStart) % 3.2

tStart = tic;
r = zeros(32,1);
```

```
for n = 1:32
    r(n) = rank(magic(n));
end
t2 = toc(tStart) % 3.2

% 4.2
A = [5 7 8; 0 1 9; 4 3 6] % crea una matriu 3x3
A(:,:,2) = [1 0 4; 3 5 6; 9 8 7] % afegeix una dimensio 2 i
 inicialitza, treient una matriu 3x3x2
% 4.3
a1 = A(1,2) % access to row 1, column 2 of the first dimension
 (default)
a2 = A(1,2,1) % normal acces to 3 dimensional matrix
a3 = A(1,2,2) % same as before but with the second dimension

% 4.4
a4 = A(3,2,2)

% 5.2
personal_record = struct('name', 'SampleName', 'age', 22, 'size', [10
 15 39])

% 5.3
age = personal_record.age

% 5.4 Yes, example:
nested_struct = struct('nested2',
 personal_record, 'nested1ex', 'sampleValue')

function [area] = calculate_rectangle_area(side1, side2)
area = side1*side2;
end
function [v] = fibonacci(n)
if n == 0
    v(1) = 1
end
if n == 1
    v(1) = 1
    v(2) = 2
end
if n > 1
    v(1) = 1
    v(2) = 1
    k = 2;
    while k <= n
        k = k + 1;
        v(k) = v(k-2) + v(k-1);
    end
end
end
function [F] = fibonacci_rec(n)
    for k = 1:n
        if k == 0
            F = [1];
```

```
        elseif k == 1
            F = [1 1];
        else
            F = fibonacci_rec(k-1);
            F(end + 1) = F(end) + F(end-1);
        end
    end
end
```

```
ans =

     6

v =

     1

v =

     1     1

fib =

     1     1     2     3     5     8

fib2 =

     1     1     2     3     5     8

t1 =

   6.4416e-04

t2 =

   6.5824e-04

t1 =

    0.0133

t2 =

    0.0031
```

```
A =

     5     7     8
     0     1     9
     4     3     6


A(:,:,1) =

     5     7     8
     0     1     9
     4     3     6


A(:,:,2) =

     1     0     4
     3     5     6
     9     8     7


a1 =

     7


a2 =

     7


a3 =

     0


a4 =

     8


personal_record =

  struct with fields:

    name: 'SampleName'
     age: 22
    size: [10 15 39]


age =

    22
```

```
nested_struct =

  struct with fields:

      nested2: [1×1 struct]
    nested1ex: 'sampleValue'
```

Published with MATLAB® R2018b