

UNIVERSITAT POLITÈCNICA DE CATALUNYA

---

# Sessió 1

---

INFORME

VISIÓ PER COMPUTADOR

*Autors*

DAVID LATORRE  
ADRIÀ AUMATELL

## EXERCICI 1:

A:

Ho podem fer simplement multiplicant els vectors, o multiplicant-los component a component:

```
1 %% apartat A
2 A_vector1 = [5 4 3 2 1];
3 A_vector2 = [1; 1; 1];
4 % Multiplicant directament els vectors
5 A_MultDirecte = A_vector2*A_vector1
6 % Multiplicant-los component a component
7 A_MultComponent = A_vector2.*A_vector1
```

Output:

```
A_MultDirecte =
     5     4     3     2     1
     5     4     3     2     1
     5     4     3     2     1
```

```
A_MultComponent =
     5     4     3     2     1
     5     4     3     2     1
     5     4     3     2     1
```

B:

Igual que l'apartat A, però en aquest cas el vector que només conté uns serà un vector fila, i l'altre vector serà un vector columna:

```
9 %% apartat B
10 B_vector1 = [1 1 1];
11 B_vector2 = [-2;-1;0];
12 % Multiplicant directament els vectors
13 B_MultDirecte = B_vector2*B_vector1
14 % Multiplicant-los component a component
15 B_MultComponent = B_vector2.*B_vector1
```

Output:

```
B_MultDirecte =  
  
    -2    -2    -2  
    -1    -1    -1  
     0     0     0  
  
B_MultComponent =  
  
    -2    -2    -2  
    -1    -1    -1  
     0     0     0
```

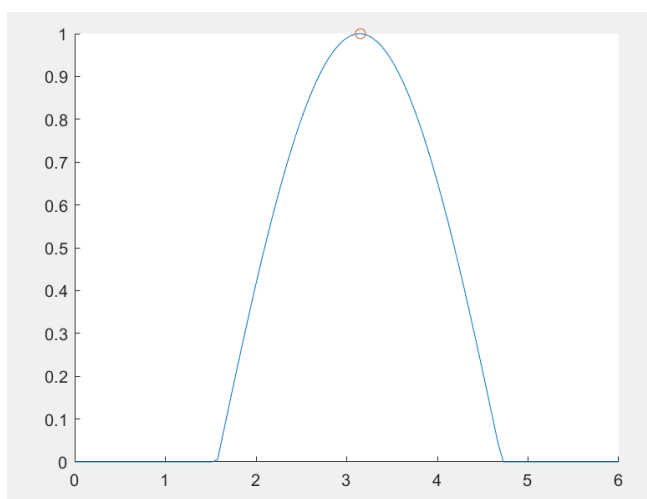
## EXERCICI 2:

Amb la funció “linspace” podem determinar el nombre d’elements que tindrà l’eix x, i amb la funció hold podem mostrar més d’una gràfica a la vegada, d’aquesta manera podem mostrar la gràfica que mostra la línia i la gràfica que mostra el punt màxim al mateix temps.

Pel que fa a l’eix y, aquest és equivalent a  $\cos(x)$  invertit, excepte que quan el valor de l’eix y és negatiu, aquest el convertim al valor 0.

```
5      x = linspace(0,6,100);  
6      y = cos(x).*(-1);  
7      y(y<0) = 0;  
8      [y_max, index_max] = max(y);  
9      x_max = x(index_max);  
10     figure  
11     hold on  
12     plot(x, y)  
13     scatter(x_max, y_max)
```

Output:

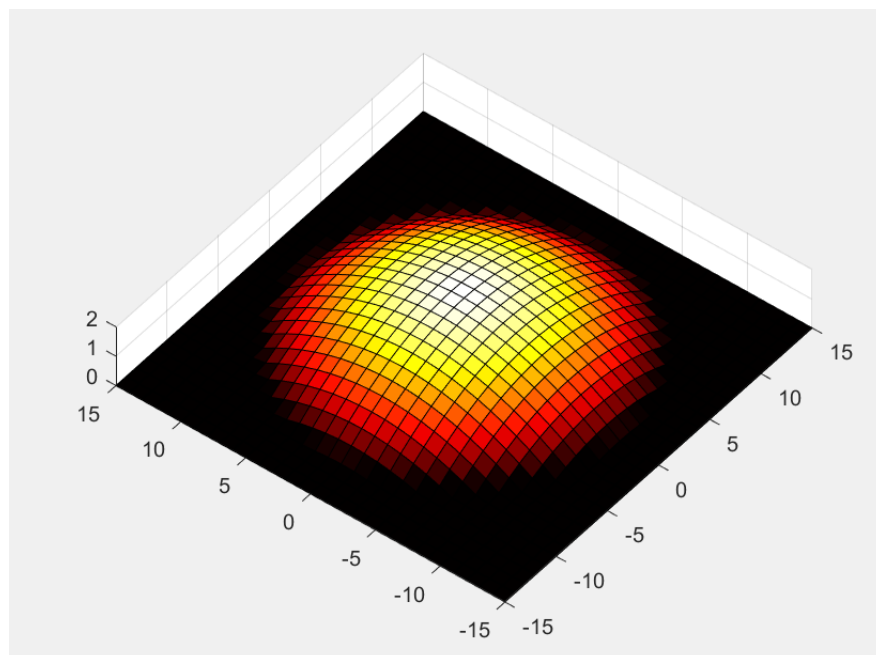


### EXERCICI 3:

Mitjançant la funció `meshgrid` podem crear els eixos `x` i `y` del plot per tal que vagin de -15 a 15. Després definim `z` en funció dels valors dels vectors `x` i `y` de manera que coincideixin aproximadament amb els de la imatge a imitar. A continuació tallem el gràfic per tal que tots els punts amb `z` inferior a zero passin a ser zero. Finalment, especifiquem que volem un plot amb colors (`surf`) amb la paleta `hot`.

```
1 [x, y] = meshgrid(-15:1:15);
2 z = -(x.^2+y.^2)/75 + 2;
3
4 %fem que els punts on z és negativa siguin 0
5 z(z<0) = 0;
6
7 %pintem el plot
8 surf(x,y,z);
9 colormap hot;
10
```

output:

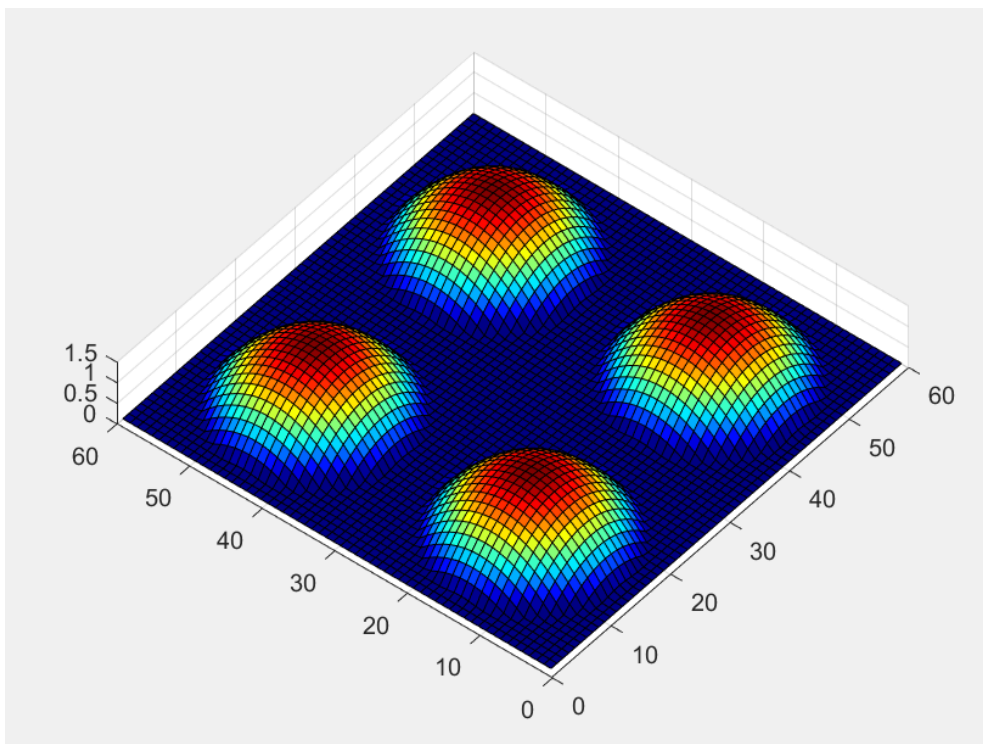


#### EXERCICI 4:

Aprofitant el plot fet a l'exercici anterior podem fer una concatenació de matrius per tal de mostrar-lo quatre vegades.

```
1 %creació de la figura
2 [x, y] = meshgrid(-14:1:15);
3 za = -(x.^2+y.^2)/100 + 1.5;
4
5 %concatenació
6 [x2, y2] = meshgrid(1:1:60);
7 z = [za za; za za];
8
9 %pintem el plot
10 z(z<0)=0;
11 colormap jet;
12 surf(x2,y2,z);
13
```

output:



## EXERCICI 5:

Per tal de crear els punts de forma aleatòria ubicats a l'espai xyz, primer de tot necessitem crear-los en espai esfèric. És a dir, a l'hora de crear els punts aleatoris, per a cada un agafarem un valor aleatori del seu angle azimuth, de l'angle d'elevació, i del valor radial (que en aquest cas es trobarà de 0 a 50).

Hem de tenir en compte que a l'hora de crear aquests valors aleatoris ho fem amb la distribució correcta, només l'angle azimuth segueix una distribució uniforme.

Finalment, un cop tenim totes les coordenades esfèriques, les passem al sistema de coordenades cartesià, i ja podem mostrar el plot.

Amb la variable "s" i "c" de scatter3(x,y,z,s,c) podem manipular respectivament la grandària i el color dels punts.

Farem que els punts que es situïn més a l'interior de les esferes siguin més grans i obscurs, mentre que els que es situïn més a l'exterior siguin més blaus i també més petits.

```
1      % creació dels punts aleatoris en sistema de coordenades esfèric
2      N = 1000;
3      theta = 2*pi*rand(N,1);
4      phi = asin(2*rand(N,1)-1);
5      radii = 50*(rand(N,1).^(1/50));
6
7      % conversió al sistema de coordenades cartesià
8      [x,y,z] = sph2cart(theta, phi, radii);
9
10     %trasllat al punt central 100,100,100
11     x = x + 100;
12     y = y + 100;
13     z = z + 100;
14
15     % plot
16     S = ((-radii)+50)*16.0;
17     C = [zeros(N,1) zeros(N,1) (radii./50).^(50)];
18     scatter3(x, y, z, S, C, 'filled')
19     axis vis3d equal, grid on, box on
20     xlabel X, ylabel Y, zlabel Z
```

Output:

