MATLAB

3D graphics

2020-09-18

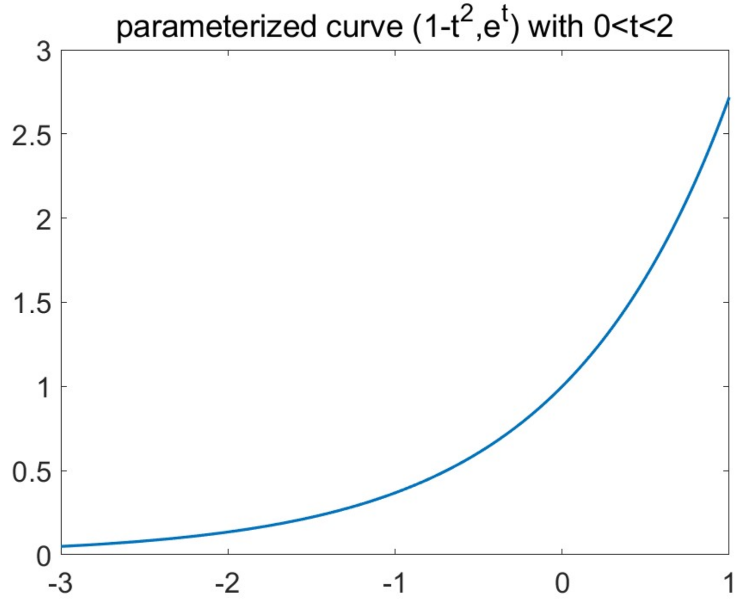
JINKYU YU (hortensia@kaist.ac.kr)

MATLAB command : plot3

MATLAB draw the graph by dotting the points.
In 2D-graphics, we learned how to draw the function $y = f(x)$.
We can generalize this way to the parameterized curve.

Let's consider the curve $\alpha(t) = (1 - t^2, e^t) \quad on \quad 0 \leq t \leq 2.$

1. Make the domain points)
   t = linspace(0, 2, 100);
2. Compute the x and y values)
   x = 1-t.^2;
   y = exp(t);
3. Draw the curve)
   figure(1), plot(x, y)



MATLAB command : plot3

One can think the function $y = f(x)$ as the parameterized curve $\alpha(x) = (x, f(x))$.
So, parameterized curve is more general version. And it can be a useful.

Let's draw the unit circle ($x^2 + y^2 = 1$).
Here is 2 way to draw the circle.

Combine the 2 functions $y = \pm\sqrt{1 - x^2}$	parameterized curve $\alpha(\theta) = (\cos(\theta), sin(\theta))$
x = linspace(-1, 1, 100);	theta = linspace(0, 2*pi, 200);
y = sqrt(1-x.^2);	x = cos(theta); y = sin(theta);
figure(1), hold on plot(x, y) plot(x, -y) hold off	figure(1), plot(x, y)

MATLAB command : plot3

Now we will draw the curve in R^3.
Just compute the 3^{rd} coordinate and use the MATLAB command plot3 instead of plot.

Let's consider the curve $\alpha(t) = (1 - t^2, e^t, \cos(t))$ on $0 \le t \le 2.$

1. Make the domain points)

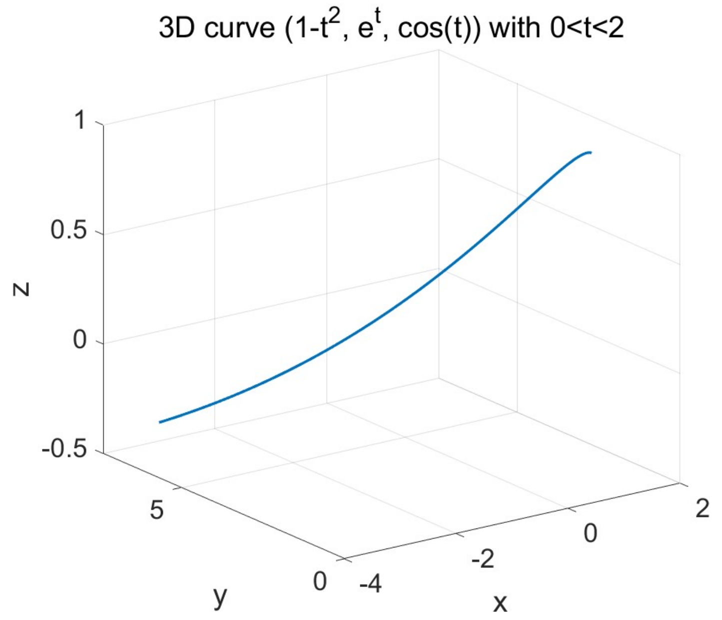
```
t = linspace(0, 2, 100);
```

2. Compute the x and y values)

```
x = 1-t.^2;
y = exp(t);
z = cos(t)
```

3. Draw the curve)

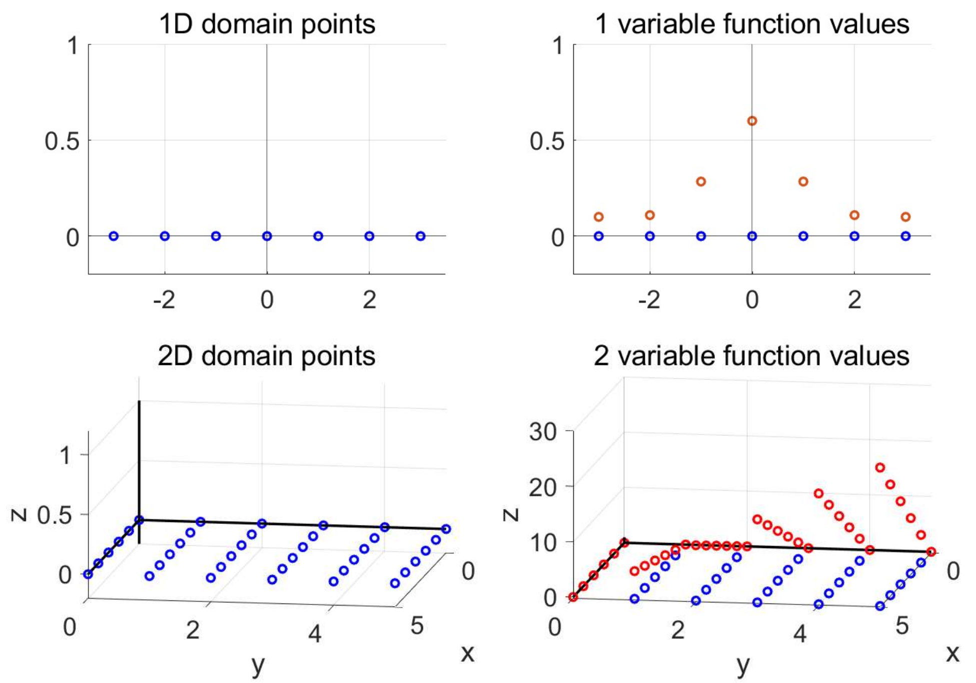
```
figure(1), plot3(x, y, z)
```



MATLAB command : meshgrid

- Make the domain points plain version.

Here we will learn how to draw the surface.
For example, consider $S(x,y) = (x, y, xy)$
Unlike curves, we need 2 variables to draw the surface.
It means that we need the domain points as a matrix not a vector.



Blue points are the domain points.
Red points are the function values.

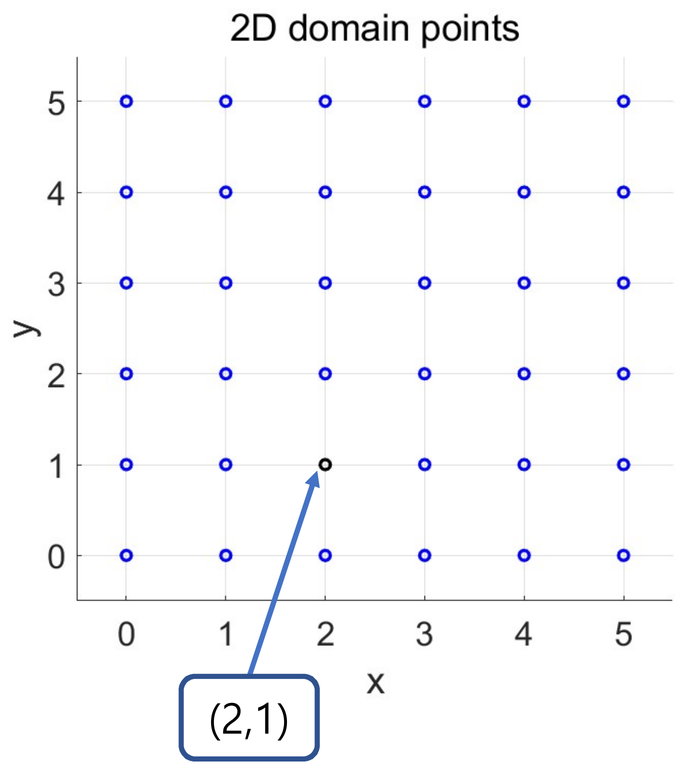
MATLAB command : meshgrid

- Make the domain points plain version.

Now focus on the 2D domain points.
Each of points has 2D vector like (2,1) for its coordinate.

Let's extract the first coordinates then it looks like

First coordinate(x)					
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5

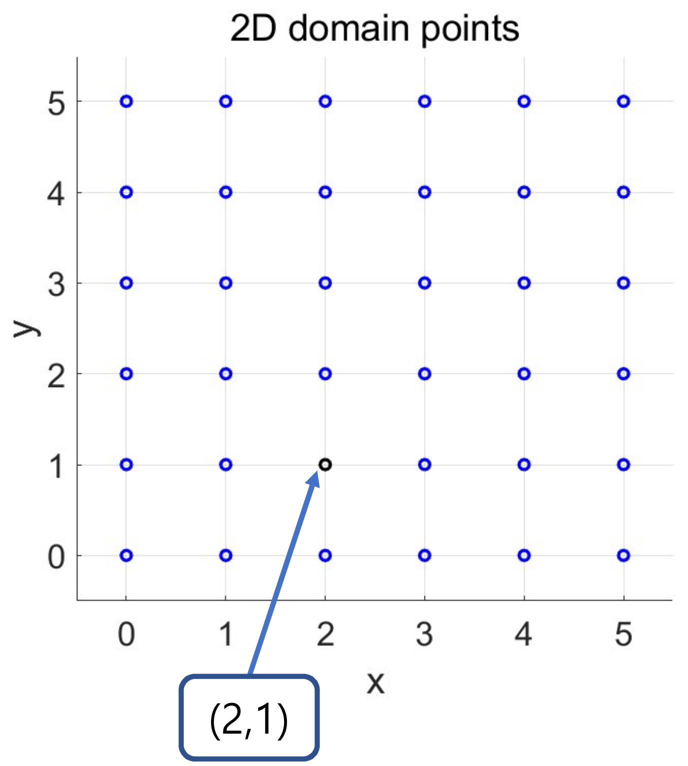


MATLAB command : meshgrid

- Make the domain points plain version.

Similarly extract the second coordinates.

Second coordinate(y)					
5	5	5	5	5	5
4	4	4	4	4	4
3	3	3	3	3	3
2	2	2	2	2	2
1	1	1	1	1	1
0	0	0	0	0	0



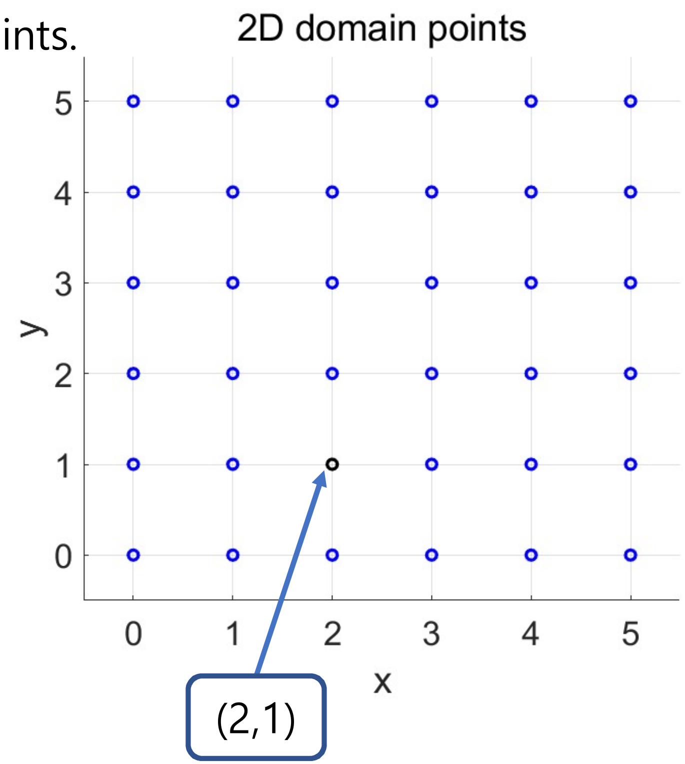
MATLAB command : meshgrid

- Make the domain points plain version.

To draw the surface, we need the 2 matrices for a domain points.

First coordinate(x)					
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5
0	1	2	3	4	5

Second coordinate(y)					
5	5	5	5	5	5
4	4	4	4	4	4
3	3	3	3	3	3
2	2	2	2	2	2
1	1	1	1	1	1
0	0	0	0	0	0



MATLAB command : meshgrid
- Make the domain points plain version.

MATLAB command "meshgrid" gives these 2 matrices.

For example, consider $S(x,y) = (x,y,xy)$ on $D = \{(x,y) \mid 0 \le x \le 1 , 2 \le y \le 4\}$.

1. Prepare the x vector and y vector.

```
x = 0: 0.1 :1;          % x is 1*11 vector.
y = 2: 0.1: 4;          % y is 1*21 vector.
```

2. Make the domain points matrices.

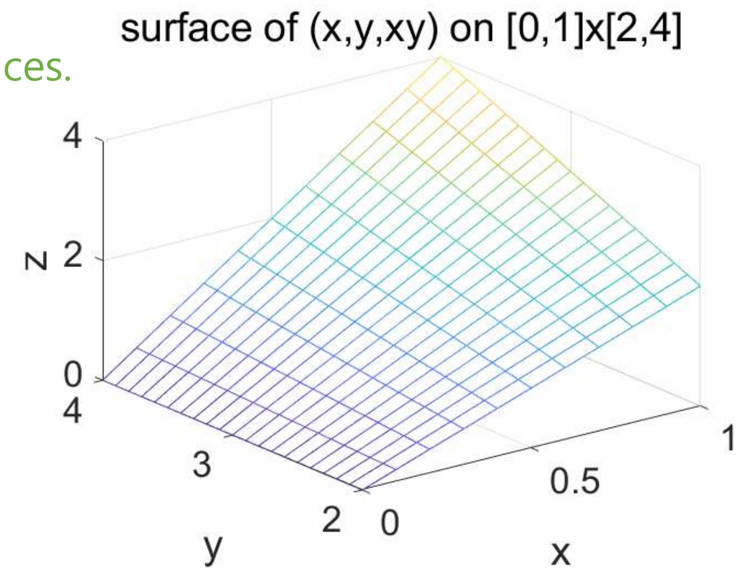
```
[X, Y] = meshgrid(x ,y);          % X and Y are 21*11 matrices.
```

3. Compute the function values.

```
Z = X.*Y;          % Z is 21*11 matrix.
```

4. Draw the surface

```
figure(1), mesh(X, Y, Z)
```



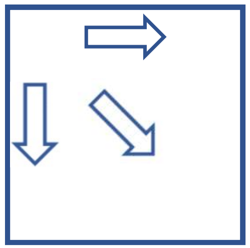
MATLAB command : meshgrid

- Make the domain points plain version.

In previous example, one can check that

$$Y = \begin{bmatrix} 2 & 2 & \cdots & 2 & 2 \\ 2.1 & 2.1 & \cdots & 2.1 & 2.1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 3.9 & 3.9 & \cdots & 3.9 & 3.9 \\ 4 & 4 & \cdots & 4 & 4 \end{bmatrix} \quad \text{not } Y = \begin{bmatrix} 4 & 4 & \cdots & 4 & 4 \\ 3.9 & 3.9 & \cdots & 3.9 & 3.9 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 2.1 & 2.1 & \cdots & 2.1 & 2.1 \\ 2 & 2 & \cdots & 2 & 2 \end{bmatrix}$$

[X, Y] = meshgrid(x ,y);

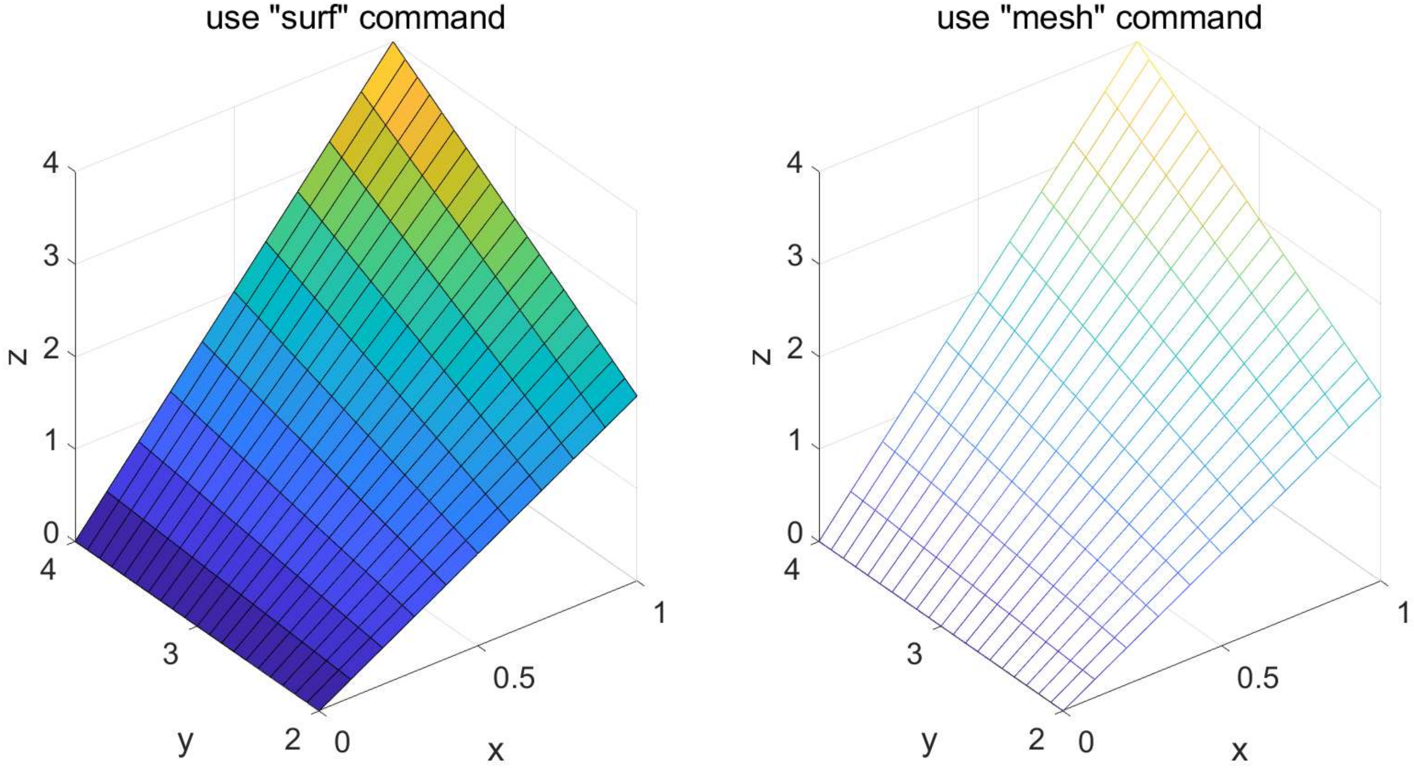


Since human and computer write the information in direction [left->right] and [top->bottom], Matrix is formed in that direction. So X is normal [left->right] and Y looks like flipped [top->bottom]. But there is no problem with the calculation and draw.

MATLAB command : surf , mesh

- Draw the surface

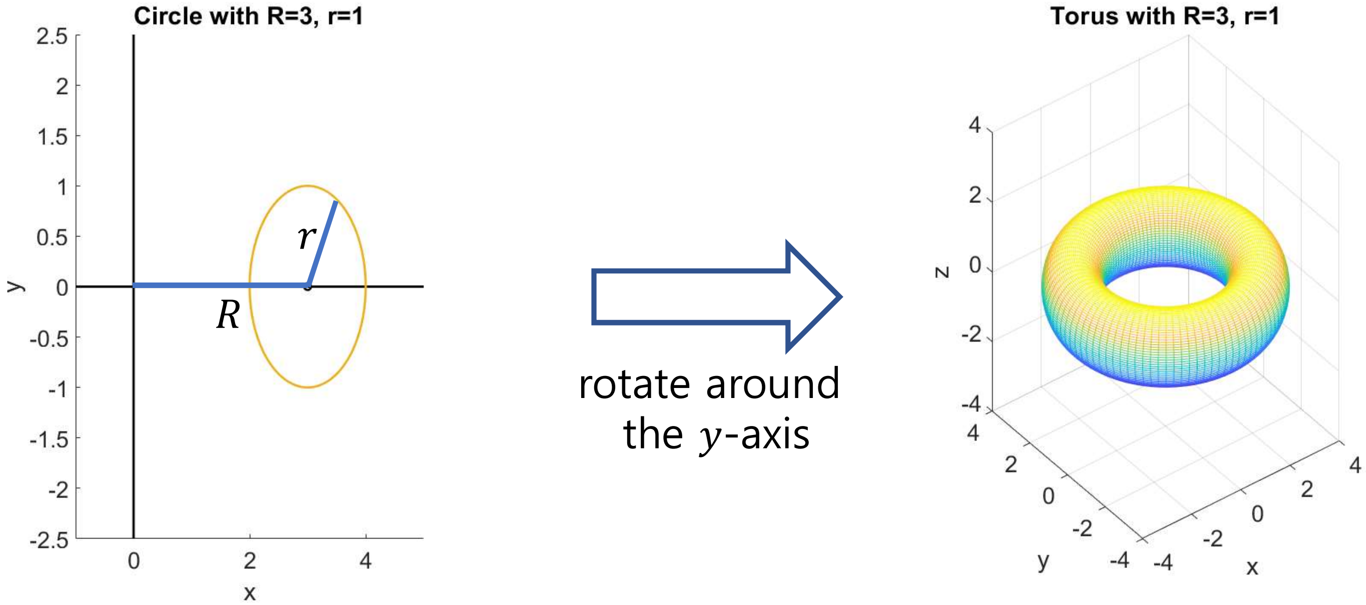
Both of MATLAB command “surf(X, Y, Z)” and “mesh(X ,Y, Z)” draw the surface with coordinates (x, y, z).
“surf(X, Y, Z)” give the colored area and “mesh(X, Y, Z)” give the blank area.



MATLAB command : surf , mesh

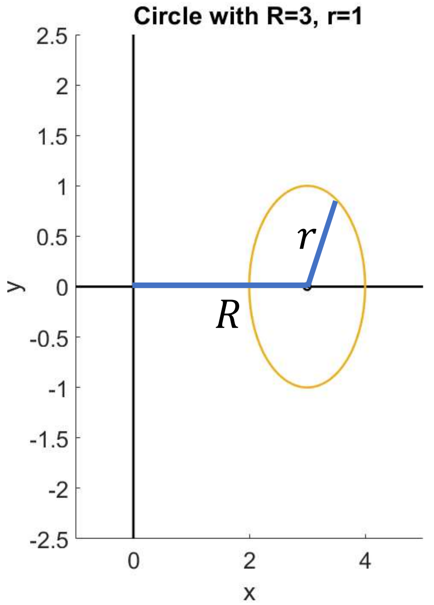
- Draw the surface of the revolution

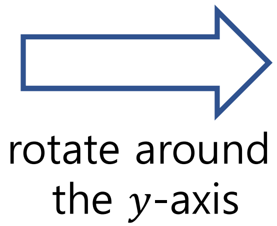
Now we will learn how to draw the parameterized surface.
For example, consider the "Torus" which looks like a donut.
By rotating the circle around the y-axis, one can get the Torus.

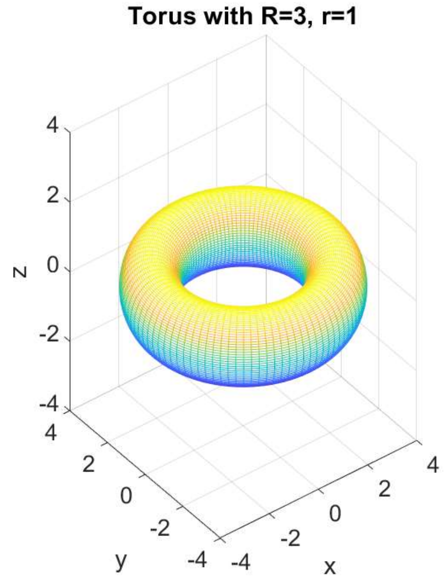


MATLAB command : surf , mesh
- Draw the surface of the revolution

One can parametrize the Circle as $\alpha(u) = \left(R + rcos(u)\ ,\ rsin(u)\right)$ with $0 \leq u \leq 2\pi$
Then the Torus can be parametrized as $S(u,v) = \left(\left(R + r * cos(u)\right)\cos(v)\,,\left(R + r * cos(u)\right)\sin(v)\,,\ rsin(u)\right)$ with $0 \leq u, v \leq 2\pi$.



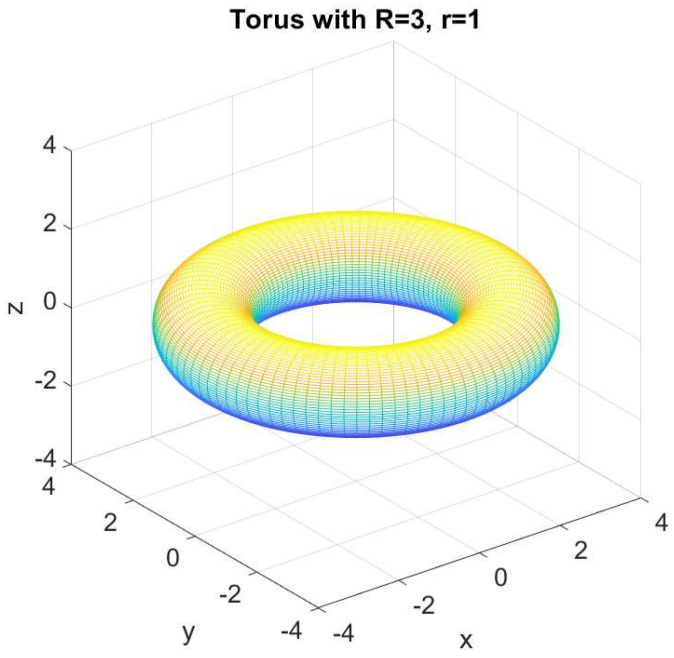




MATLAB command : surf , mesh
- Draw the surface of the revolution

To draw the Torus with $R = 3, r = 1$.

1. Prepare the u vector and v vector. (parameter u, v = angle information)
```
u = linspace(0, 2*pi, 200);
v = linspace(0, 2*pi, 200);
```
2. Make the domain points matrices. (U, V = angle information)
```
[U, V] = meshgrid(u ,v);
```
3. Compute the x, y, z coordinates.
```
X = (R + r*cos(U)) * cos(V);
Y = (R + r*cos(U)) * sin(V)
Z = r*sin(U)
```
4. Draw the surface
```
figure(1), mesh(X, Y, Z)
```



MATLAB command : contour
- Draw contour curves(level curves) in 2D domain plain.

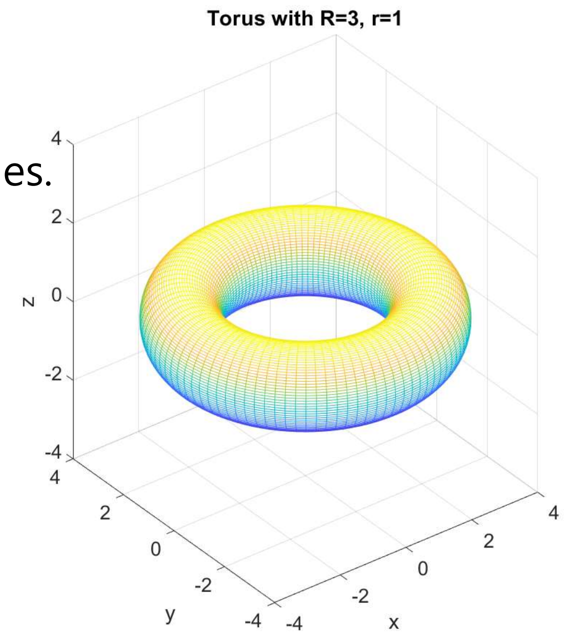
Let's assume that one computed the x, y, z coordinates as X, Y, Z.
When it is complicated to see the whole structure, one can see the contour curves to know the rough structure.

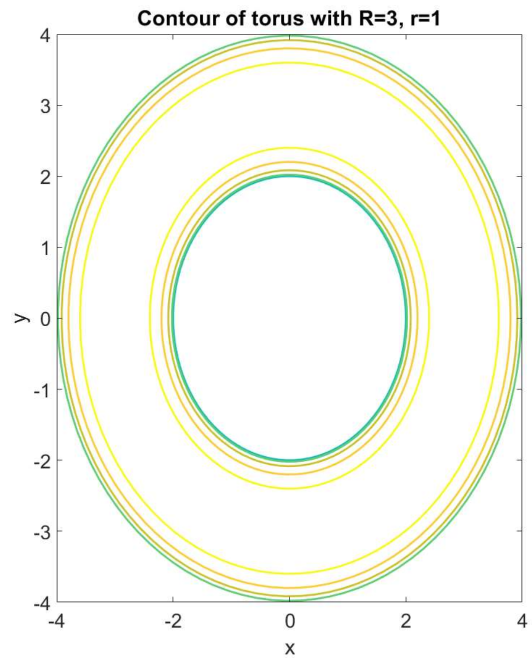
MATLAB code is like below

```
figure(1), contour(X, Y, Z);
```

MATLAB command "contour"
do not give information about values.

We do not know what value each line has.



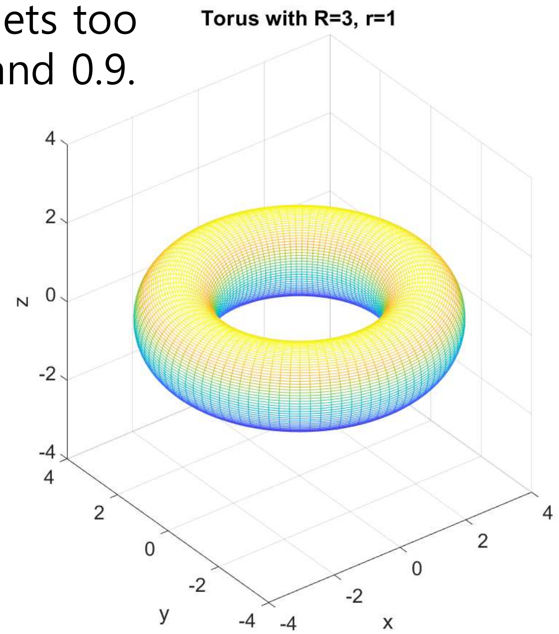


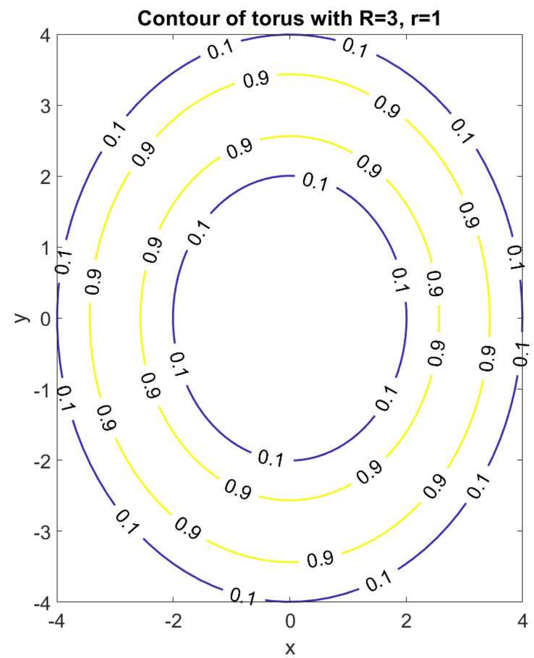
MATLAB command : contour & clabel
- Draw contour curves(level curves) in 2D domain plain.

MATLAB command “clabel” gives the value of each line.

```
figure(1),
[C, h] = contour(X, Y, Z);
clabel(C, h);
```

* If I put values on all the lines, it gets too messy, so I only put values on 0.1 and 0.9.



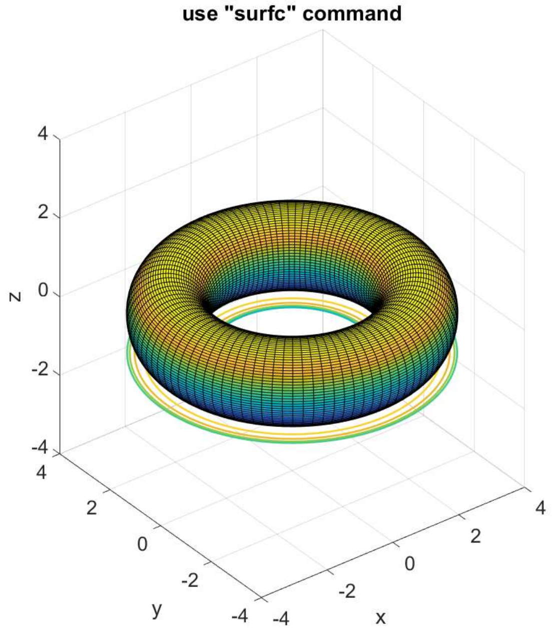


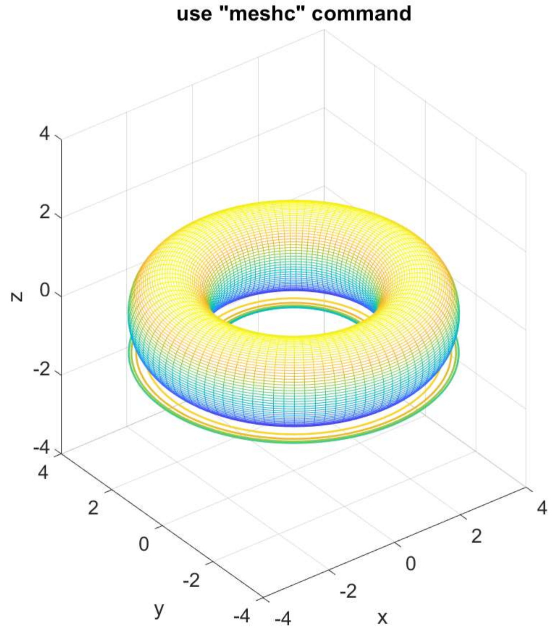
MATLAB command : surfc & meshc

- Draw the surface and contour curves(level curves) in same axis.

You can think of “surfc” as the sum of “surf” and “contour”.
Similarly, “meshc” = “mesh” + “contour”.

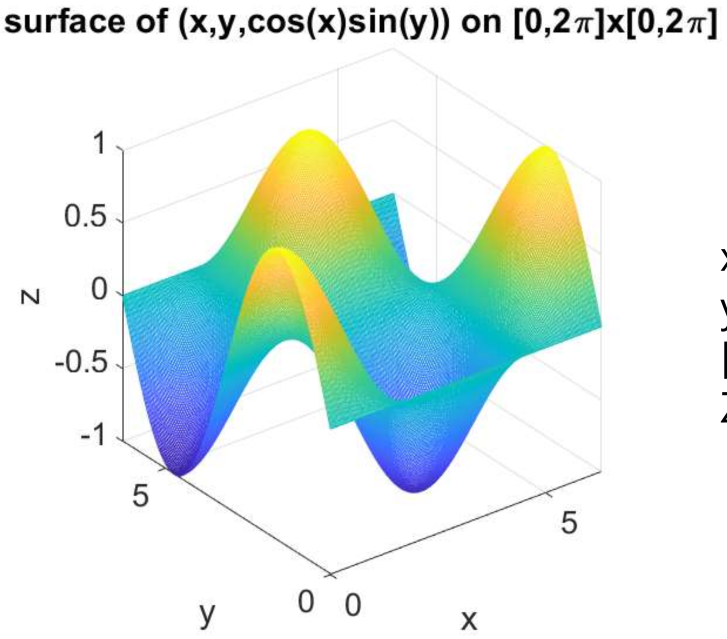
Both of “surfc” and “meshc” draw the surface on 3D
and draw the contour curves on 2D.



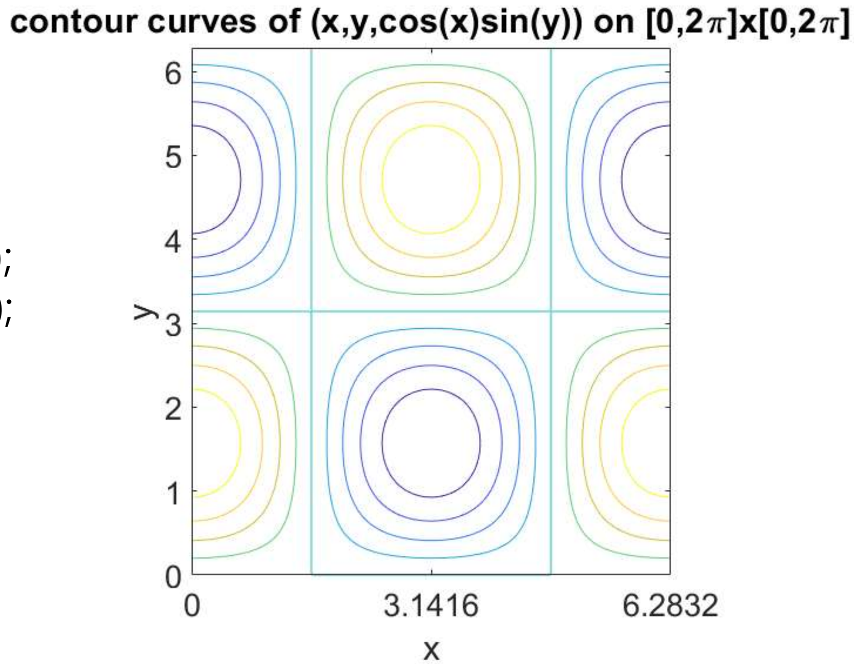


MATLAB command : gradient & quiver
- calculate the gradient & draw the vector field(gradient)

Let's consider the surface $S(x,y) = (x, y, \cos(x)\, sin(y))$ on $0 \le x, y \le 2\pi$.
One can draw the surface and contour curves.
And now we want to get the gradient vector of the function $z = \cos(x)\, sin(y)$.




```
x = linspace(0, 2*pi, 200);
y = linspace(0, 2*pi, 100);
[X, Y] = meshgrid(x, y);
Z = cos(X).*sin(Y);
```



MATLAB command : gradient & quiver

- calculate the gradient & draw the vector field(gradient)

MATLAB command “gradient” give the gradient vector.
[GX, GY] = gradient(Z, dx, dy);

Input)
Z : function to be differentiated.
dx : distance of meshgrid X.
dy : distance of meshgrid Y.

Output)
GX : first coordinate of gradient vector.
GY : second coordinate of gradient vector.

In previous example, I used the

```
x = linspace(0, 2*pi, 200);
y = linspace(0, 2*pi, 100);
[X, Y] = meshgrid(x, y);
Z = cos(X).*sin(Y);
```

So dx = $\frac{2\pi}{200}$, dy=$\frac{2\pi}{100}$.
Final code will be
[GX, GY] = gradient$\left(Z, \frac{2\pi}{200}, \frac{2\pi}{100}\right)$

MATLAB command : gradient & quiver
- calculate the gradient & draw the vector field(gradient)

This is the code for draw the gradient vector of $z = \cos(x)\, sin(y)$ on $0 \le x, y \le 2\pi$.

1. Prepare the x vector and y vector.
   x = linspace(0, 2*pi, 200); dx = $\frac{2\pi}{200}$ % x is 1*200 vector.
   y = linspace(0, 2*pi, 100); dy = $\frac{2\pi}{10}$ % y is 1*100 vector.
2. Make the domain points matrices.
   [X, Y] = meshgrid(x ,y); % X and Y are 100*200 matrices.
3. Compute the function values.
   Z = cos(X).*sin(Y); % Z is 100*200 matrix.
4. Calculate the gradient vector.
   [GX, GY] = gradient(Z, dx, dy); % GX and GY are 100*200 matrices.
5. Draw the surface

```
figure(1),
hold on
contour(X, Y, Z)
quiver(X, Y, GX, GY)     % quiver(X, Y, GX, GY) draw the GX on X coordinate
hold off                 %                      and draw the GY on Y coordinate
```

MATLAB command : gradient & quiver

- calculate the gradient & draw the vector field(gradient)

This is the code for draw the gradient vector of $z = \cos(x)\, sin(y)$ on $0 \le x, y \le 2\pi$.

1. Prepare the x vector and y vector.
   x = linspace(0, 2*pi, 200); dx = $\frac{2\pi}{200}$
   y = linspace(0, 2*pi, 100); dy = $\frac{2\pi}{100}$
2. Make the domain points matrices.
   [X, Y] = meshgrid(x ,y);
3. Compute the function values.
   Z = cos(X).*sin(Y);
4. Calculate the gradient vector.
   [GX, GY] = gradient(Z, dx, dy);
5. Draw the surface
   figure(1),
   hold on
   contour(X, Y, Z)
   quiver(X, Y, GX, GY)
   hold off