First part: Basic elements

1.5 3D graphic

1.4.2 Draw the level lines of a function of 2 variables

1.4.3 Represent an equation surface *z*=*g*(*x*,*y*)

1.4.4 Represent a parameterized surface

First part: Basic elements

1.5 3D graphic

1.5.1 Draw the level lines of a function of 2 variables

- **Function**

meshgrid	**(see example)**
mesh	(see example)
meshc	(see example)
meshz	(see example)
contour	(see example)
view	Adjust the viewing angle
grid	Add a grid
xlabel	Add a legend for the abscissa axis
ylabel	Add a legend for the ordinate axis
zlabel	Add a legend for the Z axis
title	Add a title
hold	Add a graph to the current window
figure	Create a new window
plot3	trace point by point a 3D graph
view	Adjust the viewing angle

The outline control allows you to draw the level lines of a function of 2 real variables. This function can be defined by a Matlab expression, or be defined as a user function. To trace the level lines of the G (X, Y) function for xmin <xmax and ymin <ya <ymax we proceed as follows:

creation of a mesh, with a mesh of length h, of the domain [xmin , xmax] x [ymin , ymax] using the command **meshgrid**,
[X,Y] = meshgrid(x_{min}:h:x_{max}, y_{min}:h:y_{max}).

➢ Evaluation of the function at the nodes of this mesh, either by calling the user function defining the function, Z = g(X,Y) or directly by defining the function with a MATLAB expression.

➢ Display of the level lines thanks to the command contour,contour(X,Y,Z).

First part: Basic elements

Example 1

Thus to draw the contour lines of the function $f(x,y) = yxe^{(x^2-y^2)}$ on the domain [-1 , 1] x [-1 , 1] taking a mesh of mesh of length h=0.02:

```
clc; clear all
[X,Y] = meshgrid(-1:0.02:1, -1:0.02:1);
 Z = Y*X.*exp(-X.^2-Y.^2);
 contour(X,Y,Z) , grid on
```

You can also write a user function g.m,

```
function x3 = g(x1,x2)
 x3 = x2* x1.*exp(-x1.^2-x2.^2);
```

and execute

```
>> [X,Y] = meshgrid(-1:0.02:1, -1:0.02:1);

>> Z = g(X,Y);

>> contour(X,Y,Z), grid on
```

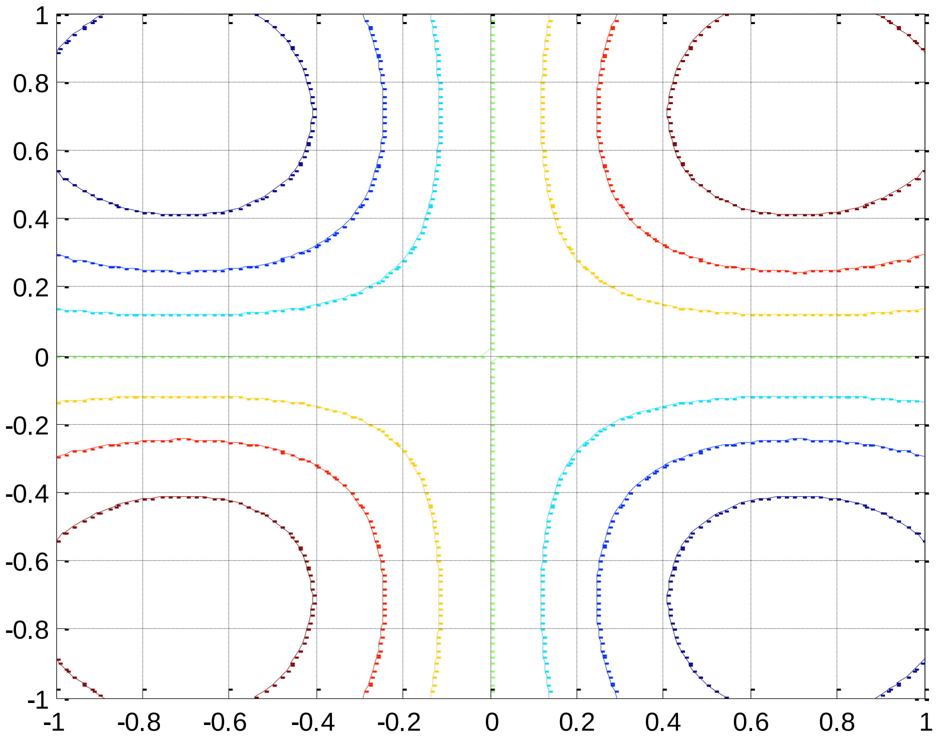

Fig. 1 – Graph of the function f(x,y) .

Example 2

First part: Basic elements

Visualization of level lines using the contour command. The number of level lines is determined automatically from the extreme values taken by the function on the domain considered. To impose the number n of level lines to be displayed, simply call the contour function with the value n as the fourth parameter, contour(X,Y,Z,n). There are two ways to display the contour line values in the figure. If we want to display the values for all level lines, we use the clabel command in the following way:

```
>>                  [C,h]                =                contour(X,Y,Z,n)
>> clabel(C,h)
```

If we want to display only the values of a few level lines, we use the clabel command in the following way :

```
>>                  [C,h]                =                contour(X,Y,Z,n)
>> clabel(C,h,'manual')
```

You can then use the mouse to select the level lines for which you want to display the value.

So to draw 40 lines of level of function $z = (x^2+2)^2 + (x^2 + y)^2$ on the domain [-1 , 1] x [-1 , 1] .

```
[X,Y] = meshgrid(-1:0.02:1, -1:0.02:1);

 Z = (X.^2+2).^2 +(X.^2+Y).^2;

[C,h] = contour(X,Y,Z,40);

 clabel(C,h,'manual')
```

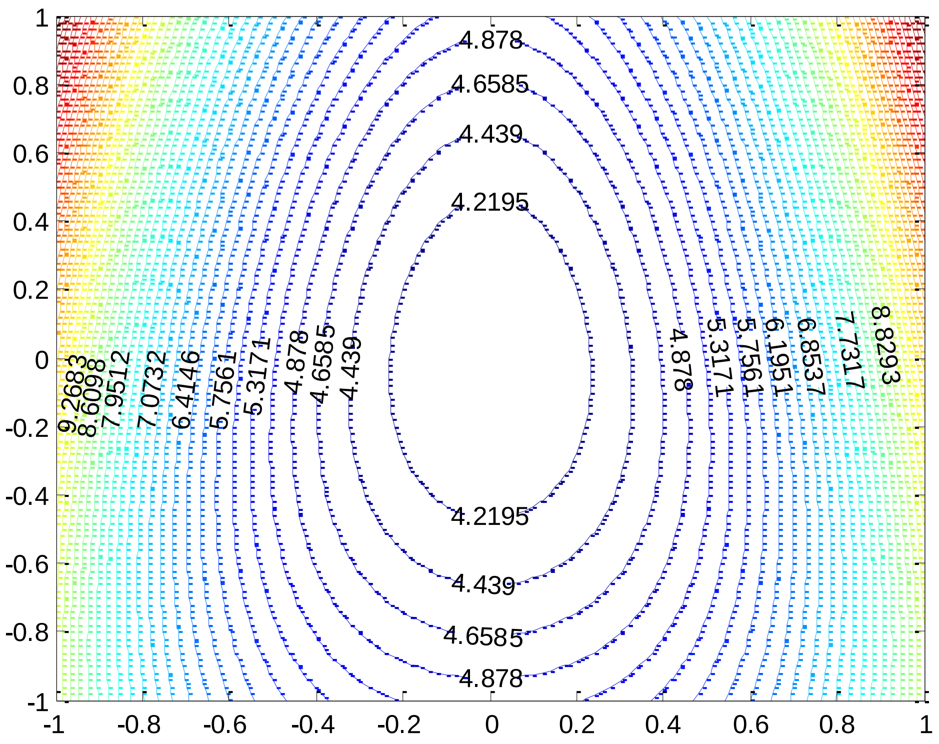

Fig. 2 – Graph of the function z(x,y) .

Example 3

Visualization of level lines using the contour command. It is possible to modify the color palette using the **colormap** command. Typing help graph3d in the MATLAB control window will get you all available color palettes.

The **contourf** command is used in the same way as the contour command. It is used to display, in addition to the level lines, a continuous gradient of colors which varies according to the values taken by the function.

```
[X,Y] = meshgrid(-1:0.02:1, -1:0.02:1);

Z = (X.^2+2).^2 +(X.^2+Y).^2;

 contourf(X,Y,Z,40);

 colormap(cool);
```

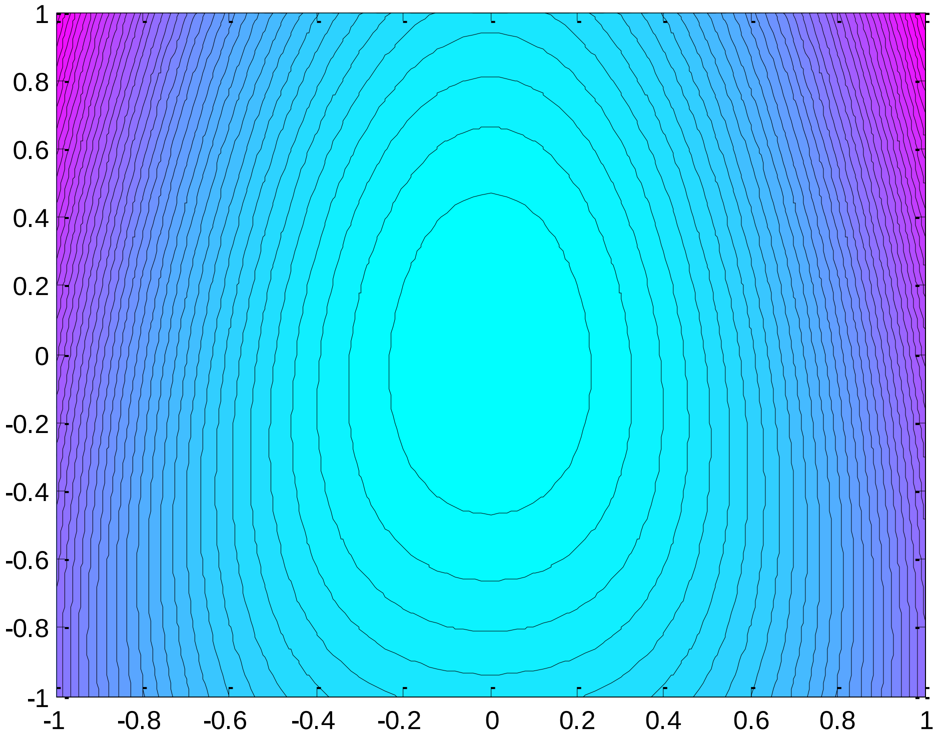

Fig. 3 – Graph of the function z(x,y) .

1.5.2 Represent an equation surface *z*=*g*(*x*,*y*)

The mesh command allows you to draw a surface with the equation z=g(x,y). The function g can be defined directly by a MATLAB expression or be defined as a user function. To plot the surface with equation z=g(x,y) for xmin< x < xmax and ymin< y < ymax we proceed as follows:

- Creation of a mesh, mesh of length H, in the field [xmin, xmax] x [ymin, ymax] thanks to the **Meshgrid** command, [X,Y] =meshgrid(x_{min}:h:x_{max}, y_{min}:h:y_{max}) .
- Evaluation of the function to the nodes of this network, either by calling for the user function defining the function,Z = g(X,Y) either directly by defining the function by an expression MATLAB.
- Surface display thanks to the mesh command,mesh(X,Y,Z).

Example

$z = 2xe^{(-x^2+y^2)}$ on the domain *[-1 , 1]* x *[-1 , 1]* with a mesh of mesh length h=0.02, we execute:

```
clc; clear all

[X,Y] = meshgrid(-1:0.02:1, -1:0.02:1);

 Z = 2*X*Y.*exp(-X.^2+Y.^2-1);

 mesh(X,Y,Z)
```

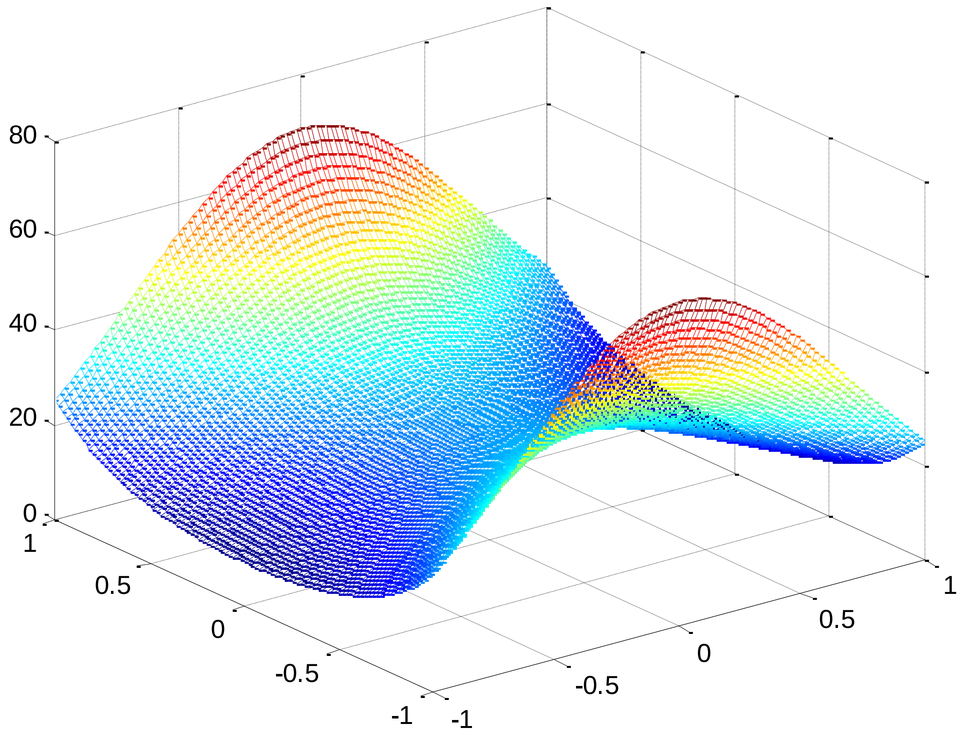

Fig. 4 – Graph of the function z(x,y) .

1.5.3 Represent a parameterized surface

The **surf** control allows you to draw a parameterized surface of equations

First part: Basic elements

$$\begin{cases} x = f_1(u,v) \\ y = f_2(u,v) \\ z = f_3(u,v) \end{cases}$$

The f= function (f1, f2, f3) can be defined directly by a MATLAB expression or be defined as a user function. To plot the parameterized area of equation (E) for umin< u < umax and vmin< v < vmax we proceed as follows:

- creating a mesh, h-length mesh, domain [xmin , xmax] x [ymin , ymax] using the command **meshgrid**, [U,V] = meshgrid(u_{min}:h:u_{max}, v_{min}:h:v_{max}). Evaluation of the function to the nodes of this network, either by calling for the user function defining the function, [x, y, z] = f (u, v) either directly by defining the function by a Matlab expression.
- Surface display thanks to the surf control, surf(X,Y,Z).

Example 1

Thus to draw the configured surface of equations:

$$\begin{cases} x = vcos(u) \\ y = v(u) \\ z = 2v \end{cases}$$

on the domain [0 , 2pi] x [0 , 2] with a mesh mesh of length h=0.05, we execute:

```
clc; clear all

[U,V] = meshgrid(0:0.05:2*pi, 0:0.05:2);

 X = 2*V.*cos(2*U);

 Y = V.*sin(2*U);

 Z = 2*U.^2+V.^2;

 surf(X,Y,Z)
```

Si la fonction est définie comme une fonction utilisateur dans le fichier G1.m,

```
function [x1, x2, x3] = G1(u,v)

x1 = 2*v.*cos(2*u);
x2 = v.*sin(2*u);
x3 = 2*u;
```

On exécute:

```
>> [U,V] = meshgrid(0:0.05:2*pi, 0:0.05:2);

>> [X,Y,Z] = G1(U,V);
>> surf(X,Y,Z)
```

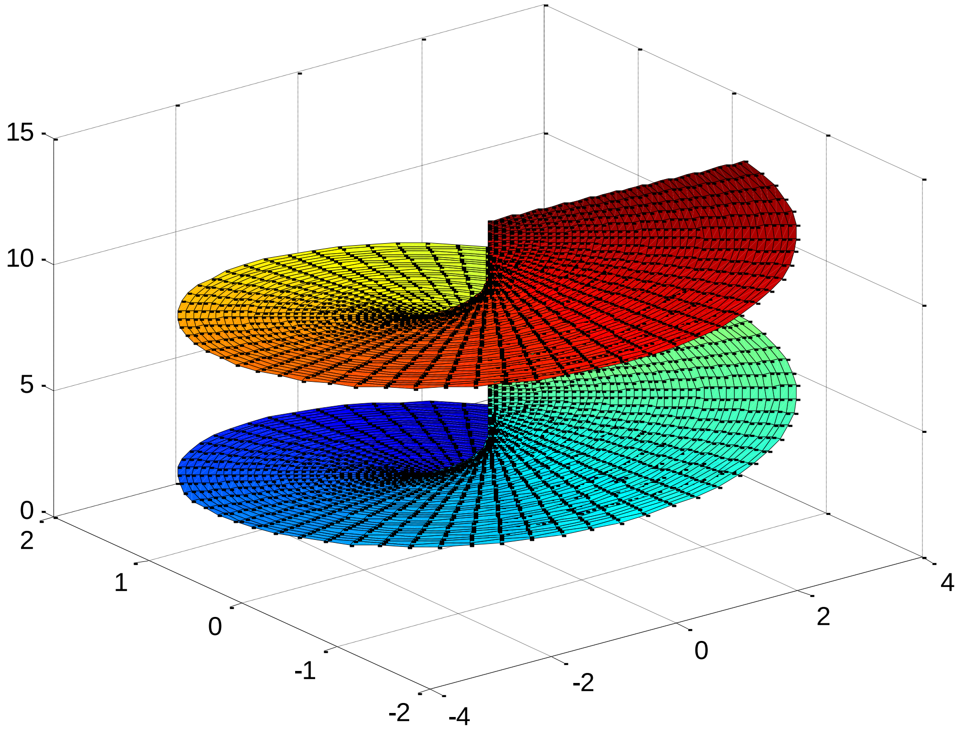

Fig. 5 – surface d'équations.

Example 2

Either parametric equation: $\begin{cases} x = cos(t) \\ y = 2tsin(t) \\ z = 5t \end{cases}$

```
clc; clear all
t = 0 : pi/200 : 10*pi;
x = t.*cos(t);
y = 2*t.*sin(t);
z = 5*t+1;
plot3( cos(t) , 2*t.*sin(t) , 5*t),grid on
```

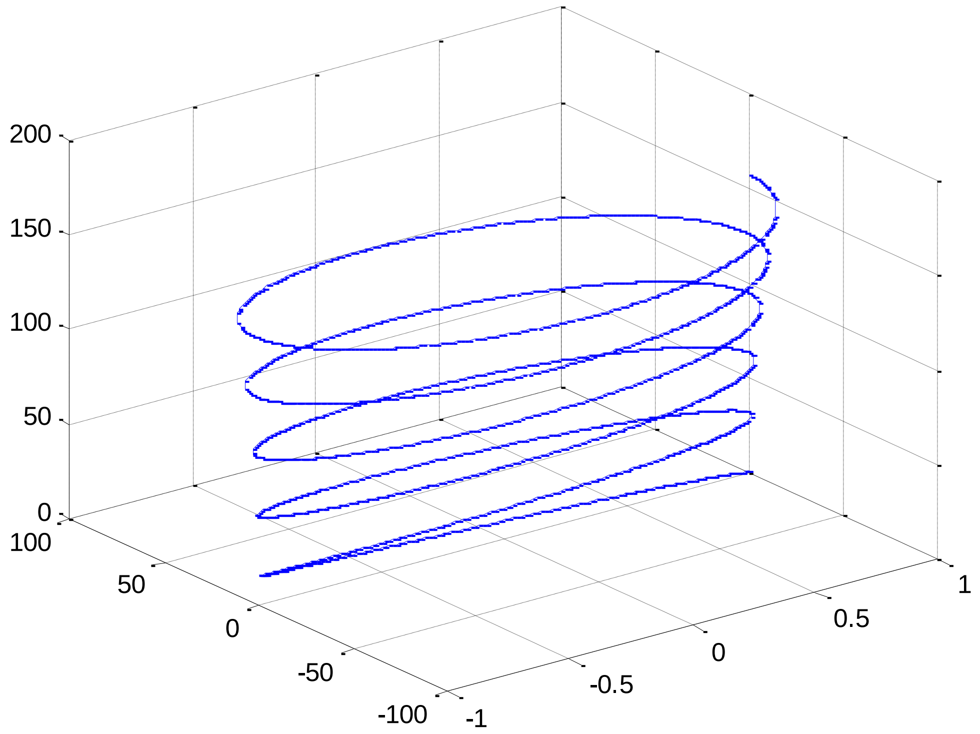

Fig. 6 – surface d'équations.

Example 3

$$z = f(x, y) = \sqrt{5x^2 + y^2}$$

```
clc; clear all
   x = -80 : 80
   y = -80 : 80
   [X , Y] = meshgrid(x , y);
   Z = sqrt (5.*X.^2 + Y.^2);
   mesh (X , Y , Z)
   grid on
```