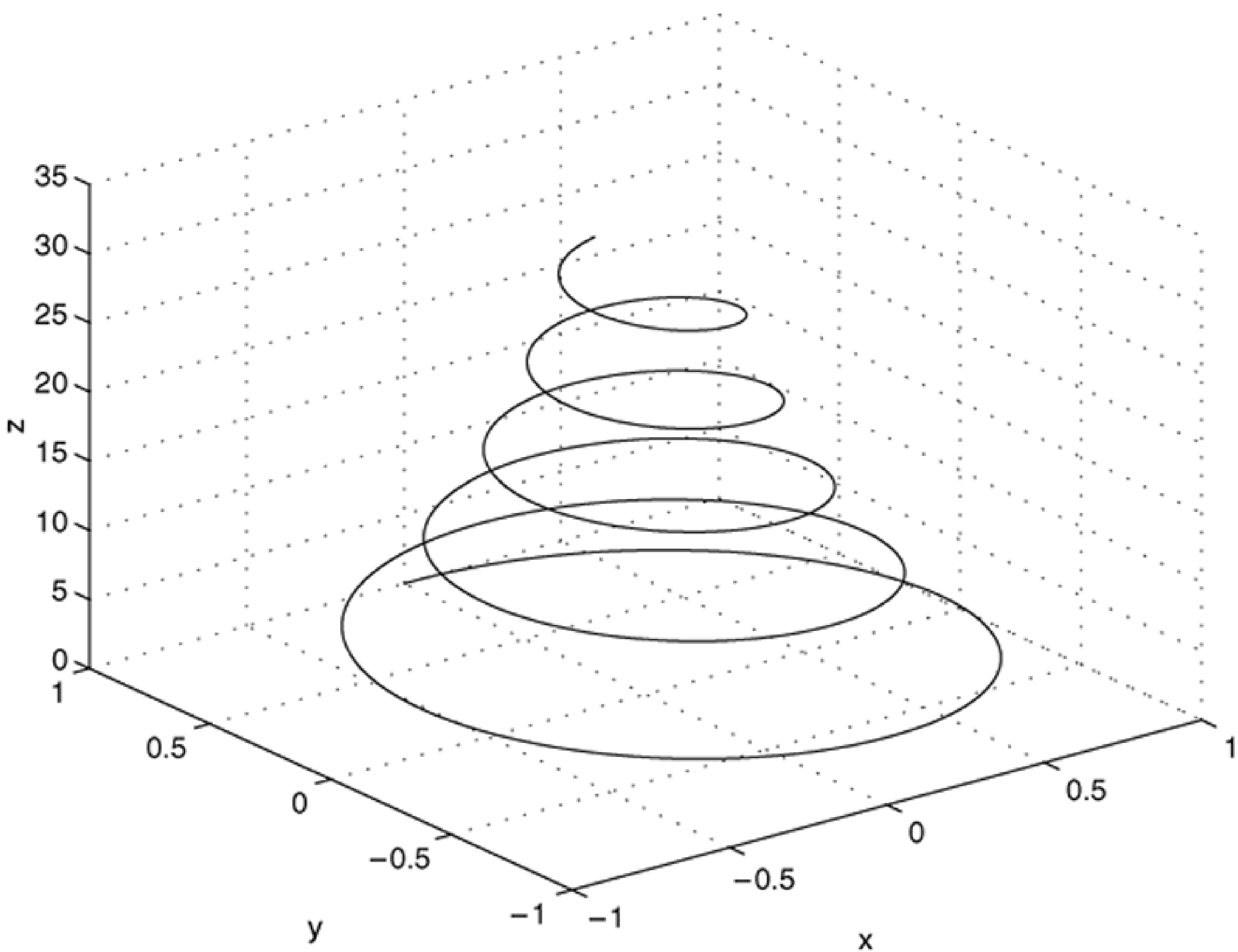
Three-Dimensional Line Plots:

The following program uses the `plot3` function to generate the spiral curve shown in Figure 5.4–1, page 247.

```
>>t = 0:pi/50:10*pi;
>>plot3(exp(-0.05*t).*sin(t),...
  exp(-0.05*t).*cos(t),t),...
  xlabel('x'),ylabel('y'),zlabel('z'),grid
```

See the next slide.

The curve $x = e^{-0.05t} \sin t$, $y = e^{-0.05t} \cos t$, $z = t$ plotted with the `plot3` function. Figure 5.4–1, page 247.



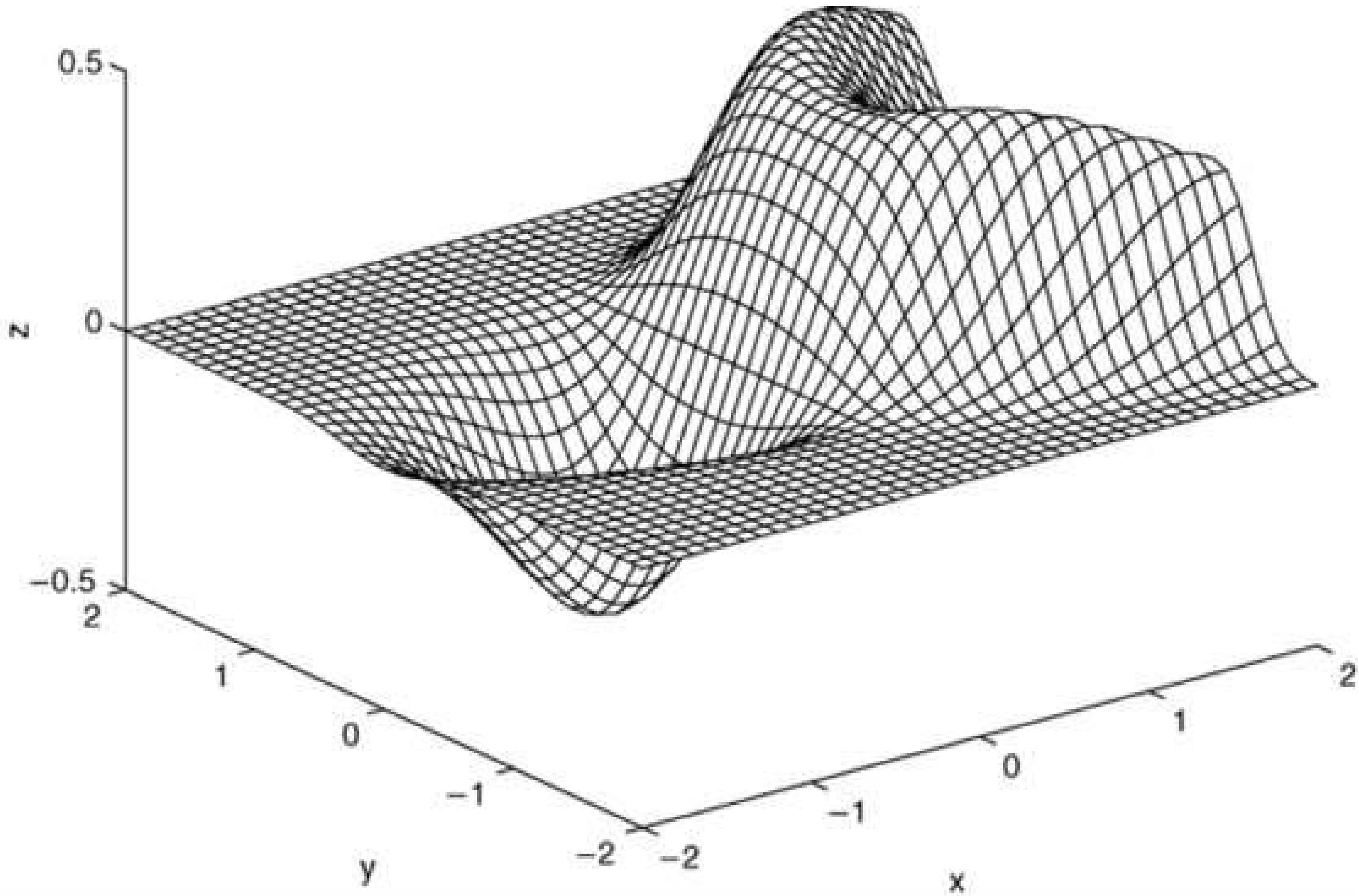
Surface Plots:

The following session shows how to generate the surface plot of the function $z = xe^{-[(x-y^2)^2+y^2]}$, for $-2 \le x \le 2$ and $-2 \le y \le 2$, with a spacing of 0.1. This plot appears in Figure 5.4–2, page 248.

```
>>[X,Y] = meshgrid(-2:0.1:2);
>>Z = X.*exp(-((X-Y.^2).^2+Y.^2));
>>mesh(X,Y,Z),xlabel('x'),ylabel('y'),...
   zlabel('z')
```

See the next slide.

A plot of the surface $z = xe^{-[(x-y^2)^2+y^2]}$ created with the `mesh` function. Figure 5.8–2

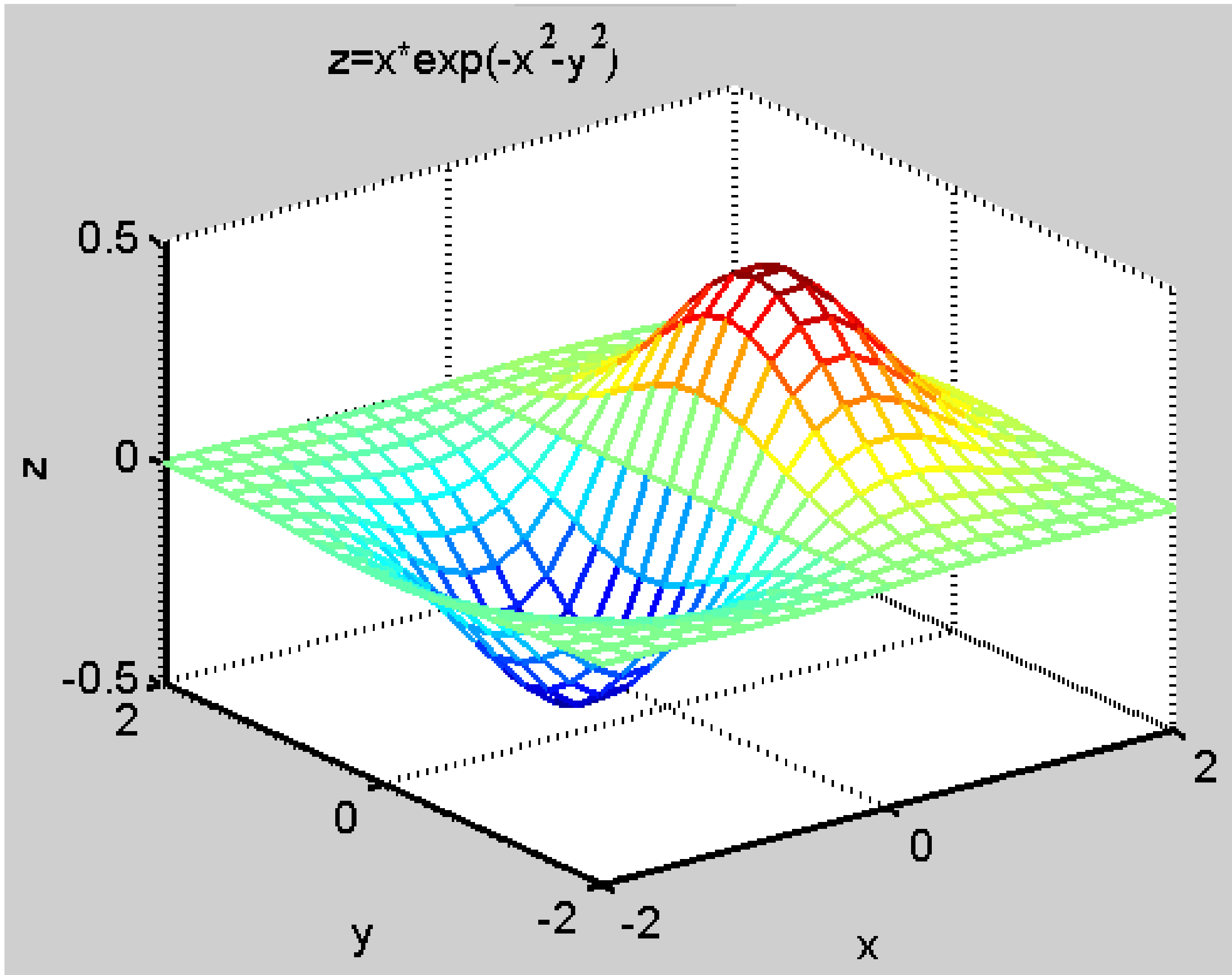


Example

```
clear ; clf ;
xa = -2:0.2:2;
ya = -2:0.2:2;
[x,y] = meshgrid(xa,ya) ;
z = x.*exp(-x.^2-y.^2) ;
mesh(x,y,z)
title('z=x*exp(-x^2-y^2)')
xlabel('x') ; ylabel('y') ; zlabel('z') ;
```

See the next slide.

A contour plot of the surface $z = xe^{-[(x-y^2)^2+y^2]}$ created with the `contour` function.



Contour

Contour of a function in a two-dimensional array can be plotted by:

```
Contour(x , y , z , level)
```

z is the 2-D array of the function

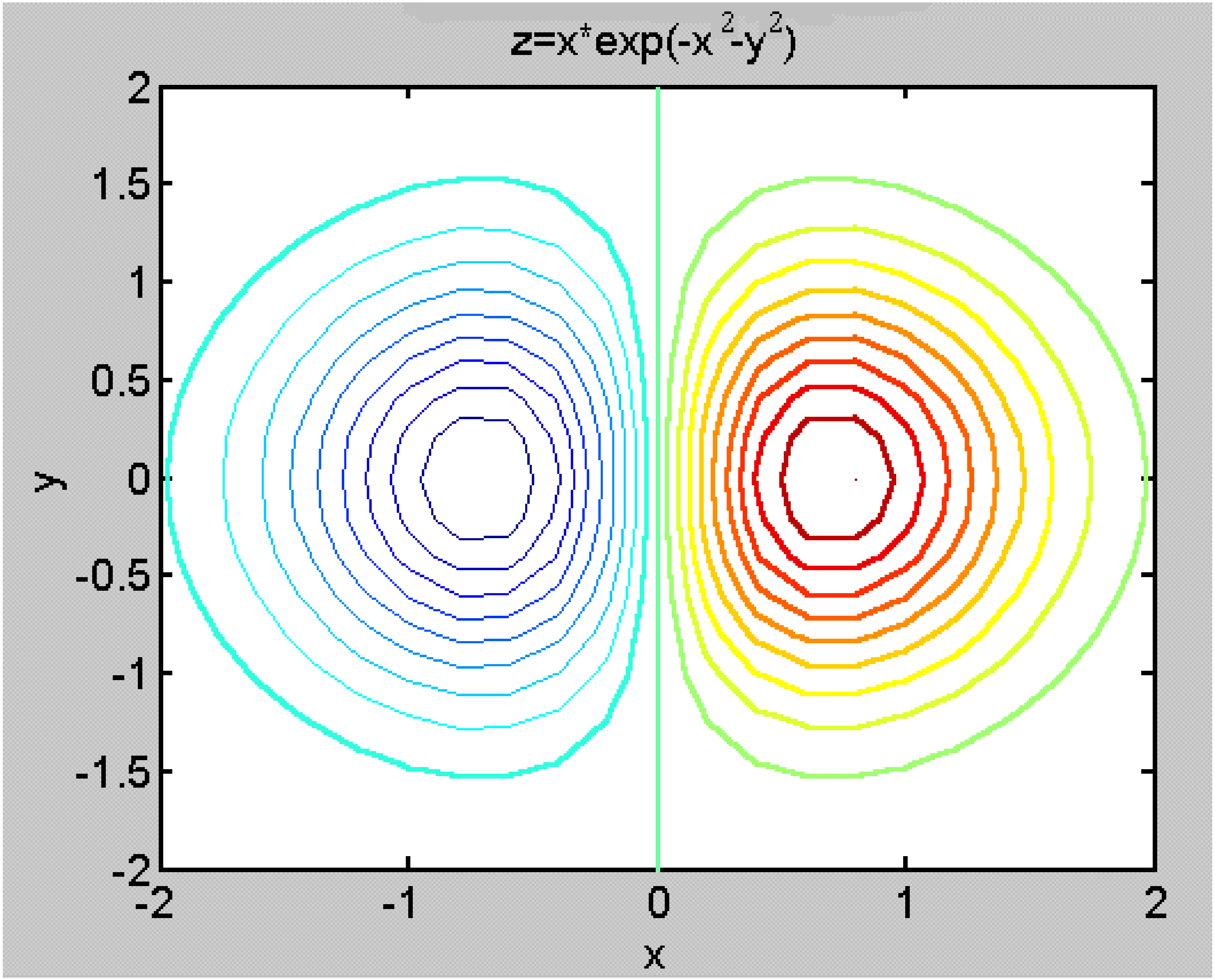
x and y are the coordinates in one or 2D arrays

Level is a vector containing the levels

The contour level is determined by dividing the minimum and maximum values of z into k-1 intervals

The following is a script for the contour plot:

```
clear ; clf ;

xa = -2 : 0.2 : 2  ;  ya = -2 : 0.2 : 2 ;

[x,y] = meshgrid(xa,ya) ;

z = x.*exp(-x.^2-y.^2) ;  zmax=max(max(z)) ;

zmin=min(min(z)) ; k=21 ;

dz=(zmax-zmin)/(k-1) ; level=zmin : dz : zmax ;
h=contour(x , y , z , level) ;

title('z=x*exp(-x^2-y^2)') ;

xlabel('x') ; ylabel('y') ;
```

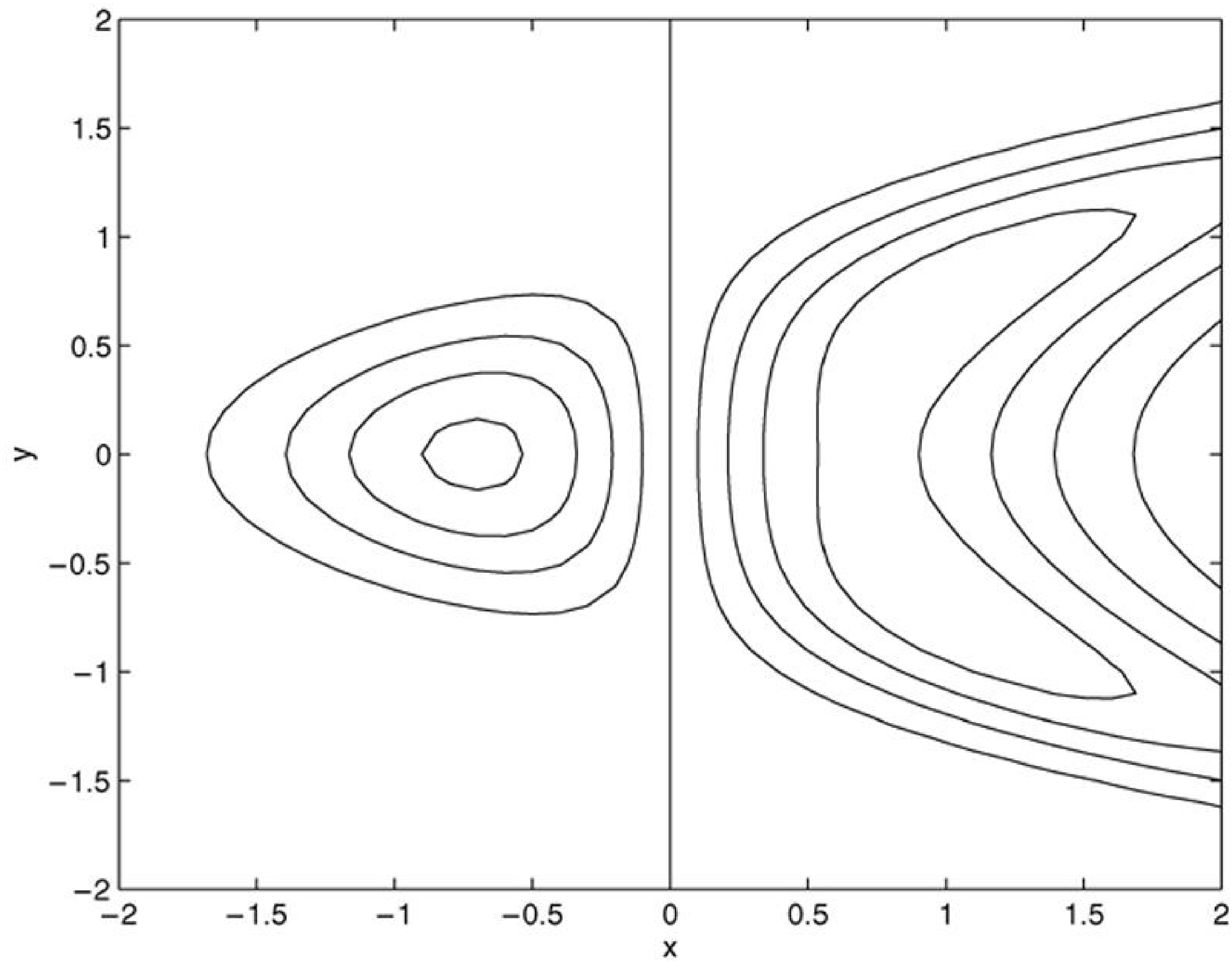

z=x*exp(-x^2-y^2)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
y
-2
-1
0
1
2
x

The following session generates the contour plot of the function whose surface plot is shown in Figure 5.8–2; namely, $z = xe^{-[(x-y^2)^2+y^2]}$, for $-2 \le x \le 2$ and $-2 \le y \le 2$, with a spacing of 0.1. This plot appears in Figure 5.4–3, page 249.

```
>>[X,Y] = meshgrid(-2:0.1:2);
>>Z = X.*exp(-((X- Y.^2).^2+Y.^2));
>>contour(X,Y,Z),xlabel('x'),ylabel('y')
```

See the next slide.

A contour plot of the surface $z = xe^{-[(x-y^2)^2+y^2]}$ created with the `contour` function.



Contour labels may be automatically annotated by : clabel(h)

h is the name of contour

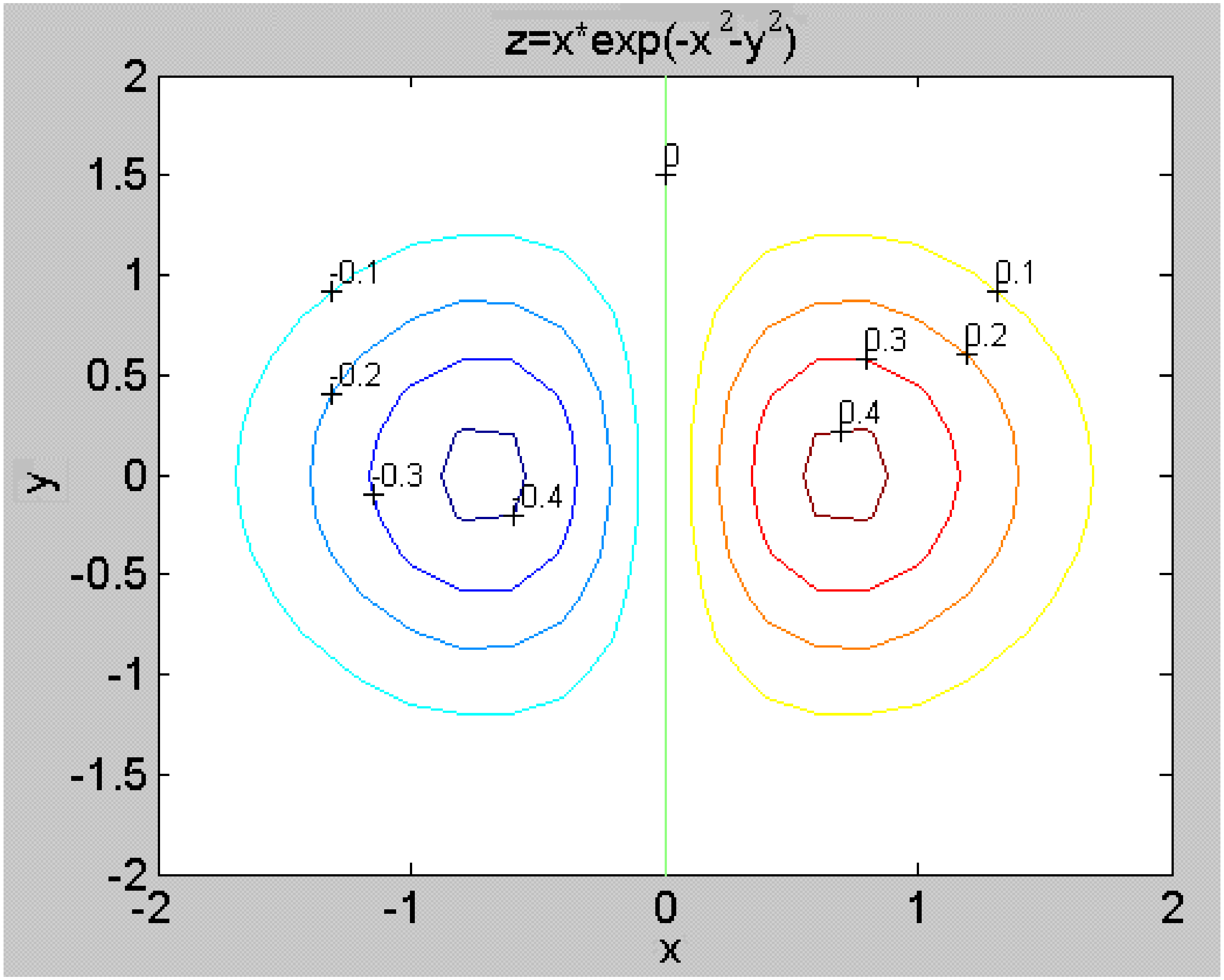
Contour labels may also be placed manually using the mouse and by :

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

Below is a script with clabel :

Example

```
clear ; clf ;

xa = -2:0.2:2; ya = -2:0.2:2;

[x,y] = meshgrid(xa,ya) ;

z = x.*exp(-x.^2-y.^2)  ;  zmax=0.5 ;

zmin=-0.5 ; k=11 ;

dz=(zmax-zmin)/(k-1) ;  level=zmin : dz :
zmax ; h=contour(x,y,z,level) ;
clabel(h,'manual') ;

title('z=x*exp(-x^2-y^2)')

xlabel('x') ;  ylabel('y')
```

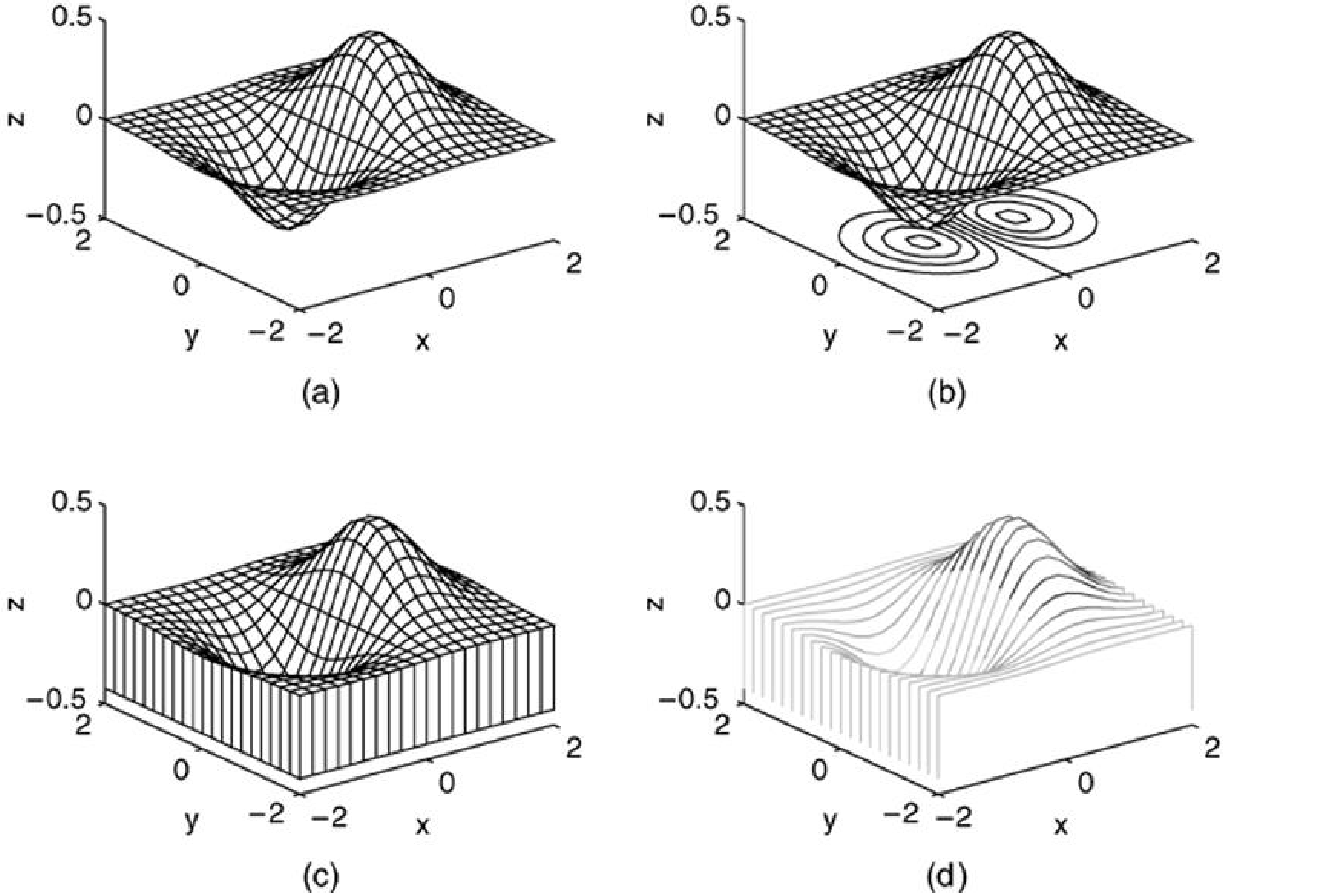

z=x*exp(-x 2-y 2)
y
x
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2
-0.1
-0.2
-0.3
-0.4
0
0.1
0.2
0.3
0.4

Three-dimensional plotting functions. Table 5.4–1, page 250.

Function	Description
`contour(x,y,z)`	Creates a contour plot.
`mesh(x,y,z)`	Creates a 3D mesh surface plot.
`meshc(x,y,z)`	Same as `mesh` but draws contours under the surface.
`meshz(x,y,z)`	Same as `mesh` but draws vertical reference lines under the surface.
`surf(x,y,z)`	Creates a shaded 3D mesh surface plot.
`surfc(x,y,z)`	Same as `surf` but draws contours under the surface.
`[X,Y] = meshgrid(x,y)`	Creates the matrices `X` and `Y` from the vectors `x` and `y` to define a rectangular grid.
`[X,Y] = meshgrid(x)`	Same as `[X,Y]=meshgrid(x,x)`.
`waterfall(x,y,z)`	Same as `mesh` but draws mesh lines in one direction only.

Plots of the surface $z = xe^{-(x^2+y^2)}$ created with the `mesh` function and its variant forms: `meshc`, `meshz`, and `waterfall`. **a) `mesh`, b) `meshc`, c) `meshz`, d) `waterfall`.**

Figure 5.4–4, page 250.



Vector plot

Quantities at grid points sometimes are required to be plotted in a vector form

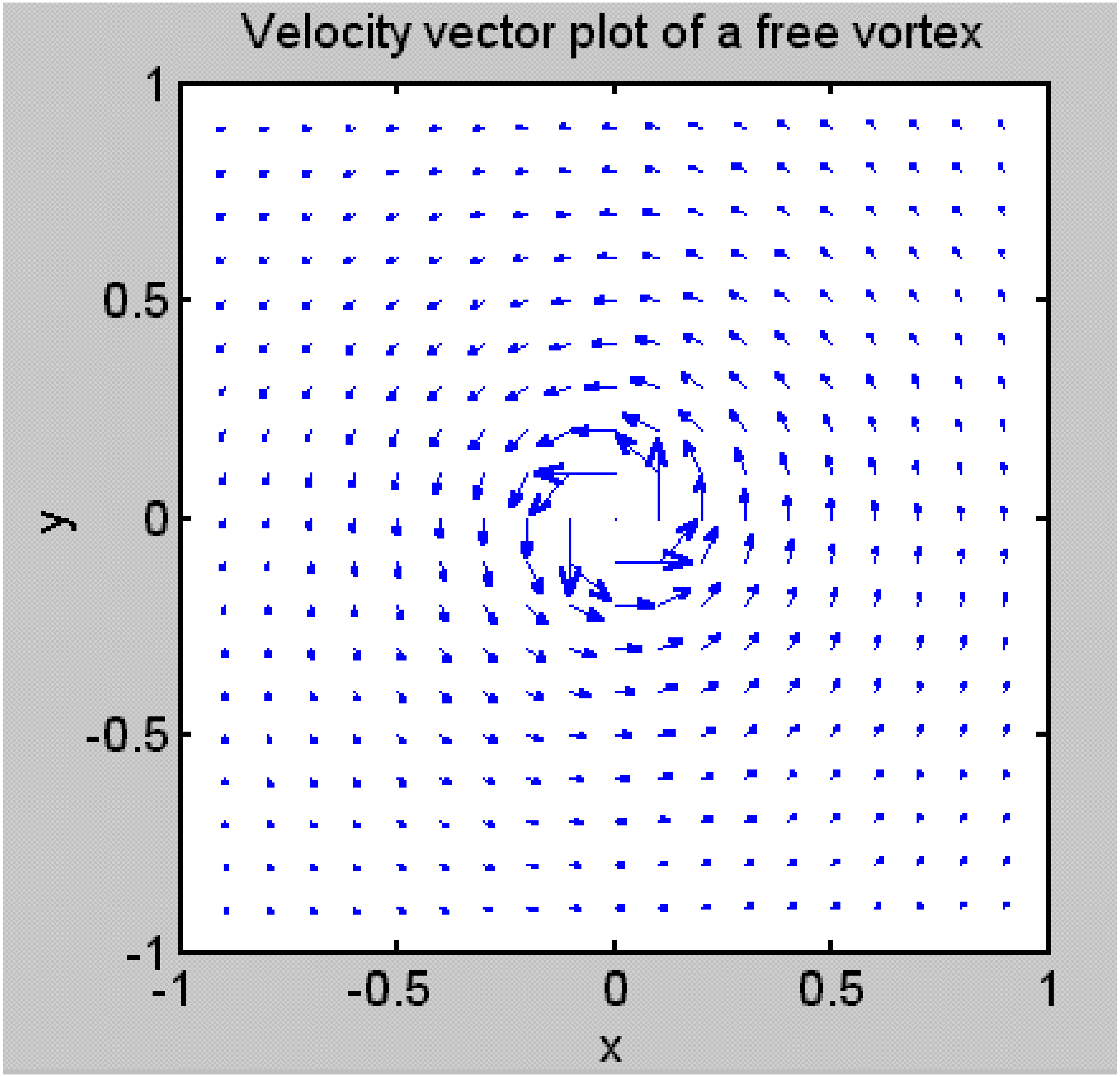
The vectors at grid points may be plotted by the **quiver** command:

```
quiver(x , y , u , v , s)
```

x	Array for x-coordinate
y	Array for y-coordinate
u	Array of the function u
v	Array of the function v
s	Scale factor for vector adjustment

```
clear , clf
xmax= 0.9 ; xmin= -xmax ; dx= 0.1 ;
ymax= xmax ; ymin= -ymax ; dy= dx ;
xm = xmin : dx : xmax   ;
ym = ymin : dy : ymax  ;
[x,y] = meshgrid(xm,ym) ;
u = -0.1*(2*y)./(x.^2+y.^2)  ;
v = -0.1*(-2*x)./(x.^2+y.^2) ;
quiver(x,y,u,v,1.5) ;
title('Velocity vector plot of  free vortex')
xlabel('x') ; ylabel('y')
axis('square')
```

This script is
an example of
a vector plot
for a free vortex
plane potential
flow


Velocity vector plot of a free vortex
1
0.5
0
-0.5
-1
y
-1
-0.5
0
0.5
1
x

Polynomial fit to a given data

A polynomial is required to fit the following data:

x	y
1.1	3.887
2.3	4.276
3.9	4.651
5.1	2.117

The required script for fitting a polynomial and plotting the data is given in the next sheet

```
clear , clf , hold off
x= [1.1,2.3,3.9,5.1] ;
y= [3.887,4.276,4.651,2.117] ;
a= polyfit(x,y,length(x)-1)
xi= 1.1:0.1:5.1 ;
yi= polyval(a,xi) ;
plot(xi,yi,'r')
text(4.2,4.5,'fit','fontsize',[14],'color','r');
xlabel('x') ; ylabel('y')
hold on
plot(x,y,'--',x,y,'+b')
```