3.3 Surfaces in Matlab

In this section we add another dimension and look at the plots of surfaces in three-space. We've already spent time in two-space, drawing graphs in the Cartesian plane pictured in **Figure 3.1**(a). In the image in **Figure 3.1**(a), we use arrows and labeling to indicate the positive direction on each axis.

To add a third dimension, we need to add another axis. It is not required to make the axes orthogonal (perpendicular), but things are much easier if each pair of axes is orthogonal. So we create third axis, which we will call the z-axis, that is orthogonal to the plane containing the x- and y-axes. To determine the positive direction, we use what is known as the *right-hand rule*. If we cup the fingers of our right hand from the x- to the y-axis, then our thumb will point in the positive direction on the z-axis (out of the printed page).

It is difficult to draw three-dimensional images on a two dimensional page or screen. In **Figure 3.1**(b), note that we've made an attempt to picture the z-axis. In this image, it is our intent to deliver the message that the z-axis is perpendicular to the xy-plane and emanates from the origin directly out of the page.

As long as we maintain the right-hand rule to orient our axes, we can depict our coordinate system in a number of different ways. The orientation we prefer to use in Matlab is shown in **Figure 3.1**(c), where the positive x-axis emanates outward from the printed page. Note that if we cup the fingers of our right hand from the x-axis to the y-axis, the thumb points upward in the direction of the positive z-axis.

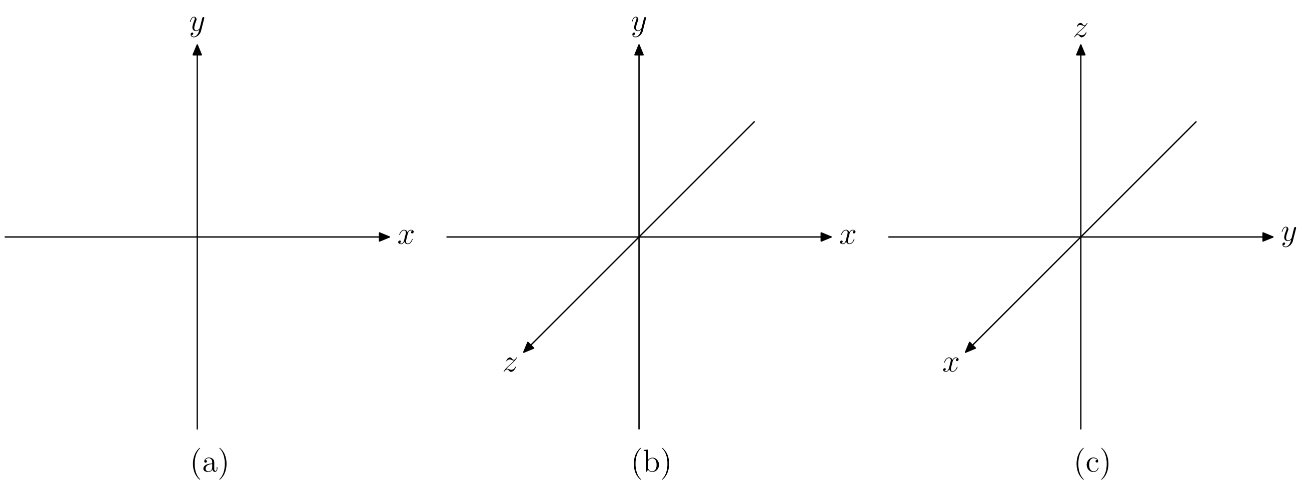


Figure 3.1. Cartesian coordinate systems obey the right-hand rule.

[1] Copyrighted material. See: http://msenux.redwoods.edu/Math4Textbook/

Plotting Functions of Two Variables

We will now begin the task of plot a function of two variables. In the same way as an equation $y = f(x)$ leads to the plotting of ordered pairs (x, y) in the Cartesian plane, the equation $z = f(x, y)$ will lead to the plotting of ordered triplets (x, y, z) in three-space. Consider, for example, the function

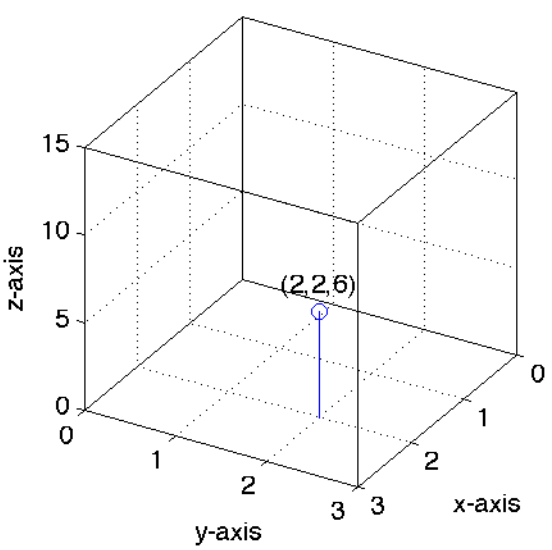
$$f(x, y) = 12 - x - 2y.$$

If we evaluate this function at the point $(x, y) = (2, 2)$, then

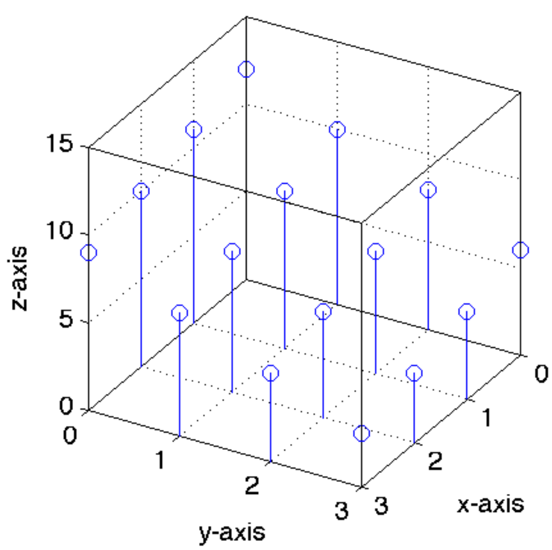
$$z = f(2, 2) = 12 - 2 - 2(2) = 6.$$

This leads to the ordered triplet $(x, y, z) = (2, 2, 6)$, which we plot in three-space, as shown in **Figure 3.2**(a).

Now, suppose that we form all possible ordered pairs (x, y), where x and y are chosen from the set $\{0, 1, 2, 3\}$. We obtain a *grid* of (x, y) pairs. Next, evaluate the function $z = f(x, y) = 12 - x - 2y$ at each (x, y) pair in this grid, then plot the resulting triplets (x, y, z), as shown in **Figure 3.2**(b).



(a)



(b)

Figure 3.2. Ploting points (x, y, z) that satisfy $z = f(x, y) = 12 - x - 27$.

If we connect each of the points in **Figure 3.2**(b) with line segments (in the same way as we connect points plotted with Matlab's **plot** command), we get the surface shown in **Figure 3.3**.

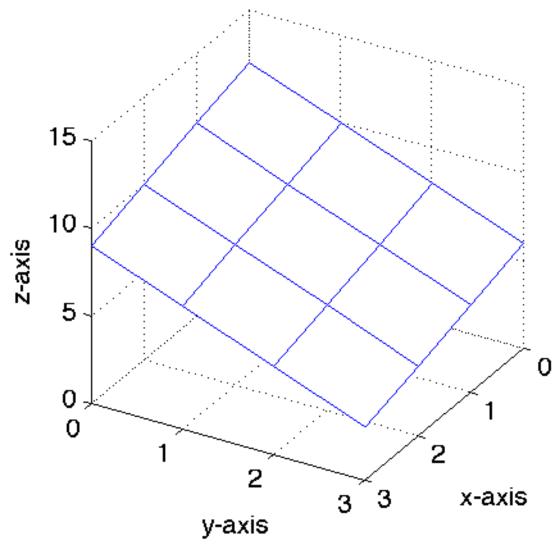


Figure 3.3. The set of points satisfying the equation $z = f(x) = 12 - x - 2y$ is a plane.

Creating the Grid. Note that in creating the image in **Figures 3.2**(b) and **3.3**, we evaluated the function $z = f(x, y) = 12 - x - 2y$ at each point of the grid shown in **Table 3.1**(a). The z-results are shown in **Table 3.1**(b).

$(0, 0)$	$(1, 0)$	$(2, 0)$	$(3, 0)$
$(0, 1)$	$(1, 1)$	$(2, 1)$	$(3, 1)$
$(0, 2)$	$(1, 2)$	$(2, 2)$	$(3, 2)$
$(0, 3)$	$(1, 3)$	$(2, 3)$	$(3, 3)$

(a)

12	11	10	9
10	9	8	7
8	7	6	5
6	5	4	3

(b)

Table 3.1. Evaluate $z = f(x, y) = 12 - x - 2y$ at each pair (x, y) in a rectangular grid.

Therefore, the question arises. How do we create a grid of (x, y) pairs in Matlab?

Because both x and y range from 0 to 3 by ones, the following command will create the grid we need.

```
>> [x,y]=meshgrid(0:3);
```

First, examine the matrix x. Note that each row of the matrix x contains the entries 0, 1, 2, and 3.

```
>> x
x =
     0     1     2     3
     0     1     2     3
     0     1     2     3
     0     1     2     3
```

Next, examine the matrix y. Note that each column contains the entries 0, 1, 2, and 3.

```
y =
     0     0     0     0
     1     1     1     1
     2     2     2     2
     3     3     3     3
```

Now, imagine superimposing the matrix y atop the matrix x. If you can hold this image in your mind, then you have a picture of the ordered pairs in **Table 3.1**(a).

It is now a simple matter to evaluate the function $z = f(x, y) = 12 - x - 2y$ at each ordered pair (x, y). Note that the following z-values agree with those found in **Table 3.1**(b).

```
>> z=12-x-2*y
z =
    12    11    10     9
    10     9     8     7
     8     7     6     5
     6     5     4     3
```

Finally, to produce the plane shown in **Figure 3.3**, execute the following commands.

```
>> mesh(x,y,z)
>> view(125,30)
>> xlabel('x-axis')
>> ylabel('y-axis')
>> zlabel('z-axis')
```

The commands **xlabel**, **ylabel**, and **zlabel** are used to label the x-, y-, and z-axes respectively. The **view** command sets the angle of the view from which an observer sees the current three-dimensional plot. The syntax is **view(ax,el)**, where **az** is the azimuth or horizontal rotation of the xy-plane, and **el** is the vertical elevation, both measured in degrees. By rotating the horizontal plane through 125 degrees, we bring it into an orientation that approximates that shown in **Figure 3.1**(c). Alternatively, the user can execute the command **mesh(x,y,z)**, then click the **Rotate 3D** icon on the toolbar, then use the mouse interactively to rotate the surface into an orientation resembling that in **Figure 3.3**.

It's not necessary that z be expressed as a function of x and y. You could have x as a function of y and z, or y as a function of x and z. Let's look at an example.

▶ **Example 1.** *Sketch the graph of $y = 1$ in three-space.*

Here, thought not explicit, we assume that y is a function of x and z. That is, we interpret $y = 1$ to mean $y = f(x, z) = 1$.

Because y is a function of x and z, we will first create a grid of points in the xz-plane. We arbitrarily let x run from -2 to 2 in increments of 1, and z run from -3 to 3 in increments of 1.

```
>> [x,z]=meshgrid(-2:2,-3:3);
```

It is again instructive to view the contents of matrix x. Note that each row of matrix x runs from -2 to 2 in increments of 1.

```
>> x
x =
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
```

Similarly, each column of z runs from -3 to 3 in increments of 1.

```
>> z
z =
    -3    -3    -3    -3    -3
    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1
     0     0     0     0     0
     1     1     1     1     1
     2     2     2     2     2
     3     3     3     3     3
```

It is important to note that the **meshgrid** command always creates two matrices of equal size. In this case, both x and z have dimensions 7×5.

Superimposing (mentally) matrix z atop matrix z gives a vision of (x, z) pairs. We must now evaluate the function $y = f(x, z) = 1$ at each pair. This is easy to do mentally, because the resulting y-value is always equal to 1, no matter what (x, z) pair you use. Thus, what we need Matlab to do is create a matrix of all ones having the same size as the matices x and z, and store the result in the variable y.

```
>> y=ones(size(x))
y =
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

Note that the command **y=ones(size(z))** would have worked equally well.

All that remains is to plot the surface with Matlab's **mesh** command. The following commands will produce the plane shown in **Figure 3.4**.

```
>> mesh(x,y,z)
>> axis tight
>> view(120,30)
>> xlabel('x-axis')
>> ylabel('y-axis')
>> zlabel('z-axis')
```

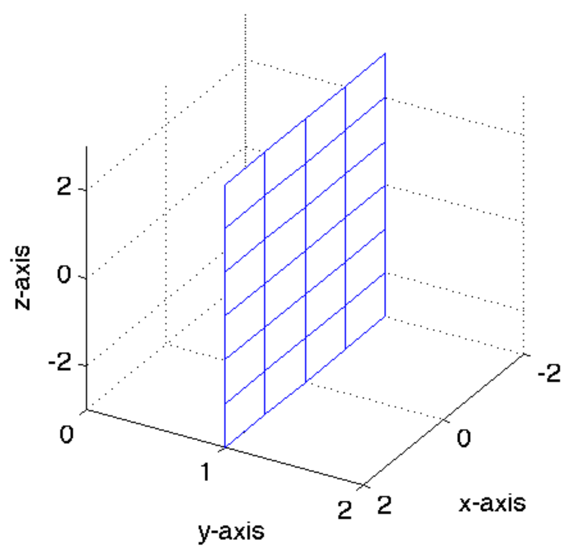


Figure 3.4. The graph of $y = 1$ in three-space.

A Bit More Interesting

We can draw a lot more interesting surfaces than simple planes with Matlab's **mesh** command. Let's look at some examples.

► **Example 2.** *Sketch the graph of* $z = f(x, y) = 9 - x^2 - y^2$ *over the rectangular domain* $D = \{(x, y) : -3 \leq x, y \leq 3\}$.

When sketching the graph of a surface that curves in space, you will again want to avoid the "Jaggies" by plotting enough points. So, let's try letting both x and y run from -3 to 3 in increments of 0.1.

```
>> [x,y]=meshgrid(-3:.1:3);
```

Evaluate the function at each point of the grid.

```
z=9-x.^2-y.^2;
```

We use of array exponentiation because we don't want to square the matrices x and y, we want to square each element of the matrices x and y. Draw the surface with the **mesh** command.

```
>> mesh(x,y,z)
```

Finally, rotate the surface and add axis labels with the following commands. The result is shown in **Figure 3.5**.

```
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```

By default, Matlab uses color to indicate height. You can add a colorbar to your plot with Matlab's **colorbar** command.

```
>> colorbar
```

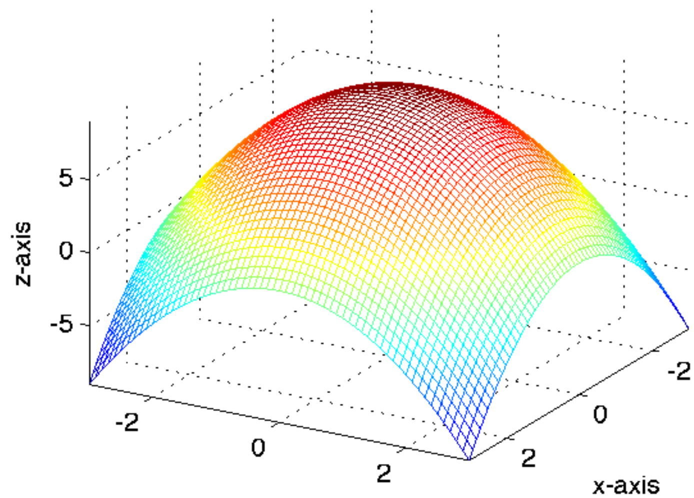


Figure 3.5. The graph of
$z = f(x, y) = 9 - x^2 - y^2$ in three-space.

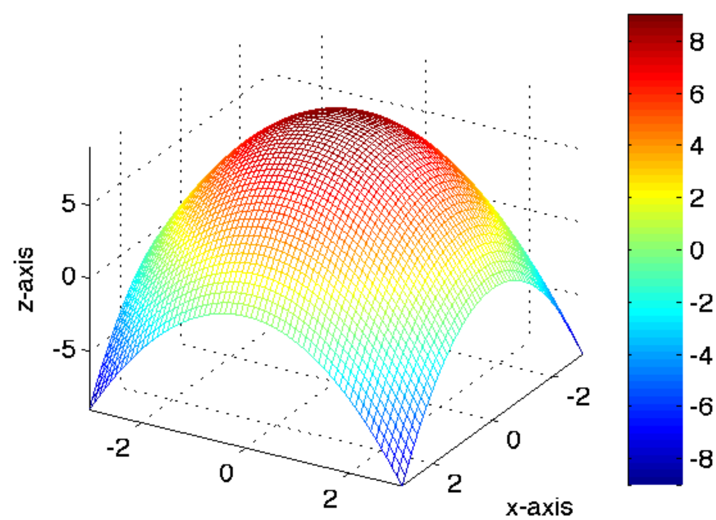


Figure 3.6. A colorbar correlates
height (z-direction) and color.

In **Figure 3.6**, you see that a colorbar has been added to the figure as a result of the last command.

The colors toward the blue end of the color map indicate low heights (z-wise) and the colors at the red end of the color map indicate high heights (z-wise). Inbetween, there is a gradual gradient from blue to red. Note that the highest points on the surface are red and the lowest points on the surface are blue. Other points on the surface are colored according to their z-height and the correlation indicated by the colorbar.

Let's look at another example.

▶ **Example 3.** *Sketch the surface defined by* $z = f(x, y) = (x^2 - y^2)e^{-x^2-y^2}$ *on the domain* $D = \{(x, y) : -3 \leq x, y \leq 3\}$.

Create the grid as in the previous example.

```
>> [x,y]=meshgrid(-3:0.1:3);
```

Remember to use array operators when calculating z.

```
>> z=(x.^2-y.^2).*exp(-x.^2-y.^2);
```

The **mesh** command provides the surface shown in **Figure 3.7**.

```
>> mesh(x,y,z)
```

Of course, you will want to label the axes and add a title.

```
>> xlabel('x-axis')
>> ylabel('y-axis')
>> zlabel('z-axis')
>> title('The graph of z = (x^2-y^2)e^{-x^2-y^2}.')
```

Sometimes you would like to add a bit of transparency to your plot, allowing the user to "see through" the mesh for details that are nomally hidden from view. Matlab's **hidden off** command provides this capability. The following command provides the image in **Figure 3.8**.

```
>> hidden off
```

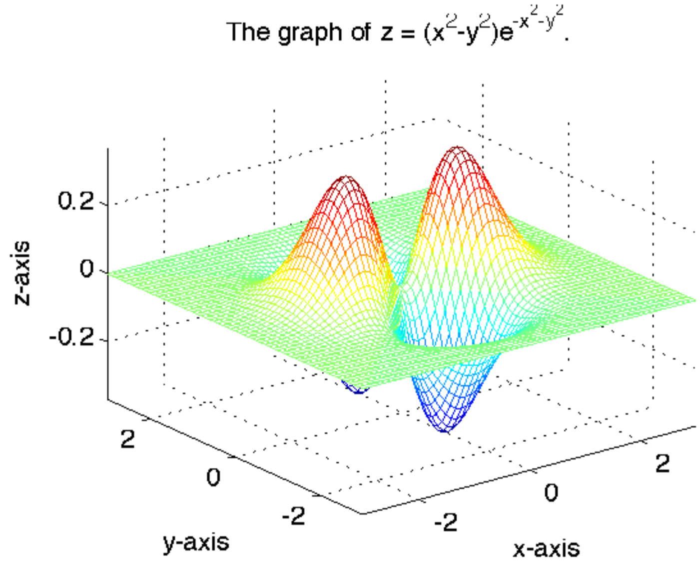


Figure 3.7. The surface defined by the function $z = f(x, y) = (x^2 - y^2)e^{-x^2-y^2}$.

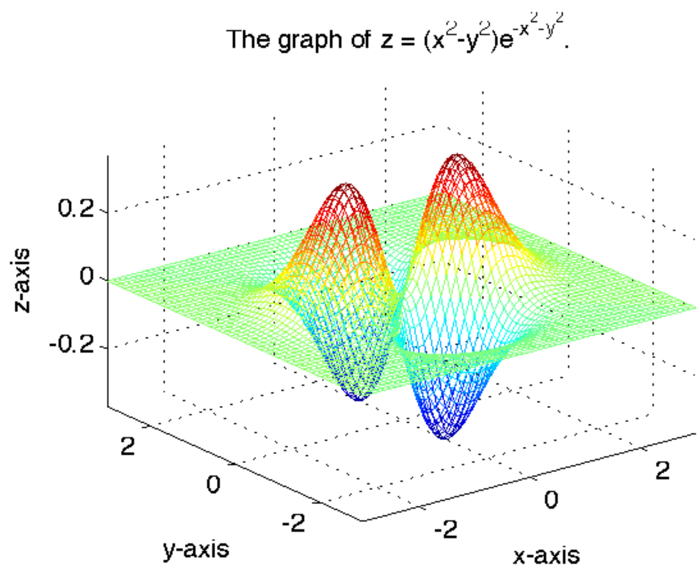


Figure 3.8. Hidden line removal allows the use to "see through" the mesh.

3.3 Exercises

In **Exercises 1-16**, use Matlab to sketch the graph of the given function on the indicated domain. Rotate each result with Matlab's **view(az,el)** command to something near the standard orientation, then add axes labels and a title containing the equation of the surface.

1. $z = f(x,y) = 10 - 2x - y$ on

$$D = \{(x,y) : \ -3 \le x, y \le 3\}.$$

2. $z = f(x,y) = 2$ on

$$D = \{(x,y) : \ -3 \le x, y \le 3\}.$$

3. $y = f(x,z) = x + 2z - 5$ on

$$D = \{(x,z) : \ -3 \le x, z \le 3\}.$$

4. $x = f(y,z) = y - 2z + 4$ on

$$D = \{(y,z) : \ -3 \le y, z \le 3\}.$$

5. $x = f(y,z) = -2$ on

$$D = \{(y,z) : \ -2 \le y, z \le 2\}.$$

6. $z = f(x,y) = 10 - 5x$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 3\}.$$

7. $y = f(x,z) = -2$ on

$$D = \{(x,z) : \ x| \le 2, \ |z| \le 5\}.$$

8. $y = f(x,z) = 12 - 4x$ on

$$D = \{(x,z) : \ -1 \le x \le 4, \ |z| \le 4\}.$$

9. $z = f(x,y) = 2x - y$ on

$$D = \{(x,y) : \ x| \le 1, \ |y| \le 3\}.$$

10. $x = f(y,z) = 4 - z$ on

$$D = \{(y,z) : \ 0 \le y \le 4, \ -1 \le z \le 5\}.$$

11. $z = f(x,y) = 1/(1 + x^2 + y^2)$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 2\}.$$

12. $z = f(x,y) = \cos(x^2 + y^2)$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 2\}.$$

13. $z = f(x,y) = \sin\sqrt{x^2 + y^2}$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 2\}.$$

14. $z = f(x,y) = \sin\sqrt{x^2 + y^2}/\sqrt{x^2 + y^2}$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 2\}.$$

15. $z = f(x,y) = xe^{-x^2 - y^2}$ on

$$D = \{(x,y) : \ |x| \le 2, \ |y| \le 3\}.$$

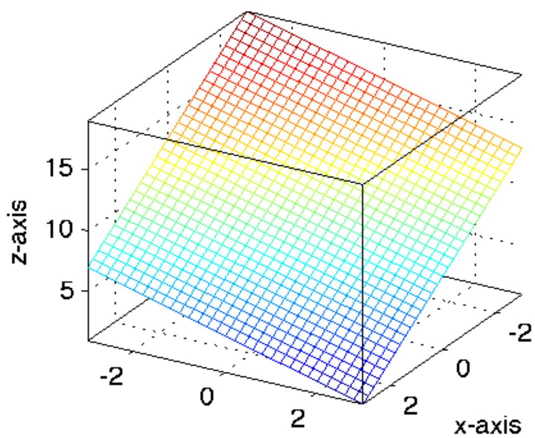
16. $z = f(x,y) = \cos x \cos y$ on

$$D = \{(x,y) : \ |x| \le 2\pi, \ |y| \le 2\pi\}.$$

3.3 Answers

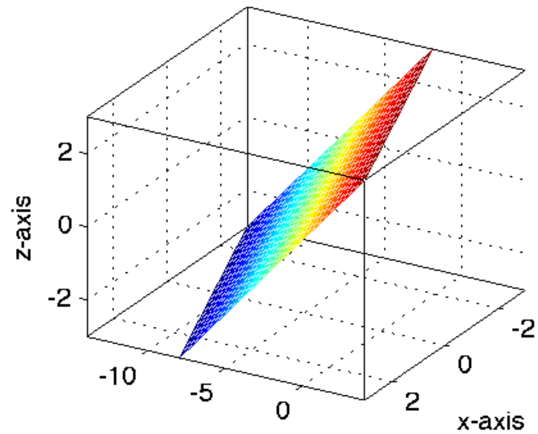
1.

```
[x,y]=meshgrid(-3:0.2:3);
z=10-2*x-y;
mesh(x,y,z)
axis tight
box on
view(120,30)
```



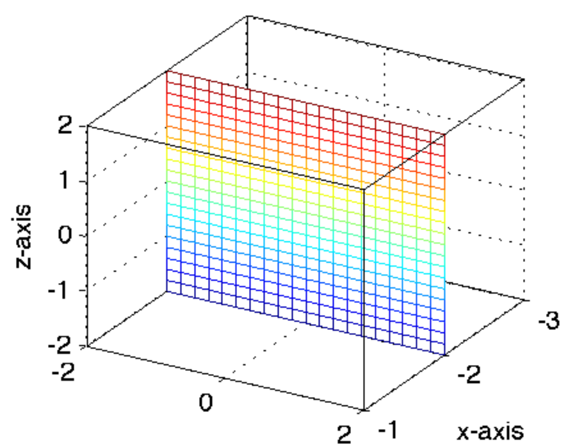
3.

```
[x,z]=meshgrid(-3:0.2:3);
y=x+2*z-5;
mesh(x,y,z)
axis tight
box on
view(120,30)
```



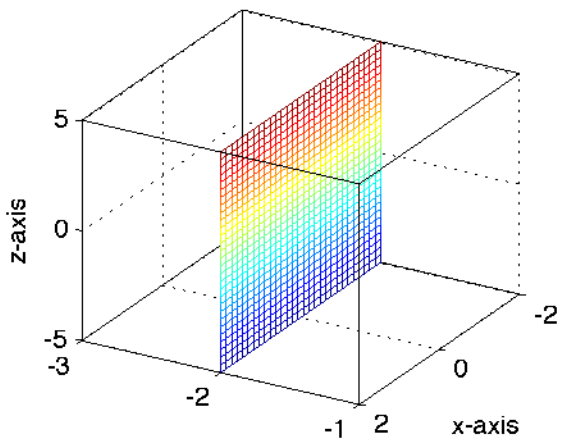
5.

```
[y,z]=meshgrid(-2:0.2:2);
x=-2*ones(size(y));
mesh(x,y,z)
axis tight
box on
view(120,30)
```



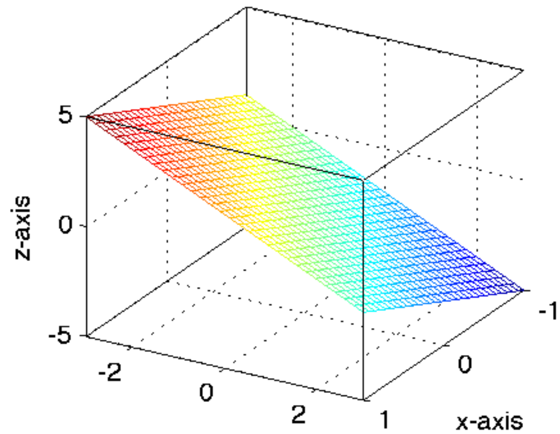
7.

```
x=linspace(-2,2,30);
z=linspace(-5,5,30);
[x,z]=meshgrid(x,z);
y=-2*ones(size(x));
mesh(x,y,z)
axis tight
box on
view(120,30)
```



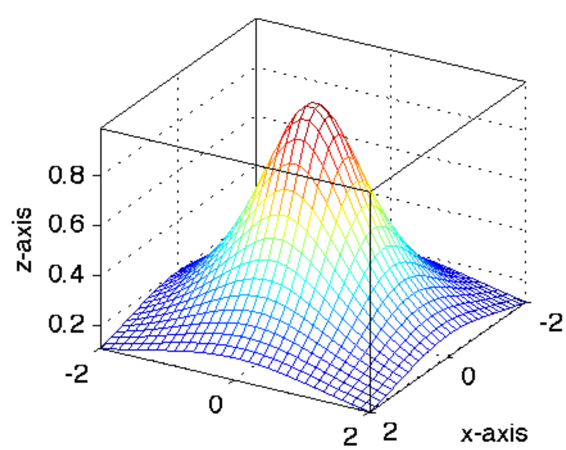
9.

```
x=linspace(-1,1,30);
y=linspace(-3,3,30);
[x,y]=meshgrid(x,y);
z=2*x-y;
mesh(x,y,z)
axis tight
box on
view(120,30)
```



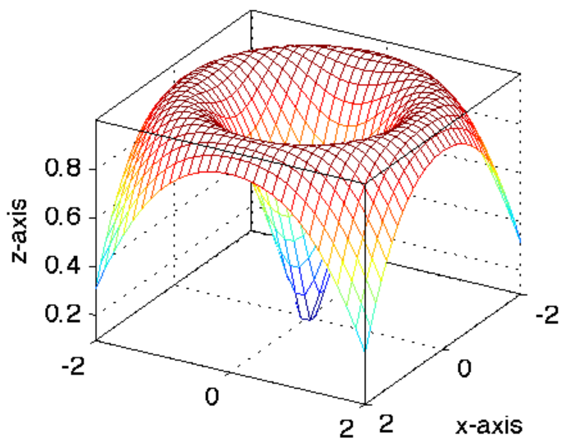
11.

```
x=linspace(-2,2,30);
y=linspace(-2,2,30);
[x,y]=meshgrid(x,y);
z=1./(1+x.^2+y.^2);
mesh(x,y,z)
axis tight
box on
view(120,30)
```



13.

```
x=linspace(-2,2,30);
y=linspace(-2,2,30);
[x,y]=meshgrid(x,y);
z=sin(sqrt(x.^2+y.^2));
mesh(x,y,z)
axis tight
box on
view(120,30)
```



15.

```
x=linspace(-2,2,30);
y=linspace(-2,2,30);
[x,y]=meshgrid(x,y);
z=x.*exp(-x.^2-y.^2);
mesh(x,y,z)
axis tight
box on
view(120,30)
```