# 3.4 Parametric Surfaces in Matlab

In this section we will again examine surfaces in three-space. However, the position of a point on the surface will be defined in terms of parameters. In this way we open a fascinating world of surfaces that you may never have seen before, in much the same way that parametric equatins opened a whole new world of curves in the plane.

Let's get started.

▶ **Example 1.** *Sketch the surface defined by the parametric equations*

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= r, \end{aligned} \tag{3.1}$$

*where* $0 \le r \le 1$ *and* $0 \le \theta \le 2\pi$.

Note that $x$, $y$, and $z$ are defined in terms of two parameters, $r$ and $\theta$, which are restriced to $0 \le r \le 1$ and $0 \le \theta \le 2\pi$. The first task is to create a grid of $(r, \theta)$ pairs, after which each pair $(r, \theta)$ will be mapped to a triplet $(x, y, z)$ on the surface via the parametric **equations (3.1)**.

First, use the **linspace** commands to create vectors $\mathbf{r}$ and $\theta$.

```
r=linspace(0,1,30);
theta=linspace(0,2*pi,30);
```

When using Matlab's **mesh** command, too few grid points will create a surface with the "Jaggies," but too many grid points will create a mesh that appears to be a solid mass of color, and you'll be unable to distinguish grid lines on the surface. The number 30 usually provides a nice number of grid points to start your exploration, then you can adjust upward or downward from there.

The next step is to create the grid of $(r, \theta)$ pairs with Matlab's **meshgrid** command.

```
[r,theta]=meshgrid(r,theta);
```

If you remove the semicolons in each of the previous commands, you'll note that **r** and $\theta$ were first vectors, but after the **meshgrid** command, $r$ and $\theta$ are matrices. If your find this overwriting using the same variable distasteful, you can try something like **[R,THETA]=meshgrid(r,theta)** instead. However, we have no further use of the *vectors* **r** and $\theta$, so we are perfectly happy overwriting $r$ and $\theta$ with the matrix output of the **meshgrid** command.

At this point, each row of the matrix $r$ contains the contents of the former vector **r**, and each column of the matrix $\theta$ contains the contents of the vector $\theta$. If you mentally superimpose the matrix $\theta$ atop the matrix $r$, you can imagine a two dimensional grid of $(r, \theta)$ pairs. We now use the parametric **equations (3.1)** to compute the triplets $(x, y, z)$ at each pair $(r, \theta)$ in the grid. Note that this requires the use of array operators, as one would expect.

```
x=r.*cos(theta);
y=r.*sin(theta);
z=r;
```

Each triplet $(x, y, z)$ is a point on the surface. We can use the **mesh** command to connect neighboring points with line segments.

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

We orient the axis, "tighten" the plot, and turn the **box on** to provided some depth to the visualization.

```
view(135,30)
axis tight
box on
```

Finally, labeling the axes helps us to visualize the orientation, as we can clearly see in **Figure 3.1**.

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

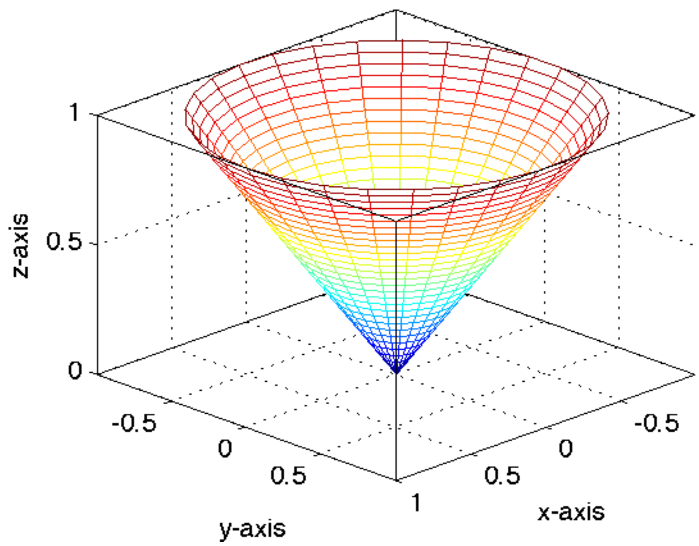



**Figure 3.1.** The surface defined by the parametric **equations (3.1)** is a cone.

Let's look at another example.

▶ **Example 2.** *Sketch the surface defined by the equations*

$$\begin{aligned} x &= \sin\phi\cos\theta \\ y &= \sin\phi\sin\theta \\ z &= \cos\phi, \end{aligned} \tag{3.2}$$

*where* $0 \leq \phi \leq \pi$ *and* $0 \leq \theta \leq 2\pi$.

First, create vectors $\phi$ and $\theta$ so that $0 \leq \phi \leq \pi$ and $0 \leq \theta \leq 2\pi$.

```
phi=linspace(0,pi,30);
theta=linspace(0,2*pi,30);
```

Now, create a grid of $(\phi, \theta)$ pairs.

```
[phi,theta]=meshgrid(phi,theta);
```

Use the parametric **equations (3.2)** to calculate surface triplets $(x, y, z)$ at each grid pair $(\phi, \theta)$. Again, array operators are required.

```
x=sin(phi).*cos(theta);
y=sin(phi).*sin(theta);
z=cos(phi);
```

We can now create a mesh of the surface with Matlab's **mesh** command.

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

We adjust the orientation, turn the **box on** to add depth of visualization, then issue **axis equal** command to show that the surface is actually a sphere, and not an ellipsoid.

```
axis equal
view(135,30)
box on
```

Finally, we annotate the axes to produce the final image in **Figure 3.2**.

```
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')T
```

Let's look at another example. It's time to have some fun!

▶ **Example 3.** *Sketch the graph of the surface defined by the parametric equations*

$$\begin{aligned} x &= 2\left[1 - e^{u/(6\pi)}\right]\cos u \cos^2\left(\frac{v}{2}\right) \\ y &= 2\left[-1 + e^{u/(6\pi)}\right]\sin u \cos^2\left(\frac{v}{2}\right) \\ z &= 1 - e^{u/(3\pi)} - \sin v + e^{u/(6\pi)}\sin v, \end{aligned} \tag{3.3}$$

*where* $0 \leq u \leq 6\pi$ *and* $0 \leq v \leq 2\pi$.

First, create vectors **u** and **v** so that $0 \leq u \leq 6\pi$ and $0 \leq v \leq 2\pi$.

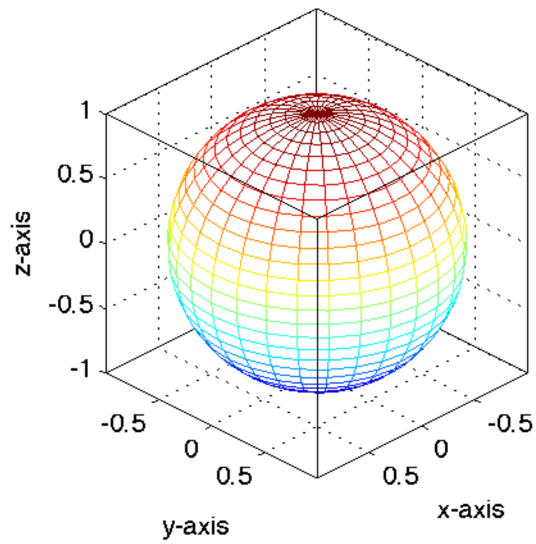



**Figure 3.2.** The surface defined by the parametric **equations (3.2)** is a sphere.

```
u=linspace(0,6*pi,60);
v=linspace(0,2*pi,60);
```

Use the vectors **u** and **v** to create a grid of $(u, v)$ pairs.

```
[u,v]=meshgrid(u,v);
```

Use the parametric **equations (3.3)** to compute $(x, y, z)$ triplets at each $(u, v)$ pair in the grid. Again, array operators are expected.

```
x=2*(1-exp(u/(6*pi))).*cos(u).*cos(v/2).^2;
y=2*(-1+exp(u/(6*pi))).*sin(u).*cos(v/2).^2;
z=1-exp(u/(3*pi))-sin(v)+exp(u/(6*pi)).*sin(v);
```

Create a mesh of the surface with Matlab's **mesh** command.

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

Adjust the orientation, set **axis equal**, and turn the **box on** to add depth to the visualization.

```
view(160,10)
axis equal
box on
```

Annotate the axes in the usual manner to produce the final "seashell" in **Figure 3.3**.

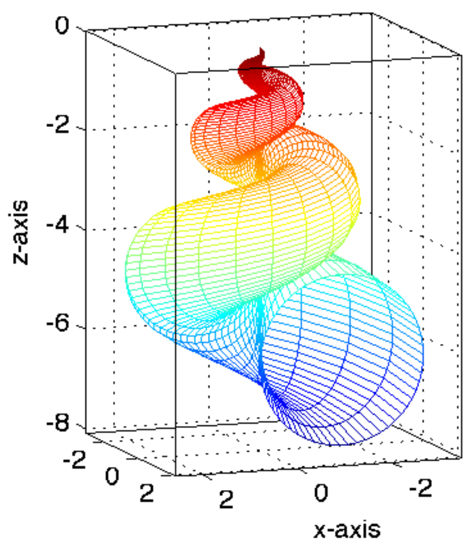



**Figure 3.3.** Can you hear the sound of the ocean?

Matlab also offers a **surf** command which colors the patches between the gridlines. We need only execute **surf(x,y,z)** to see the effect in **Figure 3.4**(a).

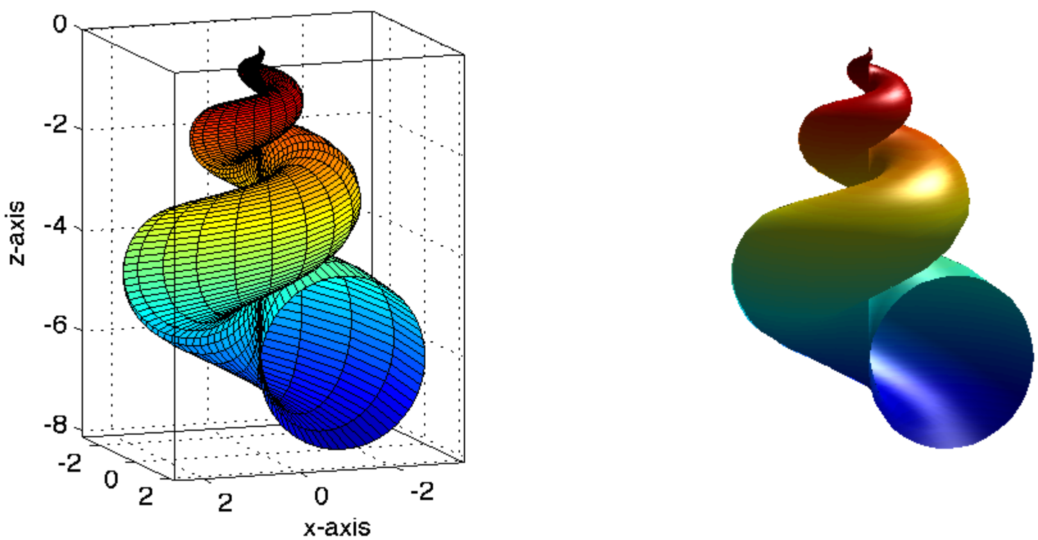
```
surf(x,y,z)
```

You might want to turn off hidden line removal and use the rotation tool on the figure toolbar to turn and twist the figure.

```
hidden off
```

Finally, Matlab offers truly stunning graphics capabilities. We can use the surf command in combination with edge and face coloring techniques, add lighting, and turn the axes off to produce an image in **Figure 3.4**(b) that is quite beautiful.

```
surf(x,y,z,...
    'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
camlight left
view(160,10)
axis equal
axis off
```




(a) (b)

**Figure 3.4.** A surface plot instead of a mesh colors the patches between meshlines on the surface. Adding lighting makes for a beautiful image.

Parametric surfaces are just too much fun! Let's look at another example.

▶ **Example 4.** *Sketch the surface defined by the parametric equations*

$$\begin{aligned} x &= \frac{v}{2}\sin\frac{u}{2} \\ y &= \left(1+\frac{v}{2}\cos\frac{u}{2}\right)\sin u \\ z &= \left(1+\frac{v}{2}\cos\frac{u}{2}\right)\cos u, \end{aligned} \tag{3.4}$$

*where* $0 \le u \le 2\pi$ *and* $-1 \le v \le 1$.

Create the grid with the constraints $0 \le u \le 2\pi$ and $-1 \le v \le 1$.

```
u=linspace(0,2*pi,30);
v=linspace(-1,1,15);
[u,v]=meshgrid(u,v);
```

Use the parametric **equations (3.4)** to compute surface triplete $(x, y, z)$ at each $(u, v)$ in the grid.

```
z=(r+v/2.*cos(u/2)).*cos(u);
y=(r+v/2.*cos(u/2)).*sin(u);
x=v/2.*sin(u/2);
```

Draw the surface.

```
surf(x,y,z)
```

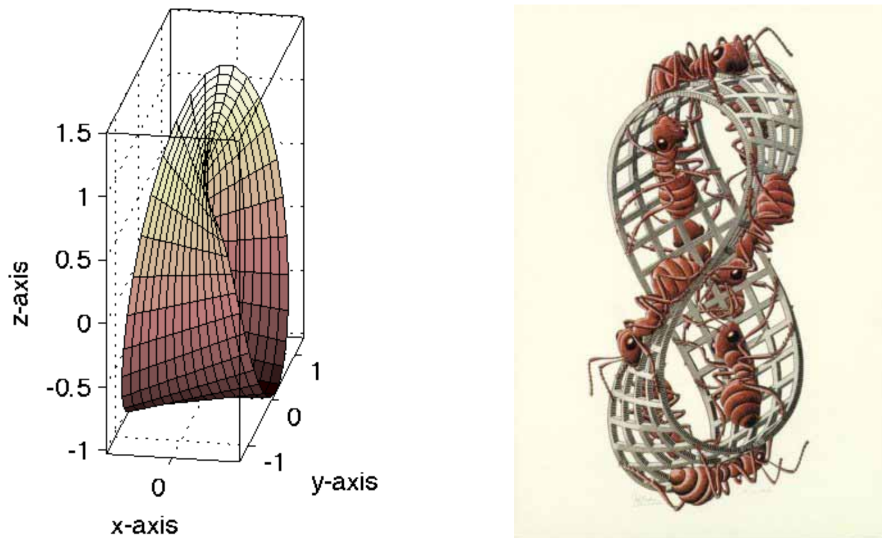
Choose the **pink** colormap[2] set the **axis equal**, and add **box on** to help with the depth of the visualization.

```
box on
axis equal
colormap pink
```

Finally, we set the orientation in order to compare our result in **Figure 3.5**(a) with the famous *Mobius Strip* drawn by Escher in **Figure 3.5**(b).

You can create a Mobius strip on your own. Simply take a long strip of paper, maybe 1-2 inches wide by 12-24 inches in length. Twist the strip once, then glue the ends together to match what you see in **Figures 3.5**(a) and (b). As the ants walk along the strip, they quickly learn that the strip has only one side, not two. The Mobius Strip is an important example that is closely studied in a subject called *differential geometry*.

[2] For information on the available colormaps in Matlab, type **help graph3d** at the command prompt.




(a) (b)

**Figure 3.5.** The classic *Mobius Strip* has only one side.

Let's look at one final example, the *Klein Bottle*.

▶ **Example 5.** *Sketch the surface defined by the parametric equations*

$$\begin{aligned} x &= \left(r + \cos\frac{u}{2}\sin v - \sin\frac{u}{2}\sin 2v\right)\cos u \\ y &= \left(r + \cos\frac{u}{2}\sin v - \sin\frac{u}{2}\sin 2v\right)\sin u \\ z &= \sin\frac{u}{2}\sin v + \cos\frac{u}{2}\sin 2v, \end{aligned} \tag{3.5}$$

*with* $r = 1$ *and* $0 \leq u, v \leq 2\pi$.

Set $r = 1$, then create the grid with constraints $0 \leq u, v \leq 2\pi$.

```
r=1;
u=linspace(0,2*pi,60);
v=linspace(0,2*pi,60);
[u,v]=meshgrid(u,v);
```

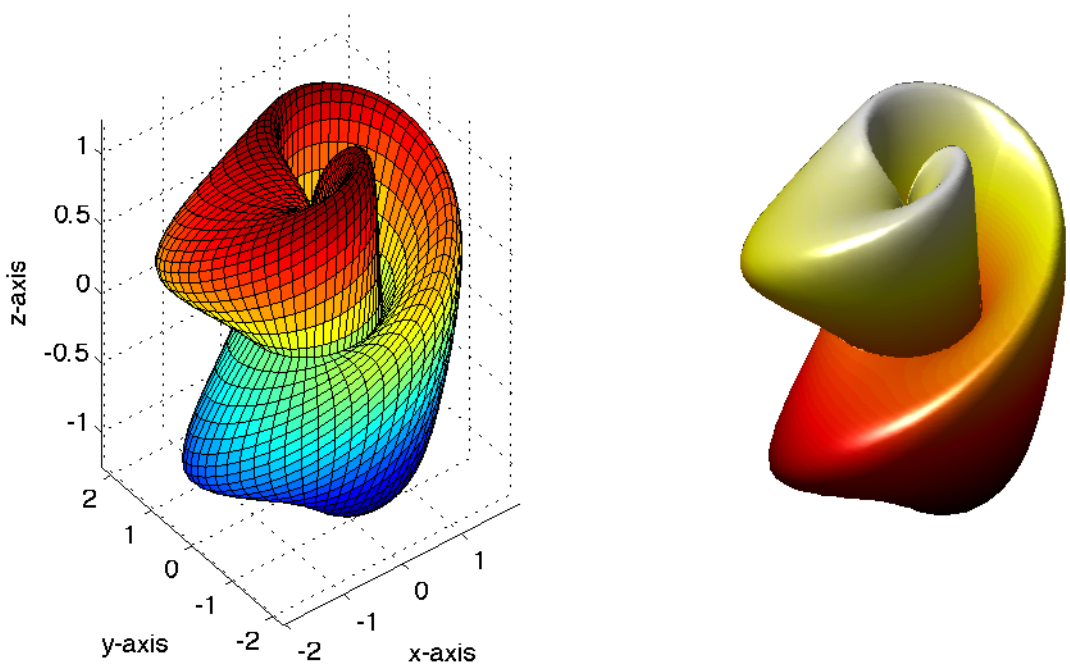
Calculate surface triplets $(x, y, z)$ at each grid pair $(u, v)$, using **equations (3.5)**.

```
x=(r+cos(u/2).*sin(v)-sin(u/2).*sin(2*v)).*cos(u);
y=(r+cos(u/2).*sin(v)-sin(u/2).*sin(2*v)).*sin(u);
z=sin(u/2).*sin(v)+cos(u/2).*sin(2*v);
```

Use the **surf** command to draw the surface, a *Klein Bottle*. We'll use the default view. The result is shown in **Figure 3.6**(a).

```
surf(x,y,z)
```

Let the creative juices flow! Do some interpretation of edge and face color, add some lighting, and voila! You obtain the beautiful image shown in **Figure 3.6**(b).




(a) (b)

**Figure 3.6.** A Klein Bottle, one plain, one pretty! Both fascinating!

Here is the code used to do the fancy shading and lighting seen in **Figure 3.6**(b).

```
surf(x,y,z,...
    'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
camlight left
colormap(hot)
axis off
```

Note the new choice of colormap, the “hot” colormap, which adds a “heated look” to the surface of our Klein Bottle. Click the rotate icon on the toolbar, then use the mouse to twist and turn and explore this fascinating surface!

Finally, check out the hand-made Klein Bottle in **Figure 3.7** crafted by an artist.

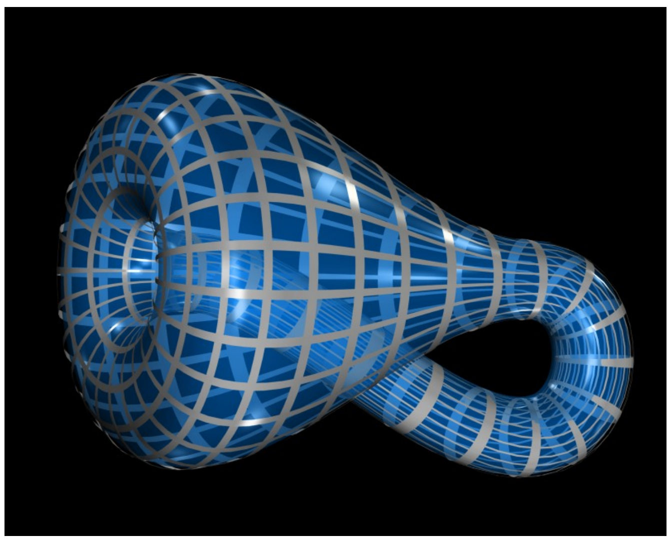


**Figure 3.7.** An artist’s rendering of a Klein Bottle.

## 3.4 Exercises

**1.** Sketch the *Enneper's Surface* defined by the parametric equations

$$x = u - u^3/3 + uv^2$$
$$y = v - v^3/3 + u^2 v$$
$$z = u^2 - v^2,$$

where $-1.5 \leq u, v \leq 1.5$.

**2.** Sketch the *Ellipsoid* defined by the parametric equations

$$x = a \cos u \sin v$$
$$y = b \sin u \sin v$$
$$z = c \cos v,$$

where $0 \leq u \leq 2\pi$ and $0 \leq v \leq \pi$. Start with $a = 3$, $b = 4$, and $c = 5$, then experiment by varying the values of $a$, $b$, and $c$. Explain the effect of varying $a$, $b$, and $c$. You will want to use **axis equal** on this exercise.

**3.** Sketch the *Hyperboloid of One Sheet* defined by the parametric equations

$$x = a \cosh(u) \cos v$$
$$y = b \cosh u \sin v$$
$$z = c \sinh u,$$

where $-2 \leq u \leq 2$ and $0 \leq v \leq 2\pi$. Start with $a = 1$, $b = 1$, and $c = 1$, then experiment by varying the values of $a$, $b$, and $c$. Explain the effect of varying $a$, $b$, and $c$. You will want to use **axis equal** on this exercise.

**4.** Sketch the *Hyperboloid of One Sheet* defined by the parametric equations

$$x = a \cosh(u) \cos v$$
$$y = b \cosh u \sin v$$
$$z = c \sinh u,$$

where $-2 \leq u \leq 2$ and $0 \leq v \leq 2\pi$. Start with $a = 1$, $b = 1$, and $c = 1$, then experiment by varying the values of $a$, $b$, and $c$. Explain the effect of varying $a$, $b$, and $c$. You will want to use **axis equal** on this exercise.

**5.** Sketch the *Hyperboloid of Two Sheets* defined by the parametric equations

$$x = a \sinh u \cos v$$
$$y = b \sinh u \sin v$$
$$z = c \cosh u,$$

where $-2 \leq u \leq 2$ and $0 \leq v \leq 2\pi$. Start with $a = 1$, $b = 1$, and $c = 1$, then hold the graph and plot again with $a = 1$, $b = 1$, and $c = -1$ to see why this is called a *Hyperboloid of Two Sheets*.

**6.** Sketch the *Lissajous Surface* defined by the parametric equations

$$x = \sin u$$
$$y = \sin v$$
$$z = \sin((d - au - bv)/c),$$

where $-\pi \leq u, v \leq \pi$. Set $a = 1$, $b = 1$, $c = 1$ and $d = 0$. You will want to use **axis equal** on this exercise.

**7.** Sketch the *Whitney Umbrella* defined by the parametric equations

$$\begin{aligned} x &= uv \\ y &= u \\ z &= v^2, \end{aligned}$$

where $-1 \le u, v \le 1$. You will want to use **axis equal** on this exercise.

**8.** Sketch the *Steiner Surface* defined by the parametric equations

$$\begin{aligned} x &= \sin 2u \cos^2 v \\ y &= \sin u \sin 2v \\ z &= \cos u \sin 2v, \end{aligned}$$

where $0 \le u \le \pi$ and $-\pi/2 \le v \le \pi/2$. You will want to use **axis equal** on this exercise.

**9.** Sketch the *Pseudosphere* defined by the parametric equations

$$\begin{aligned} x &= \cos u \sin v \\ y &= \sin u \sin v \\ z &= \cos v + \ln(\tan(v/2)), \end{aligned}$$

where $0 \le u \le 2\pi$ and $0 < v < \pi$. Note that when $v$ nears zero or $\pi$, the term $\ln(\tan(v/2)$ "leaks" to either minus or positive infinity. Keep this in mind when selecting a vector **v** to be used in an eventual grid of $(u, v)$ pairs.

**10.** Sketch *Dini's Surface* defined by the parametric equations

$$\begin{aligned} x &= \cos u \sin v \\ y &= \sin u \sin v \\ z &= \cos v + \ln(\tan(v/2)) + au, \end{aligned}$$

where $0 \le u \le 2\pi$ and $0 < v < \pi$. Note that when $v$ nears zero or $\pi$, the term $\ln(\tan(v/2)$ "leaks" to either minus or positive infinity. Keep this in mind when selecting a vector **v** to be used in an eventual grid of $(u, v)$ pairs. Experiment with different values of the constant $a$.

**11.** Sketch *Helicoid* defined by the parametric equations

$$\begin{aligned} x &= av \cos u \\ y &= av \sin u \\ z &= bu, \end{aligned}$$

where $0 \le u \le 2\pi$ and $-d < v < d$, $d > 0$. Experiment with different values of the constants $a$, $b$, and $d$, then write a short description explaining how varying each of the constants $a$, $b$, and $d$ affects the surface.

**12.** Sketch *Catenoid* defined by the parametric equations

$$\begin{aligned} x &= a \cos u \cosh(v/a) \\ y &= a \sin u \cosh(v/a) \\ z &= v, \end{aligned}$$

where $0 \le u \le 2\pi$ and $-c < v < c$, $c > 0$. Experiment with different values of the constants $a$ and $c$, then write a short description explaining how varying each of the constants $a$ and $c$ affects the surface.

**13.** Sketch *Torus* defined by the parametric equations

$$\begin{aligned} x &= (a + b \cos v) \cos u \\ y &= (a + b \cos v) \sin u \\ z &= b \sin v, \end{aligned}$$

where $0 \le u \le 2\pi$ and $0 \le v \le 2\pi$. Experiment with different values of the constants $a$ and $b$, then write a short description explaining how varying each of the constants $a$ and $b$ affects the surface.

**14.** Sketch the *Alien Space Ship*. Set $a = 0.4$ and define $w = \sqrt{1-a^2}$. Define

$$d = a\big[(w\cosh(au))^2 + (a\sin(wv))^2\big].$$

Now, define the parametric equations

$$\begin{aligned}
x &= -u + 2w^2\cosh(au)\sinh(au)/d\\
y &= 2w\cosh(au)\big[-w\cos v\cos(wv) - \sin v\sin(wv)\big]/d\\
z &= 2w\cosh(au)\big[-w\sin v\cos(wv) + \cos v\sin(wv)\big]/d,
\end{aligned}$$

where $-13.4 \le u \le 13.4$ and $-37.2 \le v \le 37.2$. Blast off!

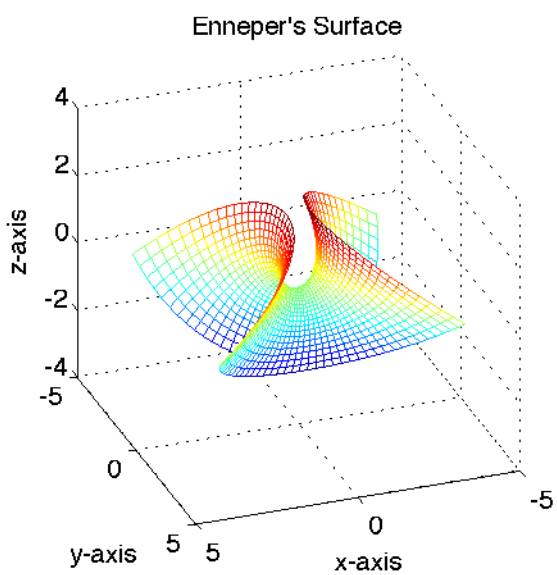
## 3.4 Answers

**1.** Create the grid and calculate **x**, **y**, and **z** at each $(u, v)$ pair of the grid.

```
u=linspace(-1.5,1.5,40);
v=u;
[u,v]=meshgrid(u,v);
x=u-u.^3/3+u.*v.^2;
y=v-v.^3/3+u.^2.*v;
z=u.^2-v.^2;
```

Draw the surface, add some annotations, and adjust the view.

```
mesh(x,y,z)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('Enneper''s Surface')
view(160,30)
```




**3.** Set some constants and create the grid.

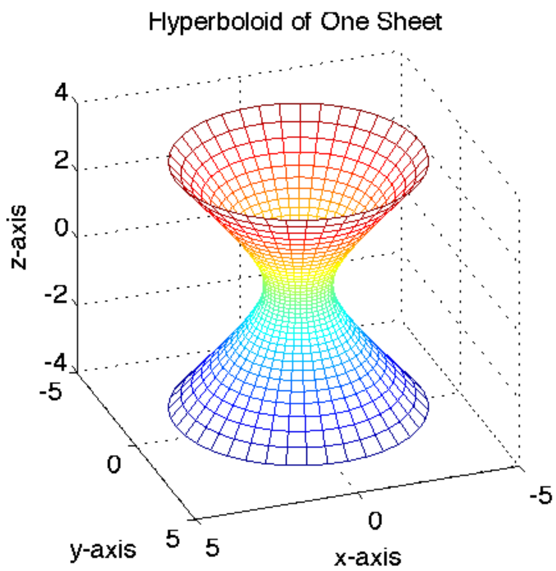
```
a=1; b=1; c=1;
u=linspace(-2,2,40);
v=linspace(0,2*pi,40);
[u,v]=meshgrid(u,v);
```

Calculate **x**, **y**,a dn **z** at each pair $(u, v)$ in the grid.

```
x=a*cosh(u).*cos(v);
y=b*cosh(u).*sin(v);
z=c*sinh(u);
```

Draw the surface, add some annotations, and adjust the view.

```
mesh(x,y,z)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('Hyperboloid of One Sheet')
view(160,30)
```




After some experimentation with values of $a$, $b$, and $c$, both $a$ and $b$ control the elliptical nature of cross sections parallel to the $xy$-plane, and $c$ contols the height of the cone. The command **axis equal** will help in this exploration.

**5.** Set some constants and create the grid.

```
a=1; b=1; c=1;
u=linspace(-2,2,40);
v=linspace(0,2*pi,40);
[u,v]=meshgrid(u,v);
```

Calculate **x**, **y**, and **z** at each $(u, v)$ in the grid.

```
x=a*sinh(u).*cos(v);
y=b*sinh(u).*sin(v);
z=c*cosh(u);
```

Draw the mesh.

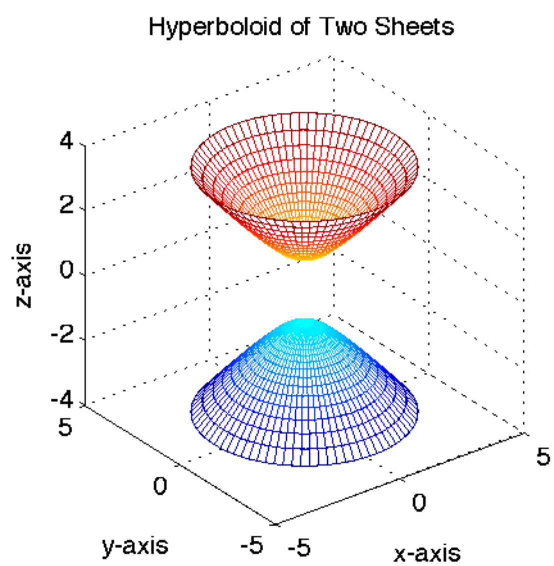
```
mesh(x,y,z)
```

Now, hold the graph, set $c = -1$, and repeat.

```
hold on
c=-1;
x=a*sinh(u).*cos(v);
y=b*sinh(u).*sin(v);
z=c*cosh(u);
mesh(x,y,z)
```

Annotate the plot.

```
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('Hyperboloid of Two Sheets')
```




**7.** Set up the grid.

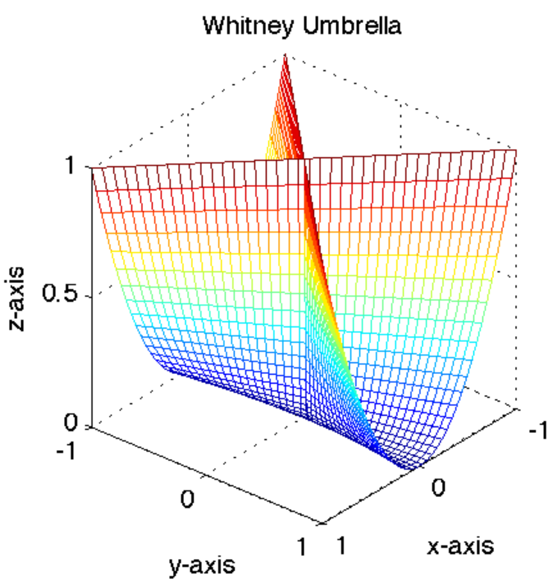
```
u=linspace(-1,1,40);
v=u;
[u,v]=meshgrid(u,v);
```

Compute **x**, **y**, and **z** at each pair $(u, v)$ in the grid.

```
x=u.*v;
y=u;
z=v.^2;
```

Draw and orient the surface, then annotate the plot.

```
mesh(x,y,z)
view(130,30)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('Whitney Umbrella')
```




**9.** Set up the grid. We chose to get fairly close to 0 and $\pi$ in computing **v**.

```
u=linspace(0,2*pi,40);
v=linspace(pi/48,47*pi/48,40);
[u,v]=meshgrid(u,v);
```

Use the parametric equations to determine **x**, **y**, and **z** at each $(u, v)$ pair of the grid.

```
x=cos(u).*sin(v);
y=sin(u).*sin(v);
z=cos(v)+log(tan(v/2));
```

We use the **surf** command with some shading and lighting. For fun, we use a different colormap.

```
surf(x,y,z,...
    'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
camlight left
colormap(copper)
```

We set **axis equal**, then **axis off**, orient the view, and add a title.

```
axis equal
axis off
view(160,10)
title('Pseudosphere.')
```


11. Set up some constants.

```
a=1; b=1; d=2;
```

Set up the grid.

```
u=linspace(0,2*pi,40);
v=linspace(-d,d,40);
[u,v]=meshgrid(u,v);
```

Use the parametric equations to determine **x**, **y**, and **z** at each $(u, v)$ pair of the grid.

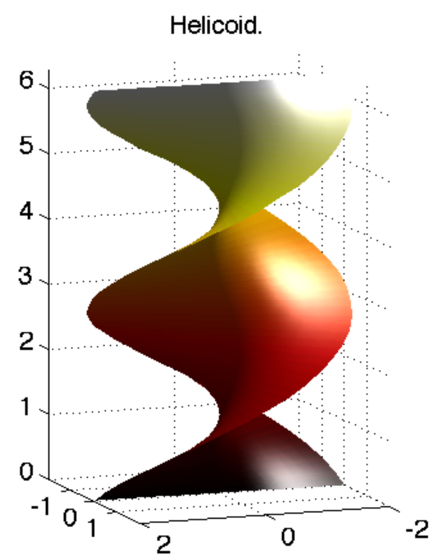
```
x=a*v.*cos(u);
y=a*v.*sin(u);
z=b*u;
```

We use the **surf** command with some shading and lighting. For fun, we use a different colormap.

```
surf(x,y,z,...
    'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
camlight left
colormap(hot)
```

We set **axis equal**, orient the view, and add a title.

```
axis equal
view(160,10)
title('Helicoid.')
```

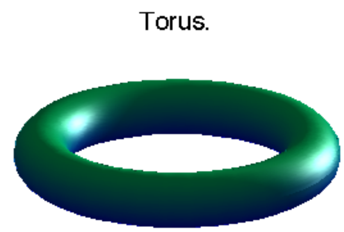



13. Set up some constants.

```
a=5; b=1;
```

Set up the grid.

```
u=linspace(0,2*pi,40);
v=u;
[u,v]=meshgrid(u,v);
```




Use the parametric equations to determine **x**, **y**, and **z** at each $(u, v)$ pair of the grid.

```
x=(a+b*cos(v)).*cos(u);
y=(a+b*cos(v)).*sin(u);
z=b*sin(v);
```

We use the **surf** command with some shading and lighting. For fun, we use a different colormap.

```
surf(x,y,z,...
    'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
camlight left
colormap(winter)
```

We set **axis equal**, turn the axes off, orient the view, and add a title.

```
axis equal
axis off
view(150,20)
title('Torus.')
```