3.5 Space Curves in Matlab

To draw the graphs of curves in the plane, we used Matlab's **plot** command. To animate curves in the plane, we used the **comet** command. To achieve the same effects in three-space, use Matlab's **plot3** and **comet3** commands. Curves that travel through three-space are called *space curves*. Let's look at some examples.

▶ **Example 1.** *Sketch the graph of the space curve defined by the parametric equations*

$$\begin{aligned} x &= a\cos\omega t \\ y &= a\sin\omega t \\ z &= bt. \end{aligned} \tag{3.1}$$

Set $a = 2$, $b = 0.1$, $\omega = 2$, *and restrict* $0 \le t \le 12\pi$.

Set $a = 2$. This constant controls the amplitude of x and y. Set $b = 0.1$. As you will see, this controls the rate at which z (height) changes with respect to time. Set $\omega = 2$. This controls the rate at which the particle circles the origin, with an angular velocity of 2 radians per second.

```
a=2;
b=0.1;
w=2;
```

Create a vector of t-values using the given constraint $0 \le t \le 12\pi$.

```
t=linspace(0,12*pi,500);
```

Use the parametric **equations (3.1)** to calculate triplets (x, y, z) on the space curve in terms of t.

```
x=a*cos(w*t);
y=a*sin(w*t);
z=b*t;
```

To get a sense of the motion, use the **comet3** command.

```
comet3(x,y,z)
```

To provide a finished plot, use Matlab's **plot3** command, then add axes labels and a title. The following commands will produce the *helix* shown in **Figure 3.1**

```
plot3(x,y,z)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('x = 2 cos(t), y = 2 sin(t), z = 0.1t.')
```

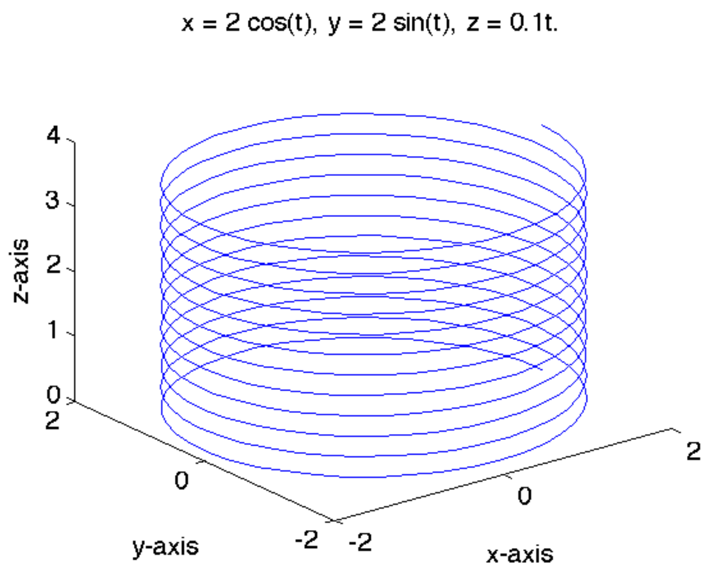


Figure 3.1. The space curve determine by the parametric **equations (3.1)** is called a *helix*.

Let's look at another example.

► **Example 2.** *Suppose that a ship travels from the south pole to the north pole keeping a fixed angle to all meridians. Then the path traveled is described by the parametric equations*

$$x = \frac{\cos t}{\sqrt{1+\alpha^2 t^2}}$$
$$y = \frac{\sin t}{\sqrt{1+\alpha^2 t^2}} \tag{3.2}$$
$$z = -\frac{\alpha t}{\sqrt{1+\alpha^2 t^2}}.$$

Set $\alpha = 0.2$ *and restrict* $-12\pi \leq t \leq 12\pi$.

Set $\alpha = 0.2$ and create a vector of t-values subject to the constraint $-12\pi \leq t \leq 12\pi$.

```
alpha=0.2;
t=linspace(-12*pi,12*pi,500);
```

Use the parametric **equations (3.2)** to compute the positions (x, y, z) on the spherical spiral as a function of time t.

```
x=cos(t)./sqrt(1+alpha^2*t.^2);
y=sin(t)./sqrt(1+alpha^2*t.^2);
z=alpha*t./sqrt(1+alpha^2*t.^2);
```

Use **comet3** to animate the motion, then follow this with Matlab's **plot3** command. This will produce the spherical spiral shown in **Figure 3.2**.

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

Handle Graphics

Many space curves are closely related with one or another particular class of surfaces. In the case of the spherical spiral, one would intuit a close relationship with a sphere. So, let's draw a sphere of appropriate radius, then superimpose the spherical spiral of **Example 2**.

In the section on Parametric Surfaces, we saw the parametric equations of a sphere of radius r, which we repeat here for convenience.

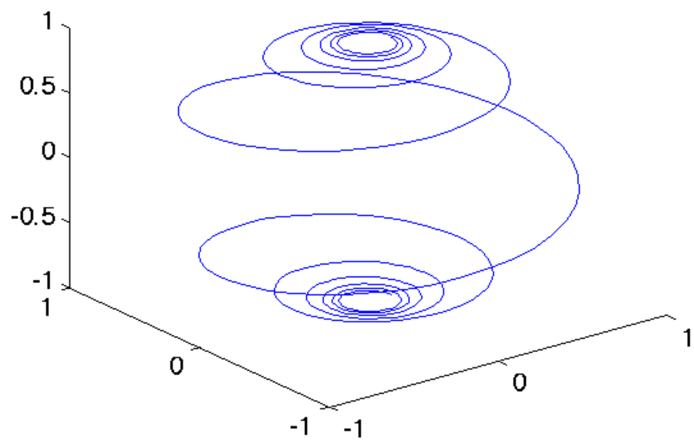


Figure 3.2. The path taken by the ship is an example of a spherical spiral.

$$
\begin{aligned}
x &= r \sin\phi \cos\theta \\
y &= r \sin\phi \sin\theta \\
z &= r \cos\phi
\end{aligned}
\tag{3.3}
$$

Set $r = 1$ and create a grid of (ϕ, θ) pairs, where $0 \le \phi \le \pi$ and $0 \le \theta \le 2\pi$.

```
r=1;
phi=linspace(0,pi,30);
theta=linspace(0,2*pi,40);
[phi,theta]=meshgrid(phi,theta);
```

Use the parametric **equations (3.3)** to compute each point (x, y, z) on the surface of the sphere as a function of each grid pair (ϕ, θ).

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

Now, plot the sphere with the **mesh** command.

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

The command **mhndl=mesh(x,y,z)** stores a "handle" to the mesh in the variable **mhandl**[2]. A handle is a numerical identifier associated with an object we place on the figure window. We've left the command **mhndl=mesh(x,y,z)** unsuppressed (no semicolon), so you can look in Matlab's command window to see the numerical value stored in **mhndl**.

Remember, **mhndl** is a "handle" that points at the mesh object we've just plotted. We can obtain a full list of property-value settings of this mesh by executing the command **get(mhndl)**. Matlab will respond with a huge list of property-value pairs for the current mesh. We are interested in three of these pairs: **EdgeColor**, **EdgeAlpha**, and **FaceAlpha**. We are going to set the edgecolor to a shade of gray, and we're going to make the edges and faces transparent to a certain degree. To set these property-value pairs, use Matlab's **set** command. The three consecutive dots are used by Matlab as a *line continuation character*. They indicate that you intend to continue the current command on the next line.

```
set(mhndl,...
    'EdgeColor',[0.6,0.6,0.6],...
    'EdgeAlpha',0.5,...
    'FaceAlpha',0.5)
```

If you type **get(mhndl)** at the prompt in the Matlab command window, you will see that these property-value pairs are changed to the settings we made above.

We will change the aspect ratio with the **axis equal** command, which makes the surface look truly spherical. We also turn off the axes with the **axis off** command.

```
axis equal
axis off
```

We reuse the parametric **equations (3.2)** form **Example 2** to compute points (x, y, z) on the spherical spiral as a function of t.

[2] You could use the variable **m_hndl** or **mhandle** or any variable you wish for the purpose of storing a handle to the mesh.

```
alpha=0.2;
t=linspace(-12*pi,12*pi,500);
x=cos(t)./sqrt(1+alpha^2*t.^2);
y=sin(t)./sqrt(1+alpha^2*t.^2);
z=alpha*t./sqrt(1+alpha^2*t.^2);
```

Instead of using the **plot3** command, we will use the **line** command. The **line** command is used to append graphics to the plot without erasing what is already there. When you use the **line** command, there is no need to use the **hold on** command.

```
lhndl=line(x,y,z)
```

Look in Matlab's command window to see that a numerical value has been assigned to the variable **lhndl**. This is a numerical identifier to the spherical spiral just plotted. Use **get(lhndl)** to obtain a list of property-value settings for the spherical spiral. We are interested in two of these pairs: **Color** and **LineWidth**, which we will now change with Matlab's **set** command.

```
set(lhndl,...
    'Color',[0.625,0,0],...
    'LineWidth',2)
```

These commands change the spherical spiral to a dark shade of red and thicken the spiral to 2pts. The result is shown in **Figure 3.3**. Because we changed the Alpha settings (transparency) of the edges and faces of the sphere, note that we can "see through" the sphere to a certain extent, making the spherical spiral on the far side of the sphere visible.

Viviani's Curve

Many new curves can be fromed from the intersection of two surfaces. For example, all of the conic sections (circle, ellipse, parabola, and hyperbola) are determined by how a plane intersects a right-circular cone (we will explore these conic sections in the exercises).

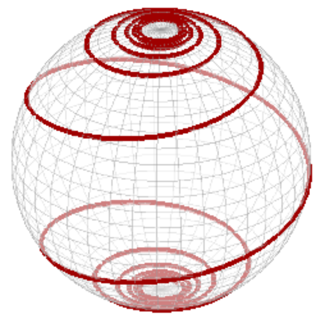


Figure 3.3. Superimposing the spherical spiral on a transparent sphere.

Such is the case with a space curve known as *Viviani's Curve*, which is the intersection of a sphere of radius $2r$ (centered at the origin) and a right-circular cylinder of radius r that is shifted r units along either the x- or y-axis.

The equation of the sphere is $x^2 + y^2 + z^2 = 4r^2$. We know that this sphere can be produced by the parametric equations

$$\begin{aligned} x &= 2r \sin\phi \cos\theta \\ y &= 2r \sin\phi \sin\theta \\ z &= 2r \cos\phi. \end{aligned} \tag{3.4}$$

We offer the following construction with few comments. It is similar to what we did in **Example 2**.

```
r=1;
phi=linspace(0,pi,30);
theta=linspace(0,2*pi,40);
[phi,theta]=meshgrid(phi,theta);

x=2*r*sin(phi).*cos(theta);
y=2*r*sin(phi).*sin(theta);
z=2*r*cos(phi);
```

Handle graphics are employed to color edges.

```
mhndl1=mesh(x,y,z)
set(mhndl1,...
    'EdgeColor',[0.6,0.6,0.6])
axis equal
axis off
```

A circle of radius r centered at the origin has equation $x^2+y^2=r^2$. If we plot the set of all (x,y,z) such that $x^2+y^2=r^2$, the result is a right-circular cylinder of radius r. Replace x with $x-r$ to get $(x-r)^2+y^2=r^2$, which will shift the cylinder r units in the x-direction. One final question remains. How can we parametrize the cylinder $(x-r)^2+y^2=r^2$?

It's fairly straightforward to show that the parametric equations

$$\begin{aligned} x &= r\cos t \\ y &= r\sin t \end{aligned} \tag{3.5}$$

produce a circle of radius r centered at the origin[3]. This can be verified with Matlab's **comet** or **plot** command[4]. To shift this r units in the x-direction add r to the equation for x to obtain

$$\begin{aligned} x &= r + r\cos t \\ y &= r\sin t. \end{aligned}$$

Thus, the parametric equations of the right-circular cylinder $(x-r)^2+y^2=r^2$ are

$$\begin{aligned} x &= r + r\cos t \\ y &= r\sin t \\ z &= z. \end{aligned}$$

The key to plotting the cylinder in Matlab is to realize that x, y, and z are functions of both t and z. That is,

$$\begin{aligned} x(t,z) &= r + r\cos t \\ y(t,z) &= r\sin t \\ z(t,z) &= z. \end{aligned} \tag{3.6}$$

Therefore, the first task is to create a grid of (t,z) pairs.

It will suffice to let $0 \le t \le 2\pi$. That should trace out the circle. If we hope to see the intersection of the sphere (radius $2r$) and the cylinder, we will need to

[3] $x^2+y^2=r^2\cos^2 t + r^2\sin^2 t = r^2(\cos^2 t+\sin^2 t) = r^2$.

[4] And **axis equal**.

have the cylinder at least as low and high in the z-directions as is the sphere of radius $2r$. Thus, limit $-2r \leq z \leq 2r$. After creating these vectors, we then create a grid of (t, z) pairs.

```
t=linspace(0,2*pi,40);
z=linspace(-2*r,2*r,20);
[t,z]=meshgrid(t,z);
```

Use the parametric **equations (3.6)** to produce points (x, y, z) on the cylinder as a function of the grid pairs (t, z).

```
x=r+r*cos(t);
y=r*sin(t);
z=z;
```

Hold the graph and plot the cylinder. Handle graphics are used to color the edges. A view is set that allows the visualization of the intersection of the sphere and cylinder. The resulting image is shown in **Figure 3.4**.

```
hold on
mhndl2=mesh(x,y,z)
set(mhndl2,...
    'EdgeColor',[0.8,0,0])
view(50,20)
```

Now, how do we get the parametrization of the curve of intersection? Recall the equations of the sphere and cylinder.

$$x^2 + y^2 + z^2 = 4r^2$$
$$(x - r)^2 + y^2 = r^2$$

If we expand and simplify the second equation, we get

$$x^2 - 2rx + y^2 = 0.$$

If we subtract this result from the first equation, we obtain

$$z^2 + 2rx = 4r^2.$$

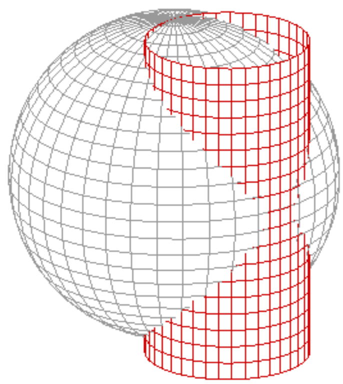


Figure 3.4. The intersection of the sphere and cylinder is called *Viviani's Curve*.

Note that all points on Viviani's curve must fall on the cylinder, where $x = r + r\cos t$. Substitute this into the last result.

$$\begin{aligned} z^2 + 2r(r + r\cos t) &= 4r^2 \\ z^2 + 2r^2 + 2r^2\cos t &= 4r^2 \\ z^2 &= 2r^2 - 2r^2\cos t. \end{aligned}$$

This can be written

$$z^2 = 4r^2\left(\frac{1-\cos t}{2}\right),$$

and the half angle identity $\sin^2(t/2) = (1-\cos t)/2$ leads to

$$z^2 = 4r^2\sin^2(t/2).$$

Normally, we should now say $z = \pm 2r\sin(t/2)$, but we will go with $z = 2r\sin(t/2)$ in the following set of parametric equations for Viviani's Curve.

$$\begin{aligned} x &= r + r\cos t \\ y &= r\sin t \\ z &= 2r\sin(t/2). \end{aligned} \tag{3.7}$$

Note that the period of $z = 2r\sin(t/2)$ is $T = 4\pi$, so if we go with only $0 \le t \le 2\pi$, we will only get positive values of z and the lower half of the curve will not be shown[5]. Thus, we use $0 \le t \le 4\pi$.

```
t=linspace(0,4*pi,200);
```

Use the parametric **equation (3.7)** to compute points (x, y, z) on Viviani's Curve in terms of t.

```
x=r+r*cos(t);
y=r*sin(t);
z=2*r*sin(t/2);
```

We plot the curve and record its "handle." We then use *handle graphics* to color the curve black ($[0, 0, 0]$ is black) and set the line width at 2 points.

```
vhndl=line(x,y,z)
set(vhndl,...
    'Color',[0,0,0],...
    'LineWidth',2)
view(50,20)
```

Setting the same view used in **Figure 3.4** produces the image in **Figure 3.5**.

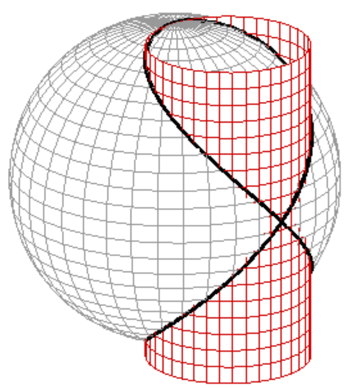


Figure 3.5. The intersection of the cylinder and sphere is the curve of Viviani.

[5] The reader should verify this statement is true.

You'll want to click the *rotate icon* on the toolbar and use the mouse to rotate and twist the figure to verify that our parametrization of Viviani's Curve is truly the intersection of the sphere and cylinder.

3.5 Exercises

In **Exercises 1-6**, perform each of the following tasks.

i. Sketch the space curve defined by the given set of parametric equations over the indicated domain.
ii. Turn the **box on**, label each axis, and provide a title.

1. On $0 \leq t \leq 1$, sketch the *Lissajous Curve*

$$\begin{aligned} x &= 3\sin(4\pi t) \\ y &= 4\sin(6\pi t) \\ z &= 5\sin(8\pi t). \end{aligned}$$

2. On $0 \leq t \leq 1$, sketch the *Lissajous Curve*

$$\begin{aligned} x &= 3\sin(4\pi t) \\ y &= 4\sin(10\pi t) \\ z &= 5\sin(14\pi t). \end{aligned}$$

3. On $0 \leq t \leq 2\pi$, sketch the *Torus Knot*

$$\begin{aligned} x &= (6.25 + 3\cos 5t)\cos 2t \\ y &= (6.25 + 3\cos 5t)\sin 2t \\ z &= 3.25\sin 5t. \end{aligned}$$

4. On $0 \leq t \leq 2\pi$, sketch the *Torus Knot*

$$\begin{aligned} x &= (7 + 2\cos 5t)\cos 3t \\ y &= (7 + 2\cos 5t)\sin 3t \\ z &= 3\sin 5t. \end{aligned}$$

5. On $0 \leq t \leq 6\pi$, sketch the *Trefoil Knot*

$$\begin{aligned} x &= \cos t(2 - \cos(2t/3)) \\ y &= \sin t(2 - \cos(2t/3)) \\ z &= -\sin(2t/3). \end{aligned}$$

6. On $0 \leq t \leq 10\pi$, sketch the *Cinquefoil Knot*

$$\begin{aligned} x &= \cos t(2 - \cos(2t/5)) \\ y &= \sin t(2 - \cos(2t/5)) \\ z &= -\sin(2t/5). \end{aligned}$$

In **Exercises 7-10**, we investigate the *conic sections*, each of which is the intersection of a plane with a right circular cone. In each exercise, perform the following tasks.

i. Draw the right circular cone having the parametric equations

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= r, \end{aligned} \tag{3.8}$$

where $0 \leq \theta \leq 2\pi$ and $-1 \leq r \leq 1$. Use handle graphics to set the **EdgeColor** to a shade of gray.
ii. Execute **hold on** to "hold" the surface plot of the right circular cone. Superimpose the plot of the given plane over the given domain, then use Matlab's handle graphics to set the **EdgeColor** to a single color of your choice.
iii. Click the rotate icon on the figure toolbar and rotate the figure to a view that emphasizes the conic section (curve of intersection) and note the azimuth and elevation. Use

these components in the **view(ax,el)** in your script to obtain an identical view. Label the axes and provide a title that includes the name the conic section that is the intersection of the given plane and the right circular cone.

7. Sketch the plane $z = 1/2$ over the domain

$$D = \{(x, y) : -1 \leq x, y \leq 1\}.$$

8. Sketch the plane $z = 0.4y + 0.5$ over the domain

$$D = \{(x, y) : -1 \leq x, y \leq 1\}.$$

9. Sketch the plane $z = y+0.25$ over the domain

$$D = \{(x, y) : -1 \leq x, y \leq 1\}.$$

10. Sketch the plane $x = 0.5$ over the domain

$$D = \{(y, z) : -1 \leq y, z \leq 1\}.$$

---

11. Sketch the torus defined by the parametric equations

$$\begin{aligned} x &= (7 + 2\cos u)\cos v \\ y &= (7 + 2\cos u)\sin v \\ z &= 3\sin u. \end{aligned}$$

Set the **EdgeColor** to a shade of gray and add transparency by setting both **FaceAlpha** and **EdgeAlpha** equal to 0.5. Set the **axis equal**. Use Matlab's **line** command to superimpose the torus knot having parmetric equations

$$\begin{aligned} x &= (7 + 2\cos 5t)\cos 2t \\ y &= (7 + 2\cos 5t)\sin 2t \\ z &= 3\sin 5t \end{aligned}$$

over the time domain $0 \leq t \leq 2\pi$. Use handle graphics to set the **Color** of the torus knot to a color of your choice and set the **LineWidth** to a thickness of 2 points.

12. Sketch the torus defined by the parametric equations

$$\begin{aligned} x &= (8 + 2\cos u)\cos v \\ y &= (8 + 2\cos u)\sin v \\ z &= 3\sin u. \end{aligned}$$

Set the **EdgeColor** to a shade of gray and add transparency by setting both **FaceAlpha** and **EdgeAlpha** equal to 0.5. Set the **axis equal**. Use Matlab's **line** command to superimpose the torus knot having parmetric equations

$$\begin{aligned} x &= (8 + 2\cos 11t)\cos 3t \\ y &= (8 + 2\cos 11t)\sin 3t \\ z &= 3\sin 11t \end{aligned}$$

over the time domain $0 \leq t \leq 2\pi$. Use handle graphics to set the **Color** of the torus knot to a color of your choice and set the **LineWidth** to a thickness of 2 points.

13. Sketch the cone defined by the parametric equations

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

where $0 \leq r \leq 1$ and $0 \leq \theta \leq 2\pi$. Set the **EdgeColor** to a shade of gray and add transparency by setting both **FaceAlpha** and **EdgeAlpha** equal to 0.5. Set the **axis equal**. Use Matlab's **line** command to superimpose the *conical spiral* having parmetric equations

$$\begin{aligned} x &= t\cos 20t \\ y &= t\sin 20t \\ z &= t \end{aligned}$$

over the time domain $0 \le t \le 1$. Use handle graphics to set the **Color** of the conical spiral to a color of your choice and set the **LineWidth** to a thickness of 2 points.

14. Sketch the cylinder defined by the parametric equations

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

where $0 \le z \le 5$ and $0 \le \theta \le 2\pi$. Set the **EdgeColor** to a shade of gray and add transparency by setting both **FaceAlpha** and **EdgeAlpha** equal to 0.5. Set the **axis equal**. Use Matlab's **line** command to superimpose the helix having parmetric equations

$$\begin{aligned} x &= 2\cos 5t \\ y &= 2\sin 5t \\ z &= t, \end{aligned}$$

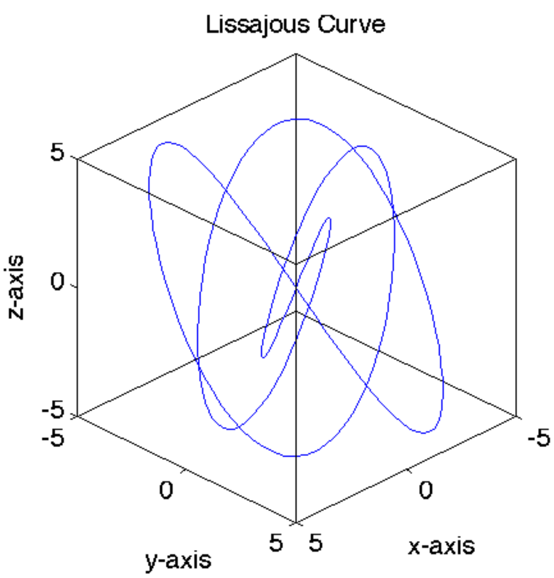
over the time domain $0 \le t \le 5$. Use handle graphics to set the **Color** of the helix to a color of your choice and set the **LineWidth** to a thickness of 2 points.

15. **Challenge**. The intersection of a sphere and a plane that passes through the center of the sphere is called a *great circle*. Sketch the equation of a sphere of radius 1, then sketch the plane $y = x$ and show that the intersection is a great circle. Find the parametric equations of this great circle and add it to your plot.

3.5 Answers

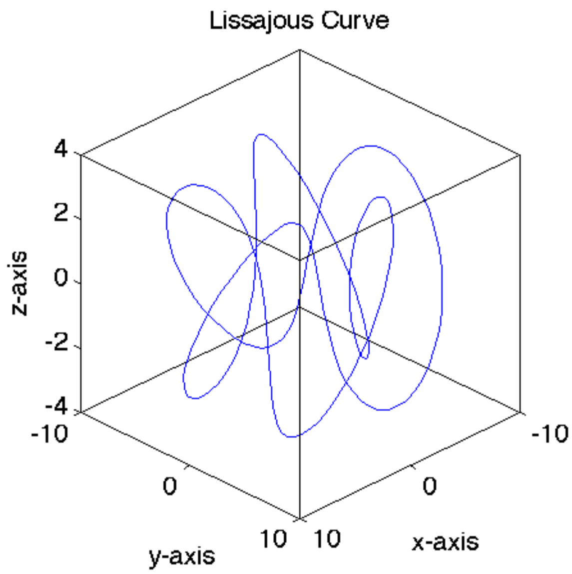
1.

```
t=linspace(0,1,200);
x=3*sin(4*pi*t);
y=4*sin(6*pi*t);
z=5*sin(8*pi*t);
plot3(x,y,z)
box on
view(135,30)
```



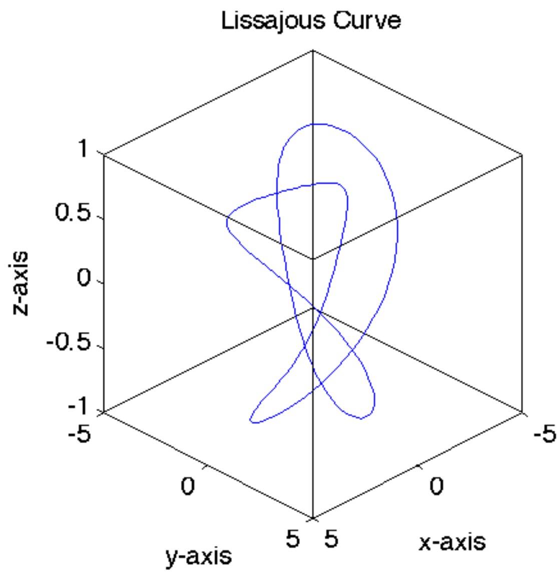
3.

```
t=linspace(0,2*pi,200);
x=(6.25+3*cos(5*t)).*cos(2*t);
y=(6.25+3\cos(5*t)).*sin(2*t);
z=3.25*sin(5*t);
plot3(x,y,z)
box on
view(135,30)
```



5.

```
t=linspace(0,6*pi,200);
x=cos(t).*(2-cos(2*t/3));
y=sin(t).*(2-cos(2*t/3));
z=-sin(2*t/3);
plot3(x,y,z)
box on
view(135,30)
```



7. Set the grid for the cone.

```
theta=linspace(0,2*pi,40);
r=linspace(-1,1,30);
[theta,r]=meshgrid(theta,r);
```

Use the parametric equations to compute x, y, and z.

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

Draw the right circular cone in a shade of gray.

```
mhndl=mesh(x,y,z)
set(mhndl,...
    'EdgeColor',[.6,.6,.6])
```

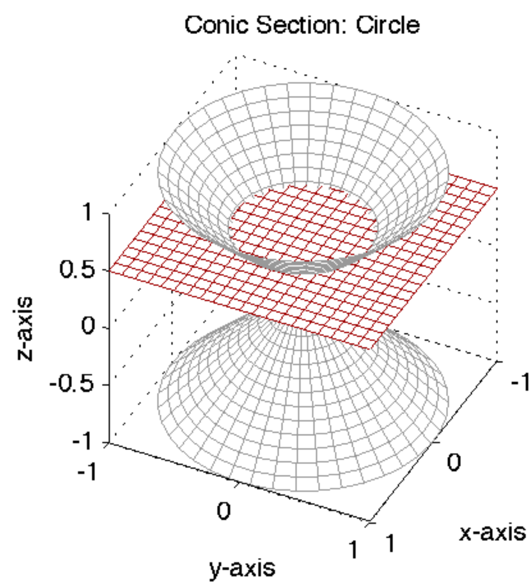
Hold the surface plot and draw the plane $z = 1/2$.

```
hold on
[x,y]=meshgrid(-1:0.2:1);
z=0.5*ones(size(x));
phndl=mesh(x,y,z);
set(phndl,...
    'EdgeColor',[0.625,0,0])
```

Adjust orientation.

```
axis equal
view(116,38)
```

Annotate the plot in the usual manner. The intersection is a circle, as indicated in the title.



9. Set the grid for the cone.

```
theta=linspace(0,2*pi,40);
r=linspace(-1,1,30);
[theta,r]=meshgrid(theta,r);
```

Use the parametric equations to

compute x, y, and z.

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

Draw the right circular cone in a shade of gray.

```
mhndl=mesh(x,y,z)
set(mhndl,...
    'EdgeColor',[.6,.6,.6])
```

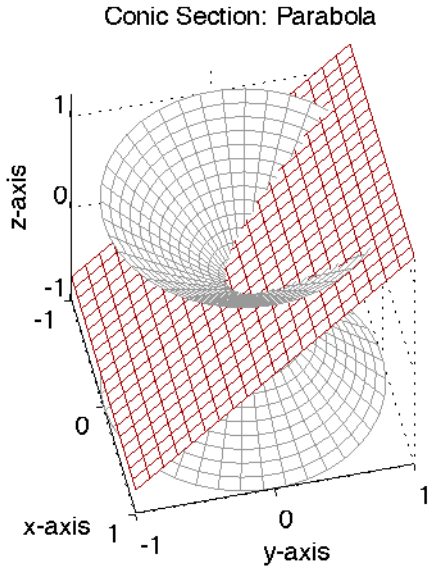
Hold the surface plot and draw the plane $z = y + 0.25$.

```
hold on
[[x,y]=meshgrid(-1:0.1:1);
z=y+0.25;
phndl=mesh(x,y,z);
set(phndl,...
    'EdgeColor',[0.625,0,0])
```

Adjust orientation.

```
axis equal
view(77,50)
```

Annotate the plot in the usual manner. The intersection is a parabola, as indicated in the title.



11. Set parameters a, b, and c.

```
a=7;b=2;c=3;
```

Set the grid for the torus.

```
u=linspace(0,2*pi,20);
v=linspace(0,2*pi,40);
[u,v]=meshgrid(u,v);
```

Use the parametric equations to compute x, y, and z.

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

Draw the torus in a shade of gray and add transparency. Set the perspective with **axis equal**.

```
mhndl=mesh(x,y,z);
set(mhndl,...
    'EdgeColor',[.6,.6,.6],...
    'FaceAlpha',0.5,...
    'EdgeAlpha',0.5);
axis equal
```

Compute x, y, and z for the torus knot over the requested time domain.

```
t=linspace(0,2*pi,200);
x=(a+b*cos(5*t)).*cos(2*t);
y=(a+b*cos(5*t)).*sin(2*t);
z=c*sin(5*t);
```

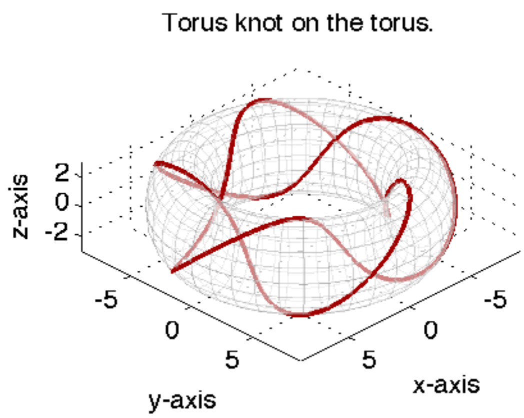
Plot the torus knot and change its color and linewidth.

```
lhndl=line(x,y,z);
set(lhndl,...
    'Color',[.625,0,0],...
    'LineWidth',2)
```

Adjust orientation.

```
view(135,30)
```

Annotate the plot in the usual manner.



13. Set the grid for the cone.

```
r=linspace(0,1,20);
theta=linspace(0,2*pi,40);
[r,theta]=meshgrid(r,theta);
```

Use the parametric equations to compute x, y, and z.

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

Draw the cone in a shade of gray and add transparency. Set the perspective with **axis equal**.

```
mhndl=mesh(x,y,z);
set(mhndl,...
    'EdgeColor',[.6,.6,.6],...
    'FaceAlpha',0.5,...
    'EdgeAlpha',0.5);
axis equal
```

Compute x, y, and z for the conical spiral over the requested time domain.

```
t=linspace(0,1,200);
x=t.*cos(20*t);
y=t.*sin(20*t);
z=t;
```

Plot the concical spiral and change its color and linewidth.

```
lhndl=line(x,y,z);
set(lhndl,...
    'Color',[.625,0,0],...
    'LineWidth',2)
```

Adjust orientation.

```
view(135,30)
```

Annotate the plot in the usual manner.