In[1]:=
```
NotebookDirectory[]
```

Out[1]=
```
C:\Dropbox\307_Files\2025\
```

Before reading this notebook evaluate the entire notebook by pressing the keyboard shortcut Alt+v+o or using the menu item:
Evaluation ▹ Evaluate Notebook

You can open all the cells below by highlighting the outermost cell and pressing the keyboard shortcut: Shift+Ctrl+{

The Beauty of Trigonometry

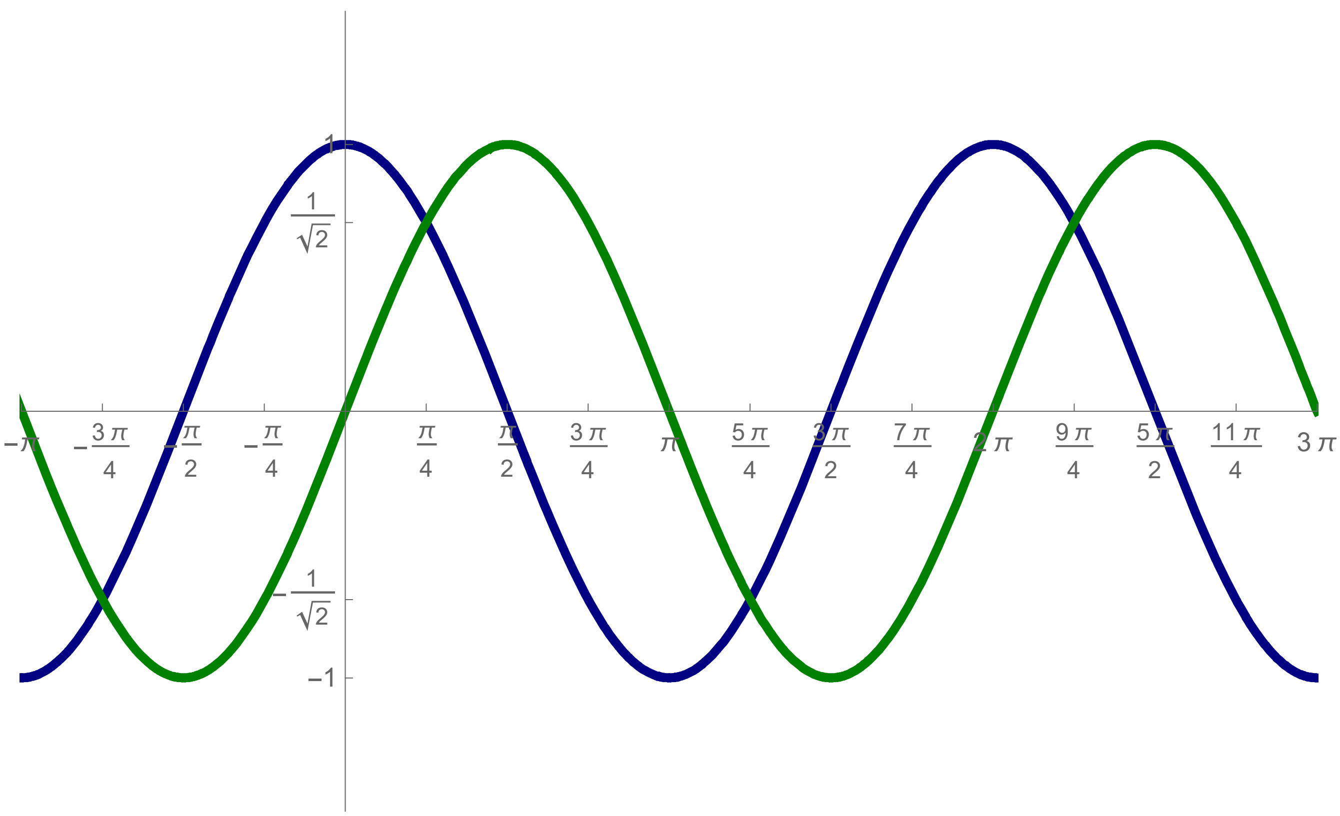
The functions Cosine and Sine

Here they are, in all their glory, Cos and Sin:

reproduce the picture below (1)

In[2]:=
```
Plot[ (* here starts Plot *)
  {Cos[x], Sin[x]}, (* ploted are Cos and Sin *)
  {x, -3 π, 3 π}, (* This is the domain for the variable *)
  (* here start Options *)
  PlotStyle → { (* here starts PlotStyle,
    choosing colors and thickness of the graphs *)
    {Thickness[0.007], RGBColor[0, 0, 0.5]},
    {Thickness[0.007], RGBColor[0, 0.5, 0]}
    (* here ends PlotStyle *)},
  PlotRange → {{-Pi, 3 Pi}, {-1.5, 1.5}}, (* choosing the plot range,
  first horizontal, then vertical *)
  Ticks → {Range[-7 π, 7 π, π/4], {-1, -√2/2, 0, √2/2, 1}},
  (* choosing the ticks on the coordinate axes, first x-axis,
  then y-axis *)
  ImageSize → 500
 (* here ends Plot *)]
```

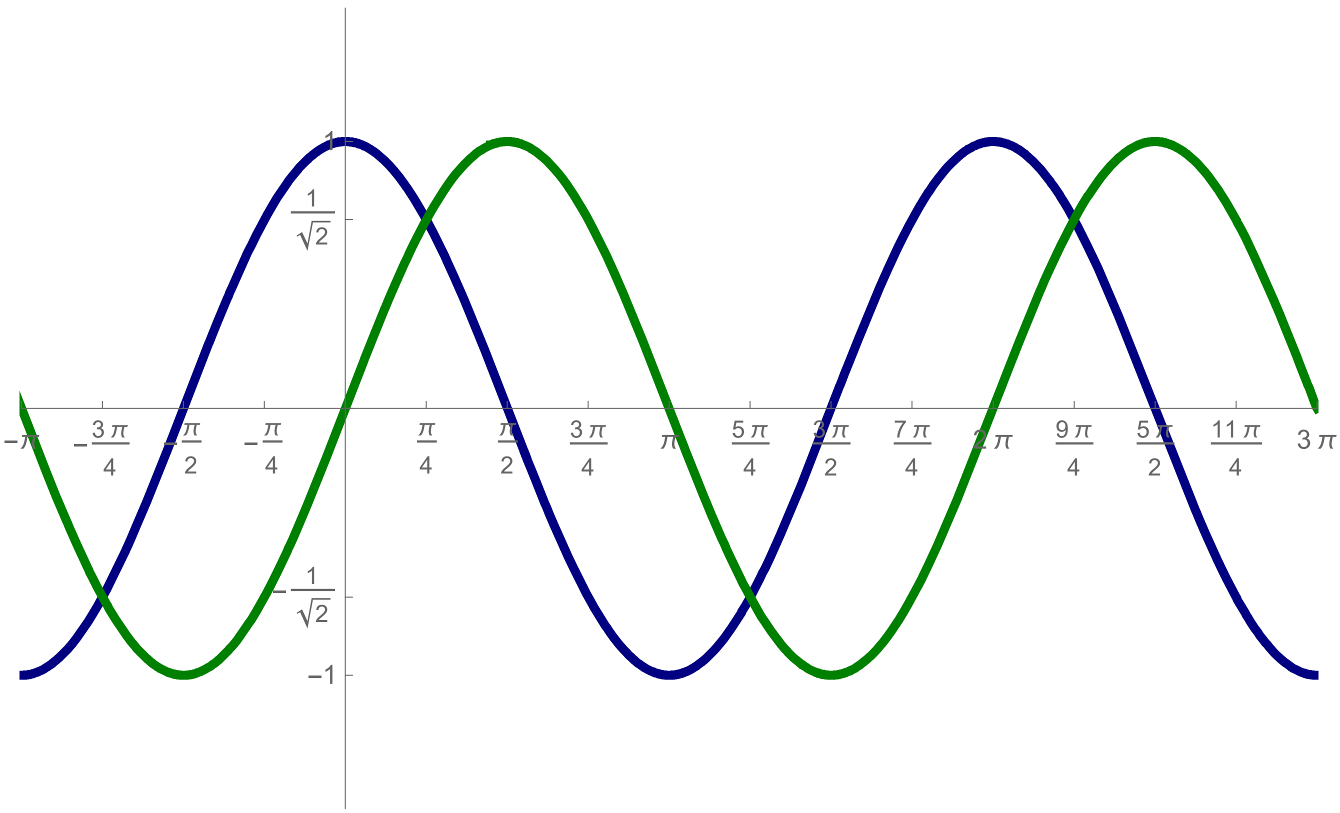
Out[2]=



Above, we see that the cosine is just a shift of the sine by Pi/2.

In[3]:=
```
Plot[
 {Sin[x + Pi / 2], Sin[x]},
 {x, -3 π, 3 π},
  PlotStyle → {
   {Thickness[0.007], RGBColor[0, 0, 0.5]},
   {Thickness[0.007], RGBColor[0, 0.5, 0]}
         },
 PlotRange → {{-Pi, 3 Pi}, {-1.5, 1.5}},
 Ticks → {Range[-7 π, 7 π, π/4], {-1, -√2/2, 0, √2/2, 1}},
 ImageSize → 500
]
```

Out[3]=



Why only one shift? Why not many? We need a new command, called Table

In[4]:=
```
Table[Sin[x], {x, 1, 10, 1}]
```

Out[4]=
```
{Sin[1], Sin[2], Sin[3], Sin[4],
 Sin[5], Sin[6], Sin[7], Sin[8], Sin[9], Sin[10]}
```

The next table will list pairs of the values of the variable x and the values of the sine function at that value of x. You will see some values of the sine that you have not seen before. For example the value of sine at x = Pi/12 is, you can read below ...

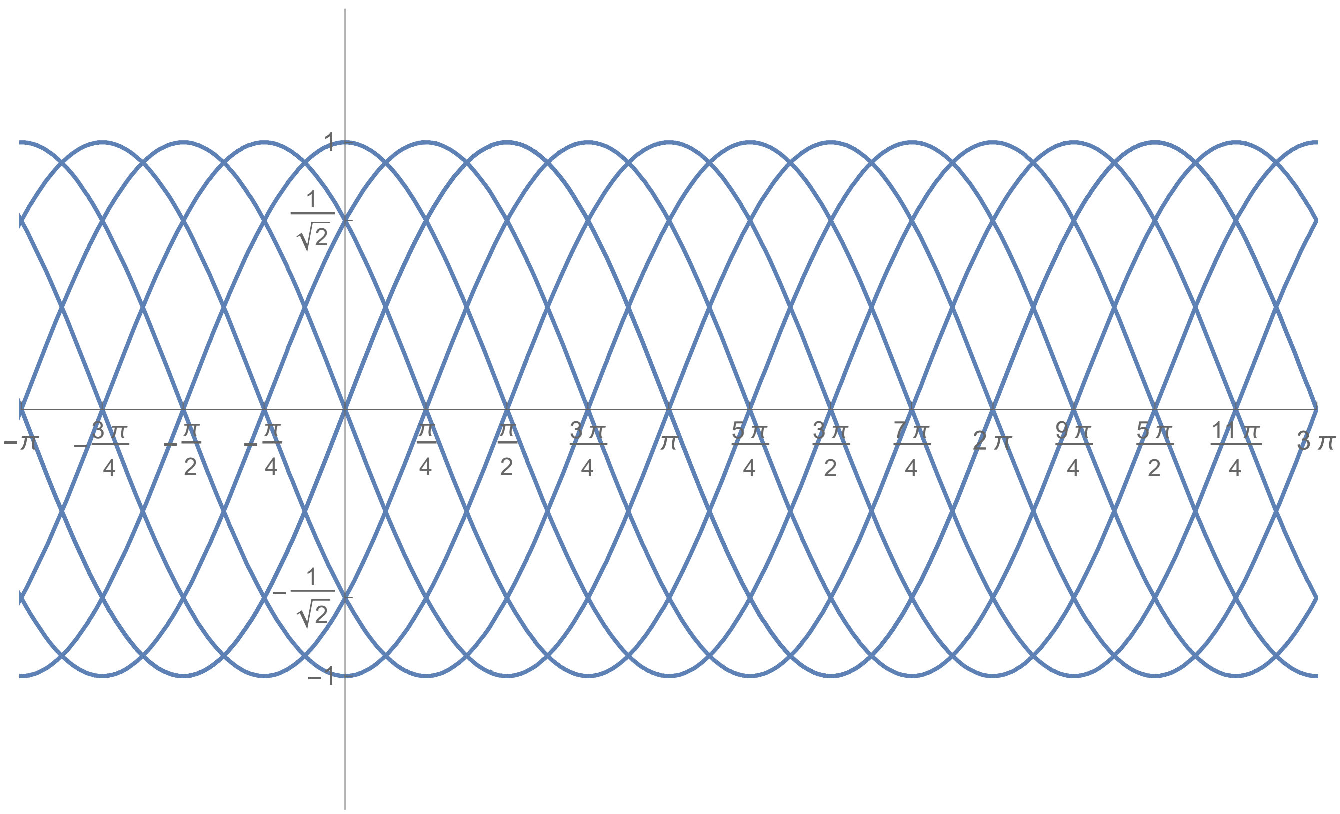
In[5]:= `Table[{x, Sin[x]}, {x, 0, 2 Pi, Pi/12}]`

Out[5]= $\left\{\{0, 0\}, \left\{\frac{\pi}{12}, \frac{-1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{\pi}{6}, \frac{1}{2}\right\}, \left\{\frac{\pi}{4}, \frac{1}{\sqrt{2}}\right\}, \left\{\frac{\pi}{3}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{5\pi}{12}, \frac{1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{\pi}{2}, 1\right\}, \left\{\frac{7\pi}{12}, \frac{1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{2\pi}{3}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{3\pi}{4}, \frac{1}{\sqrt{2}}\right\}, \left\{\frac{5\pi}{6}, \frac{1}{2}\right\}, \left\{\frac{11\pi}{12}, \frac{-1+\sqrt{3}}{2\sqrt{2}}\right\}, \{\pi, 0\}, \left\{\frac{13\pi}{12}, -\frac{-1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{7\pi}{6}, -\frac{1}{2}\right\}, \left\{\frac{5\pi}{4}, -\frac{1}{\sqrt{2}}\right\}, \left\{\frac{4\pi}{3}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{17\pi}{12}, -\frac{1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{3\pi}{2}, -1\right\}, \left\{\frac{19\pi}{12}, -\frac{1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{\frac{5\pi}{3}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{7\pi}{4}, -\frac{1}{\sqrt{2}}\right\}, \left\{\frac{11\pi}{6}, -\frac{1}{2}\right\}, \left\{\frac{23\pi}{12}, -\frac{-1+\sqrt{3}}{2\sqrt{2}}\right\}, \{2\pi, 0\}\right\}$

Let us plot many shifts, below we plot 24 of them.

In[6]:=
```
Plot[
 Table[Sin[x + sh], {sh, 0, 2 Pi, Pi/4}],
 {x, -3 π, 3 π},
 PlotRange → {{-Pi, 3 Pi}, {-1.5, 1.5}},
 Ticks → {Range[-7 π, 7 π, π/4], {-1, -√2/2, 0, √2/2, 1}},
 ImageSize → 500
]
```

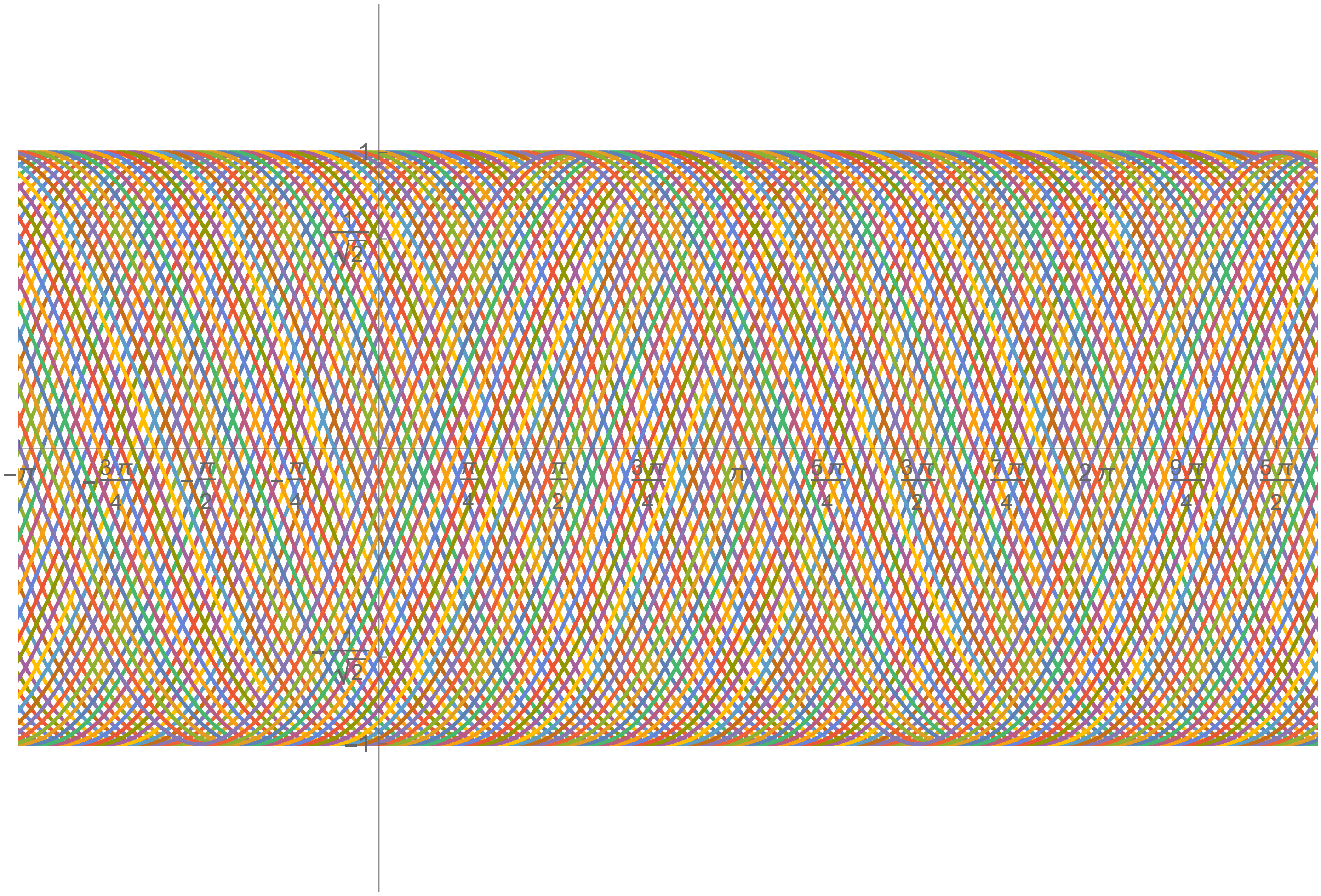
Out[6]=



A small change, I wrap Table[] in Evaluate[] and that tells Mathematica to choose different colors for the shifts.

In[79]:=
```
Plot[
 Evaluate[Table[Sin[x + sh], {sh, 0, 2 Pi, Pi/32}]],
 {x, -3 π, 3 π},
 PlotRange → {{-Pi, 3 Pi}, {-1.5, 1.5}},
 Ticks → {Range[-7 π, 7 π, π/4], {-1, -√2/2, 0, √2/2, 1}},
 ImageSize → 600
]
```

Out[79]=



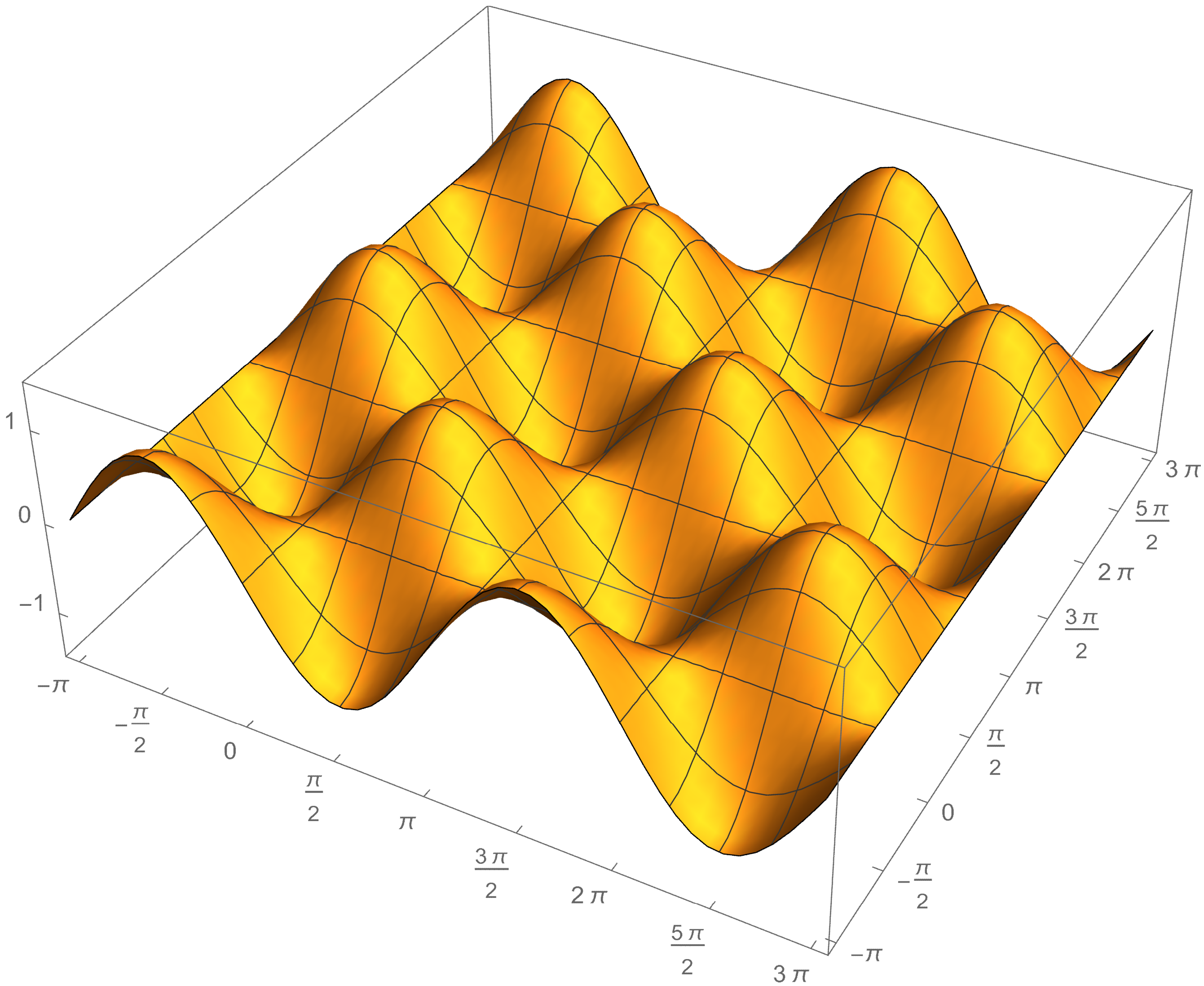
There are many other Options; to see them all remove the comment out

In[8]:=
```
(* Options[Plot] *)
```

Below is a possible two variable version of two trigonometric functions

In[9]:=
```
Plot3D[Cos[y] Sin[x], {x, -π, 3 π}, {y, -π, 3 π},
  PlotRange → {-3/2, 3/2}, PlotPoints → {51, 51},
  Ticks → {Range[-3 π, 3 π, π/2], Range[-3 π, 3 π, π/2], Range[-3, 3]},
  ImageSize → 500]
```

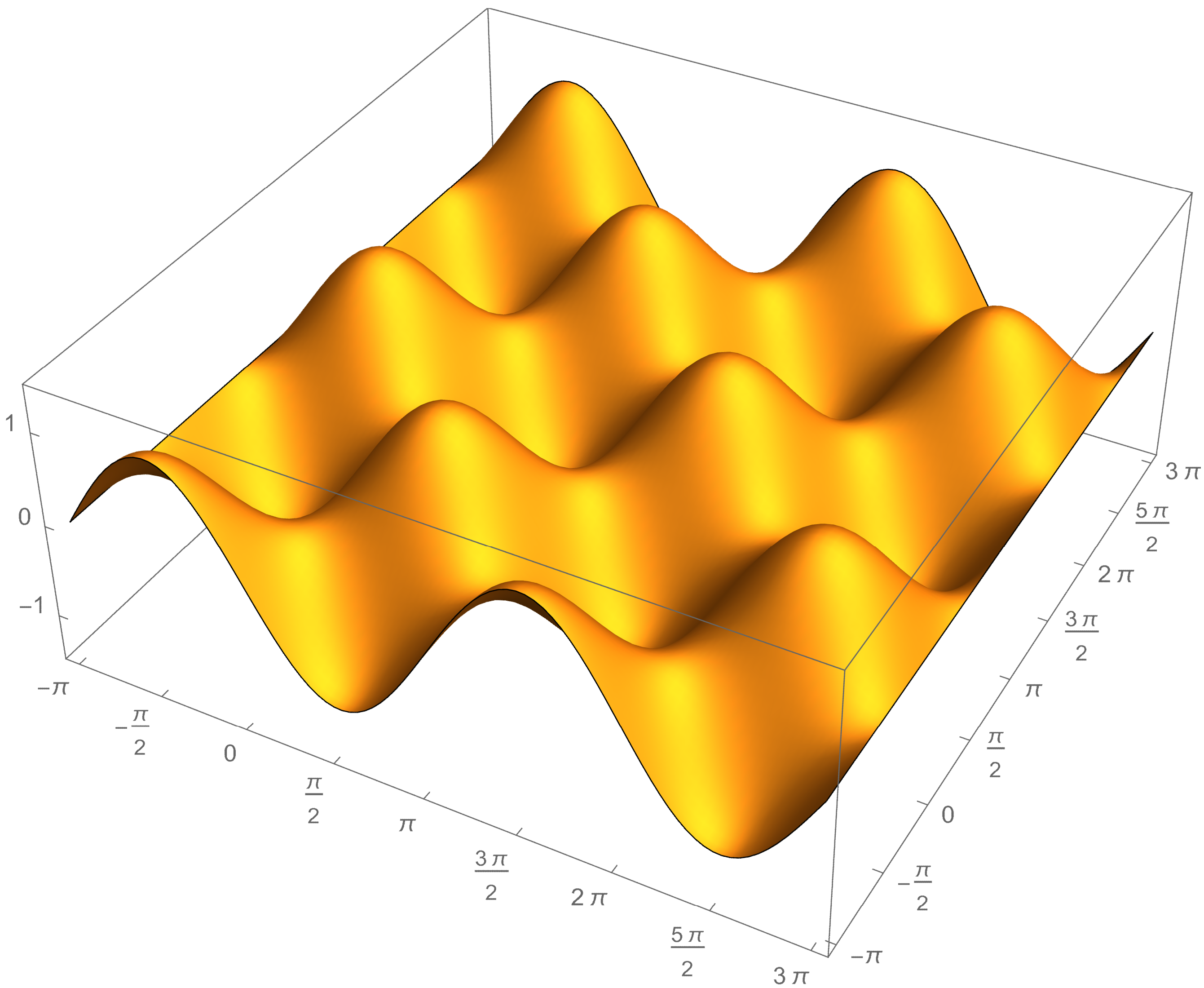
Out[9]=



reproduce the picture below (2)

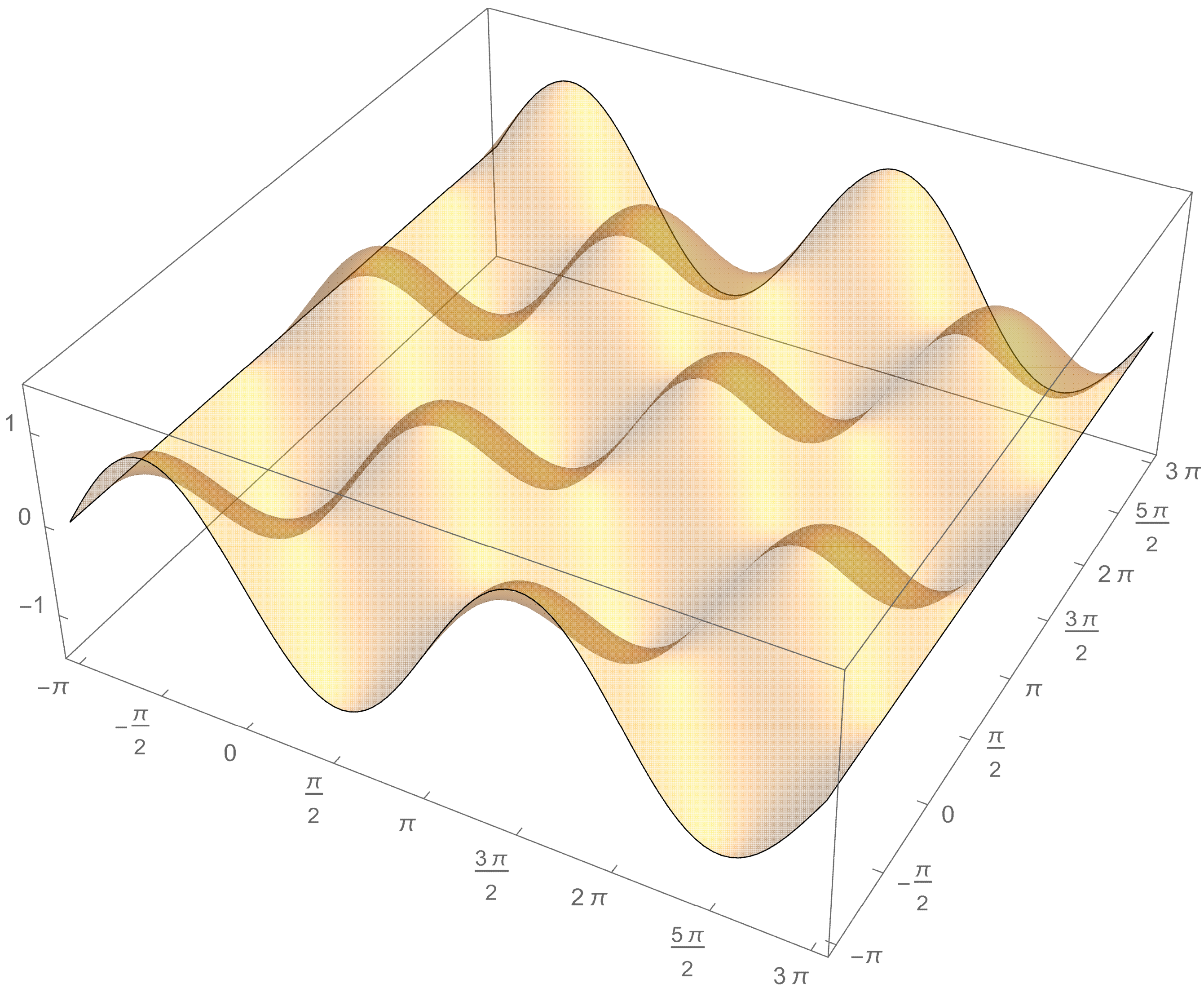
```
In[10]:= Plot3D[Cos[y] Sin[x], {x, -π, 3 π}, {y, -π, 3 π}, PlotRange → {-1.5, 1.5},
   PlotPoints → {91, 91}, Mesh → False,
   Ticks → {Range[-3 π, 3 π, π/2], Range[-3 π, 3 π, π/2], Range[-3, 3]},
   ImageSize → 500]
```

Out[10]=



In[11]:=
```
Plot3D[Cos[y] Sin[x], {x, -π, 3 π}, {y, -π, 3 π},
 PlotStyle → {Opacity[0.3]},
 PlotPoints → {91, 91}, Mesh → False,
 PlotRange → {-1.5, 1.5},
 Ticks → {Range[-3 π, 3 π, π/2], Range[-3 π, 3 π, π/2], Range[-3, 3]},
 ImageSize → 500]
```

Out[11]=



There are many other Options for Plot3D; to see them all remove the comment out

In[12]:=
```
(* Options[Plot3D] *)
```

Cosine and Sine parametrize the Unit Circle

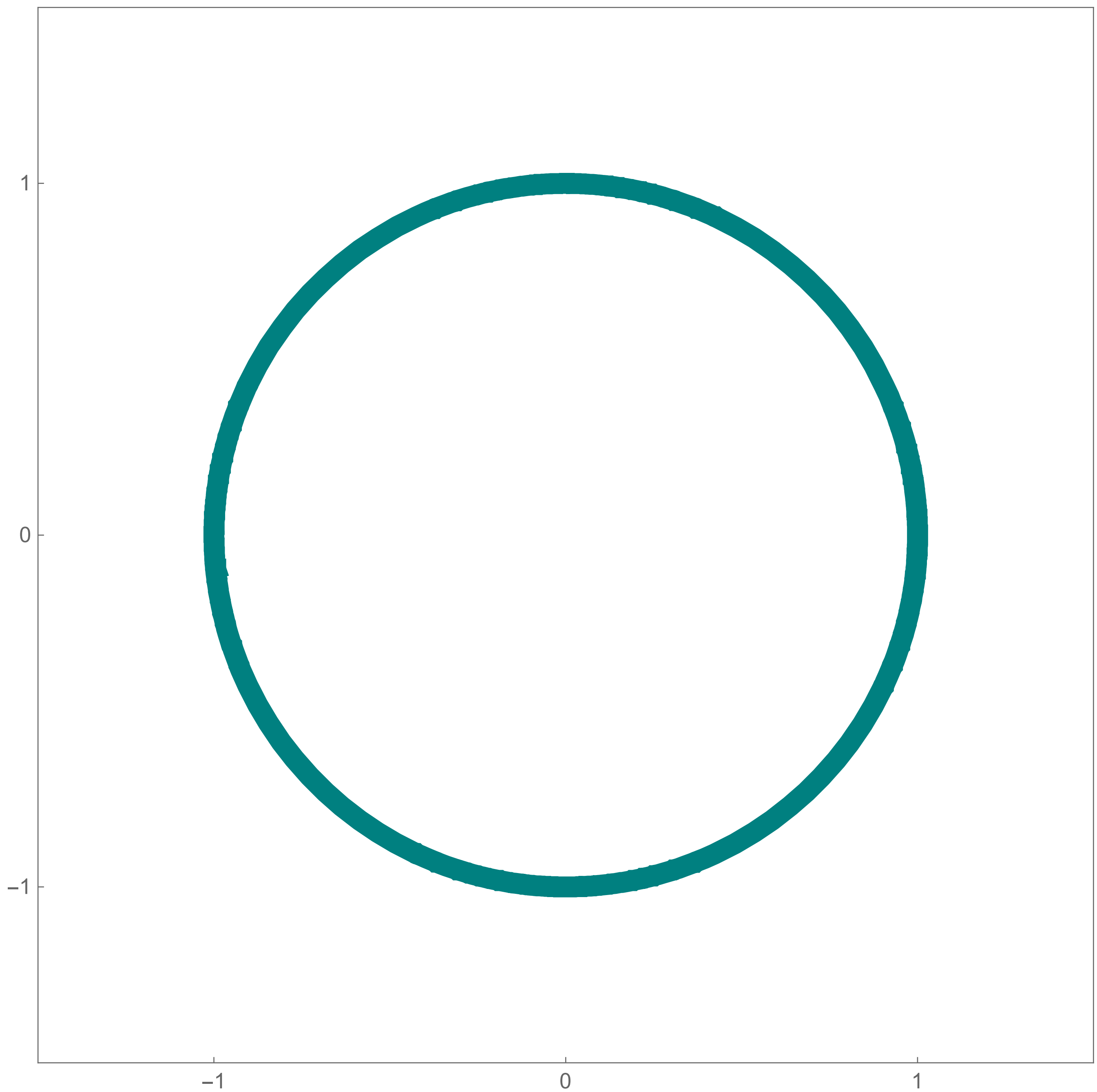
The most important property of cosine and sine is that they provide the parametric equations of the unit circle.

To plot a curve given by parametric equations we use ParametricPlot[]

reproduce the picture below (3)

In[13]:=
```
ParametricPlot[
 {Cos[t], Sin[t]}, {t, 0, 2*Pi},
 PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 500
]
```

Out[13]=

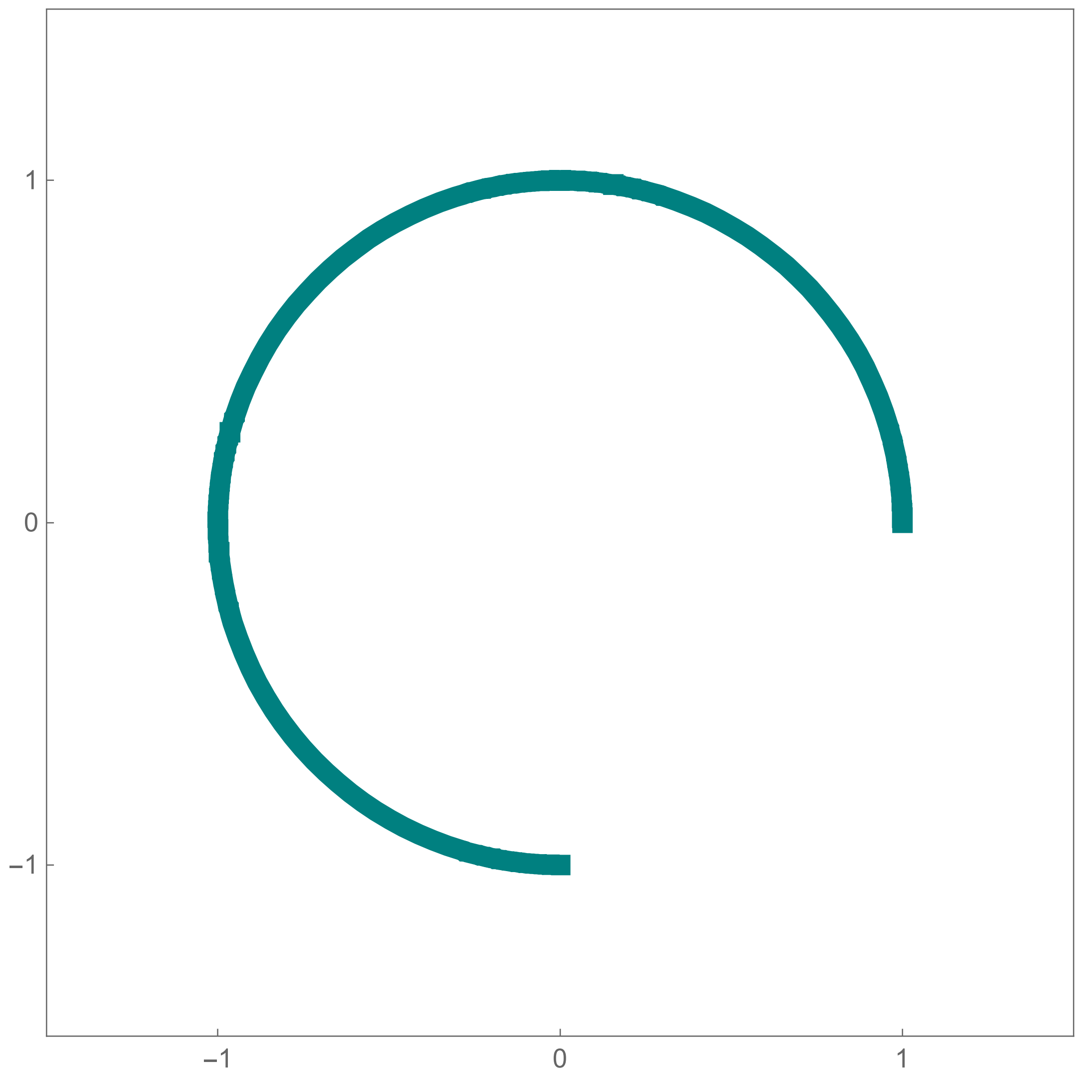


Please be aware of the role of the parameter t. If we restrict t to the interval from 0 to Pi we get the top half of the unit circle.

reproduce the picture below (4)

In[14]:=
```
ParametricPlot[
 {Cos[t], Sin[t]}, {t, 0, 3 Pi / 2},
 PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

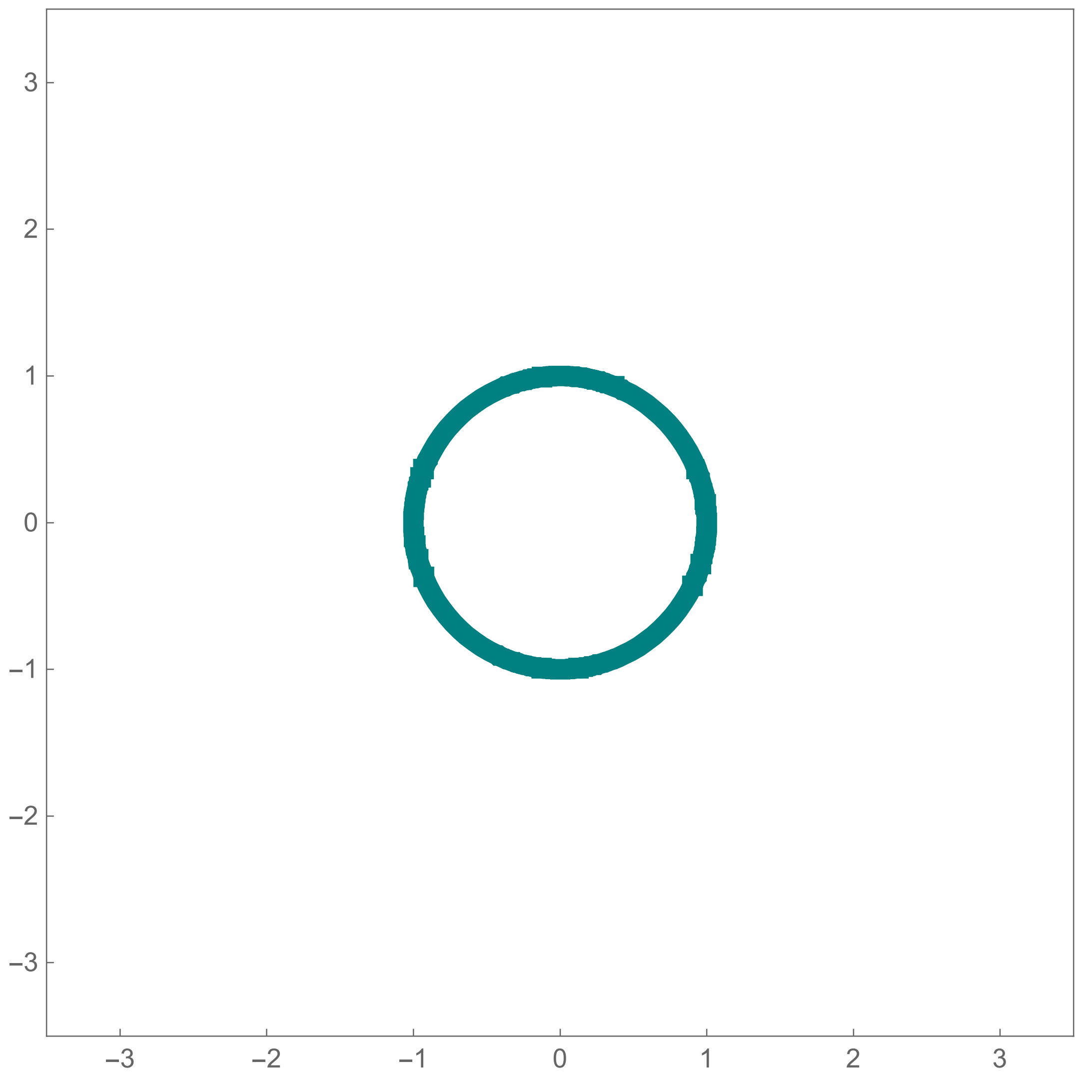
Out[14]=



We need a bigger PlotRange to explore how one can increase or decrease the radius:

In[15]:=
```
ParametricPlot[
 1 {Cos[t], Sin[t]}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

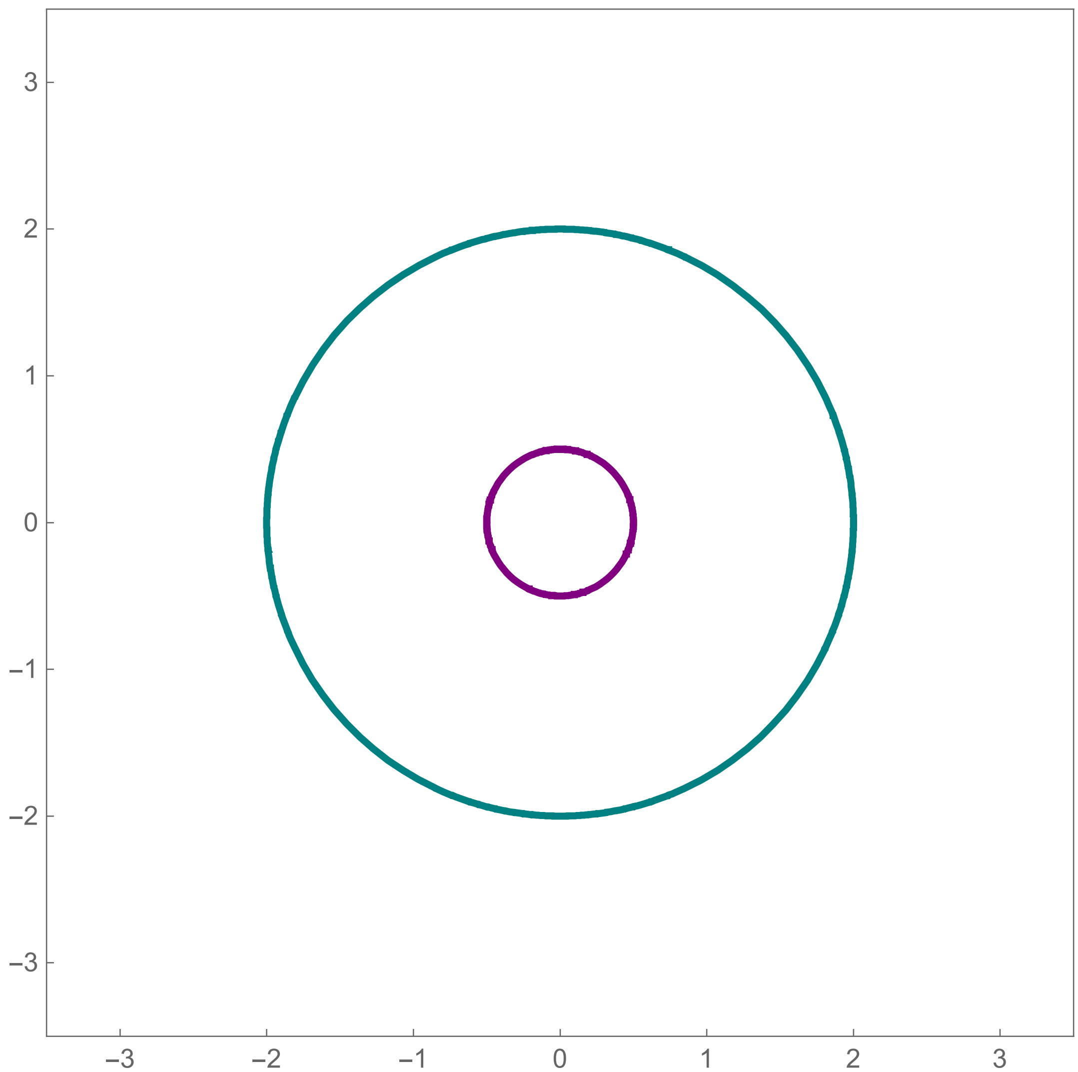
Out[15]=



And more than one circle:

In[16]:=
```
ParametricPlot[
 {2 {Cos[t], Sin[t]}, 1/2 {Cos[t], Sin[t]}}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]},
   {Thickness[0.007], RGBColor[0.5, 0, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

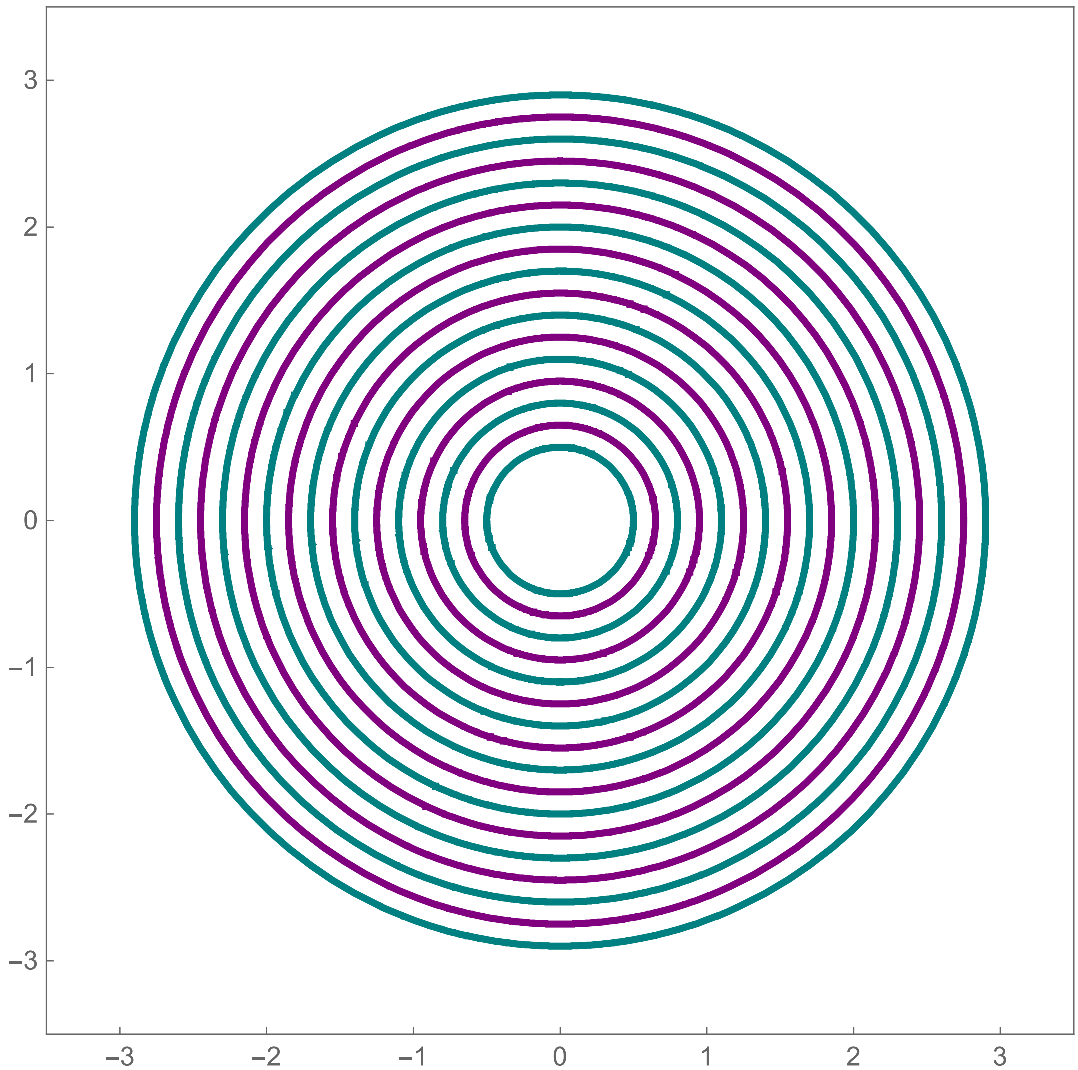
Out[16]=



Now many circles with different radii

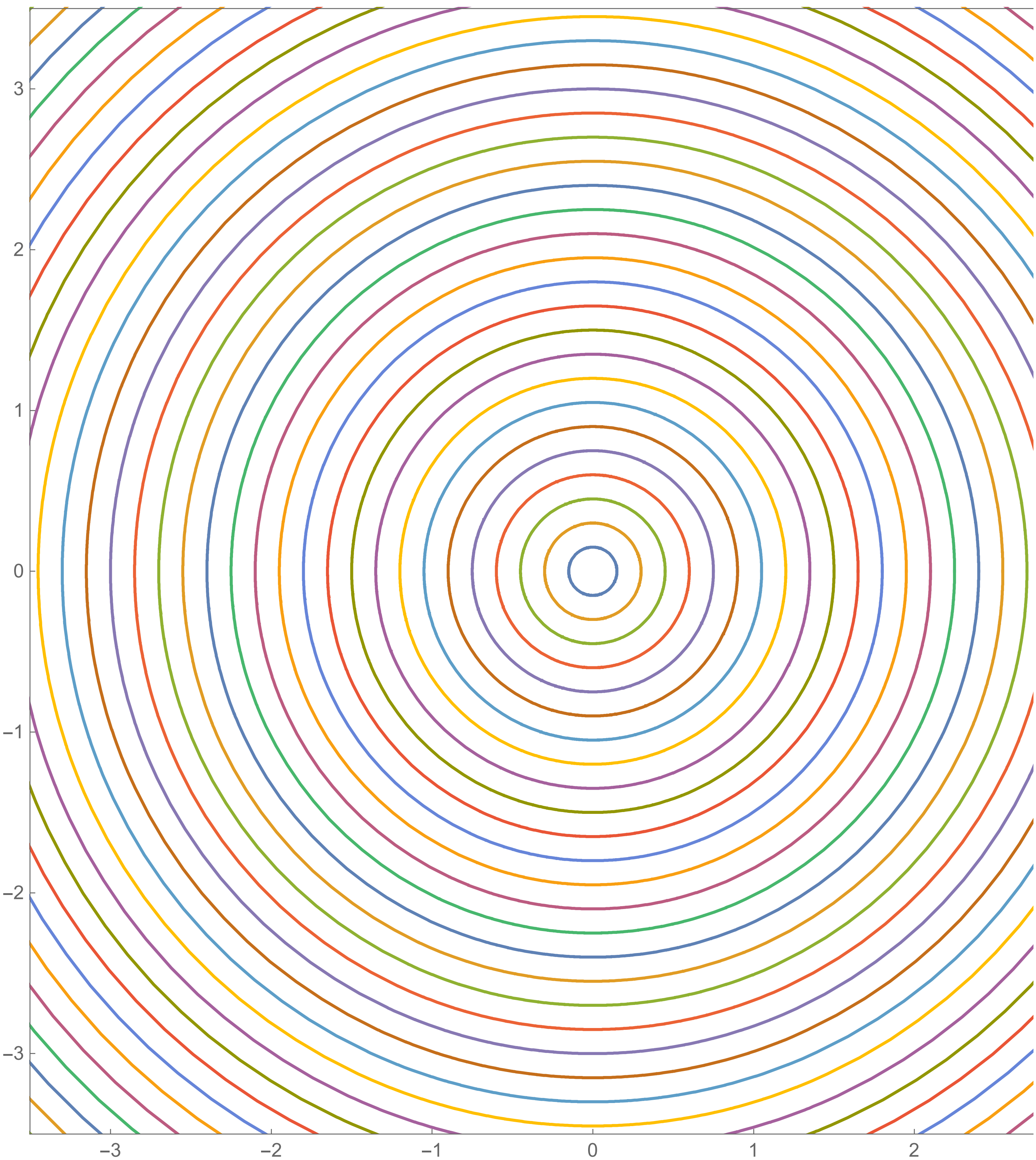
In[17]:=
```
ParametricPlot[
 Evaluate[Table[rr {Cos[t], Sin[t]}, {rr, 0.5, 3, 0.15}]], {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]},
   {Thickness[0.007], RGBColor[0.5, 0, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

Out[17]=



In[18]:=
```
ParametricPlot[
 Evaluate[Table[rr {Cos[t], Sin[t]}, {rr, 0.15, 6, 0.15}]], {t, 0, 2 Pi},
 PlotPoints → 101, PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 600
]
```

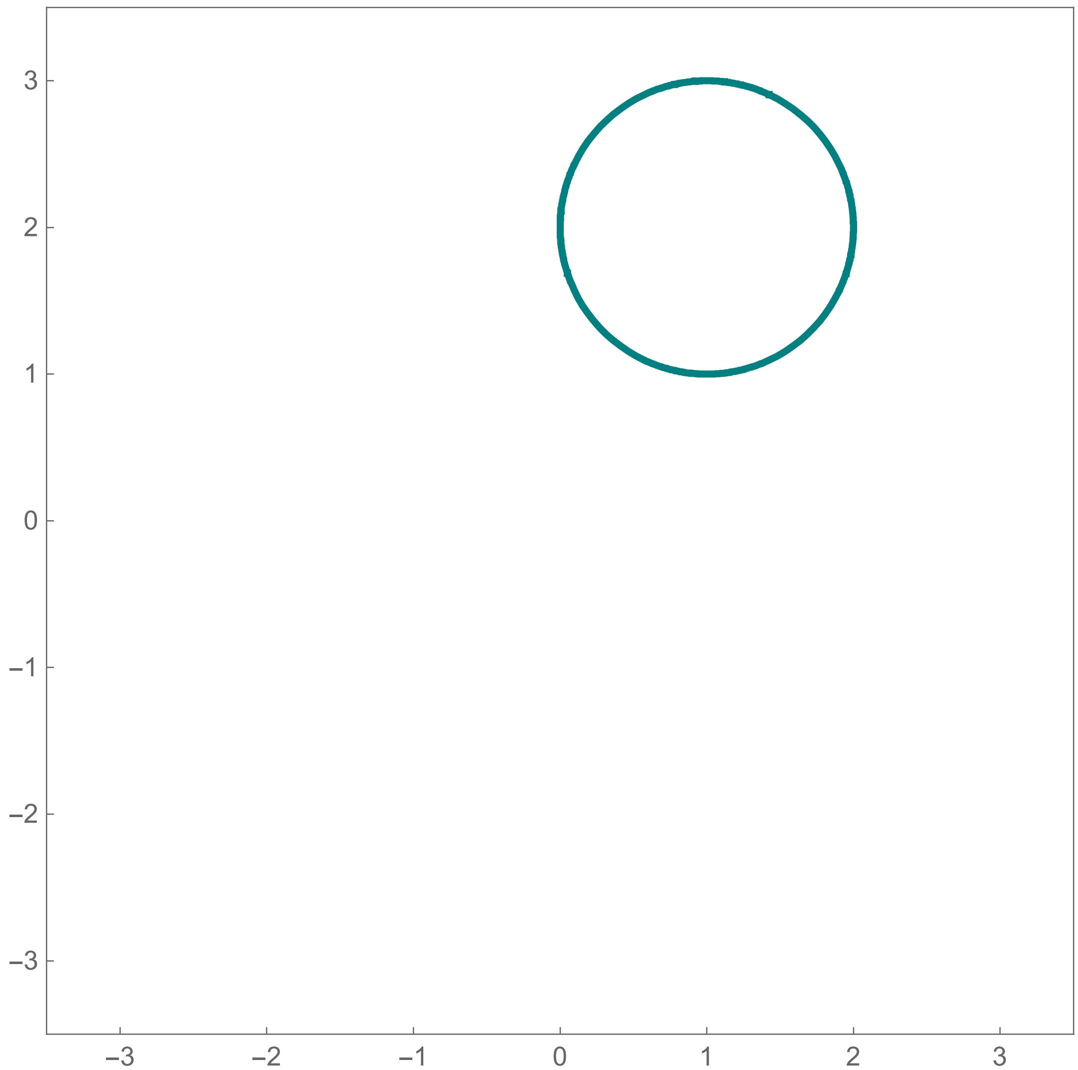
Out[18]=



We can move the circle anywhere in the plane. In the formula below you should think of {1,2} as a vector that moves the circle from the origin to the point {1,2} which becomes the new center.

In[19]:=
```
ParametricPlot[
 {1, 2} + {Cos[t], Sin[t]}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]}},
 PlotPoints → 101, PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

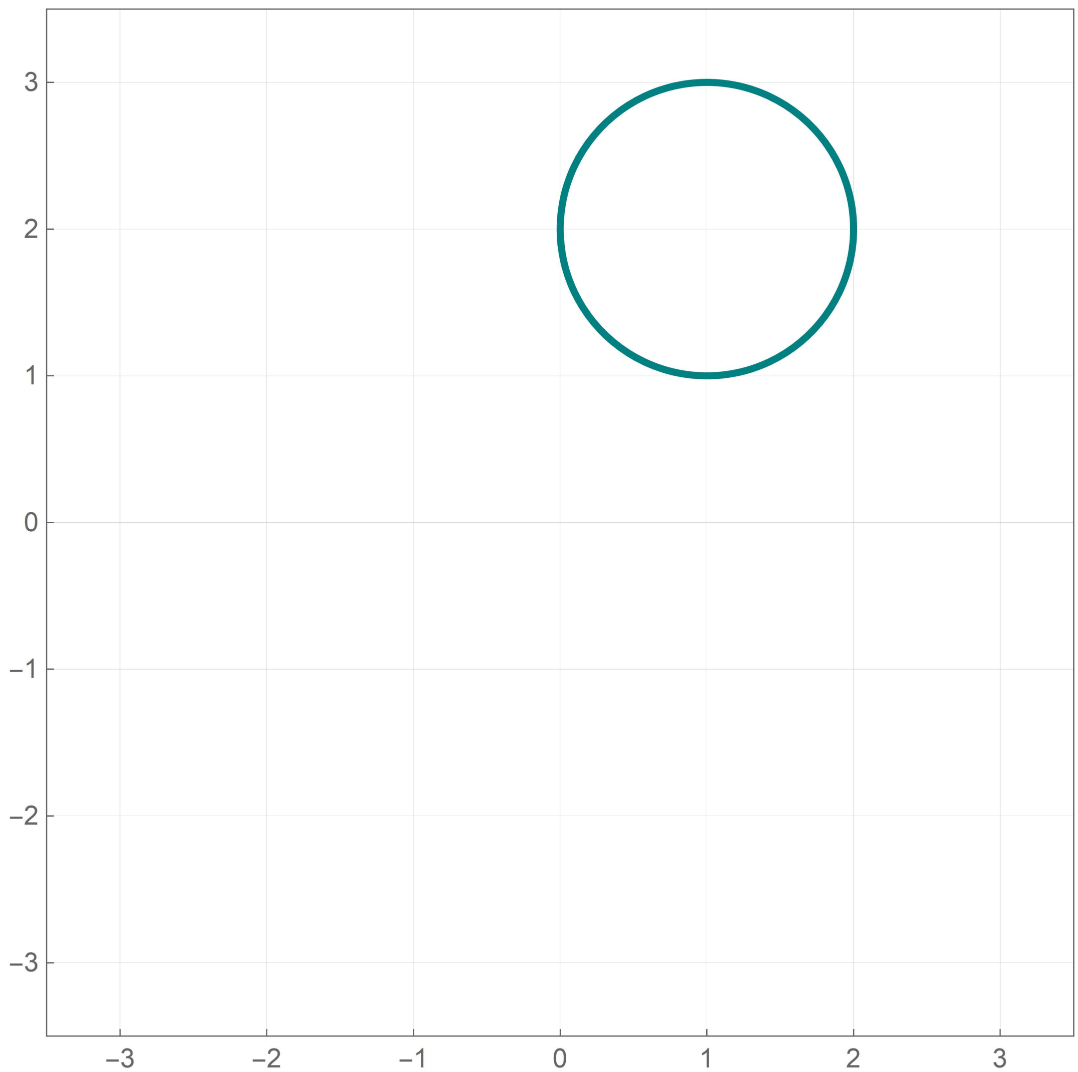
Out[19]=



It might be clearer with the GridLines:

In[20]:=
```
ParametricPlot[
 {1, 2} + {Cos[t], Sin[t]}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]}},
 PlotPoints → 101, PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True, GridLines → {Range[-3, 3, 1], Range[-3, 3, 1]},
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

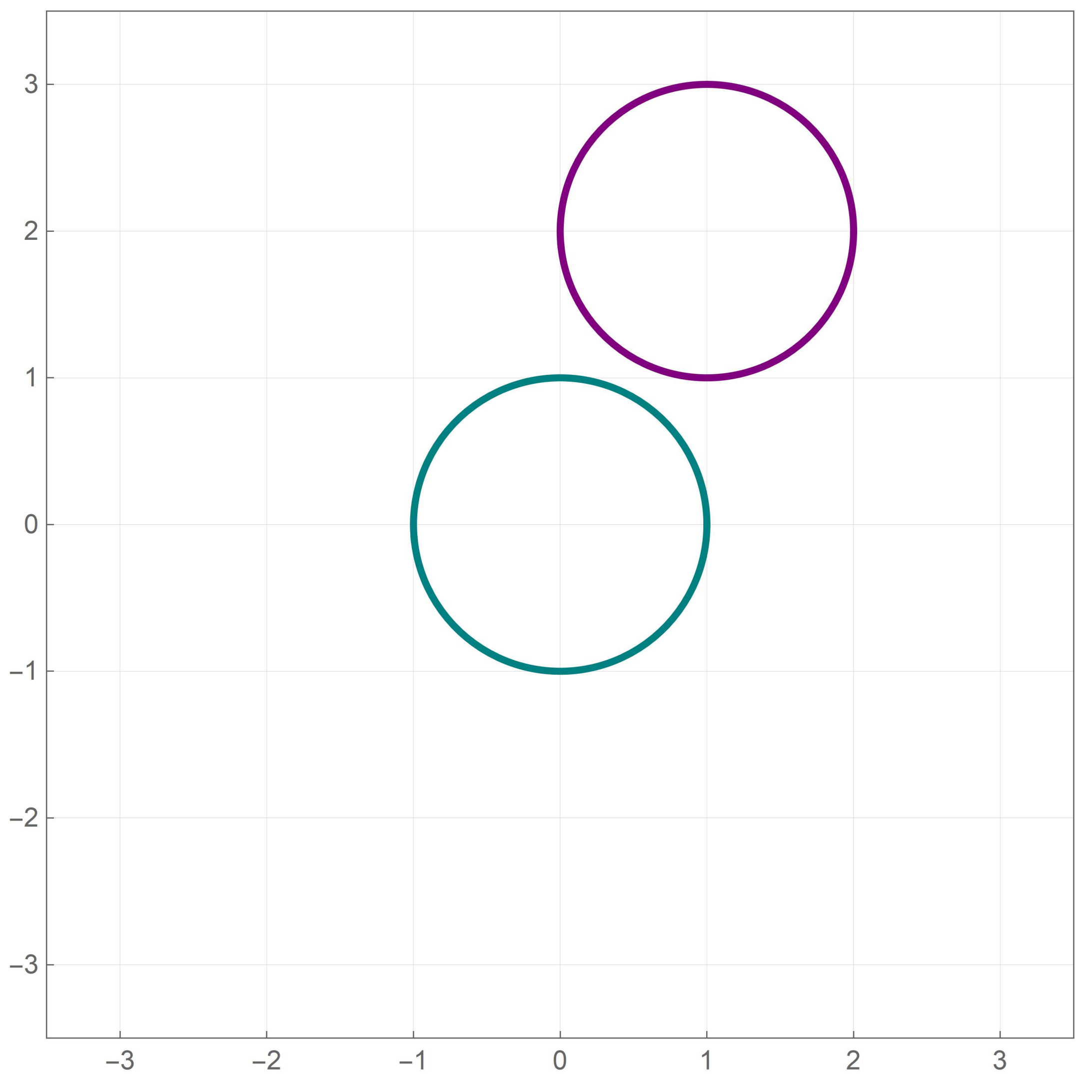
Out[20]=



Or, keeping the original circle:

In[21]:=
```
ParametricPlot[
 {{Cos[t], Sin[t]}, {1, 2} + {Cos[t], Sin[t]}}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]},
   {Thickness[0.007], RGBColor[0.5, 0, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True, GridLines → {Range[-3, 3, 1], Range[-3, 3]},
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

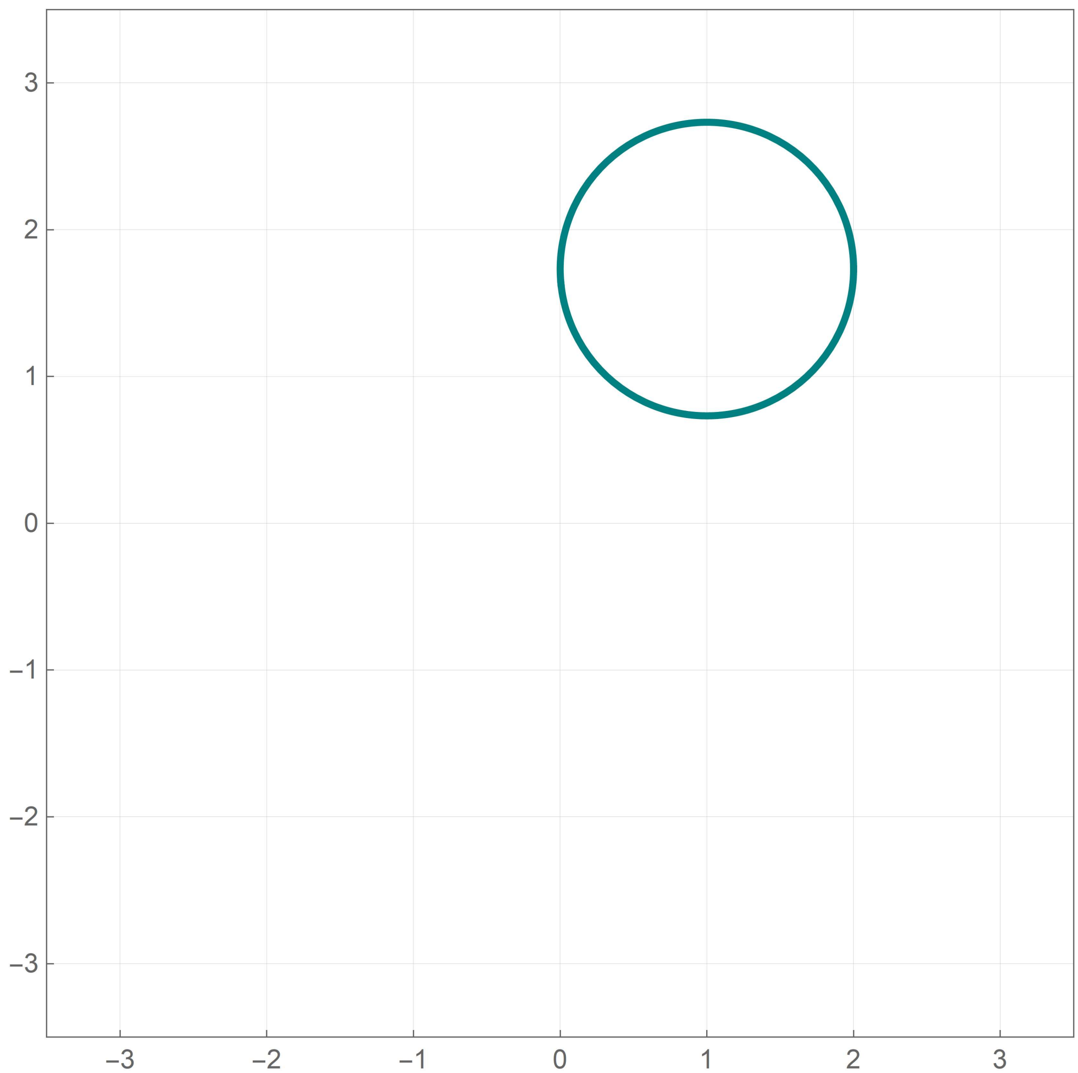
Out[21]=



Now I will move the unit circle in the direction of the vector 2 $\left\{\text{Cos}\left[\frac{\text{Pi}}{3}\right], \text{Sin}\left[\frac{\text{Pi}}{3}\right]\right\}$

In[22]:=
```
ParametricPlot[
 2 {Cos[Pi/3], Sin[Pi/3]} + {Cos[t], Sin[t]}, {t, 0, 2 Pi},
 PlotStyle → {{Thickness[0.007], RGBColor[0, 0.5, 0.5]}},
 PlotPoints → 101, PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
 Axes → False, Frame → True, GridLines → {Range[-3, 3, 1], Range[-3, 3, 1]},
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3, 1], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

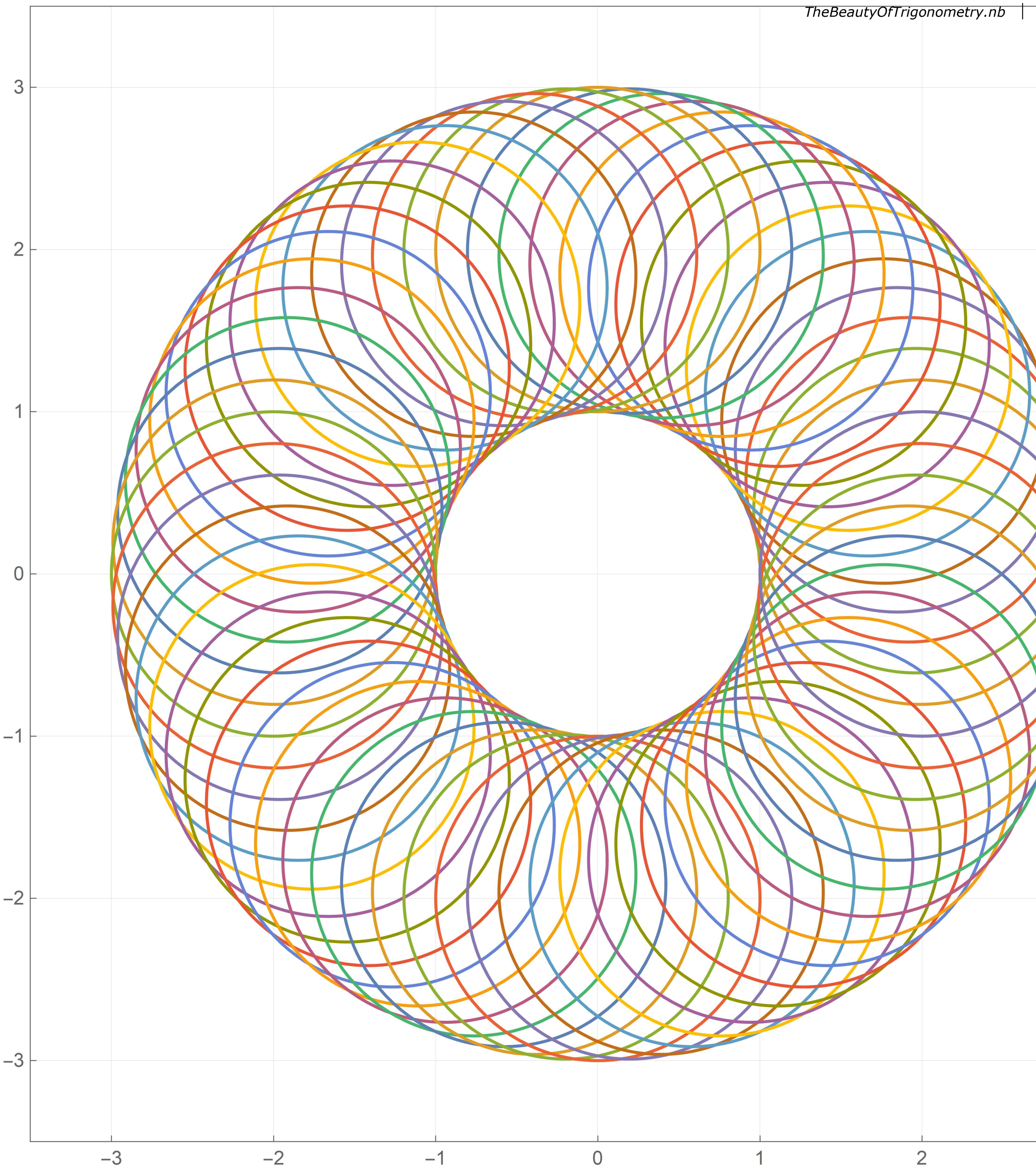
Out[22]=



Now I create many shifts in different directions:

```
In[23]:= ParametricPlot[
  Evaluate[Table[2 {Cos[an], Sin[an]} + {Cos[t], Sin[t]}, {an, 0, 2 Pi, Pi/32}]],
  {t, 0, 2 Pi}, PlotPoints → 101, PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
  Axes → False, Frame → True, GridLines → {Range[-3, 3, 1], Range[-3, 3, 1]},
  FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3, 1], {}}},
  AspectRatio → Automatic, ImageSize → 600
 ]
```

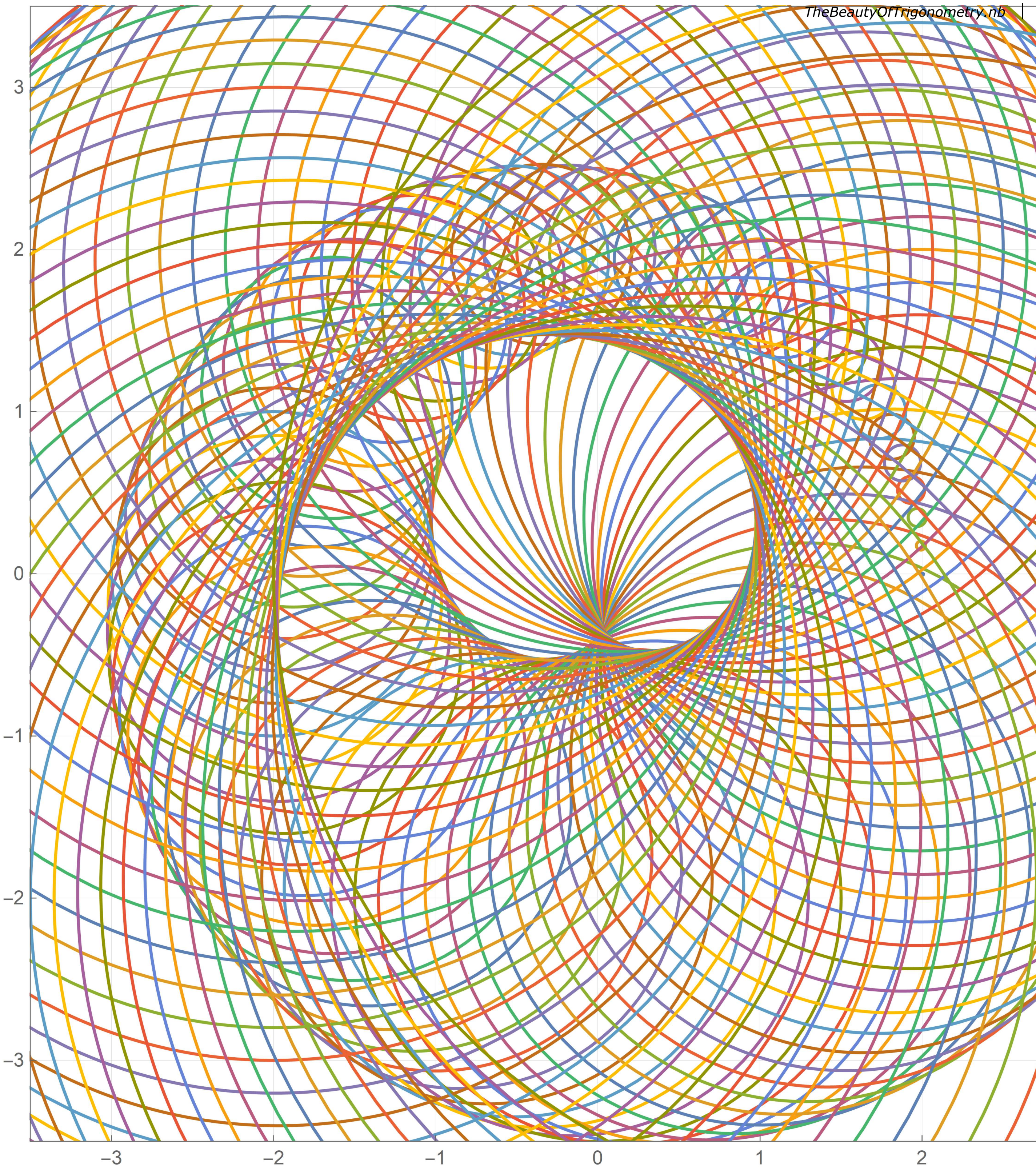
Out[23]=



Just for fun, as I shift a circle, I change it radius:

In[24]:=
```
ParametricPlot[
  Evaluate[Table[2 {Cos[an], Sin[an]} + an/Pi {Cos[t], Sin[t]},
    {an, 0, 4 Pi, Pi/36}]], {t, 0, 2 Pi}, PlotPoints → 101,
  PlotRange → {{-3.5, 3.5}, {-3.5, 3.5}},
  Axes → False, Frame → True, GridLines → {Range[-3, 3, 1], Range[-3, 3, 1]},
  FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
  AspectRatio → Automatic, ImageSize → 600
 ]
```

Out[24]=



Next, I want to color each point on the circle individually.

In[25]:=
```
Table[k, {k, 1, 20, 2}]
```

Out[25]=
```
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
```

In[26]:=
```
Table[{PointSize[0.02], Hue[t/(2 Pi)], Point[{Cos[t], Sin[t]}]},
  {t, 0, 2 Pi, Pi/16}]
```

Out[26]=
```
{{PointSize[0.02], , Point[{1, 0}]},
 {PointSize[0.02], , Point[{Cos[π/16], Sin[π/16]}]},
 {PointSize[0.02], , Point[{Cos[π/8], Sin[π/8]}]},
 {PointSize[0.02], , Point[{Cos[3 π/16], Sin[3 π/16]}]},
 {PointSize[0.02], , Point[{1/Sqrt[2], 1/Sqrt[2]}]},
 {PointSize[0.02], , Point[{Sin[3 π/16], Cos[3 π/16]}]},
 {PointSize[0.02], , Point[{Sin[π/8], Cos[π/8]}]},
 {PointSize[0.02], , Point[{Sin[π/16], Cos[π/16]}]},
 {PointSize[0.02], , Point[{0, 1}]},
 {PointSize[0.02], , Point[{-Sin[π/16], Cos[π/16]}]},
 {PointSize[0.02], , Point[{-Sin[π/8], Cos[π/8]}]},
 {PointSize[0.02], , Point[{-Sin[3 π/16], Cos[3 π/16]}]},
 {PointSize[0.02], , Point[{-1/Sqrt[2], 1/Sqrt[2]}]},
 {PointSize[0.02], , Point[{-Cos[3 π/16], Sin[3 π/16]}]},
 {PointSize[0.02], , Point[{-Cos[π/8], Sin[π/8]}]},
 {PointSize[0.02], , Point[{-Cos[π/16], Sin[π/16]}]},
 {PointSize[0.02], , Point[{-1, 0}]},
 {PointSize[0.02], , Point[{-Cos[π/16], -Sin[π/16]}]},
 {PointSize[0.02], , Point[{-Cos[π/8], -Sin[π/8]}]},
 {PointSize[0.02], , Point[{-Cos[3 π/16], -Sin[3 π/16]}]},
```

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{-\text{Sin}[\frac{3\pi}{16}], -\text{Cos}[\frac{3\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{-\text{Sin}[\frac{\pi}{8}], -\text{Cos}[\frac{\pi}{8}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{-\text{Sin}[\frac{\pi}{16}], -\text{Cos}[\frac{\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{0, -1\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Sin}[\frac{\pi}{16}], -\text{Cos}[\frac{\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Sin}[\frac{\pi}{8}], -\text{Cos}[\frac{\pi}{8}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Sin}[\frac{3\pi}{16}], -\text{Cos}[\frac{3\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Cos}[\frac{3\pi}{16}], -\text{Sin}[\frac{3\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Cos}[\frac{\pi}{8}], -\text{Sin}[\frac{\pi}{8}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{\text{Cos}[\frac{\pi}{16}], -\text{Sin}[\frac{\pi}{16}]\}]\},$

$\{\text{PointSize}[0.02], \blacksquare, \text{Point}[\{1, 0\}]\}\}$

Let us explore how the color function Hue[] works using table:

In[27]:=
```
Table[{t, Hue[t]}, {t, 0, 1, 0.1}]
```

Out[27]=
```
{{0., ■}, {0.1, ■}, {0.2, ■}, {0.3, ■}, {0.4, ■},
 {0.5, ■}, {0.6, ■}, {0.7, ■}, {0.8, ■}, {0.9, ■}, {1., ■}}
```

I am not sure that I know the names for all these colors, but, it seems that Hue[0] is red, then proceeds towards orange, then lime, and so on.

One can ask Mathematica for the RGBColor[] code from the Hue[]:

In[28]:=
```
InputForm[ColorConvert[Hue[0], "RGB"]]
```

Out[28]//InputForm=
```
RGBColor[1., 0., 0.]
```

In[29]:=
```
FullForm[ColorConvert[Hue[0], "RGB"]]
```

Out[29]//FullForm=
```
RGBColor[1.`, 0.`, 0.`]
```

For color-curious this might be interesting:

In[30]:=
```
Table[{t, InputForm[ColorConvert[Hue[t], "RGB"]]}, {t, 0, 1, 0.1}]
```

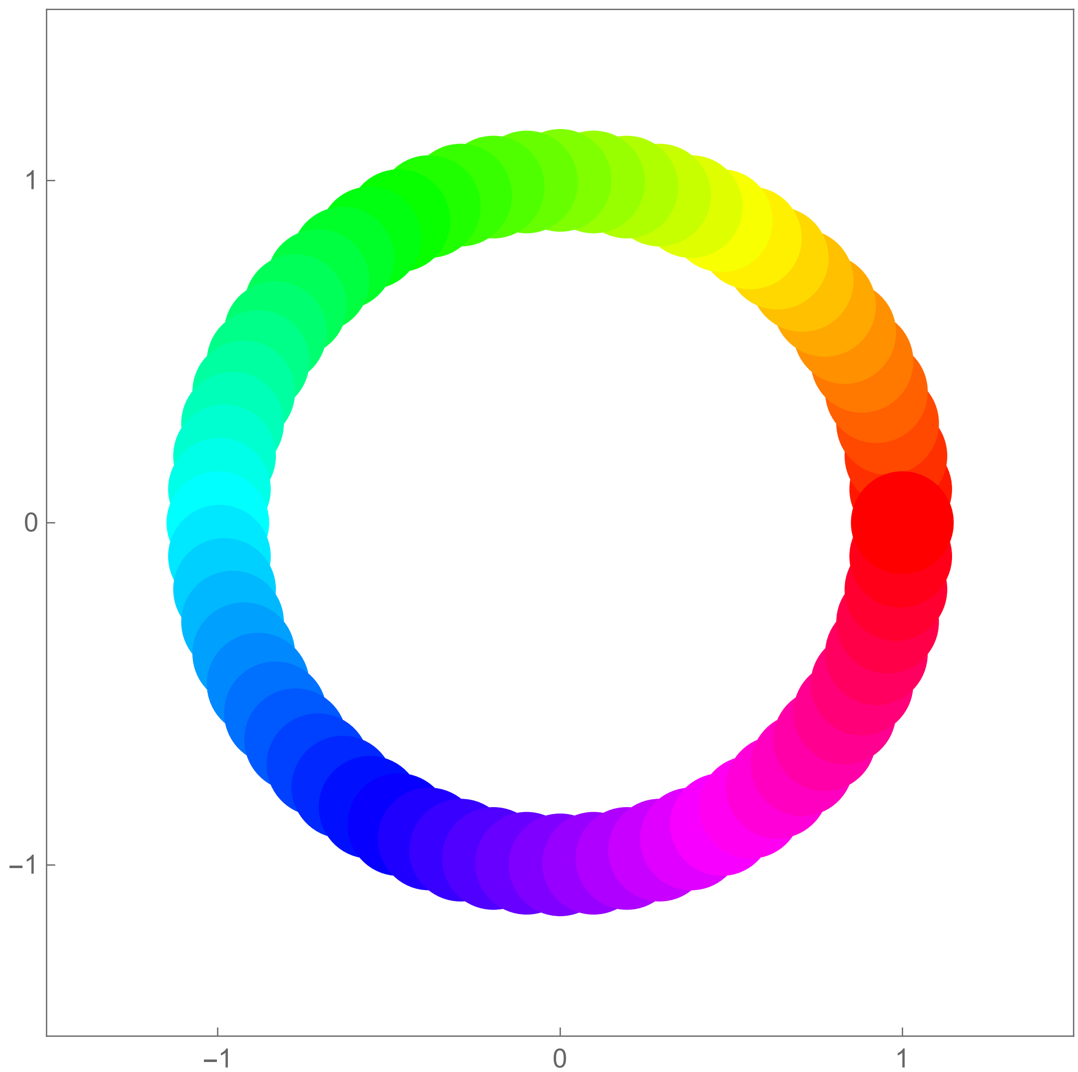
Out[30]=
```
{{0., RGBColor[1., 0., 0.]}, {0.1, RGBColor[1., 0.6000000000000001, 0.]},
 {0.2, RGBColor[0.7999999999999998, 1., 0.]},
 {0.3, RGBColor[0.19999999999999973, 1., 0.]},
 {0.4, RGBColor[0., 1., 0.40000000000000036]},
 {0.5, RGBColor[0., 1., 1.]}, {0.6, RGBColor[0., 0.39999999999999947, 1.]},
 {0.7, RGBColor[0.20000000000000018, 0., 1.]},
 {0.8, RGBColor[0.8000000000000007, 0., 1.]},
 {0.9, RGBColor[1., 0., 0.5999999999999996]}, {1., RGBColor[1., 0., 0.]}}
```

We continue exploring the unit circle, point by point. Below is thirty three quite large points on the unit circle:

reproduce the picture below (5)

In[31]:=
```
Graphics[{
  Table[{PointSize[0.1], Hue[t/(2 Pi)], Point[{Cos[t], Sin[t]}]},
   {t, 0, 2 Pi, Pi/32}]
 },
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

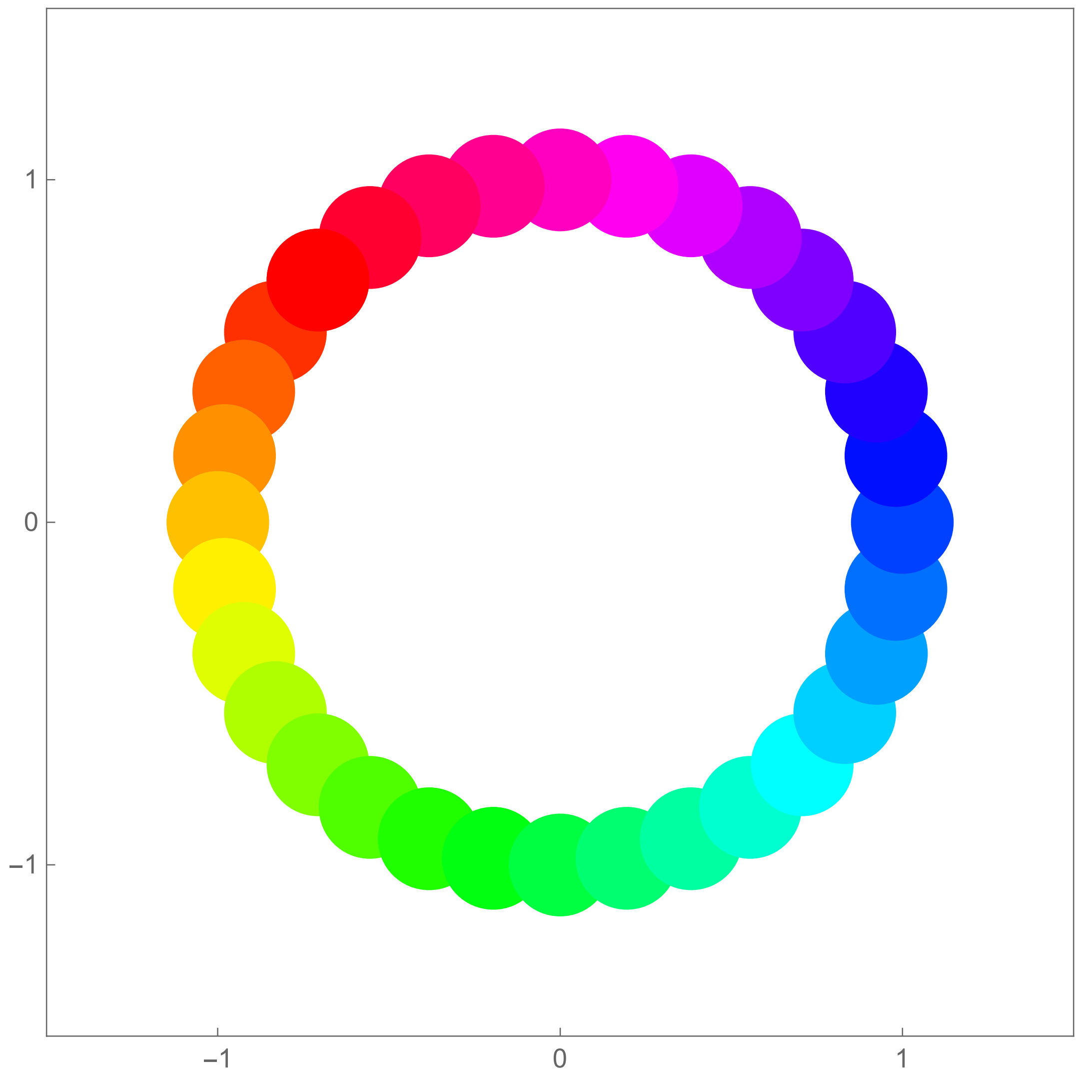
Out[31]=



One way to move the colors around would be to introduce a new variable which I call aa below. Change the value for aa to see what happens.

In[32]:=
```
Clear[aa]; aa = 3 Pi/4; Graphics[{
  Table[{PointSize[0.1], Hue[t/(2 Pi)], Point[{Cos[aa + t], Sin[aa + t]}]},
   {t, 0, 2 Pi, Pi/16}]
 },
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

Out[32]=

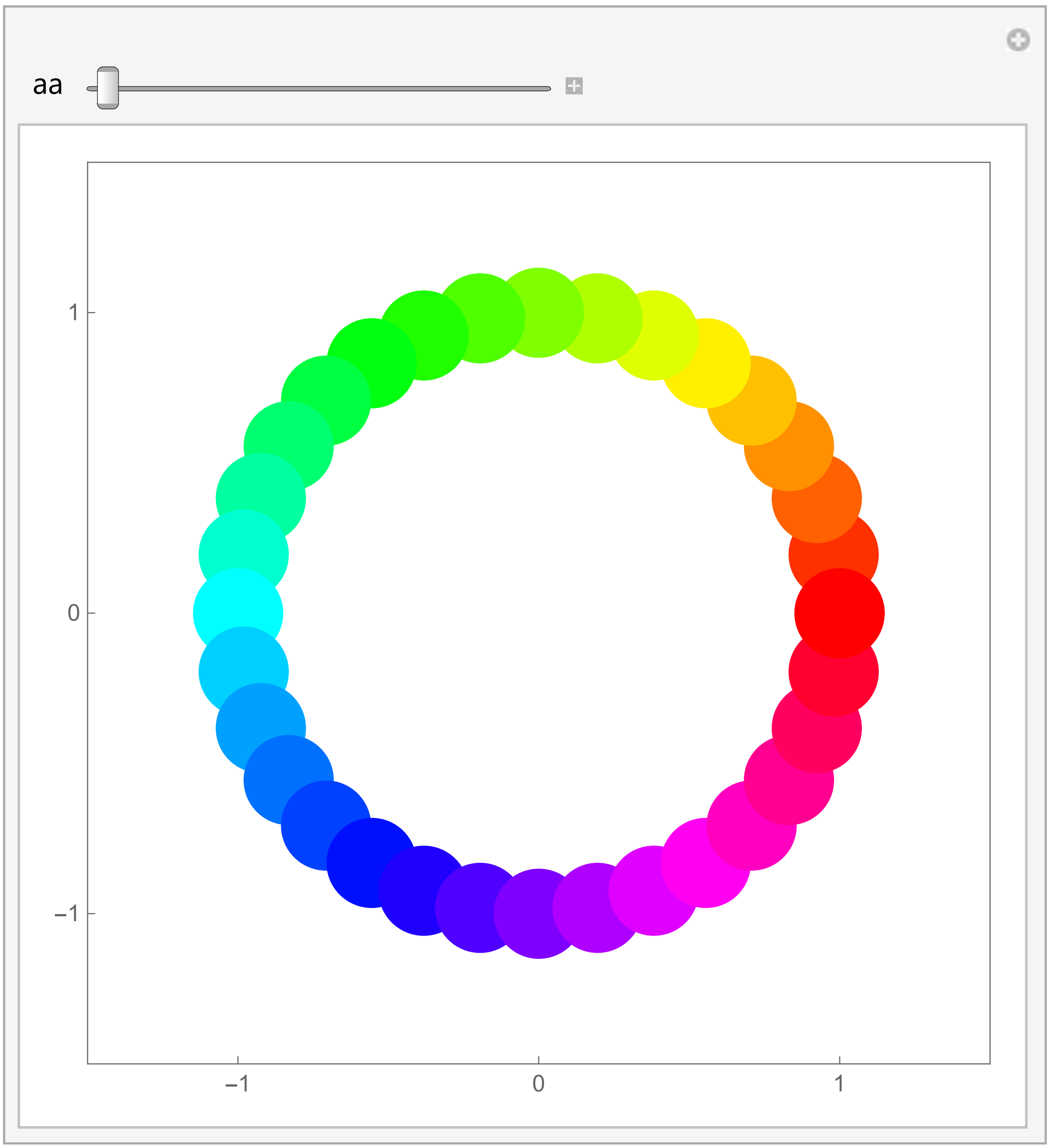


One can explore the effect of changing colors by using the command Manipulate[]

reproduce the manipulation below (6)

In[33]:=
```
Clear[aa]; Manipulate[
 Graphics[{
   Table[{PointSize[0.1], Hue[t/(2 Pi)], Point[{Cos[aa + t], Sin[aa + t]}]},
    {t, 0, 2 Pi, Pi/16}]
   },
  PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
  Axes → False, Frame → True,
  FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
  AspectRatio → Automatic, ImageSize → 400
 ], {aa, 0, 2 Pi, ControlPlacement → Top}]
```

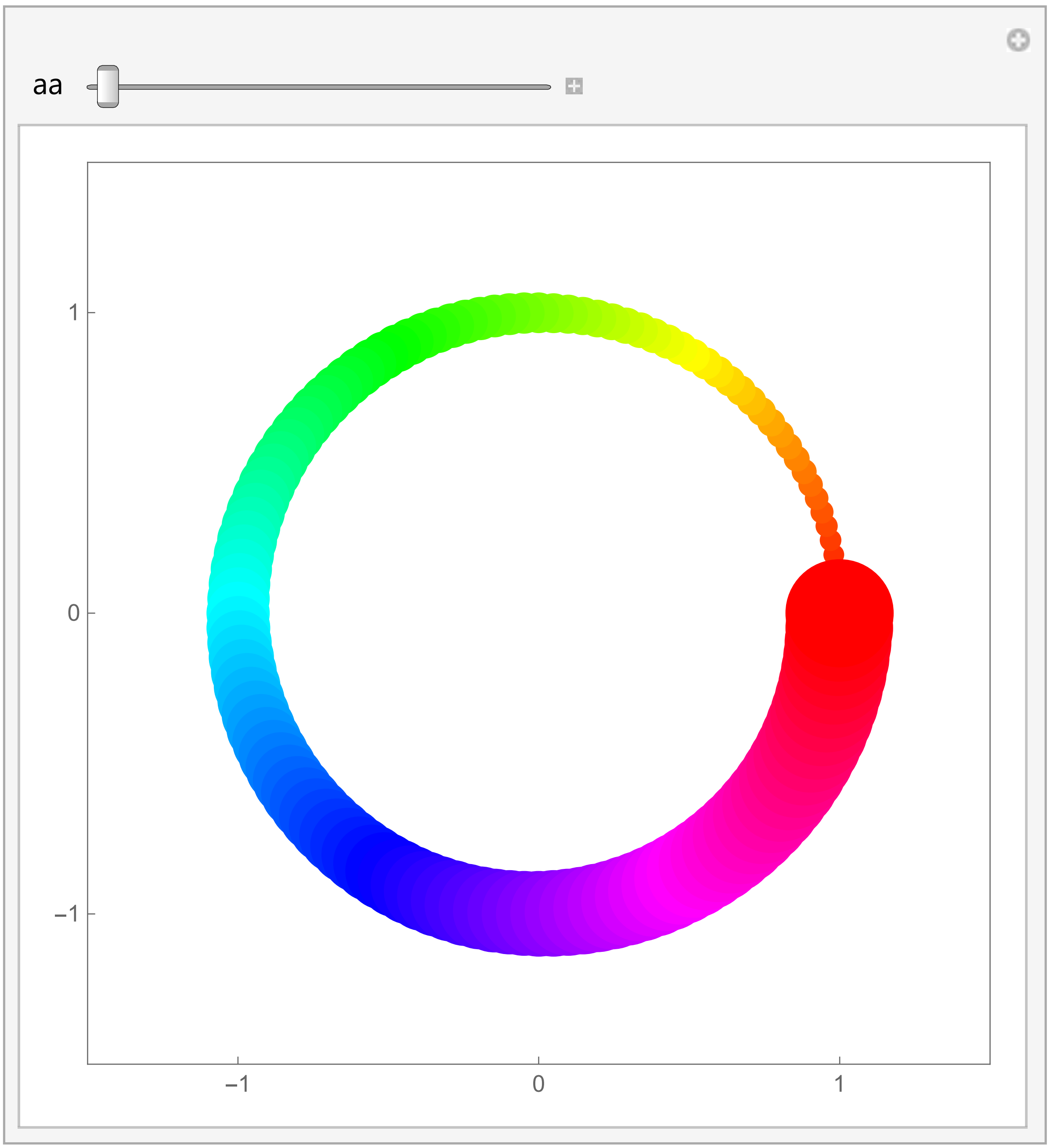
Out[33]=



reproduce the manipulation below (7)

In[34]:=
```
Clear[aa]; Manipulate[Graphics[{
    Table[{PointSize[0.02 + 0.1/(2 Pi) t], Hue[t/(2 Pi)],
      Point[{Cos[aa + t], Sin[aa + t]}]}, {t, 0, 2 Pi, Pi/64}]
   },
   PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
   Axes → False, Frame → True,
   FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
   AspectRatio → Automatic, ImageSize → 400
  ], {aa, 0, 2 Pi, ControlPlacement → Top}]
```

Out[34]=

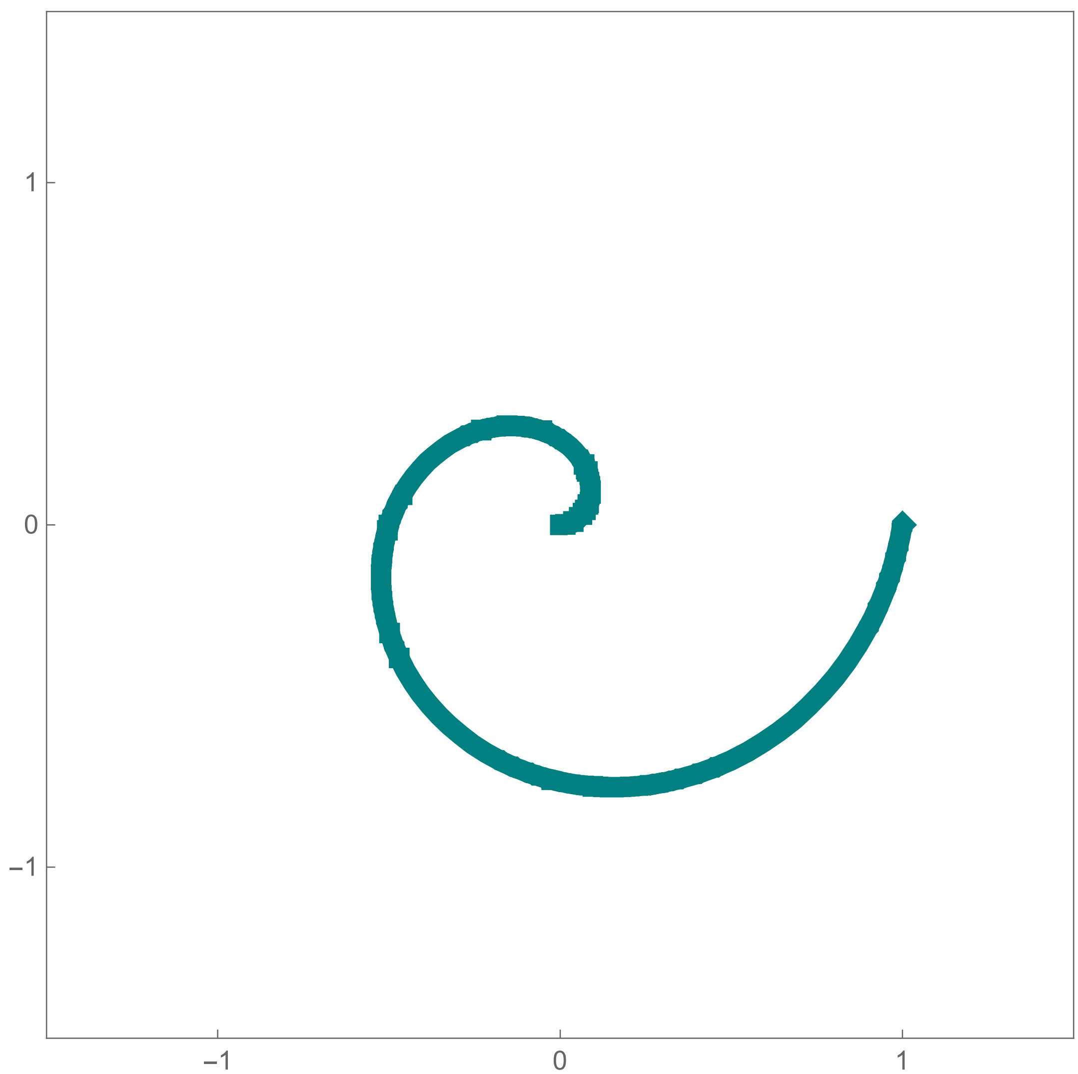


Let us explore spirals. The first spiral starts with the radius 0, then it increases to 1 as t changes from 0 to 2 Pi.

reproduce the picture below (8)

In[35]:=
```
ParametricPlot[
  t/(2 Pi) {Cos[t], Sin[t]}, {t, 0, 2*Pi},
  PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
  PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
  Axes → False, Frame → True,
  FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
  AspectRatio → Automatic, ImageSize → 400
]
```

Out[35]=

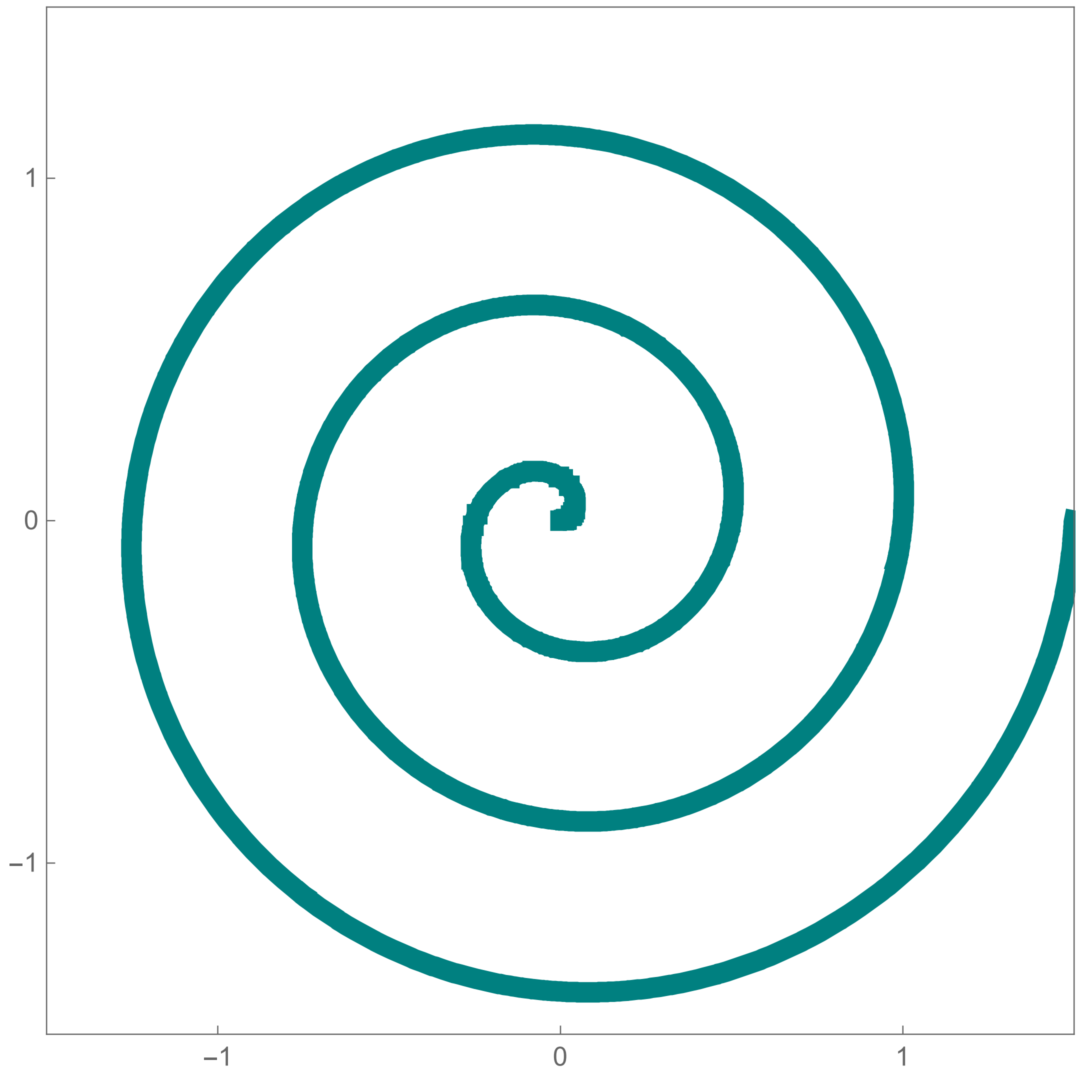


If we want a spiral to make more turns, we need to play with the changing radius. The spiral below starts with the radius 0, then the radius increases to $\frac{3}{2}$ as t changes from 0 to 6 Pi.

reproduce the picture below (9)

In[36]:=
```
ParametricPlot[
 t/(4 Pi) {Cos[t], Sin[t]}, {t, 0, 6*Pi},
 PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

Out[36]=

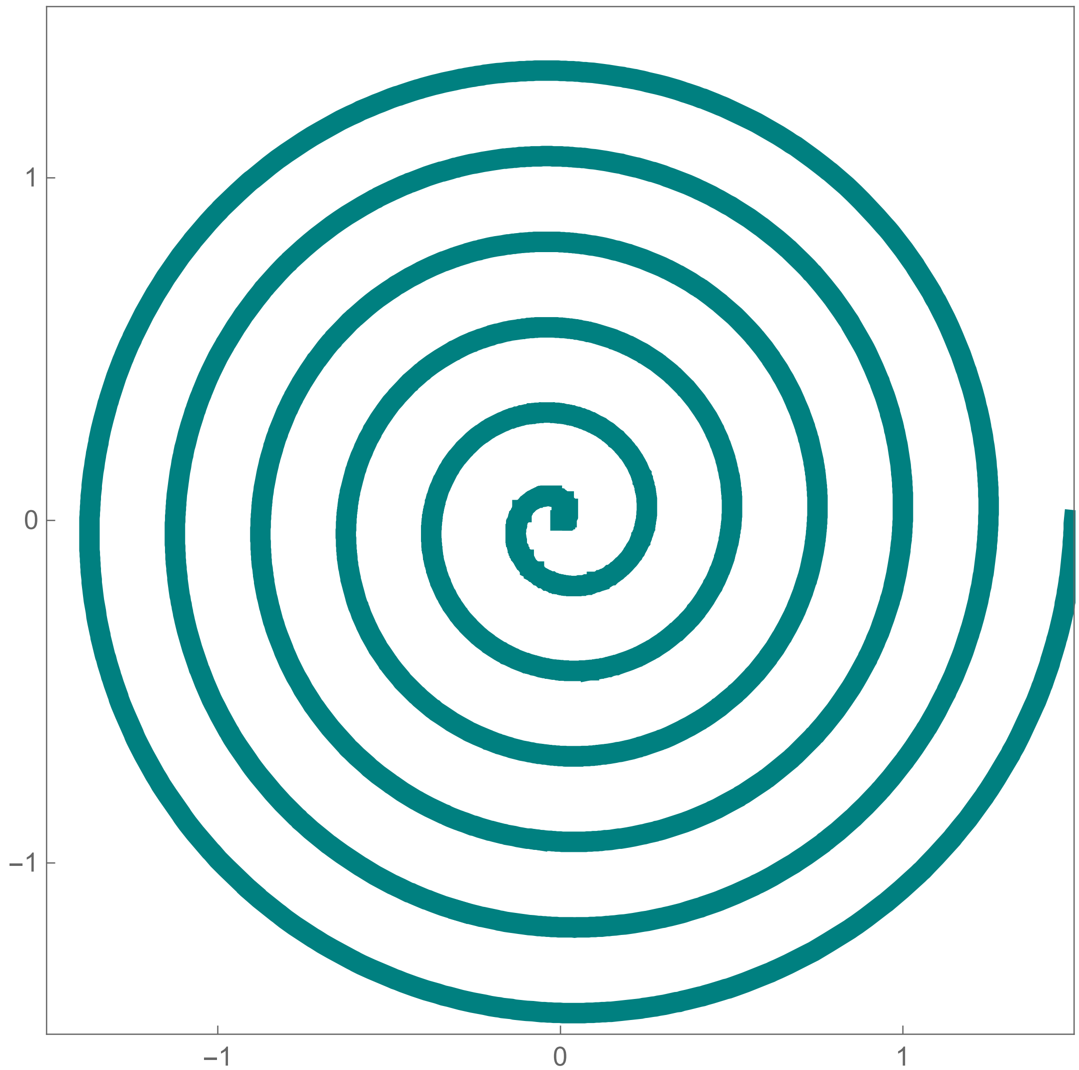


If we want a spiral to make even more turns, we need to let t change from 0 to say 12 Pi, but at the same time we need to divide the radius by 8 Pi, so to make the radius at most $\frac{3}{2}$ as t changes from 0 to 12 Pi.

reproduce the picture below (10)

In[37]:=
```
ParametricPlot[
 t/(8 Pi) {Cos[t], Sin[t]}, {t, 0, 12*Pi},
 PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
 Axes → False, Frame → True,
 FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3], {}}},
 AspectRatio → Automatic, ImageSize → 400
]
```

Out[37]=

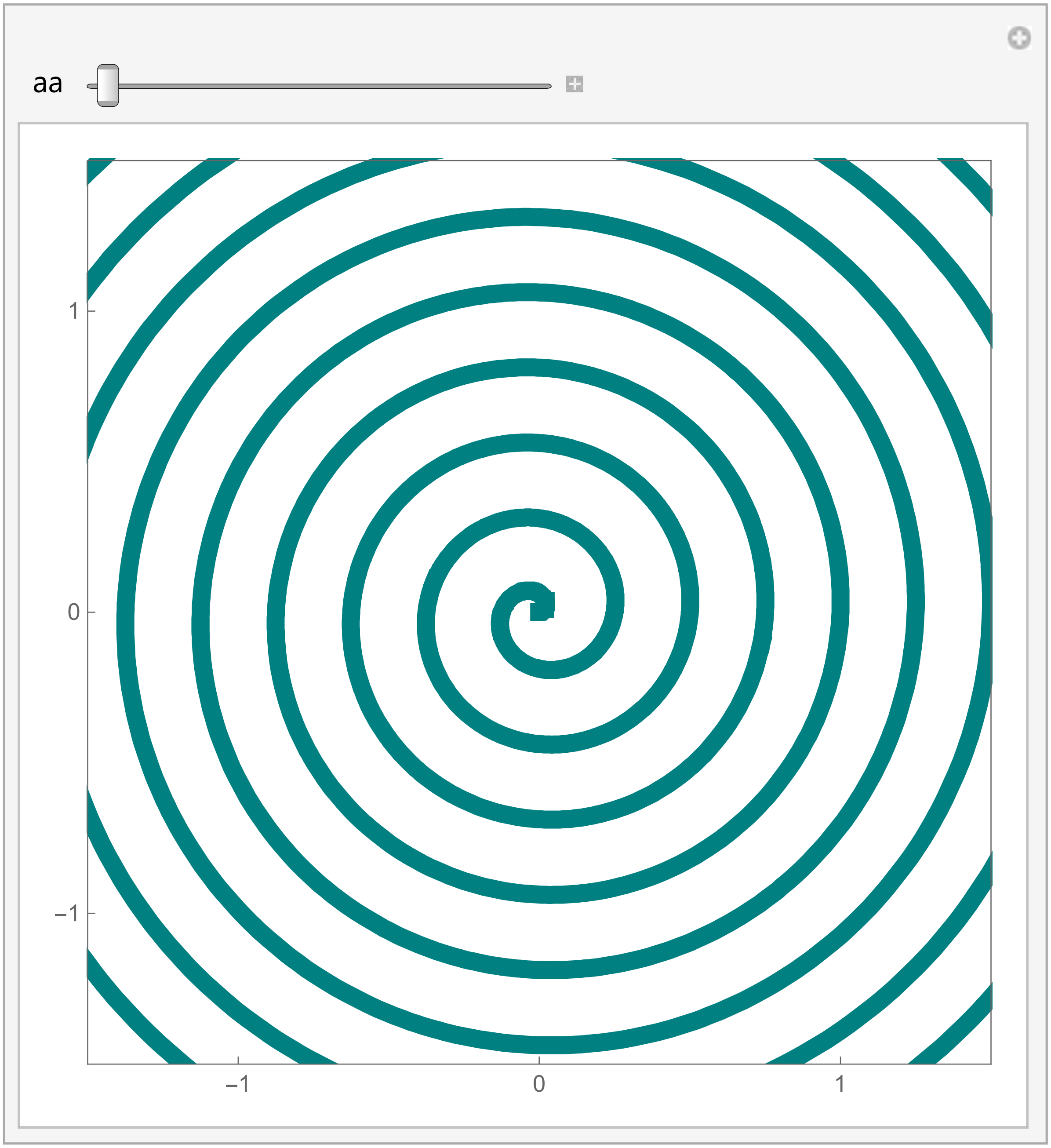


The manipulation below might be interesting.

reproduce the manipulation below (11)

In[38]:=
```
Manipulate[
 ParametricPlot[
  t/(8 Pi) {Cos[aa + t], Sin[aa + t]}, {t, 0, 29 * Pi},
  PlotStyle → {{Thickness[0.02], RGBColor[0, 0.5, 0.5]}},
  PlotPoints → 101, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
  Axes → False, Frame → True,
  FrameTicks → {{Range[-3, 3, 1], {}}, {Range[-3, 3, 1], {}}},
  AspectRatio → Automatic, ImageSize → 400
 ], {aa, 0, 2 Pi}]
```

Out[38]=

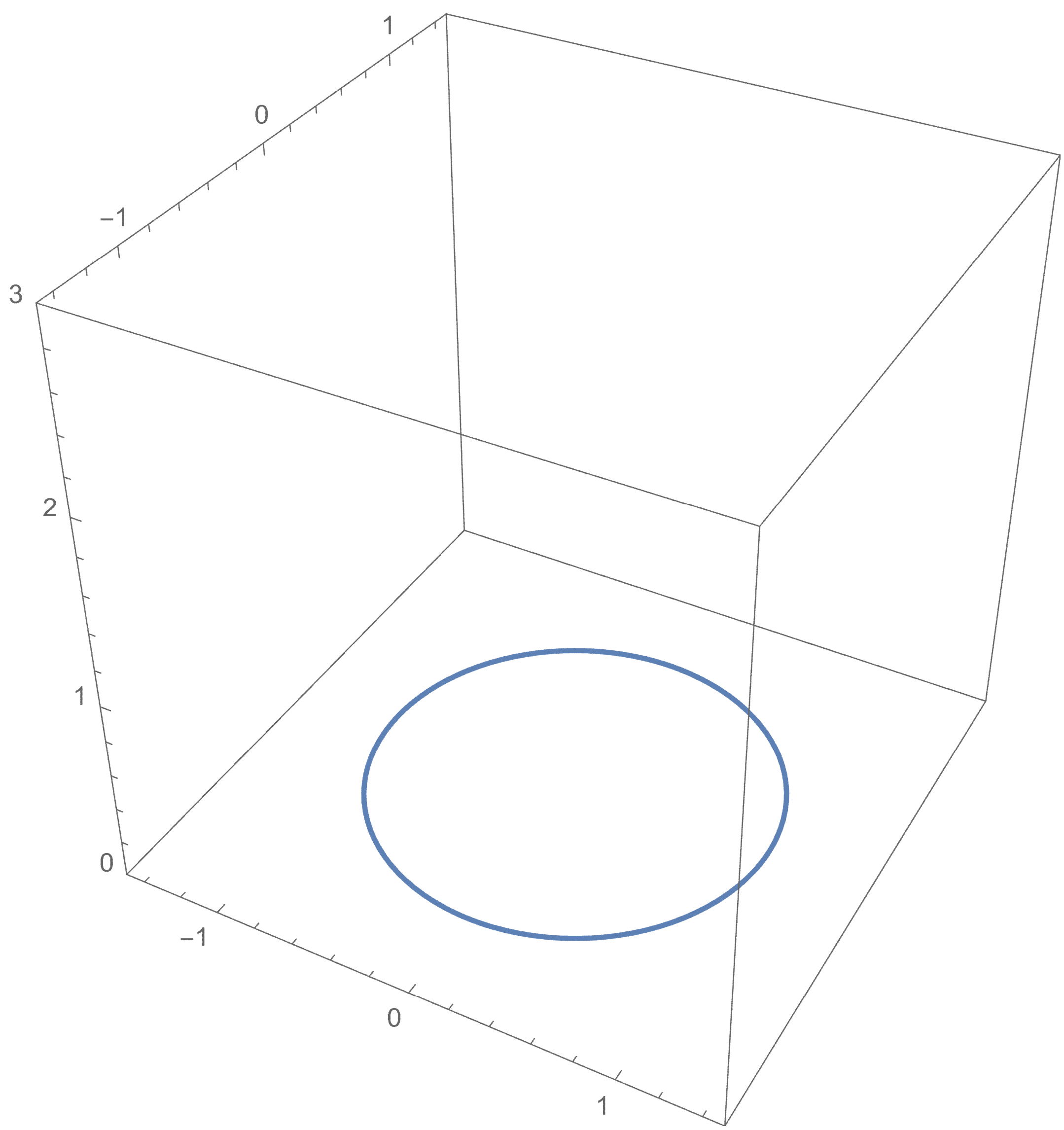


Cylinder

To show a 3-dimensional plot we use ParametricPlot3D[]. Now for the unit circle we need three coordinates, x, y, and z. To draw the unit circle in xy-plane we set z = 0.

In[39]:=
```
ParametricPlot3D[
 {Cos[t], Sin[t], 0}, {t, 0, 2 * Pi}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

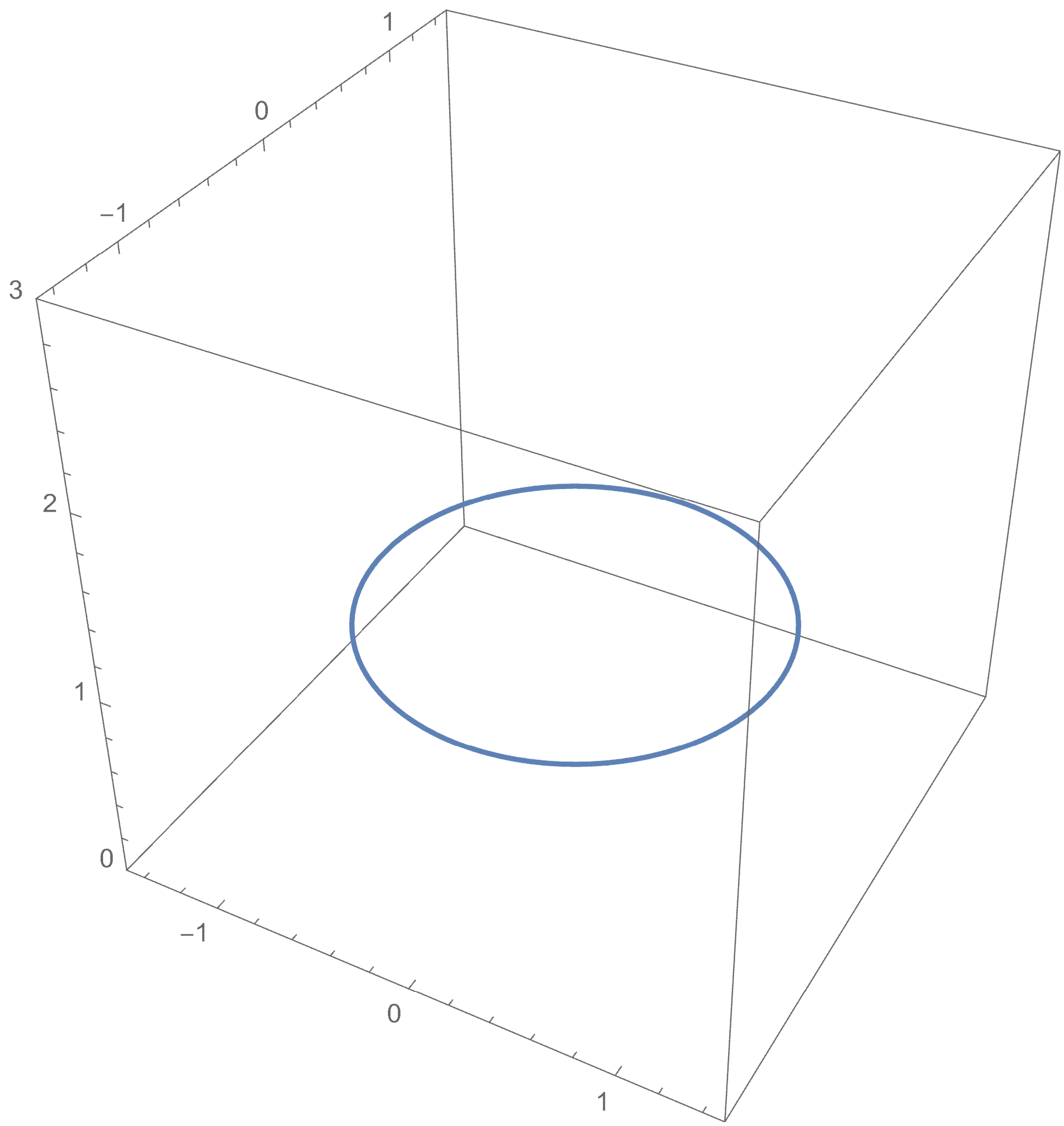
Out[39]=



But we can lift the circle to any height.

In[40]:=
```
Clear[zz]; zz = 1; ParametricPlot3D[
 {Cos[t], Sin[t], zz}, {t, 0, 2*Pi}, PlotPoints → 101,
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

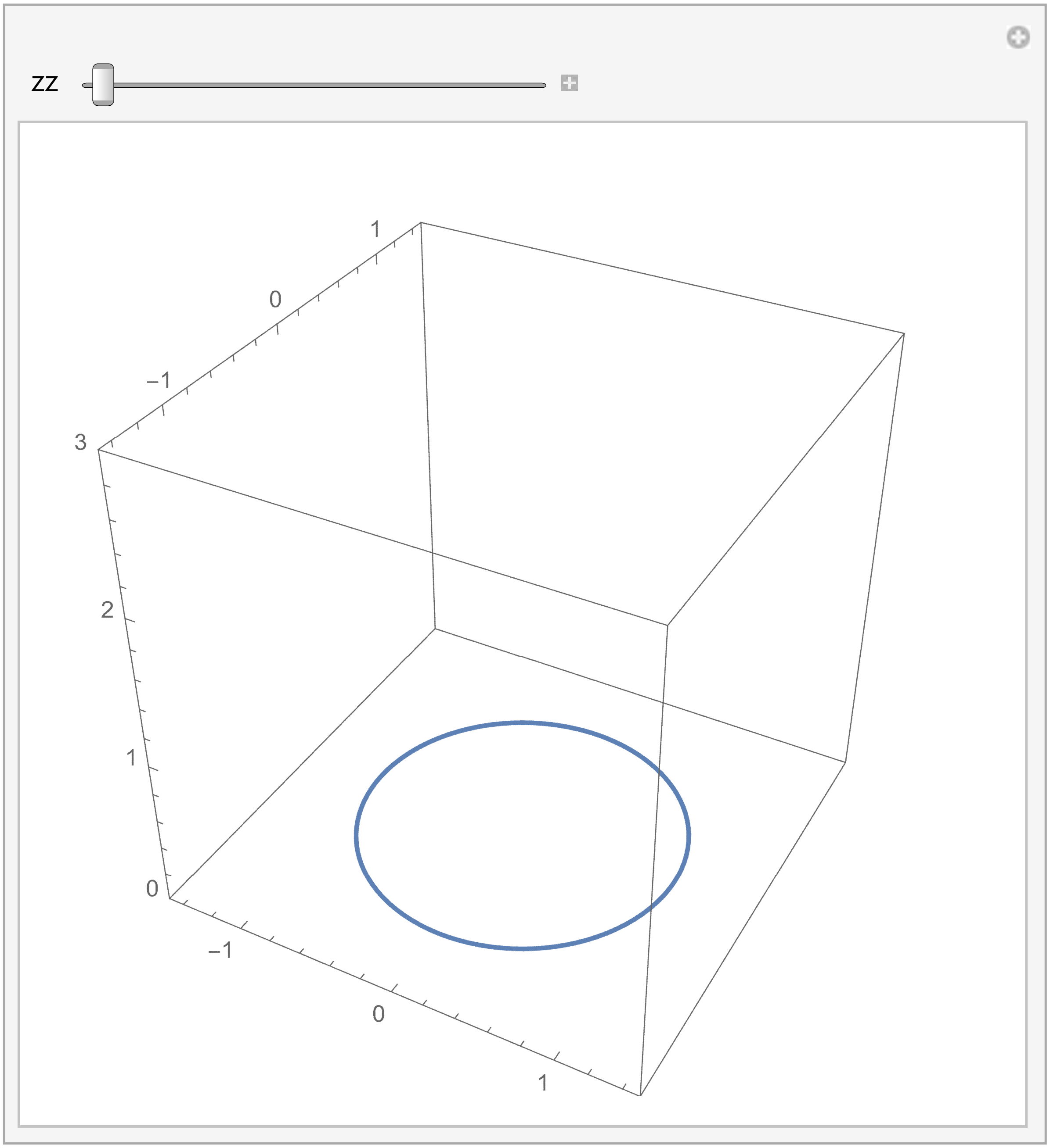
Out[40]=



Or, use Manipulate[] to further explore the lift.

In[41]:=
```
Clear[zz]; Manipulate[ParametricPlot3D[
  {Cos[t], Sin[t], zz}, {t, 0, 2*Pi}, PlotPoints → 101,
  PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ], {zz, 0, 3, ControlPlacement → Top}]
```

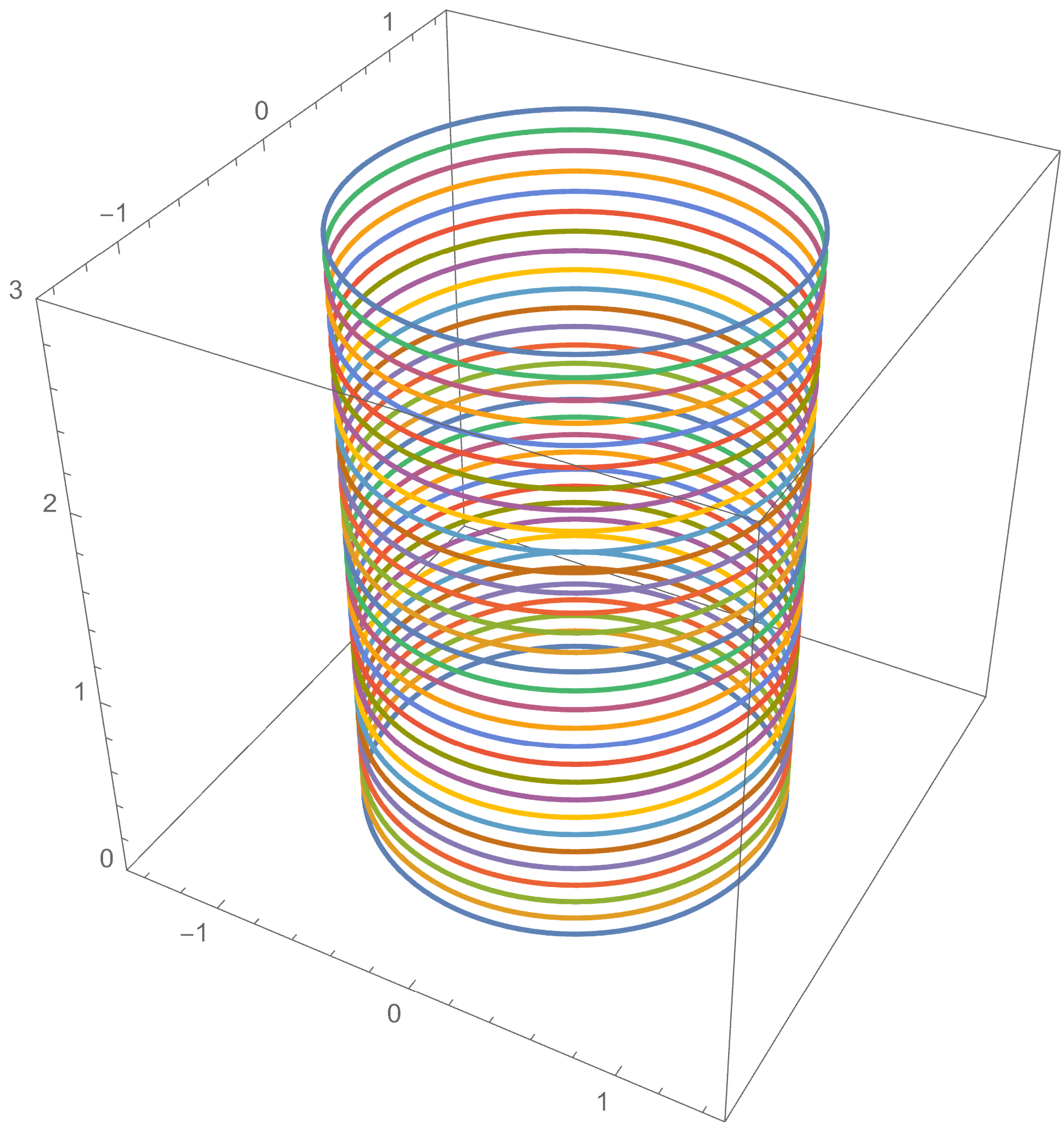
Out[41]=



Or, we can draw many circles in one picture:

In[42]:=
```
Clear[zz]; ParametricPlot3D[
 Evaluate[Table[{Cos[t], Sin[t], zz}, {zz, 0, 3, 0.1}]], {t, 0, 2*Pi},
 PlotPoints → 101, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

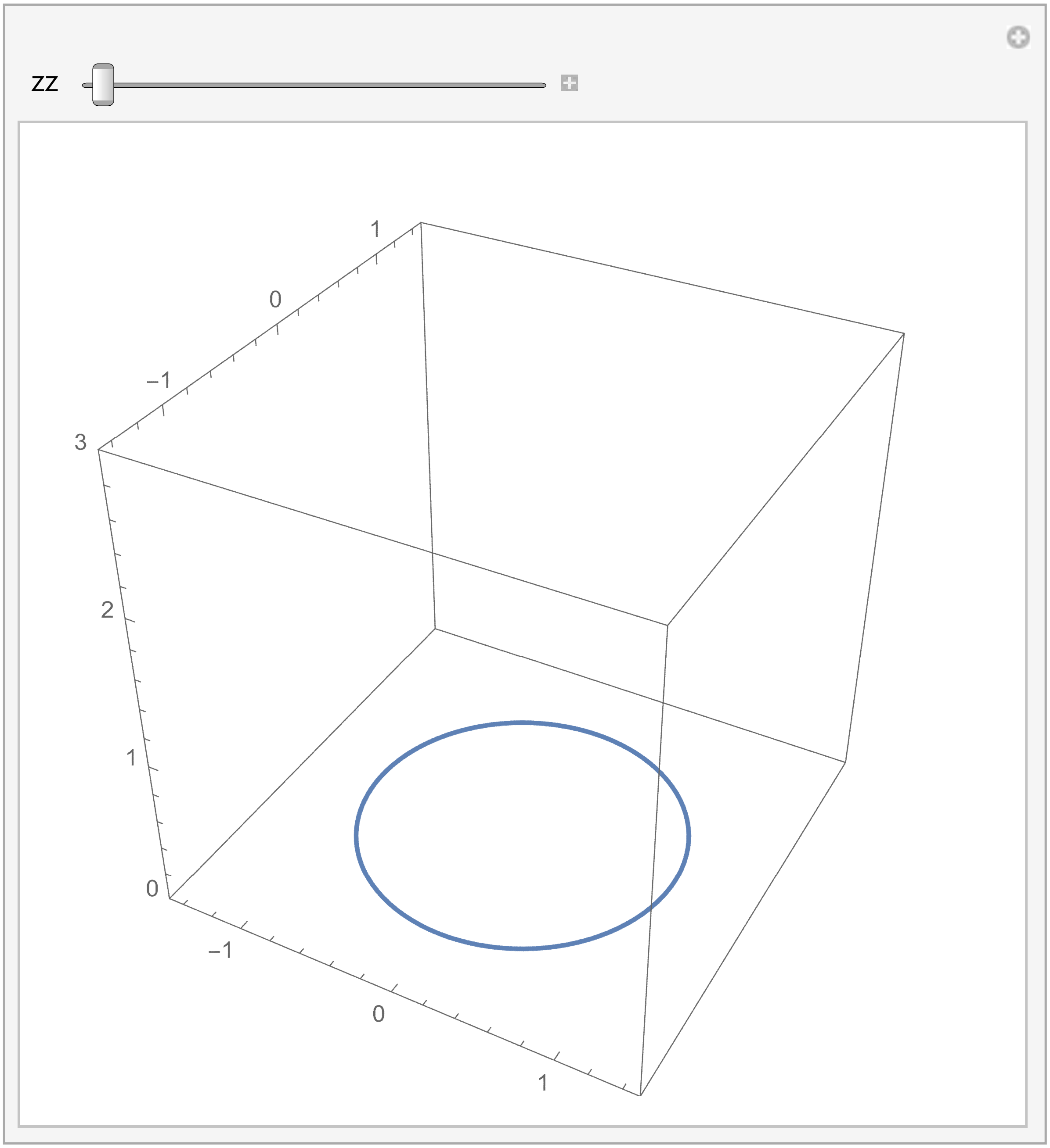
Out[42]=



Or, we can combine many circles with Manipulate:

In[43]:=
```
Clear[zz]; Manipulate[ParametricPlot3D[
  Table[{Cos[t], Sin[t], z}, {z, 0, zz, 0.1}], {t, 0, 2*Pi},
  PlotPoints → 101, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ], {zz, 0, 3, ControlPlacement → Top}]
```

Out[43]=

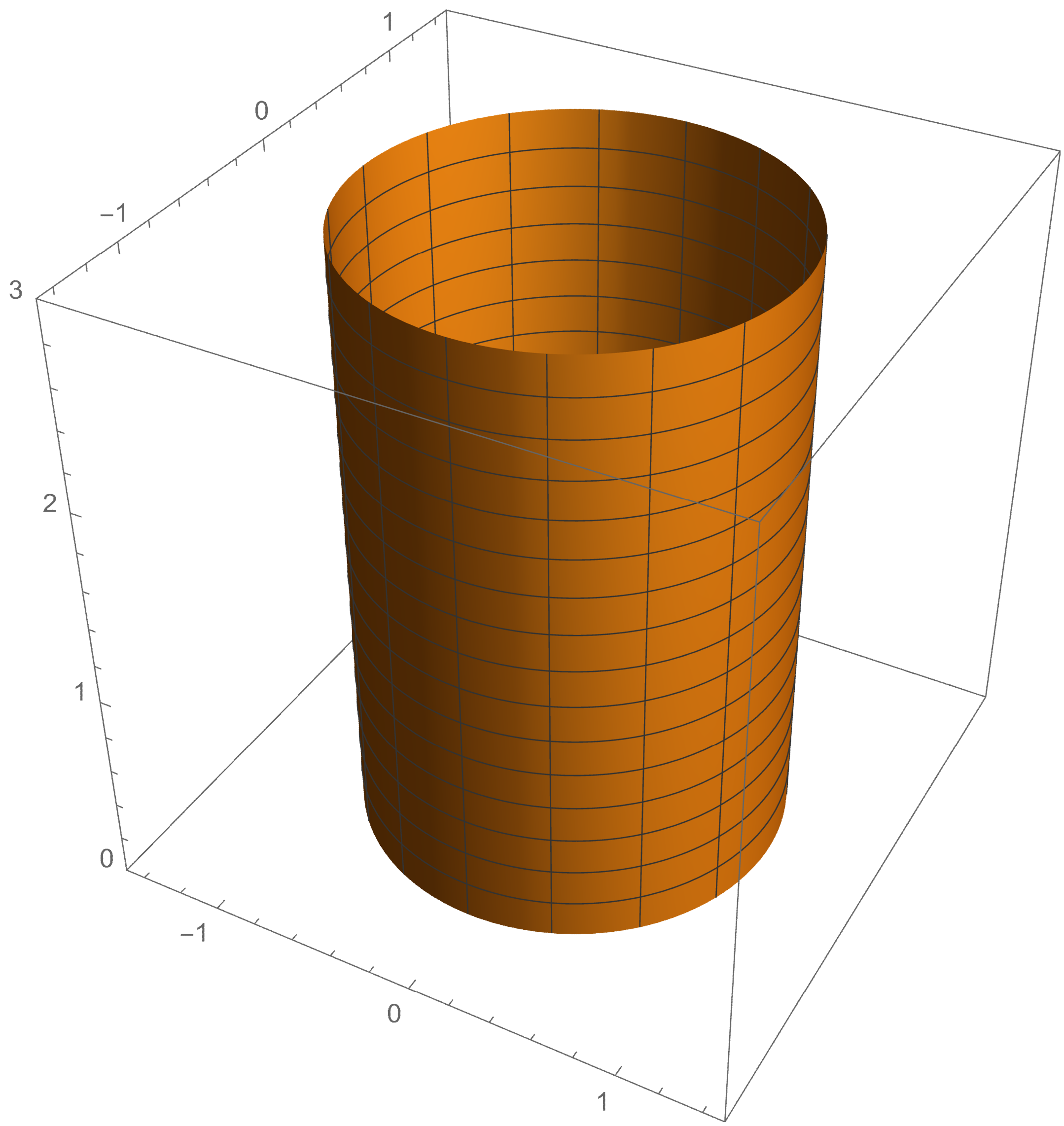


So, many circles build a cylinder:

reproduce the picture below (12)

In[44]:=
```
ParametricPlot3D[
 {Cos[t], Sin[t], z}, {z, 0, 3}, {t, 0, 2*Pi}, PlotPoints → {101, 101},
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 3}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

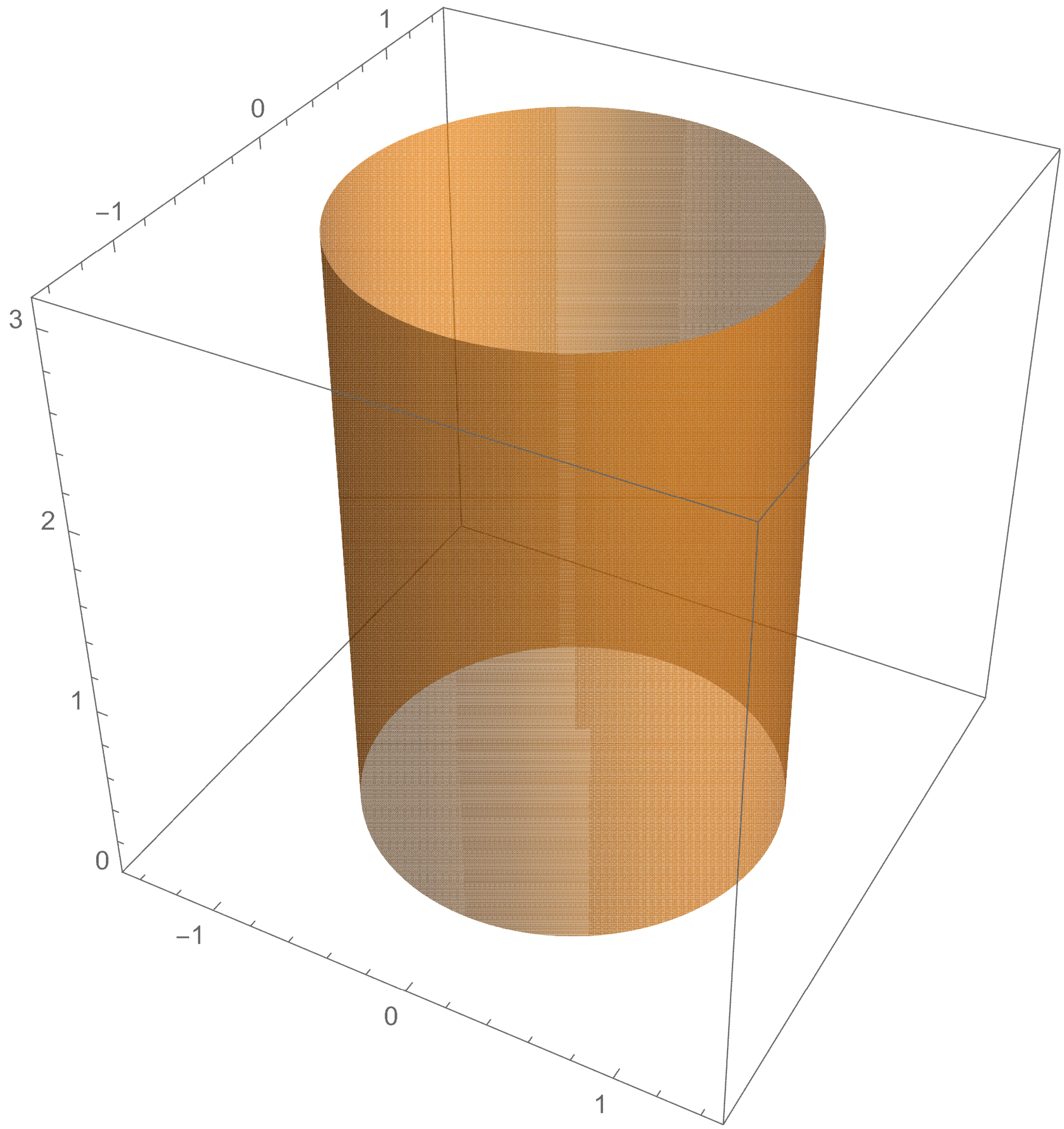
Out[44]=



Some variations with PlotStyle and Mesh: I named it here cyl

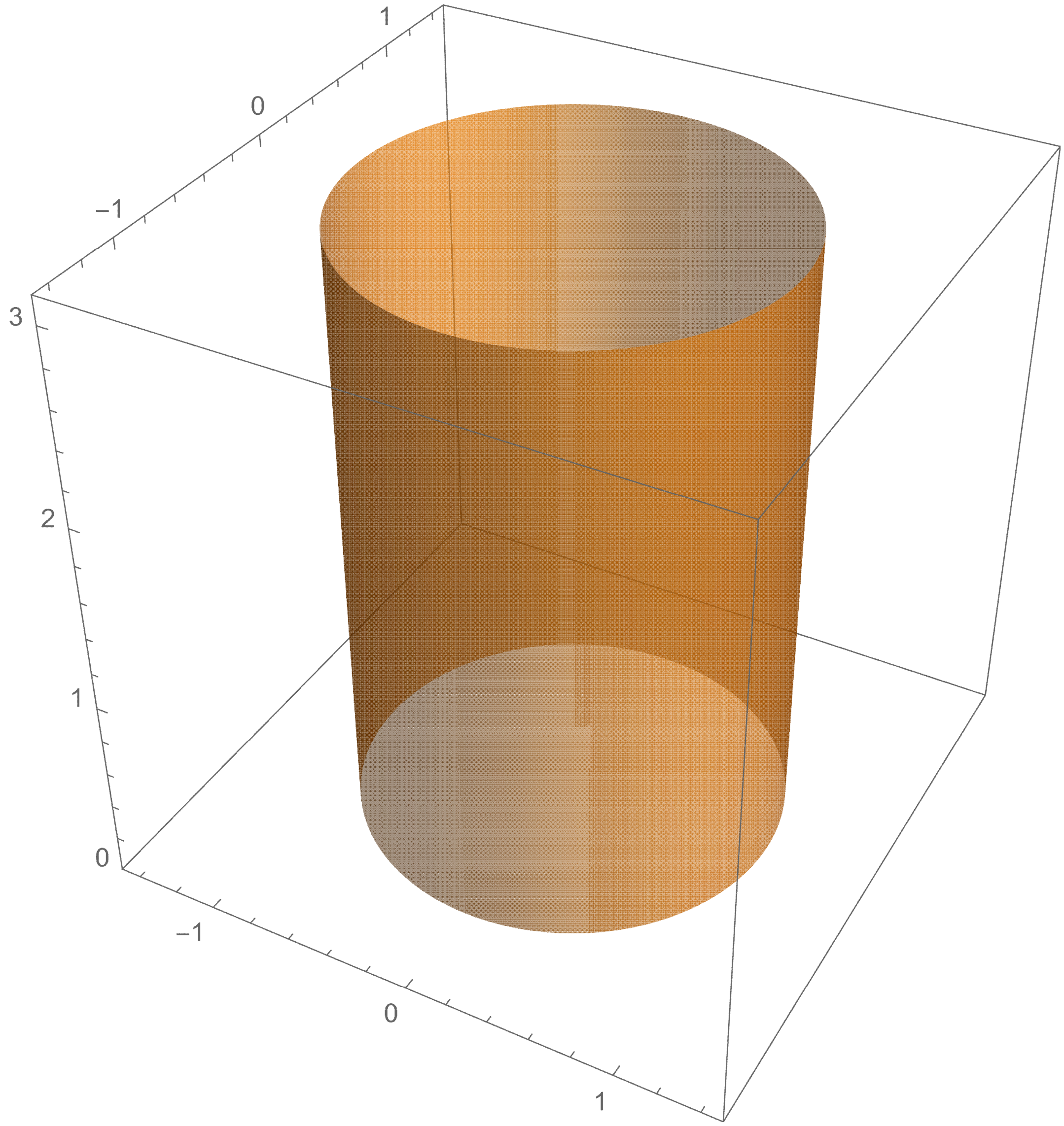
In[45]:=
```
Clear[cyl]; cyl = ParametricPlot3D[
  {Cos[t], Sin[t], z}, {z, 0, Pi}, {t, 0, 2*Pi}, PlotPoints → {101, 101},
  PlotStyle → {Opacity[0.5]}, Mesh → False ,
  PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, Pi}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]
```

Out[45]=



In[46]:= `Show[cyl]`

Out[46]=

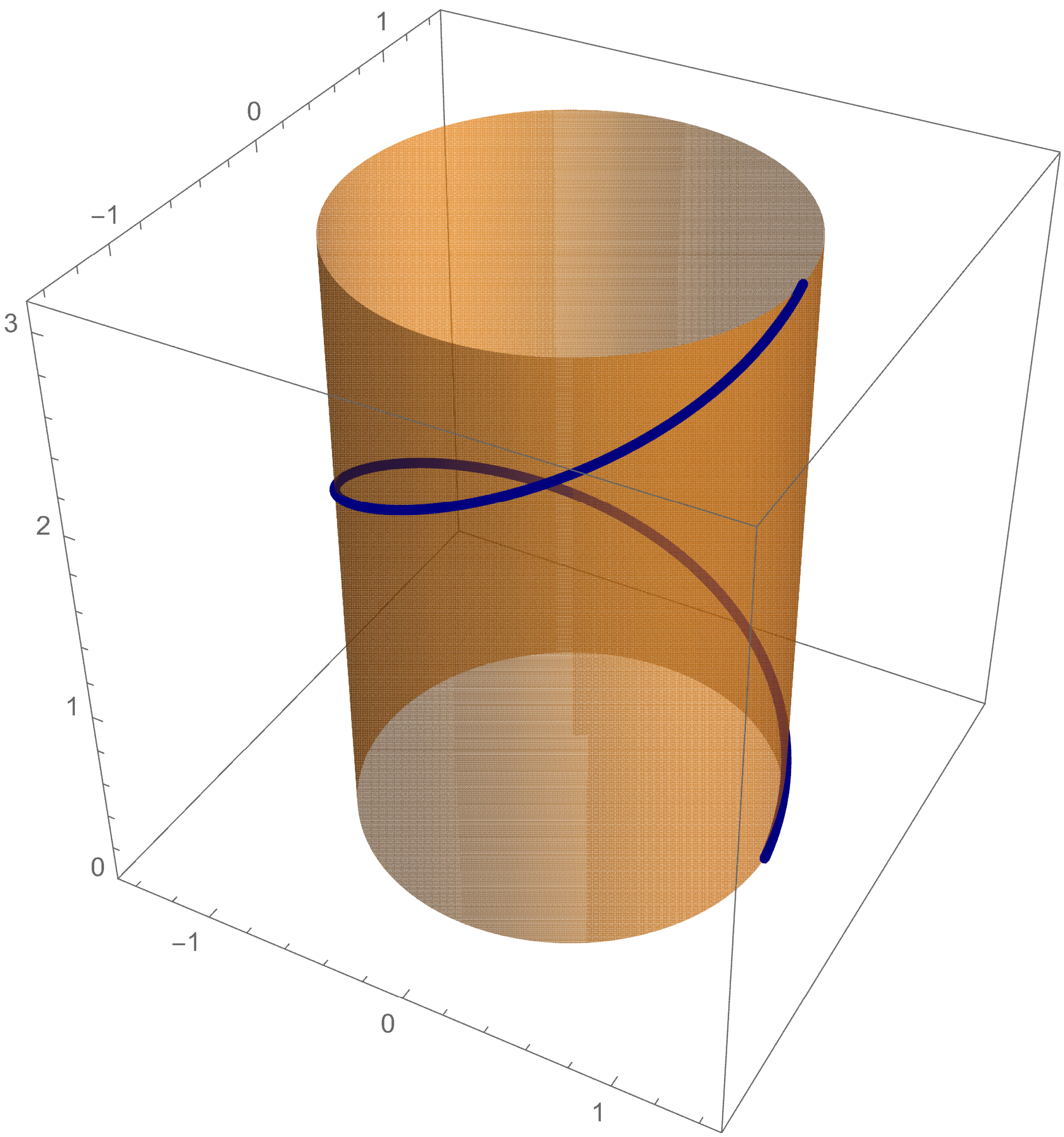


Helix

A helix is a special curve that lives on a cylinder:

In[47]:=
```
Show[cyl, ParametricPlot3D[
  {Cos[t], Sin[t], t/2}, {t, 0, 2*Pi}, PlotPoints → {101},
  PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
  PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, Pi}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]]
```

Out[47]=

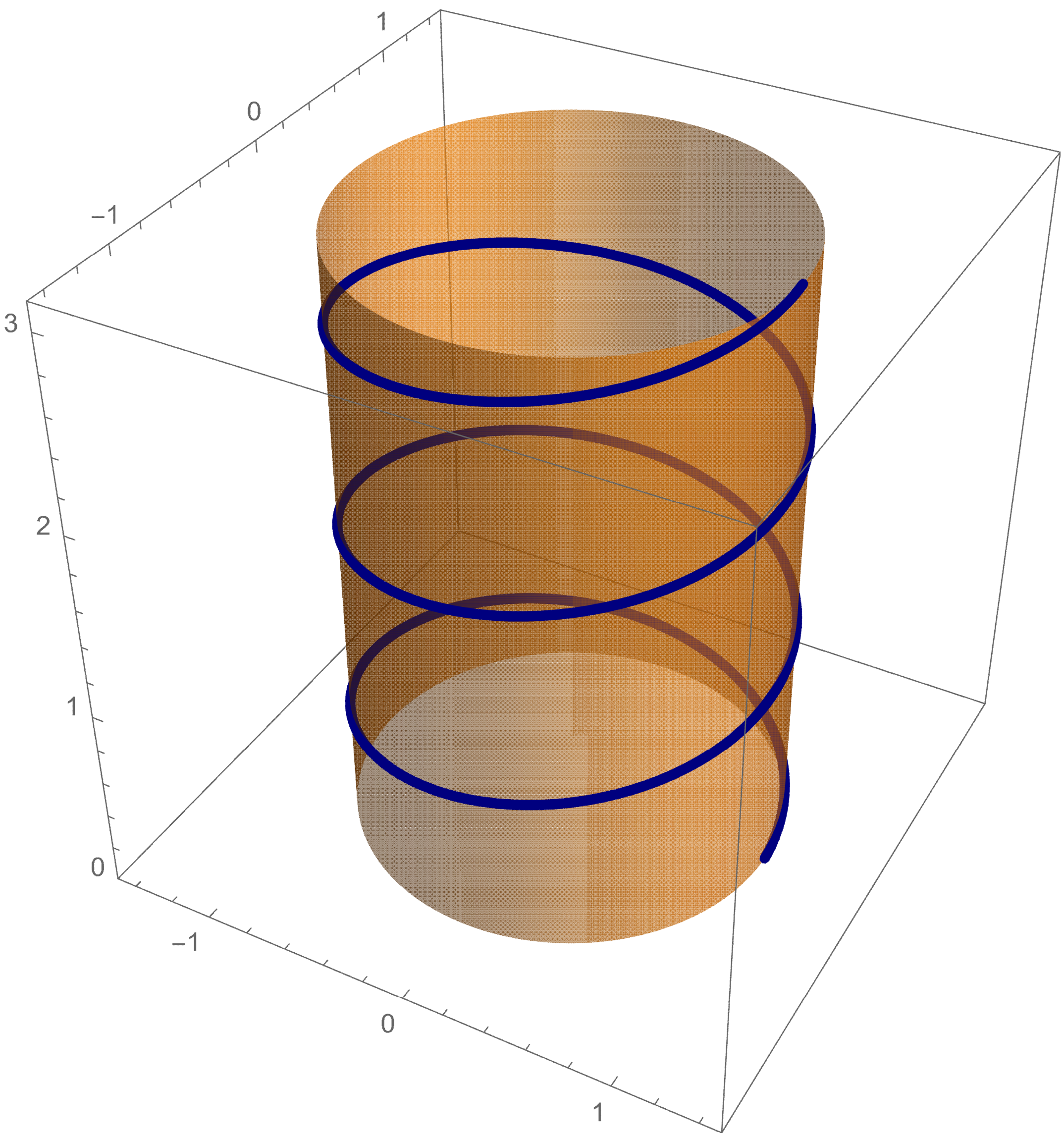


Or, winding up more as it climbs:

reproduce the picture below (13)

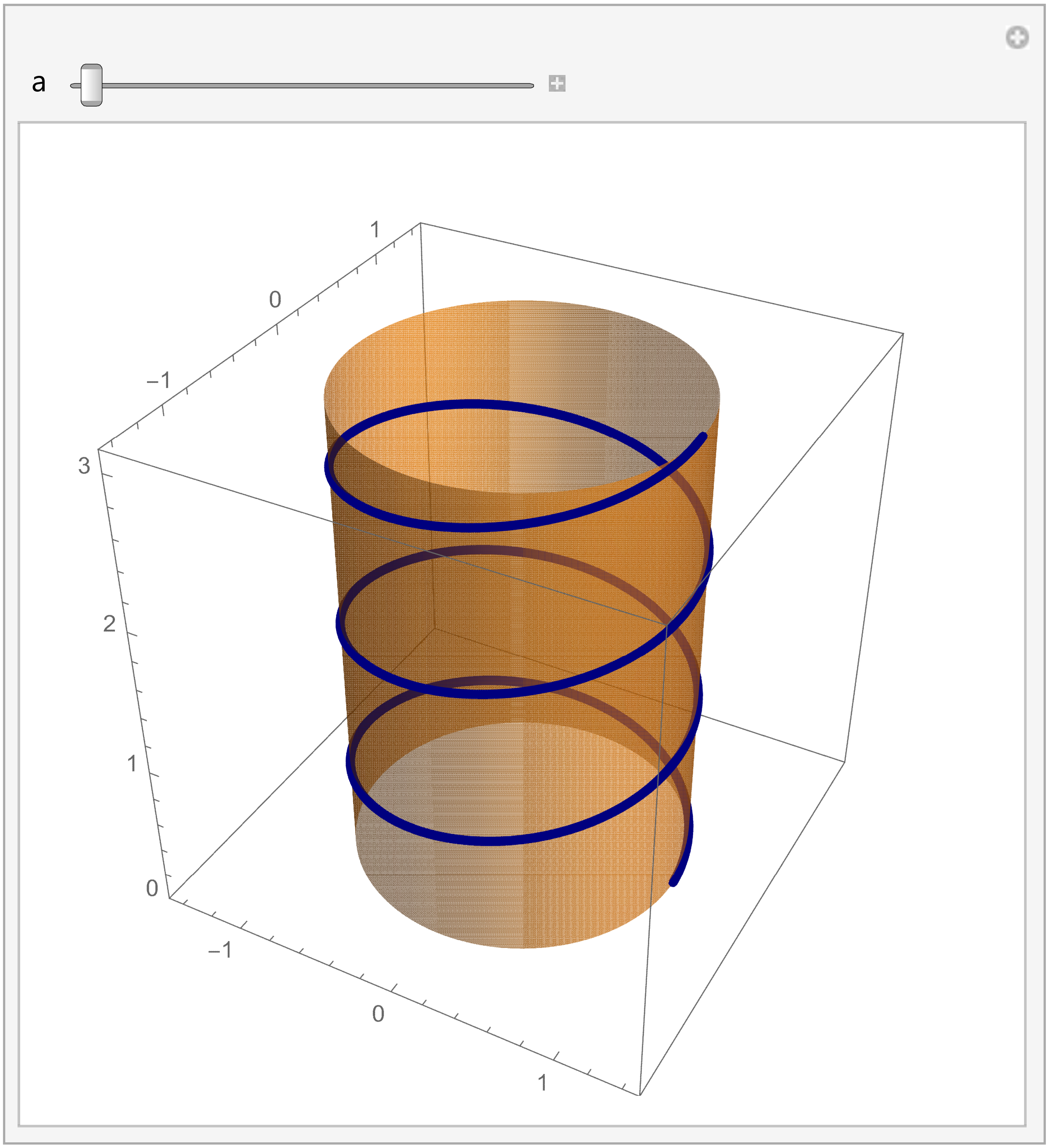
In[48]:=
```
Show[cyl, ParametricPlot3D[
   {Cos[t], Sin[t], t/6}, {t, 0, 6*Pi}, PlotPoints → {301},
   PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
   PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, Pi}},
   Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
   ImageSize → 400
  ]]
```

Out[48]=



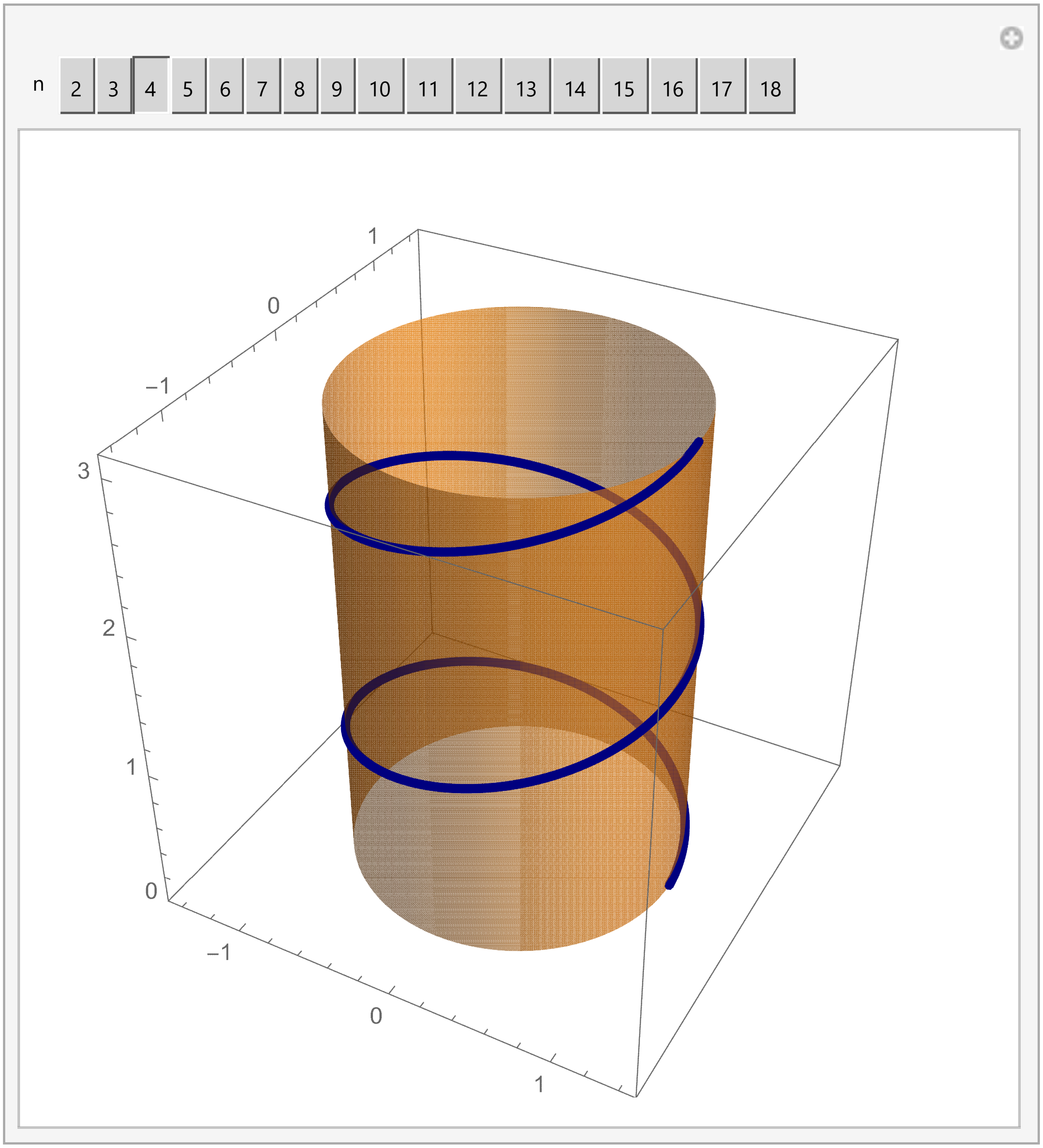
In[49]:=
```
Manipulate[Show[cyl, ParametricPlot3D[
    {Cos[a + t], Sin[a + t], t/6}, {t, 0, 6 * Pi}, PlotPoints → {301},
    PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
    PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, Pi}},
    Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
    ImageSize → 400
  ]], {a, 0, 2 Pi, ControlPlacement → Top}]
```

Out[49]=



In[50]:=
```
Manipulate[Show[cyl, ParametricPlot3D[
    {Cos[t], Sin[t], t/n}, {t, 0, n*Pi}, PlotPoints → {301},
    PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
    PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, Pi}},
    Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
    ImageSize → 400
  ]], {{n, 4}, Range[2, 18], ControlPlacement → Top, Setter}]
```

Out[50]=

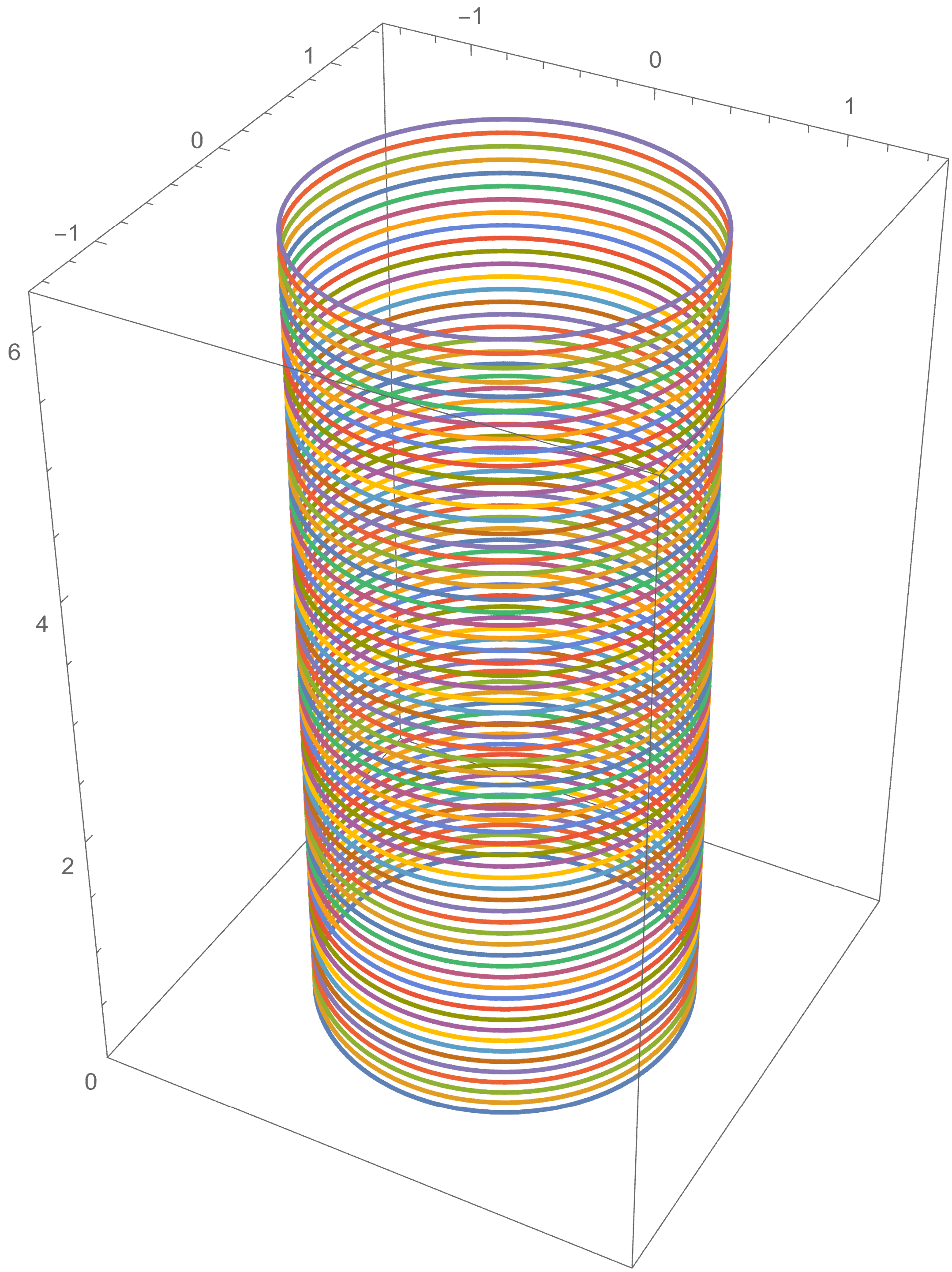


Vase

We constructed the unit cylinder by lifting the unit circle at different z-levels.

In[51]:=
```
Clear[zz]; ParametricPlot3D[
  Evaluate[Table[{Cos[t], Sin[t], zz}, {zz, 0, 2 Pi, Pi/32}]], {t, 0, 2*Pi},
  PlotPoints → 101, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {0, 2 Pi}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1.5},
  ImageSize → 400
 ]
```

Out[51]=



Next we will change the radius of the circle depending on the z-level. At the level z we will draw the circle with radius 2+Sin[z]. This will give us a nice vase. To make this construction more transparent, we will write the formula for the circle and its level separately: The circle with the radius 2+Sin[z] at the level 0 is

```
In[52]:= (2 + Sin[z]) {Cos[t], Sin[t], 0}

Out[52]= {Cos[t] (2 + Sin[z]), Sin[t] (2 + Sin[z]), 0}
```

Then we add the level

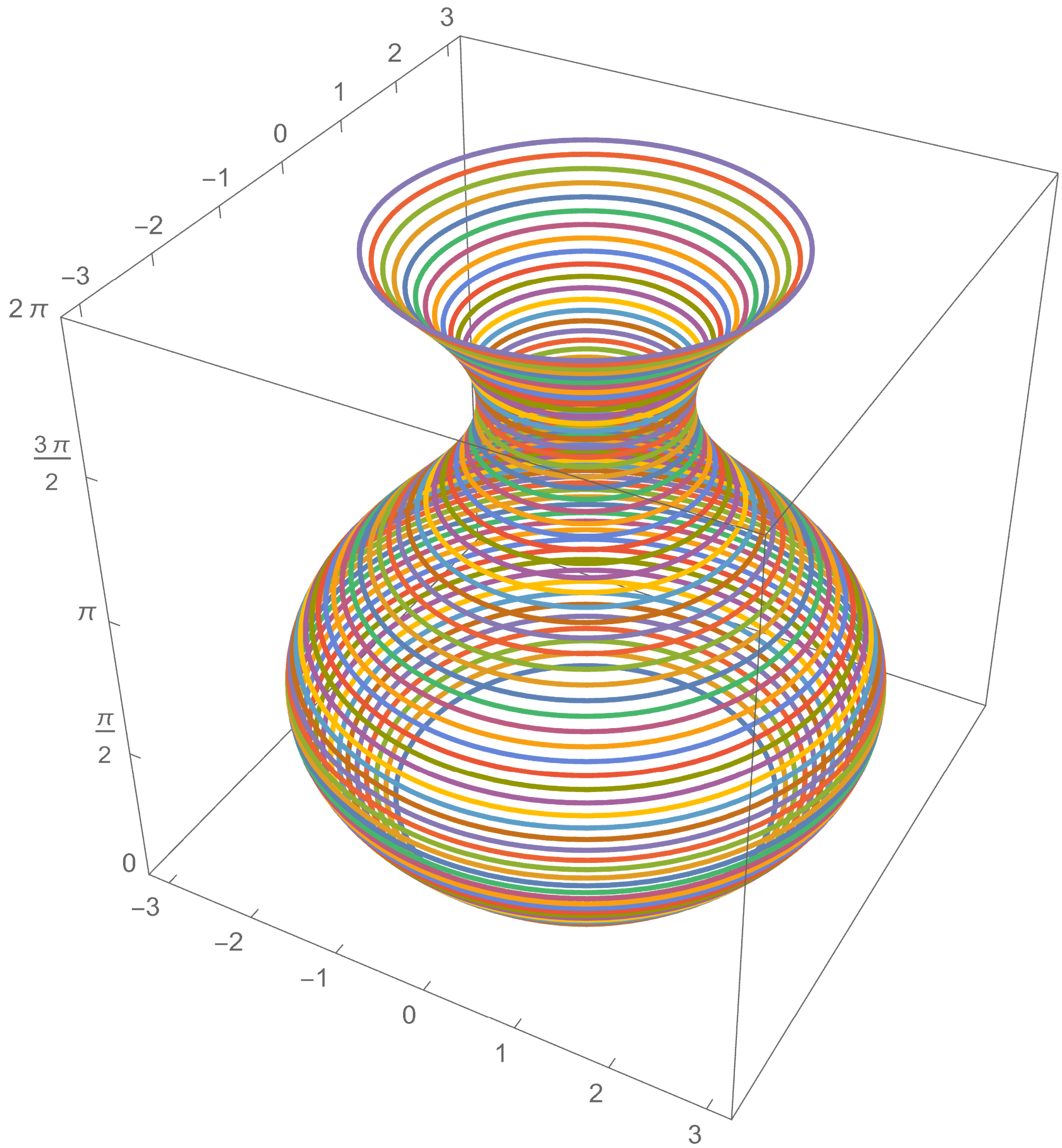
```
In[53]:= (2 + Sin[z]) {Cos[t], Sin[t], 0} + {0, 0, z}

Out[53]= {Cos[t] (2 + Sin[z]), Sin[t] (2 + Sin[z]), z}
```

Notice that all the points that we are drawing behave as vectors.

In[54]:=
```
Clear[z]; ParametricPlot3D[
 Evaluate[Table[(2 + Sin[z]) {Cos[t], Sin[t], 0} + {0, 0, z},
   {z, 0, 2 Pi, Pi/32}]], {t, 0, 2 * Pi}, PlotPoints → 101,
 PlotRange → {{-3.25, 3.25}, {-3.25, 3.25}, {0, 2 Pi}},
 Axes → True, Boxed → True,
 Ticks → {Range[-4, 4, 1], Range[-4, 4, 1], Range[-Pi, 4 Pi, Pi/2]},
 BoxRatios → {1, 1, 1}, ImageSize → 400
]
```

Out[54]=

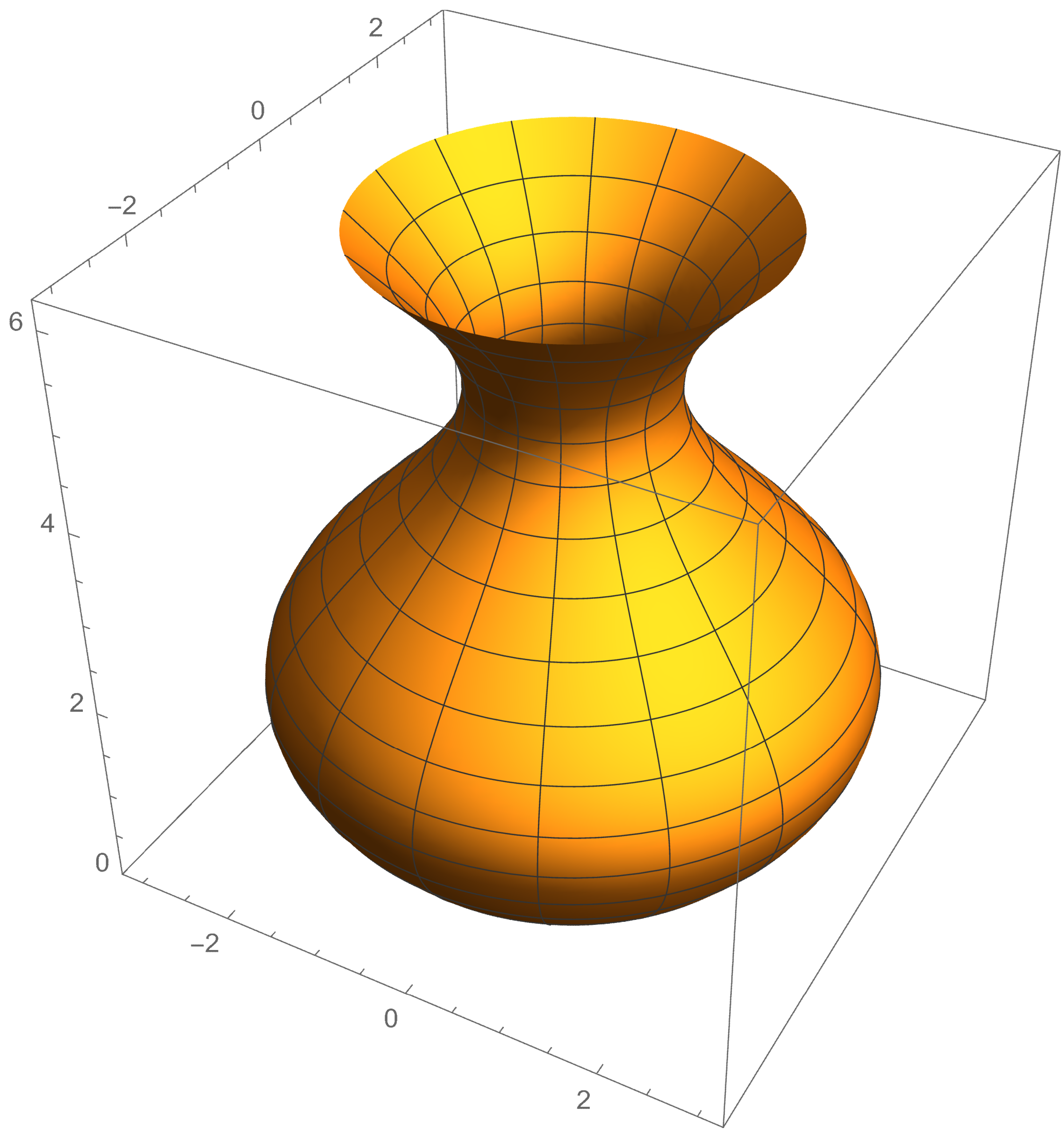


Or, drawn as a surface:

reproduce the picture below, try to produce several different vases in the homework. (14)

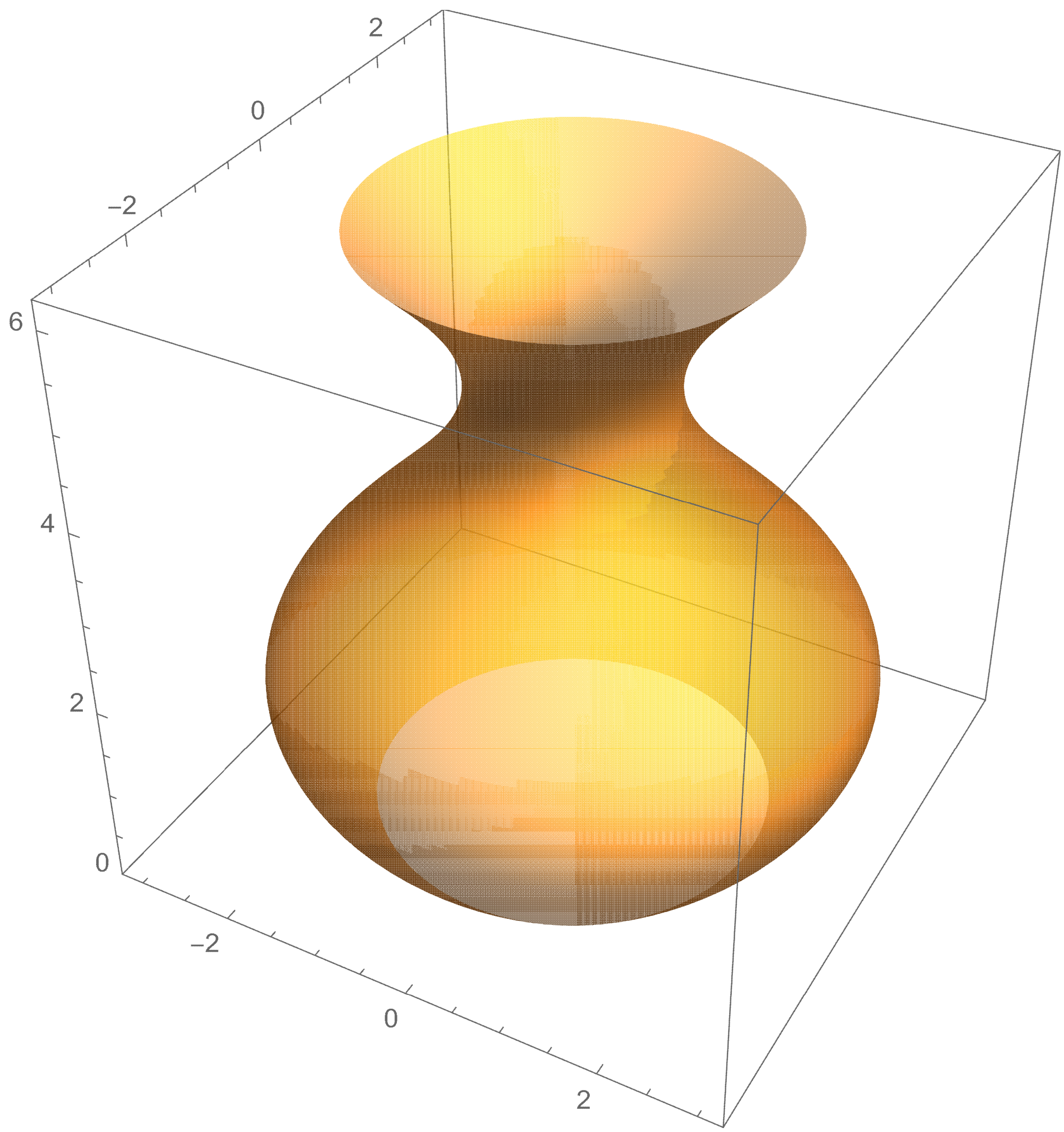
In[55]:=
```
Clear[z]; ParametricPlot3D[
 (2 + Sin[z]) {Cos[t], Sin[t], 0} + {0, 0, z}, {z, 0, 2 Pi}, {t, 0, 2 * Pi},
 PlotPoints → {101, 101},
 PlotRange → {{-3.25, 3.25}, {-3.25, 3.25}, {0, 2 Pi}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

Out[55]=



In[56]:=
```
Clear[z]; vase = ParametricPlot3D[
  (2 + Sin[z]) {Cos[t], Sin[t], 0} + {0, 0, z}, {z, 0, 2 Pi}, {t, 0, 2 * Pi},
  PlotPoints → {101, 101}, PlotStyle → {Opacity[0.5]}, Mesh → False ,
  PlotRange → {{-3.25, 3.25}, {-3.25, 3.25}, {0, 2 Pi}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]
```

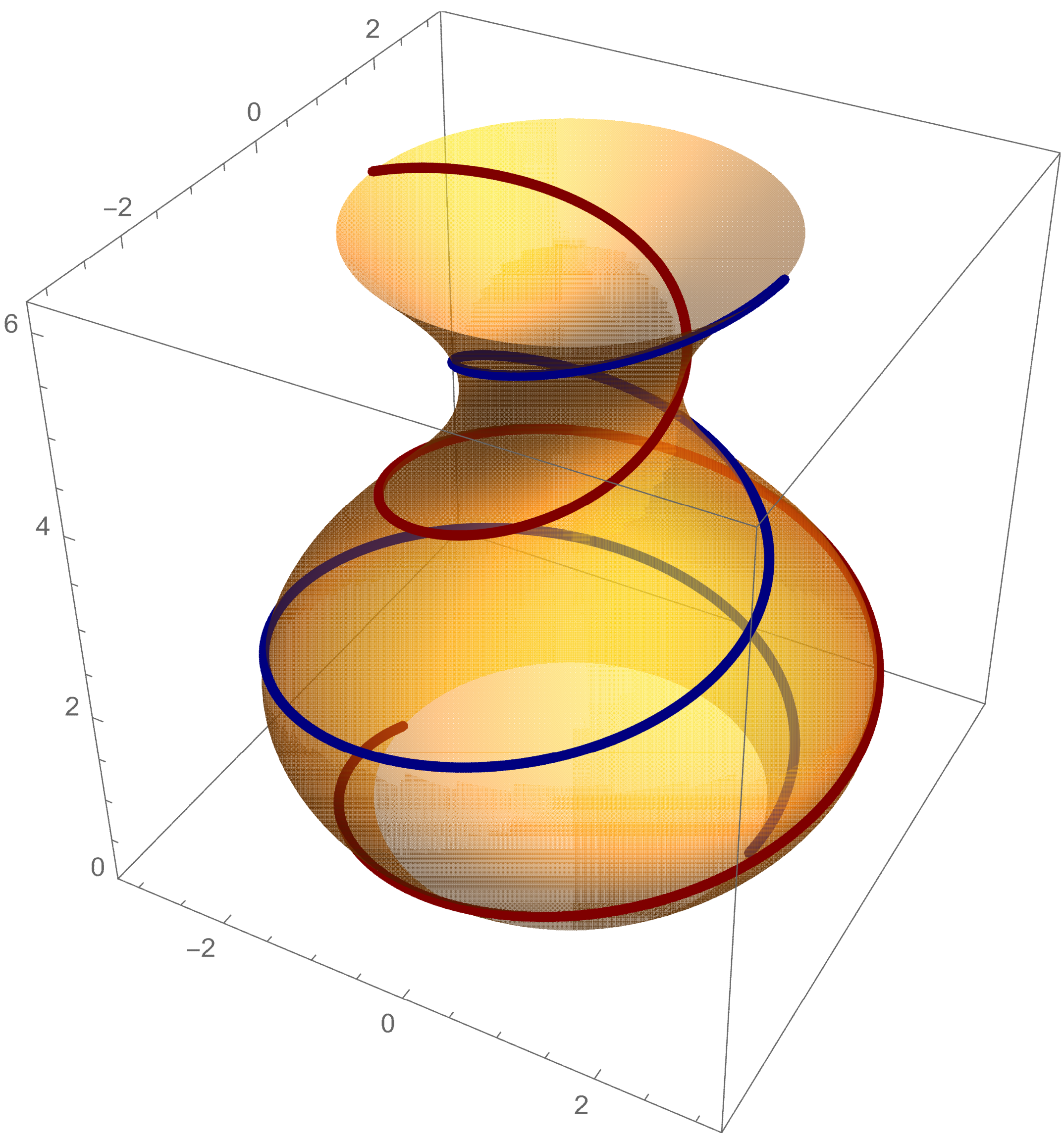
Out[56]=



Now, what would be a helix on a vase?

In[57]:=
```
Show[vase, ParametricPlot3D[
  (2 + Sin[t/2]) {Cos[t], Sin[t], 0} + {0, 0, t/2}, {t, 0, 4 * Pi},
  PlotPoints → {301}, PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]}
 ], ParametricPlot3D[
  (2 + Sin[t/2]) {Cos[Pi + t], Sin[Pi + t], 0} + {0, 0, t/2}, {t, 0, 4 * Pi},
  PlotPoints → {301}, PlotStyle → {Thickness[0.01], RGBColor[0.5, 0, 0]}
 ], PlotRange → {{-3.25, 3.25}, {-3.25, 3.25}, {0, 2 Pi}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400]
```

Out[57]=

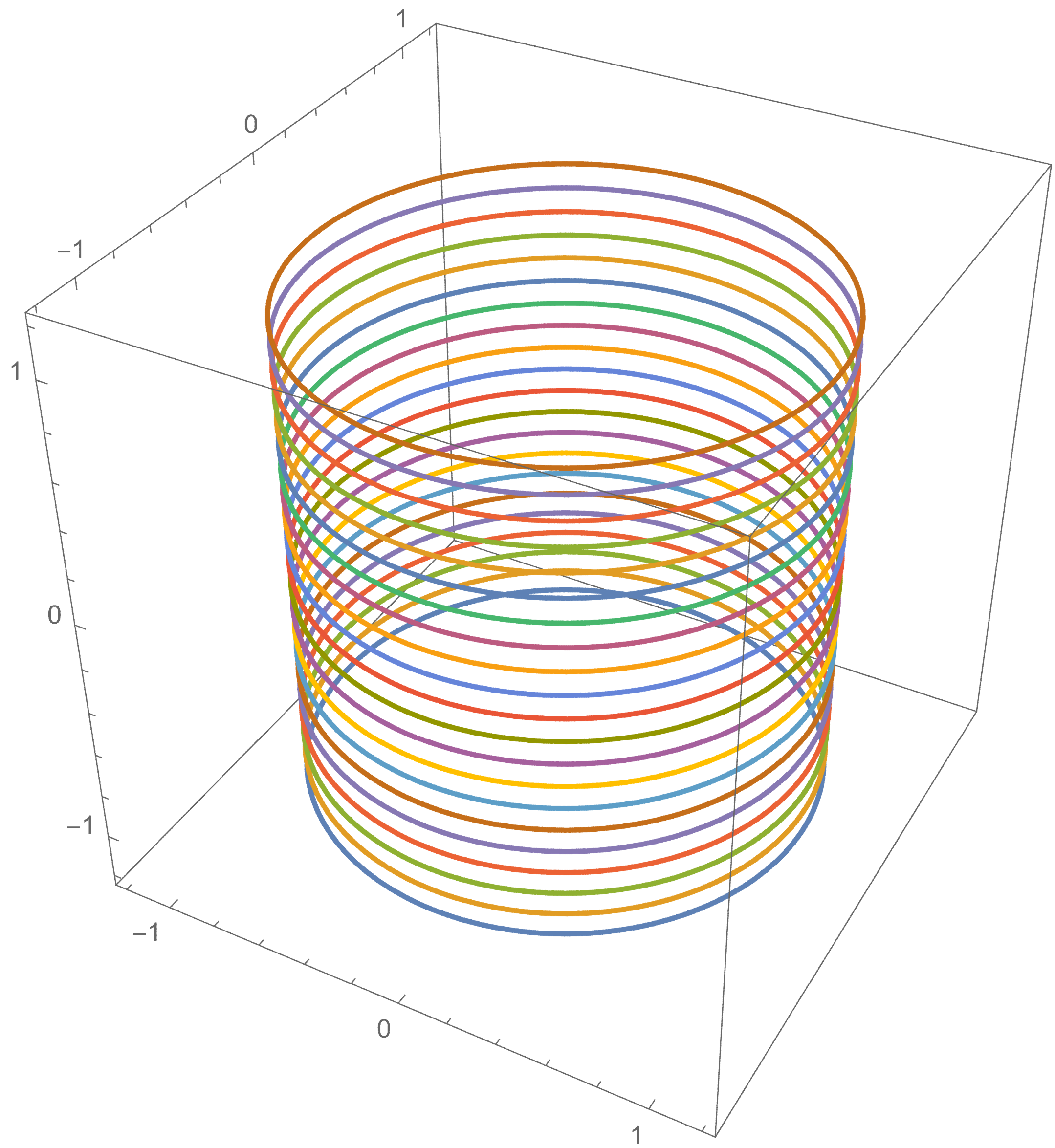


Sphere

We can think of a sphere as a collection of circles of different radius at different levels. It turns out that we have to take the radius Sin[ϕ] at the level Cos[ϕ].

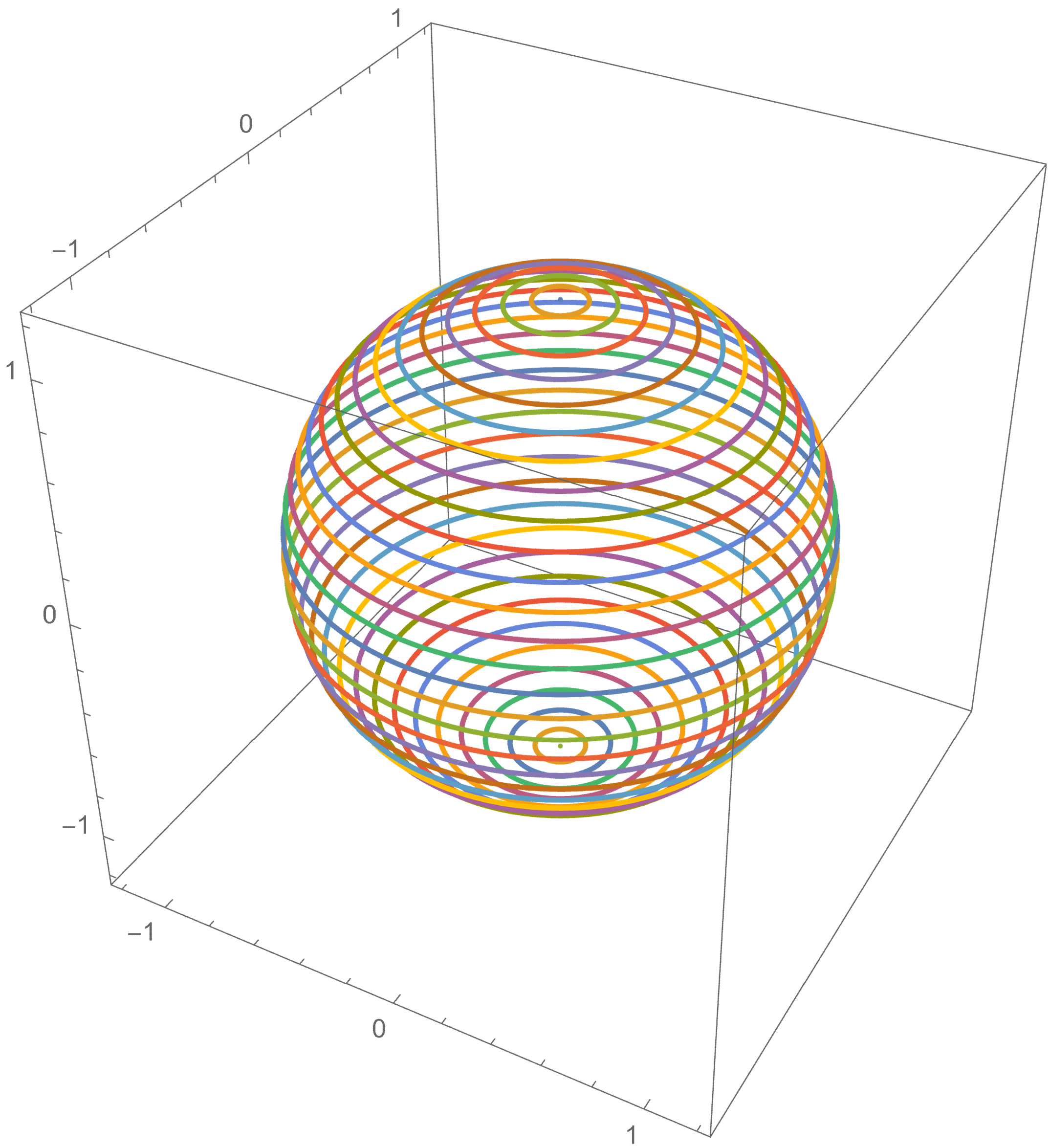
In[58]:=
```
Clear[z]; ParametricPlot3D[
 Evaluate[Table[{Cos[t], Sin[t], z}, {z, -1, 1, 0.1}]], {t, 0, 2*Pi},
 PlotPoints → 101,
 PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

Out[58]=



```
In[59]:= Clear[z]; ParametricPlot3D[
  Evaluate[Table[Sin[ϕ] {Cos[t], Sin[t], 0} + {0, 0, Cos[ϕ]},
    {ϕ, 0, Pi, Pi/32}]], {t, 0, 2 * Pi}, PlotPoints → 101,
  PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]
```

Out[59]=

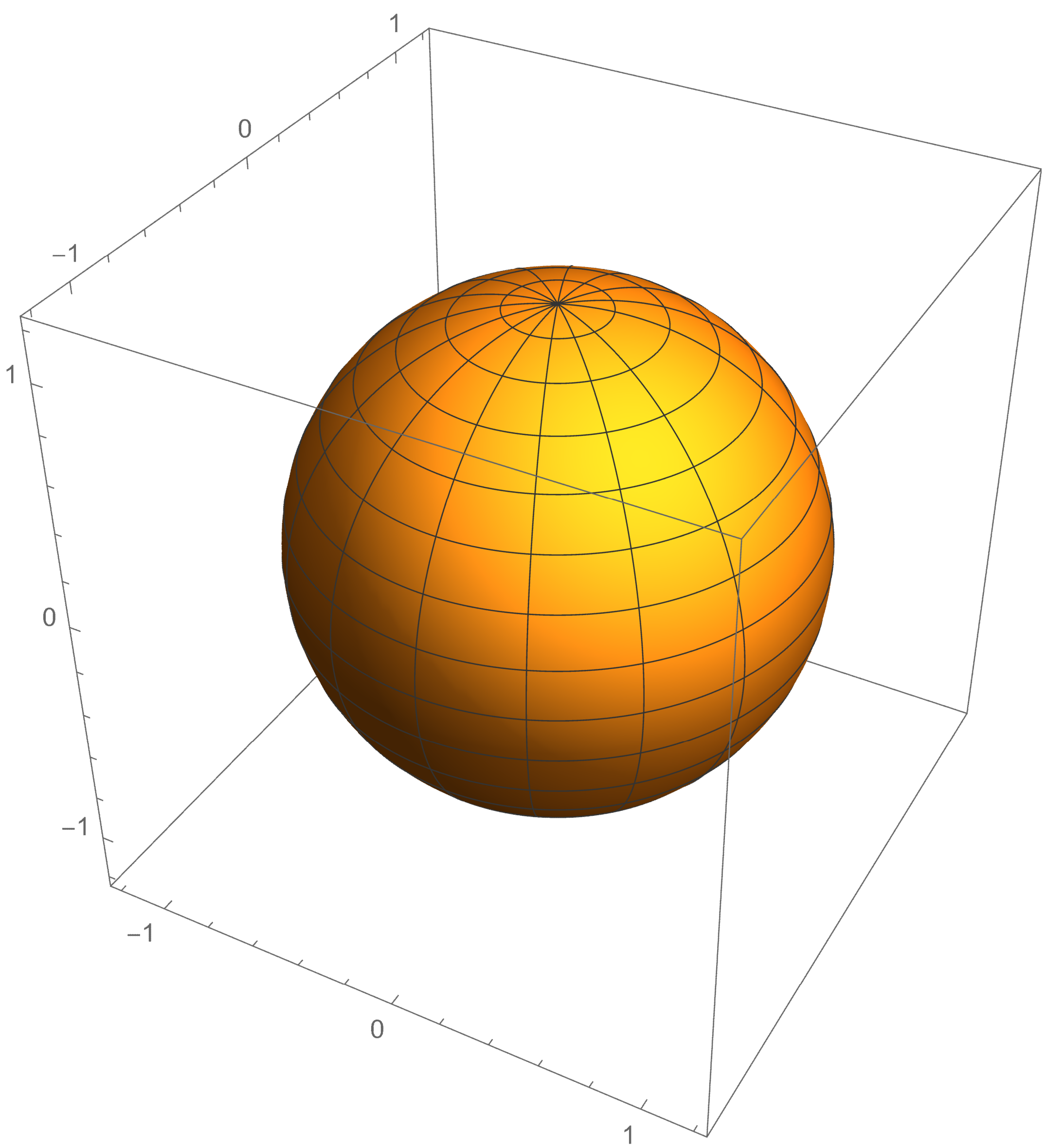


Or presented as a surface:

reproduce the picture below (15)

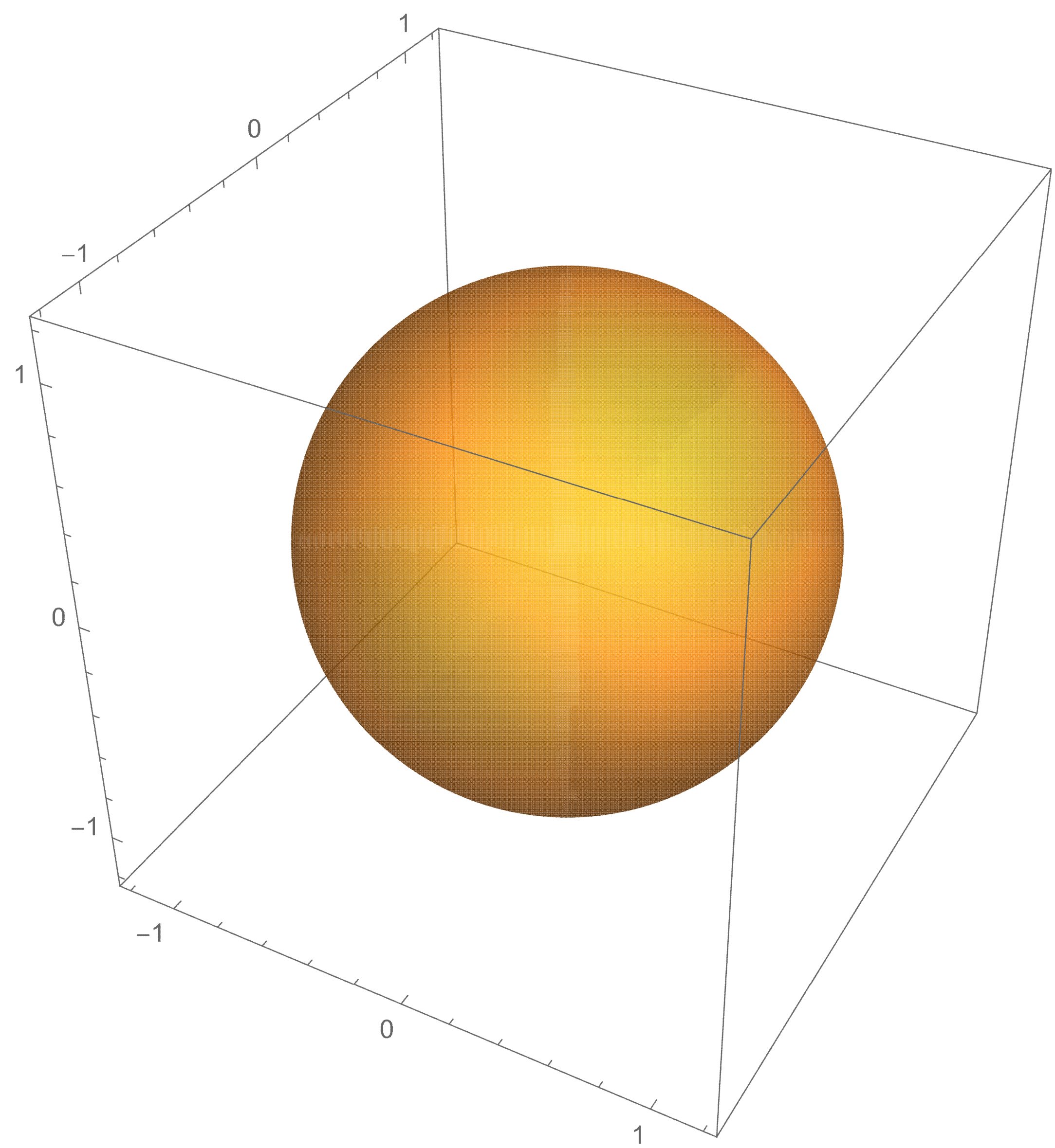
In[60]:=
```
ParametricPlot3D[
 Sin[ϕ] {Cos[t], Sin[t], 0} + {0, 0, Cos[ϕ]}, {ϕ, 0, Pi}, {t, 0, 2*Pi},
 PlotPoints → {101, 201},
 PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
 Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
 ImageSize → 400
]
```

Out[60]=



In[61]:=
```
sph = ParametricPlot3D[
  Sin[ϕ] {Cos[t], Sin[t], 0} + {0, 0, Cos[ϕ]}, {ϕ, 0, Pi}, {t, 0, 2*Pi},
  PlotPoints → {101, 201}, PlotStyle → {Opacity[0.5]}, Mesh → False,
  PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]
```

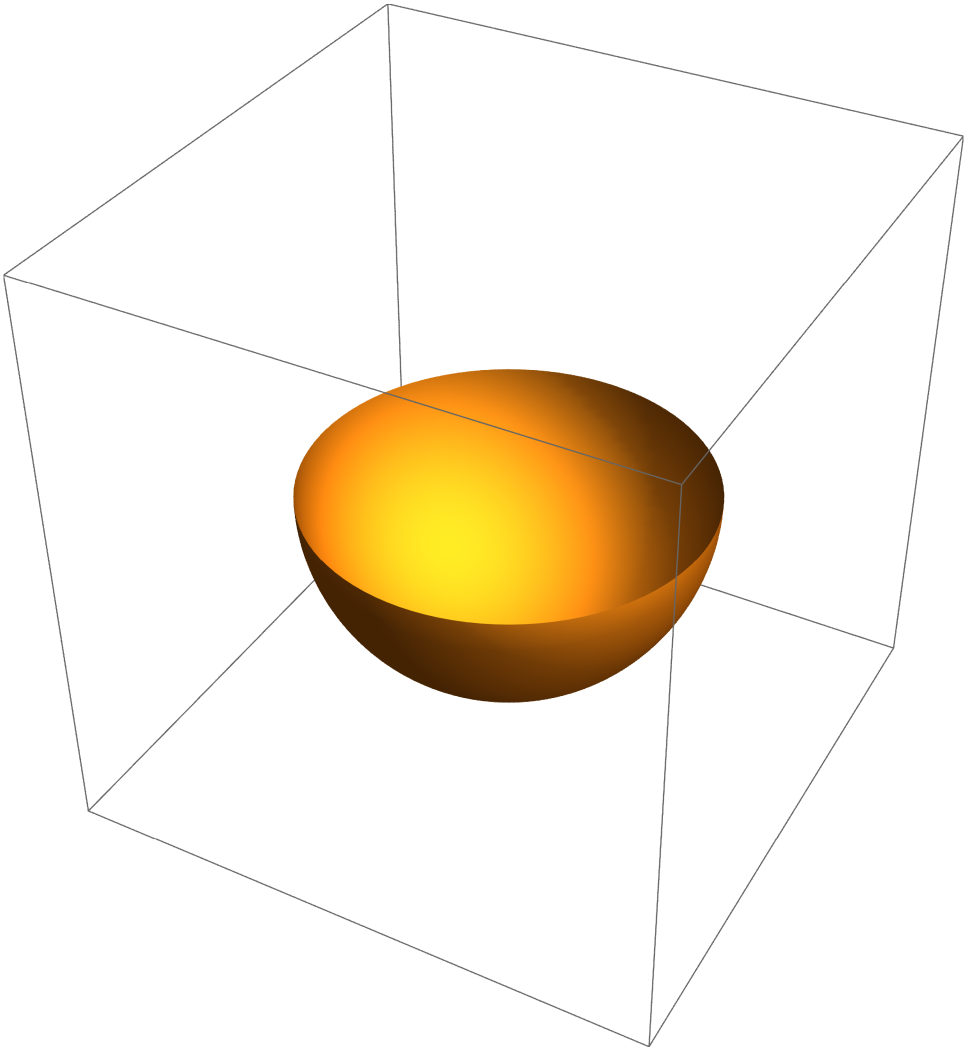
Out[61]=



Now explore how parameters t and ϕ form the unit sphere:

In[62]:=
```
ParametricPlot3D[{Cos[t] Sin[ϕ], Sin[t] Sin[ϕ], Cos[ϕ]}, {t, 0, 2 π},
 {ϕ, π/2, π}, PlotPoints → {101, 51}, Mesh → False,
 PlotRange → {{-1.5`, 1.5`}, {-1.5`, 1.5`}, {-1.5`, 1.5`}},
 Axes → False, BoxRatios → {1, 1, 1}]
```

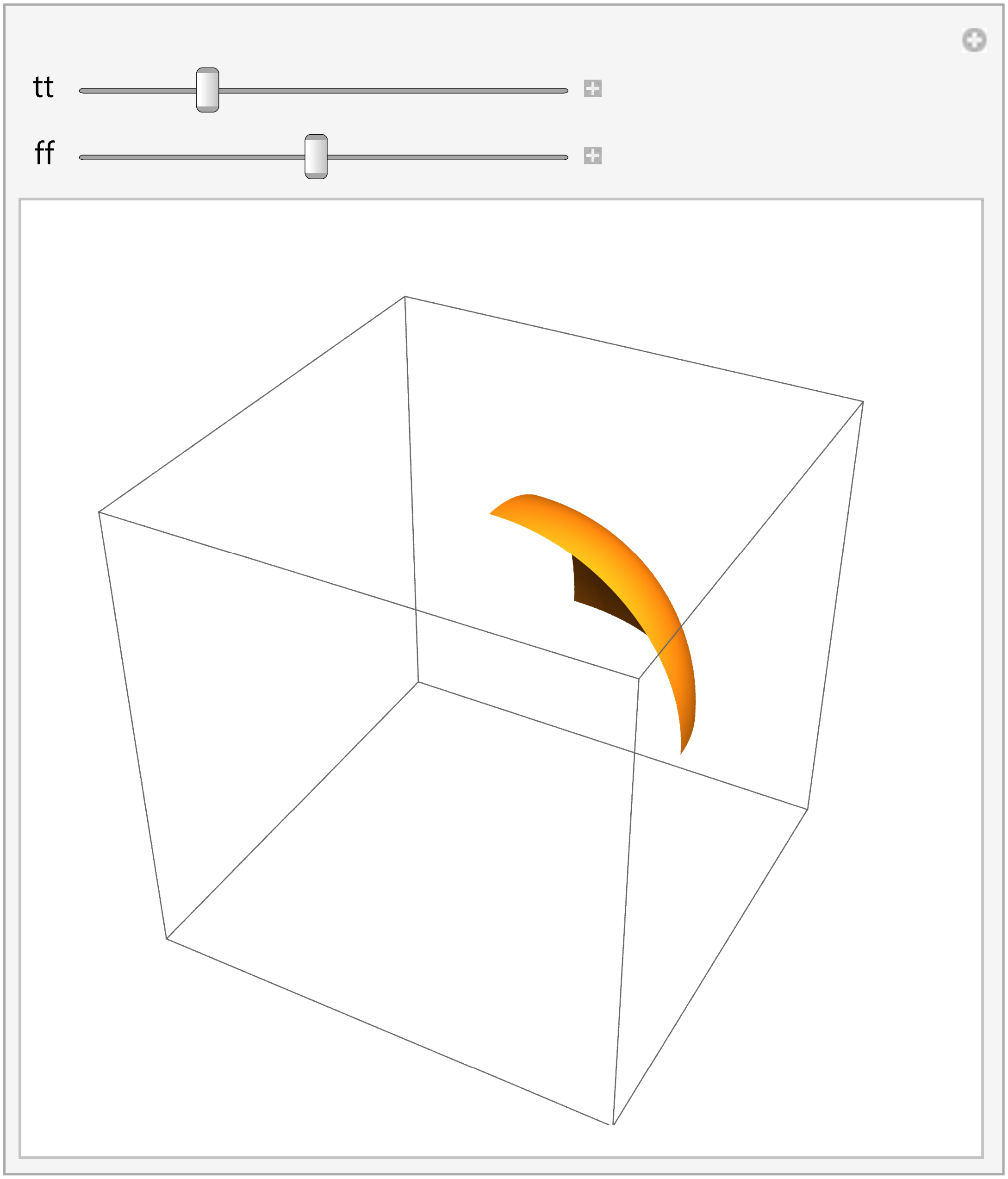
Out[62]=



With Manipulate[]

In[63]:=
```
Clear[ff, tt];
Manipulate[ParametricPlot3D[{Cos[t] Sin[ϕ], Sin[t] Sin[ϕ], Cos[ϕ]},
  {t, 0, tt}, {ϕ, 0, ff}, PlotPoints → {101, 51}, Mesh → False,
  PlotRange → {{-1.25`, 1.25`}, {-1.25`, 1.5`}, {-1.25`, 1.25`}},
  Axes → False, BoxRatios → {1, 1, 1}], {{tt, π/2}, 0.1, 2 π},
 {{ff, π/2}, 0.1, π}, ControlPlacement → Top]
```

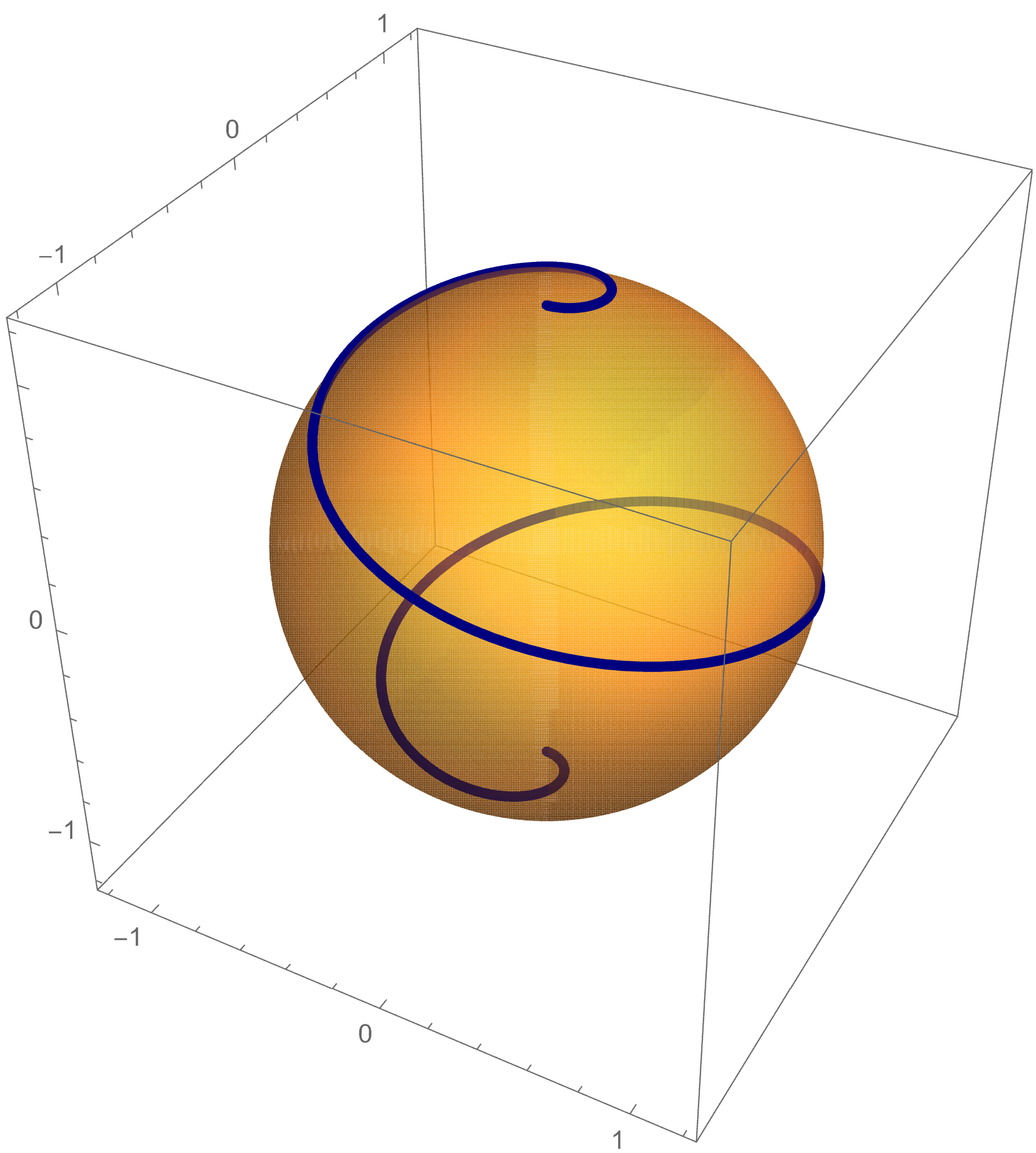
Out[63]=



A spherical helix

In[64]:=
```
Show[sph, ParametricPlot3D[
  Sin[t / 4] {Cos[t], Sin[t], 0} + {0, 0, Cos[t / 4]}, {t, 0, 4 * Pi},
  PlotPoints → {201}, PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
  PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
  Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
  ImageSize → 400
 ]]
```

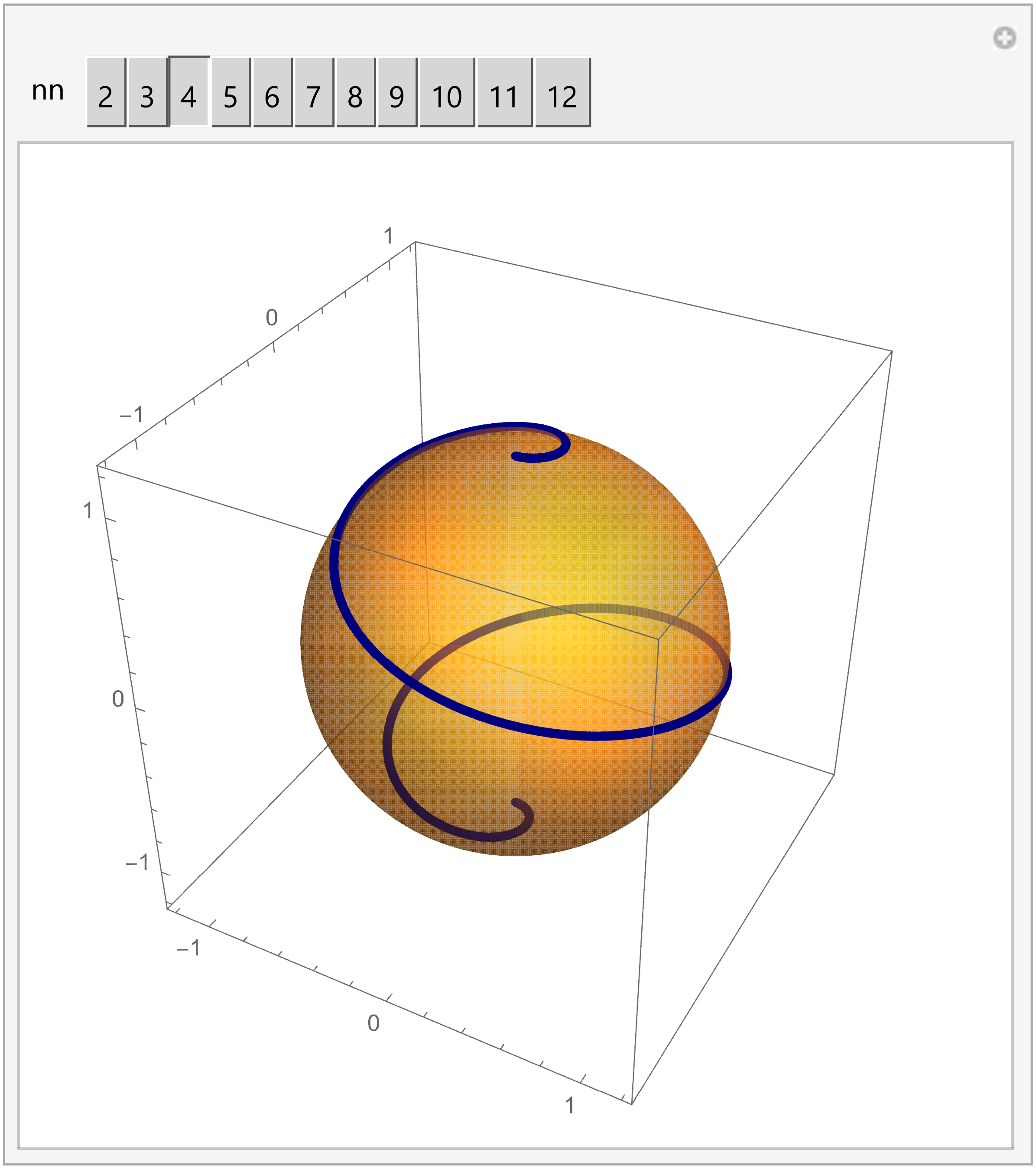
Out[64]=



And one more with Manipulate[]

In[65]:=
```
Manipulate[Show[sph, ParametricPlot3D[
    Sin[t / nn] {Cos[t], Sin[t], 0} + {0, 0, Cos[t / nn]}, {t, 0, nn * Pi},
    PlotPoints → {nn * 50}, PlotStyle → {Thickness[0.01], RGBColor[0, 0, 0.5]},
    PlotRange → {{-1.25, 1.25}, {-1.25, 1.25}, {-1.25, 1.25}},
    Axes → True, Boxed → True, Ticks → Automatic, BoxRatios → {1, 1, 1},
    ImageSize → 400
  ]], {{nn, 4}, Range[2, 12], ControlPlacement → Top, Setter}]
```

Out[65]=

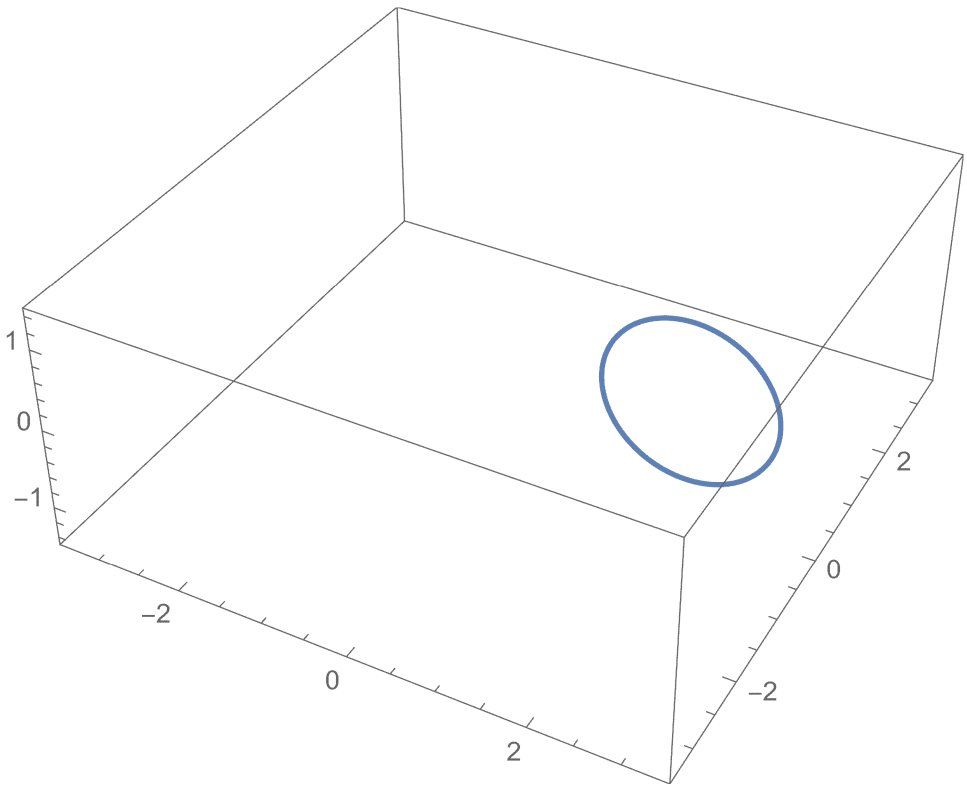


Torus

A torus is obtained when a circle in xz-plane centered at (2,0,0) is rotated around z-axis.

In[66]:=
```
ParametricPlot3D[{2, 0, 0} + {Cos[ϕ], 0, Sin[ϕ]}, {ϕ, 0, 2 π},
 PlotPoints → {101},
 PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
 Axes → True, BoxRatios → Automatic]
```

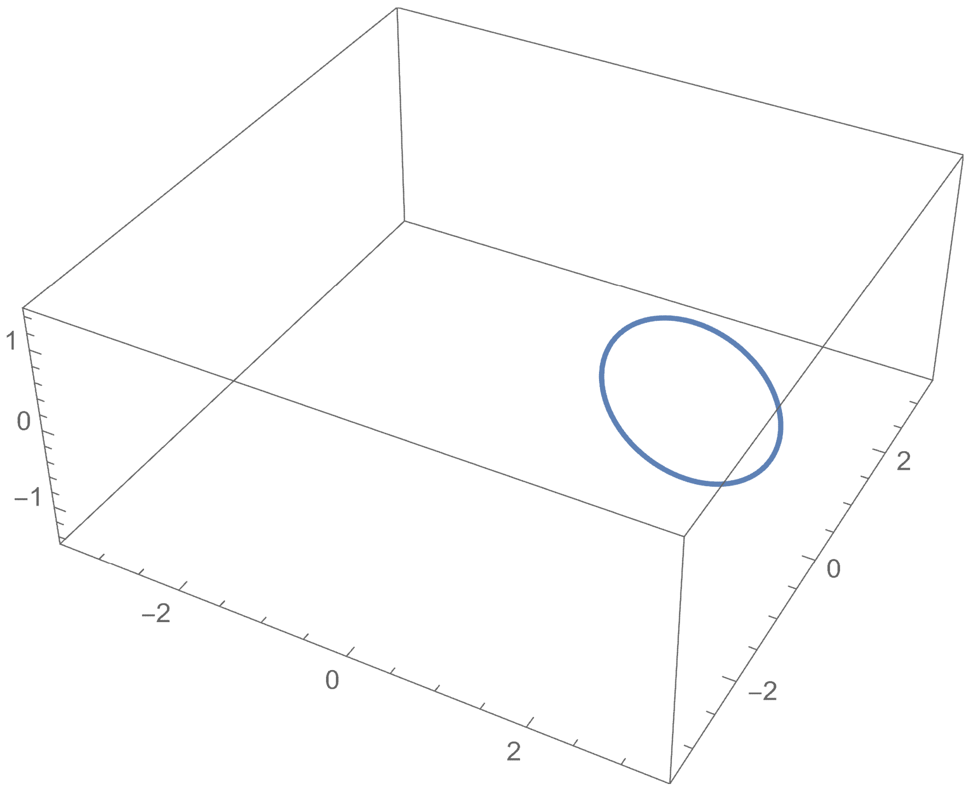
Out[66]=



It is useful to recognize the coordinate vectors in the preceding formula:

In[67]:=
```
ParametricPlot3D[2 {1, 0, 0} + Cos[ϕ] {1, 0, 0} + Sin[ϕ] {0, 0, 1},
 {ϕ, 0, 2 π}, PlotPoints → {101},
 PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
 Axes → True, BoxRatios → Automatic]
```

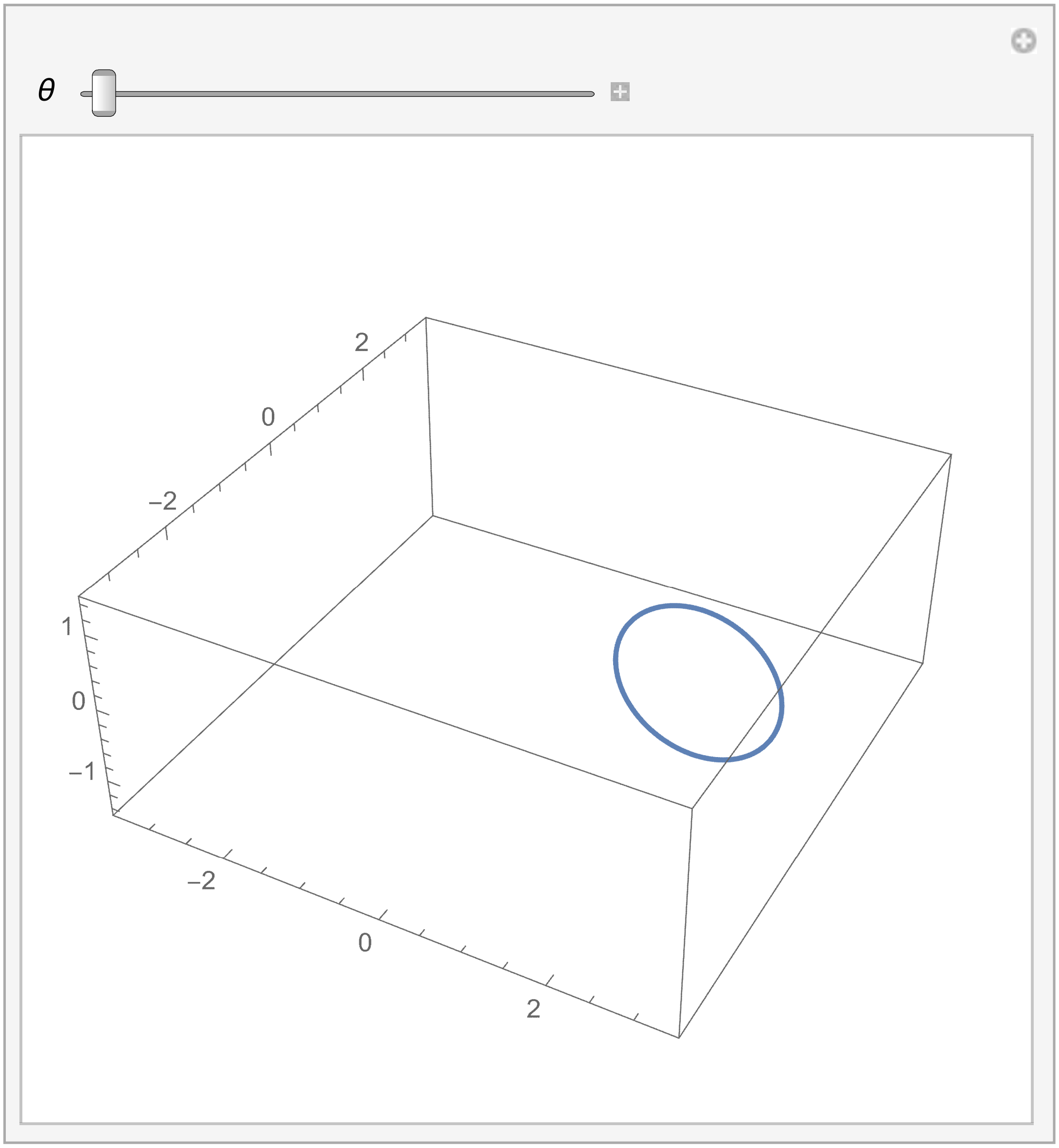
Out[67]=



To rotate the above circle around z-axis we need to replace the coordinate vector {1,0,0} with the vector in {Cos[θ],Sin[θ],0}. We illustrate this in Manipulate[]:

In[68]:=
```
Manipulate[
 ParametricPlot3D[2 {Cos[θ], Sin[θ], 0} + Cos[ϕ] {Cos[θ], Sin[θ], 0} +
   Sin[ϕ] {0, 0, 1}, {ϕ, 0, 2 π}, PlotPoints → {101},
  PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
  Axes → True, BoxRatios → Automatic], {θ, 0, 2 Pi, ControlPlacement → Top}]
```

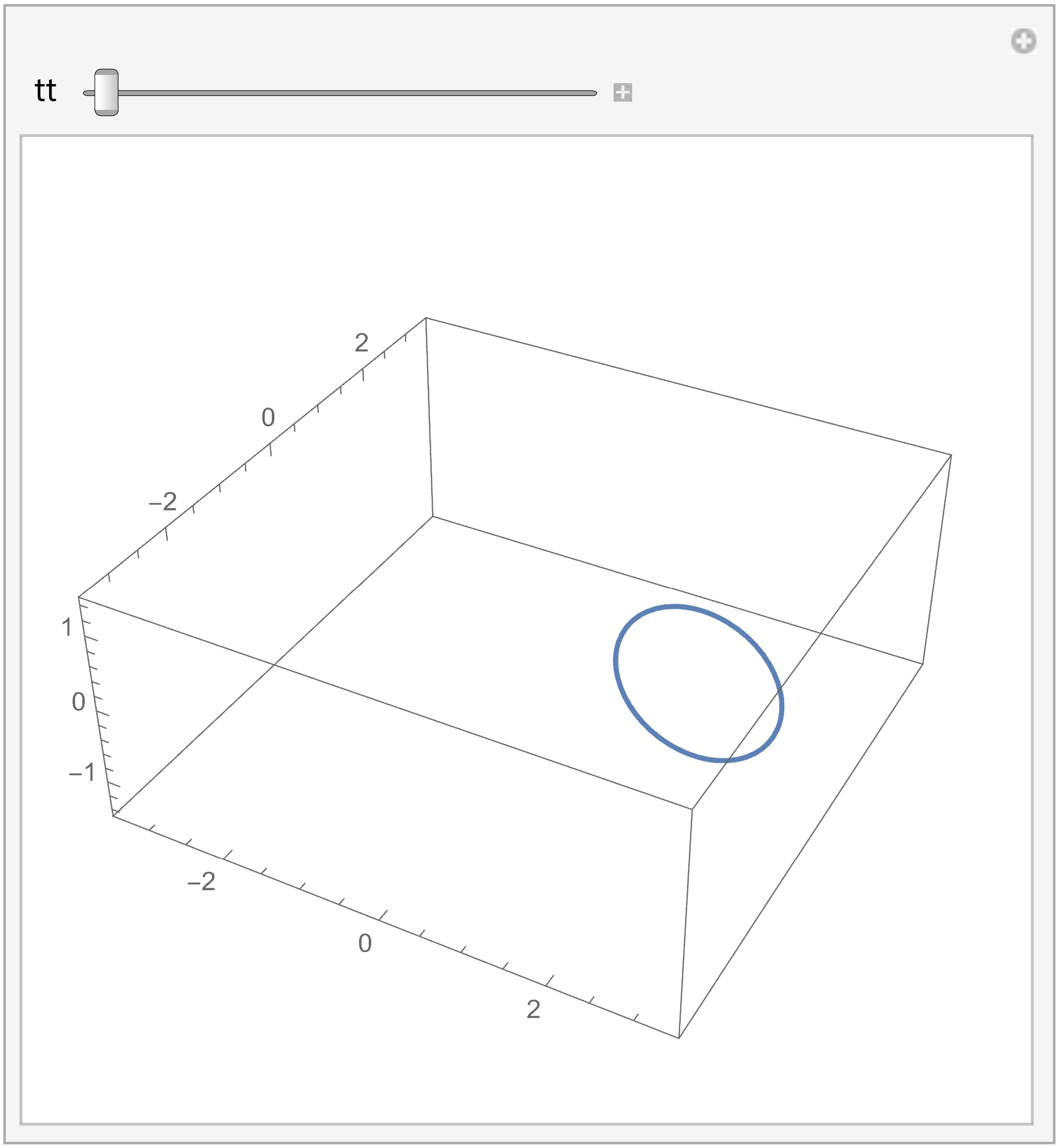
Out[68]=



Or, memorizing circles:

In[69]:=
```
Manipulate[
  ParametricPlot3D[
   Table[2 {Cos[θ], Sin[θ], 0} + Cos[ϕ] {Cos[θ], Sin[θ], 0} + Sin[ϕ] {0, 0, 1},
    {θ, 0, tt, Pi/16}], {ϕ, 0, 2 π}, PlotPoints → {51},
   PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
   Axes → True, BoxRatios → Automatic], {tt, Pi/32, 2 Pi, ControlPlacement → Top}]
```

Out[69]=

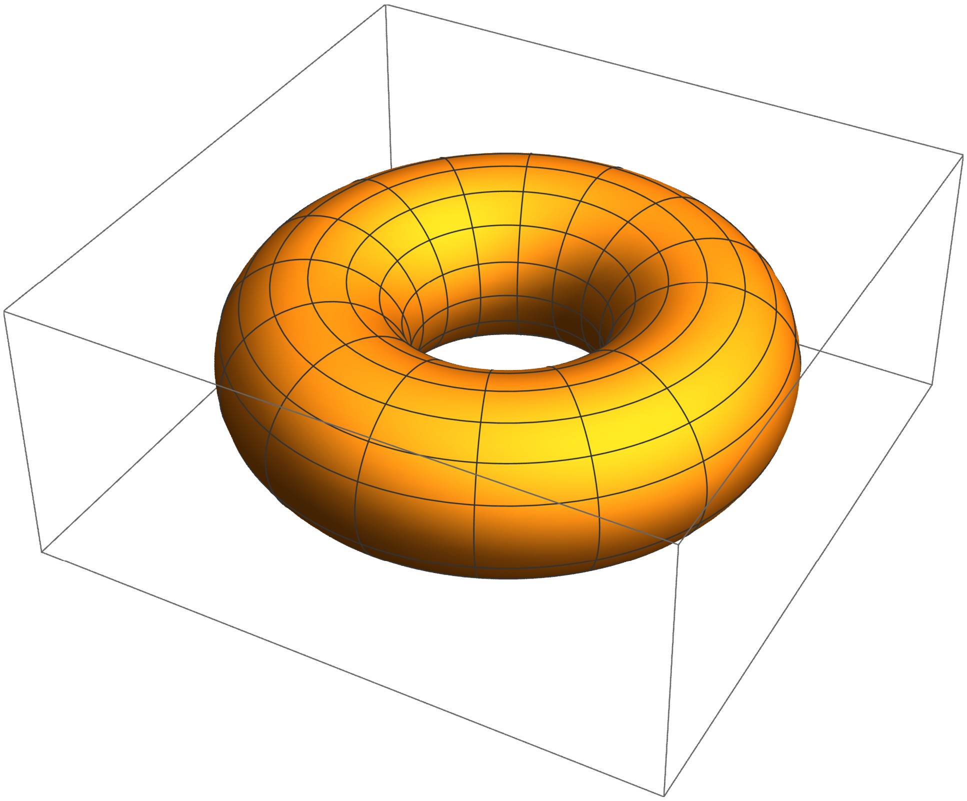


Torus as a surface:

reproduce the picture below, try to produce several different tori in your homework (16)

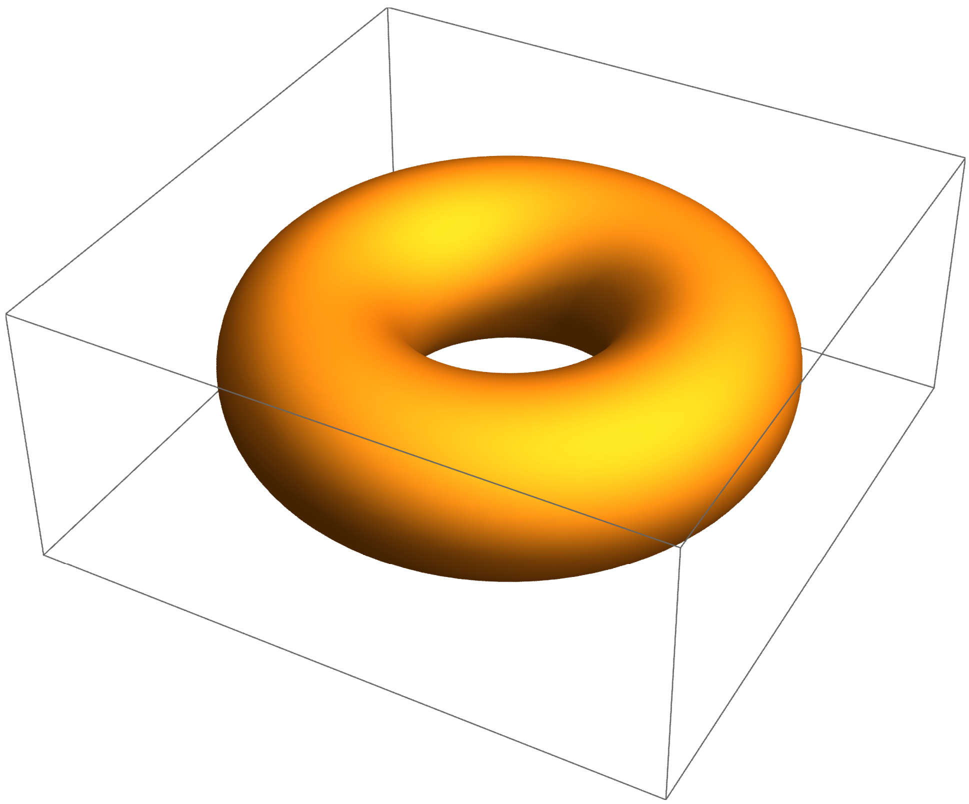
In[70]:=
```
ParametricPlot3D[{Cos[θ] (2 + Cos[ϕ]), Sin[θ] (2 + Cos[ϕ]), Sin[ϕ]},
  {θ, 0, 2 π}, {ϕ, 0, 2 π}, PlotPoints → {91, 61},
  PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
  Axes → False, BoxRatios → Automatic]
```

Out[70]=



In[71]:=
```
ParametricPlot3D[{Cos[θ] (2 + Cos[ϕ]), Sin[θ] (2 + Cos[ϕ]), Sin[ϕ]},
  {θ, 0, 2 π}, {ϕ, 0, 2 π}, PlotPoints → {91, 61}, Mesh → False,
  PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
  Axes → False, BoxRatios → Automatic]
```

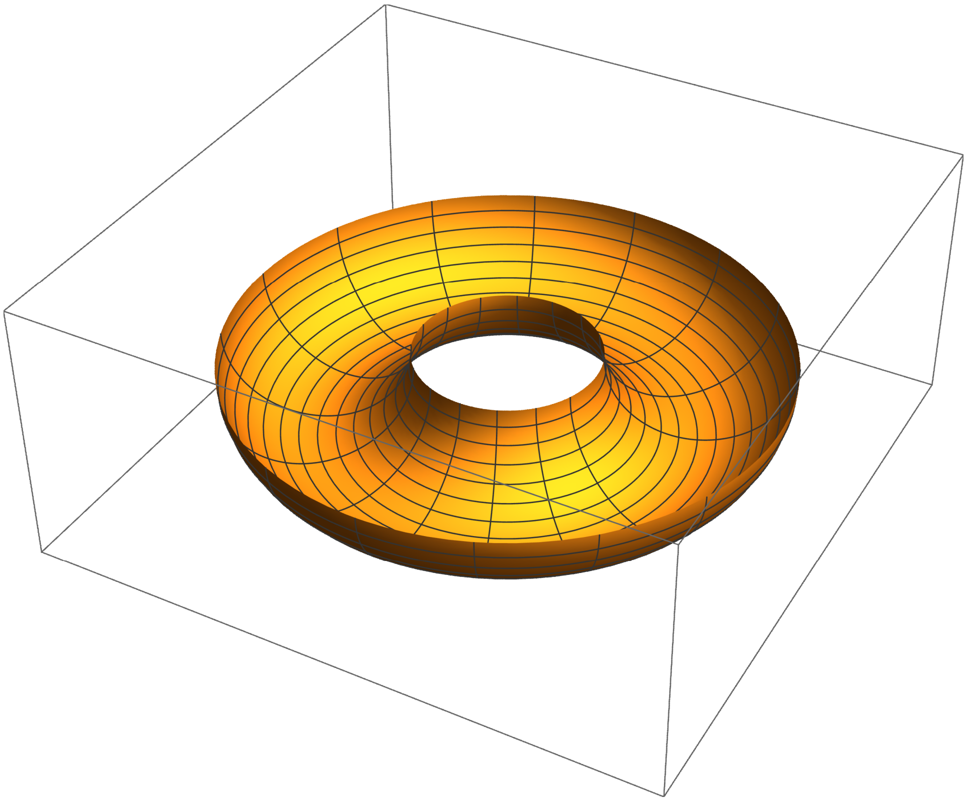
Out[71]=



Explore the role of the variables θ and ϕ:

In[72]:=
```
ParametricPlot3D[{Cos[θ] (2 + Cos[ϕ]), Sin[θ] (2 + Cos[ϕ]), Sin[ϕ]},
 {θ, 0, 2 π}, {ϕ, π, 2 π}, PlotPoints → {91, 61},
 PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
 Axes → False, BoxRatios → Automatic]
```

Out[72]=

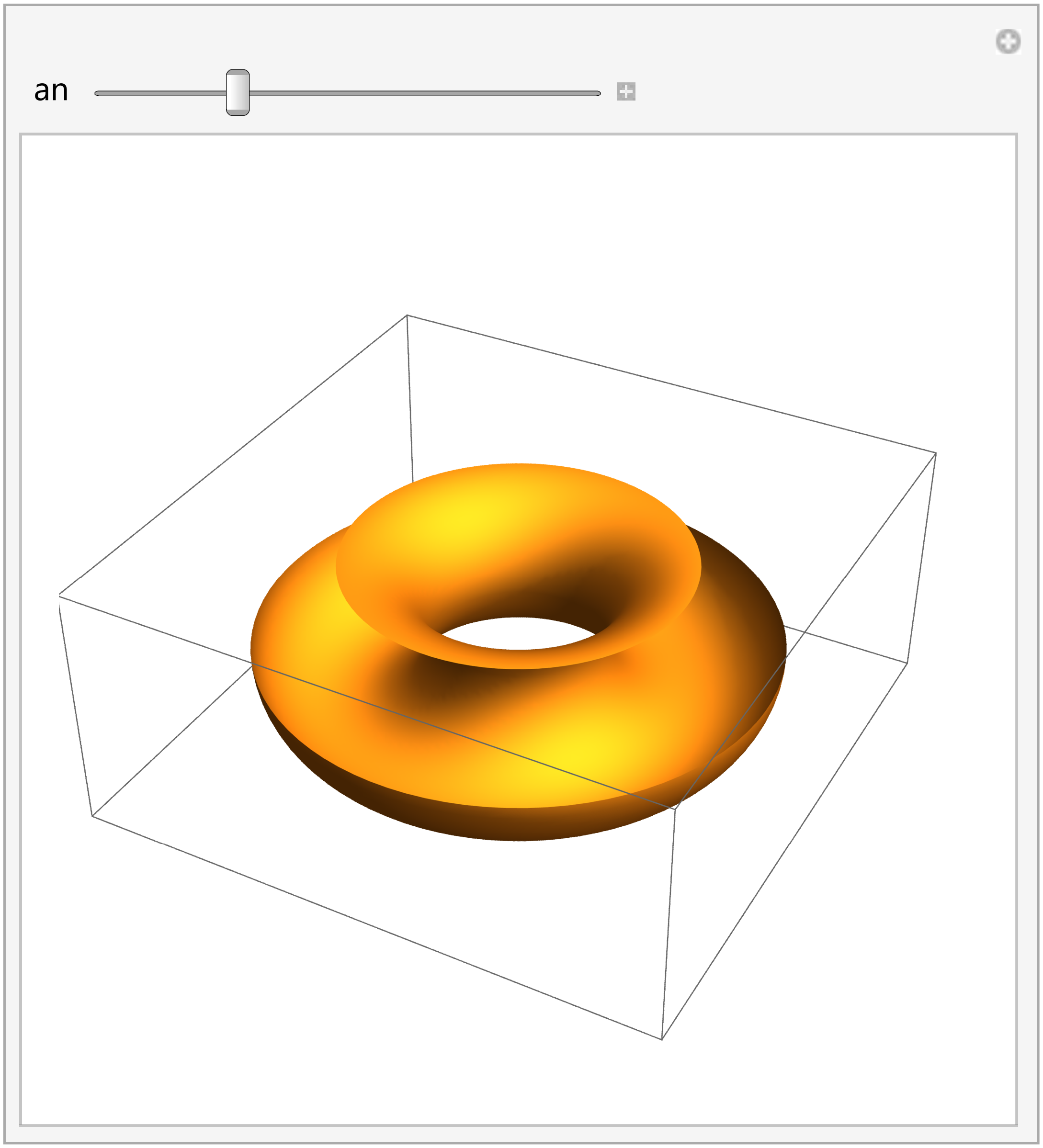


With Manipulate[], the role of ϕ:

In[73]:=
```
Clear[an];

Manipulate[
 ParametricPlot3D[{Cos[θ] (2 + Cos[ϕ]), Sin[θ] (2 + Cos[ϕ]), Sin[ϕ]},
  {θ, 0, 2 π}, {ϕ, an, 2 π}, PlotPoints → {91, 61}, Mesh → False,
  PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
  Axes → False, BoxRatios → Automatic],
 {{an, Pi / 2}, 0, 2 Pi - Pi/12, Pi/12, ControlPlacement → Top}]
```

Out[74]=

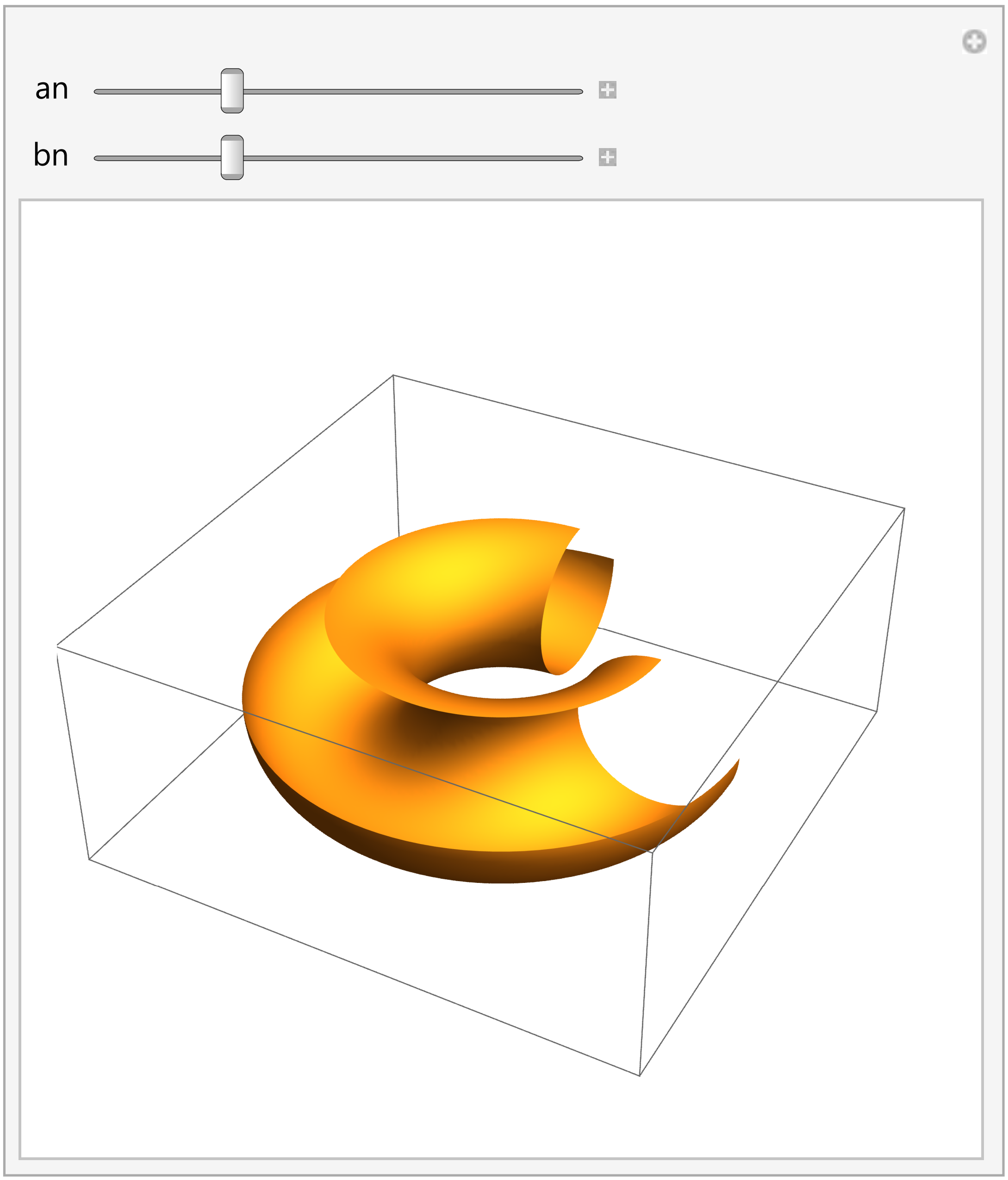


With Manipulate[], the role of both θ and ϕ:

In[75]:=
```
Clear[an, bn];

Manipulate[
 ParametricPlot3D[{Cos[θ] (2 + Cos[ϕ]), Sin[θ] (2 + Cos[ϕ]), Sin[ϕ]},
  {θ, bn, 2 π}, {ϕ, an, 2 π}, PlotPoints → {91, 61}, Mesh → False,
  PlotRange → {{-3.5`, 3.5`}, {-3.5`, 3.5`}, {-1.5`, 1.5`}},
  Axes → False, BoxRatios → Automatic],
 {{an, Pi / 2}, 0, 2 Pi - Pi/12, Pi/12, ControlPlacement → Top},
 {{bn, Pi / 2}, 0, 2 Pi - Pi/12, Pi/12, ControlPlacement → Top}]
```

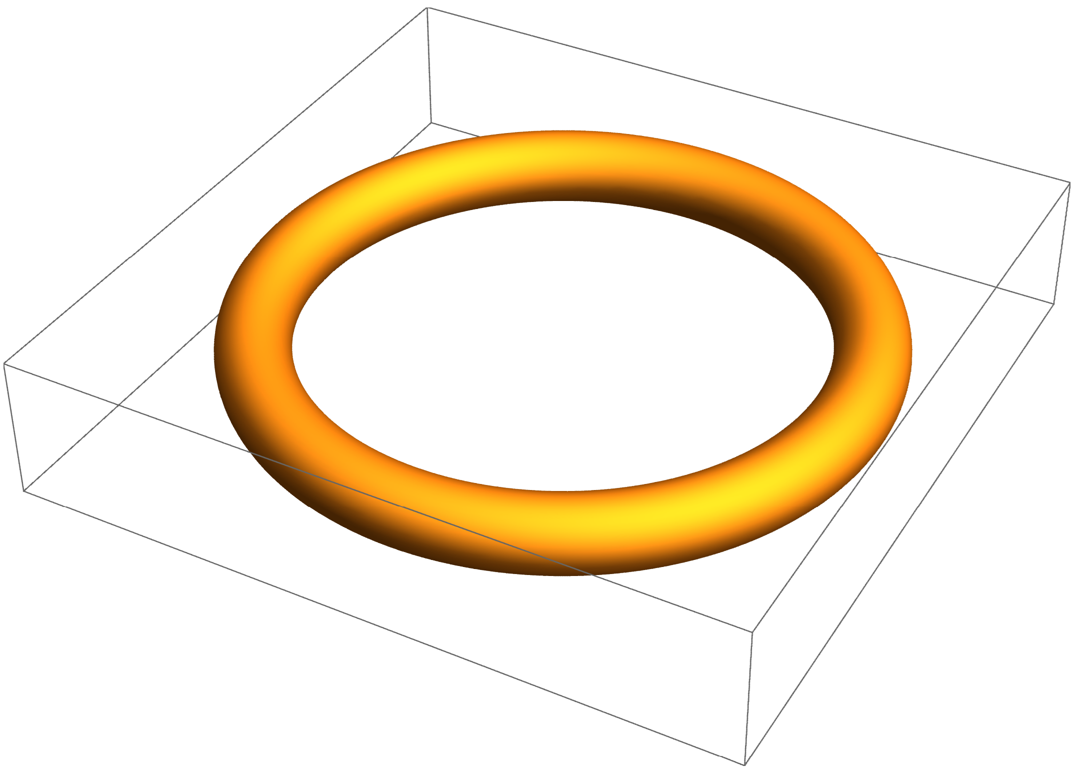
Out[76]=



There are several ways how to give a torus some life; we can make it bigger and thicker or thinner; make it a function

In[77]:=
```
Clear[Ra, ra]; Ra = 4;
ra = 0.5;
ParametricPlot3D[{Cos[θ] (Ra + ra Cos[ϕ]), Sin[θ] (Ra + ra Cos[ϕ]),
  ra Sin[ϕ]}, {θ, 0, 2 π}, {ϕ, 0, 2 π}, PlotPoints → {161, 51},
 Mesh → False,
 PlotRange → {{-Ra - ra - 0.5`, Ra + ra + 0.5`}, {-Ra - ra - 0.5`, Ra + ra + 0.5`},
   {-ra - 0.5`, ra + 0.5`}}, Axes → False, BoxRatios → Automatic,
 ImageSize → 400]
```

Out[77]=



Or, we can make torus into a helix:

In[78]:=
```
Clear[Ra, ra, h]; Ra = 4;
ra = 1;
h = 0.5`;
ParametricPlot3D[{Cos[θ] (Ra + ra Cos[ϕ]), Sin[θ] (Ra + ra Cos[ϕ]),
   ra Sin[ϕ] + θ/(2 h)}, {θ, 0, 8 π}, {ϕ, 0, 2 π}, PlotPoints → {261, 51},
 Mesh → False,
 PlotRange → {{-Ra - ra - 0.5`, Ra + ra + 0.5`}, {-Ra - ra - 0.5`, Ra + ra + 0.5`},
   {-ra - 0.5`, ra + 0.5` + 8 π}}, Axes → False, BoxRatios → {1, 1, 2},
 ImageSize → 300]
```

Out[78]=