In[1]:=
```
NotebookDirectory[]
```
Out[1]=
```
C:\Dropbox\Work\myweb\Courses\Math_pages\Math_225\
```

You can evaluate the entire notebook by using the keyboard shortcut Alt+v o, or the menu item Evaluation→Evaluate Notebook.

Starting from a familiar curve

- ***Mathematica* comments**

In the next subsubsection there is a simple picture in which I present only one point. This is to demonstrate how to plot geometric objects in *Mathematica*. For that we use Graphics[] command. One can get help on *Mathematica* commands by placing ? before the command name.

In[2]:=
```
? Graphics
```

Graphics[*primitives*, *options*] represents a two-dimensional graphical image. ≫

In the command below there is only one primitive:

In[3]:=
```
{PointSize[0.02], Blue, Point[{1, 1}]}
```
Out[3]=
```
{PointSize[0.02], RGBColor[0, 0, 1], Point[{1, 1}]}
```

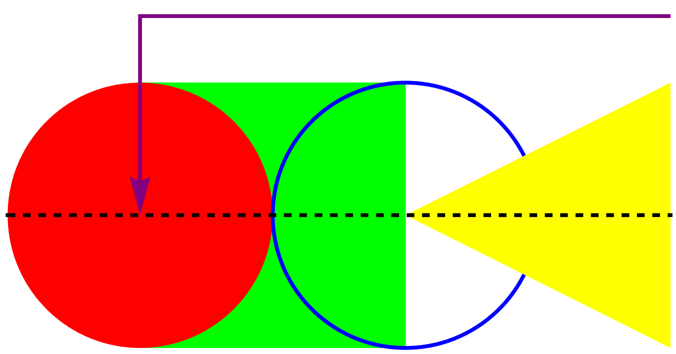
and several options, the first option being

In[4]:=
```
Frame → True
```
Out[4]=
```
Frame → True
```

The example given in *Mathematica* help is

In[5]:=
```
Graphics[{Thick, Green, Rectangle[{0, -1}, {2, 1}], Red, Disk[], Blue, Circle[{2, 0}],
  Yellow, Polygon[{{2, 0}, {4, 1}, {4, -1}}], Purple, Arrowheads[Large],
  Arrow[{{4, 3 / 2}, {0, 3 / 2}, {0, 0}}], Black, Dashed, Line[{{-1, 0}, {4, 0}}]}]
```

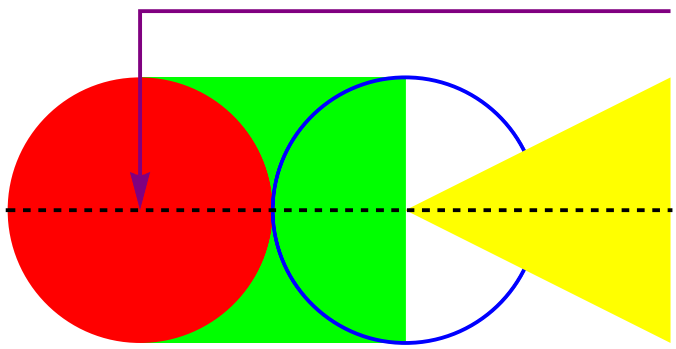
Out[5]=



This graphics command has six primitives and no options. I don't like how they write this command. In my opinion it is much nicer if we put each primitive in a separate list and all primitives we put in one list. Below is a nicer way of writing the above example

In[6]:=
```
Graphics[{ (* the list of primitives starts here *)
  {Thick, Green, Rectangle[{0, -1}, {2, 1}]}, (* the first primitive  *)
  {Red, Disk[]}, (* the second primitive  *)
  {Thick, Blue, Circle[{2, 0}]}, (* the third primitive  *)
  {Yellow, Polygon[{{2, 0}, {4, 1}, {4, -1}}]}, (* the fourth primitive  *)
  {Thick, Purple, Arrowheads[Large], Arrow[{{4, 3 / 2}, {0, 3 / 2}, {0, 0}}]},
  (* the fifth primitive  *)
  {Thick, Black, Dashed, Line[{{-1, 0}, {4, 0}}]} (* the sixth primitive  *)
 } (* the list of primitives ends here *)
]
```

Out[6]=



The only disadvantage is that we have to repeat the graphics directive Thick three times.

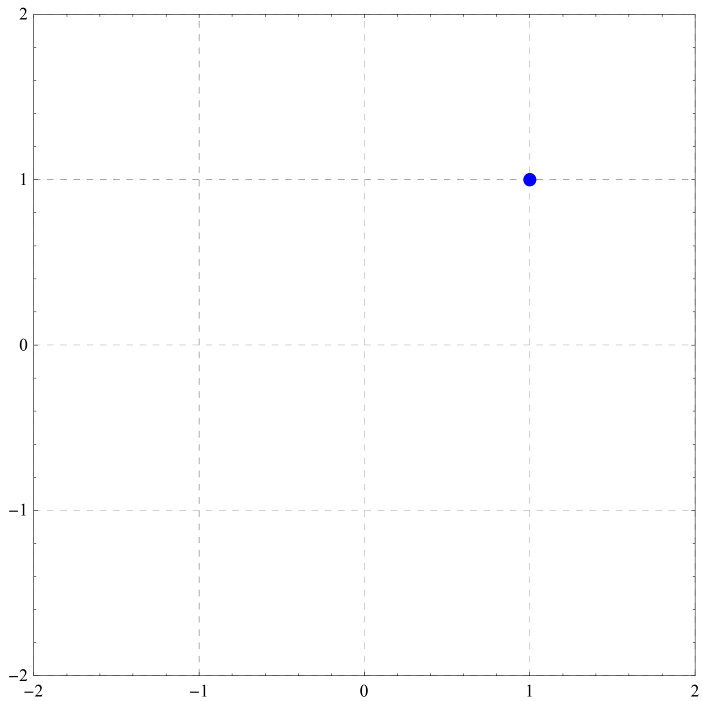
You can experiment by adding options to the above command.

- **Plotting points**

This is how to plot one point.

In[7]:=
```
Graphics[ (* Graphics[] command starts here *)
 { (* the list of primitives starts here  *)
  {PointSize[0.02], Blue, Point[{1, 1}]}
 }, (* the list of primitives ends here, the options follow  *)
 Frame → True, (* this option puts a frame around the graph *)
 PlotRange → {{-2, 2}, {-2, 2}}, (* this option determine the range of the plot *)
 AspectRatio → Automatic, (* horzontal unit = vertical unit *)
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 (* this option draws the grid lines *)
] (* Graphics[] command ends here *)
```

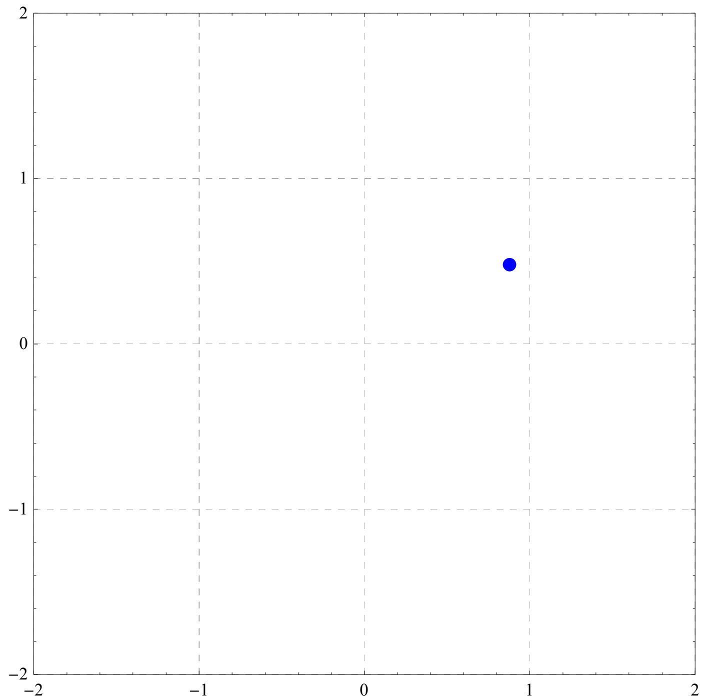
Out[7]=



Next I want to show a family of points. I do it in several steps. First I introduce a variable, t and I give this variable t a specific value 0.5. Then I plot one point with coordinates {Cos[t],Sin[t]}.

In[8]:=
```
t = .5;
Graphics[{
  {PointSize[0.02], Blue, Point[{Cos[t], Sin[t]}]}
 },
 Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

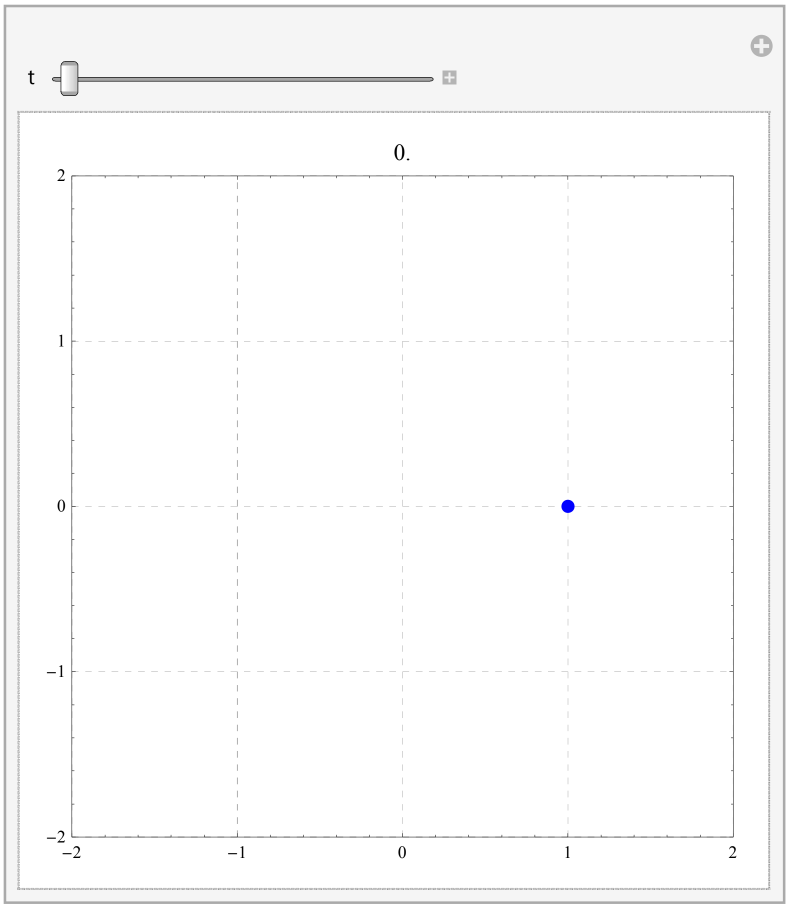
Out[9]=



Next I use command Manipulate[] to show many points with coordinates {Cos[t],Sin[t]}, as ta varies. Notice that the Graphics[] command from the previous cell is “wrapped” into Manipulate and the variable t is given range from 0 to 2π. To emphasize the change in t I show the value of t as PlotLabel.

In[10]:=
```
Clear[t];
Manipulate[ (* Manipulate[] starts here *)
 Graphics[{
   {PointSize[0.02], Blue, Point[{Cos[t], Sin[t]}]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], (* Graphics[] ends here *)
 {t, 0., 2 Pi} (* this tells Manipulate to use t in this range *)
] (* Manipulate[] ends here *)
```

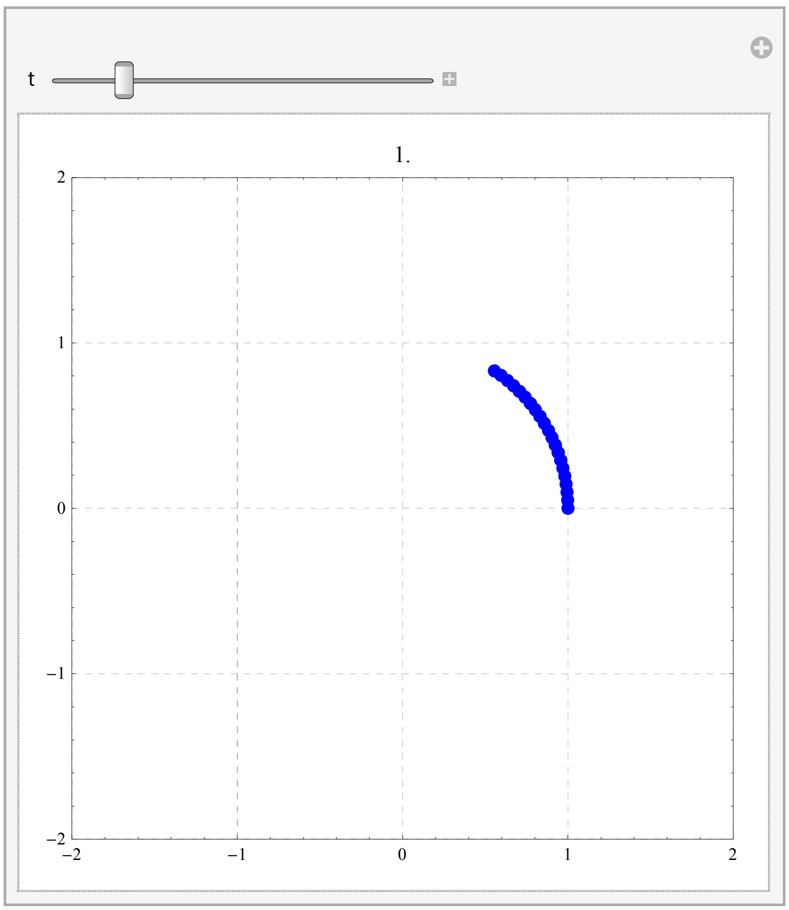
Out[11]=



In the next command I tell *Mathematica* to remember the points that have been plotted previously, so that we can see which curve is being plotted.

In[12]:=
```
Clear[t];
Manipulate[
 Graphics[{
   {PointSize[0.02], Blue, Table[Point[{Cos[v], Sin[v]}], {v, 0, t, Pi/64}]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, 1}, 0, 2 Pi, Pi/64}]
```

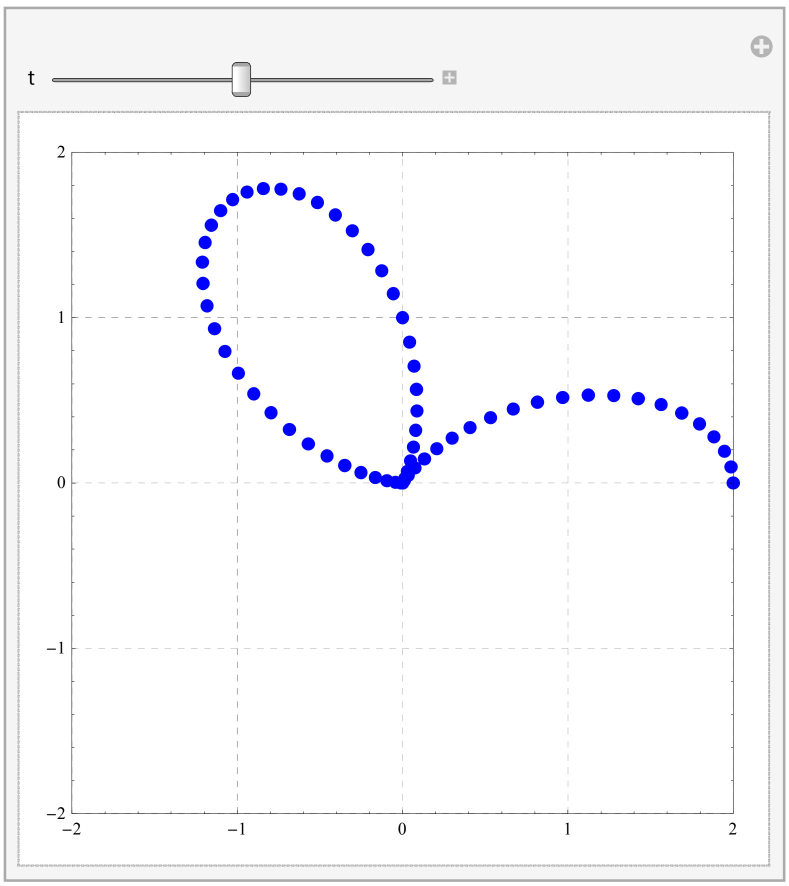
Out[13]=



In the next several plots I show variations on a unit circle. The only thing that I change is that I make the radius to be a function of t. I call that function fr[t]

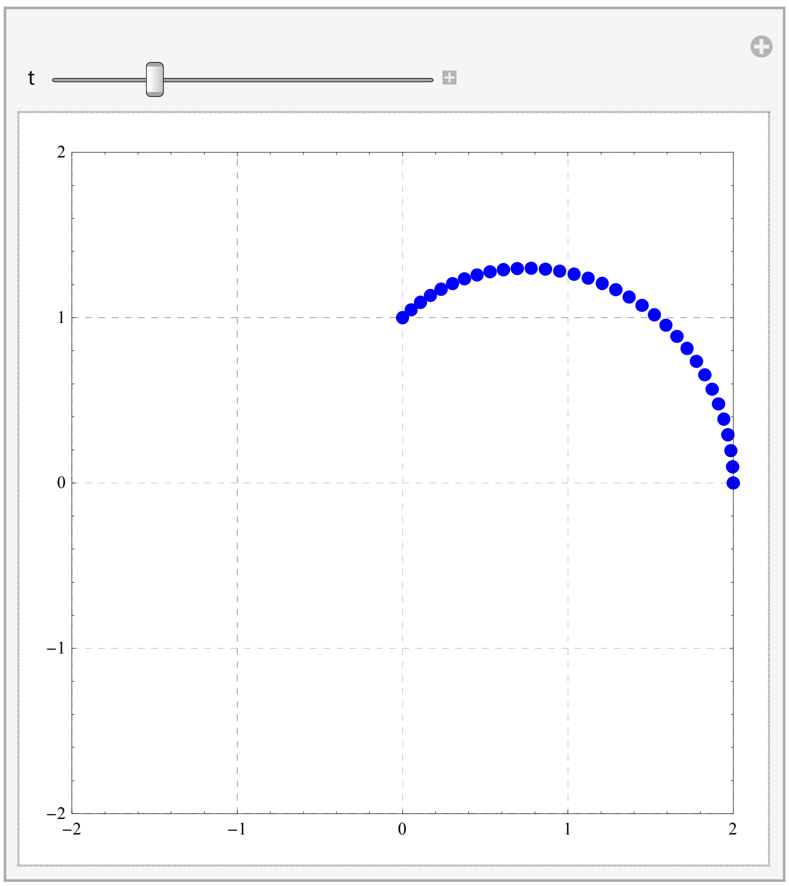
In[14]:=
```
Clear[t];

fr1[t_] := 1 + Cos[3 t];

Manipulate[
 Graphics[{
   {PointSize[0.02], Blue, Table[Point[fr1[v] {Cos[v], Sin[v]}], {v, 0, t, Pi/64}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi}, 0, 2 Pi, Pi/64}]
```

Out[16]=



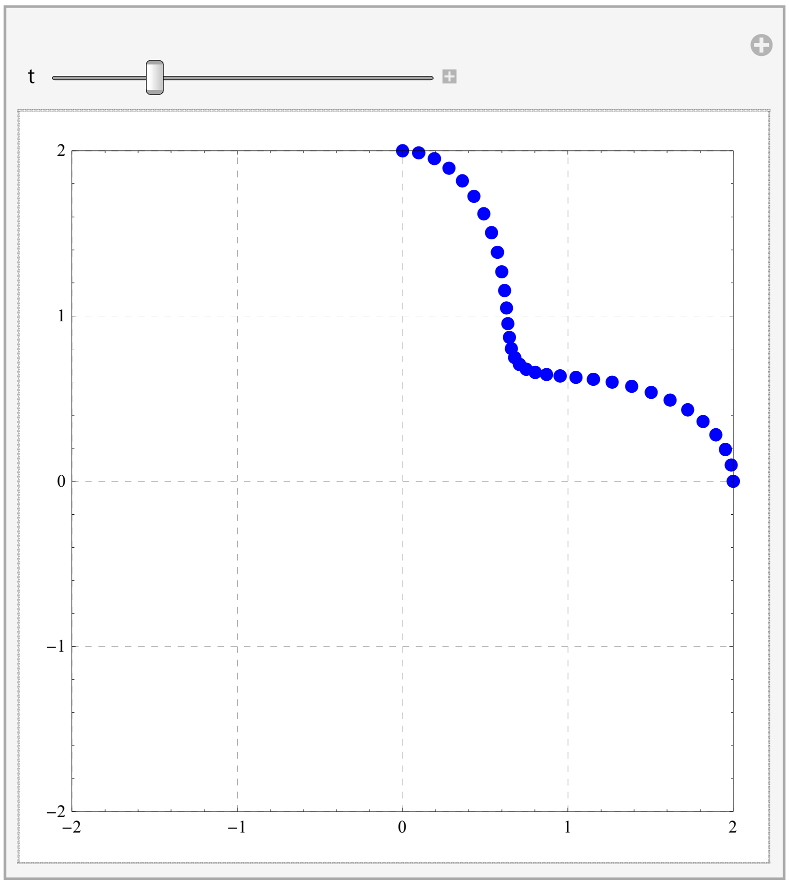
In[17]:=
```
Clear[t];

fr2[t_] := 1 + Cos[t];

Manipulate[
 Graphics[{
   {PointSize[0.02], Blue, Table[Point[fr2[v] {Cos[v], Sin[v]}], {v, 0, t, Pi/64}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi / 2}, 0, 2 Pi, Pi/64}]
```

Out[19]=



In[20]:=
```
Clear[t];

fr3[t_] := 1 + Cos[2 t]^2 ;

Manipulate[
 Graphics[{
   {PointSize[0.02], Blue, Table[Point[fr3[v] {Cos[v], Sin[v]}], {v, 0, t, Pi/64}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi / 2}, 0, 2 Pi, Pi/64}]
```

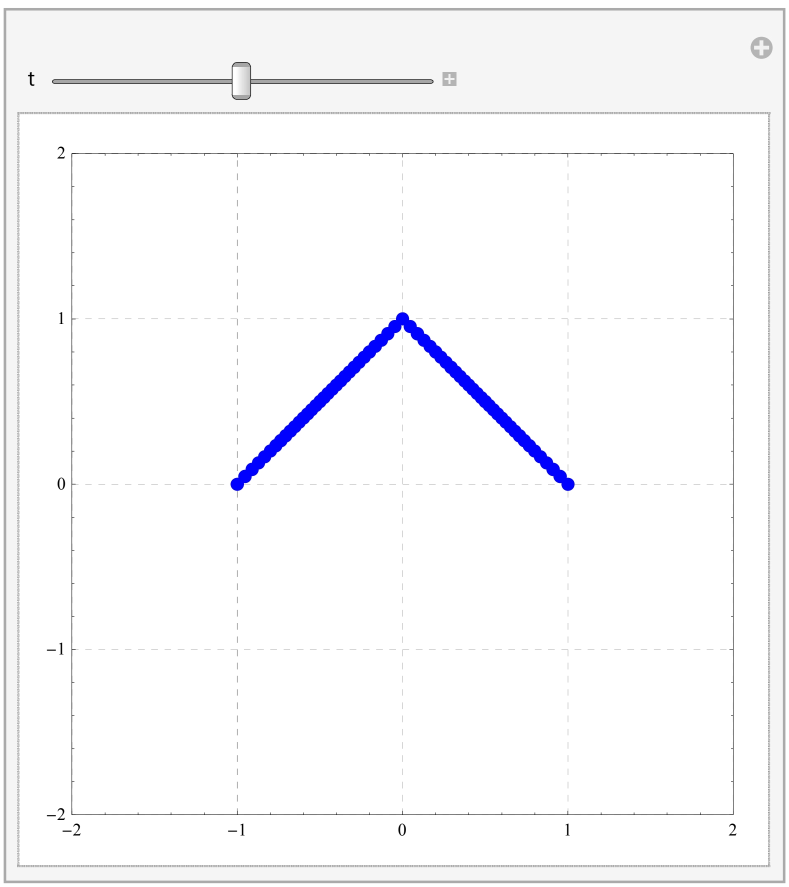
Out[22]=



The last “radius” function is more complicated. As a reward, the resulting graph is any regular n-gon. Just change 4 to any of 3,4,5,6,7, ... in fr4[v,4] in the Graphics[] command below.

In[23]:=
```
Clear[t];

fr4[t_, n_] := Cos[Pi/n] / Cos[Mod[t, 2π/n] - Pi/n] ;

Manipulate[
 Graphics[{
   {PointSize[0.02], Blue, Table[Point[fr4[v, 4] {Cos[v], Sin[v]}], {v, 0, t, Pi/64}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi}, 0, 2 Pi, Pi/64}]
```

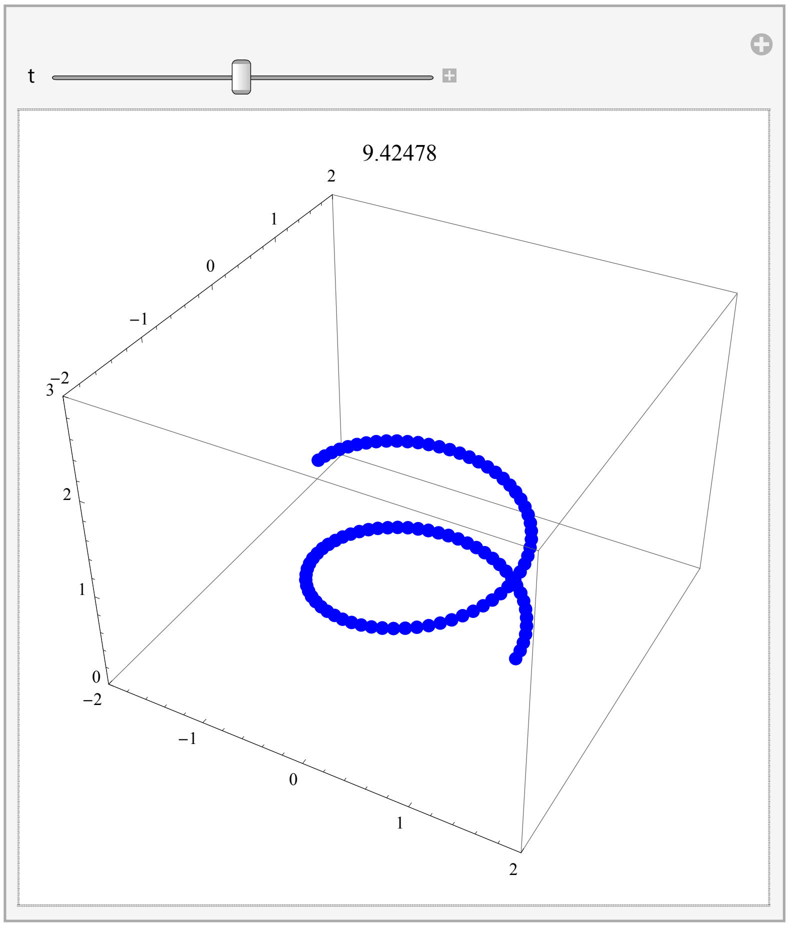
Out[25]=



In the next few examples we demonstrate curves in three-space. We start with a helix above the unit circle and which climes one unit for each complete unit circle.

In[26]:=
```
Clear[t];
Manipulate[
 Graphics3D[{
   {PointSize[0.02], Blue, Table[Point[{Cos[v], Sin[v], v/(2 Pi)}], {v, 0, t, Pi/32}]}
  }, PlotLabel → N[t],
  PlotRange → {{-2, 2}, {-2, 2}, {0, 3}}, Axes → True
 ], {{t, 3 Pi}, 0, 6 Pi, Pi/32}]
```

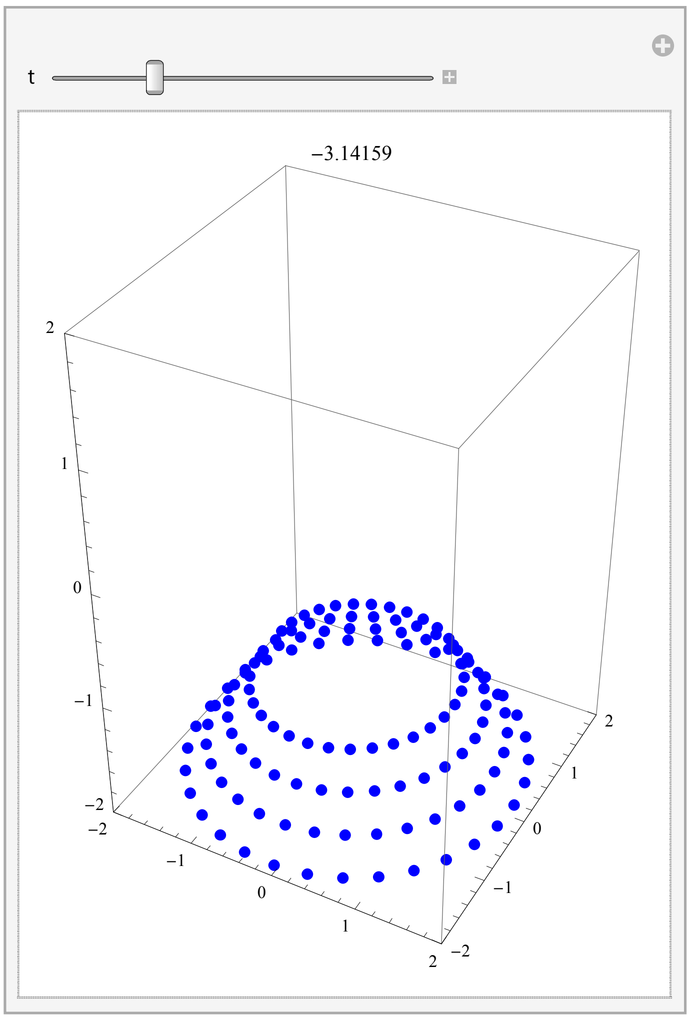
Out[27]=



The next example shows a helix like curve that climes on a cone

In[28]:=
```
Clear[t, fx, fy, fz];

fx[t_] := t/Pi Cos[4 t]; fy[t_] := t/Pi Sin[4 t]; fz[t_] := t/Pi;

Manipulate[
 Graphics3D[{
   {PointSize[0.02], Blue, Table[Point[{fx[v], fy[v], fz[v]}], {v, -2 Pi, t, Pi/128}]}
  }, PlotLabel → N[t],
  PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}}, Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}}, BoxRatios → {1, 1, 1.5}
 ], {{t, -Pi}, -2 Pi, 2 Pi, Pi/128}]
```

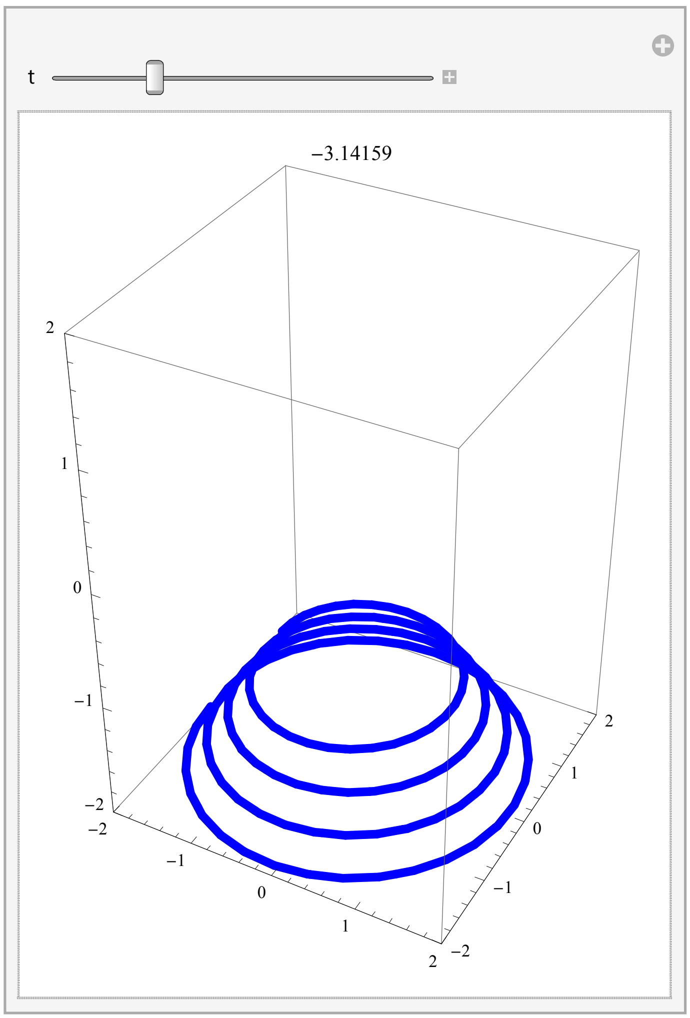
Out[30]=



The next plot is the same helix shown as line, not just a collection of points.

In[31]:=
```
Clear[t, fx, fy, fz];

fx[t_] := t/Pi Cos[4 t]; fy[t_] := t/Pi Sin[4 t]; fz[t_] := t/Pi;

Manipulate[
 Graphics3D[{
   {Thickness[0.015], Blue, Line[Table[{fx[v], fy[v], fz[v]}, {v, -2 Pi, t, Pi/128}]]}
  }, PlotLabel → N[t],
  PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}}, Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}}, BoxRatios → {1, 1, 1.5}
 ], {{t, -Pi}, -2 Pi, 2 Pi, Pi/128}]
```

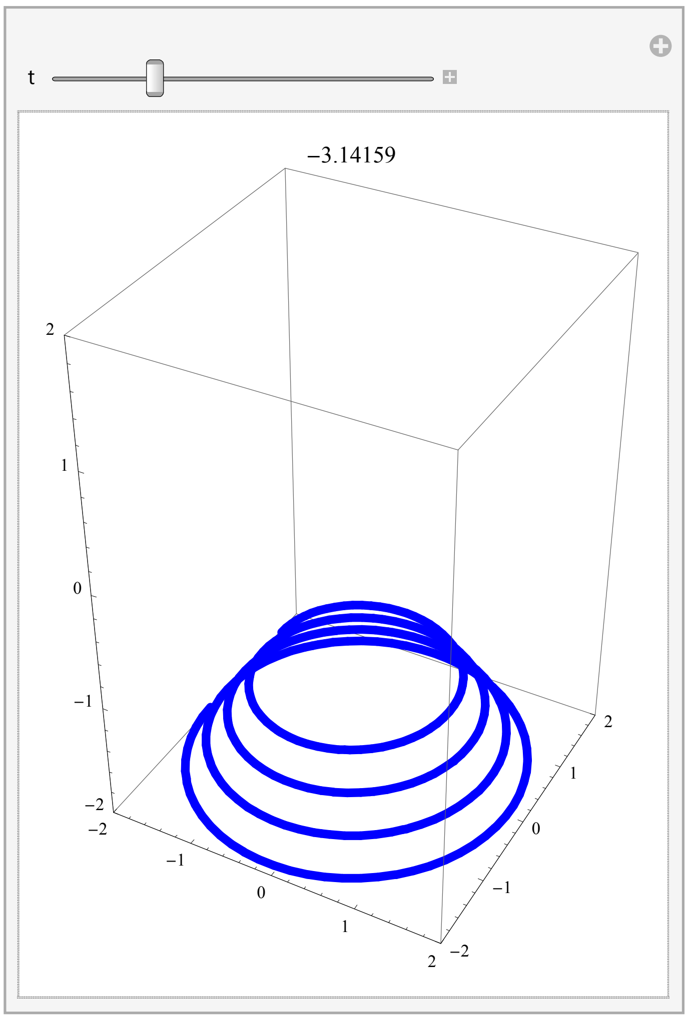
Out[33]=



The next helix is on the same cone, but winds more often then the previous one.

In[34]:=
```
Clear[t, fx, fy, fz];

fx[t_] := t/Pi Cos[8 t]; fy[t_] := t/Pi Sin[8 t]; fz[t_] := t/Pi;

Manipulate[

 Graphics3D[{

   {Thickness[0.015], Blue, Line[Table[{fx[v], fy[v], fz[v]}, {v, -2 Pi, t, Pi/(2 × 128)}]]}

  }, PlotLabel → N[t],
  PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}}, Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}}, BoxRatios → {1, 1, 1.5}
 ], {{t, -Pi}, -2 Pi, 2 Pi, Pi/(2 × 128)}]
```

Out[36]=



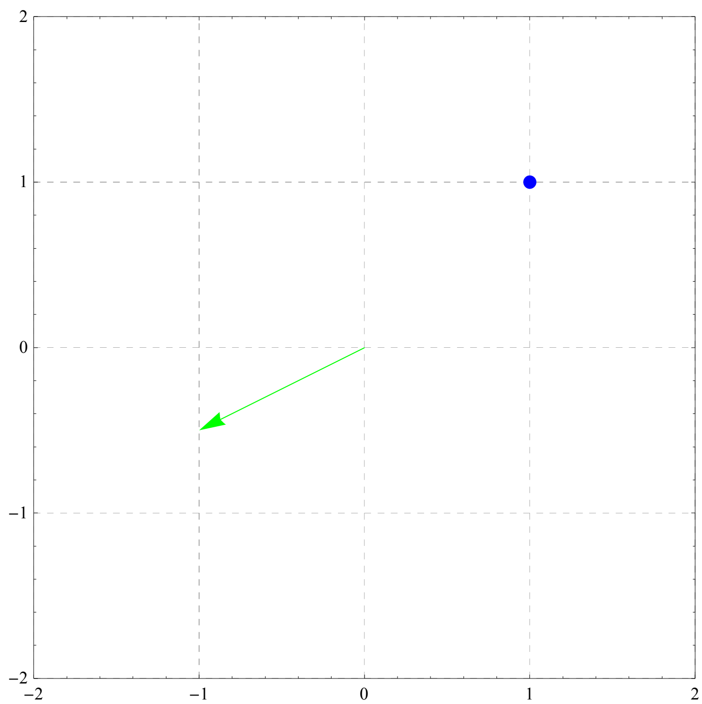
Lines

■ Point and a vector

Given a point say P and a direction given by a vector, say $\vec{v}$, how do does a point walk starting from P in the direction specified by the vector $\vec{v}$?

In[37]:=
```
pP = {1, 1}; vv = {-1, -1 / 2};

Graphics[{
   {PointSize[0.02], Blue, Point[pP]},
   {Green, Arrow[{{0, 0}, vv}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

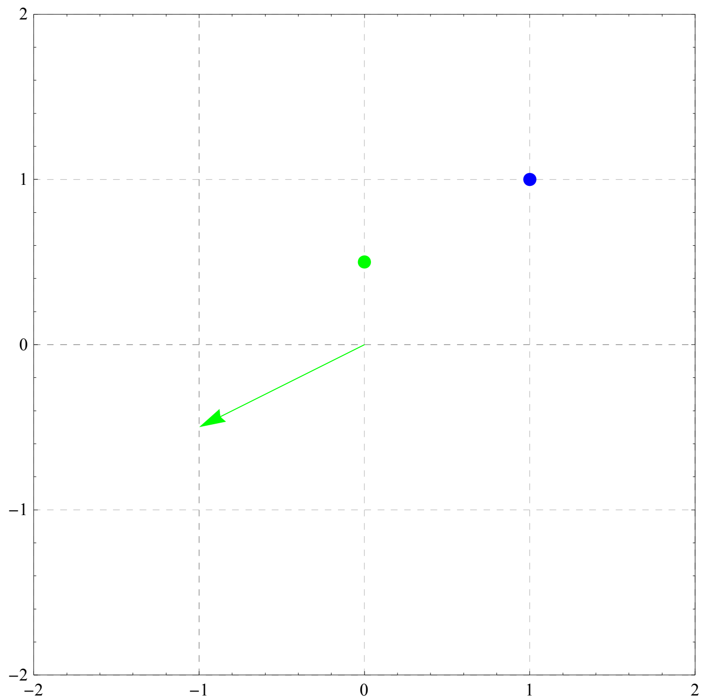
Out[38]=



After one second, the point will be at the green point whose position vector is $\overrightarrow{OP} + \vec{v}$

```
In[39]:= pP = {1, 1}; vv = {-1, -1 / 2};

t = 1;

Graphics[{
   {PointSize[0.02], Green, Point[pP + t vv]},
   {PointSize[0.02], Blue, Point[pP]},
   {Green, Arrow[{{0, 0}, vv}]}
  },
 Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

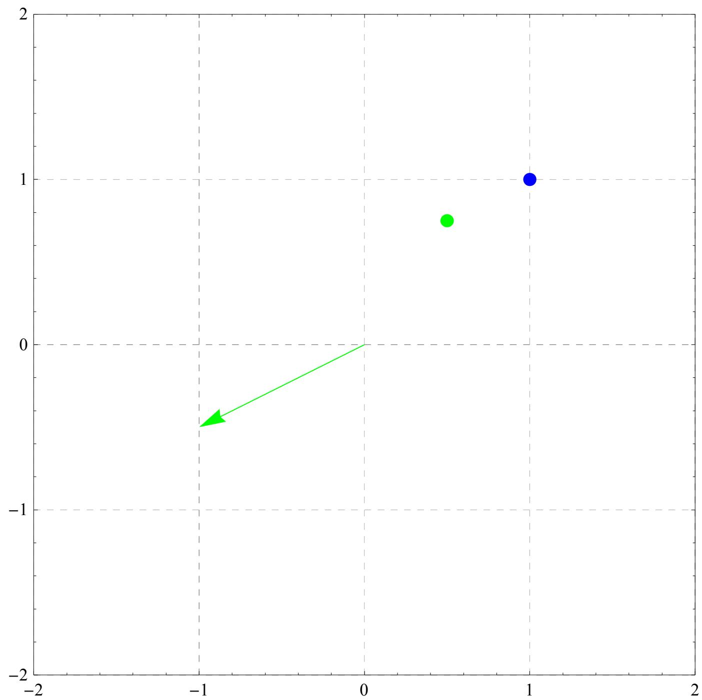
Out[41]=



After 1/2 second, the point will be at the green point whose position vector is $\overrightarrow{OP} + \frac{1}{2}\vec{v}$

In[42]:=
```
pP = {1, 1}; vv = {-1, -1 / 2};

t = 1 / 2;

Graphics[{
  {PointSize[0.02], Green, Point[pP + t vv]},
  {PointSize[0.02], Blue, Point[pP]},
  {Green, Arrow[{{0, 0}, vv}]}
 },
 Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

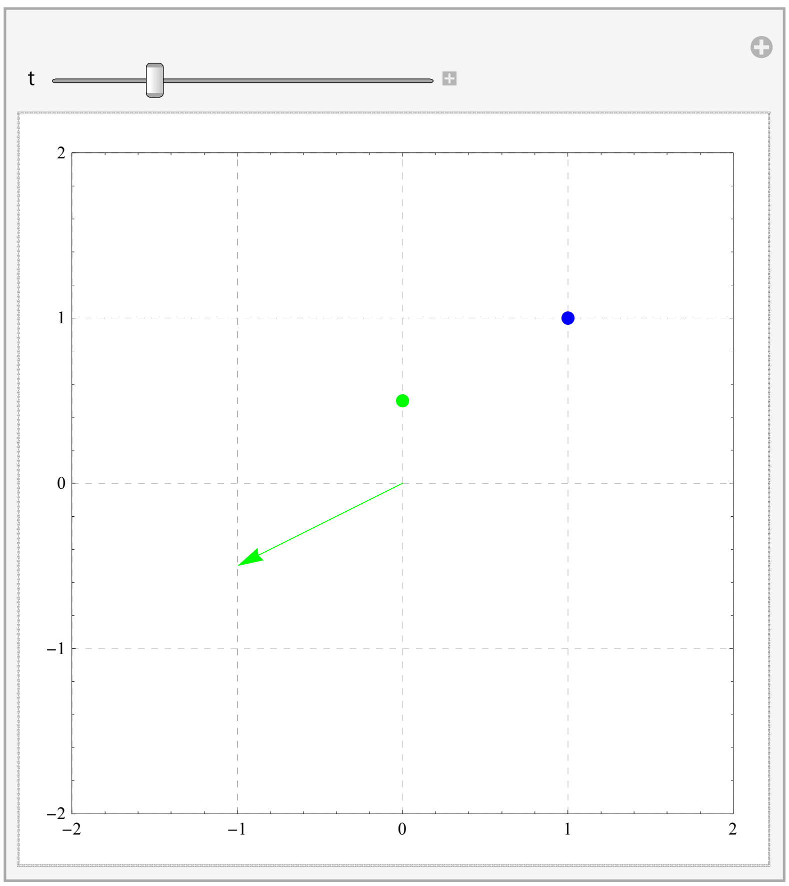
Out[44]=



Now we are ready to illustrate the motion of the point with the Manipulation[] command

In[45]:=
```
pP = {1, 1}; vv = {-1, -1 / 2};

Clear[t];
Manipulate[
 Graphics[{
   {PointSize[0.02], Green, Point[pP + t vv]},
   {PointSize[0.02], Blue, Point[pP]},
   {Green, Arrow[{{0, 0}, vv}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ],
 {{t,
   1},
  0,
  4}]
```

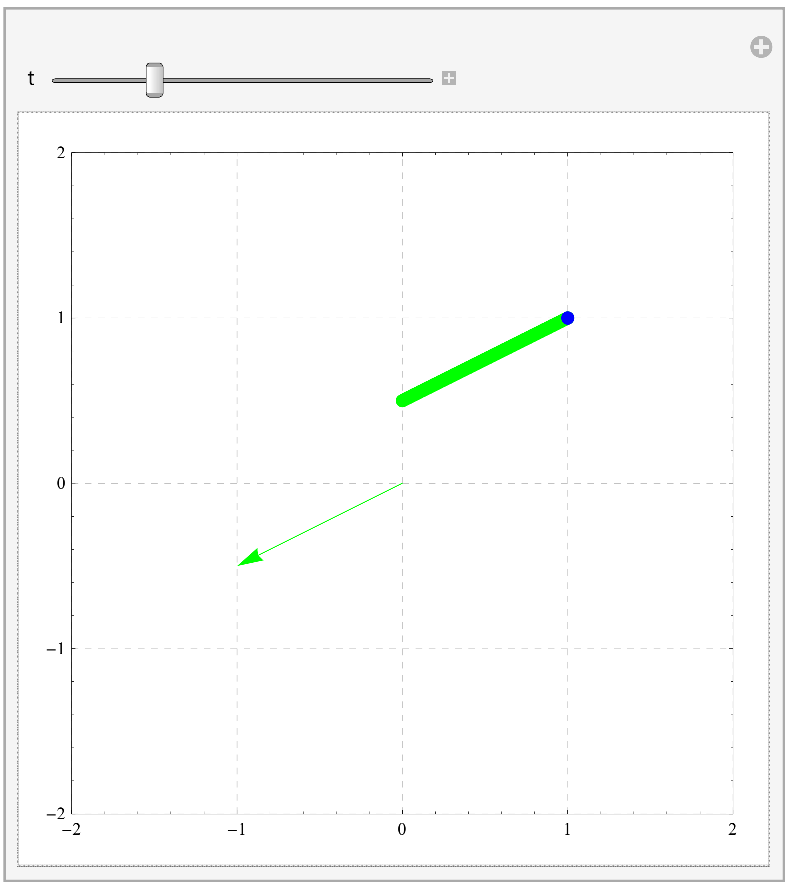
Out[47]=



The same illustration with point’s positions remembered.

In[48]:=
```
pP = {1, 1}; vv = {-1, -1 / 2};

Clear[t];
Manipulate[
 Graphics[{
   {PointSize[0.02], Green, Table[Point[pP + s vv], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP]},
   {Green, Arrow[{{0, 0}, vv}]}
  },
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ],
 {{t,
   1},
  0,
  4}]
```

Out[50]=

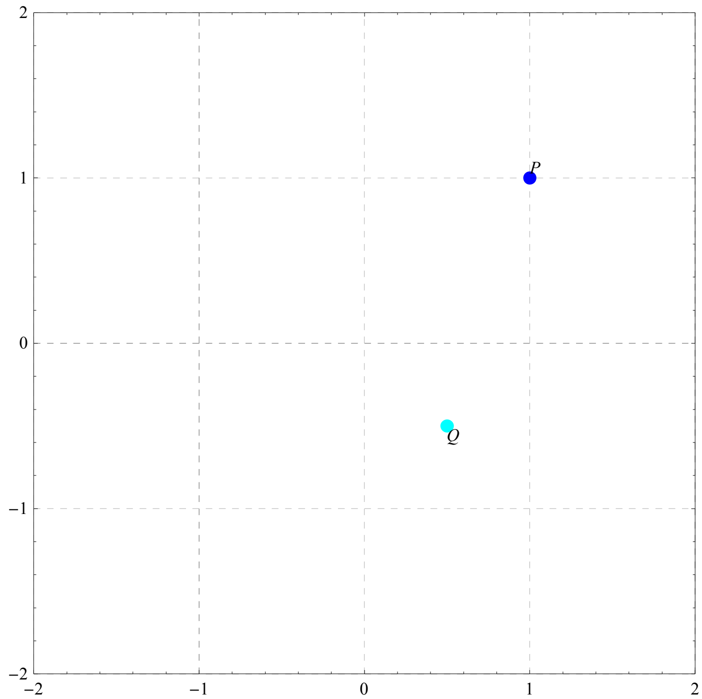


■ Two points

In this subsection I illustrate how to find the line determined by two points.

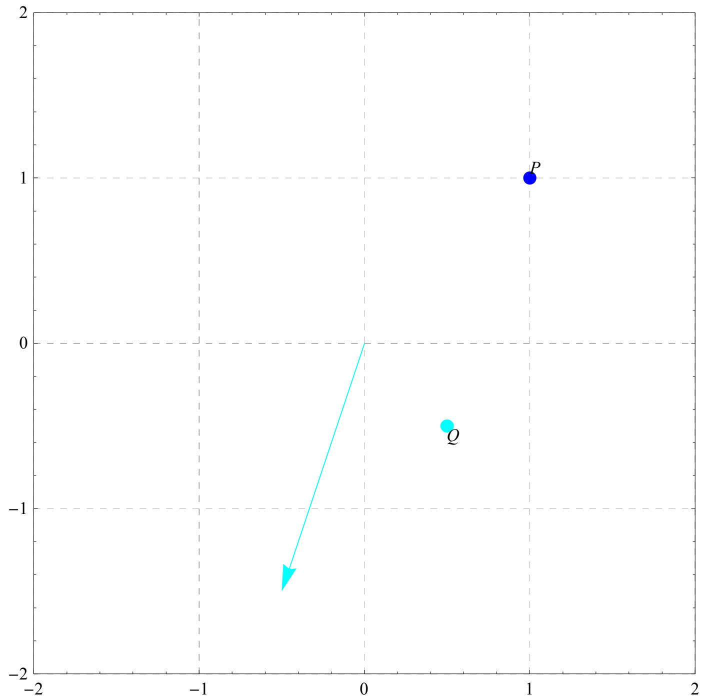
In[51]:=
```
pP = {1, 1}; pQ = {1 / 2, -1 / 2};
Graphics[{
  {PointSize[0.02], Blue, Point[pP]},
  {PointSize[0.02], Cyan, Point[pQ]},

  {Text[P, pP, {-1, -1}], Text[Q, pQ, {-1, 1}]}
 },
 Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

Out[52]=



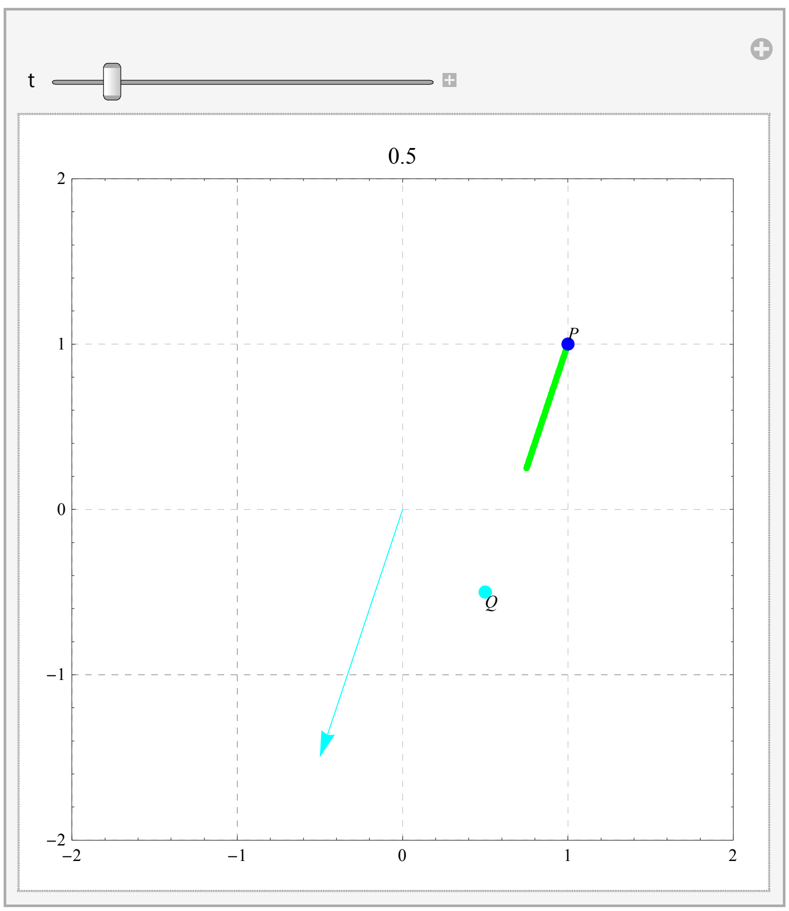
In[53]:=
```
pP = {1, 1}; pQ = {1 / 2, -1 / 2};
Graphics[{
  {PointSize[0.02], Blue, Point[pP]},
  {PointSize[0.02], Cyan, Point[pQ]},
  {Cyan, Arrow[{{0, 0}, pQ - pP}]},
  {Text[P, pP, {-1, -1}], Text[Q, pQ, {-1, 1}]}
 },
 Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

Out[54]=



In[55]:=
```
pP = {1, 1}; pQ = {1 / 2, -1 / 2};

Manipulate[
 Graphics[{
   {PointSize[0.01], Green, Table[Point[pP + s (pQ - pP)], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP]},
   {PointSize[0.02], Cyan, Point[pQ]},
   {Cyan, Arrow[{{0, 0}, pQ - pP}]},
   {Text[P, pP, {-1, -1}], Text[Q, pQ, {-1, 1}]}
  },
  PlotLabel → N[t],
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ] ,
 {{t, .5}, 0, 4}]
```

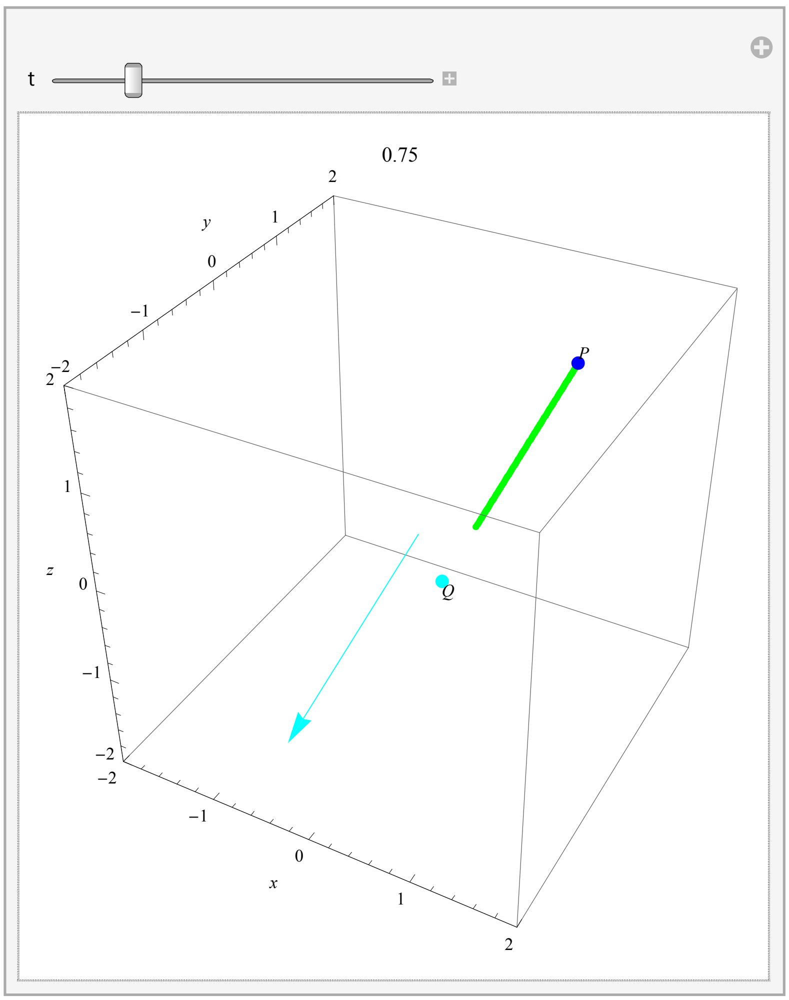
Out[56]=



The same logic applies in three dimensions:

In[57]:=
```
pP1 = {3 / 2, 1, 3 / 2}; pQ1 = {1 / 2, -3 / 2, -1 / 2};

Manipulate[
 Graphics3D[{
   {PointSize[0.01], Green, Table[Point[pP1 + s (pQ1 - pP1)], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP1]},
   {PointSize[0.02], Cyan, Point[pQ1]},
   {Cyan, Arrow[{{0, 0, 0}, pQ1 - pP1}]},
   {Text[P, pP1, {-1, -1}], Text[Q, pQ1, {-1, 1}]}
  },
  PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}},
  AxesLabel → {x, y, z}
 ],
 {{t, 3 / 4}, 0, 4}]
```

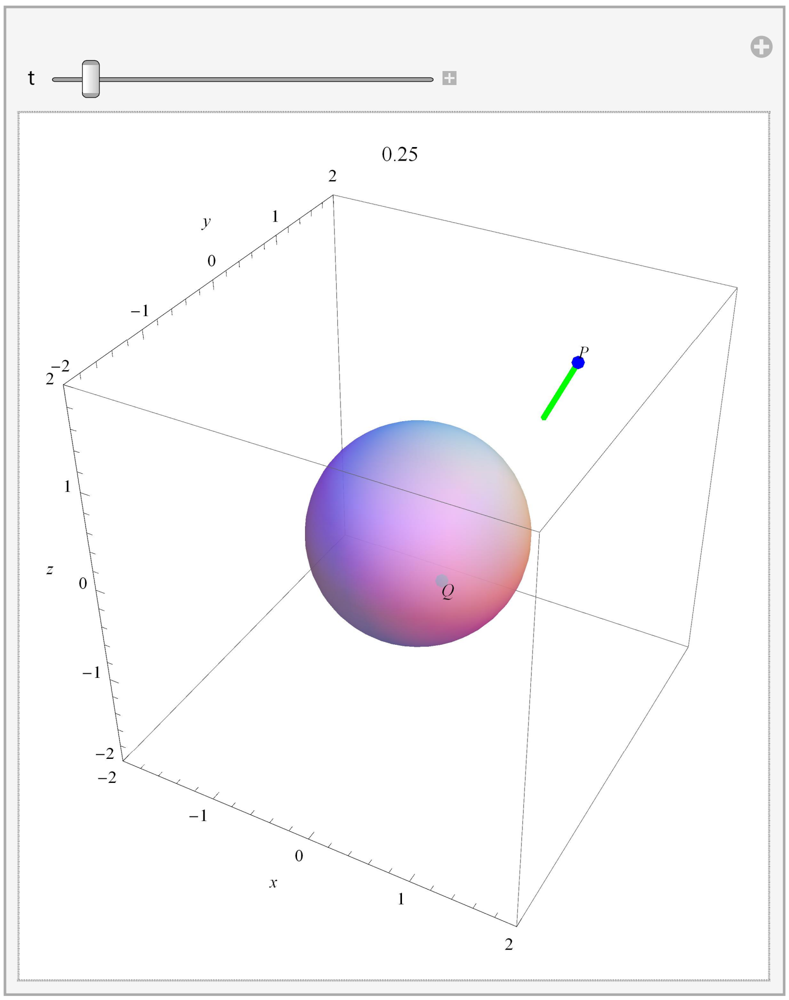
Out[58]=



■ Two points and the unit sphere

In[59]:=
```
pP1 = {3 / 2, 1, 3 / 2}; pQ1 = {1 / 4, -3 / 2, -3 / 2};

Manipulate[
 Graphics3D[{
   {PointSize[0.01], Green, Table[Point[pP1 + s (pQ1 - pP1)], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP1]},
   {PointSize[0.02], Cyan, Point[pQ1]},
   {Opacity[0.75], Sphere[{0, 0, 0}, 1]},
   {Text[P, pP1, {-1, -1}], Text[Q, pQ1, {-1, 1}]}
  },
  PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}},
  AxesLabel → {x, y, z}
 ],
 {{t, 1 / 4}, 0, 4}]
```

Out[60]=



An relevant question for the above graph would be: Does a person located at the point P sees a person located at the point Q? To answer this question we need to calculate whether the line joining P and Q intersects the unit sphere. I will do this in *Mathematica*.

In[61]:=
```
pP1 = {3 / 2, 1, 3 / 2}; pQ1 = {1 / 4, -3 / 2, -3 / 2};
```

The equation of the line joining these two points is

In[62]:=
```
pP1 + t (pQ1 - pP1)
```

Out[62]= $\left\{\frac{3}{2} - \frac{5\,t}{4},\ 1 - \frac{5\,t}{2},\ \frac{3}{2} - 3\,t\right\}$

Now we calculate if there are points on this line which are at the distance 1 from the origin

In[63]:= $\text{Solve}\left[\left(\frac{3}{2} - \frac{5\,t}{4}\right)^2 + \left(1 - \frac{5\,t}{2}\right)^2 + \left(\frac{3}{2} - 3\,t\right)^2 == 1,\ t\right]$

Out[63]= $\left\{\left\{t \to \frac{2}{269}\left(71 - \sqrt{199}\right)\right\},\ \left\{t \to \frac{2}{269}\left(71 + \sqrt{199}\right)\right\}\right\}$

Or, look for a numerical solution

In[64]:= $\text{NSolve}\left[\left(\frac{3}{2} - \frac{5\,t}{4}\right)^2 + \left(1 - \frac{5\,t}{2}\right)^2 + \left(\frac{3}{2} - 3\,t\right)^2 == 1,\ t\right]$

Out[64]= {{t → 0.422998}, {t → 0.632764}}

Yes, there are two points on the line joining P and Q which are on the unit sphere. Therefore a person located at the point P cannot see the person located at the point Q. This changes if we change the position of Q

In[65]:=
```
pP1 = {3 / 2, 1, 3 / 2}; pQ2 = {1 / 2, -3 / 2, -3 / 2};
```

The equation of the line joining these two points is

In[66]:=
```
pP1 + t (pQ2 - pP1)
```

Out[66]= $\left\{\frac{3}{2} - t,\ 1 - \frac{5\,t}{2},\ \frac{3}{2} - 3\,t\right\}$

Now we calculate if there are points on this line which are at the distance 1 from the origin

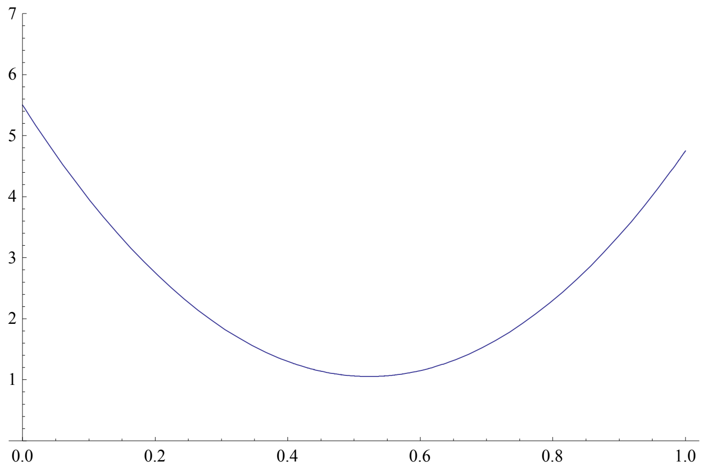
In[67]:= $\text{Solve}\left[\left(\frac{3}{2} - t\right)^2 + \left(1 - \frac{5\,t}{2}\right)^2 + \left(\frac{3}{2} - 3\,t\right)^2 == 1,\ t\right]$

Out[67]= $\left\{\left\{t \to \frac{1}{65}\left(34 - i\sqrt{14}\right)\right\},\ \left\{t \to \frac{1}{65}\left(34 + i\sqrt{14}\right)\right\}\right\}$

There are no real solutions. Therefore there are on points on the line joining P and this new Q which are on the unit sphere. Here we can calculate the closest point on this line to the unit sphere. First plot

In[68]:=
```
Plot[(3/2 - t)^2 + (1 - 5 t/2)^2 + (3/2 - 3 t)^2, {t, 0, 1}, PlotRange → {0, 7}]
```

Out[68]=



Now calculate derivative

In[69]:=
```
Simplify[(3/2 - t)^2 + (1 - 5 t/2)^2 + (3/2 - 3 t)^2]
```

Out[69]= $\frac{1}{4}\left(22 - 68\,t + 65\,t^2\right)$

In[70]:=
```
Solve[D[1/4 (22 - 68 t + 65 t^2), t] == 0, t]
```

Out[70]= $\left\{\left\{t \to \frac{34}{65}\right\}\right\}$

Thus, the closest point to the unit sphere is

In[71]:=
```
pP1 + 34/65 (pQ2 - pP1)
```

Out[71]= $\left\{\frac{127}{130}, -\frac{4}{13}, -\frac{9}{130}\right\}$

Its distance from the origin is

In[72]:= $\sqrt{\left(\left(\frac{127}{130}\right)^2 + \left(-\frac{4}{13}\right)^2 + \left(-\frac{9}{130}\right)^2\right)}$

Out[72]= $\sqrt{\frac{137}{130}}$

approximated by

In[73]:= $N\left[\sqrt{\frac{137}{130}}\right]$

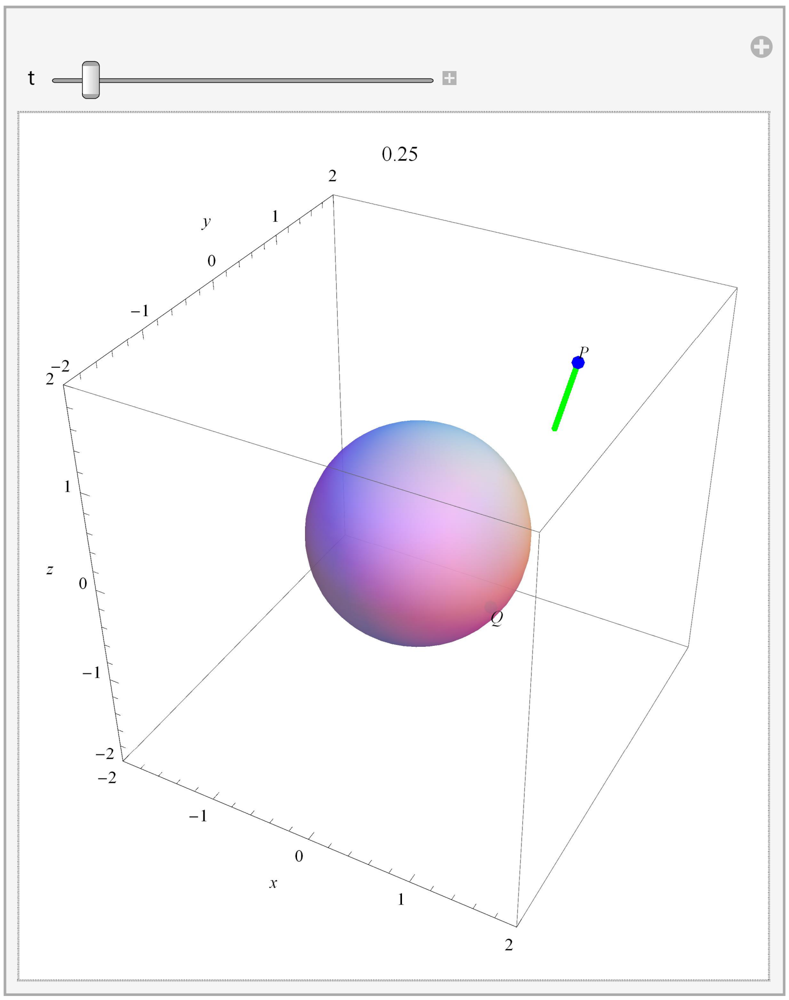
Out[73]= 1.02657

Thus this point is really close to the unit sphere.

Finally see it in three-space

In[74]:=
```
Manipulate[
 Graphics3D[{
   {PointSize[0.01], Green, Table[Point[pP1 + s (pQ2 - pP1)], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP1]},
   {PointSize[0.02], Cyan, Point[pQ2]},
   {Opacity[0.75], Sphere[{0, 0, 0}, 1]},
   {Text[P, pP1, {-1, -1}], Text[Q, pQ2, {-1, 1}]}
  },
  PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}},
  AxesLabel → {x, y, z}
 ],
 {{t, 1 / 4}, 0, 4}]
```

Out[74]=



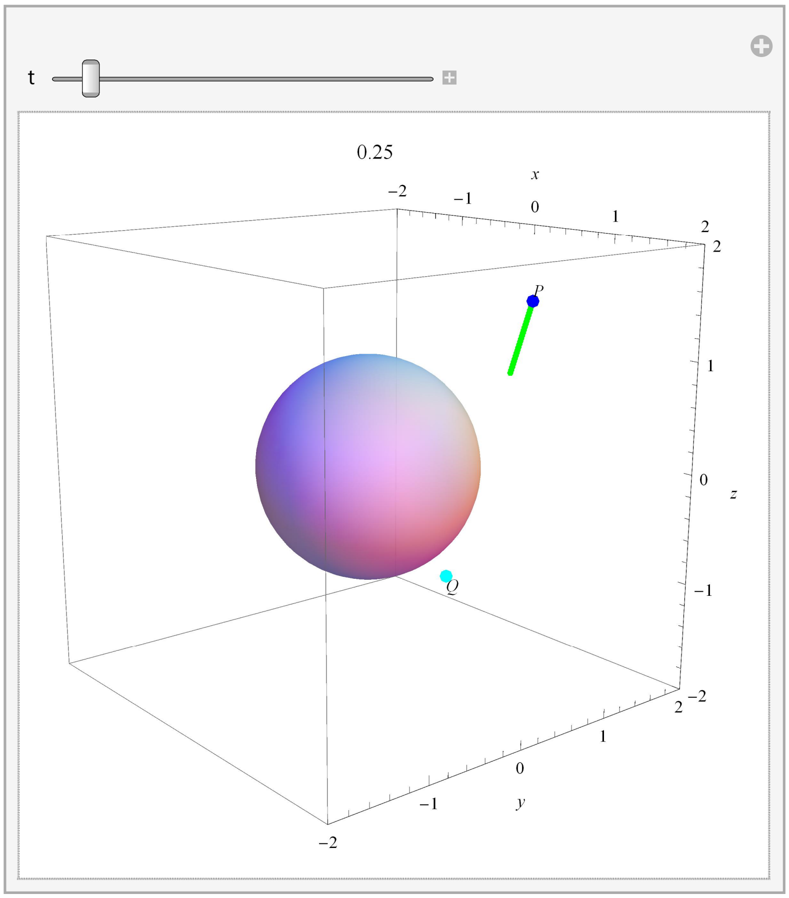
We need a different ViewPoint to see what is happening.

In[75]:=
```
VP = {2.5435418911596623`, -2.052328399828016`, 0.8765516454695087`}
```

Out[75]=
```
{2.54354, -2.05233, 0.876552}
```

In[76]:=
```
Manipulate[
 Graphics3D[{
   {PointSize[0.01], Green, Table[Point[pP1 + s (pQ2 - pP1)], {s, 0, t, .01}]},
   {PointSize[0.02], Blue, Point[pP1]},
   {PointSize[0.02], Cyan, Point[pQ2]},
   {Opacity[0.75], Sphere[{0, 0, 0}, 1]},
   {Text[P, pP1, {-1, -1}], Text[Q, pQ2, {-1, 1}]}
  },
  PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}},
  AxesLabel → {x, y, z},
  ViewPoint → VP
 ],
 {{t, 1 / 4}, 0, 4}]
```

Out[76]=



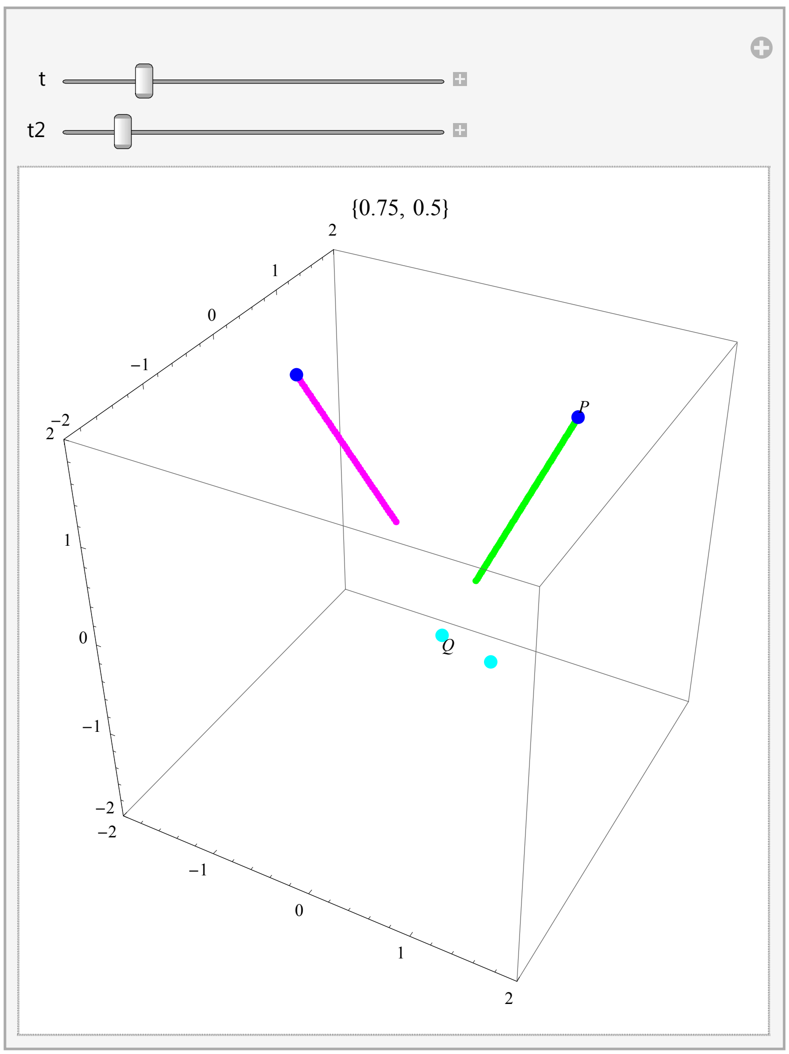
Now it is clear that this line gets very close to the unit sphere, but does not touch it.

■ Two lines

Two pairs of points determine two lines.

In[77]:=
```
pP1 = {1, 1, 3 / 2}; pQ1 = {1 / 2, -1 / 2, 0};

pP2 = {-3 / 2, 1 / 2, 3 / 2}; pQ2 = {1 / 2, 1 / 2, -1};


Manipulate[
 Graphics3D[{
   {PointSize[0.01], Green, Table[Point[pP1 + s (pQ1 - pP1)], {s, 0, t, .01}]},
   {PointSize[0.01], Magenta, Table[Point[pP2 + s2 (pQ2 - pP2)], {s2, 0, t2, .01}]},
   {PointSize[0.02], Blue, Point[pP1], , Point[pP2]},
   {PointSize[0.02], Cyan, Point[pQ1], Point[pQ2]},

   {Text[P, pP1, {-1, -1}], Text[Q, pQ1, {-1, 1}]}
  },
  PlotLabel → {N[t], N[t2]},
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}}
 ],
 {{t, .75}, 0, 4}, {{t2, .5}, 0, 4}]
```

Out[79]=



Do these lines intersect? Here is the algebraic answer. The parametric equations of these lines are

In[80]:=
```
pP1 + t (pQ1 - pP1)
```

Out[80]=
$$\left\{1-\frac{t}{2},\ 1-\frac{3\,t}{2},\ \frac{3}{2}-\frac{3\,t}{2}\right\}$$

In[81]:=
```
pP2 + s (pQ2 - pP2)
```

Out[81]=
$$\left\{-\frac{3}{2}+2\,s,\ \frac{1}{2},\ \frac{3}{2}-\frac{5\,s}{2}\right\}$$

Do they have a common point?

In[82]:=
$$\text{Solve}\left[\left\{1-\frac{t}{2}==-\frac{3}{2}+2\,s,\ 1-\frac{3\,t}{2}==\frac{1}{2},\ \frac{3}{2}-\frac{3\,t}{2}==\frac{3}{2}-\frac{5\,s}{2}\right\},\ \{s,\ t\}\right]$$

Out[82]= { }

No solutions, so these two lines do not intersect.

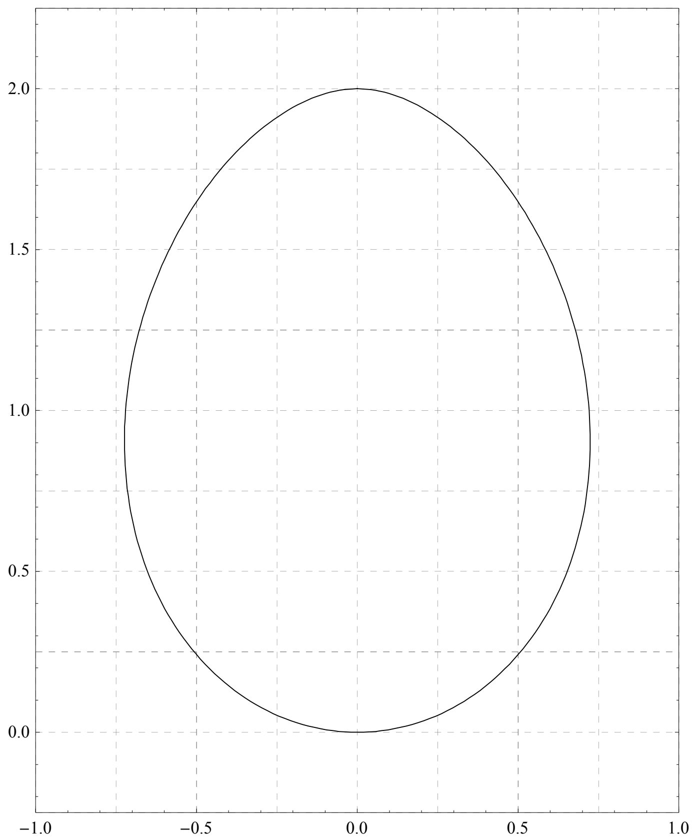
Miscellaneous

■ An egg

This parametric equation of a cross section of an egg I found on the Internet.

In[83]:=
```
Graphics[{
  Line[Table[{0.78 Cos[θ/4] Sin[θ], 1 - Cos[θ]}, {θ, -Pi, Pi, Pi/128}]]
 },
 Frame → True, PlotRange → {{-1., 1}, {-.25, 2.25}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4]}
]
```

Out[83]=

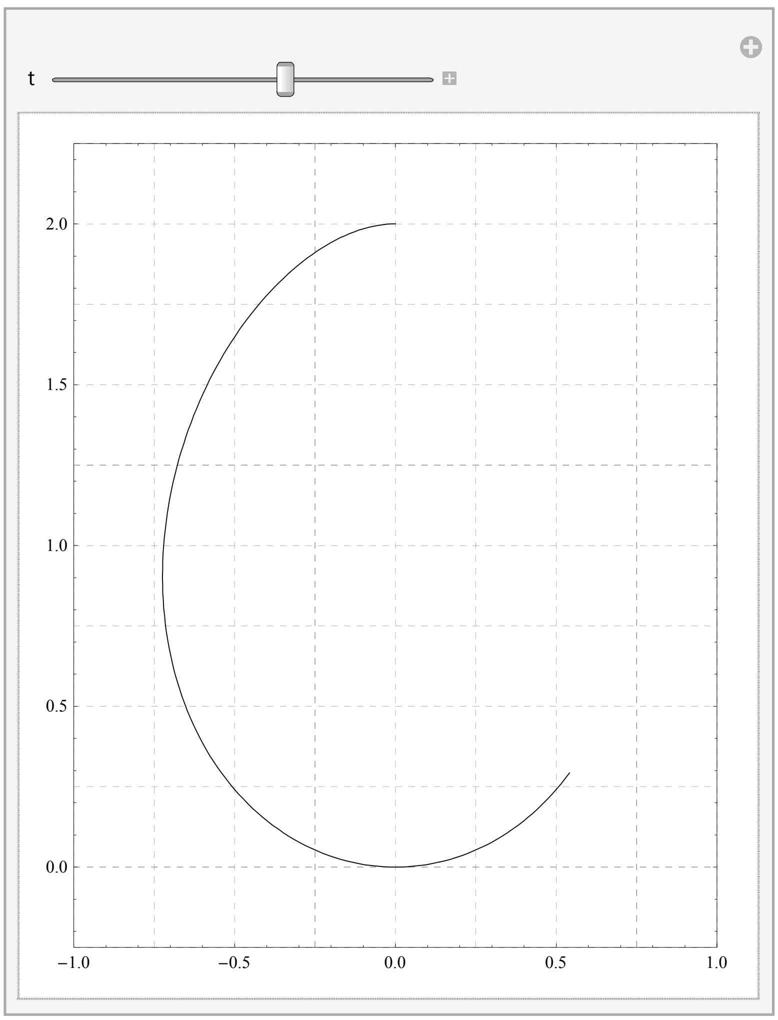


And you can draw an egg using Manipulate[]

In[84]:=

```
Manipulate[
 Graphics[{
   Line[Table[{0.78 Cos[θ/4] Sin[θ], 1 - Cos[θ]}, {θ, -Pi, t, Pi/128}]]
  },
  Frame → True, PlotRange → {{-1., 1}, {-.25, 2.25}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4]}
 ],
 {{t, Pi/4}, -Pi, Pi, Pi/128}]
```

Out[84]=



Velocity

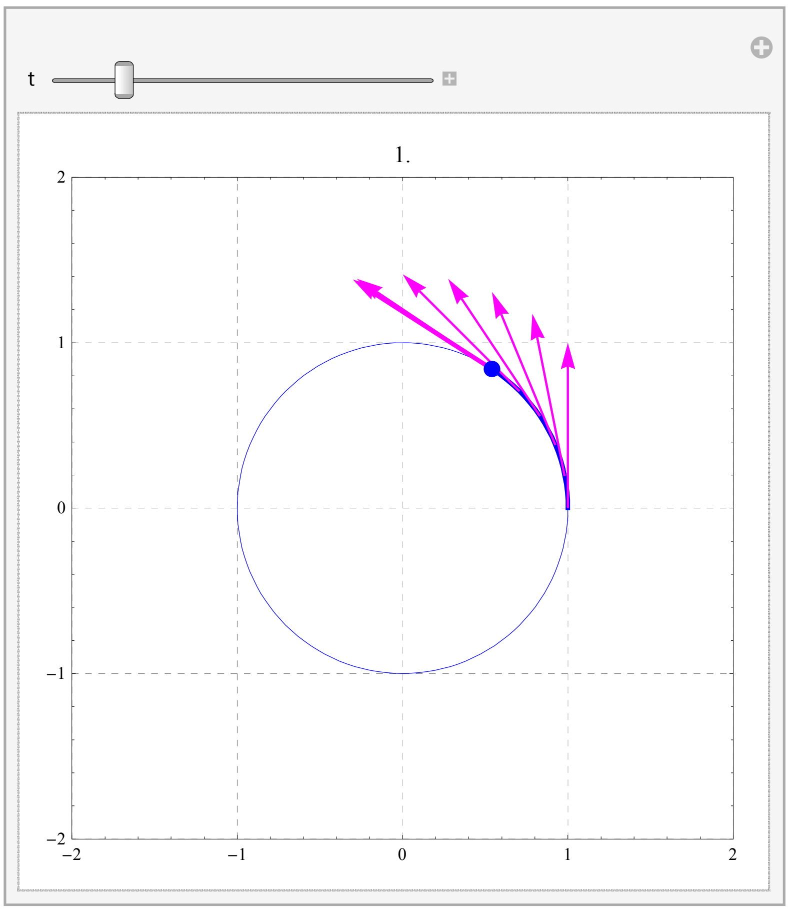
Each parametric curve studied above can be interpreted as a moving particle which leaves a trace: the parametric curve. For each curve we will name its parametric equation, find the velocity vector and illustrate on the graph of the curve.

■ The unit circle

```
In[85]:= Clear[t, r1];

In[86]:= r1[t_] := {Cos[t], Sin[t]}

In[87]:= D[r1[t], t]

Out[87]= {-Sin[t], Cos[t]}

In[88]:= Clear[v1]; v1[t_] := {-Sin[t], Cos[t]}
```

In[89]:=
```
Manipulate[
 Graphics[{
   {Thickness[0.001], Blue, Line[Table[r1[v], {v, 0, 2 Pi, Pi/64}]]},
   {Thickness[0.007], Blue, Line[Table[r1[v], {v, 0, t, Pi/64}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r1[v], r1[v] + v1[v]}], {v, 0, t, Pi/16}]},
   {Thickness[0.007], Magenta, Arrow[{r1[t], r1[t] + v1[t]}]},
   {PointSize[0.025], Blue, Point[r1[t]]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-2, 2}, {-2, 2}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, 1}, 0, 2 Pi, Pi/64}]
```

Out[89]=



■ Clover

In[90]:=
```
Clear[t, r2];
```

In[91]:=
```
r2[t_] := (1 + Cos[3 t]) {Cos[t], Sin[t]}
```

In[92]:=
```
D[r2[t], t]
```

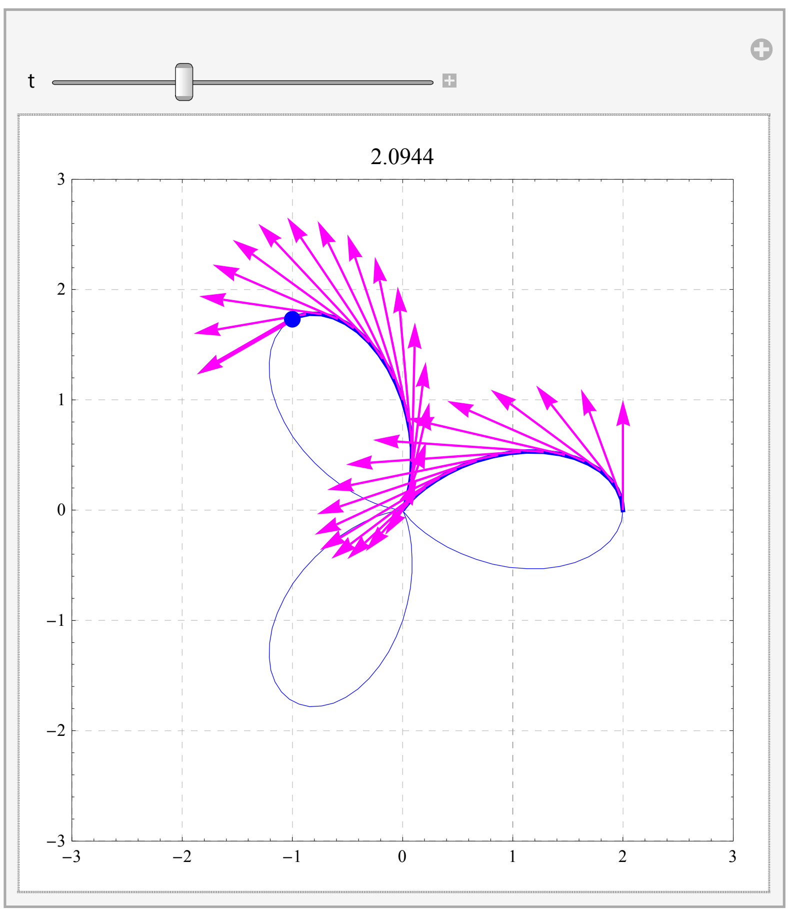
Out[92]=
```
{- (1 + Cos[3 t]) Sin[t] - 3 Cos[t] Sin[3 t], Cos[t] (1 + Cos[3 t]) - 3 Sin[t] Sin[3 t]}
```

For esthetic reasons, in the picture below I will uniformly shorten each velocity vector to half of its magnitude.

In[93]:=
```
Clear[v2];
v2[t_] := 1/2 {- (1 + Cos[3 t]) Sin[t] - 3 Cos[t] Sin[3 t], Cos[t] (1 + Cos[3 t]) - 3 Sin[t] Sin[3 t]}
```

In[94]:=
```
Manipulate[
  Graphics[{
    {Thickness[0.001], Blue, Line[Table[r2[v], {v, 0, 2 Pi, Pi/64}]]},
    {Thickness[0.007], Blue, Line[Table[r2[v], {v, 0, t, Pi/64}]]},
    {Thickness[0.0035], Magenta, Table[Arrow[{r2[v], r2[v] + v2[v]}], {v, 0, t, Pi/(3 * 16)}]},
    {Thickness[0.007], Magenta, Arrow[{r2[t], r2[t] + v2[t]}]},
    {PointSize[0.025], Blue, Point[r2[t]]}
   }, PlotLabel → N[t],
   Frame → True, PlotRange → {{-3, 3}, {-3, 3}},
   AspectRatio → Automatic,
   GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
     {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
  ], {{t, 2 Pi/3}, 0, 2 Pi, Pi/64}]
```

Out[94]=



■ **Cardioid**

In[95]:=
```
Clear[t, r3]; r3[t_] := (1 + Cos[t]) {Cos[t], Sin[t]}
```

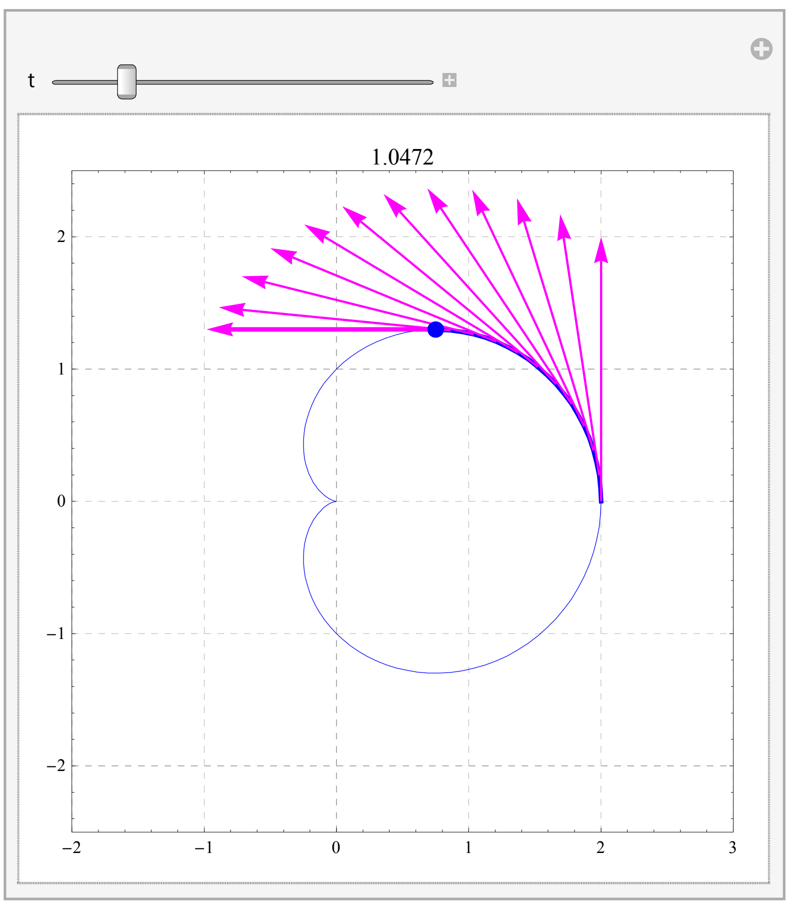
In[96]:=
```
D[r3[t], t]
```

Out[96]=
```
{-Cos[t] Sin[t] - (1 + Cos[t]) Sin[t], Cos[t] (1 + Cos[t]) - Sin[t]^2}
```

In[97]:=
```
Clear[v3]; v3[t_] := {-Cos[t] Sin[t] - (1 + Cos[t]) Sin[t], Cos[t] (1 + Cos[t]) - Sin[t]^2}
```

In[98]:=
```
Manipulate[
 Graphics[{
   {Thickness[0.001], Blue, Line[Table[r3[v], {v, 0, 2 Pi, Pi/64}]]},
   {Thickness[0.007], Blue, Line[Table[r3[v], {v, 0, t, Pi/64}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r3[v], r3[v] + v3[v]}], {v, 0, t, Pi/(2*16)}]},
   {Thickness[0.007], Magenta, Arrow[{r3[t], r3[t] + v3[t]}]},
   {PointSize[0.025], Blue, Point[r3[t]]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-2, 3}, {-2.5, 2.5}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi/3}, 0, 2 Pi, Pi/64}]
```

Out[98]=



■ Unnamed curve

In[99]:=
```
Clear[t, r4]; r4[t_] := (1 + Cos[2 t]^2) {Cos[t], Sin[t]}
```

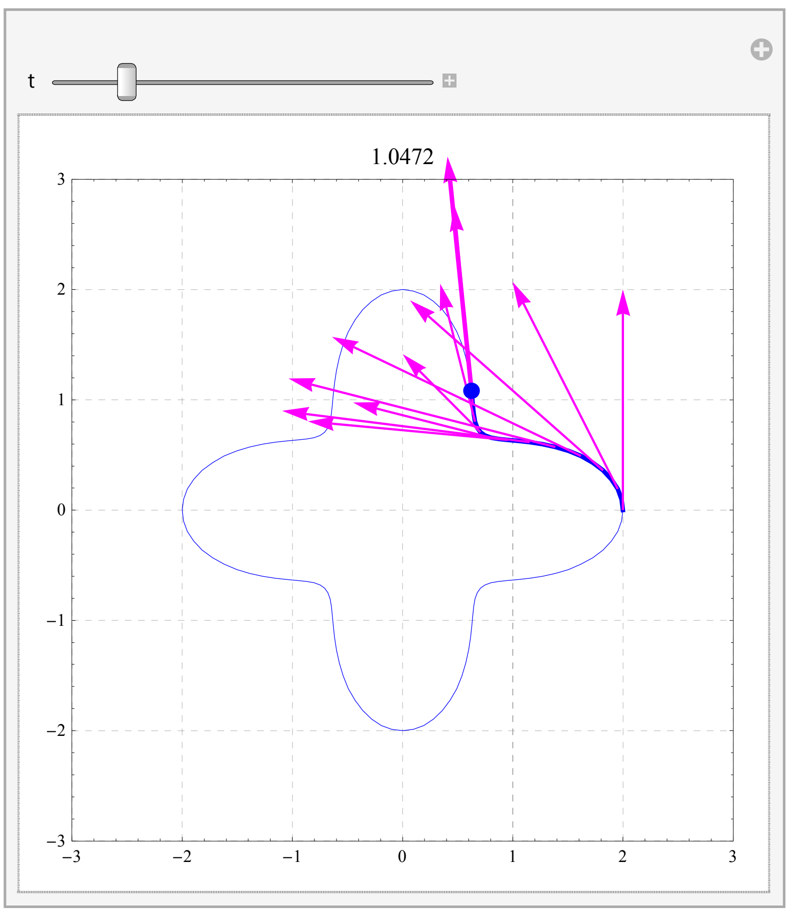
In[100]:=
```
D[r4[t], t]
```

Out[100]=
```
{-(1 + Cos[2 t]^2) Sin[t] - 4 Cos[t] Cos[2 t] Sin[2 t],
 Cos[t] (1 + Cos[2 t]^2) - 4 Cos[2 t] Sin[t] Sin[2 t]}
```

In[101]:=
```
Clear[v4]; v4[t_] := {-(1 + Cos[2 t]^2) Sin[t] - 4 Cos[t] Cos[2 t] Sin[2 t],
   Cos[t] (1 + Cos[2 t]^2) - 4 Cos[2 t] Sin[t] Sin[2 t]}
```

In[102]:=
```
Manipulate[
 Graphics[{
   {Thickness[0.001], Blue, Line[Table[r4[v], {v, 0, 2 Pi, Pi/64}]]},
   {Thickness[0.007], Blue, Line[Table[r4[v], {v, 0, t, Pi/64}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r4[v], r4[v] + v4[v]}], {v, 0, t, Pi/(2 * 16)}]},
   {Thickness[0.007], Magenta, Arrow[{r4[t], r4[t] + v4[t]}]},
   {PointSize[0.025], Blue, Point[r4[t]]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-3, 3}, {-3, 3}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
 ], {{t, Pi/3}, 0, 2 Pi, Pi/64}]
```

Out[102]=



- **Egg**

In[103]:=
```
Clear[t, r5]; r5[t_] := {0.78 Cos[t/4] Sin[t], 1 - Cos[t]}
```

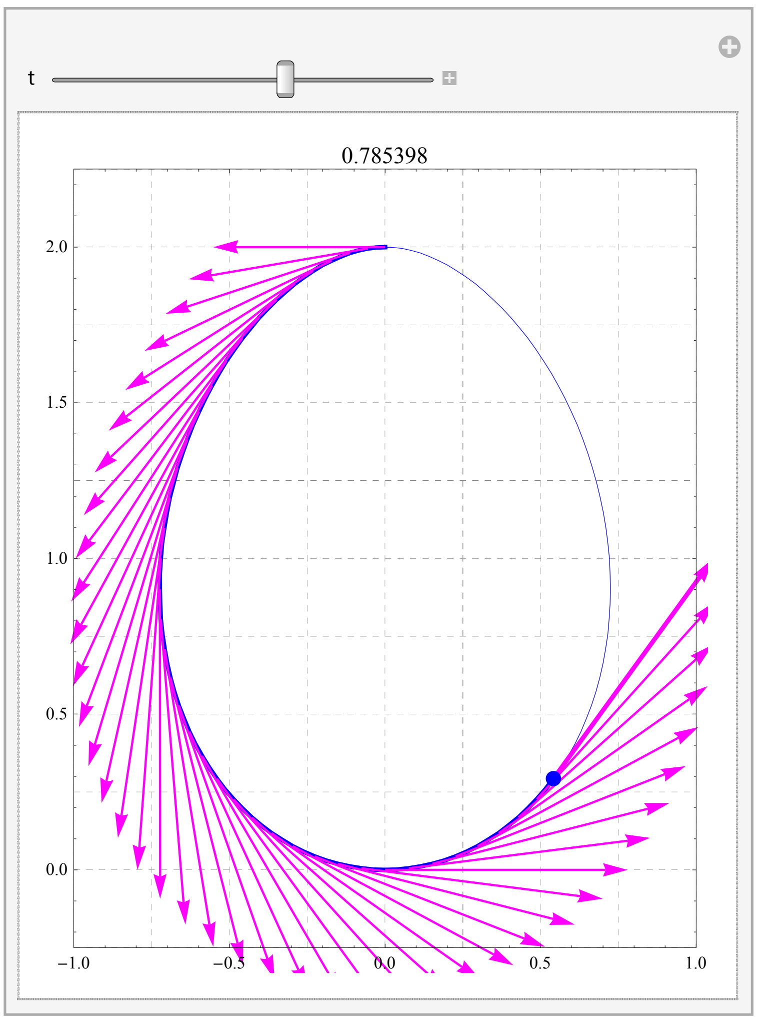
In[104]:=
```
D[r5[t], t]
```

Out[104]= $\left\{0.78 \,\text{Cos}\left[\frac{t}{4}\right] \text{Cos}[t] - 0.195 \,\text{Sin}\left[\frac{t}{4}\right] \text{Sin}[t], \text{Sin}[t]\right\}$

In[105]:=
```
Clear[v5]; v5[t_] := {0.78` Cos[t/4] Cos[t] - 0.195` Sin[t/4] Sin[t], Sin[t]}
```

In[106]:=
```
Manipulate[
 Graphics[{
   {Thickness[0.001], Blue, Line[Table[r5[v], {v, -Pi, Pi, Pi/64}]]},
   {Thickness[0.007], Blue, Line[Table[r5[v], {v, -Pi, t, Pi/64}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r5[v], r5[v] + v5[v]}], {v, -Pi, t, Pi/(2 * 16)}]},
   {Thickness[0.007], Magenta, Arrow[{r5[t], r5[t] + v5[t]}]},
   {PointSize[0.025], Blue, Point[r5[t]]}
  }, PlotLabel → N[t],
  Frame → True, PlotRange → {{-1, 1}, {-.25, 2.25}},
  AspectRatio → Automatic,
  GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4],
    {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4]}
 ], {{t, Pi/4}, -Pi, Pi, Pi/64}]
```

Out[106]=



■ Helix

In[107]:=
```
Clear[t, r6]; r6[t_] := {Sin[t], Cos[t], t/(2 Pi)}
```

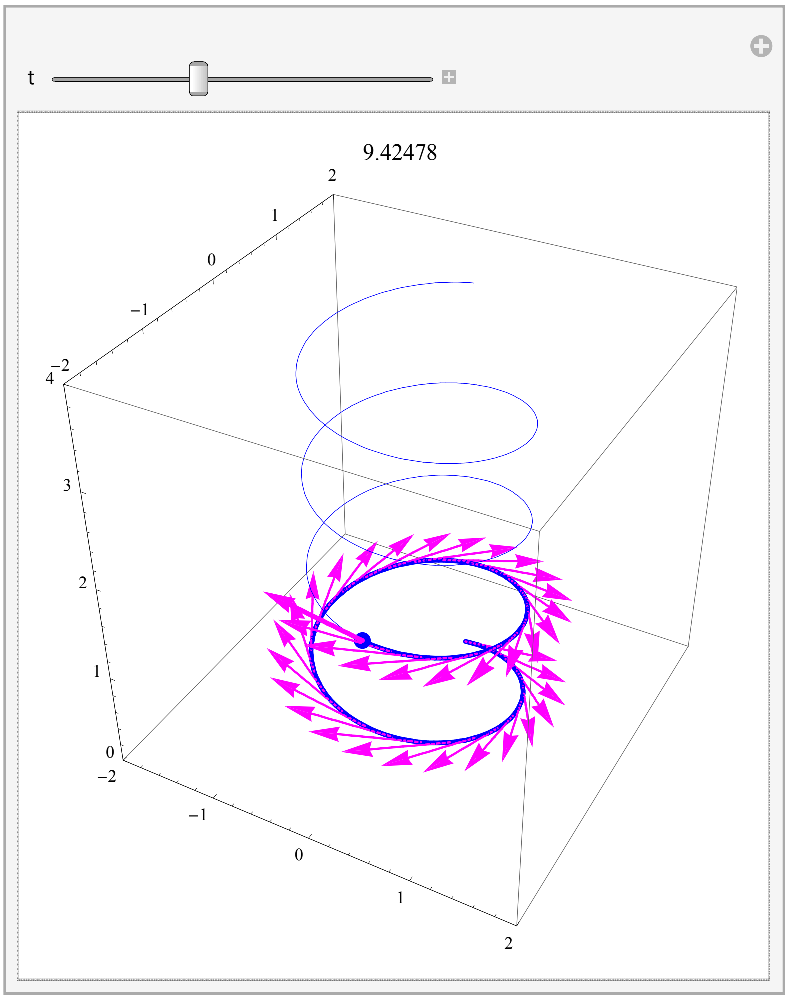
In[108]:=
```
D[r6[t], t]
```

Out[108]= $\left\{\text{Cos}[t], -\text{Sin}[t], \frac{1}{2\pi}\right\}$

In[109]:=
```
Clear[v6]; v6[t_] := {Cos[t], -Sin[t], 1/(2 π)}
```

In[110]:=
```
Manipulate[
 Graphics3D[{
   {Thickness[0.001], Blue, Line[Table[r6[v], {v, 0, 8 Pi, Pi/64}]]},
   {Thickness[0.007], Blue, Line[Table[r6[v], {v, 0, t, Pi/64}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r6[v], r6[v] + v6[v]}], {v, 0, t, Pi/12}]},
   {Thickness[0.007], Magenta, Arrow[{r6[t], r6[t] + v6[t]}]},
   {PointSize[0.025], Blue, Point[r6[t]]}
  }, PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-2, 2}, {-2, 2}, {0, 4}},
  BoxRatios → {1, 1, 1}
 ], {{t, 3 Pi}, 0, 8 Pi, Pi/64}]
```

Out[110]=



■ Conical helix

In[111]:=

In[112]:=
```
Clear[t, r7]; r7[t_] := t/Pi {Sin[8 t], Cos[8 t], 1}
```

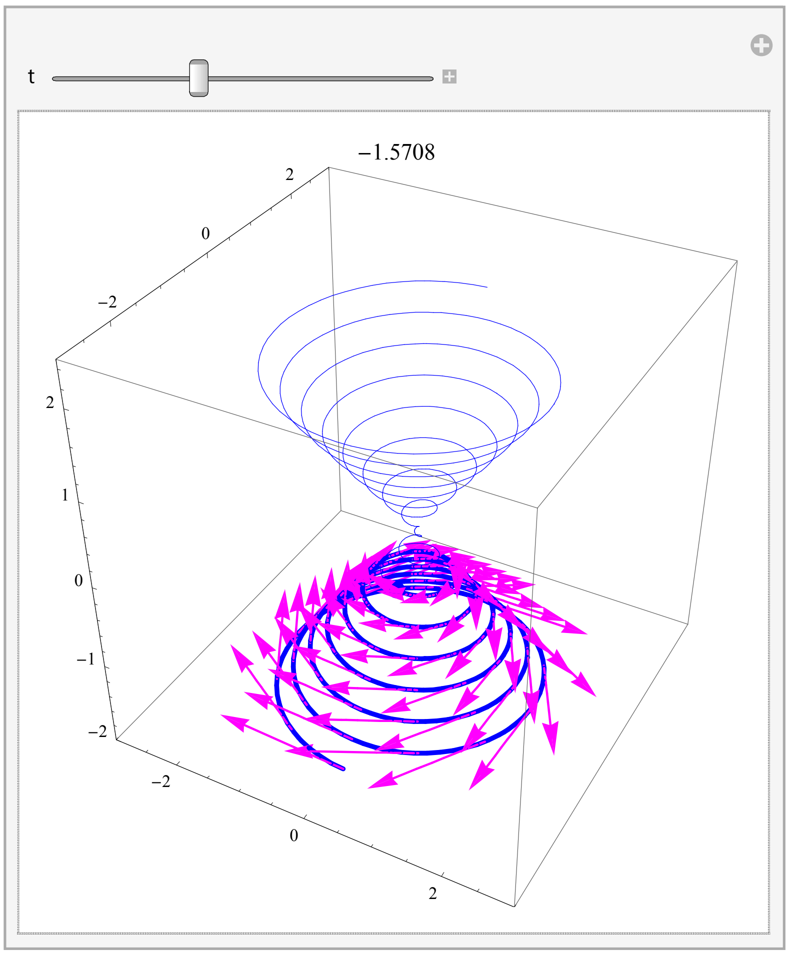
In[113]:=
```
D[r7[t], t]
```

Out[113]= $\left\{\frac{8\,t\,\mathrm{Cos}[8\,t]}{\pi}+\frac{\mathrm{Sin}[8\,t]}{\pi},\ \frac{\mathrm{Cos}[8\,t]}{\pi}-\frac{8\,t\,\mathrm{Sin}[8\,t]}{\pi},\ \frac{1}{\pi}\right\}$

In[114]:=
```
Clear[v7]; v7[t_] := 1/8 {8 t Cos[8 t]/π + Sin[8 t]/π, Cos[8 t]/π - 8 t Sin[8 t]/π, 1/π}
```

In[115]:=
```
Manipulate[
 Graphics3D[{
   {Thickness[0.001], Blue, Line[Table[r7[v], {v, -2 Pi, 2 Pi, Pi/(2 × 128)}]]},
   {Thickness[0.007], Blue, Line[Table[r7[v], {v, -2 Pi, t, Pi/(2 × 128)}]]},
   {Thickness[0.0035], Magenta, Table[Arrow[{r7[v], r7[v] + v7[v]}], {v, - 2 Pi, t, Pi/48}]},
   {Thickness[0.007], Magenta, Arrow[{r7[t], r7[t] + v7[t]}]},
   {PointSize[0.025], Blue, Point[r7[t]]}
  }, PlotLabel → N[t],
  Boxed → True, Axes → True, PlotRange → {{-3, 3}, {-3, 3}, {-2, 2.5}},
  BoxRatios → {1, 1, 1}
 ], {{t, -Pi / 2}, -2 Pi, 2 Pi, Pi/64}]
```

Out[115]=



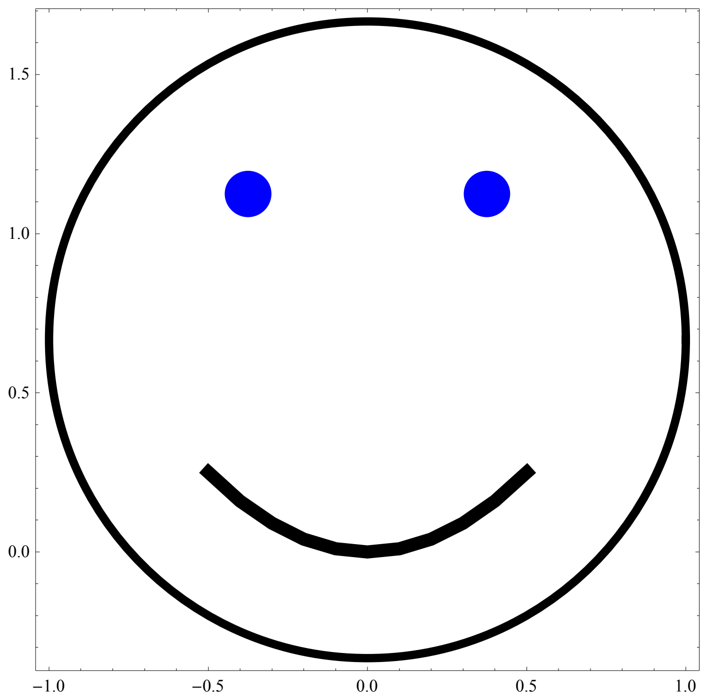
Length

■ Smile

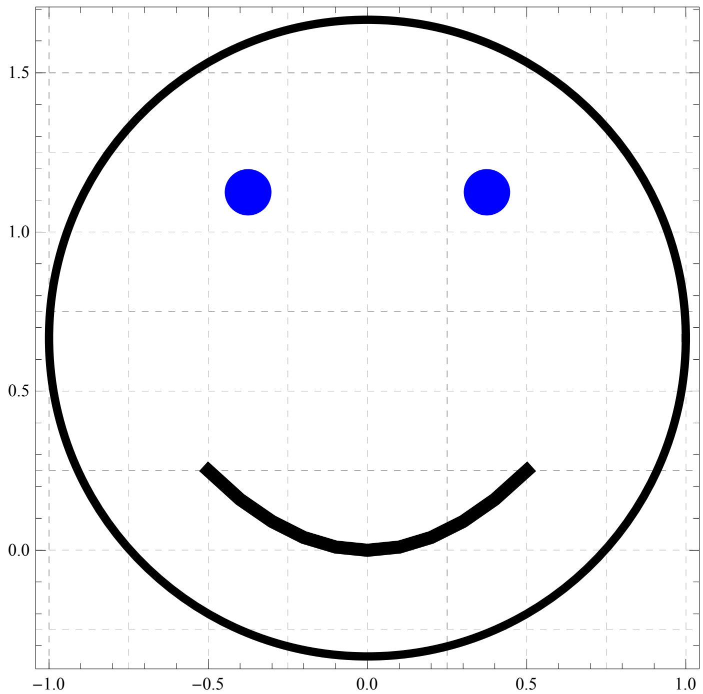
What is a smile mathematically? It could be defined as a graph of the square function near the origin; for example for x between -1/2 and 1/2.

In[116]:=
```
Graphics[{
  {Thickness[0.0125], Circle[{0, 2 / 3}, 1]},
  {Blue, PointSize[0.07], Point[{-3 / 8, 9 / 8}], Point[{3 / 8, 9 / 8}]},
  {Thickness[0.02], Line[Table[{x, x^2}, {x, -1 / 2, 1 / 2, .1}]]}
 },
 Frame → True
]
```

Out[116]=



In[117]:=
```
Graphics[{
  {Thickness[0.0125], Circle[{0, 2 / 3}, 1]},
  {Blue, PointSize[0.07], Point[{-3 / 8, 9 / 8}], Point[{3 / 8, 9 / 8}]},
  {Thickness[0.02], Line[Table[{x, x^2}, {x, -1 / 2, 1 / 2, .1}]]}
 },
 Frame → True,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10, 1 / 4]}
]
```

Out[117]=


The parametric equation of a smile is

In[118]:= `rs[t_] := {t, t^2}`

In[119]:= `D[rs[t], t]`

Out[119]= `{1, 2 t}`

Then magnitude of this vector is

In[120]:= $\sqrt{\{1, 2\,t\}.\{1, 2\,t\}}$

Out[120]= $\sqrt{1 + 4\,t^2}$

The length of this smile is

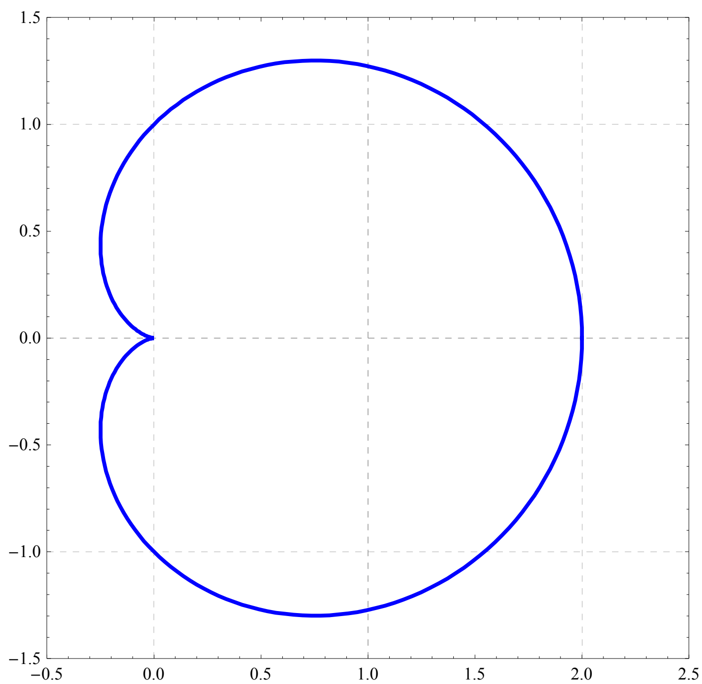
In[121]:= `Integrate[`$\sqrt{1 + 4\,t^2}$`, {t, -1 / 2, 1 / 2}]`

Out[121]= $\frac{1}{2}\left(\sqrt{2} + \text{ArcSinh}[1]\right)$

■ Cardioid

In[122]:=
```
Clear[t, rc];

rc[t_] := (1 + Cos[t]) {Cos[t], Sin[t]};

Graphics[{
  {Thick, Blue, Line[Table[rc[v], {v, 0, 2 Pi, Pi/128}]]}
 },
 Frame → True, PlotRange → {{-.5, 2.5}, {-1.5, 1.5}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

Out[124]=



In[125]:= `rc[t]`

Out[125]= `{Cos[t] (1 + Cos[t]), (1 + Cos[t]) Sin[t]}`

In[126]:= `FullSimplify[D[rc[t], t]]`

Out[126]= `{- (1 + 2 Cos[t]) Sin[t], Cos[t] + Cos[2 t]}`

In[127]:=
```
FullSimplify[
 {- (1 + 2 Cos[t]) Sin[t], Cos[t] + Cos[2 t]}.{- (1 + 2 Cos[t]) Sin[t], Cos[t] + Cos[2 t]}]
```

Out[127]= `2 (1 + Cos[t])`

The length of the cardioid is

In[128]:= `Integrate[Sqrt[2 (1 + Cos[t])], {t, 0, 2 Pi}]`

Out[128]= 8

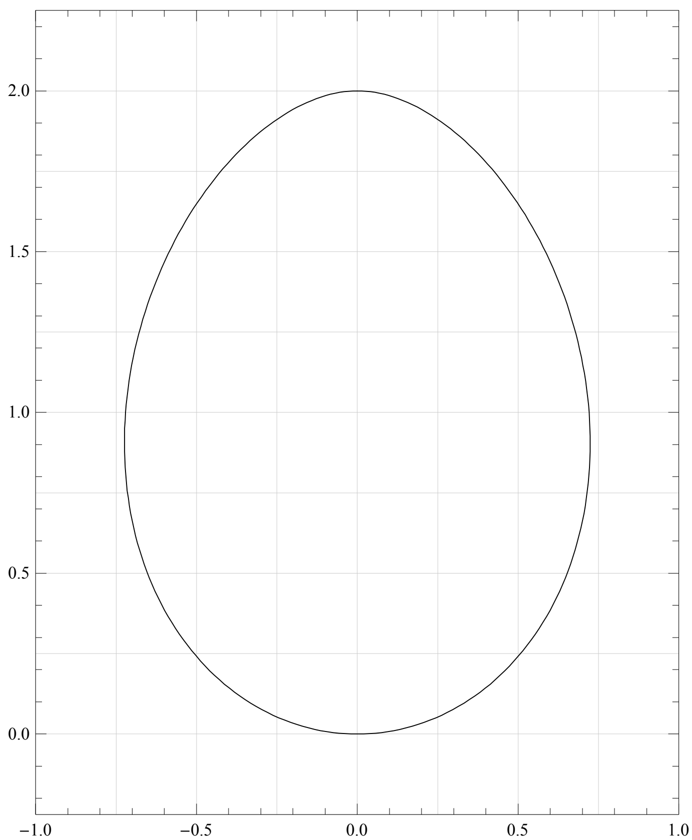
In[129]:=
```
Integrate[√(2 (1 + Cos[t])) , t]
```

Out[129]=
$$2\sqrt{2}\ \sqrt{1+\mathrm{Cos}[t]}\ \mathrm{Tan}\left[\frac{t}{2}\right]$$

- **Egg**

In[130]:=
```
Graphics[{
  Line[Table[{0.78 Cos[θ/4] Sin[θ], 1 - Cos[θ]}, {θ, -Pi, Pi, Pi/128}]]
 },
 Frame → True, PlotRange → {{-1., 1}, {-.25, 2.25}},
 AspectRatio → Automatic, GridLines → {{#, {GrayLevel[0.8]}} & /@ Range[-10, 10, 1 / 4],
   {#, {GrayLevel[0.8]}} & /@ Range[-10, 10, 1 / 4]}
]
```

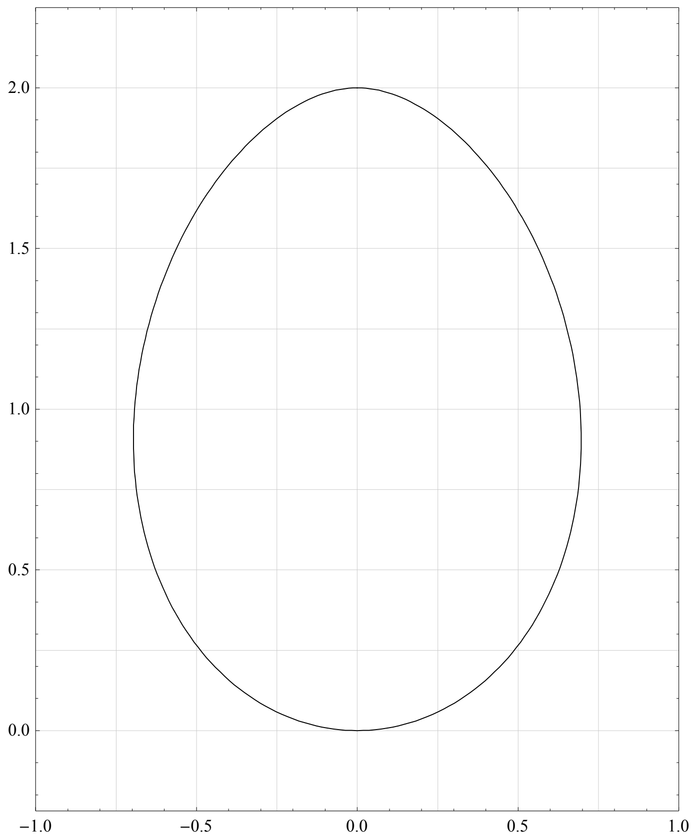
Out[130]=



I will modify this egg to

In[131]:=
```
Graphics[{
  Line[Table[{3/4 Cos[θ/4] Sin[θ], 1 - Cos[θ]}, {θ, -Pi, Pi, Pi/128}]]
 },
 Frame → True, PlotRange → {{-1., 1}, {-.25, 2.25}},
 AspectRatio → Automatic, GridLines → {{#, {GrayLevel[0.8]}} & /@ Range[-10, 10, 1 / 4],
   {#, {GrayLevel[0.8]}} & /@ Range[-10, 10, 1 / 4]}
]
```

Out[131]=



In[132]:=
```
D[{3/4 Cos[θ/4] Sin[θ], 1 - Cos[θ]}, θ]
```

Out[132]= $\left\{\frac{3}{4}\,\mathrm{Cos}\left[\frac{\theta}{4}\right]\,\mathrm{Cos}[\theta]-\frac{3}{16}\,\mathrm{Sin}\left[\frac{\theta}{4}\right]\,\mathrm{Sin}[\theta],\ \mathrm{Sin}[\theta]\right\}$

In[133]:=
```
FullSimplify[
 {3/4 Cos[θ/4] Cos[θ] - 3/16 Sin[θ/4] Sin[θ], Sin[θ]}.{3/4 Cos[θ/4] Cos[θ] - 3/16 Sin[θ/4] Sin[θ], Sin[θ]}]
```

Out[133]= $\frac{9\left(3\,\mathrm{Cos}\left[\frac{3\theta}{4}\right]+5\,\mathrm{Cos}\left[\frac{5\theta}{4}\right]\right)^2}{1024}+\mathrm{Sin}[\theta]^2$

The integral below is a difficult integral, it takes too long to evaluate.

In[134]:=
```
(* Integrate[√(1/10249 (3 Cos[3θ/4]+5 Cos[5θ/4])^2+Sin[θ]^2),{θ,-Pi,Pi}] *)
```

So, find a numerical approximation

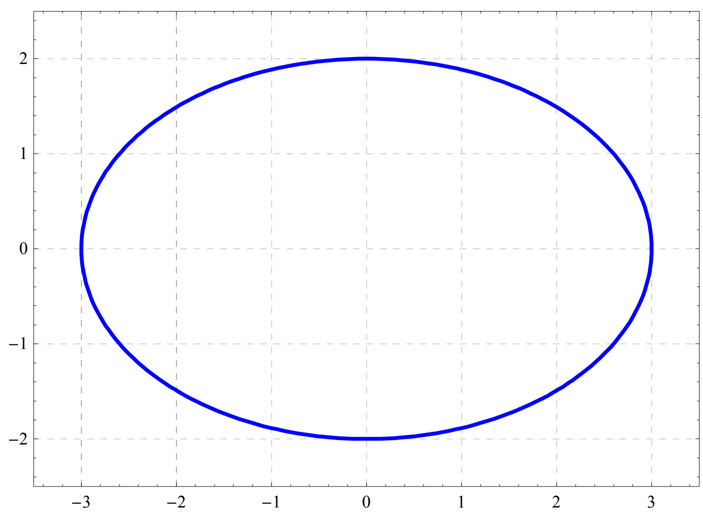
In[135]:=
```
NIntegrate[Sqrt[(1/1024  9 (3 Cos[3θ/4] + 5 Cos[5θ/4])^2 + Sin[θ]^2)], {θ, -Pi, Pi}]
```

Out[135]= 5.34129

- **Ellipse**

In[136]:=
```
Clear[t, a, b, rel];

rel[t_, a_, b_] := {a Cos[t], b Sin[t]};

Graphics[{
  {Thick, Blue, Line[Table[rel[v, 3, 2], {v, 0, 2 Pi, Pi/128}]]}
 },
 Frame → True, PlotRange → {{-3.5, 3.5}, {-2.5, 2.5}},
 AspectRatio → Automatic,
 GridLines → {{#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10],
   {#, {GrayLevel[0.5], Dashing[{0.01, 0.01}]}} & /@ Range[-10, 10]}
]
```

Out[138]=



In[139]:= `D[rel[t, a, b], t]`

Out[139]= {-a Sin[t], b Cos[t]}

In[140]:= `{-a Sin[t], b Cos[t]}.{-a Sin[t], b Cos[t]}`

Out[140]= $b^2 \text{Cos}[t]^2 + a^2 \text{Sin}[t]^2$

Thus, the length of the specific ellipse that we plotted above is

In[141]:=
```
Integrate[Sqrt[2^2 Cos[t]^2 + 3^2 Sin[t]^2], {t, 0, 2 Pi}, Assumptions :> And[a > 0, b > 0]]
```

Out[141]= $8\,\text{EllipticE}\left[-\frac{5}{4}\right]$

This shows that this integral is not calculable using the functions that we learn in Pre-calculus. A numerical approximation is

In[142]:=
```
N[8 EllipticE[-5/4]]
```

Out[142]= 15.8654

We can expect that the general case will involve EllipticE function. However, to calculate the general integral one needs to use an option for the Integral[].

Calculating the general integral takes 48 seconds

In[143]:=
```
(* Timing[Integrate[√(b^2 Cos[t]^2+a^2 Sin[t]^2),{t,0,2Pi},Assumptions:→And[a>0,b>0]]] *)
```

It is a little easier to calculate

In[144]:=
```
Timing[Integrate[√(Cos[t]^2 + (a)^2 Sin[t]^2), {t, 0, 2 Pi}, Assumptions :→ And[a > 0]]]
```

Out[144]= $\{27.752,\ 4\,\text{EllipticE}[1-a^2]\}$

Then the general integral equals

In[145]:=
```
4 b EllipticE[1 - (a/b)^2]
```

Out[145]= $4\,b\,\text{EllipticE}\left[1-\frac{a^2}{b^2}\right]$

since $\sqrt{b^2\,\text{Cos}[t]^2 + a^2\,\text{Sin}[t]^2} = b\sqrt{\text{Cos}[t]^2 + \left(\frac{a}{b}\right)^2\text{Sin}[t]^2}$

It is clear that exchanging the role of a and b does not change the length of an ellipse. Therefore $4\,b\,\text{EllipticE}\left[1-\left(\frac{a}{b}\right)^2\right] = 4\,a\,\text{EllipticE}\left[1-\left(\frac{b}{a}\right)^2\right]$. It is interesting that *Mathematica* does not know that the preceding expressions are equal

In[146]:=
```
FullSimplify[b EllipticE[1 - a^2/b^2] - a EllipticE[1 - b^2/a^2], And[a > 0, b > 0]]
```

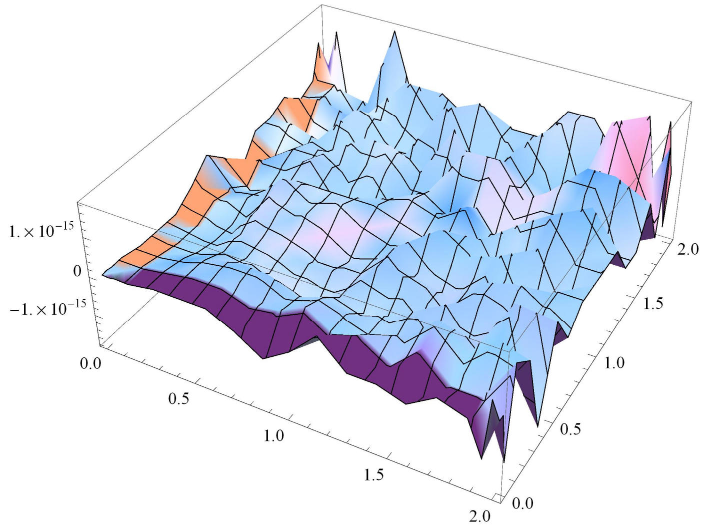
Out[146]= $b\,\text{EllipticE}\left[1-\frac{a^2}{b^2}\right] - a\,\text{EllipticE}\left[1-\frac{b^2}{a^2}\right]$

The above expression should simplify to 0.

In[147]:= `Plot3D[a EllipticE[1 - b^2/a^2] - b EllipticE[1 - a^2/b^2], {a, 0, 2}, {b, 0, 2}]`

$$\text{Plot3D}\left[a\,\text{EllipticE}\left[1-\frac{b^2}{a^2}\right]-b\,\text{EllipticE}\left[1-\frac{a^2}{b^2}\right],\{a,0,2\},\{b,0,2\}\right]$$

Out[147]=

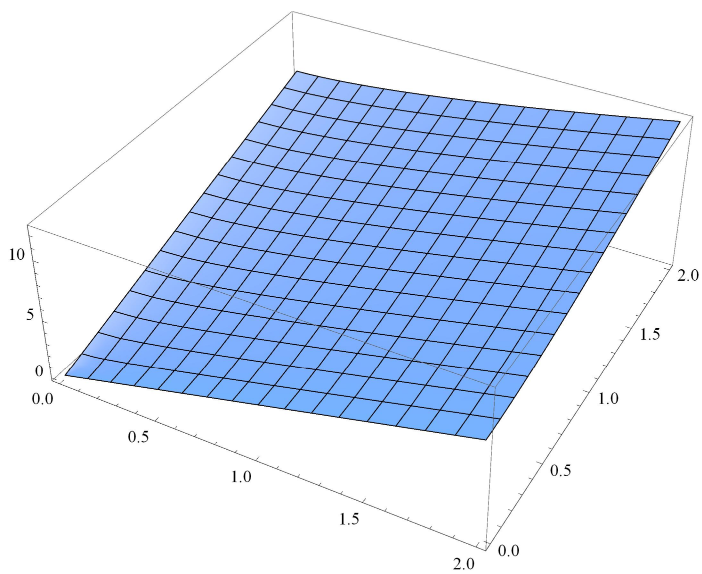


Now explore the function for the length of an ellipse as a function of *a* and *b*.

In[148]:=

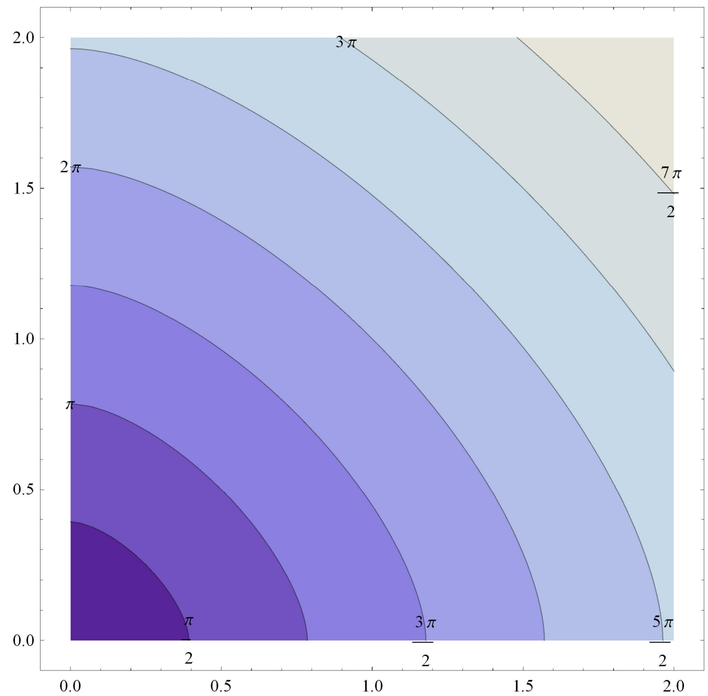
$$\text{Plot3D}\left[4\,b\,\text{EllipticE}\left[1-\frac{a^2}{b^2}\right],\{a,0,2\},\{b,0,2\}\right]$$

Out[148]=



In[149]:=
```
Show[ContourPlot[4 b EllipticE[1 - a^2/b^2], {a, 0, 2}, {b, 0, 2},
  Contours → 2 Pi Range[0, 2, 1/4], ContourLabels → All], PlotRangePadding → 0.1]
```

Out[149]=



In[150]:=
```
Show[ContourPlot[4 b EllipticE[1 - a^2/b^2], {a, 0, 2},
  {b, 0, 2}, Contours → 2 Pi Range[0, 2, 1/4], ContourLabels →
  (Text[Framed[#3], {#3/(2 Pi), #3/(2 Pi)}, Background → White] &)], PlotRangePadding → 0.1]
```

Out[150]=