```
SetOptions[EvaluationNotebook[], CellContext → Notebook]
```

Vectors

A vector in *Mathematica* is a list of numbers,e.g.

```
{0, 1, 0.5}
```

```
{0, 1, 0.5}
```

A list in *Mathematica* is a sequence of objects enclosed in curly brackets. Vectors are lists of numbers or symbols which are not themselves lists. For example

{1, 2, 3} and {a, b, c} are vectors but {1, {2, 3}} is not a vector.

Vectors of the same dimension can be added and multiplied by scalars:

```
{2, 3, 4} + {1, 2, 3}
```

```
{3, 5, 7}
```

```
λ {2, 3, 4}
```

```
{2 λ, 3 λ, 4 λ}
```

Mathematica has a built-in function whcih returns the length of a vector, called norm

```
? Norm
```

Norm[*expr*] gives the norm of a number, vector, or matrix.
Norm[*expr*, *p*] gives the *p*–norm. ≫

For exmple :

```
Norm[{1, 1, 1}]
```

$$\sqrt{3}$$

However, in symbolic expressions Norm assumes that all the variables are complex:

```
Norm[{a, b, c}]
```

$$\sqrt{\text{Abs}[a]^2 + \text{Abs}[b]^2 + \text{Abs}[c]^2}$$

It is possible to deal with this by using Simplify with Assuming as follows:

```
Assuming[Element[{a, b, c}, Reals], Simplify[Norm[{a, b, c}]]]
```

$$\sqrt{a^2 + b^2 + c^2}$$

but it is more efficient to define vector length by means of the dot product (scalar product) of vectors

```
{a, b, c}.{e, f, g}
```

```
a e + b f + c g
```

We will call our new function norm (we use a small letter to distinguish it from the built-in Norm):

```
norm[v_]:=√(v.v)
```

```
norm[{a, b}]
```

$$\sqrt{a^2 + b^2}$$

```
norm[{a, b, c}]
```

$$\sqrt{a^2 + b^2 + c^2}$$

Mathematica can verify the traingle and Cauchy-Schwarz inequality but only for a small number of variables:

```
Reduce[ForAll[{u, v, p, q}, Element[{u, v, p, q}, Reals],
  Norm[{u, v } + {p, q}] ≤ Norm[{u, v}] + Norm[{p, q}]]]
```

```
True
```

```
Reduce[ForAll[{u, v, p, q}, Element[{u, v, p, q}, Reals],
  Abs[{u, v}.{p, q}] ≤ Norm[{p, q}] Norm[{u, v}]]]
```

```
True
```

Derivatives in Mathematica

■ Derivative of an expression

```
expr = x^2 y Exp[1 / z]
```

$$e^{\frac{1}{z}}\, x^2\, y$$

```
D[expr, x]
```

$$2\, e^{\frac{1}{z}}\, x\, y$$

```
D[expr, z]
```

$$-\frac{e^{\frac{1}{z}}\, x^2\, y}{z^2}$$

```
D[expr, v]
```

```
0
```

```
D[expr, {z, 2}] // Simplify
```

$$\frac{e^{\frac{1}{z}}\, x^2\, y\, (1 + 2\, z)}{z^4}$$

$$\frac{\partial^2 \mathrm{expr}}{\partial x\, \partial y}$$

$$2\, e^{\frac{1}{z}}\, x$$

```
D[expr, {{x, y, z}}]
```

$$\left\{2\, e^{\frac{1}{z}}\, x\, y,\ e^{\frac{1}{z}}\, x^2,\ -\frac{e^{\frac{1}{z}}\, x^2\, y}{z^2}\right\}$$

```
D[expr, {{x, y, z}, 2}]
```

$$\begin{pmatrix} 2e^{1/z}y & 2e^{1/z}x & -\frac{2e^{1/z}xy}{z^2} \\ 2e^{1/z}x & 0 & -\frac{e^{1/z}x^2}{z^2} \\ -\frac{2e^{1/z}xy}{z^2} & -\frac{e^{1/z}x^2}{z^2} & \frac{2e^{1/z}yx^2}{z^3}+\frac{e^{1/z}yx^2}{z^4} \end{pmatrix}$$

```
m = D[expr, {{x, y, z}, 3}]
```

$$\begin{pmatrix} \left\{0, 2e^{1/z}, -\frac{2e^{1/z}y}{z^2}\right\} & \left\{2e^{1/z}, 0, -\frac{2e^{1/z}x}{z^2}\right\} & \left\{-\frac{2e^{1/z}y}{z^2}, -\frac{2e^{1/z}x}{z^2}, \frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}\right\} \\ \left\{2e^{1/z}, 0, -\frac{2e^{1/z}x}{z^2}\right\} & \{0,0,0\} & \left\{-\frac{2e^{1/z}x}{z^2}, 0, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}\right\} \\ \left\{-\frac{2e^{1/z}y}{z^2}, -\frac{2e^{1/z}x}{z^2}, \frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}\right\} & \left\{-\frac{2e^{1/z}x}{z^2}, 0, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}\right\} & \left\{\frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}, -\frac{6e^{1/z}yx^2}{z^4}-\frac{6e^{1/z}yx^2}{z^5}-\frac{e^{1/z}yx^2}{z^6}\right\} \end{pmatrix}$$

```
m[[1, 1, 3]]
```

$$-\frac{2e^{\frac{1}{z}}y}{z^2}$$

```
Grad[f[x, y, z], {x, y, z}]
```

$$\left\{f^{(1,0,0)}[x, y, z], f^{(0,1,0)}[x, y, z], f^{(0,0,1)}[x, y, z]\right\}$$

```
? Grad
```

Grad[f, {x_1, …, x_n}] gives the gradient $(\partial f/\partial x_1, \ldots, \partial f/\partial x_n)$.
Grad[f, {x_1, …, x_n}, *chart*] gives the gradient in the coordinates *chart*. ≫

```
? *Tensor*
```

▼ System`

LeviCivitaTensor	TensorDimensions	TensorProduct	TensorRank	TensorSymmetry	TensorWedge
TensorContract	TensorExpand	TensorQ	TensorReduce	TensorTranspose	

TensorSymmetry[*tensor*] gives the symmetry of *tensor* under permutations of its slots.
TensorSymmetry[*tensor*, *slots*] gives the symmetry under permutation of the specified list of slots. ≫

```
TensorQ[m]
```

```
True
```

```
TensorDimensions[m]
```

```
{3, 3, 3}
```

```
TensorTranspose[m]
```

$$\begin{pmatrix} \left\{0, 2e^{1/z}, -\frac{2e^{1/z}y}{z^2}\right\} & \left\{2e^{1/z}, 0, -\frac{2e^{1/z}x}{z^2}\right\} & \left\{-\frac{2e^{1/z}y}{z^2}, -\frac{2e^{1/z}x}{z^2}, \frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}\right\} \\ \left\{2e^{1/z}, 0, -\frac{2e^{1/z}x}{z^2}\right\} & \{0,0,0\} & \left\{-\frac{2e^{1/z}x}{z^2}, 0, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}\right\} \\ \left\{-\frac{2e^{1/z}y}{z^2}, -\frac{2e^{1/z}x}{z^2}, \frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}\right\} & \left\{-\frac{2e^{1/z}x}{z^2}, 0, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}\right\} & \left\{\frac{4e^{1/z}xy}{z^3}+\frac{2e^{1/z}xy}{z^4}, \frac{2e^{1/z}x^2}{z^3}+\frac{e^{1/z}x^2}{z^4}, -\frac{6e^{1/z}yx^2}{z^4}-\frac{6e^{1/z}yx^2}{z^5}-\frac{e^{1/z}yx^2}{z^6}\right\} \end{pmatrix}$$

```
TensorSymmetry[m]
```

```
Symmetric[{1, 2, 3}]
```

Derivative of a function

```
f[x_, y_] := x^2 + y^2
```

```
Derivative[1, 0][f][x, y]
```

$2x$

```
Derivative[0, 1][f][x, y]
```

2 y

```
Derivative[0, 1][f]
```

2 #2 &

```
Derivative[1, 1][Function[{x, y}, x y^2]]
```

Function[{x, y}, 2 y]

```
Derivative[1, 1][#1 #2^3 &][u, v]
```

$3\,v^2$

```
Composition[f, g]'[x]
```

f′[g[x]] g′[x]

```
D[f[g[x]], x]
```

f′[g[x]] g′[x]

```
f[x_, y_] := {x^2 - y^2 + 2 x, x^3 - 3 x^2 + 1}
```

```
D[f[x, y], {{x, y}}]
```

$$\begin{pmatrix} 2x+2 & -2y \\ 3x^2-6x & 0 \end{pmatrix}$$

```
? Derivative
```

f' represents the derivative of a function f of one argument.
Derivative[n_1, n_2, ...][f] is the general form, representing
a function obtained from f by differentiating n_1 times with respect to the first argument, n_2 times with respect to the second argument, and so on. ≫

```
g[t_] := {Sin[t], Cos[t]}
```

```
g'[t]
```

{Cos[t], -Sin[t]}

Parametric plots of curves in 3D and 2D

A parametric curve in $\mathbb{R}^n$ is defined the image of a smooth function $r: I \to \mathbb{R}^n$, where I is some interval (not necessarily finite). The same curve may be defined by more than one parametrisations.

$$r_1(t\_) := \{\cos(t), \sin(t), t\}$$

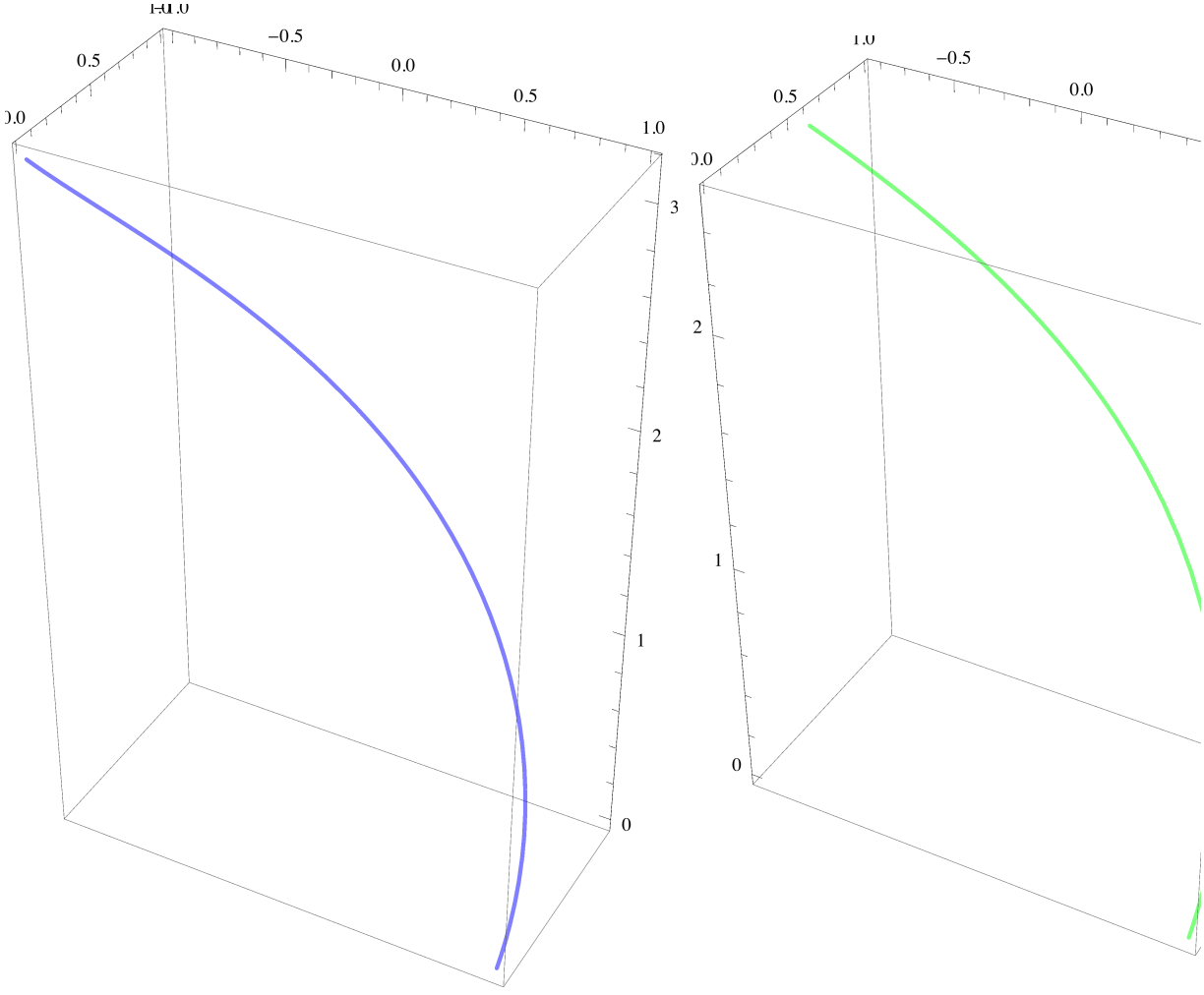
$$r_2(t\_) := \left\{\frac{1-t^2}{t^2+1}, \frac{2t}{t^2+1}, 2\tan^{-1}(t)\right\}$$

To plot a curve in $\mathbb{R}^3$ we use the built-in *Mathematica* function ParametricPlot3D. (In $\mathbb{R}^2$ ParametricPlot).

```
p1 = ParametricPlot3D[r1[t], {t, 0, Pi}, PlotStyle → Directive[Opacity[0.5], Blue]];
p2 = ParametricPlot3D[r2[t], {t, 0, Pi}, PlotStyle → Directive[Opacity[0.5], Green]];
```

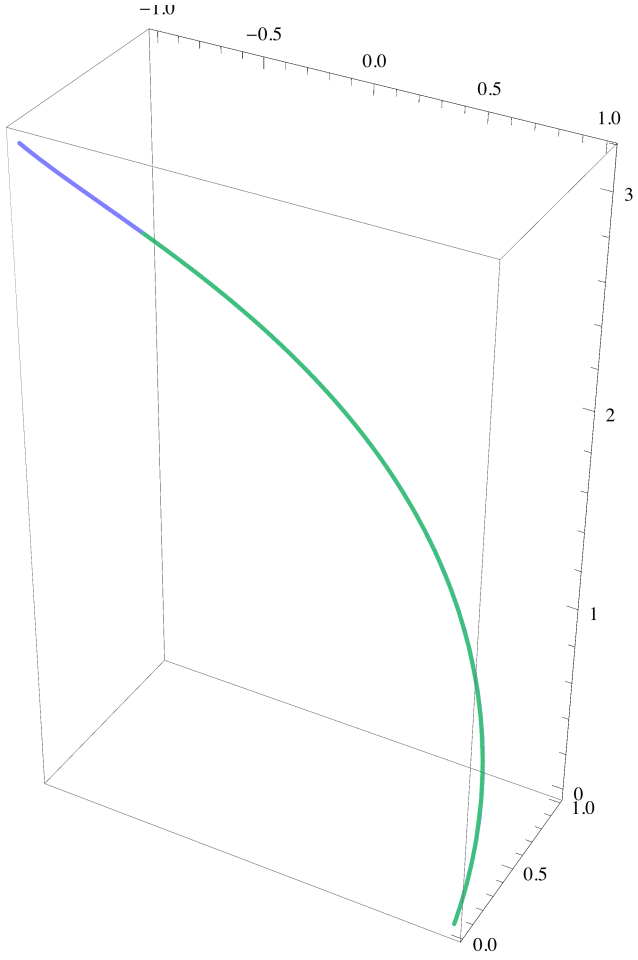
We can use the function Grid to show the two curves next to each other, but their relationship is not made clear:

```
Grid[{{p1, p2}}]
```



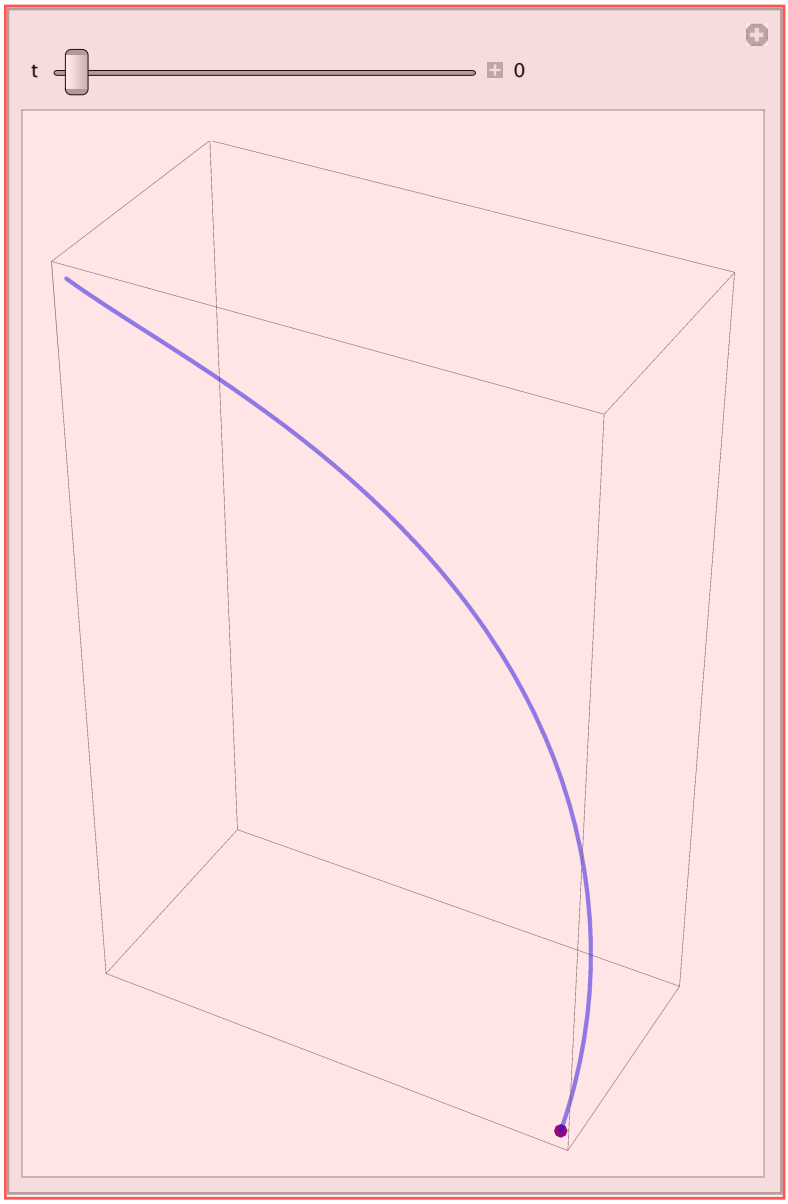
We can use Show to combine them into one picture:

```
Show[p1, p2]
```



Clearly the two parametrisations describe the same curve. We can see the dynamics of the motion. The dynamics is created using the *Mathematica* function Manipulate.

```
Manipulate[Show[Graphics3D[
    {PointSize[0.02], Red, If[t ≤ Pi, Point[r₁[t]], Point[r₁[Pi]]], Purple, Point[r₂[t]]}],
  p₁], {t, 0, 20, Appearance → "Labeled"}, SaveDefinitions → True]
```



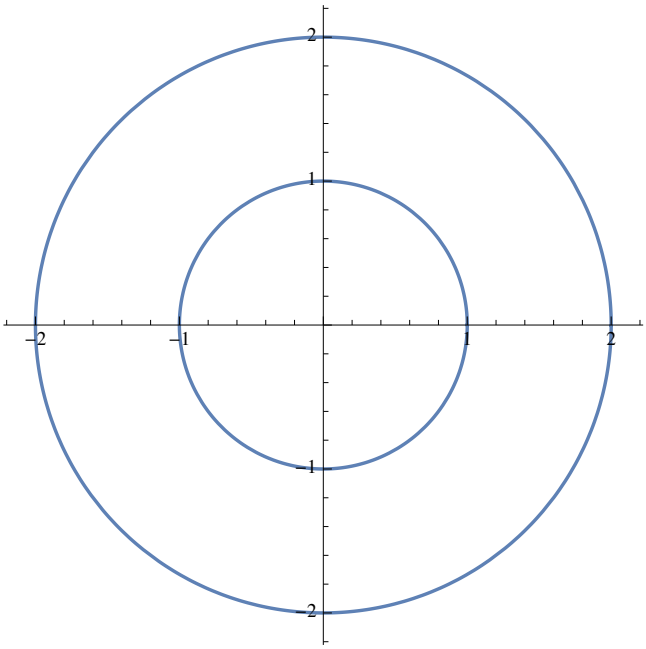
We can also use ParametricPlot for more complicted functions, e.g. piecewise defined functions

```
Piecewise
```

Piecewise[{{val_1, $cond_1$}, {val_2, $cond_2$}, …}] represents a piecewise function with values val_i in the regions defined by the conditions $cond_i$.
Piecewise[{{val_1, $cond_1$}, …}, val] uses default value val if none of the $cond_i$ apply. The default for val is 0. ≫

```
f[t_] := Piecewise[{{{2 Cos[t], 2 Sin[t]}, 0 ≤ t ≤ 2 Pi}, {{Cos[t], Sin[t]}, 2 Pi ≤ t ≤ 4 Pi}}]
```

```
ParametricPlot[f[t], {t, 0, 4 Pi}]
```




```
curveLength[g_, t_, {a_, b_}] := NIntegrate[Norm[D[g, t]], {t, a, b}]
```

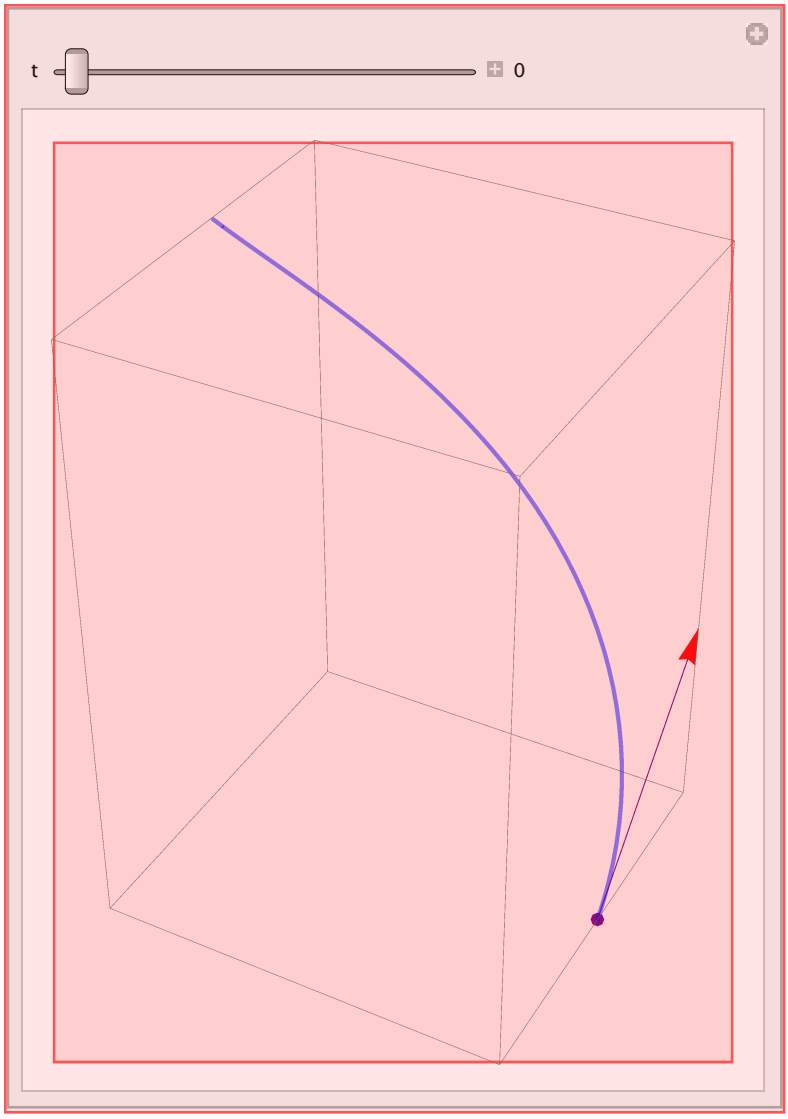
```
curveLength[r1[t], t, {0, Pi}]
```

```
4.44288
```

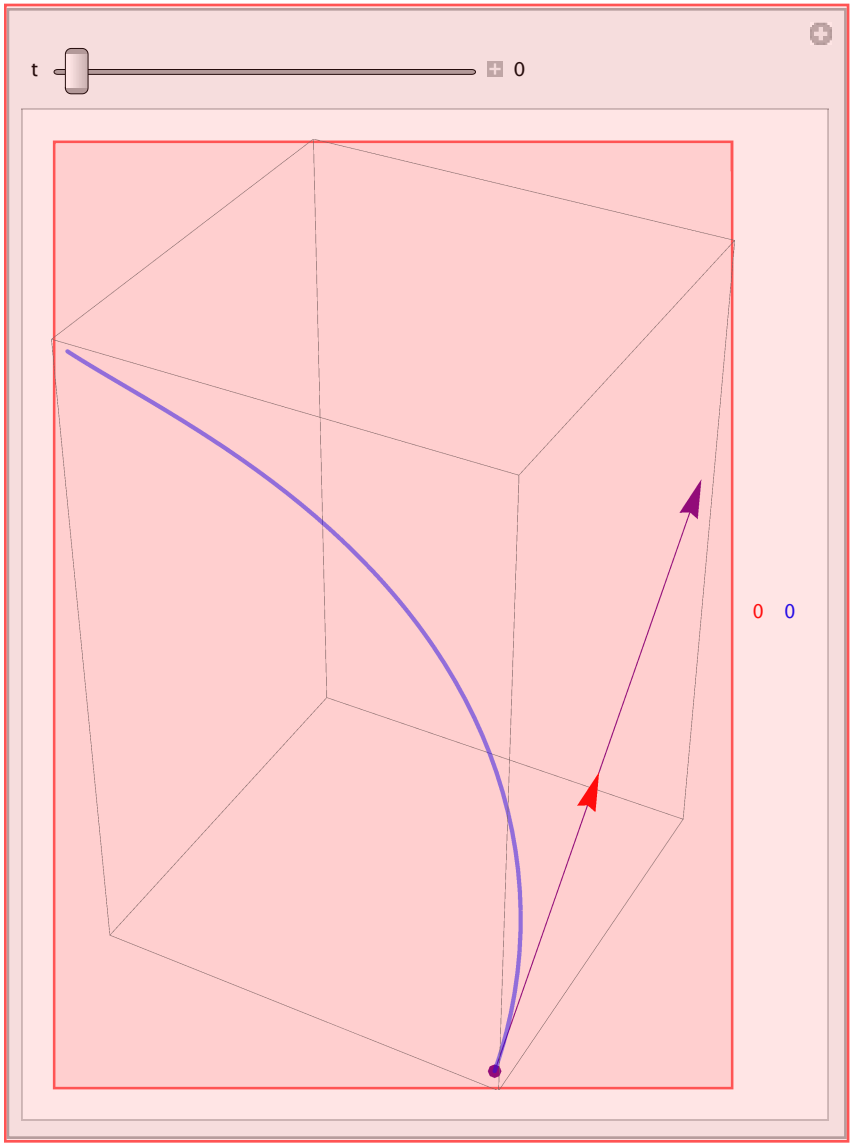
```
curveLength[r2[t], t, {0, ∞}]
```

```
4.44288
```

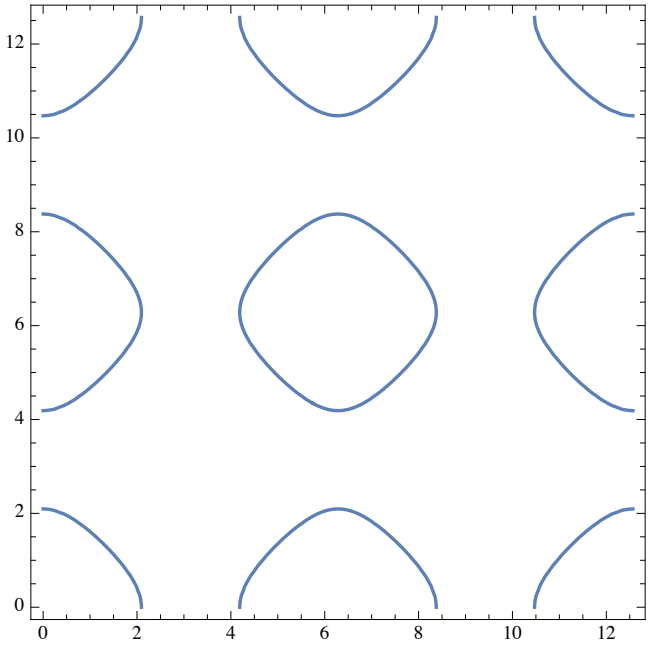
```
Manipulate[Show[Graphics3D[{PointSize[0.02], Red,
     If[t ≤ Pi, {Point[r1[t]], Arrow[{r1[t], r1[t] + r1'[t]}]}, Point[r1[Pi]]],
     Purple, Point[r2[t]], , Arrow[{r2[t], r2[t] + r2'[t]}]}],
  p1, PlotRange → {{-1, 1}, {-1, 1}, {0, 3}}],
 {t, 0, 20, Appearance → "Labeled"}, SaveDefinitions → True]
```



```
Manipulate[Grid[{{Show[Graphics3D[{PointSize[0.02], Red,
        If[t ≤ Pi, {Point[r1[t]], Arrow[{r1[t], r1[t] + r1'[t]}]}, Point[r1[Pi]]],
        Purple, Point[r2[t]], , Arrow[{r2[t], r2[t] + r2'[t]}]}], p1],
    Style[curveLength[r1[s], s, {0, Min[Pi, t]}], Red],
    Style[curveLength[r2[s], s, {0, t}], Blue]}}],
 {t, 0, 20, Appearance → "Labeled"}, SaveDefinitions → True]
```

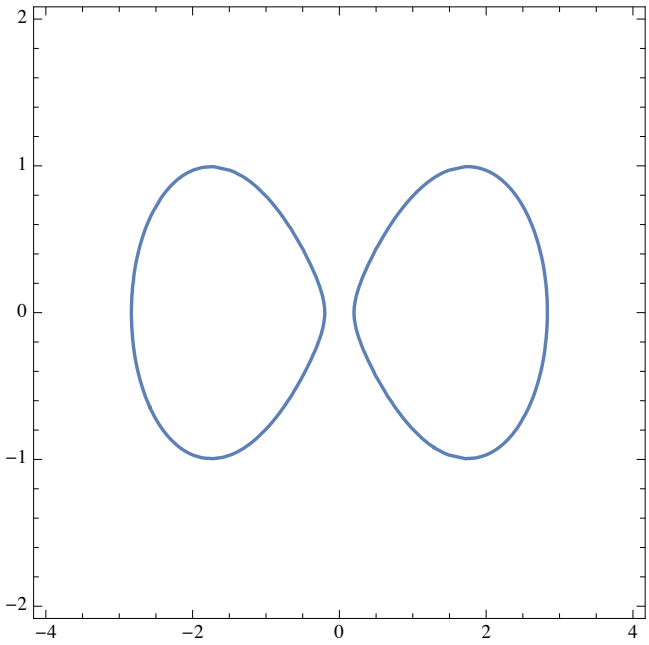


```
ContourPlot[Cos[x] + Cos[y] == 1 / 2, {x, 0, 4 Pi}, {y, 0, 4 Pi}]
```

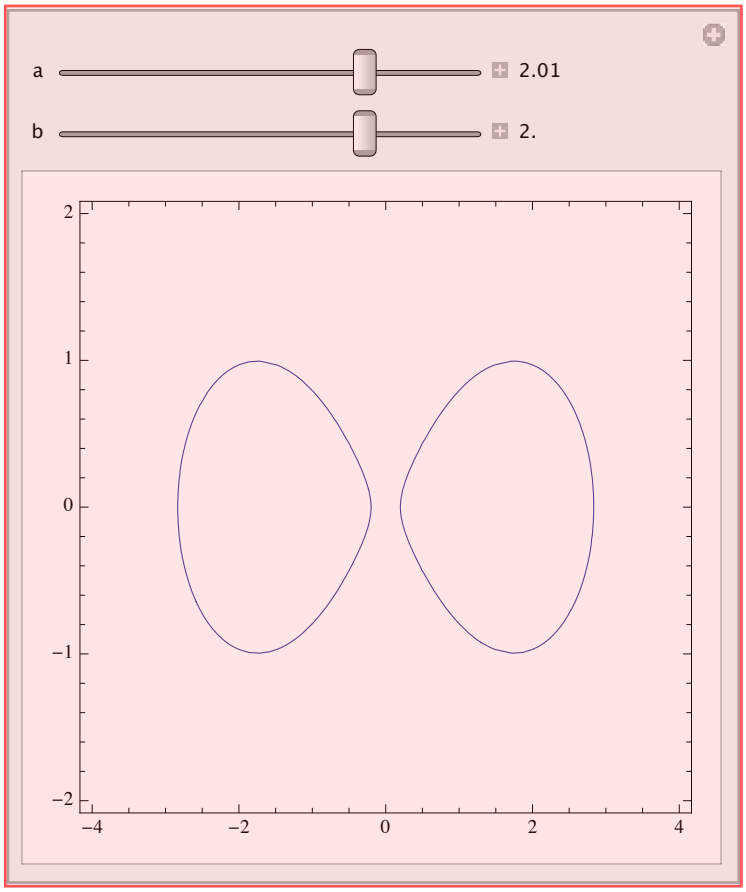



```
cassini[a_, b_][x_, y_] := (x^2 + y^2 + a^2)^2 - b^4 - 4 * a^2 * x^2
```

```
ContourPlot[cassini[2.01, 2][x, y] == 0, {x, -4, 4}, {y, -2, 2}]
```



```
Manipulate[ContourPlot[cassini[a, b][x, y] == 0, {x, -4, 4}, {y, -2, 2}],
 {{a, 2.01, "a"}, -4, 4, Appearance → "Labeled"},
 {{b, 2., "b"}, -4, 4, Appearance → "Labeled"}, SaveDefinitions → True]
```



Singular and regular points of curves

```
Clear[f]
```

For a curve

$$f(x, y) = 0$$

$$\left(f^{(1,0)}(x, y) \quad f^{(0,1)}(x, y) \right) \neq \{0, 0\}$$

For a curve

$$\{x, y\} = \{\phi(t), \psi(t)\}$$

$$\lim_{t \to 0} \frac{\frac{\partial\{\phi(t),\psi(t)\}}{\partial t}}{\left\| \frac{\partial\{\phi(t),\psi(t)\}}{\partial t} \right\|}$$

exists.