GRAPHICS

ContourPlot
Contour plot3D
ListContourPlot

DensityPlot
DensityPlot3D

VectorPlot
VectorPlot3D

ParametricPlot
ParametricPlot3D

SphericalPlot3D
SphericalHarmonicY

ContourPlot

ContourPlot creates contours of an expression involving two variables.
The contours are the curves on which the expression is constant.
The contours are drawn on a rectangle.
Ranges for each variable in the expression can be given.

ContourPlot

ContourPlot[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
generates a contour plot of f as a function of x and y.

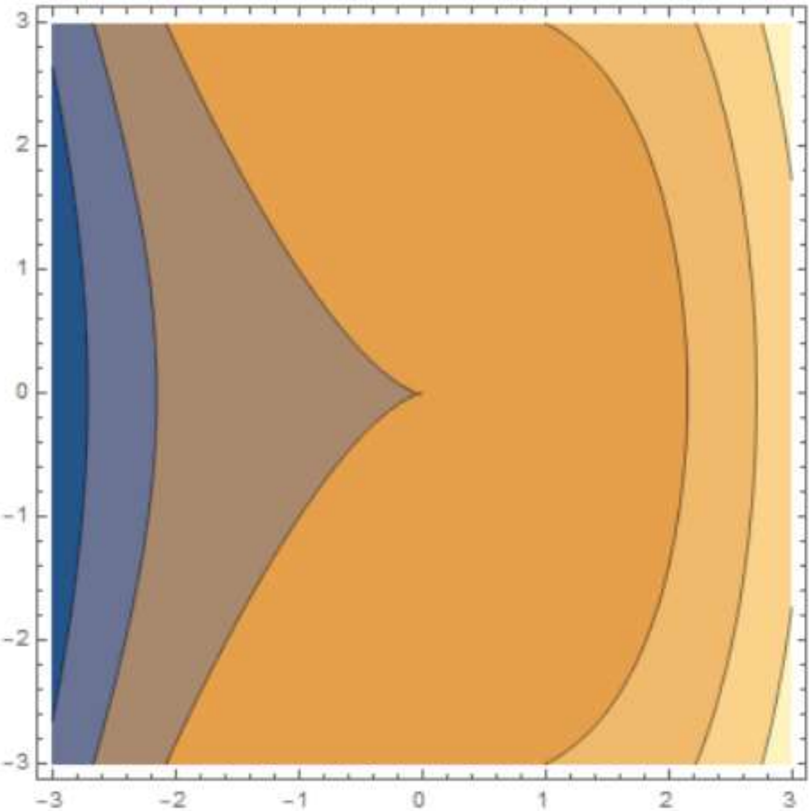
ContourPlot[f == g, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
plots contour lines for which $f = g$.

ContourPlot[{f_1 == g_1, f_2 == g_2, ...}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
plots several contour lines.

ContourPlot[..., {x, y} ∈ reg]
takes the variables {x, y} to be in the geometric region reg.

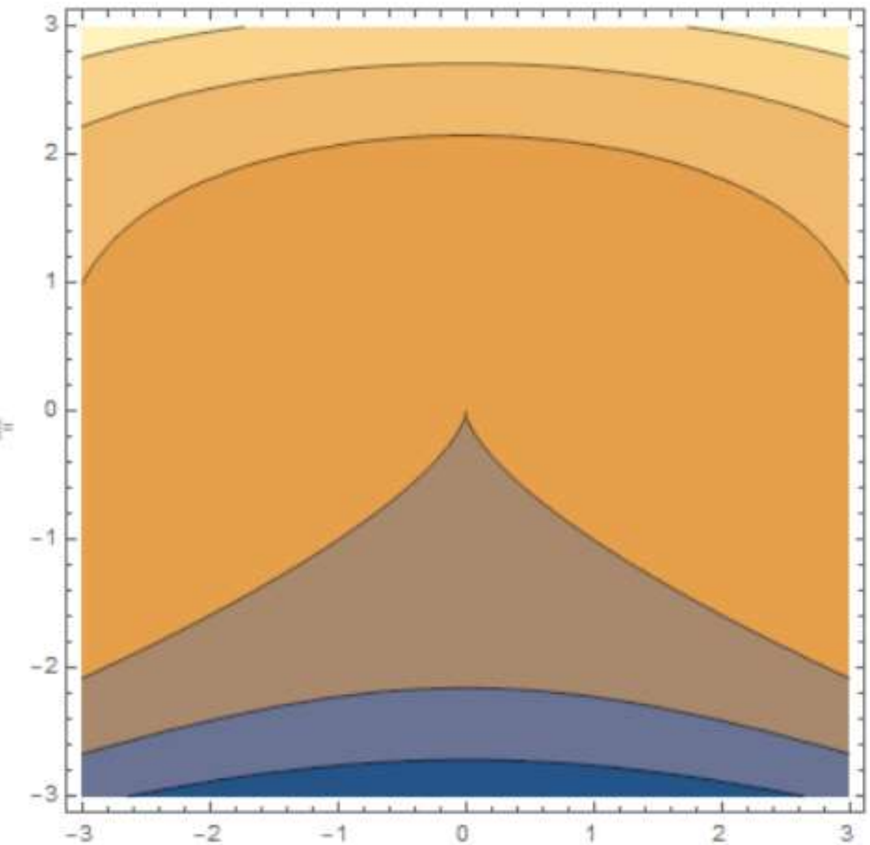
In[2]:= `ContourPlot[x^3 + y^2, {x, -3, 3}, {y, -3, 3}]`

Out[2]=



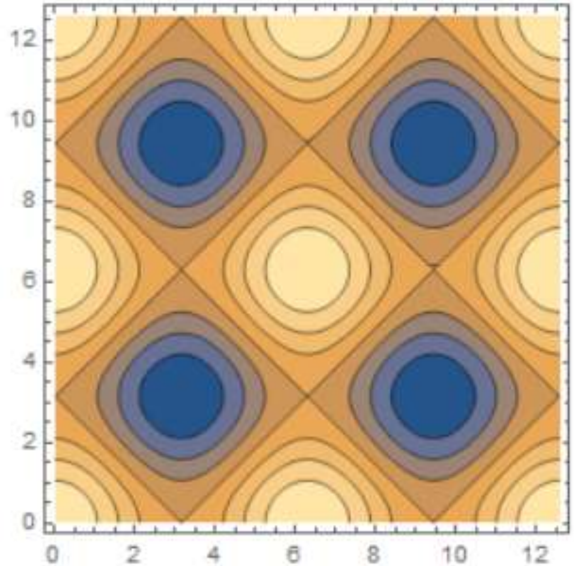
In[17]:= `ContourPlot[x^2 + y^3, {x, -3, 3}, {y, -3, 3}]`

Out[17]=



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

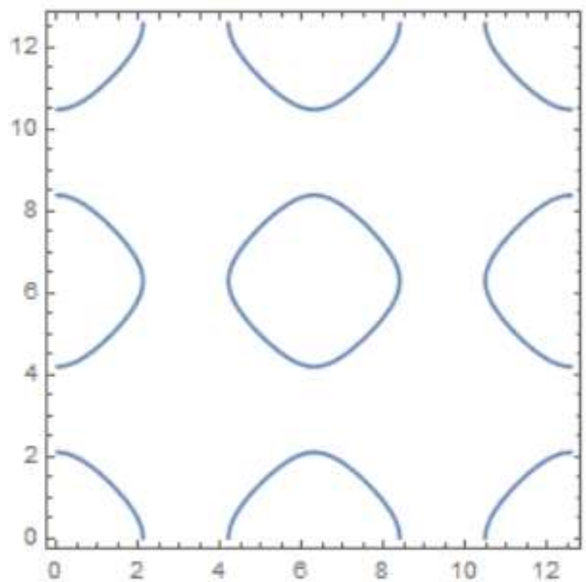
Out[1]=



Plot an equation:

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

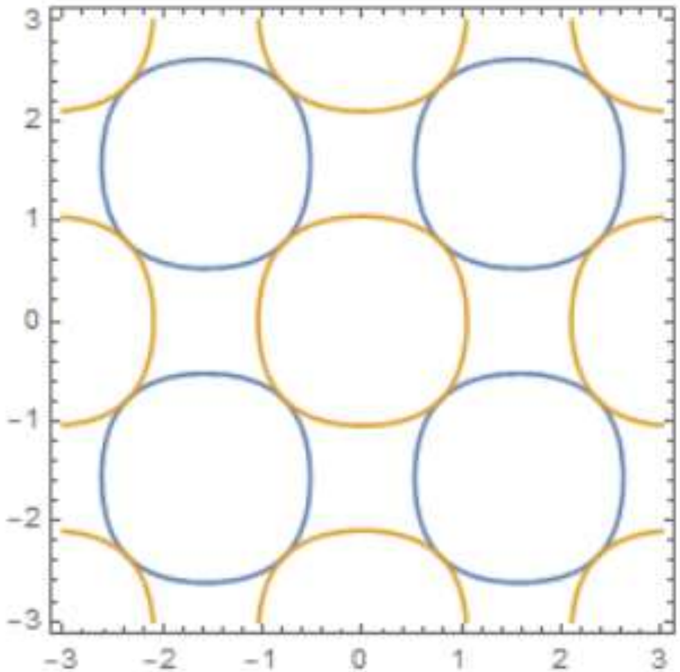
Out[1]=



Plot several equations:

```
In[1]:= ContourPlot[{Abs[Sin[x] Sin[y]] == 0.5, Abs[Cos[x] Cos[y]] == 0.5}, {x, -3, 3}, {y, -3, 3}]
```

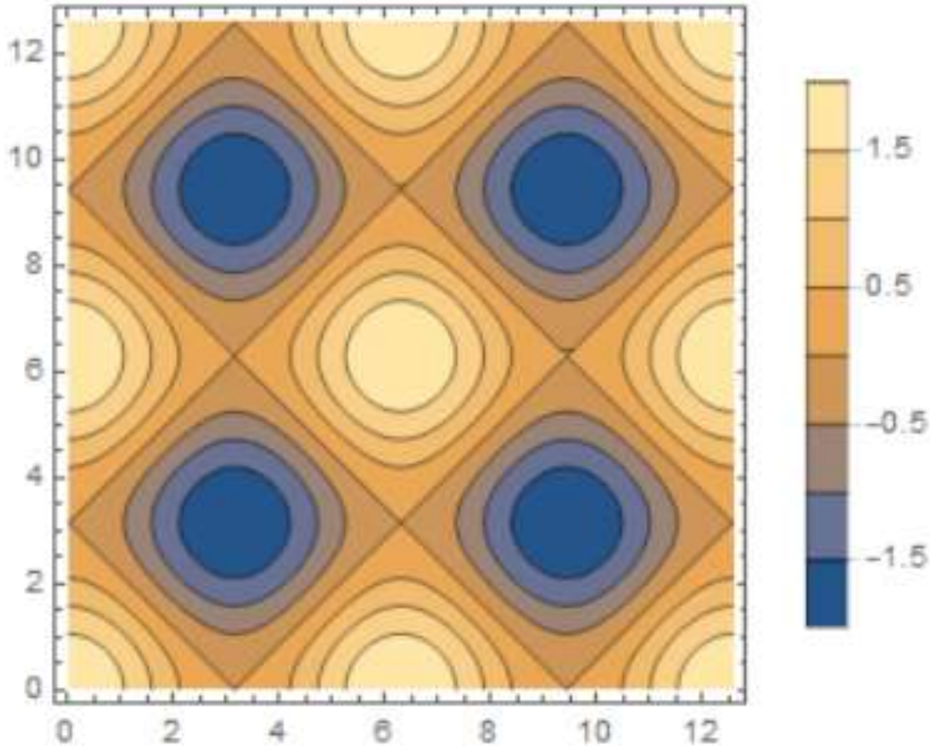
Out[1]=



Show a legend for the contours:

In[1]:= `ContourPlot[Cos[x] + Cos[y], {x, 0, 4 Pi}, {y, 0, 4 Pi}, PlotLegends → Automatic]`

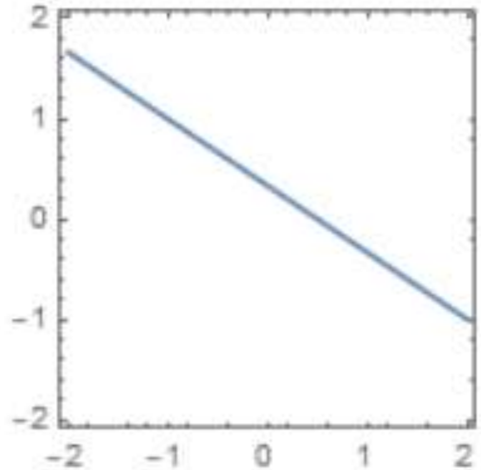
Out[1]=



Simple shapes, including a line:

```
In[1]:= ContourPlot[2 x + 3 y == 1, {x, -2, 2}, {y, -2, 2}]
```

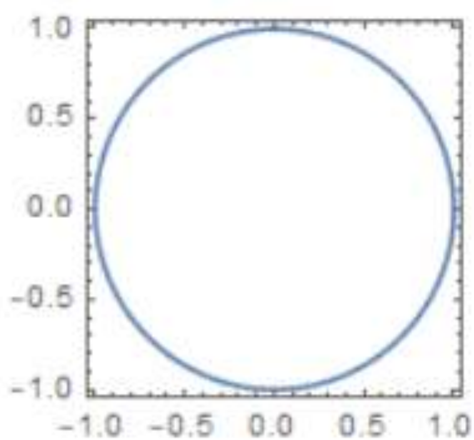
Out[1]=



Circle:

```
In[2]:= ContourPlot[x^2 + y^2 == 1, {x, -1, 1}, {y, -1, 1}]
```

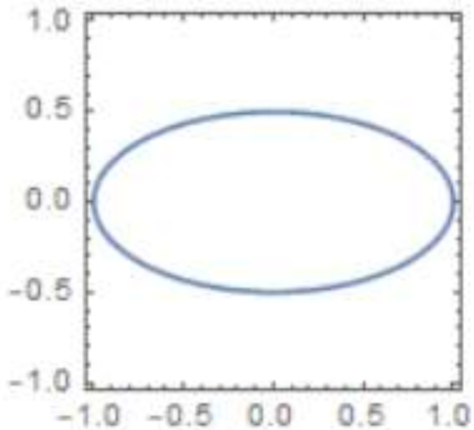
Out[2]=



Ellipse:

```
In[3]:= ContourPlot[x^2 + (2 y)^2 == 1, {x, -1, 1}, {y, -1, 1}]
```

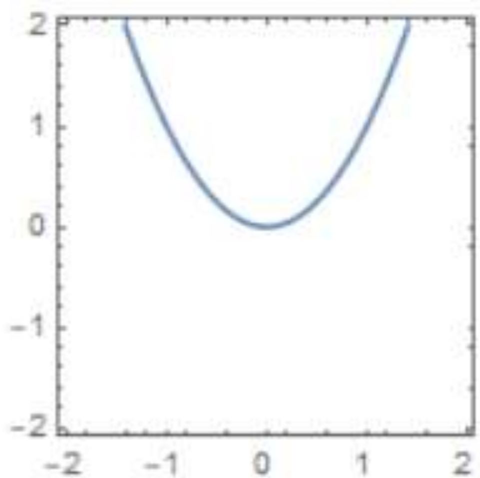
Out[3]=



Parabola:

```
In[4]:= ContourPlot[x^2 == y, {x, -2, 2}, {y, -2, 2}]
```

Out[4]=



ContourPlot3D

ContourPlot3D[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, {z, z_{min}, z_{max}}]
produces a three–dimensional contour plot of f as a function of x, y, and z.

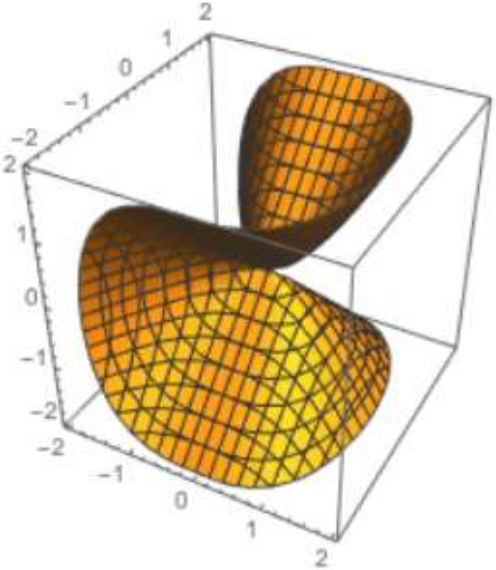
ContourPlot3D[f == g, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, {z, z_{min}, z_{max}}]
plots the contour surface for which $f = g$.

- The contour surfaces plotted by ContourPlot3D can contain disconnected parts.
- By default, ContourPlot3D shows each contour level as an opaque white surface, with normals pointing outward.

Plot a 3D contour surface:

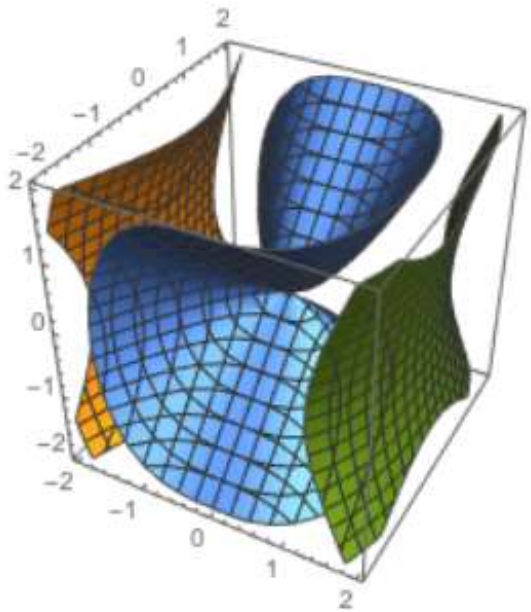
In[1]:= `ContourPlot3D[x^3 + y^2 - z^2 == 0, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]`

Out[1]=



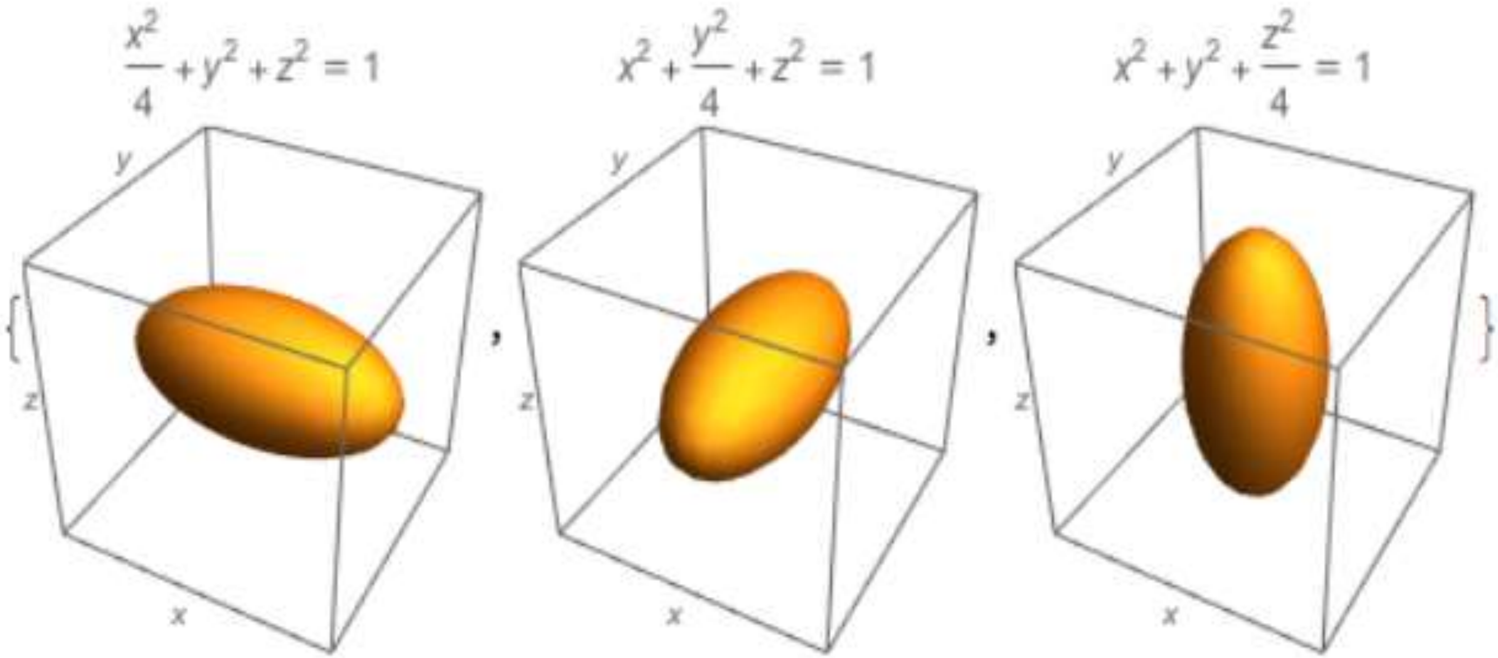
In[1]:= `ContourPlot3D[x^3 + y^2 - z^2, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]`

Out[1]=



```
In[1]:= Table[ContourPlot3D[Evaluate[e], {x, -2, 2}, {y, -2, 2},
          {z, -2, 2}, Mesh → None, AxesLabel → Automatic, Ticks → None, PlotLabel → e],
         {e, {(x / 2) ^2 + y^2 + z^2 == 1, x^2 + (y / 2) ^2 + z^2 == 1, x^2 + y^2 + (z / 2) ^2 == 1}}]
```

Out[1]=



ListContourPlot

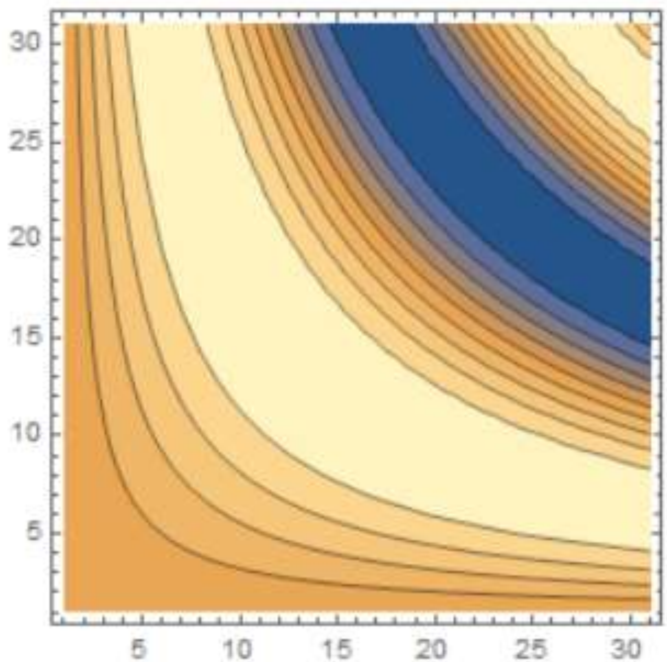
ListContourPlot[*array*]
generates a contour plot from an array of height values.

ListContourPlot[{{x_1, y_1, f_1}, {x_2, y_2, f_2}, ...}]
generates a contour plot from values defined at specified points.

```
In[1]:= data = Table[Sin[x y], {x, 0, 3, .1}, {y, 0, 3, .1}];

In[2]:= ListContourPlot[data]
```

Out[2]=




```
ListContourPlot[Table[Sin[x y], {x, 0, 3, .1}, {y, 0, 3, .1}]]
```

DensityPlot

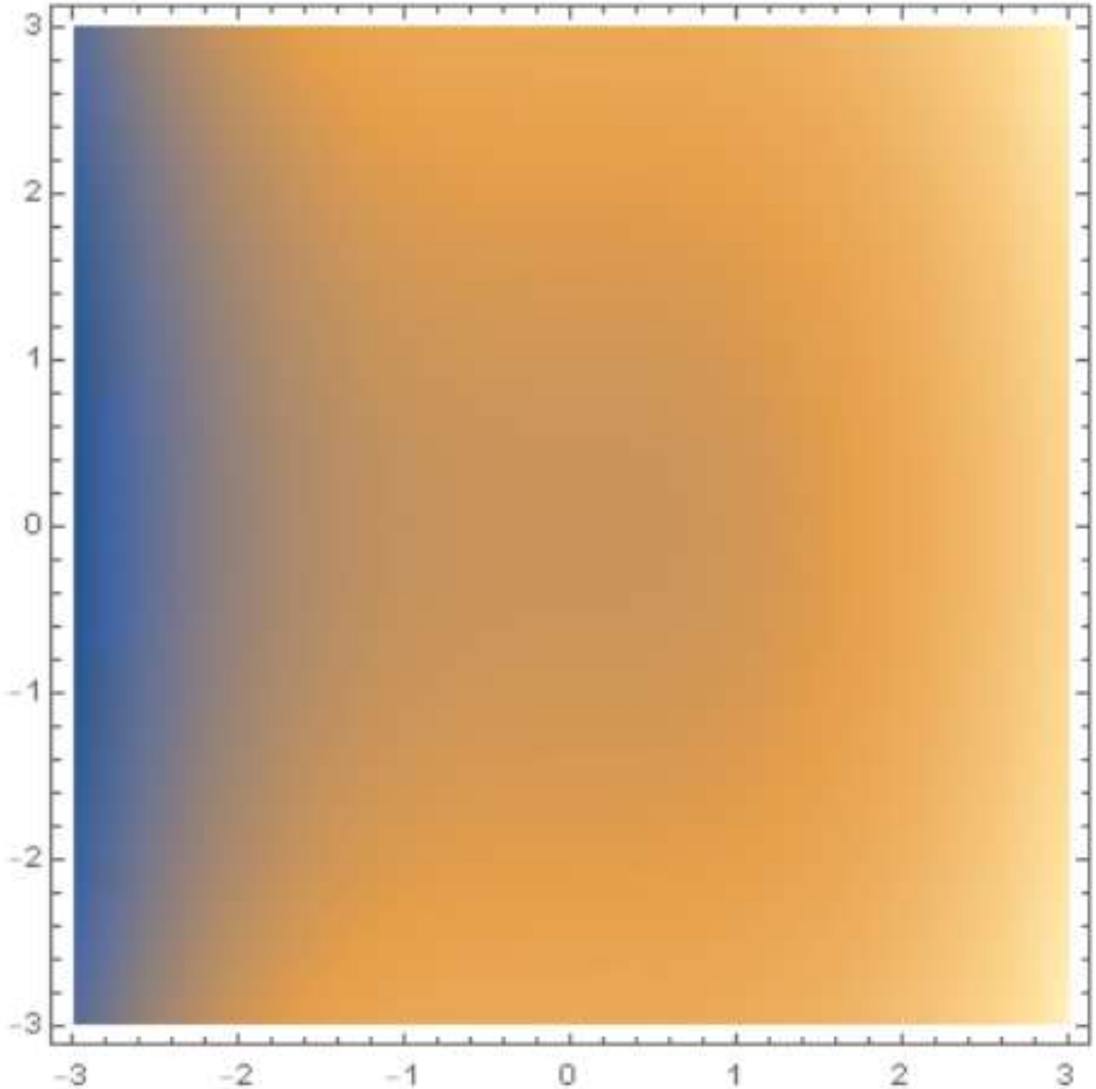
This has the syntax similar to that of the ContourPlot[ ],
but it has different options.
For an expression involving two variables,
this produces a shading of chosen rectangle.

DensityPlot

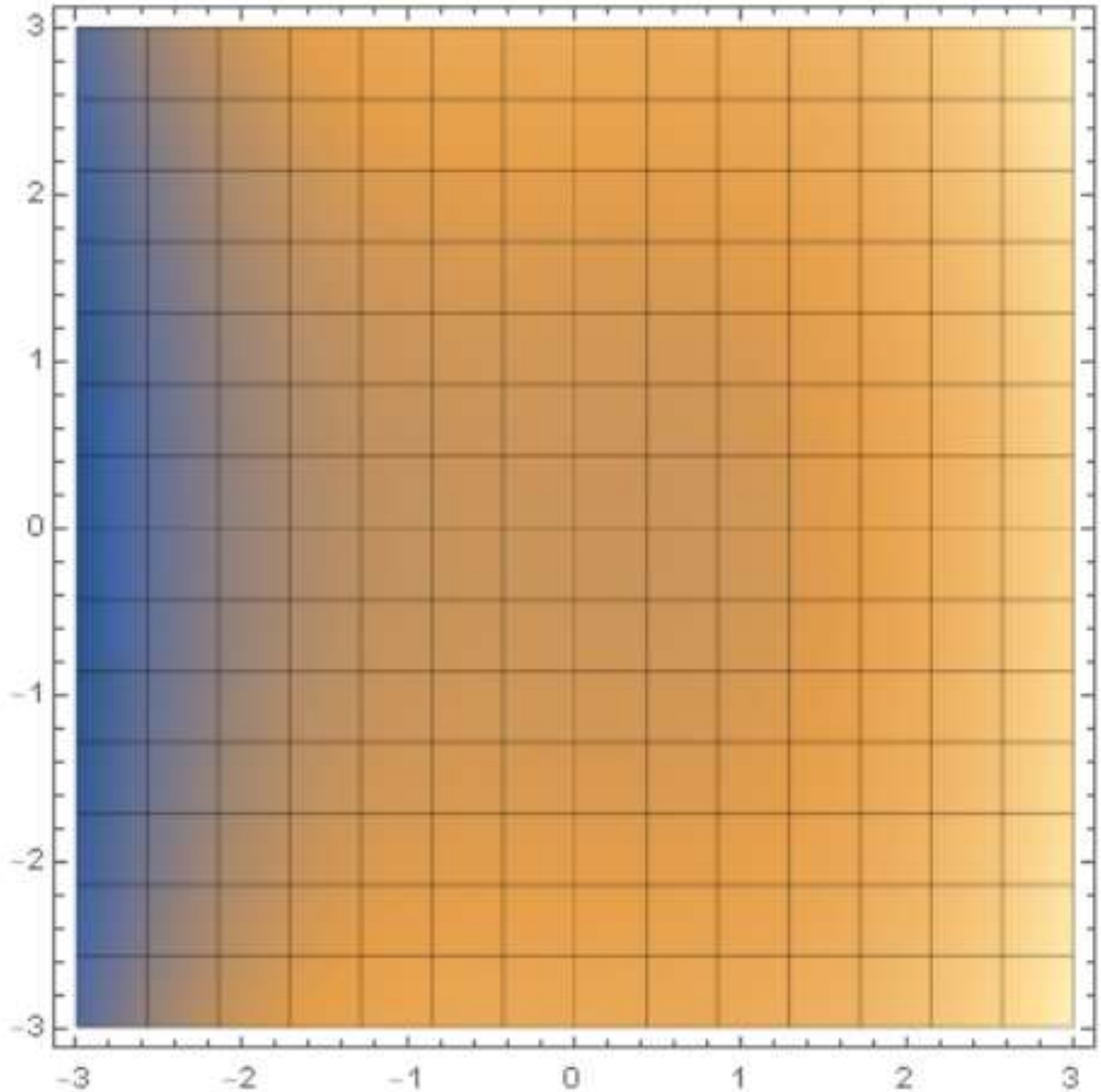
DensityPlot[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
makes a density plot of f as a function of x and y.

DensityPlot[f, {x, y} ∈ reg]
takes the variables {x, y} to be in the geometric region reg.

```
DensityPlot[x^3 + y^2, {x, -3, 3}, {y, -3, 3}]
```



```
DensityPlot[x^3 + y^2, {x, -3, 3}, {y, -3, 3}, Mesh → Full]
```



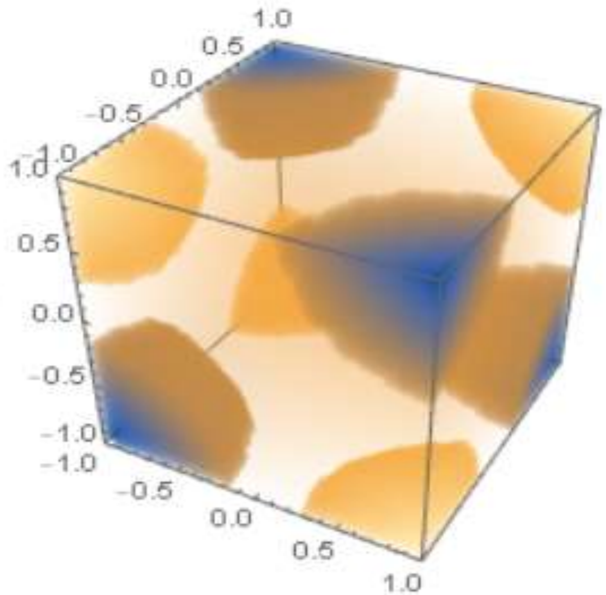
DensityPlot3D

DensityPlot3D[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, {z, z_{min}, z_{max}}]
makes a density plot of f as a function of x, y, and z.

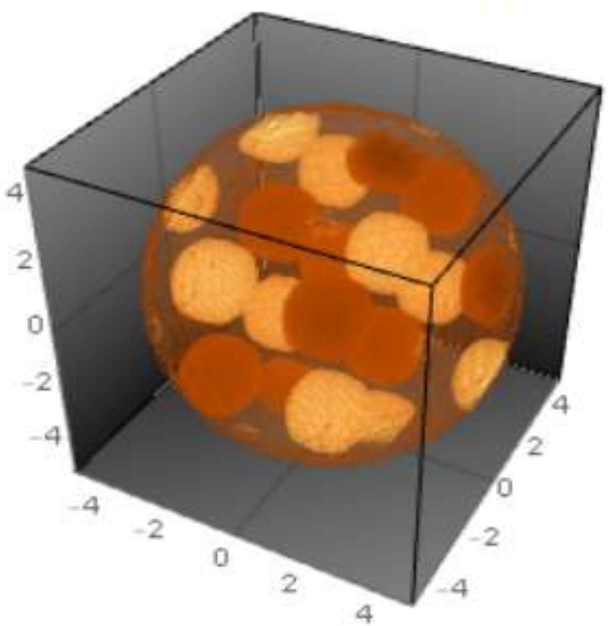
DensityPlot3D[f, {x, y, z} $\in$ *reg*]
takes the variables to be in the geometric region *reg*.

- DensityPlot3D by default generates colorized output.
- At positions where f does not evaluate to a real number, data is taken to be missing and is rendered transparently.
- DensityPlot3D treats the variables x, y, and z as local, effectively using Block.
- DensityPlot3D has attribute HoldAll, and evaluates f only after assigning specific numerical values to x, y, and z.

```
DensityPlot3D[x y z, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```




```
DensityPlot3D[Sin[x] Cos[y] Sin[z], {x, y, z} ∈ Ball[{0, 0, 0}, 5], PlotTheme → "Marketing"]
```



VectorPlot

VectorPlot[{v_x, v_y}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
generates a vector plot of the vector field {v_x, v_y} as a function of x and y.

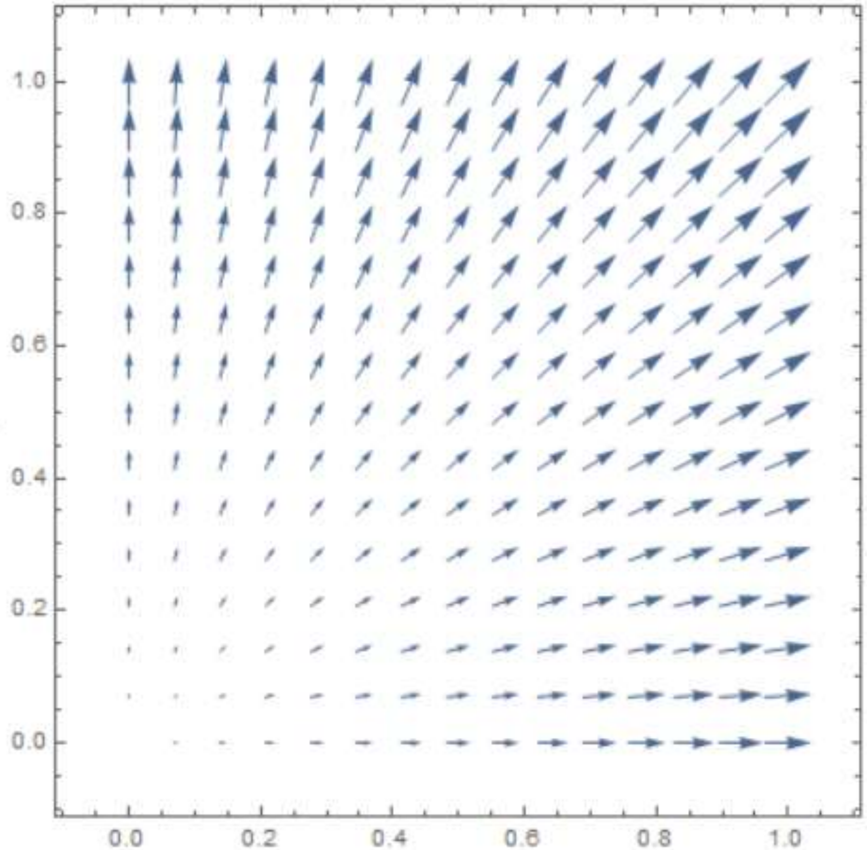
VectorPlot[{{v_x, v_y}, {w_x, w_y}, ...}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}]
plots several vector fields.

VectorPlot[..., {x, y} ∈ *reg*]
takes the variables {x, y} to be in the geometric region *reg*.

- VectorPlot by default shows vectors from the vector field at a regular grid of positions.
- VectorPlot omits any vectors for which the v_i etc. do not evaluate to real numbers.
- VectorPlot treats the variables x and y as local, effectively using Block.

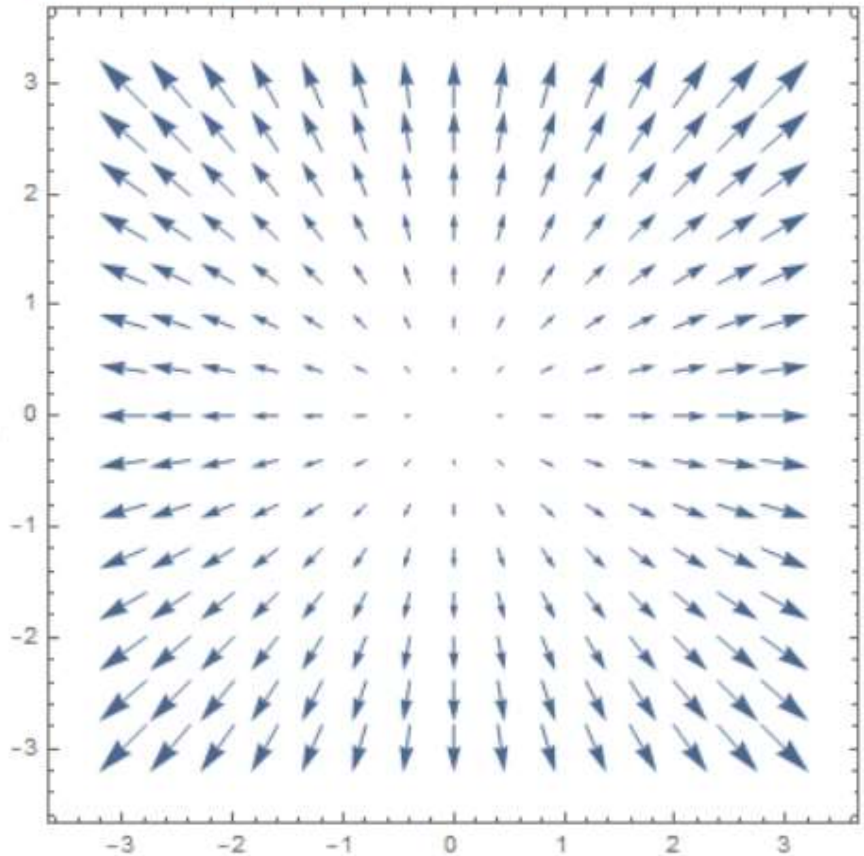
In[22]:= `VectorPlot[{x, y}, {x, 0, 1}, {y, 0, 1}]`

Out[22]=



```
In[34]:= VectorPlot[{x, y}, {x, -3, 3}, {y, -3, 3}]
```

Out[34]=

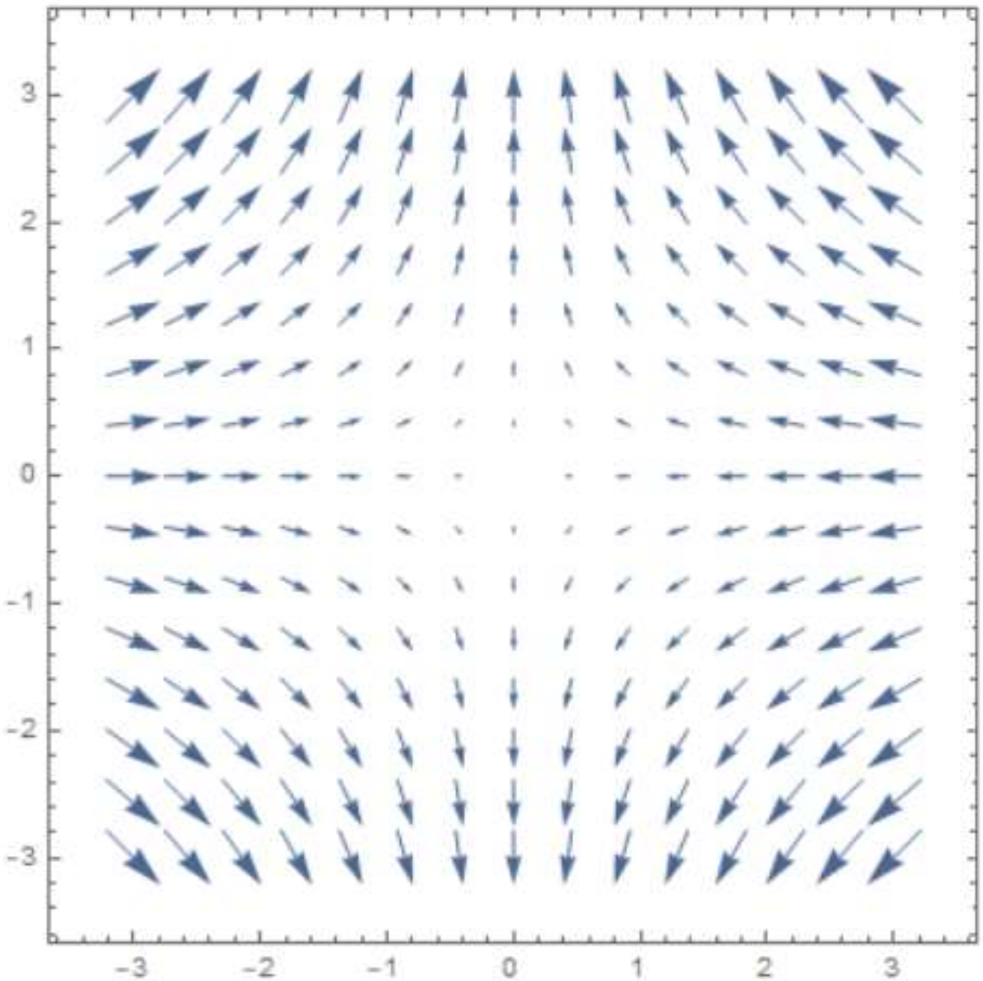



```
Table[{x, y}, {x, -3, 3}, {y, -3, 3}]; MatrixForm[%]
```

itrixForm=

$$\begin{pmatrix} \begin{pmatrix}-3\\-3\end{pmatrix} & \begin{pmatrix}-3\\-2\end{pmatrix} & \begin{pmatrix}-3\\-1\end{pmatrix} & \begin{pmatrix}-3\\0\end{pmatrix} & \begin{pmatrix}-3\\1\end{pmatrix} & \begin{pmatrix}-3\\2\end{pmatrix} & \begin{pmatrix}-3\\3\end{pmatrix} \\ \begin{pmatrix}-2\\-3\end{pmatrix} & \begin{pmatrix}-2\\-2\end{pmatrix} & \begin{pmatrix}-2\\-1\end{pmatrix} & \begin{pmatrix}-2\\0\end{pmatrix} & \begin{pmatrix}-2\\1\end{pmatrix} & \begin{pmatrix}-2\\2\end{pmatrix} & \begin{pmatrix}-2\\3\end{pmatrix} \\ \begin{pmatrix}-1\\-3\end{pmatrix} & \begin{pmatrix}-1\\-2\end{pmatrix} & \begin{pmatrix}-1\\-1\end{pmatrix} & \begin{pmatrix}-1\\0\end{pmatrix} & \begin{pmatrix}-1\\1\end{pmatrix} & \begin{pmatrix}-1\\2\end{pmatrix} & \begin{pmatrix}-1\\3\end{pmatrix} \\ \begin{pmatrix}0\\-3\end{pmatrix} & \begin{pmatrix}0\\-2\end{pmatrix} & \begin{pmatrix}0\\-1\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}0\\2\end{pmatrix} & \begin{pmatrix}0\\3\end{pmatrix} \\ \begin{pmatrix}1\\-3\end{pmatrix} & \begin{pmatrix}1\\-2\end{pmatrix} & \begin{pmatrix}1\\-1\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}1\\1\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\3\end{pmatrix} \\ \begin{pmatrix}2\\-3\end{pmatrix} & \begin{pmatrix}2\\-2\end{pmatrix} & \begin{pmatrix}2\\-1\end{pmatrix} & \begin{pmatrix}2\\0\end{pmatrix} & \begin{pmatrix}2\\1\end{pmatrix} & \begin{pmatrix}2\\2\end{pmatrix} & \begin{pmatrix}2\\3\end{pmatrix} \\ \begin{pmatrix}3\\-3\end{pmatrix} & \begin{pmatrix}3\\-2\end{pmatrix} & \begin{pmatrix}3\\-1\end{pmatrix} & \begin{pmatrix}3\\0\end{pmatrix} & \begin{pmatrix}3\\1\end{pmatrix} & \begin{pmatrix}3\\2\end{pmatrix} & \begin{pmatrix}3\\3\end{pmatrix} \end{pmatrix}$$

```
VectorPlot[{-x, y}, {x, -3, 3}, {y, -3, 3}]
```

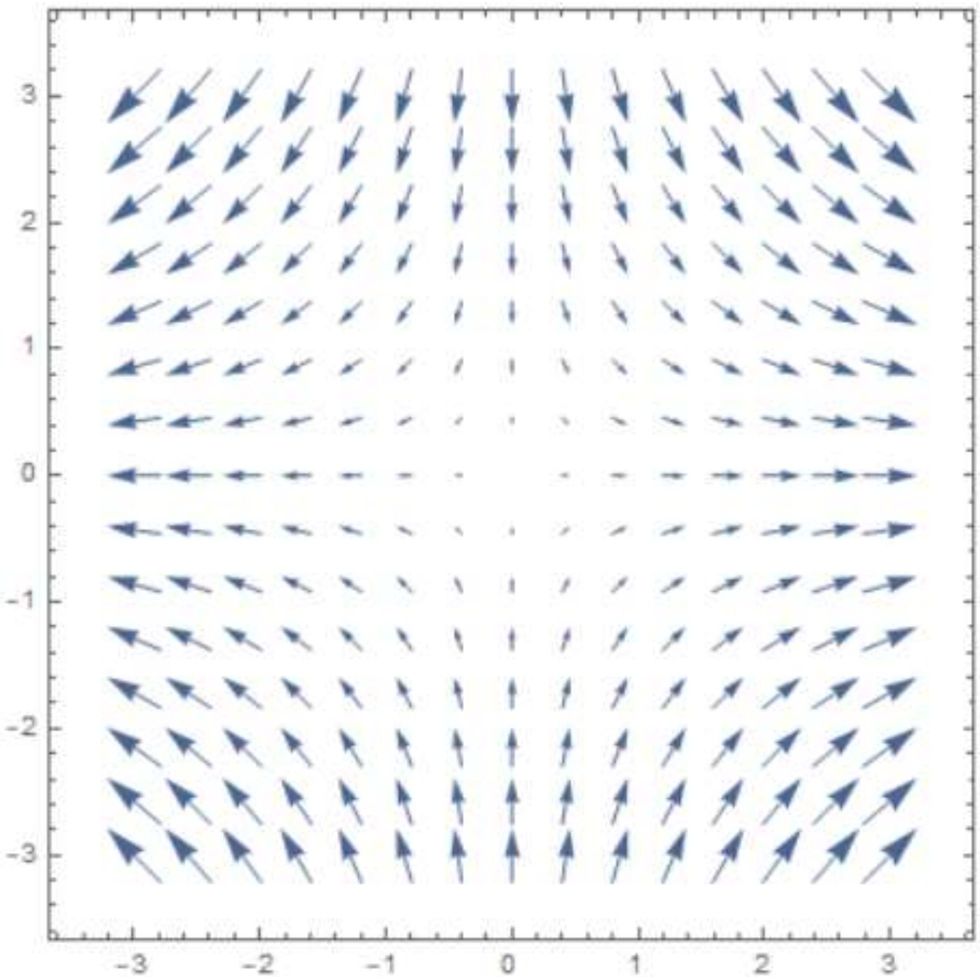



```
Table[{-x, y}, {x, -3, 3}, {y, -3, 3}]; MatrixForm[%]
```

trixForm=

$$\begin{pmatrix} \begin{pmatrix}3\\-3\end{pmatrix} & \begin{pmatrix}3\\-2\end{pmatrix} & \begin{pmatrix}3\\-1\end{pmatrix} & \begin{pmatrix}3\\0\end{pmatrix} & \begin{pmatrix}3\\1\end{pmatrix} & \begin{pmatrix}3\\2\end{pmatrix} & \begin{pmatrix}3\\3\end{pmatrix} \\ \begin{pmatrix}2\\-3\end{pmatrix} & \begin{pmatrix}2\\-2\end{pmatrix} & \begin{pmatrix}2\\-1\end{pmatrix} & \begin{pmatrix}2\\0\end{pmatrix} & \begin{pmatrix}2\\1\end{pmatrix} & \begin{pmatrix}2\\2\end{pmatrix} & \begin{pmatrix}2\\3\end{pmatrix} \\ \begin{pmatrix}1\\-3\end{pmatrix} & \begin{pmatrix}1\\-2\end{pmatrix} & \begin{pmatrix}1\\-1\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}1\\1\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\3\end{pmatrix} \\ \begin{pmatrix}0\\-3\end{pmatrix} & \begin{pmatrix}0\\-2\end{pmatrix} & \begin{pmatrix}0\\-1\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}0\\2\end{pmatrix} & \begin{pmatrix}0\\3\end{pmatrix} \\ \begin{pmatrix}-1\\-3\end{pmatrix} & \begin{pmatrix}-1\\-2\end{pmatrix} & \begin{pmatrix}-1\\-1\end{pmatrix} & \begin{pmatrix}-1\\0\end{pmatrix} & \begin{pmatrix}-1\\1\end{pmatrix} & \begin{pmatrix}-1\\2\end{pmatrix} & \begin{pmatrix}-1\\3\end{pmatrix} \\ \begin{pmatrix}-2\\-3\end{pmatrix} & \begin{pmatrix}-2\\-2\end{pmatrix} & \begin{pmatrix}-2\\-1\end{pmatrix} & \begin{pmatrix}-2\\0\end{pmatrix} & \begin{pmatrix}-2\\1\end{pmatrix} & \begin{pmatrix}-2\\2\end{pmatrix} & \begin{pmatrix}-2\\3\end{pmatrix} \\ \begin{pmatrix}-3\\-3\end{pmatrix} & \begin{pmatrix}-3\\-2\end{pmatrix} & \begin{pmatrix}-3\\-1\end{pmatrix} & \begin{pmatrix}-3\\0\end{pmatrix} & \begin{pmatrix}-3\\1\end{pmatrix} & \begin{pmatrix}-3\\2\end{pmatrix} & \begin{pmatrix}-3\\3\end{pmatrix} \end{pmatrix}$$

```
VectorPlot[{x, -y}, {x, -3, 3}, {y, -3, 3}]
```




```
Table[{x, -y}, {x, -3, 3}, {y, -3, 3}]; MatrixForm[%]
```

trixForm=

$$\begin{pmatrix} \begin{pmatrix} -3 \\ 3 \end{pmatrix} & \begin{pmatrix} -3 \\ 2 \end{pmatrix} & \begin{pmatrix} -3 \\ 1 \end{pmatrix} & \begin{pmatrix} -3 \\ 0 \end{pmatrix} & \begin{pmatrix} -3 \\ -1 \end{pmatrix} & \begin{pmatrix} -3 \\ -2 \end{pmatrix} & \begin{pmatrix} -3 \\ -3 \end{pmatrix} \\ \begin{pmatrix} -2 \\ 3 \end{pmatrix} & \begin{pmatrix} -2 \\ 2 \end{pmatrix} & \begin{pmatrix} -2 \\ 1 \end{pmatrix} & \begin{pmatrix} -2 \\ 0 \end{pmatrix} & \begin{pmatrix} -2 \\ -1 \end{pmatrix} & \begin{pmatrix} -2 \\ -2 \end{pmatrix} & \begin{pmatrix} -2 \\ -3 \end{pmatrix} \\ \begin{pmatrix} -1 \\ 3 \end{pmatrix} & \begin{pmatrix} -1 \\ 2 \end{pmatrix} & \begin{pmatrix} -1 \\ 1 \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1 \\ -1 \end{pmatrix} & \begin{pmatrix} -1 \\ -2 \end{pmatrix} & \begin{pmatrix} -1 \\ -3 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 3 \end{pmatrix} & \begin{pmatrix} 0 \\ 2 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} & \begin{pmatrix} 0 \\ -2 \end{pmatrix} & \begin{pmatrix} 0 \\ -3 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 3 \end{pmatrix} & \begin{pmatrix} 1 \\ 2 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ -1 \end{pmatrix} & \begin{pmatrix} 1 \\ -2 \end{pmatrix} & \begin{pmatrix} 1 \\ -3 \end{pmatrix} \\ \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 2 \\ 2 \end{pmatrix} & \begin{pmatrix} 2 \\ 1 \end{pmatrix} & \begin{pmatrix} 2 \\ 0 \end{pmatrix} & \begin{pmatrix} 2 \\ -1 \end{pmatrix} & \begin{pmatrix} 2 \\ -2 \end{pmatrix} & \begin{pmatrix} 2 \\ -3 \end{pmatrix} \\ \begin{pmatrix} 3 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 2 \end{pmatrix} & \begin{pmatrix} 3 \\ 1 \end{pmatrix} & \begin{pmatrix} 3 \\ 0 \end{pmatrix} & \begin{pmatrix} 3 \\ -1 \end{pmatrix} & \begin{pmatrix} 3 \\ -2 \end{pmatrix} & \begin{pmatrix} 3 \\ -3 \end{pmatrix} \end{pmatrix}$$

VectorPlot3D

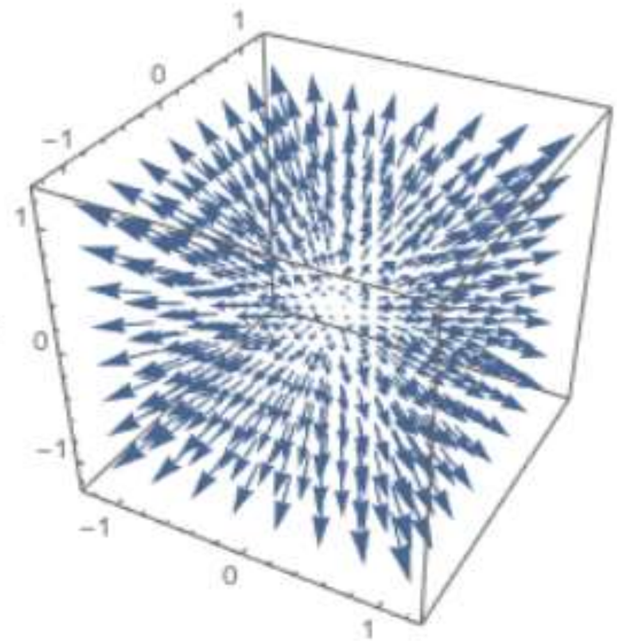
VectorPlot3D[{v_x, v_y, v_z}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, {z, z_{min}, z_{max}}]
generates a 3D vector plot of the vector field {v_x, v_y, v_z} as a function of x, y, and z.

VectorPlot3D[{$field_1$, $field_2$, ...}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, {z, z_{min}, z_{max}}]
plots several vector fields.

Plot a vector field:

```
In[1]:= VectorPlot3D[{x, y, z}, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```

Out[1]=



ParametricPlot

ParametricPlot[ ] draws the curve formed by a pair of expression

{x[t] , y[t]} as the parameter t varies.

ParametricPlot[{f_x, f_y}, {u, u_{min}, u_{max}}]
generates a parametric plot of a curve with x and y coordinates f_x and f_y as a function of u.

ParametricPlot[{{f_x, f_y}, {g_x, g_y}, ...}, {u, u_{min}, u_{max}}]
plots several parametric curves.

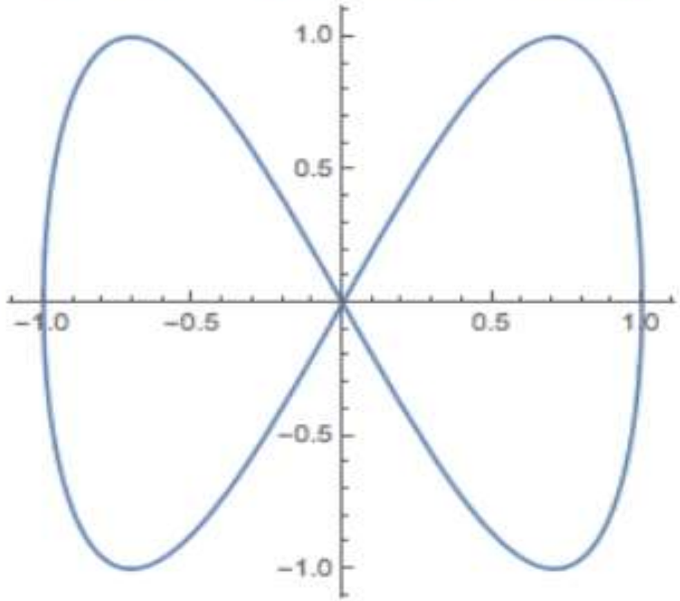
ParametricPlot[{f_x, f_y}, {u, u_{min}, u_{max}}, {v, v_{min}, v_{max}}]
plots a parametric region.

ParametricPlot[{{f_x, f_y}, {g_x, g_y}, ...}, {u, u_{min}, u_{max}}, {v, v_{min}, v_{max}}]
plots several parametric regions.

ParametricPlot[..., {u, v} ∈ reg]
takes parameters {u, v} to be in the geometric region reg.

In[1]:= `ParametricPlot[{Sin[u], Sin[2 u]}, {u, 0, 2 Pi}]`

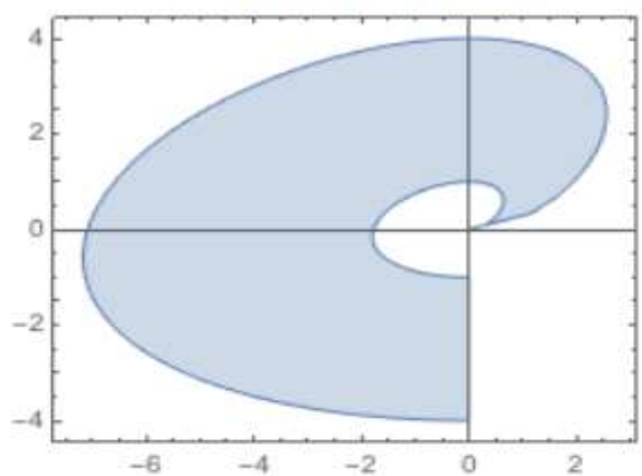
Out[1]=



Plot a parametric region:

In[1]:= `ParametricPlot[r^2 { Sqrt[t] Cos[t], Sin[t]}, {t, 0, 3 Pi / 2}, {r, 1, 2}]`

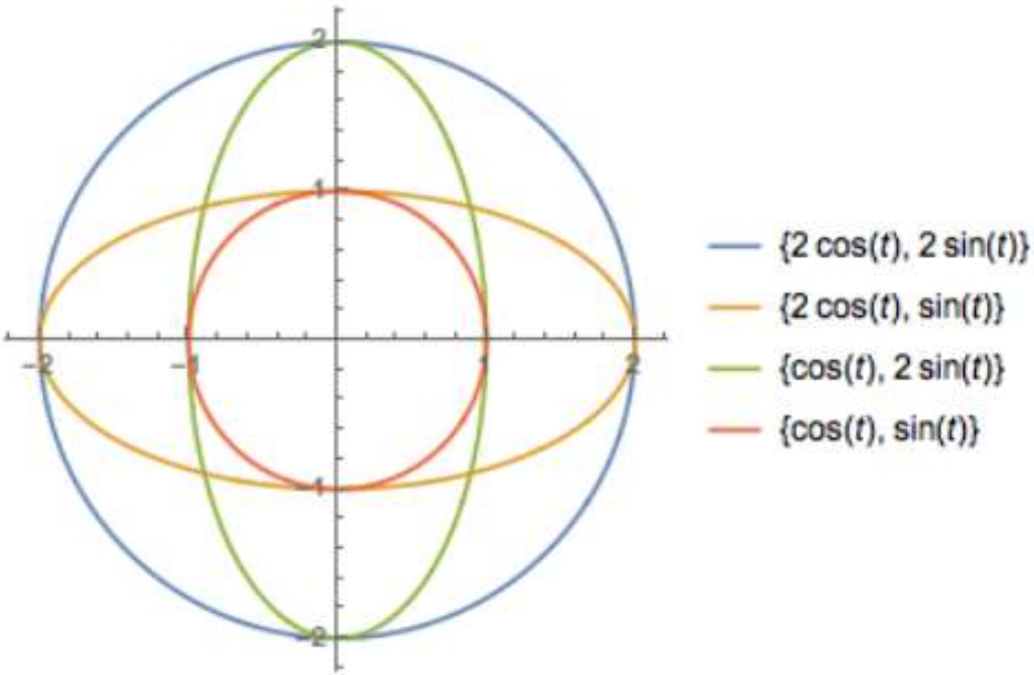
Out[1]=



Plot several parametric curves with a legend:

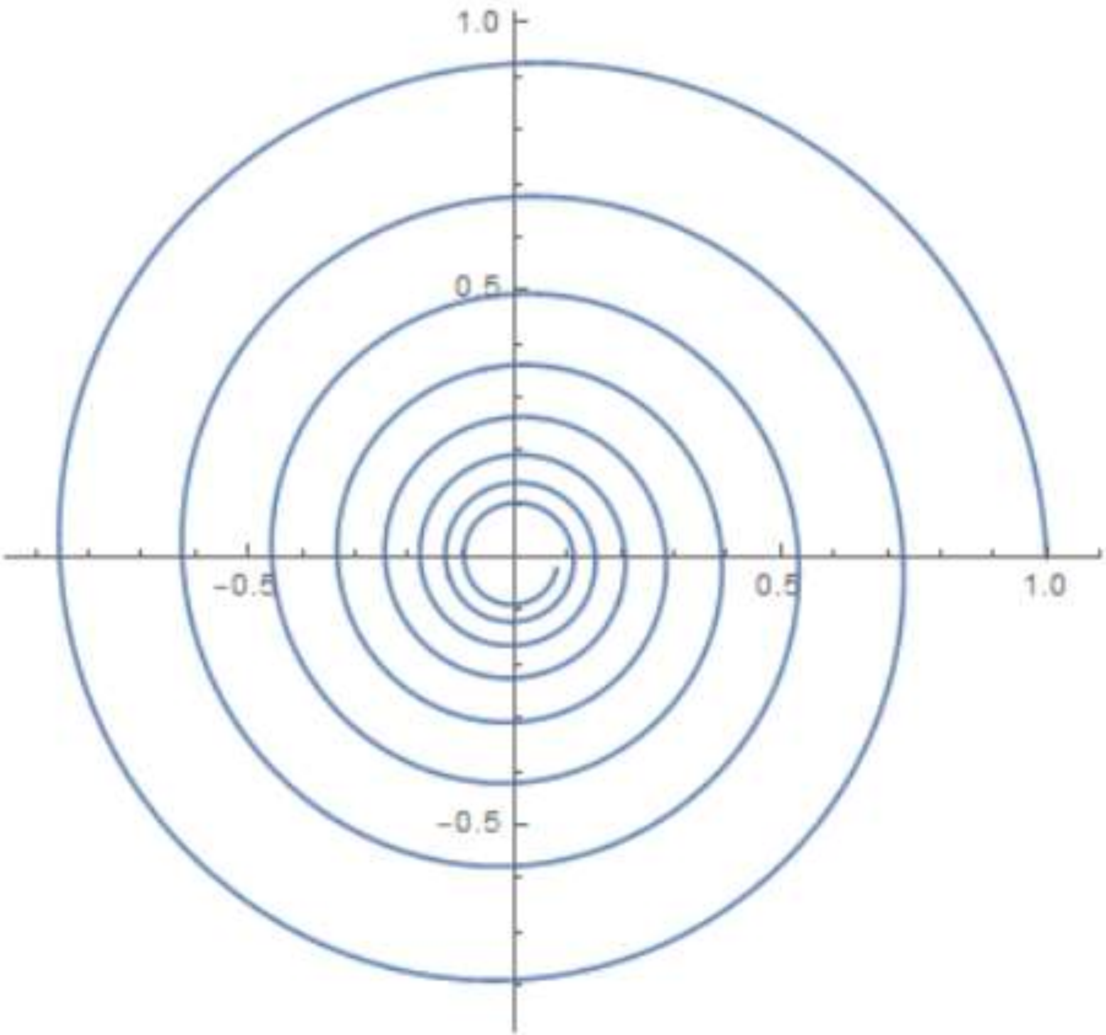
```
In[1]:= ParametricPlot[{{2 Cos[t], 2 Sin[t]}, {2 Cos[t], Sin[t]}, {Cos[t], 2 Sin[t]}, {Cos[t], Sin[t]}},
          {t, 0, 2 Pi}, PlotLegends → "Expressions"]
```

Out[1]=



In[58]:=
```
ParametricPlot[{Exp[-t / 20] Cos[t], Exp[-t / 20] Sin[t]}, {t, 0, 50}]
```

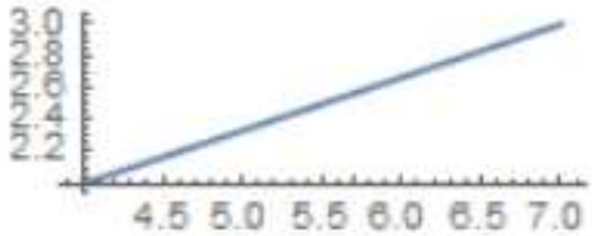
Out[58]=



Simple parametric curves including a line:

```
In[1]:= ParametricPlot[{3 u + 4, u + 2}, {u, 0, 1}]
```

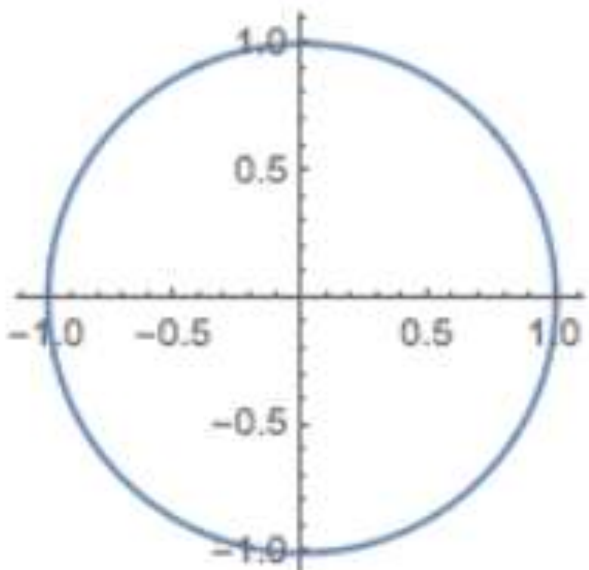
Out[1]=



Circle:

```
In[2]:= ParametricPlot[{Cos[u], Sin[u]}, {u, 0, 2 Pi}]
```

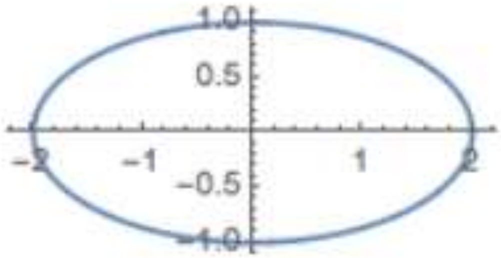
Out[2]=



Ellipse:

```
In[3]:= ParametricPlot[{2 Cos[u], Sin[u]}, {u, 0, 2 Pi}]
```

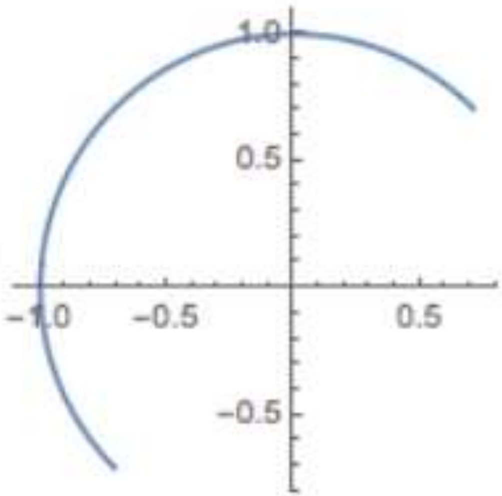
Out[3]=



Circle segment:

```
In[4]:= ParametricPlot[{Cos[u], Sin[u]}, {u, Pi / 4, 5 Pi / 4}]
```

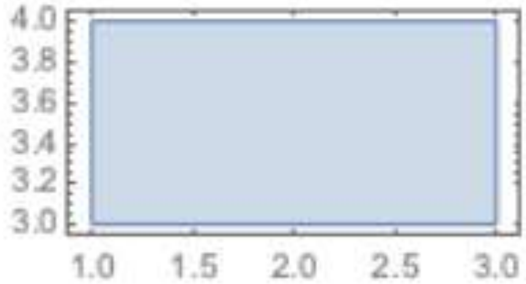
Out[4]=



Simple parametric regions including a rectangle:

```
In[1]:= ParametricPlot[{u, v}, {u, 1, 3}, {v, 3, 4}]
```

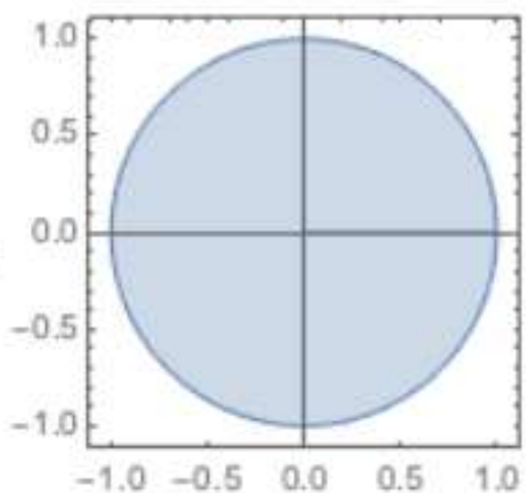
Out[1]=



Disk:

```
In[2]:= ParametricPlot[{v Cos[u], v Sin[u]}, {u, 0, 2 Pi}, {v, 0, 1}]
```

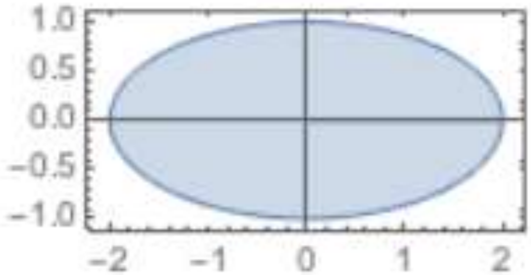
Out[2]=



Ellipse:

```
In[3]:= ParametricPlot[{2 v Cos[u], v Sin[u]}, {u, 0, 2 Pi}, {v, 0, 1}]
```

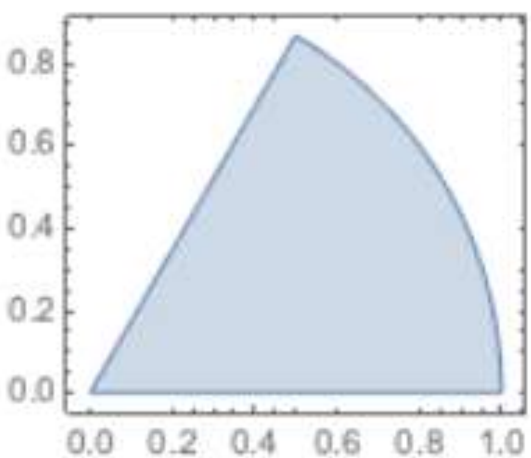
Out[3]=



Disk segment:

```
In[4]:= ParametricPlot[{v Cos[u], v Sin[u]}, {u, 0, Pi / 3}, {v, 0, 1}]
```

Out[4]=



ParametricPlot3D

This plots a parametrically defined three-dimensional curve (or a parametrically defined three-dimensional surface). It is part of the collection of Graphics packages, which must be loaded explicitly before it can be used,

ParametricPlot3D[{f_x, f_y, f_z}, {u, u_{min}, u_{max}}]

produces a three-dimensional space curve parametrized by a variable u which runs from u_{min} to u_{max}.

ParametricPlot3D[{f_x, f_y, f_z}, {u, u_{min}, u_{max}}, {v, v_{min}, v_{max}}]

produces a three-dimensional surface parametrized by u and v.

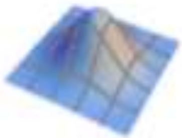
ParametricPlot3D[{{f_x, f_y, f_z}, {g_x, g_y, g_z} …} …]

plots several objects together.

ParametricPlot3D[…, {u, v} ∈ *reg*]

takes parameters {u, v} to be in the geometric region *reg*.

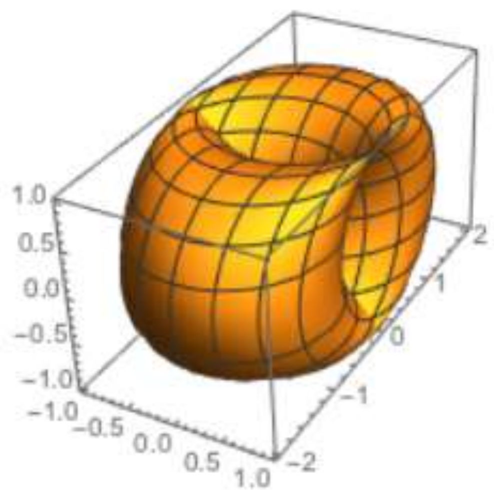
Themes that affect 3D surfaces include:

	"DarkMesh"	dark mesh lines
	"GrayMesh"	gray mesh lines
	"LightMesh"	light mesh lines
	"ZMesh"	vertical mesh lines
	"ThickSurface"	add thickness to surfaces

Plot a parametric surface:

```
In[1]:= ParametricPlot3D[{Cos[u], Sin[u] + Cos[v], Sin[v]}, {u, 0, 2 π}, {v, -π, π}]
```

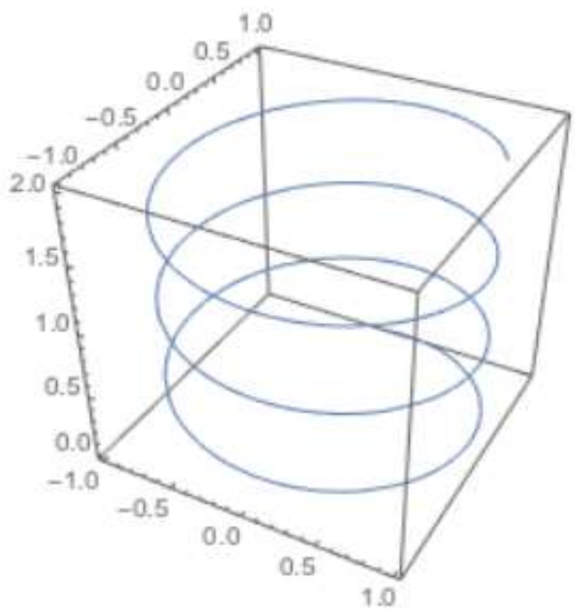
Out[1]=



Plot a parametric space curve:

```
In[1]:= ParametricPlot3D[{Sin[u], Cos[u], u / 10}, {u, 0, 20}]
```

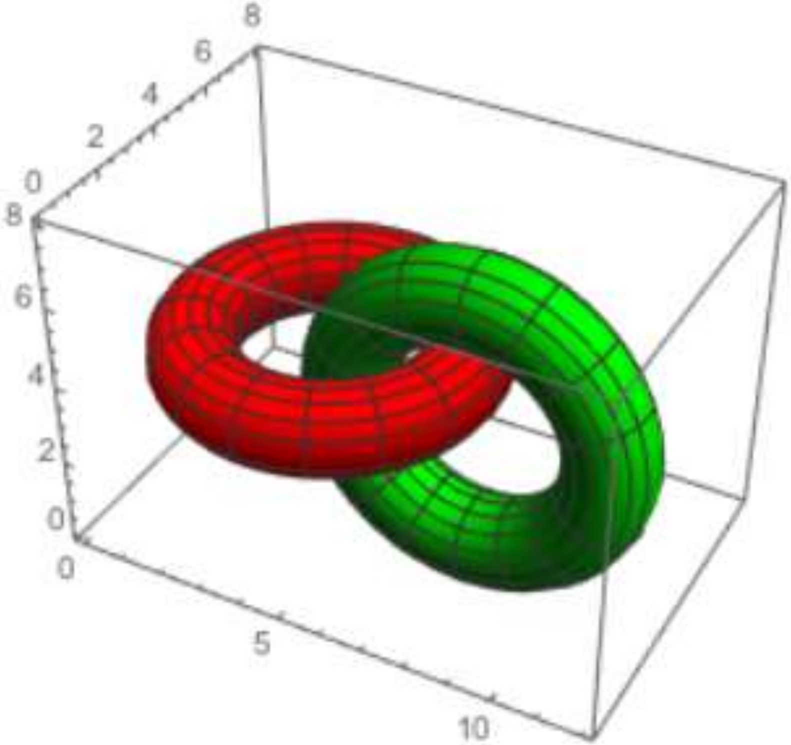
Out[1]=



Plot multiple parametric surfaces:

```
In[1]:= ParametricPlot3D[{{4 + (3 + Cos[v]) Sin[u], 4 + (3 + Cos[v]) Cos[u], 4 + Sin[v]},
          {8 + (3 + Cos[v]) Cos[u], 3 + Sin[v], 4 + (3 + Cos[v]) Sin[u]}},
         {u, 0, 2 Pi}, {v, 0, 2 Pi}, PlotStyle → {Red, Green}]
```

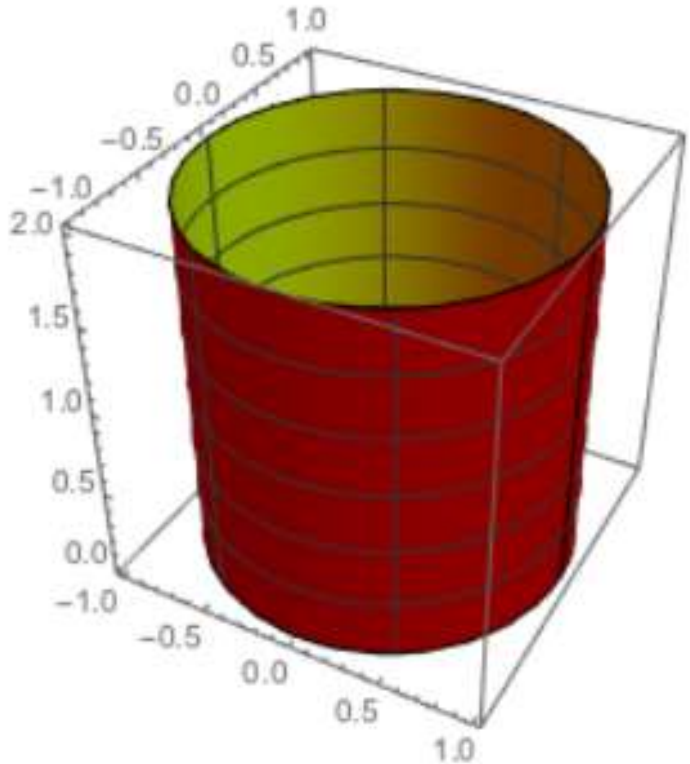
Out[1]=



Cylinder:

In[3]:=
```
ParametricPlot3D[{Cos[u], Sin[u], 2 v}, {u, 0, 2 Pi}, {v, 0, 1},
 Mesh → 5, BoundaryStyle → Black, PlotStyle → FaceForm[Red, Yellow]]
```

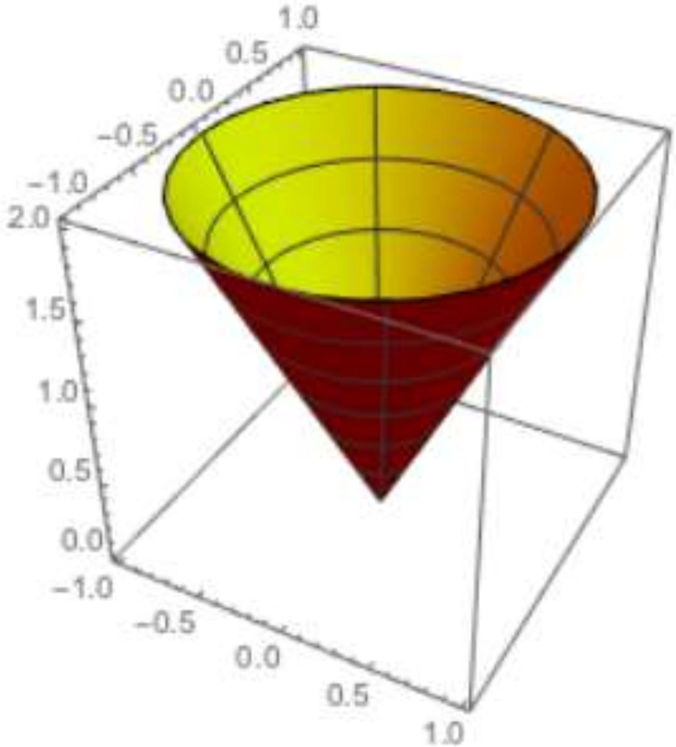
Out[3]=



Cone:

```
In[4]:= ParametricPlot3D[{v Cos[u], v Sin[u], 2 v}, {u, 0, 2 Pi},
  {v, 0, 1}, Mesh → 5, BoundaryStyle → Black, PlotStyle → FaceForm[Red, Yellow]]
```

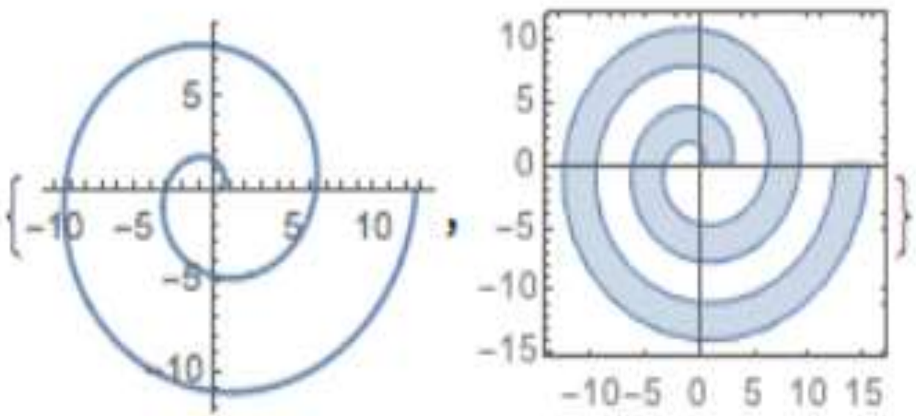
Out[4]=



Use ParametricPlot for curves and regions in two dimensions:

```
In[1]:= {ParametricPlot[{u Cos[u], u Sin[u]}, {u, 0, 4 Pi}],
         ParametricPlot[{(u + v) Cos[u], (u + v) Sin[u]}, {u, 0, 4 Pi}, {v, 0, 3}]}
```

Out[1]=



SphericalPlot3D

SphericalPlot3D[r, θ, ϕ]
generates a 3D plot with a spherical radius r as a function of spherical coordinates θ and ϕ.

SphericalPlot3D[r, {θ, θ_{min}, θ_{max}}, {ϕ, ϕ_{min}, ϕ_{max}}]
generates a 3D spherical plot over the specified ranges of spherical coordinates.

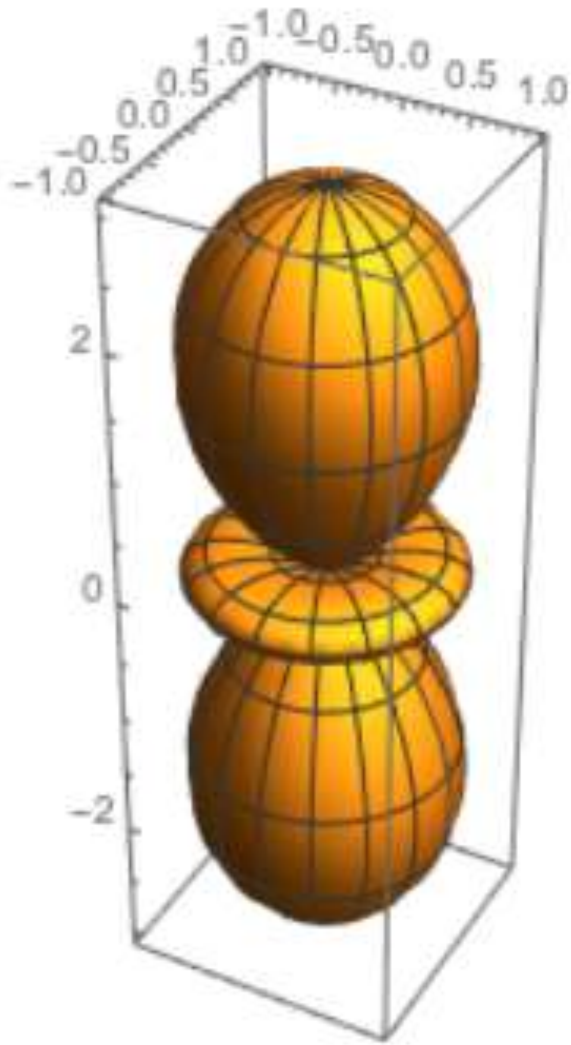
SphericalPlot3D[{r_1, r_2, …}, {θ, θ_{min}, θ_{max}}, {ϕ, ϕ_{min}, ϕ_{max}}]
generates a 3D spherical plot with multiple surfaces.

- The angles θ and ϕ are measured in radians.
- $\pi/2-\theta$ corresponds to "latitude"; θ is 0 at the "north pole", and π at the "south pole".
- ϕ corresponds to "longitude", varying from 0 to 2π counterclockwise looking from the north pole.
- SphericalPlot3D[r, θ, ϕ] takes θ to have range 0 to π, and ϕ to have range 0 to 2π.
- The x, y, z position corresponding to r, θ, ϕ is $r\sin(\theta)\cos(\phi)$, $r\sin(\theta)\sin(\phi)$, $r\cos(\theta)$. The variables θ and ϕ can have any values. The surfaces they define can overlap radially.

Plot a spherical surface:

```
In[1]:= SphericalPlot3D[1 + 2 Cos[2 θ], {θ, 0, Pi}, {ϕ, 0, 2 Pi}]
```

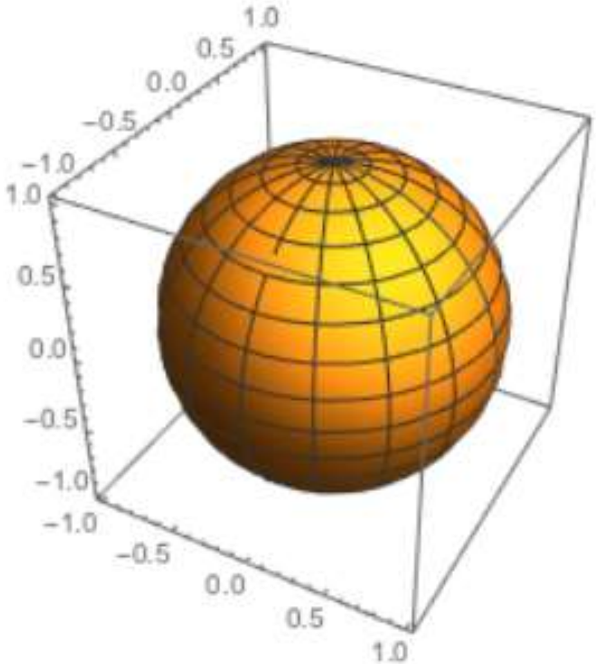
Out[1]=



Plot a sphere:

```
In[1]:= SphericalPlot3D[1, {θ, 0, Pi}, {ϕ, 0, 2 Pi}]
```

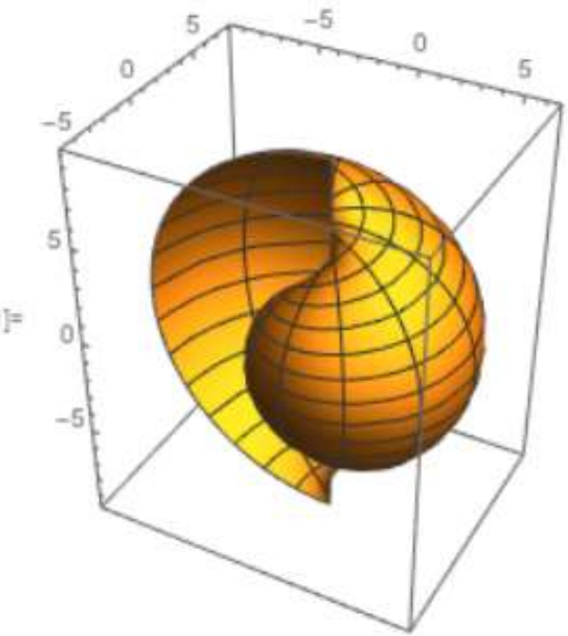
Out[1]=



A spiraling shell:

```
In[1]:= SphericalPlot3D[ϕ, {θ, 0, Pi}, {ϕ, 0, 3 Pi}]
```

Out[1]=



SphericalHarmonicY

SphericalHarmonicY[l, m, θ, ϕ]
gives the spherical harmonic $Y_l^m(\theta, \phi)$.

- Mathematical function, suitable for both symbolic and numerical manipulation.
- The spherical harmonics are orthonormal with respect to integration over the surface of the unit sphere.
- For $l \geq 0$, $Y_l^m(\theta, \phi) = \sqrt{(2l+1)/(4\pi)}\ \sqrt{(l-m)!/(l+m)!}\ P_l^m(\cos(\theta))\ e^{i m \phi}$ where P_l^m is the associated Legendre function.
- For $l \leq -1$, $Y_l^m(\theta, \phi) = Y_{-(l+1)}^m(\theta, \phi)$.

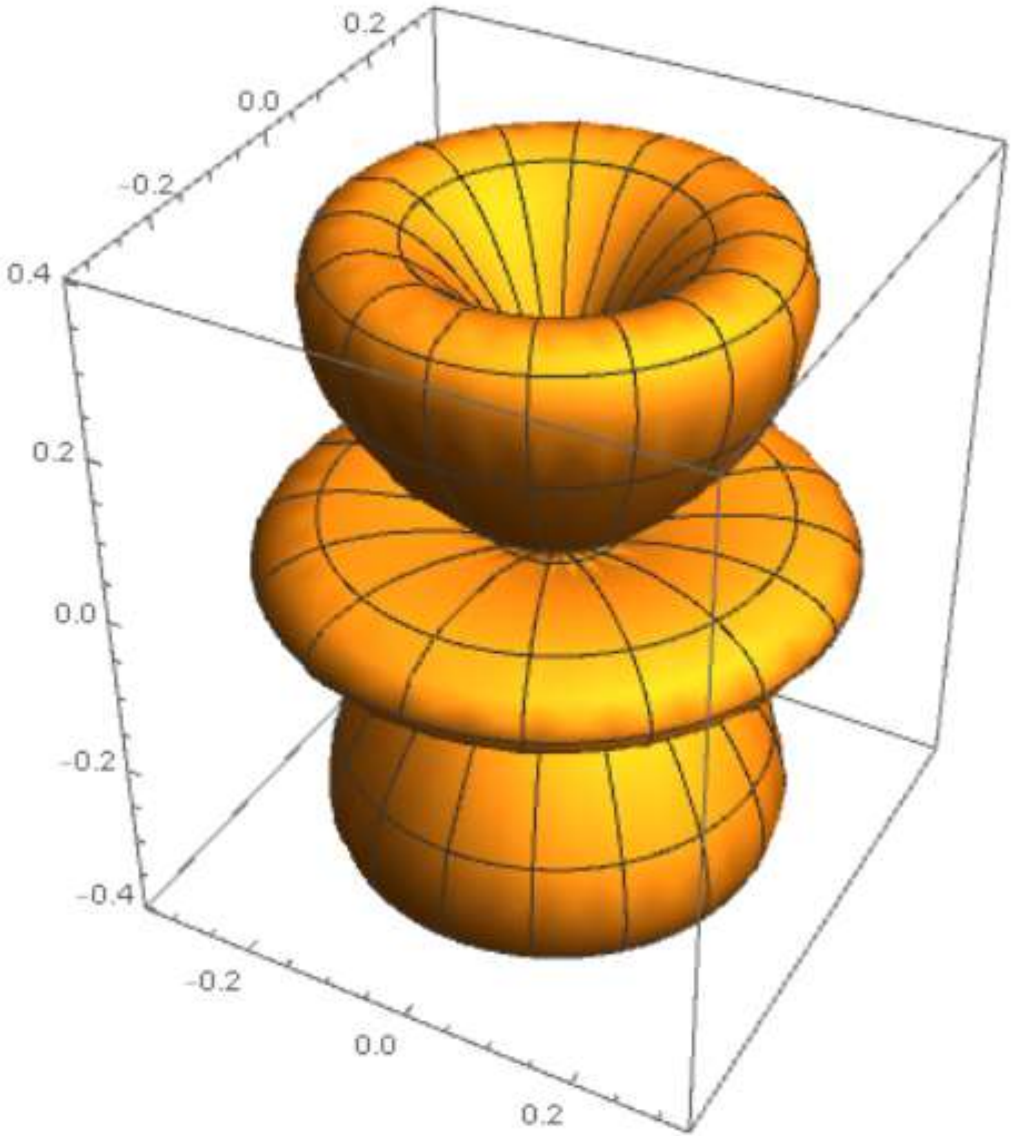
In[1]:= `SphericalHarmonicY[3, 1, θ, ϕ]`

Out[1]= $-\frac{1}{8} e^{i \phi} \sqrt{\frac{21}{\pi}} \left(-1 + 5 \operatorname{Cos}[\theta]^2\right) \operatorname{Sin}[\theta]$

In[1]:= `SphericalHarmonicY[n, m, θ, ϕ] // TraditionalForm`

Out[1]//TraditionalForm=
$Y_n^m(\theta, \phi)$

```
SphericalPlot3D[Abs[SphericalHarmonicY[3, 1, theta, phi]], {theta, 0, Pi}, {phi, 0, 2 Pi}]
```



```
SphericalPlot3D[Abs[SphericalHarmonicY[5, 1, theta, phi]], {theta, 0, Pi}, {phi, 0, 2 Pi}]
```