Chapter 14
Plots

After typing a *Mathematica* command in its notebook interface, you can send it to the kernel (which performs calculations) by pressing Shift-Enter. The result appears in the output cell which follows your input cell. Both are nested in an outer cell representing a calculation step. Later you can return to this input cell, edit the command, and execute it again. The old output will be replaced by the new result. It is allowed to type several commands in a single input cell, but this is not convenient—don't do so unless you have good reasons.

Mathematica Help contains all the necessary information. The Help menu contains Documentation Center, Function Navigator, and Virtual Book (among other things). You can quickly get help for a specific function if you select it with the mouse and press F1.

14.1 2D Plots

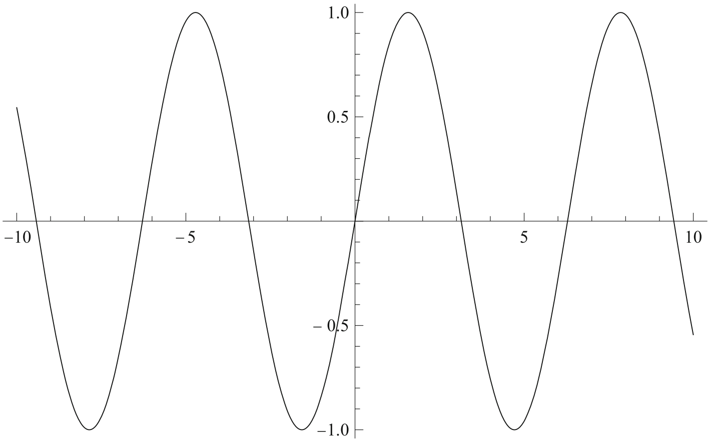
Function Plot

See Help → Virtual Book → Visualization and Graphics → Graphics and Sound → Basic Plotting, and Help → Function Navigator → Visualization and Graphics → Function Visualization → Plot for more details.

A. Grozin, *Introduction to Mathematica® for Physicists*, Graduate Texts in Physics, DOI 10.1007/978-3-319-00894-3_14,

In[1] := Plot[Sin[x], {x, −10, 10}]

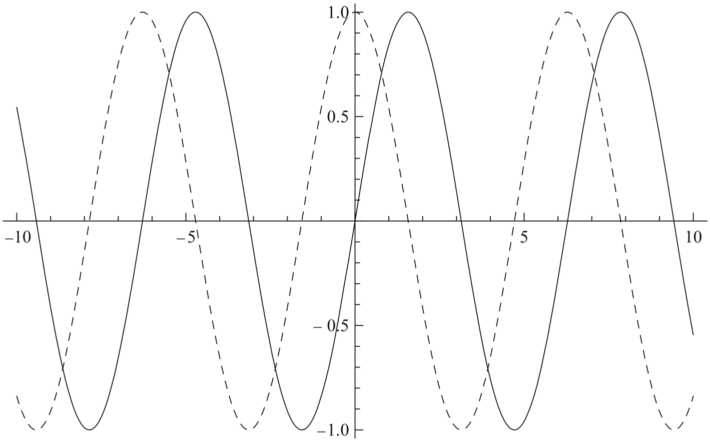
Out[1] =



Several Functions

In[2] := Plot[{Sin[x], Cos[x]}, {x, −10, 10}]

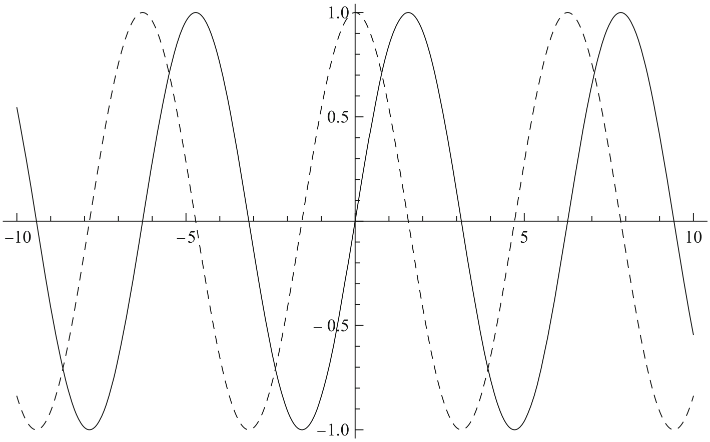
Out[2] =



You can set colors and styles of the curves (Virtual Book → Visualization and Graphics → Graphics and Sound → Options for Graphics; Function Navigator → Visualization and Graphics → Options and Styling → Plotting Options → PlotStyle).

In[3] := Plot[{Sin[*x*], Cos[*x*]}, {*x*, −10, 10}, PlotStyle−>{Red, {Blue, Dashed}}]

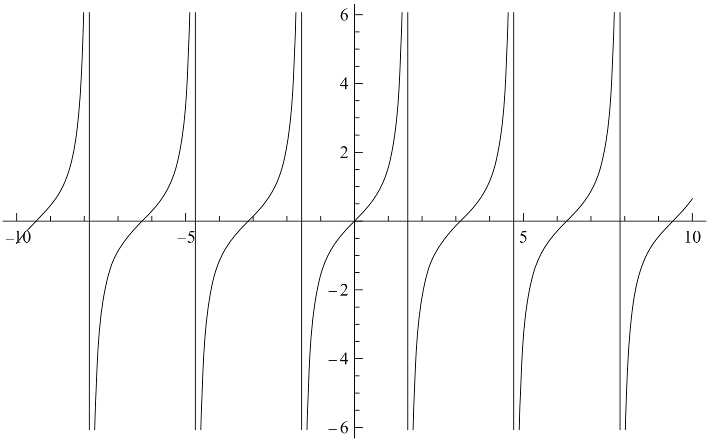
Out[3] =



Unbounded Function

In[4] := Plot[Tan[*x*], {*x*, −10, 10}]

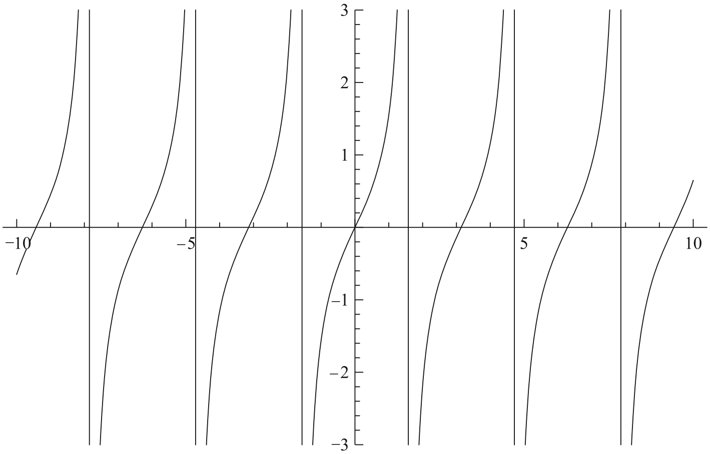
Out[4] =



Mathematica has chosen some *y* scale. How to set it? Find in the Help.

In[5] := Plot[Tan[x], {x, −10, 10}, PlotRange−>{−3, 3}]

Out[5] =

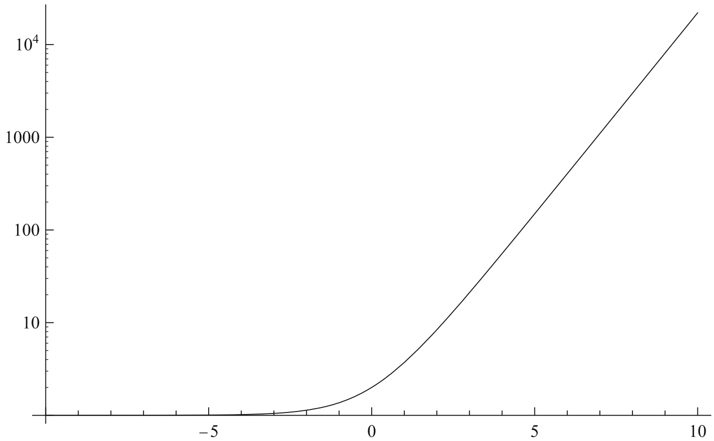


Logarithmic Scale

If our function is positive and varies by orders of magnitude in our region, it is convenient to use logarithmic scale in y. If the independent variable also varies by orders of magnitude, the x-axis scale also should be logarithmic (Function Navigator → Visualization and Graphics → Function Visualization → LogPlot, LogLogPlot).

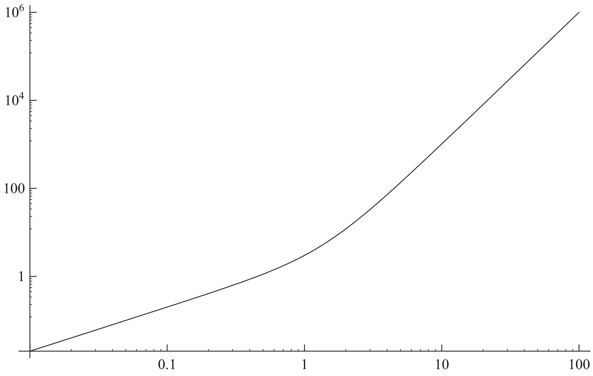
In[6] := LogPlot[Exp[x] + 1, {x, −10, 10}]

Out[6] =



In[7] := LogLogPlot[*x*^3 + 2 * *x*, {*x*, 10^ − 2, 10^2}]

Out[7] =

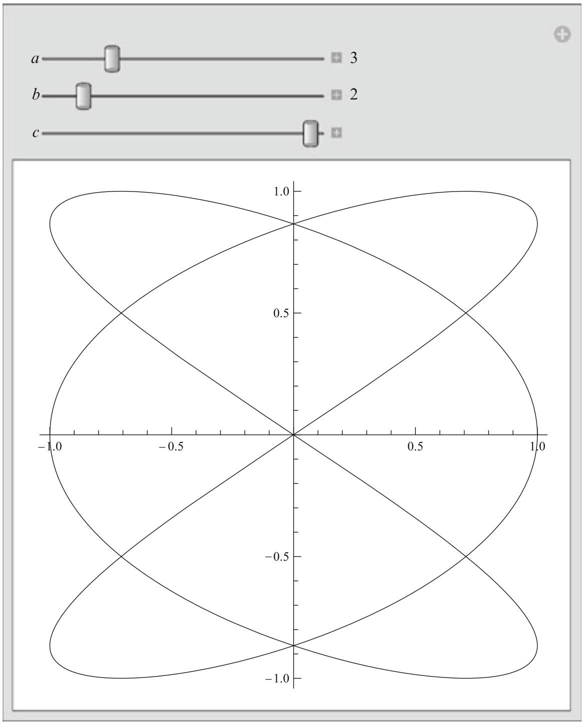


Parametric Curve

Lissajous figures.

In[8] := Manipulate[ParametricPlot[{Sin[*a* * *t* + *c*], Sin[*b* * *t*]}, {*t*, 0, 2 * Pi}],
{*a*, 1, 10, 1, Appearance−>"Labeled"},
{*b*, 1, 10, 1, Appearance−>"Labeled"}, {{*c*, Pi/2}, 0, Pi/2}]

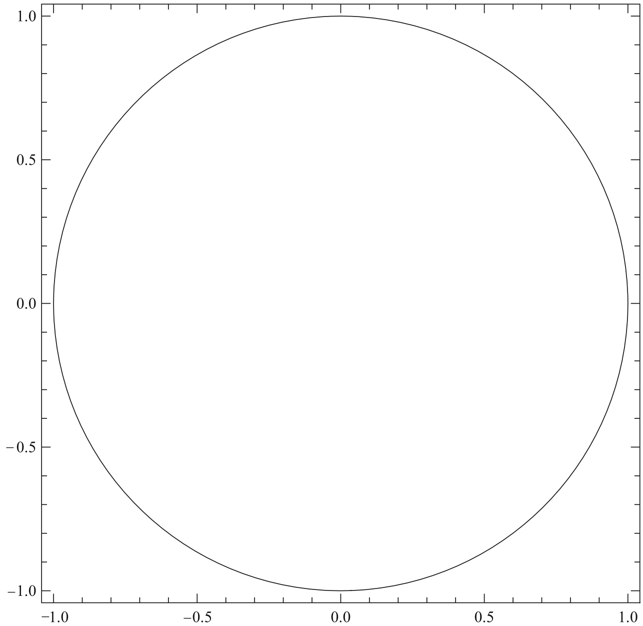
Out[8] =



Implicit Plots

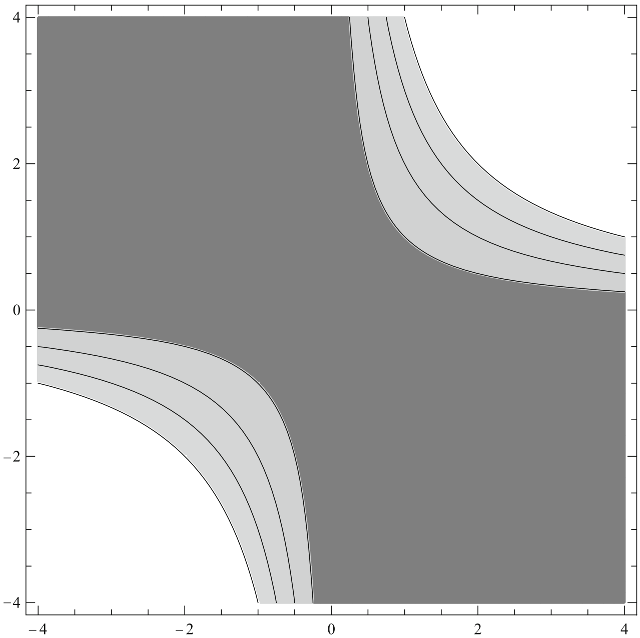
In[9] := ContourPlot[x^2 + y^2 == 1, {x, −1, 1}, {y, −1, 1}]

Out[9] =



In[10] := ContourPlot[x ∗ y, {x, −4, 4}, {y, −4, 4}, Contours−>{1, 2, 3, 4}]

Out[10] =



Experimental Points

In real life they are being read from a file. We don't have such a file at hand, and therefore we'll generate a list of "experimental" points according to some formulas.

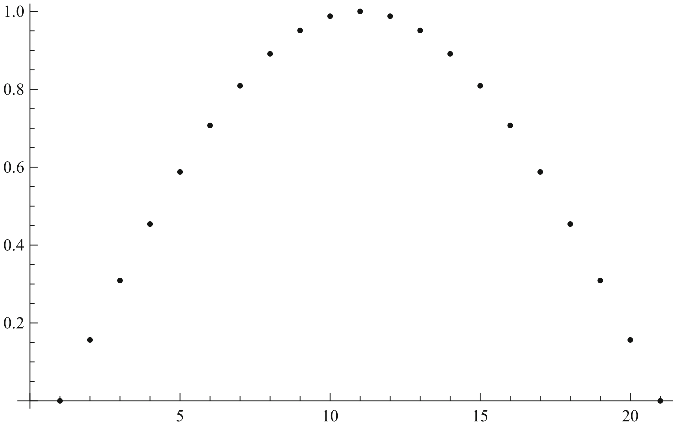
In[11] := *l* = Table[*N*[Sin[Pi * *n*/20]], {*n*, 0, 20}]

Out[11] = {0., 0.156434, 0.309017, 0.45399, 0.587785, 0.707107, 0.809017, 0.891007, 0.951057, 0.987688, 1., 0.987688, 0.951057, 0.891007, 0.809017, 0.707107, 0.587785, 0.45399, 0.309017, 0.156434, 0.}

Function Navigator → Visualization and Graphics → Data Visualization → ListPlot.

In[12] := p1 = ListPlot[*l*]

Out[12] =



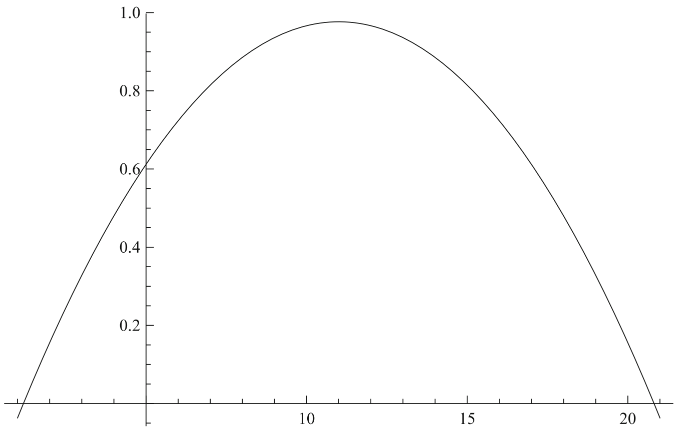
Let's try to fit these points by a quadratic polynomial.

In[13] := *f* = Fit[*l*, {1, *x*, *x*^2}, *x*]

Out[13] = $-0.24953 + 0.222936x - 0.0101334x^2$

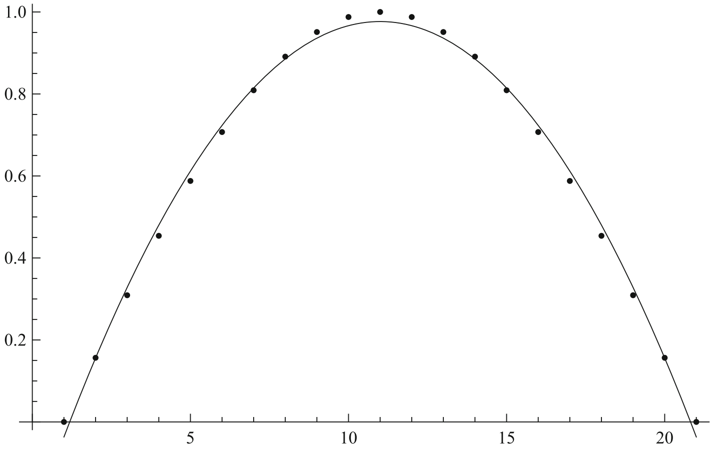
In[14] := p2 = Plot[*f*, {*x*, 1, 21}]

Out[14] =



And now the curve and the points on a single plot (Function Navigator → Visualization and Graphics → Data Visualization → Annotation & Combination → Show).

In[15] := Show[p1, p2]

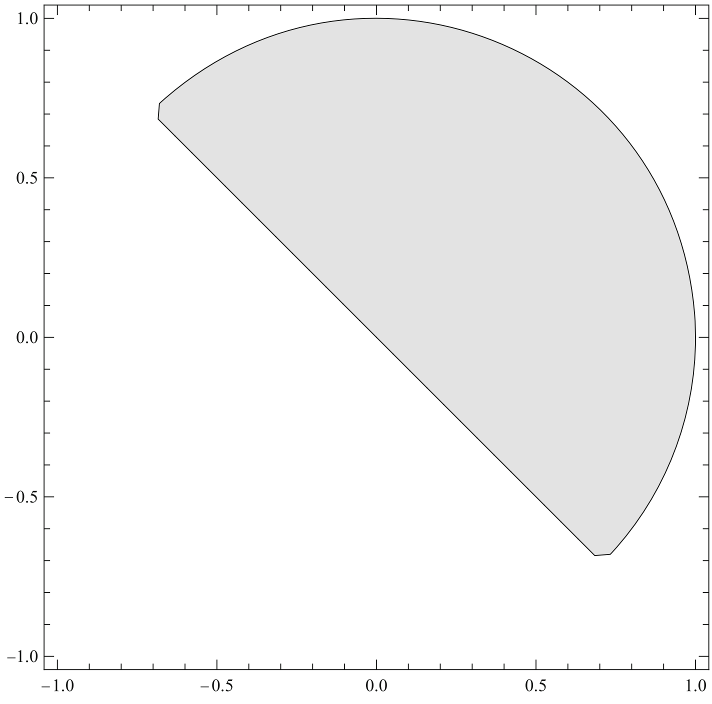
Out[15] = 

In[16] := Clear[*l*, *f*, p1, p2]

Inequalities

Function Navigator → Visualization and Graphics → Function Visualization → RegionPlot.

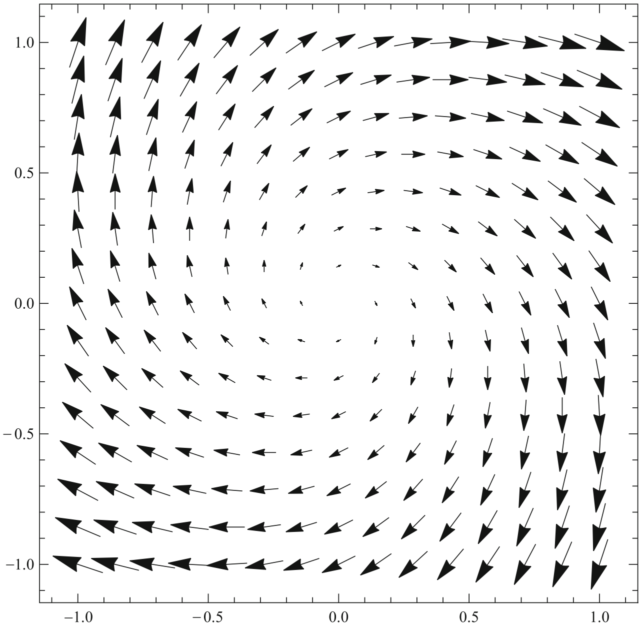
In[17] := RegionPlot[x^2 + y^2 < 1&&x + y > 0, {x, −1, 1}, {y, −1, 1}]

Out[17] = 

Vector Fields

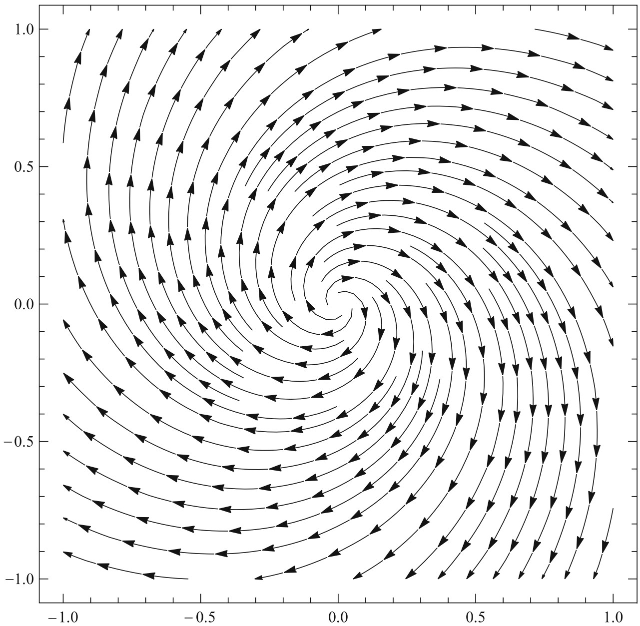
In[18] := VectorPlot[{$y+0.5*x,-x+0.5*y$},{$x,-1,1$},{$y,-1,1$}]

Out[18] =



In[19] := StreamPlot[{$y+0.5*x,-x+0.5*y$},{$x,-1,1$},{$y,-1,1$}]

Out[19] =



14.2 3D Plots

Hat

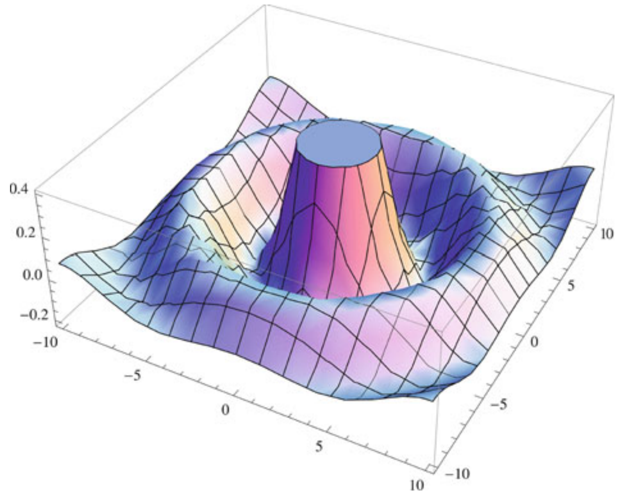
Let's draw a hat with a wavy pent. First define a function.

In[20] := f[x_, y_] := With[{r = Sqrt[x^2 + y^2]}, Sin[r]/r]

Virtual Book → Visualization and Graphics → Three-Dimensional Surface Plots; Function Navigator → Visualization and Graphics → Function Visualization → Plot3D.

In[21] := Plot3D[f[x,y], {x, −10, 10}, {y, −10, 10}]

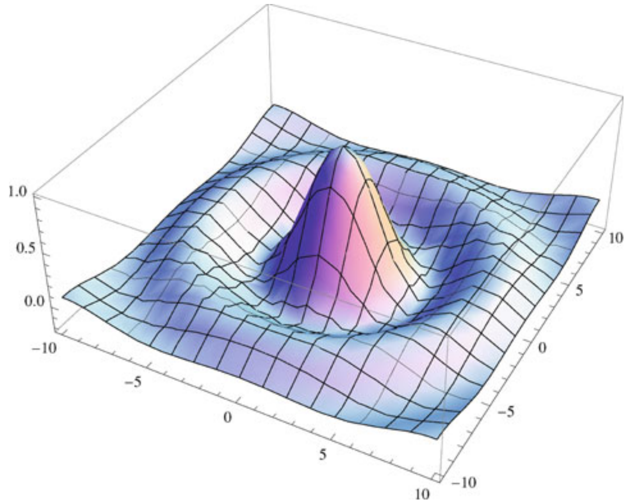
Out[21] =



And why is the top of the hat cut off?

In[22] := Plot3D[f[x,y], {x, −10, 10}, {y, −10, 10}, PlotRange−>All]

Out[22] =



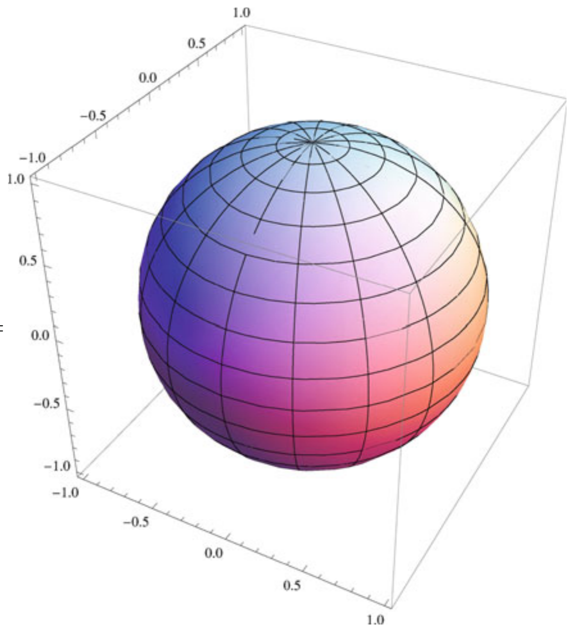
All 3D plots can be rotated by the mouse. If you press Shift, the mouse will move the plot; and if you press Ctrl, then it will resize it.

In[23] := Clear[*f*]

Sphere

**In[24] := ParametricPlot3D[{Sin[θ] ∗ Cos[ϕ], Sin[θ] ∗ Sin[ϕ], Cos[θ]},
{θ, 0, Pi}, {ϕ, 0, 2 ∗ Pi}]**

Out[24] =

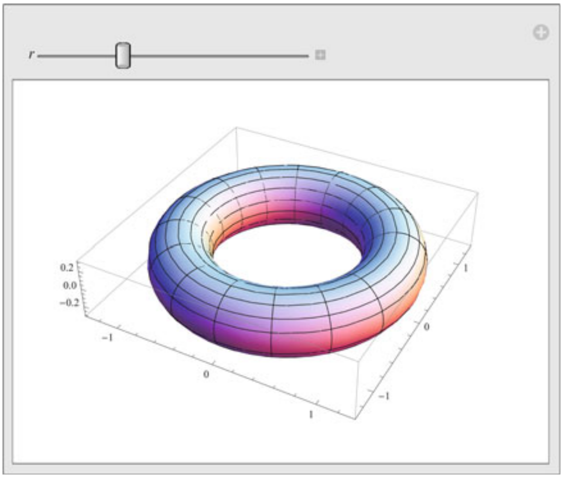


Donut

Take a point on the *x*-axis at a distance *R* from the origin; draw a circle of a radius *r* around it in the *x*, *z* plane; and rotate it around the *z* axis. You will get a donut (torus). Let *R* be 1; *r* can be tuned by the mouse from 0 to 1, with the initial value 0.3.

In[25] := *R* = 1;

In[26] := Manipulate[
ParametricPlot3D[
{(*R* + *r* ∗ Cos[θ]) ∗ Cos[ϕ], (*R* + *r* ∗ Cos[θ]) ∗ Sin[ϕ], *r* ∗ Sin[θ]},
{θ, 0, 2 ∗ Pi}, {ϕ, 0, 2 ∗ Pi}],
{{*r*, 0.3}, 0, 1}]

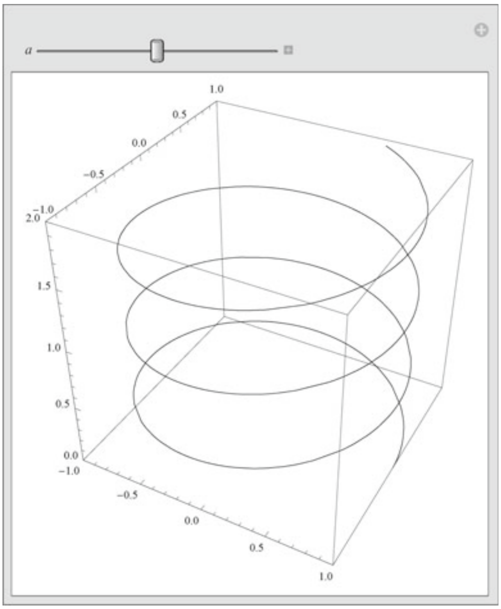
Out[26] = 

In[27] := Clear[*R*]

Spiral

ParametricPlot3D can draw curves, too.

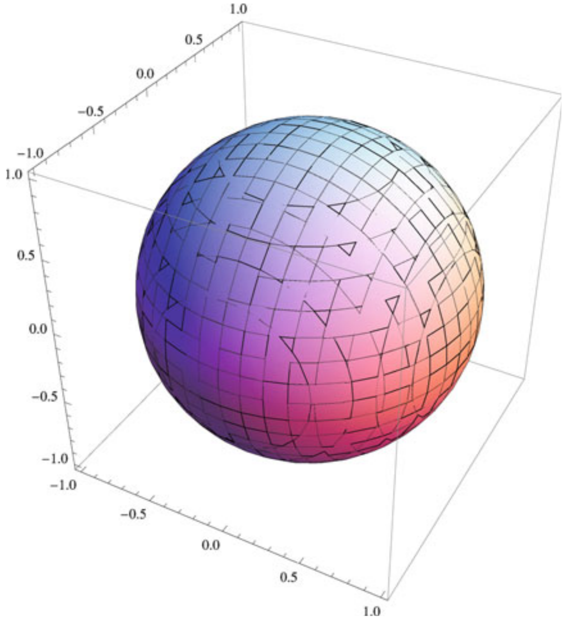
**In[28] := Manipulate[ParametricPlot3D[{Cos[*t*], Sin[*t*], *a* ∗ *t*}, {*t*, 0, 20},
PlotRange−>{{−1, 1}, {−1, 1}, {0, 2}}],
{{*a*, 0.1}, 0, 0.2}]**

Out[28] = 

Implicit Surface

```
In[29] := ContourPlot3D[x^2 + y^2 + z^2 == 1, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```

Out[29] =



Inequalities

```
In[30] := RegionPlot3D[x^2 + y^2 + z^2 < 1&&x + y + z > 0,
    {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
    ViewPoint->{-2, -10, 2}, PlotPoints->100]
```

Out[30] =