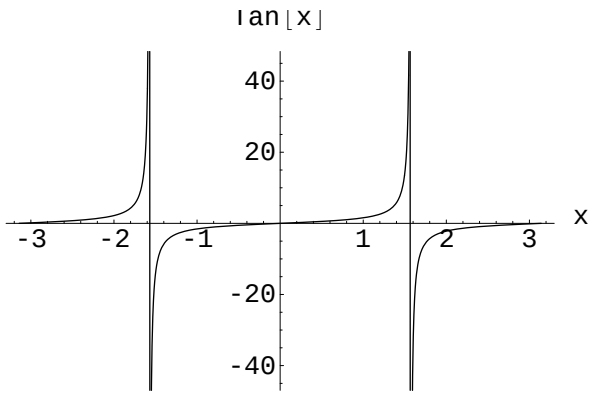
Mathematica Graphics

Dennis Silverman
Mathematical Physics 212 B
Department of Physics and Astronomy
U . C . Irvine

■ Simple Plotting

In[11]:= **Plot[Tan[x], {x, -Pi, Pi}, AxesLabel -> {"x", "Tan[x]"}]**



Out[11]= **-** Graphics **-**

? Plot

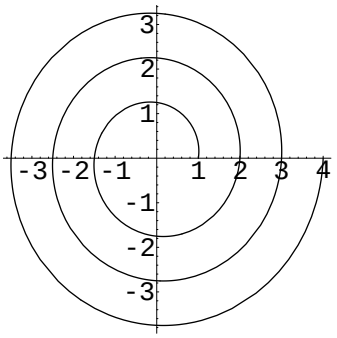
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi.

? ListPlot

ListPlot[{y1, y2, ... }] plots a list of values. The x coordinates for each point are taken to be 1, 2, ... ListPlot[{{x1, y1}, {x2, y2}, ... }] plots a list of values with specified x and y coordinates.

ListPlots are also provided with most other types of fancier plots .

```
ParametricPlot[{Cos[t] (1 + t / (2 Pi)), Sin[t] (1 + t / (2 Pi))}, {t, 0, 6 Pi},
 AspectRatio -> 1]
```




```
- Graphics -
```

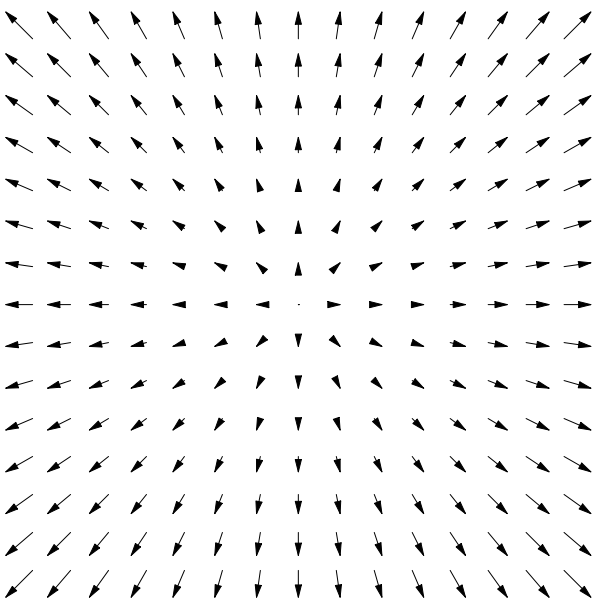
```
In[1]:= << Graphics`Graphics`
```

```
This provides BarChart[list], StackedBarChart, PieChart[list], LogPlot,
LogLogPlot, and PolarPlot .

<< Graphics`Graphics3D`
This provides BarChart3D, ScatterPlot3D, ShadowPlot3D, ListSurfacePlot3D,
```

■ Vector Fields – 2 Dimensional

```
In[8]:= << Graphics`PlotField`
        PlotVectorField[{x, y}, {x, -1, 1}, {y, -1, 1}]
```



```
Out[9]= - Graphics -
```

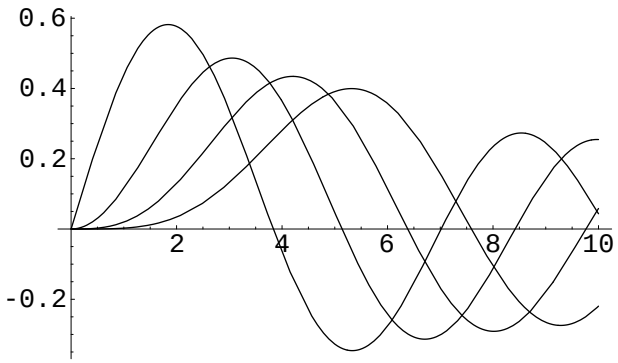
```
This also provides PlotGradientField .
```

■ Plotting From a Table

Evaluate[Table ..., means that the table is only produced once, and then the plot proceeds, to save time .

```
g1 := Plot[Evaluate[Table[BesselJ[n, x], {n, 4}]], {x, 0, 10}]
```

```
g1
```



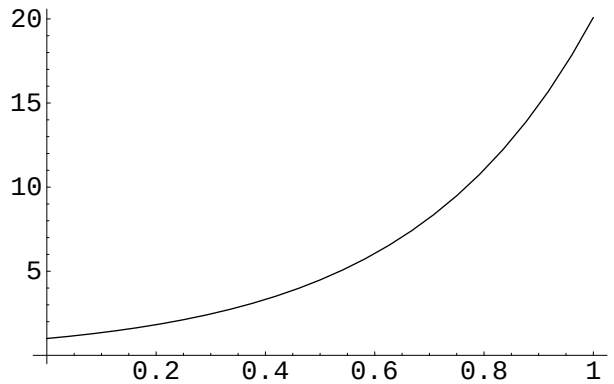
- Graphics -

For a numerical solution, an interpolating function is set up automatically, and then the function is evaluated in the plot routine by using Evaluate, without resolving the equation for each plot point .

```
In[20]:= NDSolve[{y'[x] == 3 y[x], y[0] == 1}, y, {x, 0, 1}]

Out[20]= {{y → InterpolatingFunction[{{0., 1.}}, <>]}}

In[21]:= Plot[Evaluate[y[x] /. %], {x, 0, 1}]
```




```
Out[21]= - Graphics -
```

■ Operations on Graphs

By clicking on a graph with the left mouse button or with graphics commands, you can move, resize, Rotate3D, etc . the graphics .

A sequence of graphics can be animated with :
ShowAnimation[{g1, g2, ...}] .

■ Graphics Output

Mathematica generates Postscript graphics,
and they can also be output as encapsulated postscript to fit as figures in a larger page .

Print graphics with : PSPrint[graphics objects] .

Save graphics with : Display["file", graphics] .

■ Graphics Options

```
?? Plot

Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
  xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi.

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio -> GoldenRatio^(-1), Axes -> Automatic, AxesLabel -> None,
   AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
   ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
   Epilog -> {}, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
   FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
   MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
   PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
   Prolog -> {}, RotateLabel -> True, Ticks -> Automatic, DefaultFont :> $DefaultFont,
   DisplayFunction :> $DisplayFunction, FormatType :> $FormatType, TextStyle :> $TextStyle}
```

We note here :

PlotPoints -> 100, default is 15

PlotRange -> {{0, 1}, {-2, 2}}, default is Automatic

Gridlines -> Automatic, default is none .

PlotLabel -> "Title", can be used .

AxesLabel -> ylabel, like "Time", or both labels in {xlabel, ylabel}

Frame -> True, adds a frame, the default being False .

Options[function] lists all options for any function in Mathematica

Options[Plot]

SetOptions[function, option -> value, ...] resets the defaults .

■ Operations With Plots

Show[plot] redraws the plot, and you can add new options .

Show[plot1, plot2, ...] combines plots .

Show[GraphicsArray[{plot1, plot2, ...}] shows side by side plots .

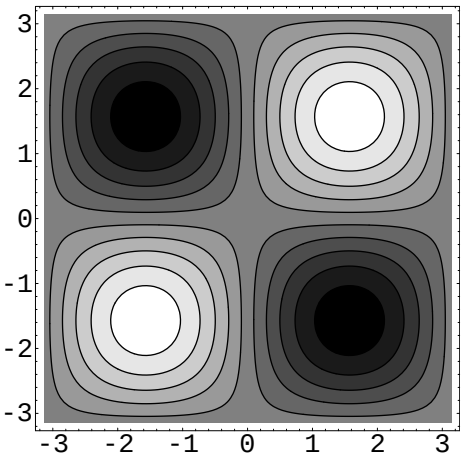
Show[GraphicsArray[{{plot11, plot12}, {plot21, plot22}}] makes a two by two array of plots .

The option : GraphicsSpacing -> {h, v} sets certain fractions h and v of the graphs' width and height for their spacing .

■ Three Dimensional Graphics

■ Contour Plotting

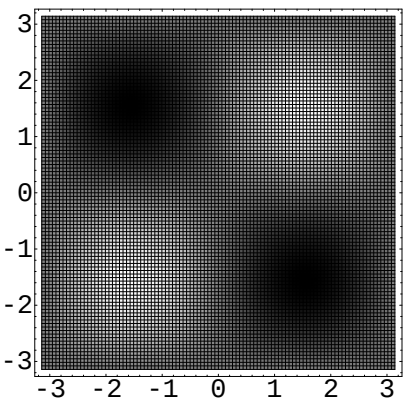
```
ContourPlot[Sin[x] Sin[y], {x, -Pi, Pi}, {y, -Pi, Pi}, PlotPoints -> 100]
```



- ContourGraphics -

■ Density Plotting

```
DensityPlot[Sin[x] Sin[y], {x, -Pi, Pi}, {y, -Pi, Pi}, PlotPoints -> 100]
```

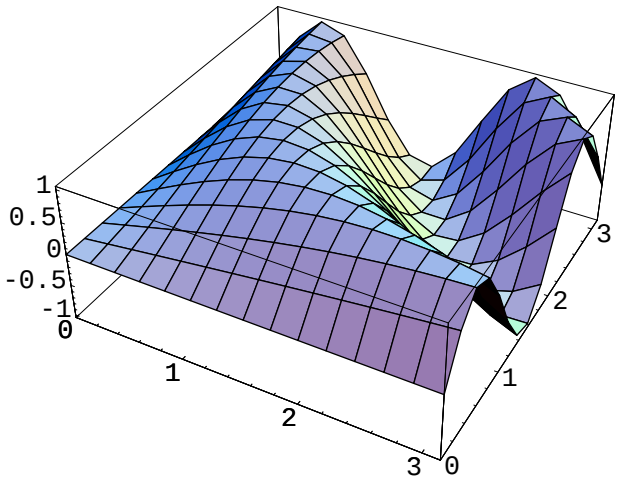


- DensityGraphics -

■ 3D Plots

```
g3d := Plot3D[Sin[x y], {x, 0, Pi}, {y, 0, Pi}]
```

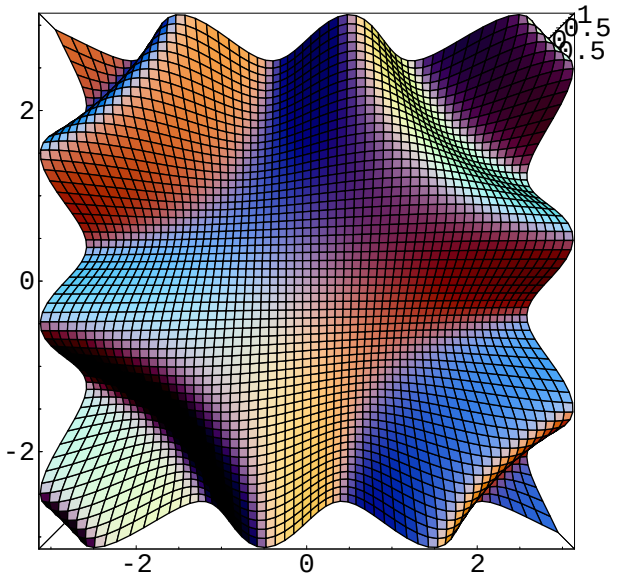
```
g3d
```




```
- SurfaceGraphics -
```

```
This can be modified with the options :
ViewPoint -> {x, y, z} point to view from, and
PlotPoints -> {nx, ny} .
```

```
Plot3D[Sin[x y], {x, -Pi, Pi}, {y, -Pi, Pi}, ViewPoint -> {0, 0, 2},
 PlotPoints -> {60, 60}]
```



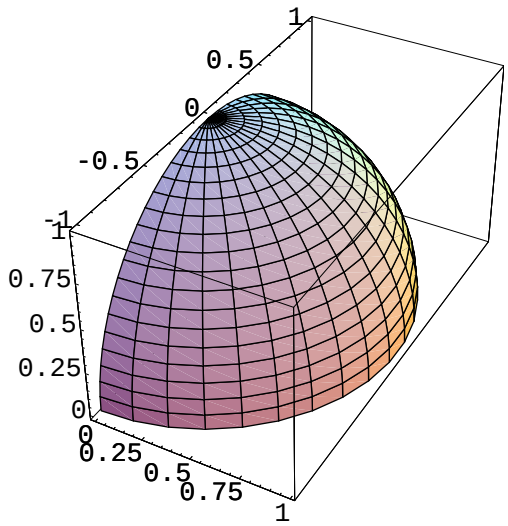

```
- SurfaceGraphics -
```

```
?ListPlot3D

ListPlot3D[array] generates a three-dimensional plot of a surface representing
  an array of height values. ListPlot3D[array, shades] generates a plot with each
  element of the surface shaded according to the specification in shades.
```

■ Parametric Plots

```
ParametricPlot3D[{Sin[τ] Cos[ϕ], Sin[τ] Sin[ϕ], Cos[τ]}, {τ, 0, Pi / 2},
 {ϕ, -Pi / 2, Pi / 2}]
```

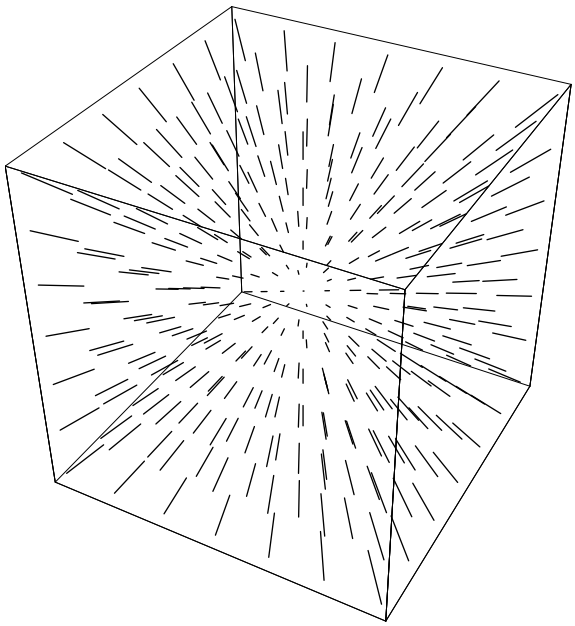



```
- Graphics3D -
```

■ Vector Fields – 3 Dimensional

```
In[23]:= << Graphics`PlotField3D`
```

```
PlotVectorField3D[{x, y, z}, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```



```
- Graphics3D -
```

```
In[25]:= PlotGradientField3D[1 / Sqrt[x^2 + y^2 + z^2], {x, 0.3, 1}, {y, 0.3, 1}, {z, 0.3, 1}]
```

■ Adding Text

Besides the AxesLabel and PlotLabel options, direct text can be added by the Text command, where expr in the following has double quotes "text" around the text .

```
Text[expr, {x, y}] has text centered at {x, y} . It can also be placed with
Text[expr, {x, y}, {-1, 0}] left hand end at x, y;
                    {0, -1} centered above x, y;
                    {0, 1}  centered below x, y, and etc .
```

■ Animation

```
In[1]:= << Graphics`Animation`
```

On the menu Cell there is Animate Selected Graphics . Equivalent is SelectionAnimate[object] with the option AnimateDisplayTime -> 6 in seconds for each cell .

```
Animate[plot dependent on t, {t, tmin, tmax}]
```

In this example we make nine plots without axes, with a fixed vertical plot range for uniformity, and without displaying them as they are created :

```
Table[Plot3D[BesselJ[0, Sqrt[x^2 + y^2] + t], {x, -10, 10}, {y, -10, 10},
  Axes -> False, PlotRange -> {-0.5, 1.0}, DisplayFunction -> Identity],
 {t, 0, 8}] // Short
```

```
{- SurfaceGraphics -, <<1>>, <<1>>, - <<15>> -}
```

In[3]:=
```
ShowAnimation[%]
```