Multivariable Calculus

This notebook will introduce you to some *Mathematica* tools which will help you explore topics in multivariable calculus corresponding to chapters 8 through 12 of the text.

■ Vectors and Matrices

This section will show you how to define vectors and matrices with *Mathematica* and how to perform operations on vectors and matrices.

Entering a matrix

Suppose want to enter the matrix $\begin{pmatrix} 1 & 0 & 2 \\ 5 & 3 & 7 \\ 6 & 2 & 9 \\ 4 & -3 & 1 \end{pmatrix}$ in an input statement. To do this from the keyboard, enter a list and within this list, enter row one as a list, then row 2 as a list and so forth.

```
In[1]:=  m = {{1, 0, 2}, {5, 3, 7}, {6, 2, 9}, {4, -3, 1}}

Out[1]=  {{1, 0, 2}, {5, 3, 7}, {6, 2, 9}, {4, -3, 1}}
```

Typically, a matrix is denoted by a capital letter, but since you want to avoid conflicts with *Mathematica* commands (which always begin with a capital letter), use a lower case letter when assigning names to vectors and matrices. For example, we named the matrix above *m* instead of using a capital *M*.

To output a matrix in the form of a matrix use `// MatrixForm`.

```
In[2]:=  m // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 2 \\ 5 & 3 & 7 \\ 6 & 2 & 9 \\ 4 & -3 & 1 \end{pmatrix}$$

To enter a matrix using the *BasicInput* palette, click on the button containing $\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$. To add rows, press **Shift-Enter** and to add columns, press **Shift-,** (i.e. press the comma key while holding down the **Shift** key).

In[3]:= **n =** $\begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \\ \mathbf{5} & \mathbf{6} & \mathbf{7} & \mathbf{8} \\ \mathbf{9} & \mathbf{10} & \mathbf{11} & \mathbf{12} \end{pmatrix}$

```
Out[3]=  {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}
```

Execute `n // MatrixForm` or `% // MatrixForm` to see the matrix in its natural form.

```
In[4]:= % // MatrixForm
```

Out[4]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

Matrix Operations

A period key is used to multiply two matrices. Below, the product *mn* is computed.

```
In[5]:= m.n // MatrixForm
```

Out[5]//MatrixForm=

$$\begin{pmatrix} 19 & 22 & 25 & 28 \\ 83 & 98 & 113 & 128 \\ 97 & 114 & 131 & 148 \\ -2 & 0 & 2 & 4 \end{pmatrix}$$

Use + and – to add and subtract matrices, respectively. To multiply a scalar constant and a matrix, place a space between the constant and the matrix. Here are some examples.

In[6]:= `Clear[a, b];`

$$a = \begin{pmatrix} 12 & 3 & -9 \\ -5 & 5 & 0 \end{pmatrix};$$

$$b = \begin{pmatrix} 41 & -22 & 3 \\ 6 & -9 & 78 \end{pmatrix};$$

```
a + b // MatrixForm
a - b // MatrixForm
3 a // MatrixForm
```

Out[9]//MatrixForm=

$$\begin{pmatrix} 53 & -19 & -6 \\ 1 & -4 & 78 \end{pmatrix}$$

Out[10]//MatrixForm=

$$\begin{pmatrix} -29 & 25 & -12 \\ -11 & 14 & -78 \end{pmatrix}$$

Out[11]//MatrixForm=

$$\begin{pmatrix} 36 & 9 & -27 \\ -15 & 15 & 0 \end{pmatrix}$$

Determinants

The `Det` command is used to compute the determinant of a matrix.

In[12]:= $m = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix};$

```
Det[m]
```

Out[13]= -2

In[14]:=
```
Clear[m];
m = (5 8 4
     2 8 9
     3 4 7);
Det[m]
```
Out[16]= 140

Vectors

A vector is just a list of numbers in *Mathematica*. For example, (1, 2, 3) or $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ is entered as {1, 2, 3}.

In[17]:=
```
u = {1, 2, 3}
```
Out[17]= {1, 2, 3}

In[18]:=
```
u // MatrixForm
```
Out[18]//MatrixForm=

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Matrix-vector Multiplication

Use a period to multiply a matrix and a vector. Study the following example.

In[19]:=
```
Clear[m, x, y, z, u];
m = ( 1 9 7
     -3 2 7
      6 5 4); u = {x, y, z};
m.u
```
Out[21]= {x + 9 y + 7 z, -3 x + 2 y + 7 z, 6 x + 5 y + 4 z}

In[22]:=
```
m.u // MatrixForm
```
Out[22]//MatrixForm=

$$\begin{pmatrix} x + 9 y + 7 z \\ -3 x + 2 y + 7 z \\ 6 x + 5 y + 4 z \end{pmatrix}$$

An example of a linear function $L[u]$, where $u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, is shown below.

In[23]:=
```
L[x_, y_, z_] = m.u
```
Out[23]= {x + 9 y + 7 z, -3 x + 2 y + 7 z, 6 x + 5 y + 4 z}

Below the value of $L\left[\begin{pmatrix}2\\4\\6\end{pmatrix}\right]$ is computed along with $2\,L\left[\begin{pmatrix}1\\2\\3\end{pmatrix}\right]$, illustrating the linear property (8) in Section 8.4 of the text.

```
In[24]:= L[2, 4, 6]
         2 L[1, 2, 3]
Out[24]= {80, 44, 56}
Out[25]= {80, 44, 56}
```

The following example illustrates linear property (7) in Section 8.4 of the text.

```
In[26]:= L[1, 2, 3] + L[7, 8, 9]
         L[8, 10, 12]
Out[26]= {182, 80, 146}
Out[27]= {182, 80, 146}
```

Dot and Cross Products of Vectors

The package *VectorAnalysis* contained in the *Calculus* directory contains the commands `CrossProduct` and `DotProduct`, which can be used to compute the cross product and dot product of two vectors. These two commands are used below to compute $\begin{pmatrix}1\\2\\4\end{pmatrix}\times\begin{pmatrix}2\\3\\6\end{pmatrix}$ and then the dot product of the same two vectors.

```
In[28]:= << Calculus`VectorAnalysis`

In[29]:= Clear[u, v];
         u = {1, 2, 4};
         v = {2, 3, 6};
         uCrossv = CrossProduct[u, v]
         udotv = DotProduct[u, v]

Out[32]= {0, 2, -1}
Out[33]= 32
```

Therefore $\begin{pmatrix}1\\2\\4\end{pmatrix}\times\begin{pmatrix}2\\3\\6\end{pmatrix}=\begin{pmatrix}0\\2\\-1\end{pmatrix}$ and the dot product is 32. Of course, you can also compute these values without using the add-on package. Below, we form the matrix $\begin{pmatrix}i & j & k\\1 & 2 & 4\\2 & 3 & 6\end{pmatrix}$ and then compute the determinant of this matrix to find the cross product.

```
In[34]:= m = {{i, j, k}, u, v}
Out[34]= {{i, j, k}, {1, 2, 4}, {2, 3, 6}}
```

In[35]:=
```
m // MatrixForm
```

Out[35]//MatrixForm=

$$\begin{pmatrix} i & j & k \\ 1 & 2 & 4 \\ 2 & 3 & 6 \end{pmatrix}$$

In[36]:=
```
Det[m]
```

Out[36]= 2 j − k

The period operator can be used to compute the dot product of two vectors.

In[37]:=
```
u.v
```

Out[37]= 32

Volume of a Parallelepiped

The following example illustrates how to compute the volume of a parallelepiped determined by $\begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}$, $\begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix}$, and $\begin{pmatrix} 1 \\ 10 \\ 14 \end{pmatrix}$.

In[38]:=
```
a = {1, 2, 5}; b = {2, 0, -4}; c = {1, 10, 14};
m = {a, b, c}
MatrixForm[m]
```

Out[39]= {{1, 2, 5}, {2, 0, −4}, {1, 10, 14}}

Out[40]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 5 \\ 2 & 0 & -4 \\ 1 & 10 & 14 \end{pmatrix}$$

`Abs[exp]` represents the absolute value of an expression *exp*.

In[41]:=
```
volume = Abs[Det[m]]
```

Out[41]= 76

■ Plotting Planes

The *Mathematica* command `Plot[f[x, y], {x, a, b}, {y, c, d}]` is used to plot the graph of $z = f(x, y)$ for values of x from a to b and for values of y from c to d. Suppose you wish to plot the plane $2x + 3y + 5z = 10$ for values of x from a to b and for values of y from c to d.

To plot the equation of a plane $\mathrm{ax} + \mathrm{by} + \mathrm{cz} = d$, where $c \neq 0$, first solve the equation for z.

In[42]:=
```
Clear[x, y, z];
Solve[2 x + 3 y + 5 z == 10, z]
```

Out[43]= $\left\{\left\{z \to \frac{1}{5}\ (10 - 2\,x - 3\,y)\right\}\right\}$

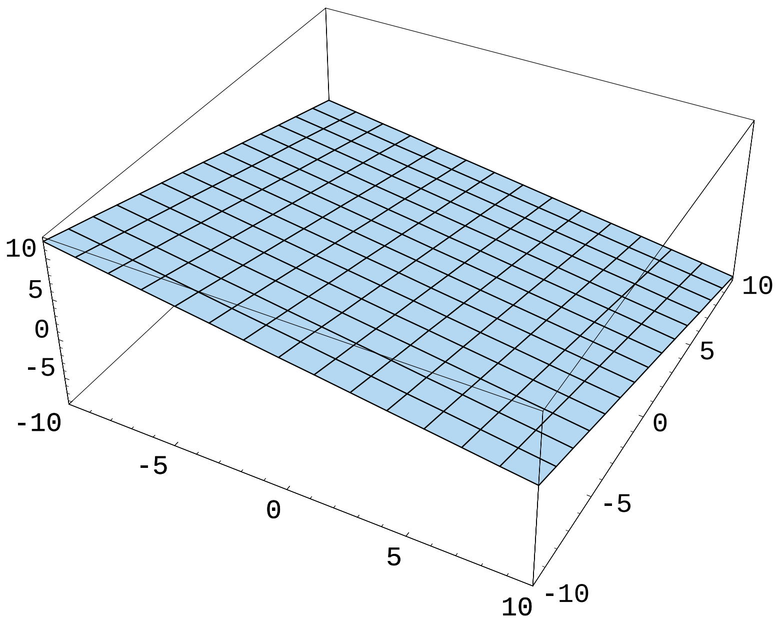
So for this example, $z = f(x, y) = \frac{1}{5}(10 - 2x - 3y)$. We now define the function $f(x, y)$ of two variables.

In[44]:= `f[x_, y_] = %[[1, 1, 2]]`

Out[44]= $\frac{1}{5}(10 - 2x - 3y)$

Now the `Plot` command is used to graph the corresponding plane.

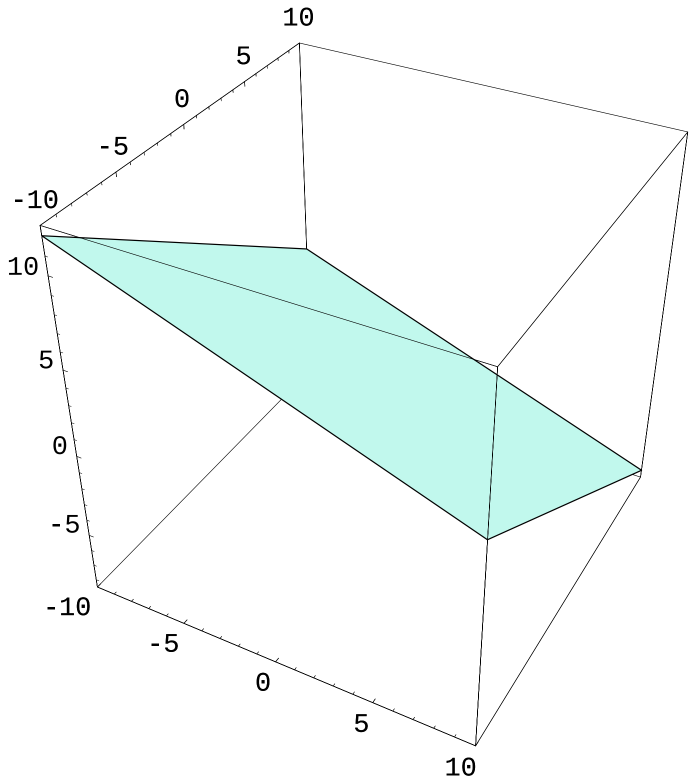
In[45]:= `Plot3D[f[x, y], {x, -10, 10}, {y, -10, 10}]`



Out[45]= - SurfaceGraphics -

Mathematica plots a 3-dimensional surface, like the plane above, by connecting together a sequence of small rectangles. The option `PlotPoints` → 2 is used to decrease the number of rectangles being plotted. The other option, BoxRatios → {*w*, *l*, *h*}, will plot the surface so that the ratio of width to length to height is *w* to *l* to *h*.

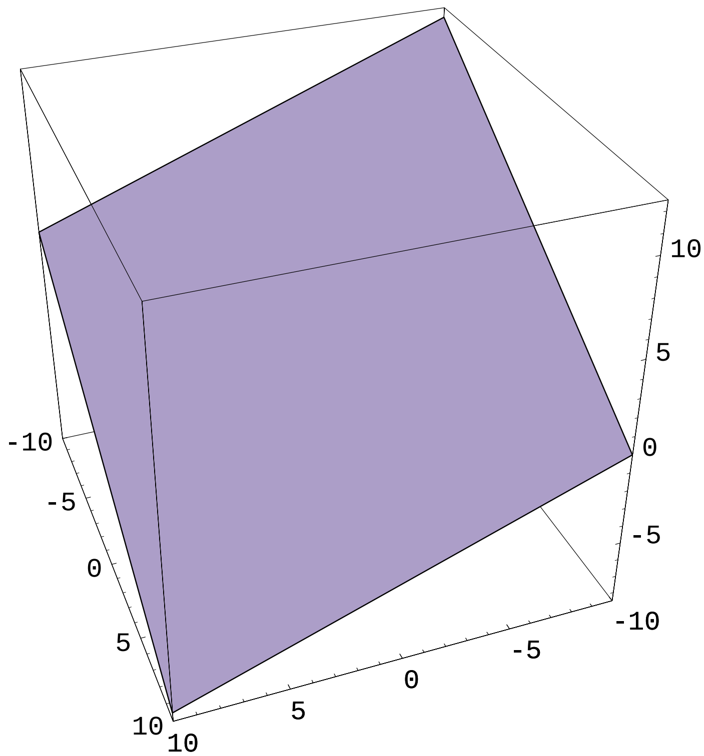
In[46]:= **Plot3D**[**f**[**x, y**], {**x, -10, 10**},
{**y, -10, 10**}, **PlotPoints → 2, BoxRatios → {1, 1, 1**}]



Out[46]= - SurfaceGraphics -

Now suppose you want to view the surface from another point of view. You could begin by copying and pasting the last input command into the cell below and then place a comma immediately after the last option BoxRatios → {1, 1, 1}. Then select **Input**, and then **3D ViewPoint Selector...** . Then place the cursor on one of the corners of the cube in the 3D ViewPoint Selector window and drag the mouse, while holding down the mouse button, to change the point of view. Click on the Paste button once you get the desired position. Then execute the plot command. Here is an example.

```
In[47]:= Plot3D[f[x, y], {x, -10, 10}, {y, -10, 10}, PlotPoints → 2,
          BoxRatios → {1, 1, 1}, ViewPoint -> {0.994, 2.660, 1.889}]
```

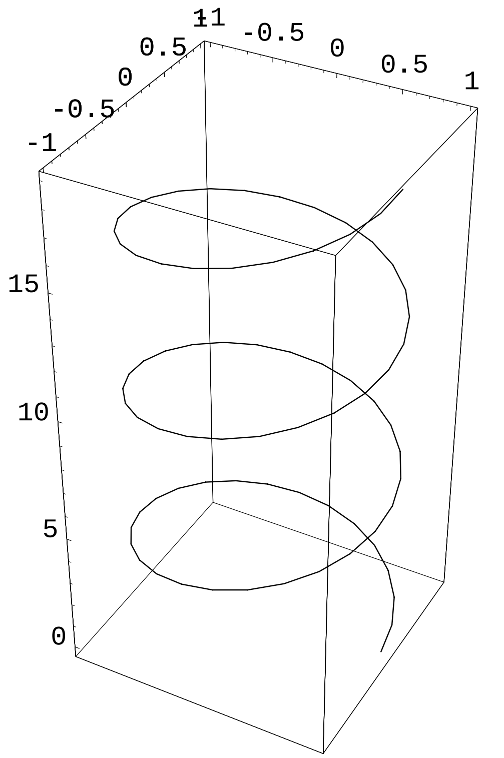



```
Out[47]= - SurfaceGraphics -
```

■ Plotting Space Curves

The command `ParametricPlot3D[{x[t], y[t], z[t]}, {t, a, b}]` is used to plot the space curve $\langle x(t),\ y(t),\ z(t)\rangle$ for values of t from a to b. Here is an example of a helix.

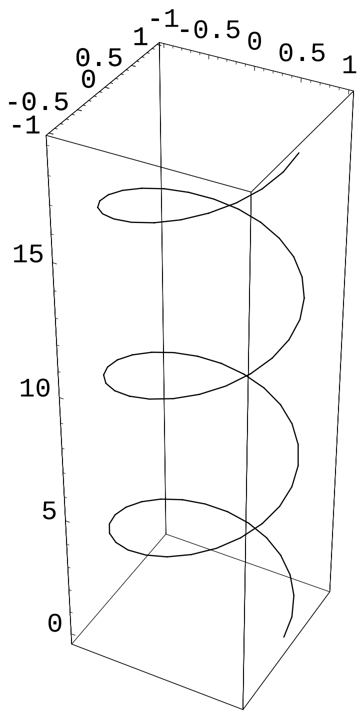
In[48]:= **ParametricPlot3D**[{**Cos**[**t**], **Sin**[**t**], **t**}, {**t**, **0**, **6** π}, **BoxRatios** → {**1**, **1**, **2**}]



Out[48]= - Graphics3D -

The `BoxRatios` option is now used to adjust the width to length to height ratios.

```
In[49]:= ParametricPlot3D[{Cos[t], Sin[t], t}, {t, 0, 6 π}, BoxRatios → {1, 1, 3}]
```

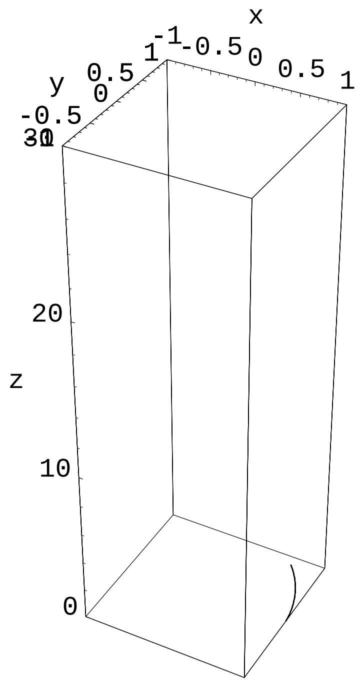


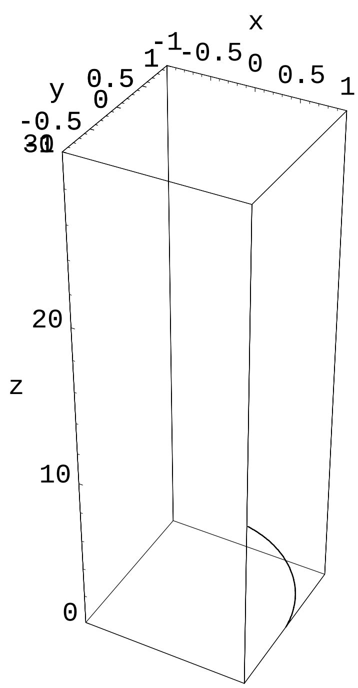

```
Out[49]= - Graphics3D -
```

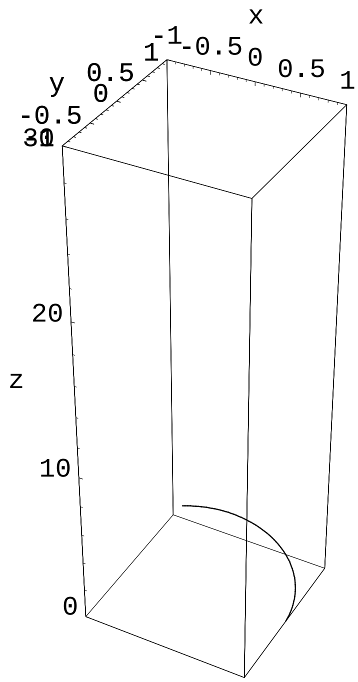
■ Animating Motion

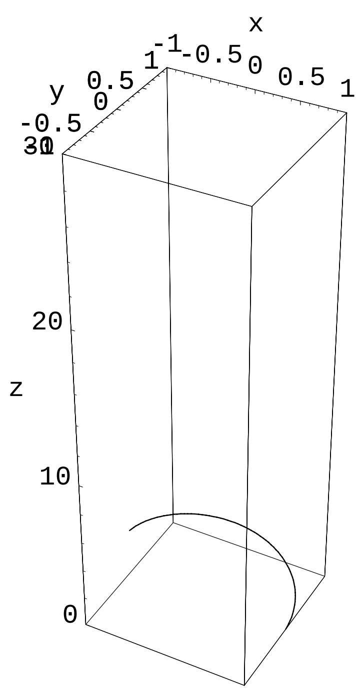
Study the following input command. The `Table` command is used here to animate the motion of the space curve as t increases from 0 to 8π. To animate the frames, double-click on any one of the frames displayed in the output below.

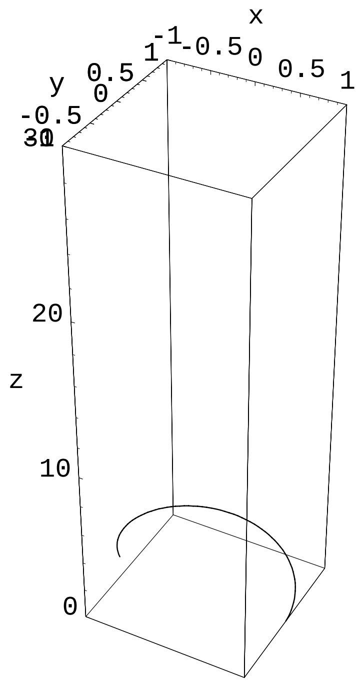
```
In[50]:= Table[ParametricPlot3D[{Cos[t], Sin[t], t}, {t, 0, k},
            BoxRatios → {1, 1, 3}, PlotRange → {{-1, 1}, {-1, 1}, {0, 30}},
            AxesLabel → {x, y, z}], {k, π/4, 8 π, π/4}];
```

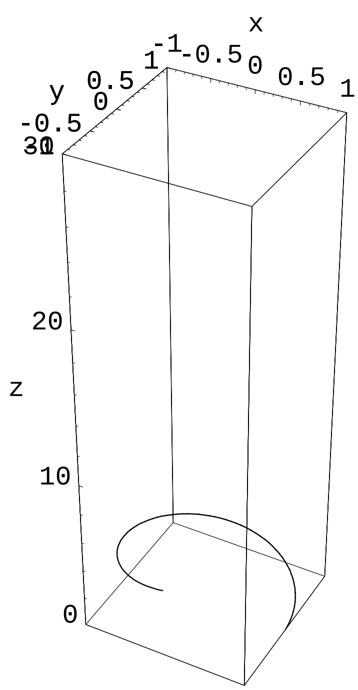

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

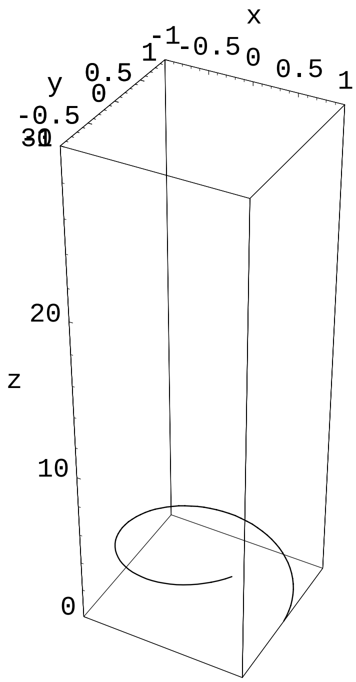

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

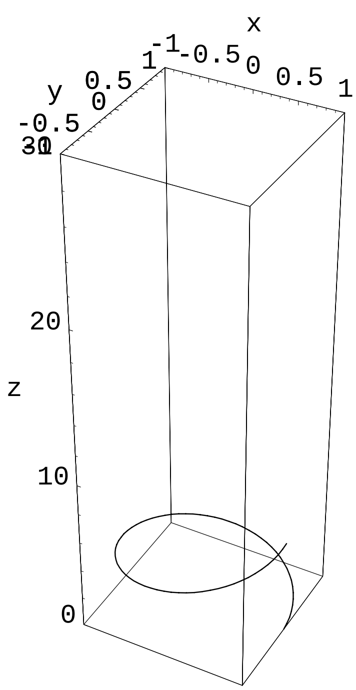

x
-1
-0.5
0
0.5
1
y
0.5
0
-0.5
1
30
20
z
10
0

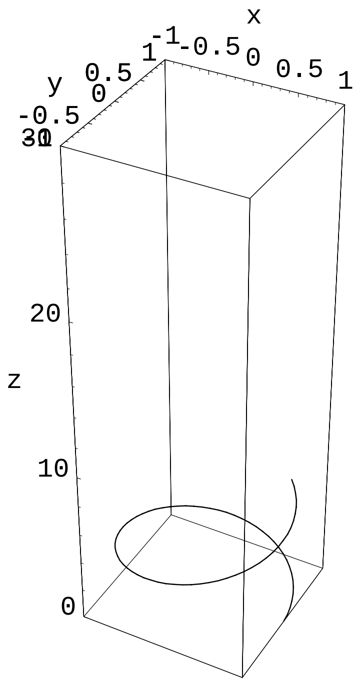

x
-1
-0.5
0
0.5
1
y
0.5
0
-0.5
1
30
20
z
10
0

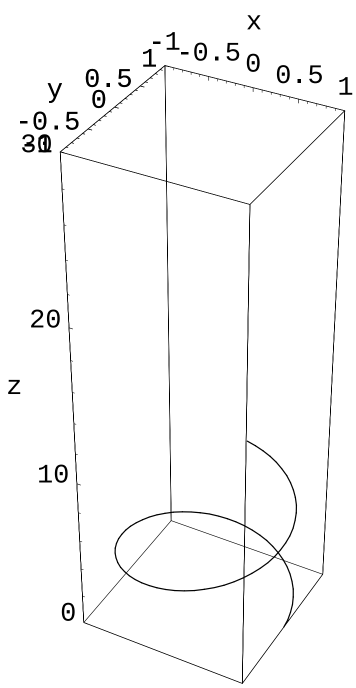

x
-1 -0.5 0 0.5 1
y
1 0.5 0 -0.5 -1
z
0
10
20
30

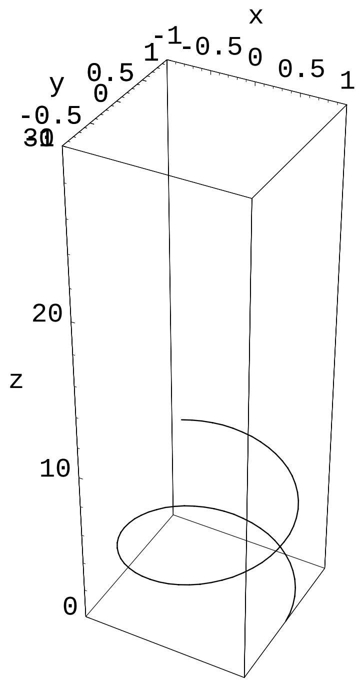

x
-1 -0.5 0 0.5 1
y
1 0.5 0 -0.5 -1
z
0
10
20
30

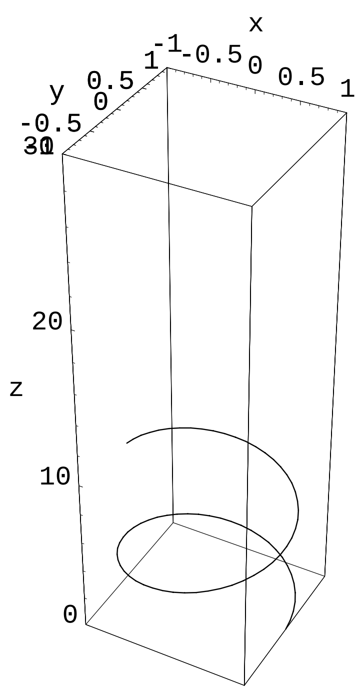

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
z
30
20
10
0

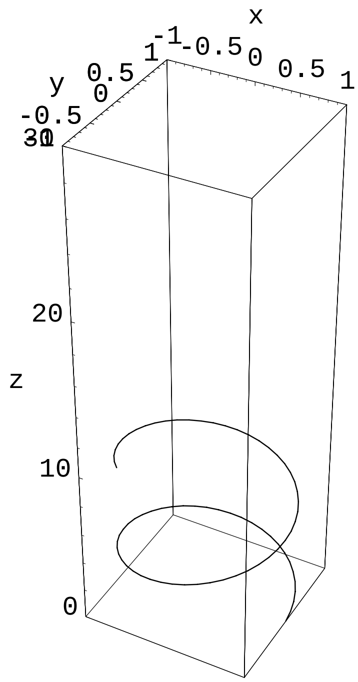

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
z
30
20
10
0

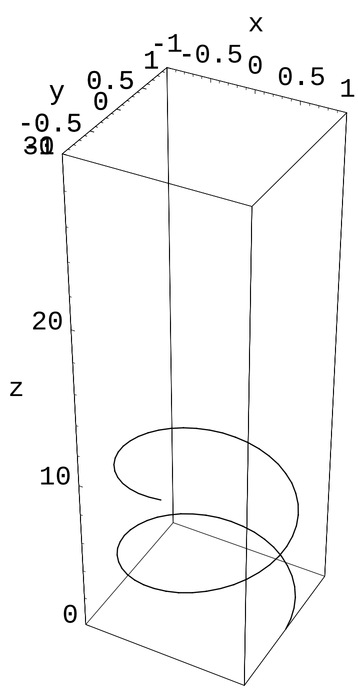

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

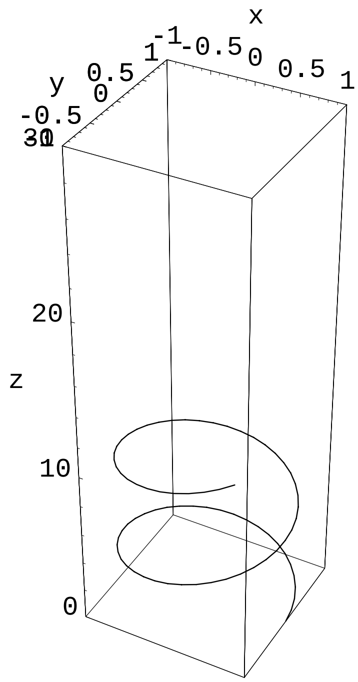

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

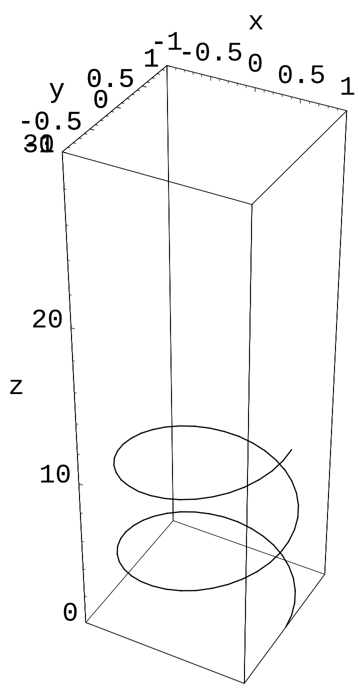

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

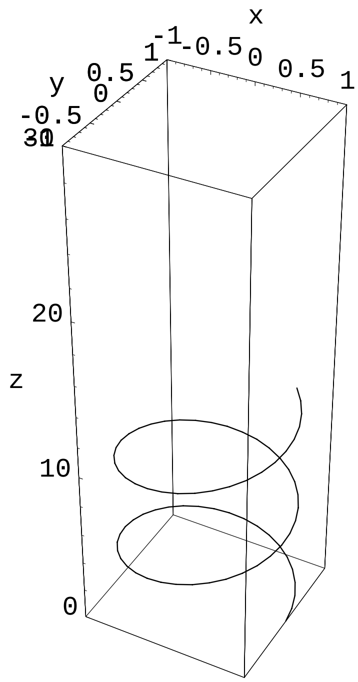

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

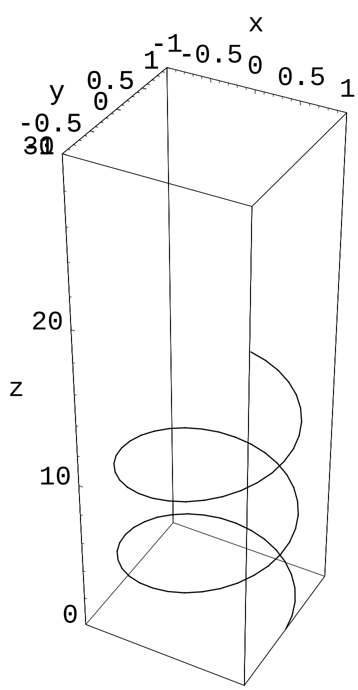

x
-1
-0.5
0
0.5
1
1
0.5
y
0
-0.5
30
20
z
10
0

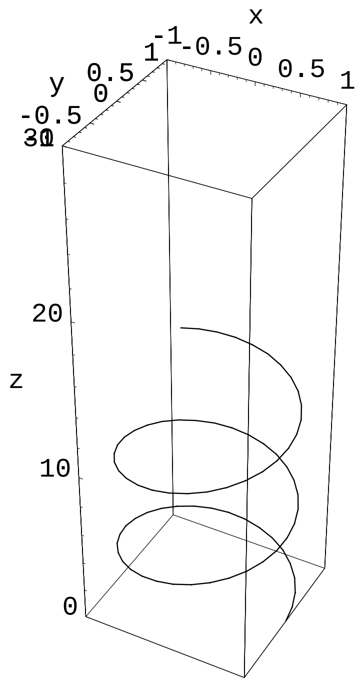

x
-1
-0.5
0
0.5
1
1
0.5
y
0
-0.5
30
20
z
10
0

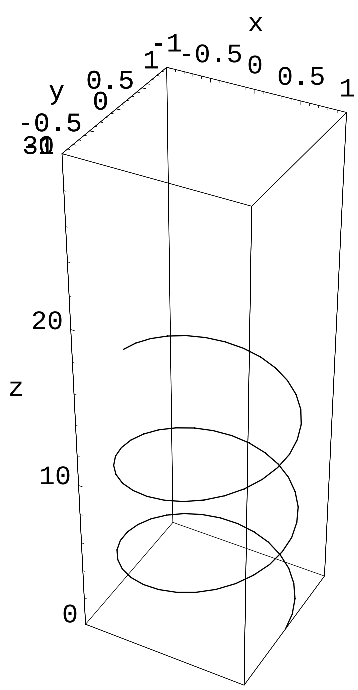

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0

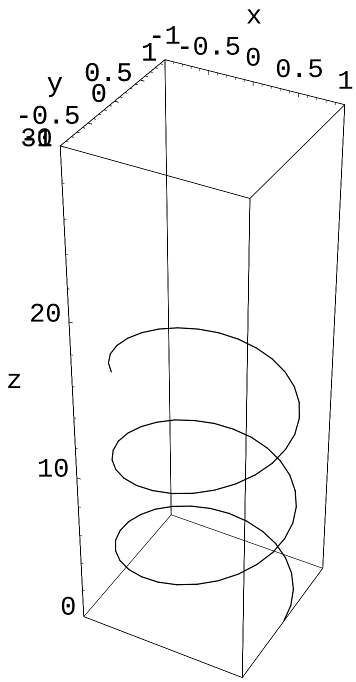

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0

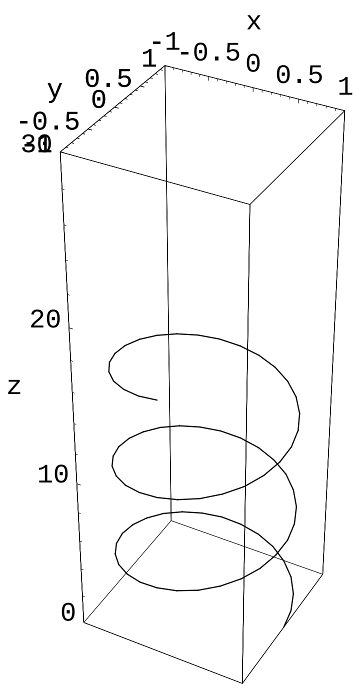

x
-1 -0.5 0 0.5 1
y
1 0.5 0 -0.5 -1
30
20
z
10
0

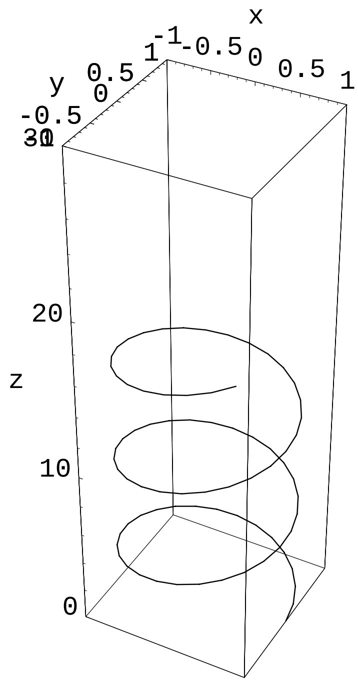

x
-1 -0.5 0 0.5 1
y
1 0.5 0 -0.5 -1
30
20
z
10
0

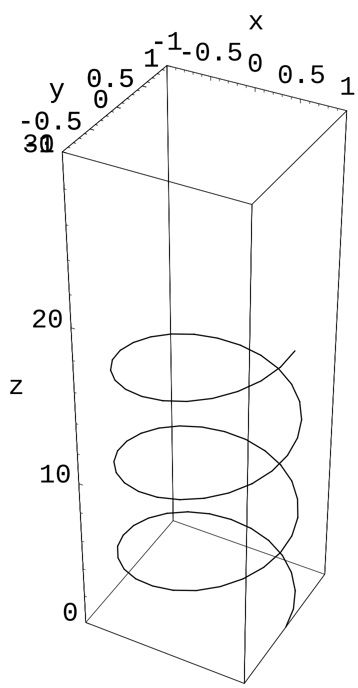

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

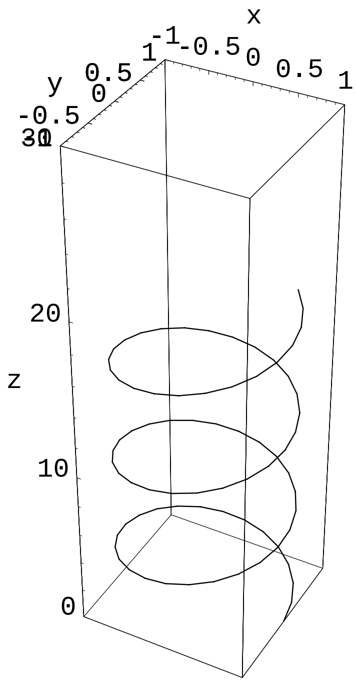

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0

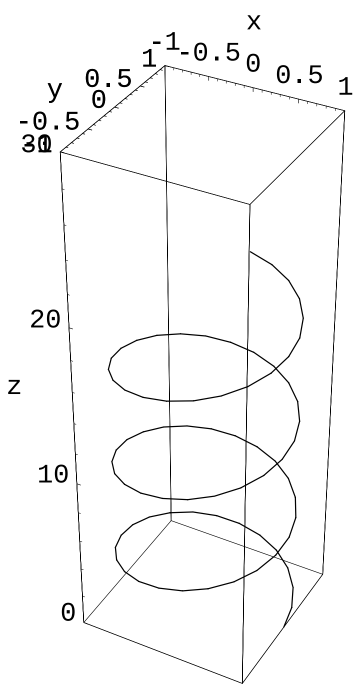

x
-1
-0.5
0
0.5
1
1
0.5
y
0
-0.5
30
20
z
10
0

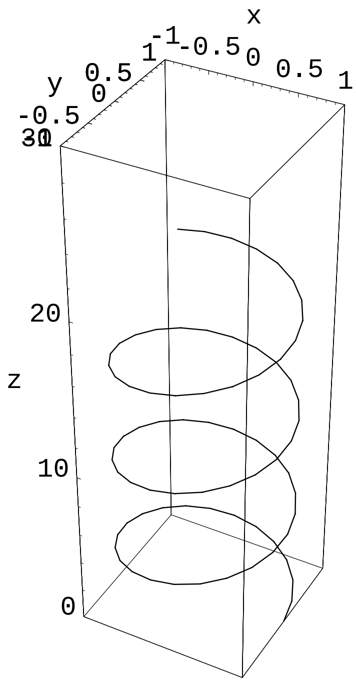

x
-1
-0.5
0
0.5
1
1
0.5
y
0
-0.5
30
20
z
10
0

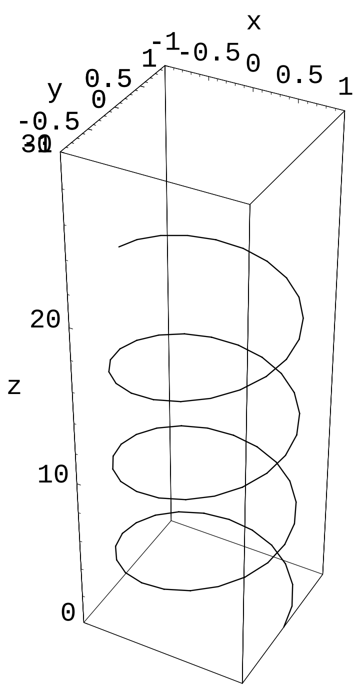

x
-1
-0.5
0
0.5
1
y
0.5
1
0
-0.5
30
20
z
10
0

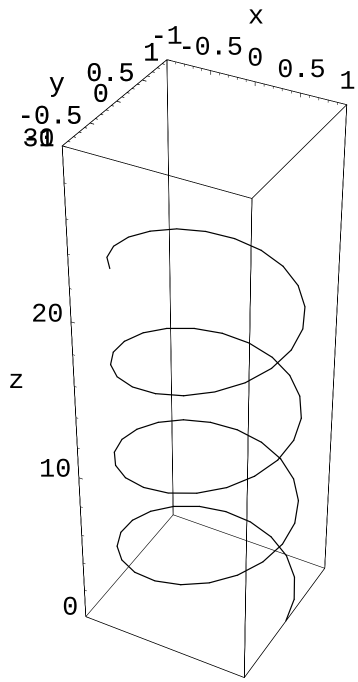

x
-1
-0.5
0
0.5
1
y
0.5
1
0
-0.5
30
20
z
10
0

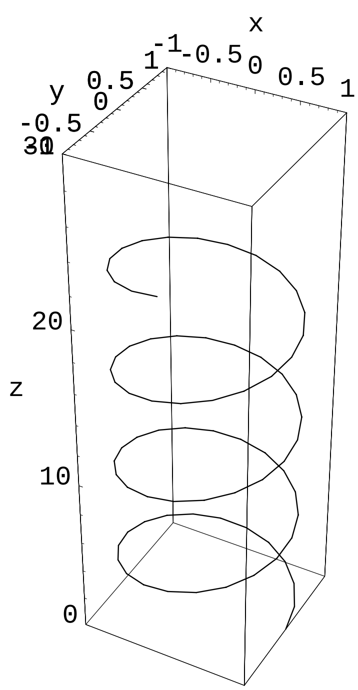

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0

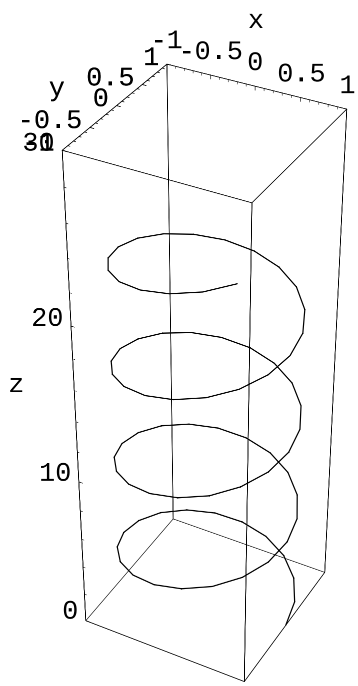

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0

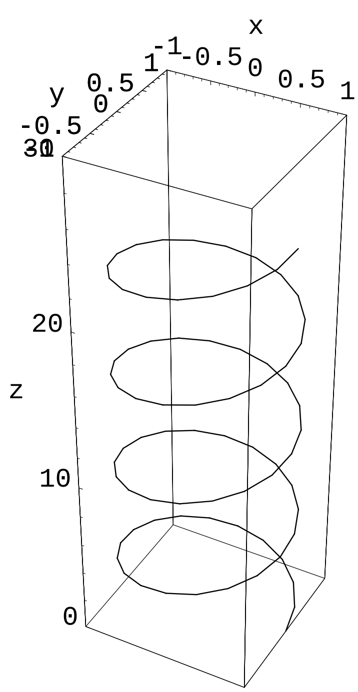

x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0


x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
30
20
z
10
0


x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
z
30
20
10
0


x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
z
30
20
10
0


x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0


x
-1
-0.5
0
0.5
1
y
1
0.5
0
-0.5
-1
z
30
20
10
0

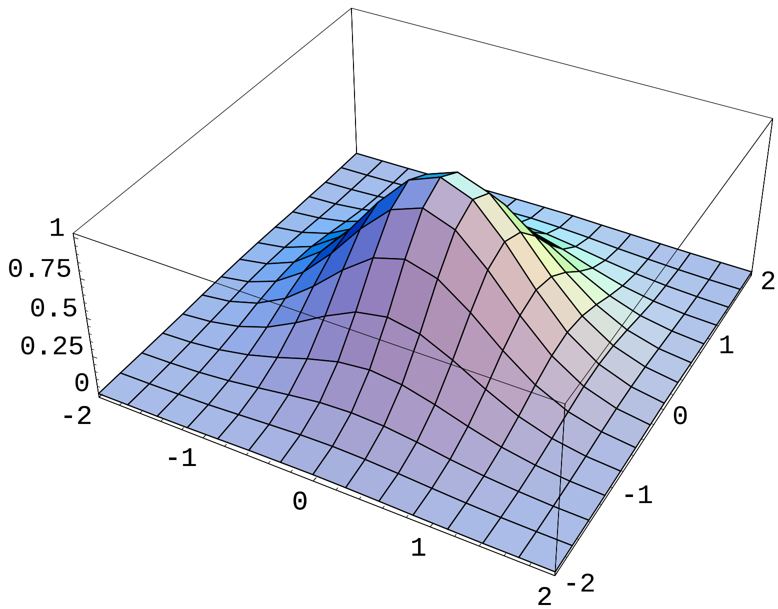
■ Surfaces

Mathematica is a great tool for helping you visualize 3-dimensional surfaces. This portion of the notebook will provide you with a few examples.

Level Curves

The next input cell provides you with an example of how to define and plot a surface $z = f(x, y)$.

In[51]:=
```
f[x_, y_] = e^-(x^2+y^2);
bell = Plot3D[f[x, y], {x, -2, 2}, {y, -2, 2}]
```



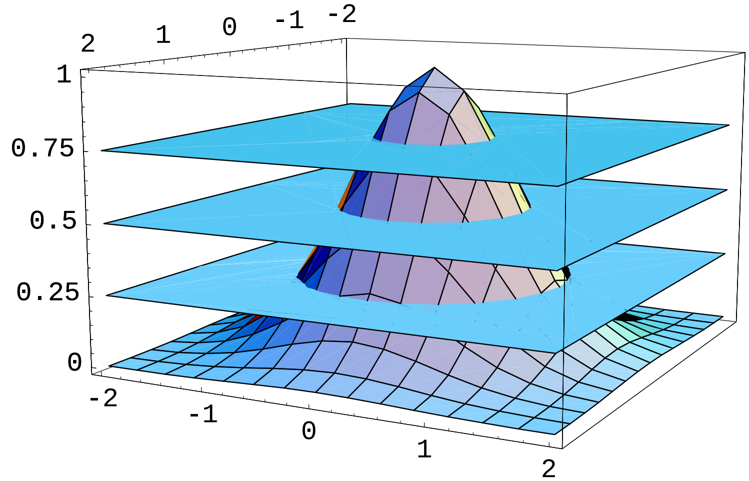
Out[52]= - SurfaceGraphics -

Level curves occur when horizontal planes intersect a surface. In the following input cell, three plots of horizontal planes $z = 0.25$, $z = 0.5$ and $z = 0.75$ are formed.

In[53]:=
```
pl1 = Plot3D[.25, {x, -2, 2}, {y, -2, 2},
   PlotPoints → 2, DisplayFunction → Identity];
pl2 = Plot3D[.5, {x, -2, 2}, {y, -2, 2}, PlotPoints → 2,
   DisplayFunction → Identity];
pl3 = Plot3D[.75, {x, -2, 2}, {y, -2, 2}, PlotPoints → 2,
   DisplayFunction → Identity];
```

Now the `Show` command is used to plot these planes together with the bell-shaped surface above. Note that the intersection of each plane and the surface appears to be a circle.

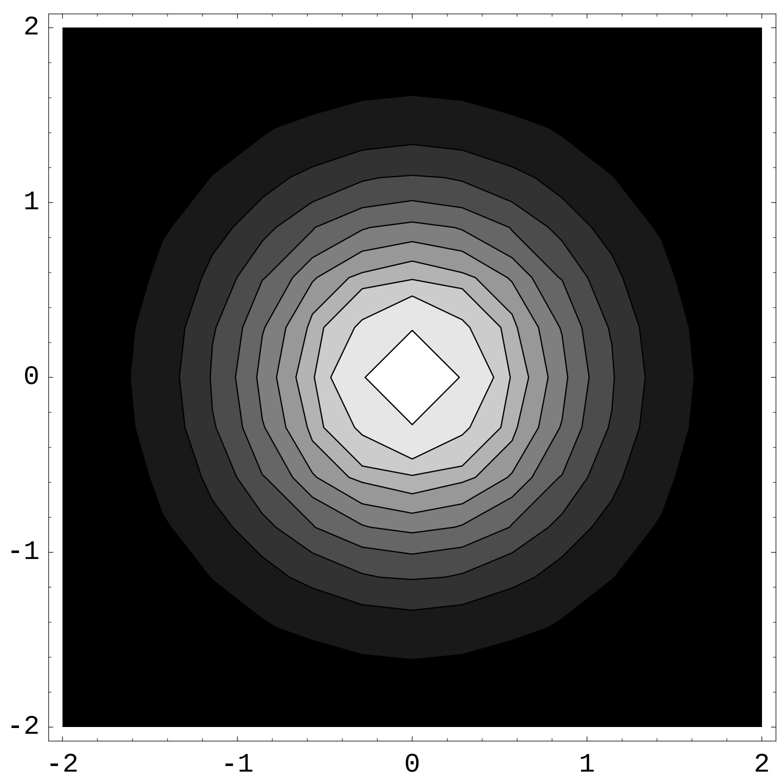
```
In[56]:= Show[{bell, pl1, pl2, pl3}, DisplayFunction → $DisplayFunction,
          ViewPoint -> {2.908, 1.499, 0.563}, BoxRatios → {1, 1, .6}]
```




```
Out[56]= - Graphics3D -
```

The `ContourPlot` command can be used to visualize the level curves. The lighter-shaded surfaces correspond to higher points on the surface.

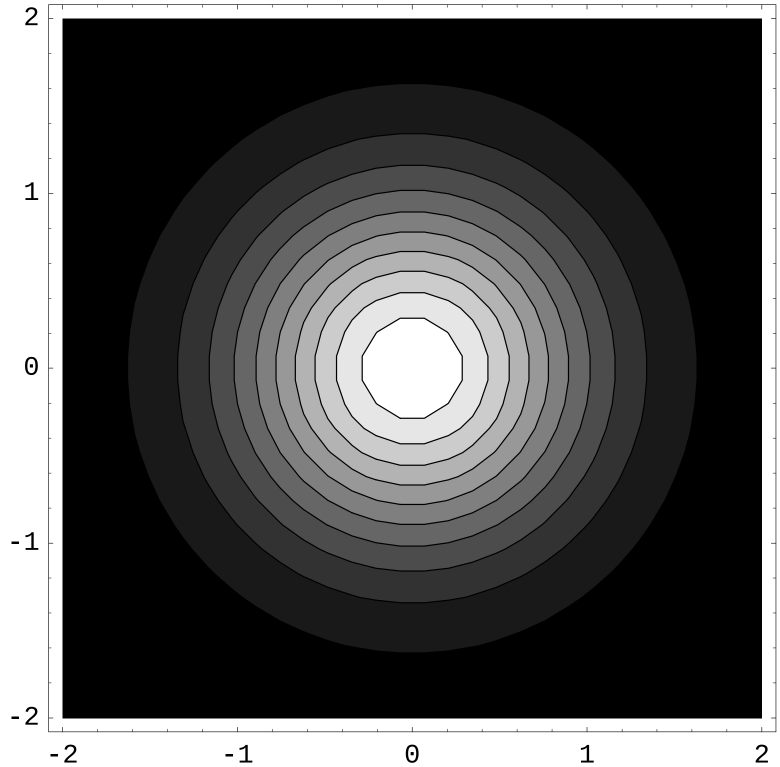
```
In[57]:= ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 2}]
```




```
Out[57]= - ContourGraphics -
```

The graph above is a bit choppy - the `PlotPoints -> n` option makes the curves smoother.

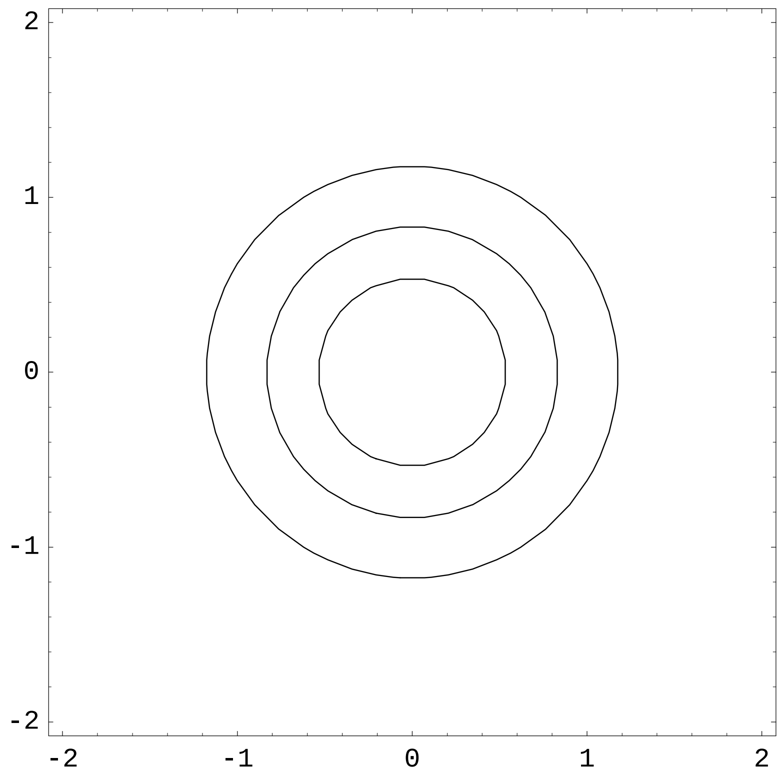
In[58]:= `ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 2}, PlotPoints → 30]`



Out[58]= `- ContourGraphics -`

The option `ContourShading` → `False` can be added to eliminate the shading and the option Contours → {.25, .5, .75} is added so that only the contour curves corresponding to $z = 0.25$, $z = 0.5$ and $z = 0.75$ are formed.

In[59]:=
```
ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 2},
 Contours → {.25, .5, .75}, ContourShading → False, PlotPoints → 30]
```



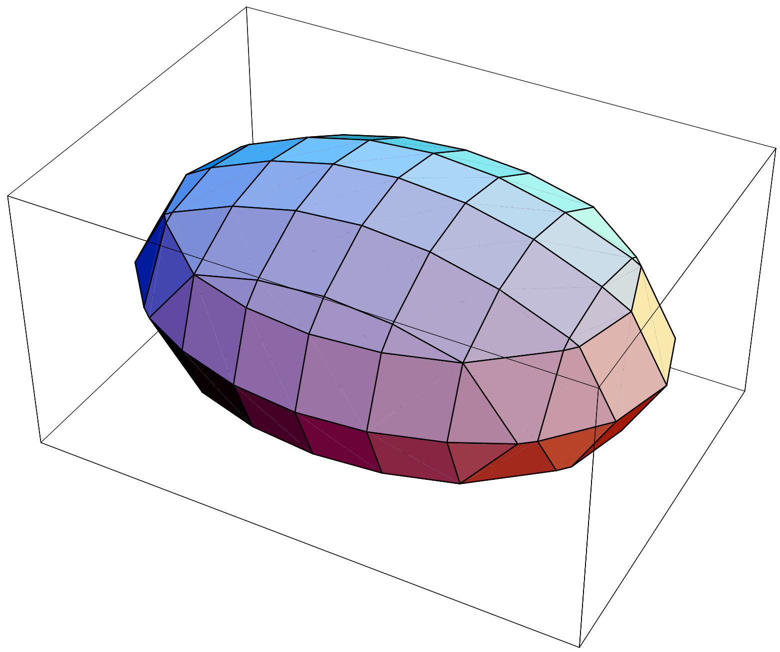
Out[59]= - ContourGraphics -

Level Surfaces

The command `ContourPlot3D[f[x, y, z], {x, a, b}, {y, c, d}, {z, e, f}, Contours → {k}]`, contained in the add-on package *ContourPlot3D*, will plot the level surface $f(x, y, z) = k$.

In[60]:=
```
<< Graphics`ContourPlot3D`
```

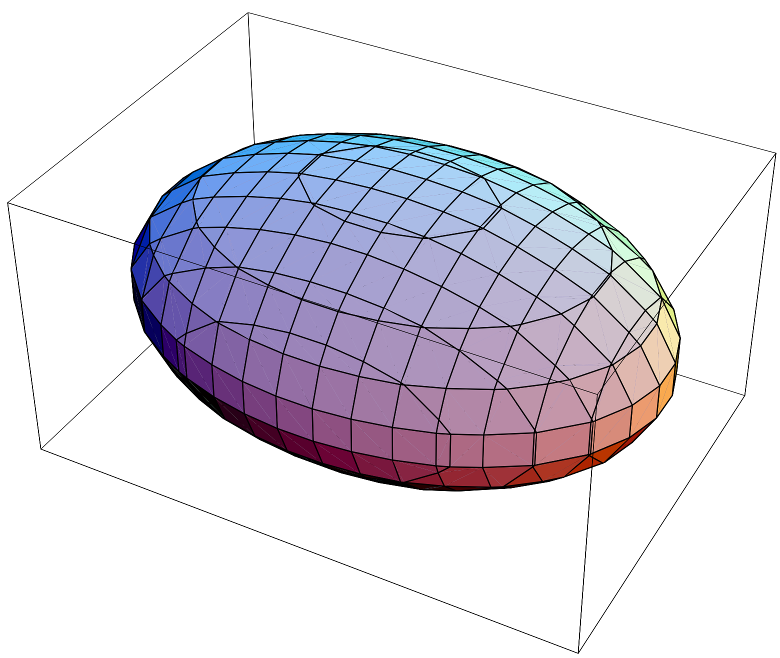
```
In[61]:= ContourPlot3D[x^2 + 2 y^2 + 4 z^2, {x, -4, 4}, {y, -4, 4}, {z, -4, 4}, Contours → {16}]
```



```
Out[61]= - Graphics3D -
```

The PlotPoints option below increases the number of points plotted so that the surface looks more smooth.

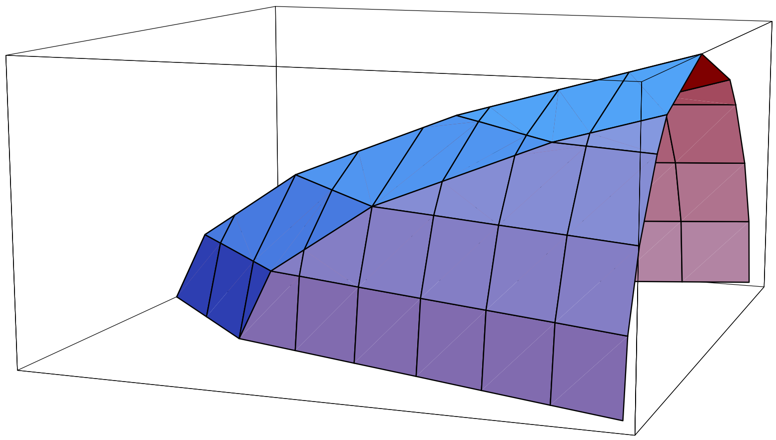
```
In[62]:= ContourPlot3D[x^2 + 2 y^2 + 4 z^2, {x, -4, 4}, {y, -4, 4}, {z, -4, 4},
          Contours → {16}, PlotPoints → {4, 6}, AxesLabel → {x, y, z}]
```



```
Out[62]= - Graphics3D -
```

Next, a plot of the surface $z = \sqrt{y - x^2}$ is obtained by first squaring both sides of the equation and then rearranging the equation to obtain $x^2 - y + z^2 = 0$, where $z \geq 0$.

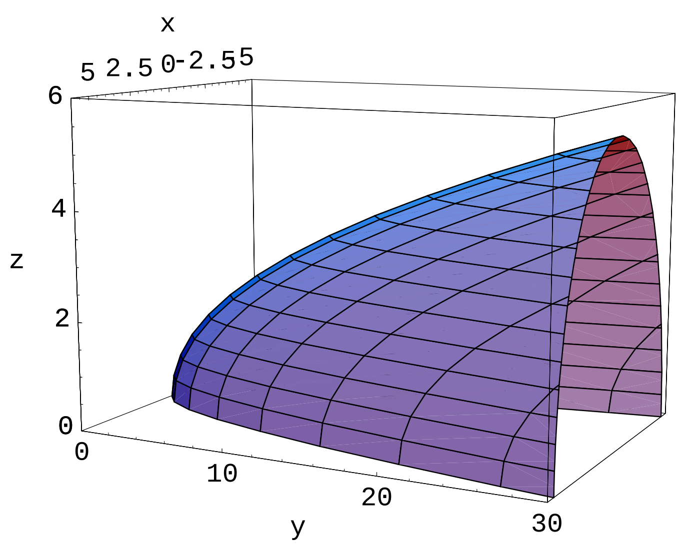
In[63]:= **ContourPlot3D[x^2 - y + z^2, {x, -4, 4}, {y, 0, 4}, {z, 0, 4}, Contours → {0}, PlotPoints → {3, 5}, AxesLabel → {x, y, z}, ViewPoint -> {3.072, 1.125, 0.563}]**



Out[63]= - Graphics3D -

Another method for plotting the surface above is described here. The command `ParametricPlot3D[{f[u, v], g[u, v], h[u, v]}, {y, a, b}, {v, c, d}]` is used to plot the surface defined parametrically by $x = f(u, v)$, $y = g(u, v)$, $z = h(u, v)$. So, if $x^2 - y + z^2 = 0$, then $y = x^2 + z^2$. So if $x = f(u, v) = u$ and $z = g(u, v) = v$, then $y = h(u, v) = u^2 + v^2$.

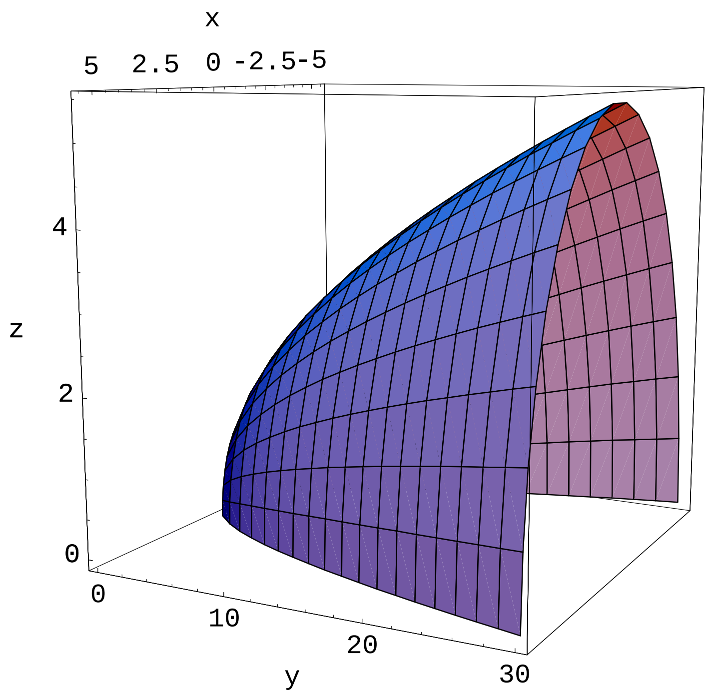
In[64]:= **Clear[u, v];**
ParametricPlot3D[{u, $u^2 + v^2$, v}, {v, 0, 6}, {u, -10, 10}, BoxRatios → {1, 1.5, 1}, ViewPoint -> {2.908, 1.499, 0.563}, PlotRange → {{-6, 6}, {0, 30}, {0, 6}}, PlotPoints → {15, 30}, AxesLabel → {x, y, z}]



Out[65]= - Graphics3D -

Still, another way to plot the graph is to first see that $\frac{x^2}{y} + \frac{z^2}{y} = 1$ and note that this equation is satisfied if $\frac{x^2}{y} = \sin^2(t)$ and $\frac{z^2}{y} = \cos^2(t)$. Then this implies $x = \sqrt{y}\sin(t)$ and $z = \sqrt{y}\cos(t)$. We can then use these equations to graph the corresponding surface.

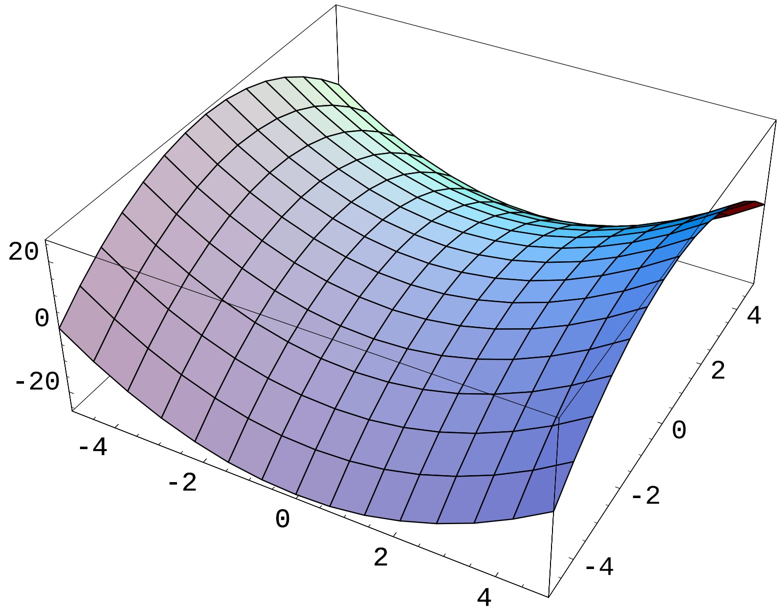
```
In[66]:= ParametricPlot3D[{√y Sin[t], y, √y Cos[t]},
  {y, 0, 30}, {t, -π/2, π/2}, ViewPoint -> {2.908, 1.499, 0.563},
  BoxRatios → {1, 1, 1}, AxesLabel → {x, y, z}]
```




```
Out[66]= - Graphics3D -
```

The command `Plot3D` can also be used to plot some surfaces of the form $z = f(x, y)$. Here is an example where the hyperbolic paraboloid $z = x^2 - y^2$ is plotted.

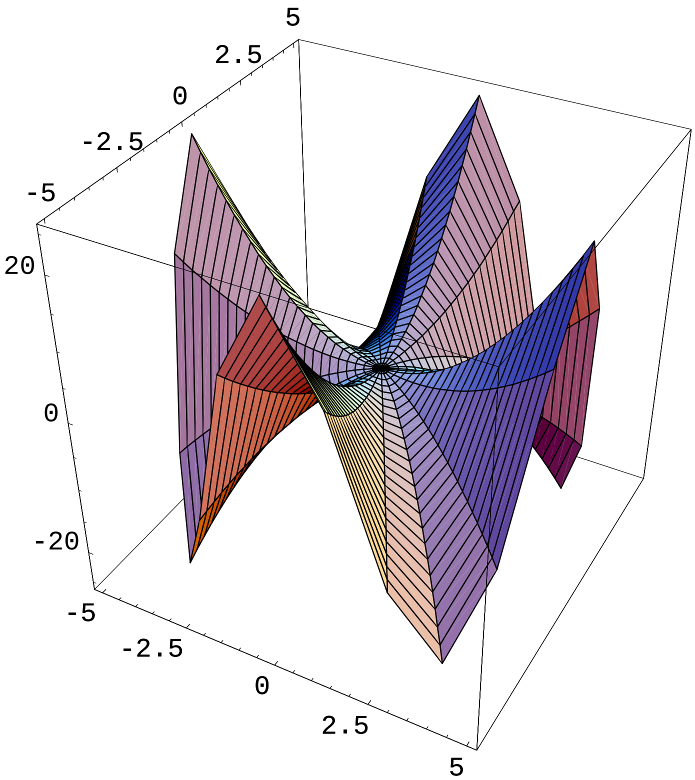
In[67]:= **Plot3D**[x^2 - y^2, {x, -5, 5}, {y, -5, 5}]



Out[67]= - SurfaceGraphics -

Here is an example where the `ParametricPlot3D` command is used to plot the surface corresponding to Example 4 of Section 9.5 in your text.

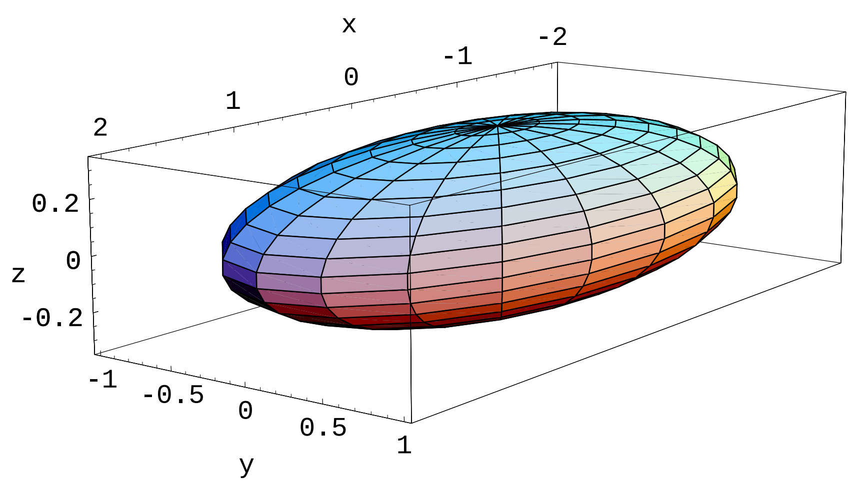
```
In[68]:= ParametricPlot3D[{r Cos[t], r Sin[t], r^2 Cos[4 t]},
          {t, 0, 2 π}, {r, 0, 5}, BoxRatios → {1, 1, 1}, PlotPoints → 25]
```




```
Out[68]= - Graphics3D -
```

The following example demonstrates how to plot the graph of the equation described in Example 10 of Section 9.5.

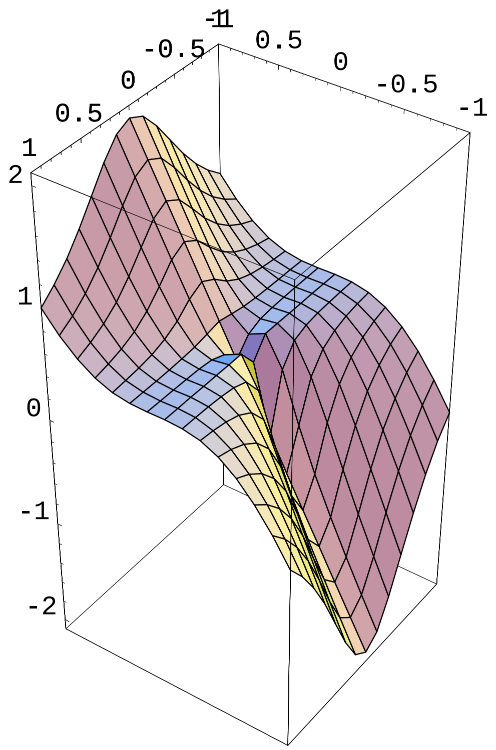
```
In[69]:= ParametricPlot3D[{2 Sin[ϕ] Cos[θ], Sin[ϕ] Sin[θ], 1/3 Cos[ϕ]},
          {θ, 0, 2 π}, {ϕ, 0, π}, BoxRatios → {2, 1, .5},
          AxesLabel → {x, y, z}, ViewPoint -> {2.573, 1.893, 0.666}]
```




```
Out[69]= - Graphics3D -
```

Here is a plot of the discontinuous function found in Example 3 of Section 9.6 of your textbook.

In[70]:= **Plot3D**$\left[\frac{2\,x^3 + y^2}{x^2 + 2\,y^2}\right.$**, {x, -1, 1}, {y, -1, 1},**
BoxRatios → {1, 1, 2}, ViewPoint -> {-1.657, 2.346, 2.295}]



Out[70]= `- SurfaceGraphics -`

■ Partial Derivatives

Computing partial derivatives is demonstrated here using the function $f(x,\,y) = e^{-x^2-y^2}$.

In[71]:= **Clear[f];**
f[x_, y_] = $e^{-x^2-y^2}$

Out[72]= $e^{-x^2-y^2}$

To compute $\frac{\partial f}{\partial x}$, execute `D[f[x, y], x]` or use the palette button containing $\partial_\square\ \square$ and enter ∂_x `f[x, y]`.

In[73]:= **D[f[x, y], x]**

Out[73]= $-2\,e^{-x^2-y^2}\,x$

In[74]:= ∂_x **f[x, y]**

Out[74]= $-2\,e^{-x^2-y^2}\,x$

The following input and output statements demonstrates how to compute the second order partial derivative $f_{xy}(x,\,y)$.

In[75]:= `D[f[x, y], x, y]`

Out[75]= $4\, e^{-x^2-y^2}\, x\, y$

In[76]:= $\partial_{x,y}$ `f[x, y]`

Out[76]= $4\, e^{-x^2-y^2}\, x\, y$

Suppose you want to compute $f_x(.5, .75)$. You might be tempted to try the following.

In[77]:= ∂_x `f[.5, .75]`

Out[77]= 0

You do not get the correct value of $f_x(.5, .75)$ in the output above since *Mathematica* first replaces x and y with the values of .5 and .75, respectively, and then evaluates the derivative of a constant, which always equals 0. What you need to do is use the replacement operator `/.` after the derivative is evaluated.

In[78]:= ∂_x `f[x, y] /. {x → .5, y → .75}`

Out[78]= −0.443747

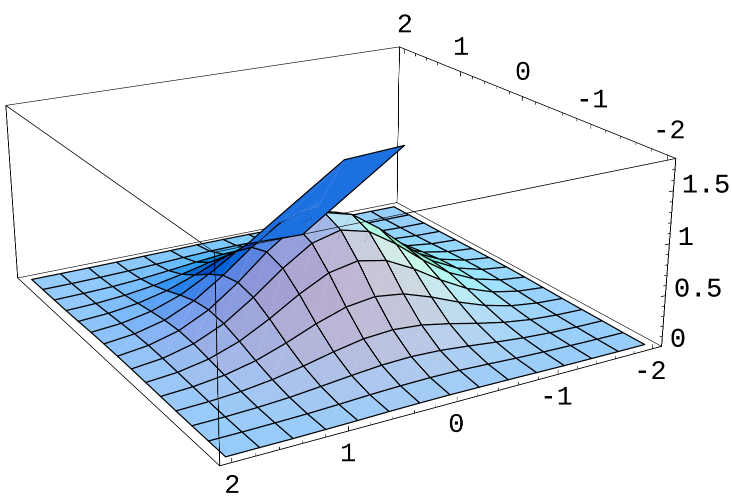
Suppose you want to find the equation of the plane tangent to $f(x, y) = e^{-x^2-y^2}$ at $(x, y) = (0.5, 0.75)$. Then all you need to do is apply equation (8) in Section 10.1.

```
In[79]:= z = f[.5, .75] + (∂x f[x, y] /. {x → .5, y → .75}) (x − .5) +
            (∂y f[x, y] /. {x → .5, y → .75}) (y − .75) // Simplify
```

Out[79]= 1.16484 − 0.443747 x − 0.665621 y

The plane and surface are plotted below. Study the input statement below.

```
In[80]:= pl = Plot3D[z, {x, −.5, .5}, {y, −.5, .5},
            PlotPoints → 2, DisplayFunction → Identity];
         bell = Plot3D[f[x, y], {x, −2, 2}, {y, −2, 2}, DisplayFunction → Identity];
         Show[{pl, bell}, DisplayFunction → $DisplayFunction,
          ViewPoint -> {-2.848, 1.647, 1.219}, PlotRange → {0, 1.8}]
```



Out[82]= - Graphics3D -

■ Double Integration

To evaluate a double integral such as $\int_{-1}^{2}\int_{y^2}^{y+2}(4-x)\,\mathrm{dx}\,\mathrm{dy}$, enter $\int_{\square}^{\square}\square\, d\square$ twice for the *BasicInput* palette. Use the **Tab** key to move from one box to the next.

In[83]:= $\int_{-1}^{2}\int_{y^2}^{y+2}(4-x)\,dx\,dy$

Out[83]= $\frac{54}{5}$

In[84]:= **N[%]**

Out[84]= 10.8