2.9.10 Coordinate Systems for Three-Dimensional Graphics

Whenever *Mathematica* draws a three-dimensional object, it always effectively puts a cuboidal box around the object. With the default option setting `Boxed -> True`, *Mathematica* in fact draws the edges of this box explicitly. But in general, *Mathematica* automatically "clips" any parts of your object that extend outside of the cuboidal box.

The option `PlotRange` specifies the range of x, y and z coordinates that *Mathematica* should include in the box. As in two dimensions the default setting is `PlotRange -> Automatic`, which makes *Mathematica* use an internal algorithm to try and include the "interesting parts" of a plot, but drop outlying parts. With `PlotRange -> All`, *Mathematica* will include all parts.

This loads a package defining various polyhedra.

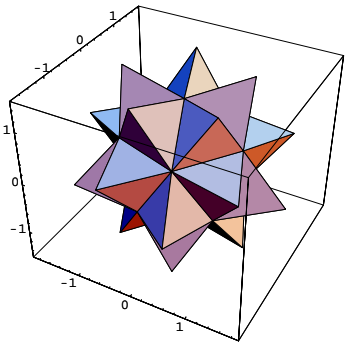
```
In[1]:= <<Graphics`Polyhedra` ;
```

This creates a stellated icosahedron.

```
In[2]:= stel = Stellate[Icosahedron[ ]] ;
```

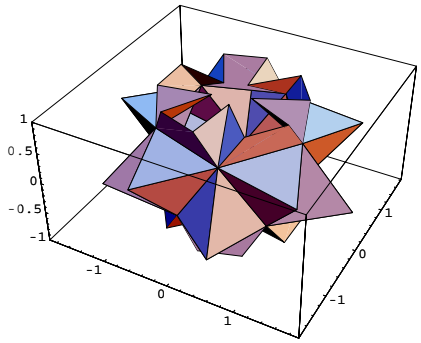
Here is the stellated icosahedron, drawn in a box.

```
In[3]:= Show[Graphics3D[stel], Axes -> True]
```



With this setting for `PlotRange`, many parts of the stellated icosahedron lie outside the box, and are clipped.

```
In[4]:= Show[%, PlotRange -> {-1, 1}]
```



Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

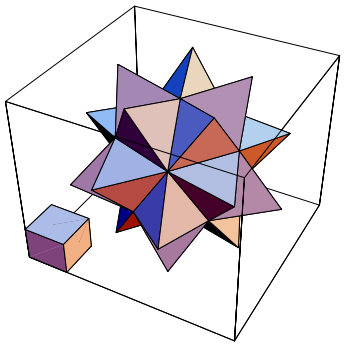
Much as in two dimensions, you can use either "original" or "scaled" coordinates to specify the positions of elements in three-dimensional objects. Scaled coordinates, specified as `Scaled[{sx, sy, sz}]` are taken to run from 0 to 1 in each dimension. The coordinates are set up to define a right-handed coordinate system on the box.

`{x, y, z}`	original coordinates
`Scaled[{sx, sy, sz}]`	scaled coordinates, running from 0 to 1 in each dimension

Coordinate systems for three-dimensional objects.

This puts a cuboid in one corner of the box.

```
In[5]:= Show[Graphics3D[{stel,
          Cuboid[Scaled[{0, 0, 0}],
            Scaled[{0.2, 0.2, 0.2}]]}]]
```



Once you have specified where various graphical elements go inside a three-dimensional box, you must then tell *Mathematica* how to draw the box. The first step is to specify what shape the box should be. This is analogous to specifying the aspect ratio of a two-dimensional plot. In three dimensions, you can use the option `BoxRatios` to specify the ratio of side lengths for the box. For `Graphics3D` objects, the default is `BoxRatios -> Automatic`, specifying that the shape of the box should be determined from the ranges of actual coordinates for its contents.

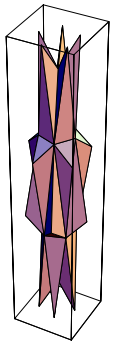
`BoxRatios -> {xr, yr, zr}`	specify the ratio of side lengths for the box
`BoxRatios -> Automatic`	determine the ratio of side length from the range of actual coordinates (default for `Graphics3D`)
`BoxRatios -> {1, 1, 0.4}`	specify a fixed shape of box (default for `SurfaceGraphics`)

Specifying the shape of the bounding box for three-dimensional objects.

Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

This displays the stellated icosahedron in a tall box.

```
In[6]:= Show[Graphics3D[stel], BoxRatios -> {1, 1, 5}]
```



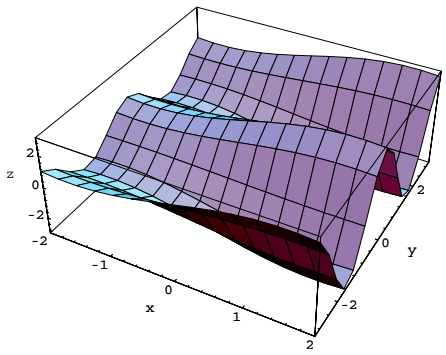
To produce an image of a three-dimensional object, you have to tell *Mathematica* from what view point you want to look at the object. You can do this using the option `ViewPoint`.

Some common settings for this option were given on page 184. In general, however, you can tell *Mathematica* to use any view point, so long as it lies outside the box.

View points are specified in the form `ViewPoint -> {`*sx*`, `*sy*`, `*sz*`}`. The values *si* are given in a special coordinate system, in which the center of the box is `{0, 0, 0}`. The special coordinates are scaled so that the longest side of the box corresponds to one unit. The lengths of the other sides of the box in this coordinate system are determined by the setting for the `BoxRatios` option. For a cubical box, therefore, each of the special coordinates runs from $-1/2$ to $1/2$ across the box. Note that the view point must always lie outside the box.

This generates a picture using the default view point `{1.3, -2.4, 2}`.

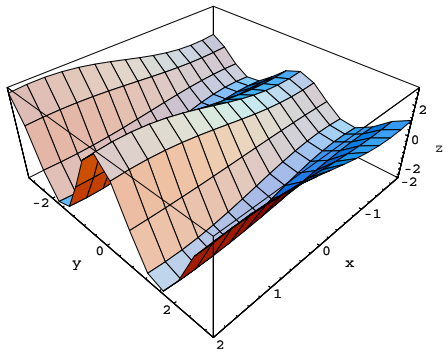
```
In[7]:= surf = Plot3D[(2 + Sin[x]) Cos[2 y],
                      {x, -2, 2}, {y, -3, 3},
                        AxesLabel -> {"x", "y", "z"}]
```



Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

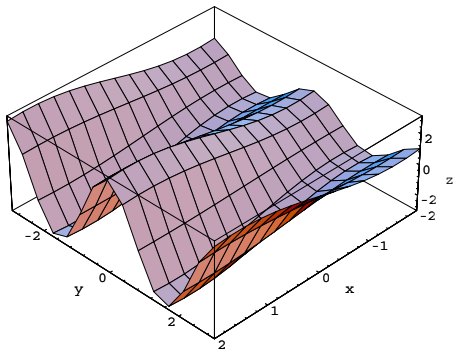
This is what you get with a view point close to one of the corners of the box.

```
In[8]:= Show[%, ViewPoint -> {1.2, 1.2, 1.2}]
```



As you move away from the box, the perspective effect gets smaller.

```
In[9]:= Show[%, ViewPoint -> {5, 5, 5}]
```



option name	*default value*	
`ViewPoint`	`{1.3, -2.4, 2}`	the point in a special scaled coordinate system from which to view the object
`ViewCenter`	`Automatic`	the point in the scaled coordinate system which appears at the center of the final image
`ViewVertical`	`{0, 0, 1}`	the direction in the scaled coordinate system which appears as vertical in the final image

Specifying the position and orientation of three-dimensional objects.

Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

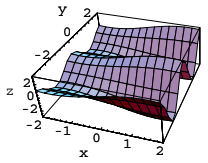
In making a picture of a three-dimensional object you have to specify more than just *where* you want to look at the object from. You also have to specify how you want to "frame" the object in your final image. You can do this using the additional options `ViewCenter` and `ViewVertical`.

`ViewCenter` allows you to tell *Mathematica* what point in the object should appear at the center of your final image. The point is specified by giving its scaled coordinates, running from 0 to 1 in each direction across the box. With the setting `ViewCenter -> {1/2, 1/2, 1/2}`, the center of the box will therefore appear at the center of your final image. With many choices of view point, however, the box will not appear symmetrical, so this setting for `ViewCenter` will not center the whole box in the final image area. You can do this by setting `ViewCenter -> Automatic`.

`ViewVertical` specifies which way up the object should appear in your final image. The setting for `ViewVertical` gives the direction in scaled coordinates which ends up vertical in the final image. With the default setting `ViewVertical -> {0, 0, 1}`, the z direction in your original coordinate system always ends up vertical in the final image.

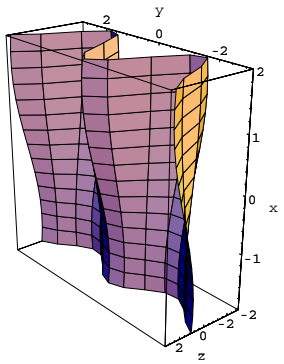
With this setting for `ViewCenter`, a corner of the box appears in the center of your image.

```
In[10]:= Show[surf, ViewCenter -> {1, 1, 1}]
```



This setting for `ViewVertical` makes the x axis of the box appear vertical in your image.

```
In[11]:= Show[surf, ViewVertical -> {1, 0, 0}]
```



Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

When you set the options ViewPoint, ViewCenter and ViewVertical, you can think about it as specifying how you would look at a physical object. ViewPoint specifies where your head is relative to the object. ViewCenter specifies where you are looking (the center of your gaze). And ViewVertical specifies which way up your head is.

In terms of coordinate systems, settings for ViewPoint, ViewCenter and ViewVertical specify how coordinates in the three-dimensional box should be transformed into coordinates for your image in the final display area.

For some purposes, it is useful to think of the coordinates in the final display area as three dimensional. The x and y axes run horizontally and vertically, respectively, while the z axis points into the page. Positions specified in this "display coordinate system" remain fixed when you change ViewPoint and so on. The positions of light sources discussed in the next section are defined in this display coordinate system.

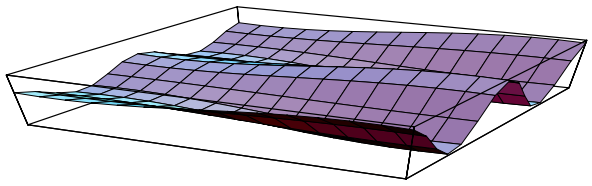
Box coordinate system	measured relative to the box around your object
Display coordinate system	measured relative to your final display area

Coordinate systems for three-dimensional graphics.

Once you have obtained a two-dimensional image of a three-dimensional object, there are still some issues about how this image should be rendered. The issues however are identical to those that occur for two-dimensional graphics. Thus, for example, you can modify the final shape of your image by changing the AspectRatio option. And you specify what region of your whole display area your image should take up by setting the PlotRegion option.

This modifies the aspect ratio of the final image.

```
In[12]:= Show[surf, Axes -> False, AspectRatio -> 0.3]
```



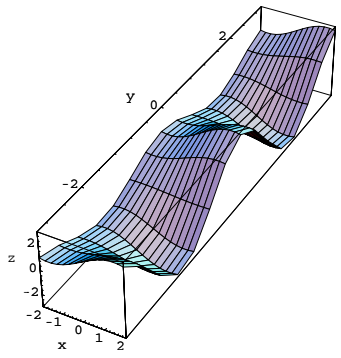
Mathematica usually scales the images of three-dimensional objects to be as large as possible, given the display area you specify. Although in most cases this scaling is what you want, it does have the conse-

Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

quence that the size at which a particular three-dimensional object is drawn may vary with the orientation of the object. You can set the option `SphericalRegion -> True` to avoid such variation. With this option setting, *Mathematica* effectively puts a sphere around the three-dimensional bounding box, and scales the final image so that the whole of this sphere fits inside the display area you specify. The sphere has its center at the center of the bounding box, and is drawn so that the bounding box just fits inside it.

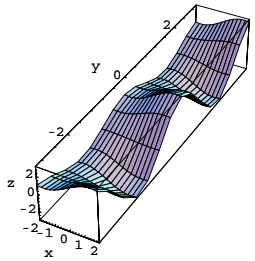
This draws a rather elongated version of the plot.

```
In[13]:= Show[surf, BoxRatios -> {1, 5, 1}]
```



With `SphericalRegion -> True`, the final image is scaled so that a sphere placed around the bounding box would fit in the display area.

```
In[14]:= Show[%, SphericalRegion -> True]
```



By setting `SphericalRegion -> True`, you can make the scaling of an object consistent for all orientations of the object. This is useful if you create animated sequences which show a particular object in several different orientations.

Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

`SphericalRegion -> False`	scale three-dimensional images to be as large as possible
`SphericalRegion -> True`	scale images so that a sphere drawn around the three-dimensional bounding box would fit in the final display area

Changing the magnification of three-dimensional images.

Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.