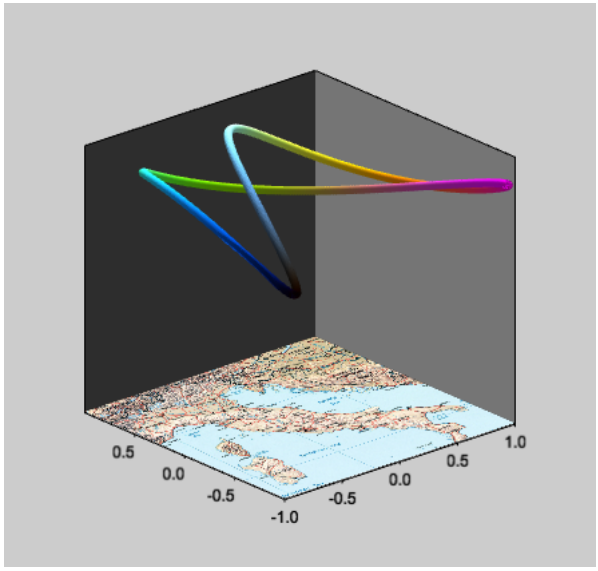
- A 2D matrix of Z values
- A 3D RGB wave of type unsigned byte with 3 layers
- A 3D RGBA wave of type unsigned byte with 4 layers

The latter two formats are created by the **ImageLoad** operation.

By default, the image is displayed at the bottom of the display volume but using rotation, translation and scaling options you can place the image anywhere within it. If you need to register the image relative to the data you can modify the axis range of the Gizmo display or change the scaling of the image. The scaling is uniform about the center of the image.

Igor includes a flight path example that combines a dense scatter plot with a Gizmo image object. In this case the image consists of a map:



To find out more open the Flight Path Demo experiment.

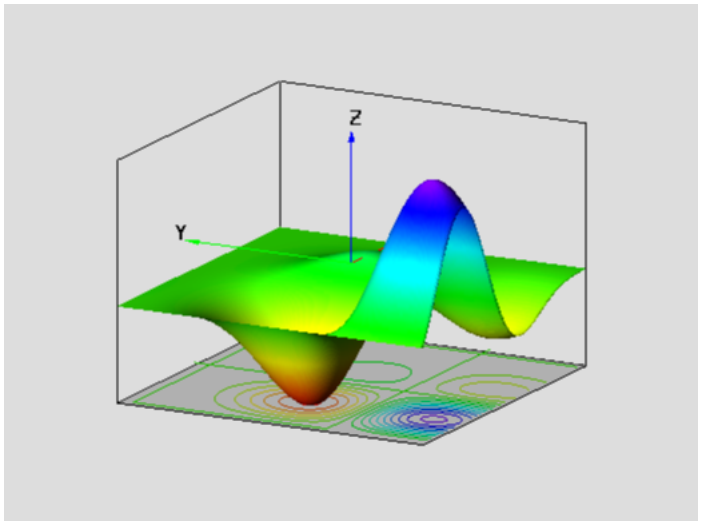
Surface Plots

A surface plot consists of a sheet connecting a grid of data values. You can specify surface colors from built-in color tables or custom color waves; see **Color Waves** on page II-430 for details. In addition to the surface, data values can also be displayed as points or as grid lines which can have their own color specification.

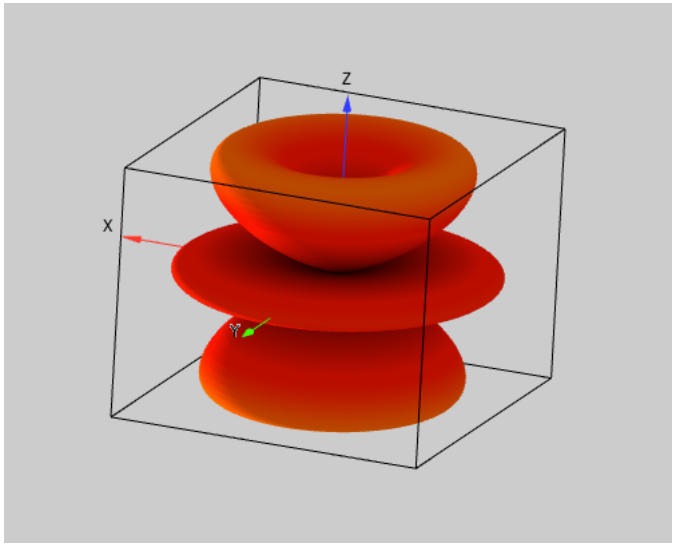
Typically data consist of an MxN 2D matrix of Z values which comprise the surface; see Surface Object Data Formats for details. Also supported are parametric surfaces which are 3D waves where each successive layer contains of the X, Y, and Z values in order; see **Parametric Surface Data Formats** on page II-452 for details.

The full list of available options is documented under **ModifyGizmo**.

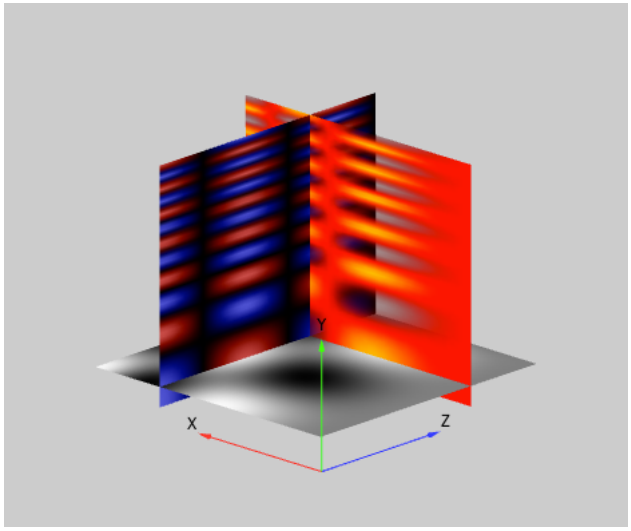
This example shows a surface plot with a contour map at the bottom:



This example shows a parametric surface plot of a spherical harmonic function:



This example shows orthogonal slices of volumetric data:



To display the data by sampling on non-orthogonal slices you can create an arbitrary parametric surface and color it using data sampled at the vertices of the parametric surface. Here is an example.

Suppose you have the following simple 3D data set:

```
Make/O/N=(100,100,100) ddd=z
```

Now create a parametric surface that describes a sphere in this range of data. Here is the MakeSphere function from the GizmoSphere demo experiment:

```
Function MakeSphere(pointsx,pointsy)
   Variable pointsx,pointsy

   Variable i,j,rad
   Make/O/N=(pointsx,pointsy,3) sphereData
   Variable anglePhi,angleTheta
   Variable dPhi,dTheta

   dPhi=2*pi/(pointsx-1)
   dTheta=pi/(pointsy-1)
   Variable xx,yy,zz
   Variable sig

   for(j=0;j<pointsy;j+=1)
      angleTheta=j*dTheta
      zz=sin(angleTheta)
      if(angleTheta>pi/2)
         sig=-1
      else
         sig=1
      endif
      for(i=0;i<pointsx;i+=1)
         anglePhi=i*dPhi
         xx=zz*cos(anglePhi)
         yy=zz*sin(anglePhi)
         sphereData[i][j][0]=xx
         sphereData[i][j][1]=yy
         sphereData[i][j][2]=sig*sqrt(1-xx*xx-yy*yy)
      endfor
   endfor
End
```

You can execute the function like this:

```
MakeSphere(100,100)
```

Then shift the result to the limits of the sample data using

```
sphereData*=48          // Slightly inside the boundary
sphereData+=50
```

Next create a scale wave that will contain the samples of the data at the vertices of the parametric surface:

```
Make/N=(100,100)/O scaleWave
scaleWave=Interp3D(ddd,sphereData[p][q][0],sphereData[p][q][1],sphereData[p][q][2])
```

To create a color wave we first open a Gizmo window and then have Gizmo create the color wave:

```
NewGizmo
AppendToGizmo surface=root:sphereData,name=surface0
ModifyGizmo modifyObject=surface0,objectType=surface,property={ srcMode,4}
ModifyGizmo makeParametricColorWave={sphereData,scaleWave,Rainbow,0}
```