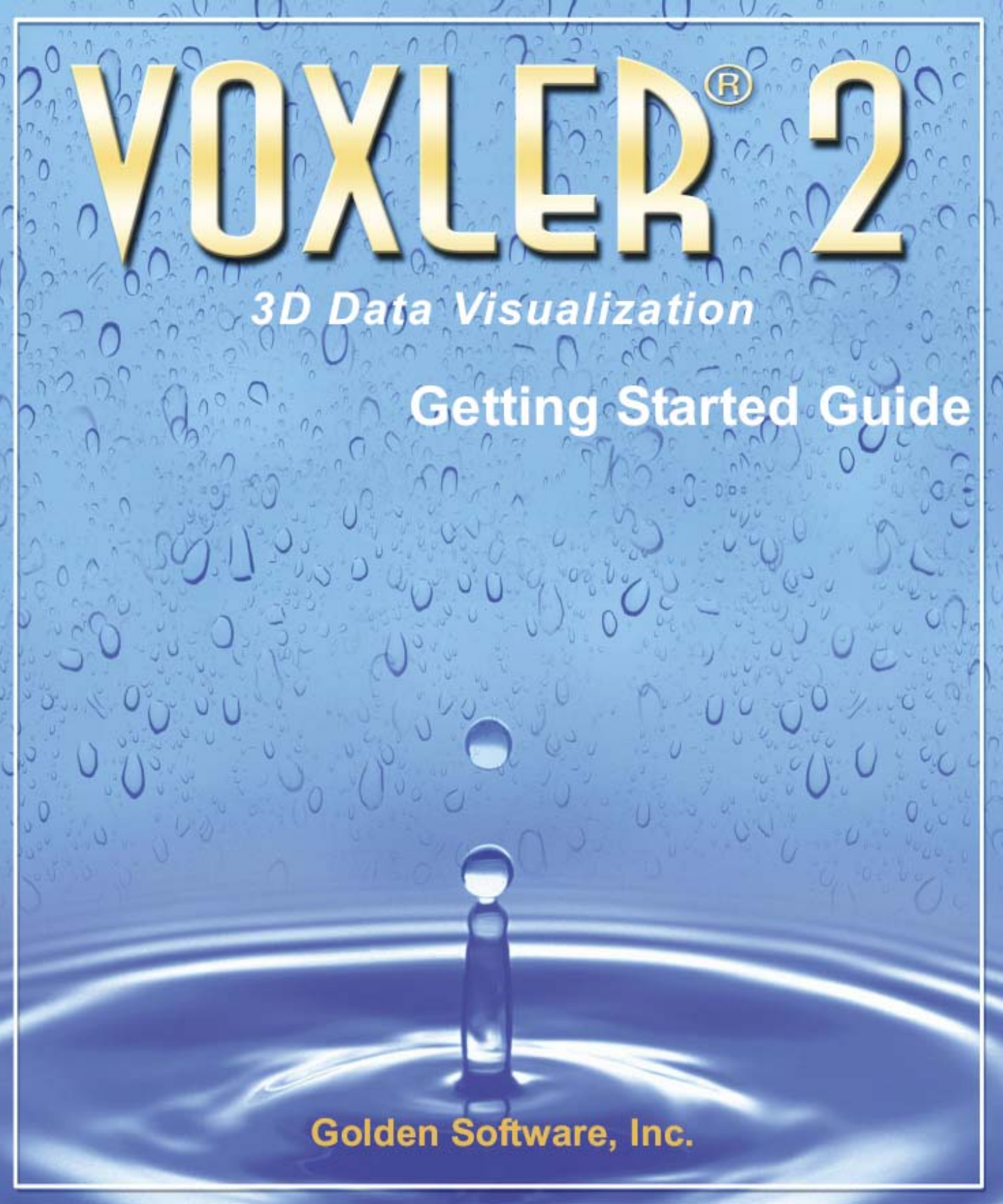
VOXLER® 2

3D Data Visualization

Getting Started Guide

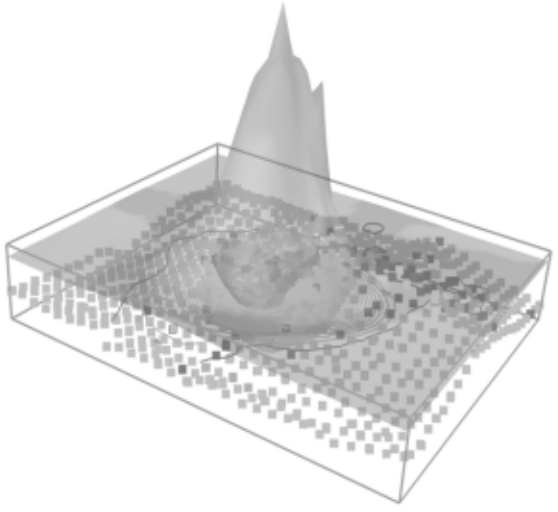


Golden Software, Inc.

Voxler®

Getting Started Guide

3D Data Visualization



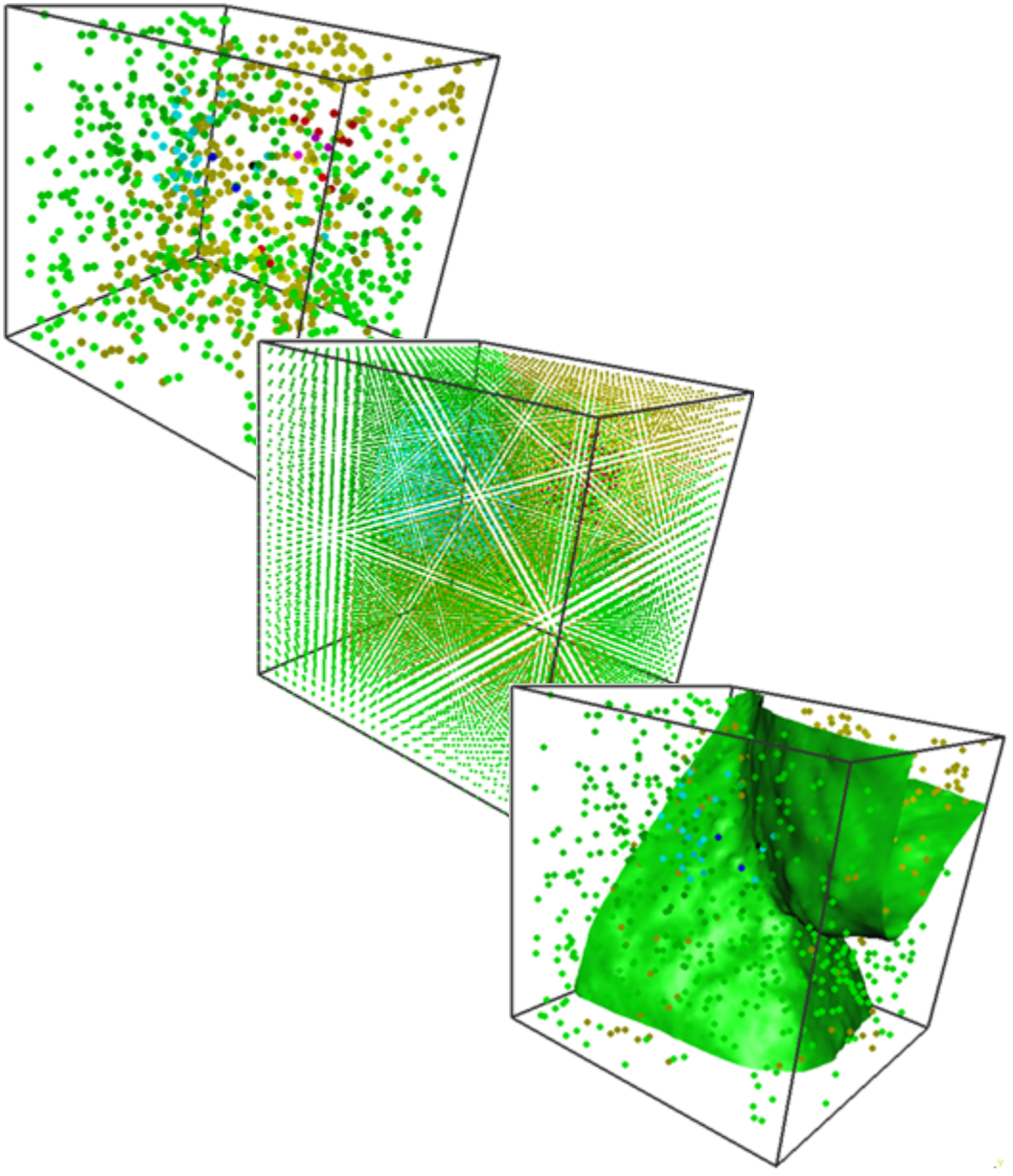
Golden Software, Inc.
809 14th Street, Golden, Colorado 80401-1866, U.S.A.
Phone: 303-279-1021 Fax: 303-279-0909
www.goldensoftware.com

COPYRIGHT NOTICE

Title page and back cover image courtesy of Bernhard Hochwimmer, Dart Mining, Australia

May 2010

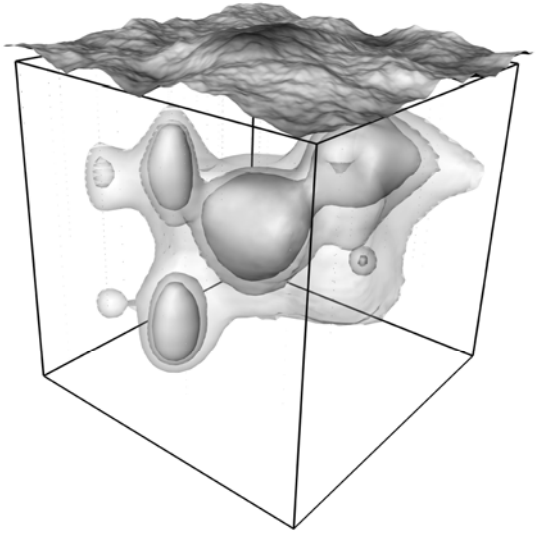
Table of Contents



Introduction to Voxler

Welcome to **Voxler**, a three-dimensional scientific visualization program, oriented primarily toward volumetric rendering and three-dimensional data display. While the emphasis is on three-dimensional volumes, **Voxler** can also utilize two-dimensional grids including DEM files, images, and scattered point data. **Voxler** can display streamlines, vector plots, contour maps, isosurfaces, image slices, three-dimensional scatter plots, direct volume rendering, and more. Computational modules include three-dimensional gridding, resampling, numerous lattice operations, and image processing. **Voxler** is designed for displaying XYZC data, where C is a variable at each X, Y, and Z location.

With **Voxler**, you can create stunning graphics output for your true three-dimensional models. Models can be sliced, displayed at any angle, and even animated with a simple mouse movement. Standard or custom colorization can be applied to the models.



Create stunning 3D graphics like this one by combining multiple map types.

Who Uses Voxler?

People from many different disciplines use **Voxler**. The geosciences generate large amounts of volumetric data from drill cores, seismic studies, ground penetrating radar, subsurface mapping, and remote sensing. Another source of data is from medical imaging generated by CT and MRI scans. Meteorological data, high-resolution microscopy, flow fields, and groundwater modeling are also sources for volumetric data. **Voxler** users include archeologists, climatologists, educators, engineers, doctors, hydrogeologists, geologists, geophysicists, medical researchers, students, and more. Anyone wanting to visualize the relationship of their three-dimensional data with stunning graphical output will benefit from **Voxler's** powerful features!

New features of **Voxler 2** are summarized:

- Online at: www.goldensoftware.com/products/voxler/voxlernew.shtml
- In the online help by using the **Help | Contents** command

System Requirements

The minimum system requirements for **Voxler** are:

- Windows XP, Vista, 7, or higher
- 512 MB RAM minimum for simple data sets, minimum 1 GB RAM recommended for large data sets or complicated networks
- At least 100 MB free hard disk space
- 1024 x 768 or higher monitor resolution with a minimum 16-bit color depth
- A video card with OpenGL hardware acceleration recommended

Installation Directions

Installing **Voxler 2** requires logging onto a computer with an account that has Administrator rights. Golden Software does not recommend installing **Voxler 2** over any previous versions of the program. **Voxler 2** can co-exist with older versions (i.e. **Voxler**) as long as both versions are installed in different directories, which is the default. For detailed installation instructions, refer to the README.RTF file.

To install **Voxler** from a CD:

1. Insert the **Voxler** CD into the CD-ROM drive. The install program automatically begins on most computers. If the installation does not begin automatically, double-click on the AUTORUN.EXE file located on the **Voxler** CD.
2. Choose *Install Voxler* from the **Voxler Auto Setup** dialog to begin the installation.

To install **Voxler** from a download:

1. Download **Voxler** according to the directions you received.
2. Double-click on the downloaded file to begin the installation process.

Updating Voxler

To update **Voxler**, open the program and choose the **Help | Check for Update** command. This will launch the Internet Update program which will check Golden Software's servers for any updates. If there is an update for your version of **Voxler** (i.e. **Voxler 2.0** to **Voxler 2.1**), you will be prompted to download the update.

When **Voxler** is initially launched, you will be prompted to allow the program to automatically check for updates. It is highly recommended that you select *Yes*. You can turn this update option on, off, or adjust the update interval by choosing the **Tools | Options** command. The update options are available on the **General** page of the **Options** dialog.

Uninstalling Voxler

Windows XP: To uninstall **Voxler**, go to the Windows Control Panel and double-click on Add/Remove Programs. Select "**Voxler 2**" from the list of installed applications. Click the *Remove* button to uninstall **Voxler 2**.

Windows Vista and 7: To uninstall **Voxler** when using the *Regular Control Panel Home*, click the *Uninstall a program* link. Select "**Voxler 2**" from the list of installed applications. Click the *Uninstall* button to uninstall **Voxler 2**.

Windows Vista: To uninstall **Voxler** when using the *Classic View*, go to the Windows Control Panel and double-click on Programs and Features. Select "**Voxler 2**" from the list of installed applications. Click the *Uninstall* button to uninstall **Voxler 2**.

A Note about the Documentation

The **Voxler** documentation includes this getting started guide and online help. Use the **Help | Contents** command in the program to access the detailed online help. Information about each command and feature of **Voxler** is included in the online help. In the event the information you need cannot be located in the online help, other sources of **Voxler** help include our support forum, FAQs, knowledge base, and contacting our technical support engineers.

If you prefer printed documentation, you may print the online help in part or in full. See the *Printing the Online Help* section on page 43 in this getting started guide for more information.

Various font styles are used throughout the **Voxler** documentation. **Bold** text indicates menu commands, dialog names, window names, and page names. *Italic* text indicates items within a dialog such as group box names, options, module names, and

field names. For example, the **Save As** dialog contains a *Save as type* drop-down list. Bold and italic text may occasionally be used for emphasis.

In addition, menu commands appear as **File | New**. This means, "click the **File** menu at the top of the **Voxler** window, then click on **New** within the **File** menu list." The first word is always the menu name, followed by the commands within the menu list.

Voxler Basics

In **Voxler**, data are imported and connected to various modules to accomplish a useful task such as displaying scatter plots, isosurfaces, heightfields, and image slices. The type of data determines what type of module can be attached to it. There are three main types of data: *point sets, lattices,* and *geometry*.

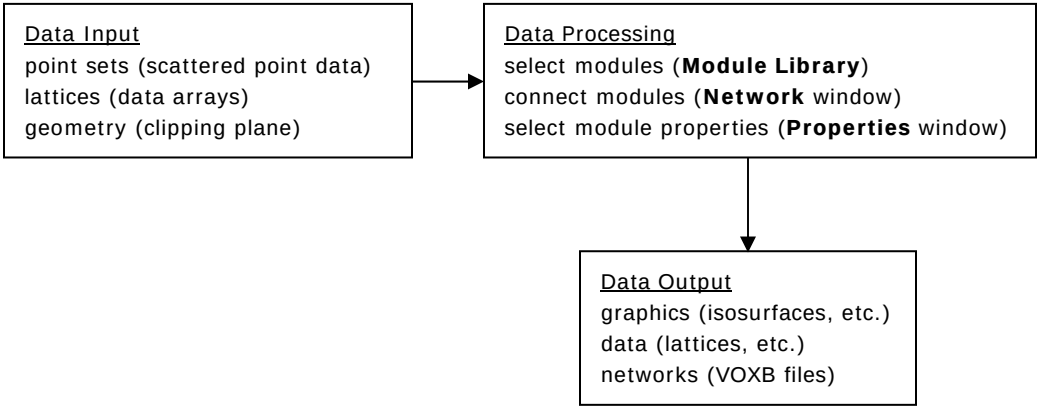
Point sets contain one or more three-dimensional point locations. Each point has an X, Y, and Z coordinate along with optional data components. A point can have any number of components. A component is a data variable associated with each point. Usually, this variable is the data you are trying to visualize, such as concentration.

A lattice consists of a one-, two-, or three-dimensional array of data. A one-dimensional lattice is a line of data. A two-dimensional lattice is similar to an image or a **Surfer** grid. A three-dimensional lattice defines a three-dimensional volume. Each node (or point) in the lattice can contain one or more components or data values. A single component lattice has a single data value (component) associated with each node. A multiple component lattice has two or more components per node.

Geometry consists of triangles, texture maps, line segments, etc. Geometry is collected at the end of the pipeline and displayed in the **Viewer** window. Geometry can be added from a data file, such as a DXF, SHP, BLN, or IV file, or can be created with a module, such as a contour or isosurface.

A module is a data set or process to be applied to a data set. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input ports, modify the data, and pass it along through the output ports. Modules represent data processing, such as filtering or resampling data. Modules also perform tasks such as creating isosurfaces and contour maps. All modules are located in the **Module Library**.

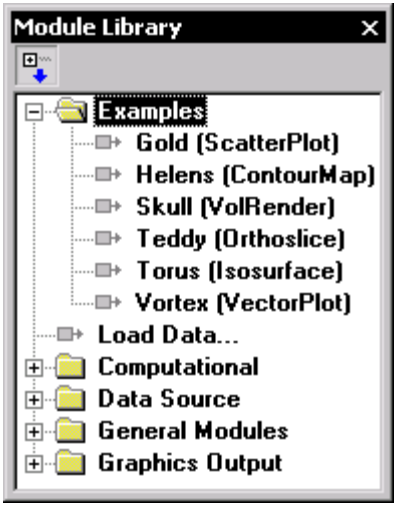
Voxler uses a visualization network to represent data, processing paths, and output. All data and modules for the current project are visible in the **Network** window. Select a module to modify the module properties in the **Properties** window. The graphical output from a module is displayed in the **Viewer** window. The **Network** window, **Module Library**, **Properties** window, and **Viewer** window are discussed in more detail in the following sections. **Voxler** files can be saved, exported to a variety of image formats, or recorded using the video capture command.



*Several data types can be loaded into **Voxler**. Once the data are loaded, you can select modules applicable to the data. The data can be visualized and exported.*

Three-Minute Tour

We have included several example files with **Voxler** so that you can quickly see some of **Voxler's** capabilities. The example files do not include all of **Voxler's** many data types, modules, and features. After opening an example file, the **Network** window is a good source of information as to what is included in each file. Example files are located at C:\Program Files\Golden Software\Voxler 2\Examples, by default.

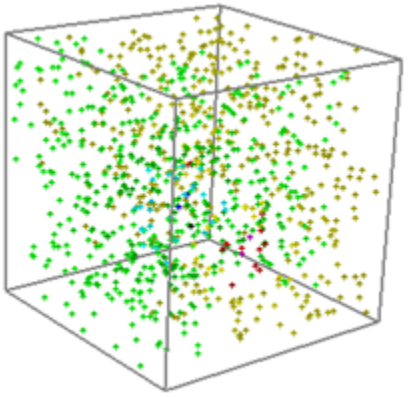


Double-click on an example name to open the file and view its contents.

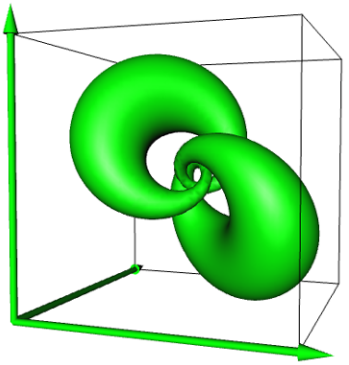
View Example Voxler Files

To view the example **Voxler** files:

1. Open **Voxler**.
2. Choose the **View | Module Library** command to display the **Module Library**. A check mark indicates the library is visible. The library is located on the left side of the window by default. The example files are located in the *Examples* folder at the top of the **Module Library**.
3. Double-click on an example file, such as *Torus (Isosurface)* to open the file in the **Viewer** window. Each time you double-click on a file, the new file opens in the **Viewer** window and the previous file closes.



The example file Gold (ScatterPlot) *displays a* ScatterPlot *module using* Pine Tree Line *symbols.*



The example file Torus (Isosurface) *displays an* Isosurface *module using a* TestLattice.

Using Voxler

To create a three-dimensional model in **Voxler**, you will need to start with data. **Voxler** supports several different data types. See the *Data Types* section on page 20 for detailed information on supported data types. Modules are attached to data in order to display the data or make adjustments to the data. See the *Modules* section on page 21 for detailed information on available modules.

To load a data file and create a scatter plot using **Voxler**:

1. Open **Voxler**.
2. Choose the **File | Load Data** command.
3. In the **Load Data** dialog, select the data file and click the *Open* button.
4. In the **Data Import Options** dialog, set the file format options. You can select *Delimiters* and how to treat text. Click the *OK* button.
5. In the **Select Data Columns** dialog, set the *X, Y, Z,* and *Component* information. Set any additional *Options* and click the *OK* button.
6. The data loads into **Voxler** and is displayed as a data module in the **Network** window. Adding modules to the data set will create a visualization pipeline in the **Network** window allowing you to create the output you desire.
7. Right-click on the data module and choose the **Graphics Output | ScatterPlot** command to add a scatter plot module. The output is displayed in the **Viewer** window.
8. Select the *ScatterPlot* module in the **Network** window and the properties are displayed in the **Properties** window. Adjust the properties as desired.
9. Choose the **File | Save Network As** command. Enter a *File name* in the **Save As** dialog and click the *Save* button to save your **Voxler** project.

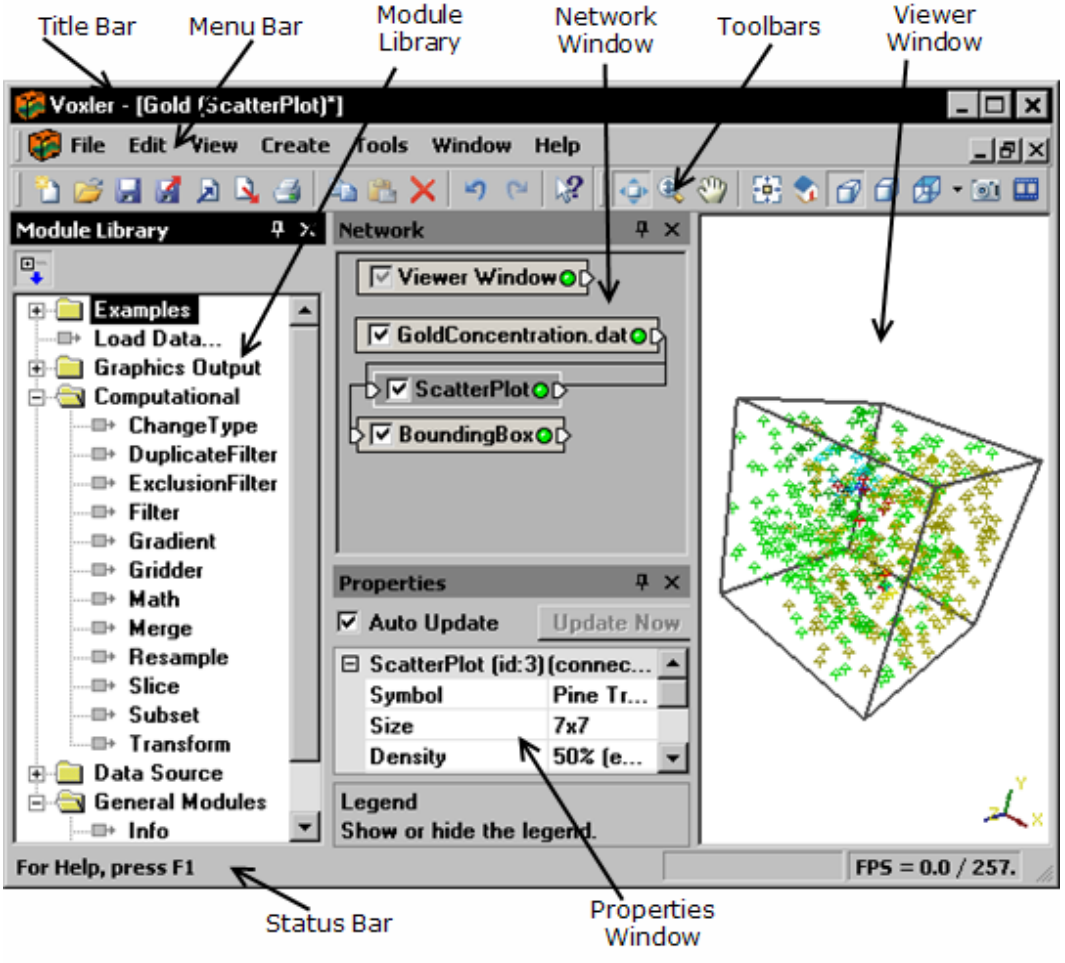
The tutorial lessons starting on page 29 contain detailed instructions on using **Voxler**. It is highly recommended that you complete the tutorial before beginning work in **Voxler**. Advanced tutorial lessons are available in the online help using the **Help | Tutorial** command.

Voxler User Interface Overview

Voxler uses multi-threading to keep the user interface responsive, even with computationally intensive background tasks. The user interface architecture is based on a single-document, multi-view model. This allows a document to exist with any number of view windows open on the document.

User Interface

The **Voxler** user interface consists of the title bar, menu bar, toolbars, **Module Library**, **Network** window, **Properties** window, **Viewer** window, and status bar.



*The **Voxler** user interface includes several toolbars and windows.*

User Interface Functions

The following table summarizes the function of each component of the user interface.

Component Name	Component Function
Title Bar	The title bar lists the program name plus the saved **Voxler** [.VOXB] file name (if any). An asterisk (*) after the file name indicates the file has been modified.
Menu Bar	The menu bar contains the commands used to run **Voxler**.
Toolbars	The toolbars contain **Voxler** tool buttons, which are usually shortcuts to menu commands. Move the cursor over each button to display a screen tip describing the command. Toolbars can be docked or floating.
Module Library	The **Module Library** lists example files and provides quick access to modules. The **Module Library** is initially docked on the far left side of the screen.
Network Window	The **Network** window displays the visualization network, consisting of loaded data files, modules, and connections. The **Network** window is initially docked in the middle top position of the screen.
Properties Window	The **Properties** window displays the properties of the module currently selected in the **Network** window. The **Properties** window is initially docked in the middle bottom position on the screen.
Viewer Window	The **Viewer** window contains the graphics output as directed by the modules in the **Network** window. The **Viewer** window is initially located to the far right side of the screen.
Status Bar	The status bar shows information about the activity in **Voxler**. The status bar is divided into three sections. The left section displays help messages and progress text. The middle section displays a progress gauge for various tasks, such as loading large data files. The right section displays the estimated time remaining for long tasks and the frames per second display.

Changing the Window Layout

The windows, toolbars, and menu bar display in a docked view by default; however, they can also be displayed as floating windows. The visibility, size, and position of each item may also be changed. Refer to the *Changing the Window Layout* topic in the online help for more information on layout options such as *position, visibility,* and *size.*

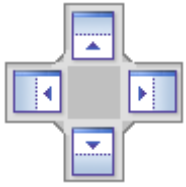
Visibility

Use the **View | Toolbars**, **View | Status Bar**, **View | Network Window**, **View | Properties Window**, and **View | Module Library** commands to turn these components on or off.

You can increase **Viewer** window space by minimizing the windows with *Auto Hide*. To hide the window, click on the button in the upper right corner of the **Module Library**, **Network**, or **Properties** windows. The window hides on the left or right side of **Voxler** with a small tab containing the window name. To view the window, place the cursor directly over the tab. Move the mouse away from the window to return the window to the hidden position. Click the button again to return the window to normal display mode.

Docking Mechanism

Use the **Tools | Options** command and set the *User interface style* to *Visual Studio 2005* or *Visual Studio 2008* to activate the docking mechanism feature that allows for easy docking of windows. With these visual looks, left-click the title bar of a window and drag it to a new location while holding down the left mouse button. The docking mechanism displays with arrow indicators as you move the window.



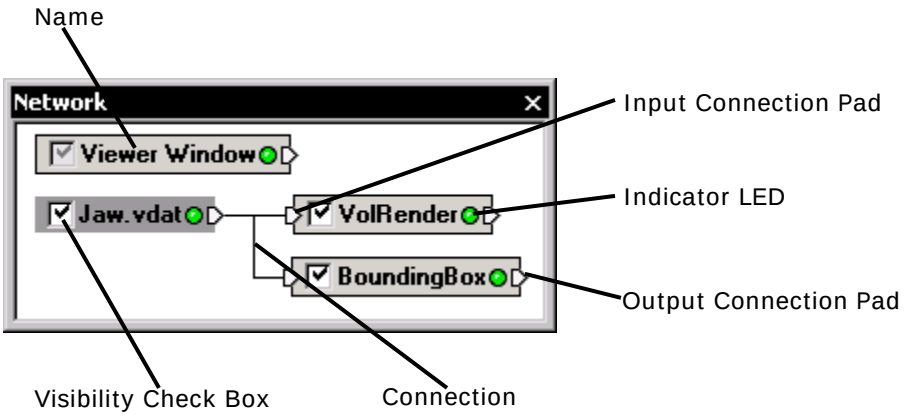
The docking mechanism has docking indicators.

When the cursor touches one of the docking indicators in the docking mechanism, a blue rectangle shows the window docking position. Release the left mouse button to allow the window to be docked in the specific location.

The Network Window

Voxler uses a **Network** window to show a graphical representation of data and processes performed in the project. All data, modules, and processing paths for the current project are visible in the **Network** window. Modules are connected to perform a desired task. A module is a data set or process to be applied to a data set. Modules accept data on their input connection pads, modify the data, and pass it along through the output connection pads. The final output from the pipeline is usually a graphical representation of data, such as a bounding box or 3D volrender.

A small rectangle, that can be selected and dragged with the mouse, represents each module in the **Network** window. The module rectangles have the following components:



Most modules contain a name, visibility check box, input connection pad, output connection pad, indicator LED light, and a connection line.

☑	The visibility check box indicates whether a module's output is visible in the **Viewer** window. Check the box to display a module and all "downstream" (connected) modules; uncheck the box to hide a module and all downstream modules. A gray check mark indicates that a module is disabled because of a hidden upstream module.
Name	Each module is named with the loaded data file name or by the function performed by the module. You can change the name with the **Edit | Rename Module** command. Alternatively, right-click the module and select **Rename Module**.
●	The indicator LED is a small round "light" showing module status. ▪ Green: the module is up to date ▪ Yellow: the module has been modified and needs to be updated ▪ Red: the module is in an error state
▷	An input connection pad is located on the left side of the module. An output connection pad is located on the right side of the module. The presence of connection pads indicates that a module may be connected to other modules. Only modules with the appropriate type of data may be connected.

Selecting and Deselecting Modules

To move a module, save a module, or change any of the module properties, select it by clicking the module rectangle in the **Network** window. The selected module is highlighted and its properties are displayed in the **Properties** window. Note that only a single module may be selected at a time. Click in the **Network** window outside all modules to deselect all modules.

Context Menus

Right-click a module to display a context menu containing various commands that can be applied to the module. The commands include:

- A list of modules that may be connected to the selected module. This is the fastest and easiest way to build a network. Simply right-click an existing module and select a module to connect.
- A **Connect** command which allows you to connect existing modules interactively. The command name changes depending on the selected module type.
- A **Save Data** command, which allows you to save the module's data output.
- A **Copy Module** command, which copies all properties of the selected module. This is useful for creating an exact duplicate of an existing module.
- A **Rename Module** command, which allows you to rename the selected module to any other name.
- A **Delete Module** command, which removes the selected module from the **Network** window. Deleting a module removes the links to other modules.

Right-click in the **Network** window without a module selected to display a different context menu that includes commands to load a new data set, create standalone modules, or paste a copied module. Standalone modules, such as *Annotation*, do not require an input connection.

Keyboard Commands

The following keyboard commands are available in the **Network** window:

- The DELETE key deletes the selected module if the **Network** window is active.
- The TAB key cycles through the modules in the order they were added.
- SHIFT+TAB cycles through the modules in the reverse order.
- The ARROW keys move the selected module.

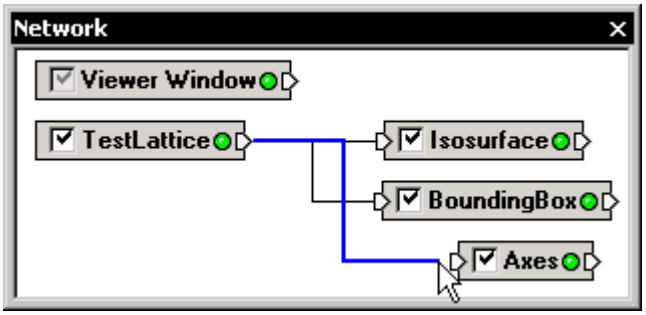
Viewer Window Module

The *Viewer Window* module is automatically created by **Voxler** and cannot be deleted or made invisible in the **Network** window. This module is used to control various settings that affect the entire scene, such as background color.

Connecting Modules

Module connections display the flow of data from one module to another. Connect a module to other compatible modules by dragging a connection line from a connection pad on one module to another. The line is initially blue, but turns yellow when the cursor touches a compatible module. Release the mouse button to complete the connection. The line turns black when the connection is complete.

Alternatively, you can click on a connection pad, release the mouse button, and move the mouse without dragging. A blue connection line is drawn. The connection line turns yellow when the cursor is over a compatible module. Click the mouse button a second time to connect the two modules.



Click on a connection pad and drag the mouse to another module's input connection pad to create a connection.

Press the ESC key or click in an empty portion of the **Network** window to cancel a connection in progress.

Some modules accept more than one input connection or provide more than one output connection. In these cases, a context menu is displayed when you click on the connection pad. Select the connection you want and proceed as previously described.

Disconnecting Modules

To disconnect a module from the network, follow the directions above to make the same connection a second time. The disconnected module remains in the **Network** window and retains all other connections.

Arranging Modules

Modules can be rearranged in the **Network** window. You may need to move modules to see some connections clearly. Click on a module and drag it to a new location to move it or use the ARROW keys for fine adjustment.

Tips on Working with Modules

Modules can be connected and edited to see the effect of different parameters on the data. For example, a point set can be loaded, connected to a *Gridder* module, and displayed as an *Isosurface* module. You can experiment with the gridding parameters and immediately see the new output in the isosurface. The *Isosurface* module automatically recognizes that the lattice has changed and updates itself.

Multiple modules can be connected to a single module output connection to stack effects in the **Viewer** window. For example, *Contours*, *Isosurface*, and *OrthoImage* modules could all be connected to a single lattice module. The output from all three modules is correctly combined in the scene.

Change to the module location, visibility, and connection status can be undone using the **Edit | Undo** command.

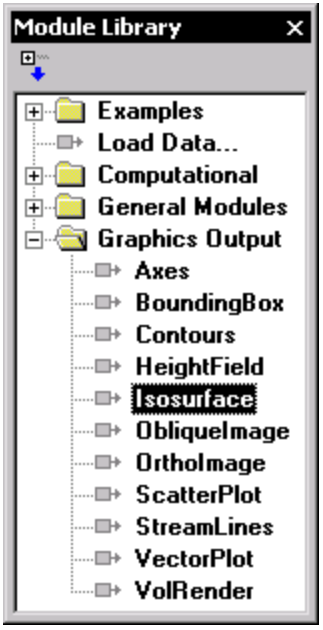
The style of the connection lines can be customized using the **Tools | Options** command.

The Module Library

The **Module Library** displays a list of available modules and other commands. You can add the modules to the **Network** window by double-clicking on the module in the **Module Library**. The item is added to the **Network** window. Depending on how the module is added, the module may appear in the **Network** window already connected to other modules.

Add Modules to the Network Window from the Module Library

Double-click a module in the *Graphics Output, Computational, Data Source,* or *General Modules* folder in the **Module Library** to add it to the **Network** window. If a module is selected in the **Network** window and the module in the **Module Library** is compatible, the two are connected. Otherwise, the module is loaded in the **Network** window without connections to any existing modules. You can also drag and drop a module from the **Module Library** to the **Network** window.



Double-click on a module name to add it to the Network.

Examples

Several simple example networks are displayed at the top of the window. Double-click an example to load it into **Voxler**.

Show All Modules

The toolbar at the top of the **Module Library** window contains a *Show All Modules* button  . If the button is depressed, all modules are listed. If *Show All Modules* is not depressed, only those modules that are compatible with the currently selected module are displayed.

Expand or Collapse

The + and - buttons indicate the folder can be expanded or collapsed to show or hide additional information. To expand a folder, click on the + control, select the item and press the plus key (+) on the numeric keypad, or press the right ARROW key on your keyboard. To collapse a folder, click on the - control, select the item and press the minus key (-) on the numeric keypad, or press the left ARROW key. You can also double-click on a folder name to expand or collapse the folder. This applies to any situation where you see a (+) or (-) in **Voxler**.

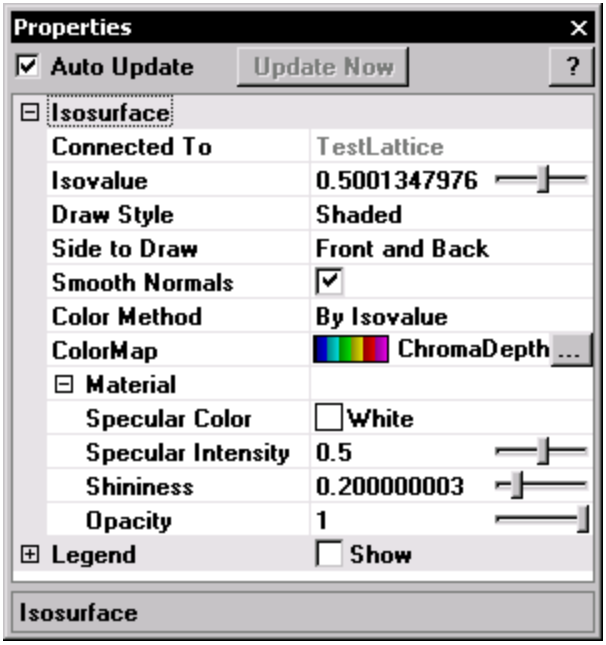
The Properties Window

The **Properties** window displays the properties of the module currently selected in the **Network** window. A property is a setting or parameter used by the module to control its behavior. Each property is displayed in a list within the window with one property per line. Click on a module in the **Network** window to display its properties in the **Properties** window.

Properties Window Sections

The **Properties** window consists of several sections described below.

Auto Update turns on or off automatic updating of the network when a change is made to the properties. If a process takes a long time or if you want to make several changes to different



*A module's properties are set in the **Properties** window.*

modules before updating the **Viewer** window, then uncheck *Auto Update* to turn this feature off. The *Update Now* button manually updates the network and any modified modules when *Auto Update* is unchecked. The button is enabled whenever there are pending changes to the network.

The main component of the **Properties** window is a list of properties used by the selected module. This list has two columns: the left column contains the property name. The right column contains the controls used to change the property. Click on the property control in the right column to change the property's value. Drag the vertical line between the left and right column to adjust the column width. If a module's properties contain subsections, a (+) or (-) is located to the left of the name. Click on the plus or minus icon to expand or collapse the list. For example, an *Axes* module contains *Axes*, *X Axis*, *Y Axis*, and *Z Axis* sections. Additional properties, such as the *Show Grid*, can be changed by opening these sections.

A simple help section is available at the bottom of the **Properties** window for help on the selected property. The help area is turned on or off with the **Tools | Options** command. The horizontal dividing line at the top of this section can be dragged up or down. Click the  button at the top of the **Properties** window to display more detailed information in the online help file about the module currently displayed in the **Properties** window.

Changing Properties

Voxler provides several different types of controls that allow you to customize the selected module behavior. For example, if an *Isosurface* module is selected in the **Network** window, properties specific to the isosurface are displayed in the **Properties** window. A list of the types of controls and how to use them appears in the online help topic, *The Properties Window*.

Some property list items only provide information. These items are disabled (grayed) to indicate they cannot be changed. Occasionally, some properties may not be valid due to other selections or data restrictions. These options are disabled as well.

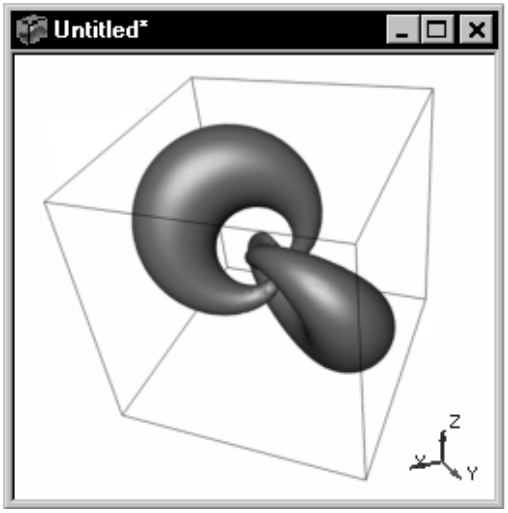
The Viewer Window

The **Viewer** window displays the three-dimensional graphical output produced by the visualization network. All visible items in the **Network** window are displayed in the **Viewer** window.

The **Viewer** window behavior is controlled by several settings from the toolbar or the **View** menu. A virtual camera is used to view, rotate, zoom, and pan the scene. An axis triad, located in the lower right corner, displays the global coordinate system orientation.

Camera

In **Voxler**, the camera is a metaphor for how we view the scene in the **Viewer** window. A scene is the collection of all graphical output. The camera is a virtual device used to view the scene. The camera has a point of view that is controlled by panning, zooming, rotating, and lighting. The camera position and the center target of the scene can be changed with the **View | Camera Properties** command.



*The **Viewer** window displays the graphics output modules and an axis triad.*

Projection

The camera can display graphics in perspective or orthographic projections. Specify the projection with the **View | Projection** command. The projection affects how the three-dimensional scene is drawn in the **Viewer** window.

The most distinguishing characteristic of perspective projection is foreshortening: the farther an object is from the camera, the smaller it appears in the final image. Perspective projection emulates the human eye so scenes appear more realistic or lifelike—larger when viewed closely, smaller when viewed from a distance.

Orthographic projection produces a parallel projection with no distortion for distance. As a result, it is sometimes difficult to determine how far an object is from you when viewing it in orthographic projection. This view is useful, however, when you need to measure distances or angles, or exactly align objects in three-dimensional space.

Headlight

The **Viewer** window includes a built-in headlight at the camera position pointing in the same direction as the camera. This allows all geometry to be seen even if no explicit lights have been added to the network. The headlight is turned on or off with **View | Headlight**. If the headlight is turned off, the scene is dark unless one or more explicit *Light* modules are added.

Home

The camera remembers a "home" position and orientation. Click the  button on the toolbar, click the HOME key on the keyboard, or choose the **View | Home** command to return the camera to the last stored home position.

Set a new home position by holding down the CTRL key and clicking the  button on the toolbar, by choosing the **View | Set Home** command, or by right-clicking in the **Viewer** window and selecting **Set Home** from the context menu. The home position is automatically set when a large change in the **Viewer** window occurs due to a significant change in network output when the **Tools | Options** *Enable AutoZoom* option is checked.

Draw Style

Choose the method that overlapping objects are drawn in the **Viewer** window with the **View | Still Draw Style** options. These commands are useful if a particular display mode takes too long to render and you want to temporarily display graphics in a faster drawing format such as **Wireframe**, **Low Resolution**, or **Bounding Box**. By choosing a faster drawing format, the scene will redraw quicker, but some elements may appear incorrect.

A similar command, **View | Animating Draw Style**, applies to the graphics when the camera position is being changed, such as when the graphic is spinning or while changing the zoom level. If a faster drawing format, such as **Wireframe** or **Bounding Box** is used, the rotating objects may appear incorrect.

Transparency

There are several transparency modes available through **View | Transparency Type**. Different modes work better for various types of geometry. Some modes provide faster rendering while others give better quality. Occasionally, you will need to experiment with transparency modes to find the best display.

If you find that your transparent graphics are partially opaque at certain orientations, try selecting **View | Transparency Type | Sorted object, sorted triangle blend** or **View | Transparency Type | Sorted Object Blend**. These methods provide a more accurate transparency mode, but are also slower and more memory-intensive than

others. If you find that you are running out of memory or rendering is taking a very long time, choose the **View | Transparency Type | Blend** command. This method usually provides a reasonable tradeoff between accuracy and speed.

Mouse

Choose the **Tools | Options** command to open the **Options** dialog. The **Mouse** page allows you to customize the mouse button assignments for the pan, zoom, rotation, and context menu control.

Trackball

The **Viewer** window is in trackball mode by default. If the **Viewer** window is not in trackball mode, choose the **View | Trackball** command.

Trackball mode allows you to click in the **Viewer** window and rotate or spin the graphics with a virtual trackball. The trackball can be thought of as a transparent sphere covering the entire scene. To rotate a graphic, click in the **Viewer** window and drag the mouse. To spin the graphic, click in the **Viewer** window, drag the mouse, and release the mouse button while dragging. To stop the spin, click anywhere in the **Viewer** window. Record the spinning graphic as an [.AVI] file with the **Edit | Capture Video** command.

For convenience, the following modifiers are allowed while in trackball mode:

- Hold down SHIFT while dragging to pan the camera
- Hold down CTRL while dragging up or down to zoom in or out

Fit to Window

Occasionally, the geometry is no longer visible in the **Viewer** window, perhaps because new geometry has been added or the existing coordinates were modified. The **View | Fit to Window** allows you to zoom in to fit all the geometry into the **Viewer** window. After the **Fit to Window** command is used, the camera is moved until the scene fills the window extents.

Zoom

The **Zoom In/Out** command zooms in and out as the mouse is dragged up and down or when the mouse wheel is moved. To use this feature, choose the **View | Zoom In/Out** command. Hold down the left mouse button and drag it up or down in the **Viewer** window to zoom in or out. As you drag the mouse up, the screen is zoomed in. As you drag the mouse down, the screen is zoomed out. You can also use the mouse wheel to zoom in or out. Move the wheel away from you to zoom in and toward you to zoom out.

Pan

The **Viewer** window can be panned. This allows you to move the graphics in the **Viewer** window without changing the level of magnification or rotation. This is useful when you would like to look at a different portion of the geometry at the current scale. To use this feature, choose the **View | Pan** command. Hold down the left mouse button and move the cursor around the window to pan the scene.

Data

Voxler supports several different file types. See the *File Format Chart* in the online help (**Help | Contents**) for a detailed list of supported file formats. Data are passed from one module to another through the visualization network to accomplish tasks such as displaying isosurfaces, heightfields, and image slices.

Data Types

The three main types of data include: point sets, lattices, and geometry.

Point Sets

Point sets contain one or more three-dimensional point locations. Each location has an X, Y, and Z coordinate along with optional data components. Point data are collections of XYZ points in space, optionally with associated data values. Occasionally, this is called "XYZC data" where XYZ represent the three-dimensional position and C represents one or more data values at that position.

Lattices

A lattice consists of a one-, two-, or three-dimensional data array. An array is a regular, structured matrix of points. A one-dimensional lattice is a line of data. Examples of two-dimensional lattices include bitmaps or **Surfer** grid files. A three-dimensional lattice defines a three-dimensional volume. Each node (or point) in the lattice can contain one or more components or data values. Lattices are further categorized by the node geometry: uniform, rectilinear, and curvilinear. Detailed information about the lattice geometry and components are found in the online help.

Geometry

Geometry consists of triangles, texture maps, line segments, and other objects. Geometry is collected at the end of the pipeline and displayed in the **Viewer** window. Geometry is usually represented internally using integer or single precision floating point.

Modules

A module is a data set or process to be applied to a data set. Modules are the building blocks from which the final output is constructed. Modules accept data on their input connection pads, modify the data, and pass it along through the output connection pads. There are four types of modules: computational, data source, general, and graphics output. Detailed information about each module is located in the online help. Basic information about each module is listed below.

Computational Modules

Computational modules alter the data by changing their type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates.

ChangeType

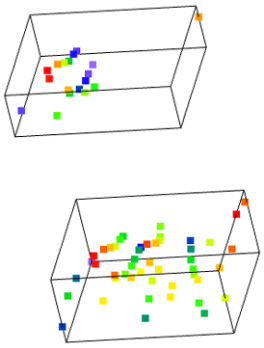
This module changes the lattice or point set data type from one primitive type (e.g. integer, float, etc.) to another. Smaller types save memory at the expense of reduced numeric precision. All components of the input data set are converted. The *ChangeType* module changes the data components type only, not the coordinates. Use the *Transform* module to change the coordinates.

DuplicateFilter

This module removes duplicate data points in a point set. Duplicate data are two or more data points having nearly identical X, Y, and Z coordinates. The *DuplicateFilter* properties control the definition of a duplicate point. Several options are available for determining which point, if any, to keep when points are considered duplicates.

ExclusionFilter

This module excludes data points in a point set according to a user-specified Boolean function. Some functions available are IF, AND, OR, NOT, and several comparison operations (=, <, >, etc). See the complete list of functions and operators in the online help on the *Mathematical Functions* page.



Custom exclusion filters can be applied to the data. The bottom scatter plot contains the data before the filter was applied and the top scatter plot shows the data after the filter was applied.

Filter

This module applies a digital filter to a uniform lattice. The lattice may be two-dimensional (images) or three-dimensional (volumes). Each filter reads the input lattice, performs a particular filtering operation on the data values in the lattice nodes, and sends the results to the output lattice. The input and output lattices are always the same size and type.

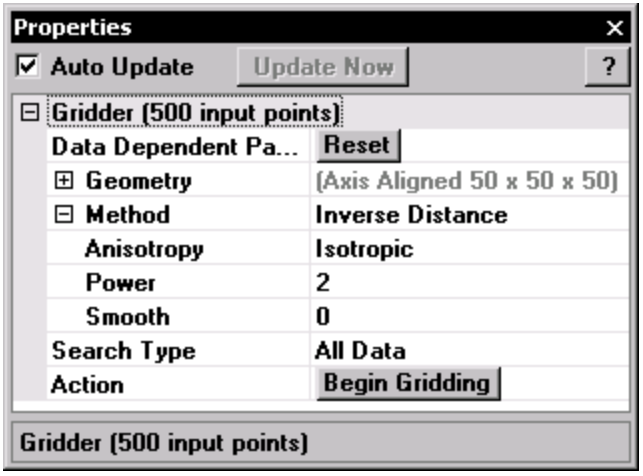
Filter module computations include data statistics such as local minimum, maximum, median, average, standard deviation; and image modification such as brightness and contrast.

Gradient

This module computes a gradient field from a single component of a two- or three-dimensional lattice. A gradient is a three-dimensional vector pointing in the direction of greatest slope. The output lattice contains three-component data at each lattice node. A centered difference algorithm is used to calculate the gradient. The output lattice geometry is identical to the input lattice geometry.

Gridder

This module interpolates scattered point data onto a uniform lattice. You can specify the output lattice range and resolution along with the interpolation method and associated parameters. Since gridding can take quite a while to execute, it is necessary to click the *Begin Gridding* button in the **Properties** window to start the process.



Click the Begin Gridding *button to start the gridding process.*

Math

This module creates a new output lattice by applying a numeric expression to one or more input lattices. The output lattice is calculated one node at a time by applying the numeric expression to the input lattice nodes.

Merge

This module combines two or more input lattices into a single uniform output lattice. You can specify the output lattice range and resolution.

Resample

This module allows the resolution of a lattice to be changed. This is performed by computing new data values at each output lattice node by interpolating the data values from the input lattice. The *Resample* module does not perform extrapolation.

Slice

This module creates a two-dimensional slice through a three-dimensional input lattice. The plane orientation may be preset to one of the local axis planes or in an arbitrary direction.

Subset

This module extracts a particular region of interest for further analysis. You can specify the geometric range, sampling frequency, and data components of the subset.

Transform

This module transforms the X, Y, and Z coordinates of an input point set or lattice using a standard 4x4 transformation matrix. The order of transformations is: scaling, rotation, and translation. Rotation and scaling are performed around the object's center.

Data Source Modules

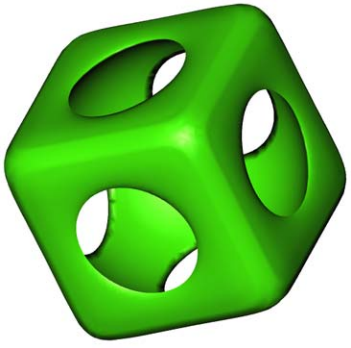
Data source modules serve as the source of raw data. The data may be loaded from a file or created from mathematical functions.

Load Data

The **File | Load Data** command creates a new source module by loading a file. The module is named with the loaded file name. See the *File Format Chart* in the online help for more details on supported file formats.

FunctionLattice

This module creates a new uniform lattice from a user-defined function. You can specify the output lattice range, resolution, number of components, and mathematical equations for defining each component.



One of the pre-defined test lattices, Wiffle Cube*, can be used for experimenting with modules and their properties.*

TestLattice

This module generates a variety of lattices for testing and experimenting with various modules. You can specify the output lattice range, resolution, and data type.

General Modules

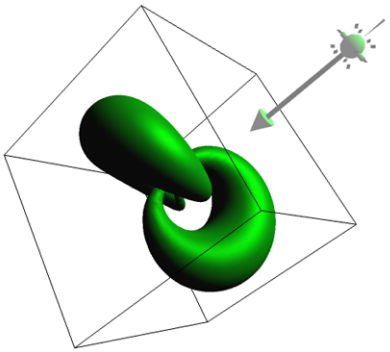
General modules display module information and provide custom lighting in the **Viewer** window.

Info

This module displays information about the connected module, such as data limits, number of components, and component type.

Light

This module creates a new directional, point, or spot light and adds it to the scene. Lights are cumulative. Every time a new light is added, it makes the scene a little brighter. You can add approximately eight lights to the scene. To view a scene with only light modules, uncheck the **View | Headlight** command to turn off the global light.



The Light *module adds directional, point, or spot lights to a scene.*

Viewer Window

This module contains various properties that affect the entire scene, such as background color. The *Viewer Window* module is automatically created. The *Viewer Window* module is displayed in the **Network** window; it is not listed in the **Module Library** since it always exists and cannot be deleted. The *Viewer Window* module only controls the options for the current **Viewer** window. To change the default **Viewer** window settings, choose **Tools | Options** and click on the **Colors** tab. The *New viewer window background* controls the color of future **Viewer** windows.

Graphics Output

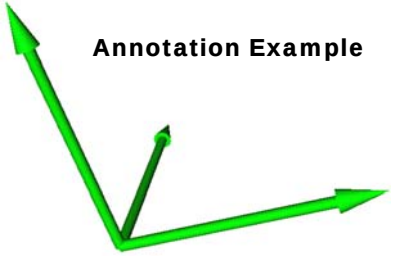
Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input.

Annotation

This module creates a text string that is always parallel to the screen. By default, the current date and time is used as the text string. You can enter your own text in the **Properties** window. Use the *Text* module to anchor the text to the scene.

Axes

This module creates a set of axes. The axes are

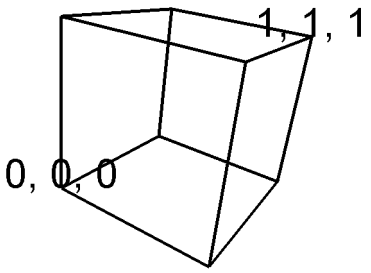


*Add axes to the **Viewer** window by double-clicking on the* Axes *module in the **Module Library**. Add* Annotation *or* Text *modules for titles, comments, or date/time.*

attached to an input point set or lattice. The axis labels are planar, although the plane orientation can be changed in the **Properties** window. A grid can also be displayed between any two axes. By default, the X axis is red, the Y axis is green, and the Z axis is blue. These colors can be changed in the **Properties** window for the existing axes or in the **Tools | Options** dialog for default conditions for future axes.

BoundingBox

This module draws a bounding box around the input module extents. Additionally, labels can be displayed for the minimum and maximum corners. The labels are displayed as screen-aligned text centered on the minimum and maximum corners.



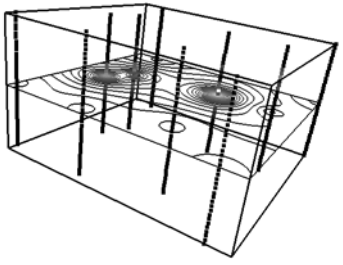
This graphic displays a BoundingBox *with labels.*

ClipPlane

This module clips input geometry according to a user-defined clipping plane. All geometry on one side of the plane is not drawn (clipped). The side that is clipped and the location of the clipping can be altered in the **Properties** window.

Contours

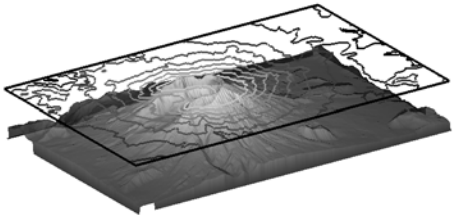
This module generates contour lines for a two-dimensional data set or for slices of a three-dimensional data set. Contour lines represent the boundary between data less than a given level (threshold) and data greater than the level. For three-dimensional data sets, the *Contours* module creates a planar slice through the lattice and contours the two-dimensional slice. Contour lines are colored by mapping data values to colors through a ColorMap.



This graphic displays a Contour *module on the XY Plane.*

HeightField

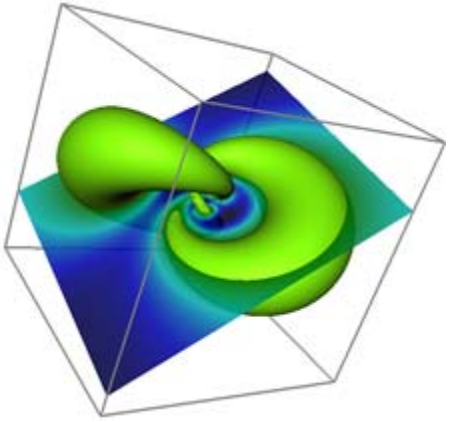
This module displays a lattice slice in three dimensions. The slice is scaled in the perpendicular direction by a user-specified data component and scale factor. The surface is colored by mapping the data values through a ColorMap.



The HeightField *module displays a lattice slice in three dimensions. A* Contours *module is displayed above the* HeightField.

Isosurface

This module creates an isosurface through an input lattice. An isosurface is a surface of constant value in a three-dimensional volume. The isosurface value is set in the *Isovalue* property in the **Properties** window. The isosurface separates regions less than the selected *Isovalue* from regions greater than the selected *Isovalue*. All points on the isosurface have the same value (the isovalue).



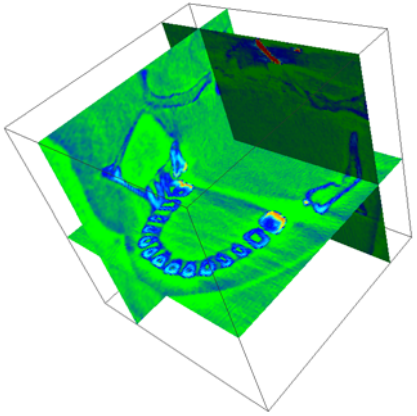
This is the Knotted Torus test lattice displayed as an Isosurface *with an* ObliqueImage *module.*

ObliqueImage

This module displays a color image on a two-dimensional cutting plane through a lattice. In medical terminology, this is known as a multi-planar reconstruction (MPR). The slice is represented using colors mapped through a ColorMap for scalar data, or as direct RGBA colors for lattices containing color data.

OrthoImage

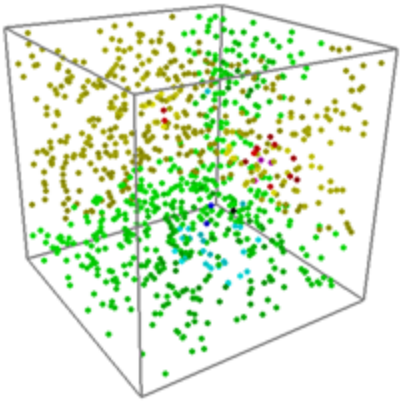
This module displays an orthogonal slice through a lattice parallel to one of the three axial planes (XY, XZ, or YZ). Orthogonal indicates elements are perpendicular or at right angles. The slice is represented by mapping data to a ColorMap for scalar data, or as direct RGBA colors if the lattice already contains color data. The scalar to color mapping may be specified with a linear gray mapping function with contrast enhancement or with a ColorMap.



This graphic displays three OrthoImage *modules: XY, XZ, and YZ.*

ScatterPlot

This module displays a set of symbols at each point of a point set or each node of a lattice. The symbols are screen-aligned and do not scale or "tilt" as the camera is changed. The symbol positions, however, are maintained in three dimensions.



The ScatterPlot *module displays symbols at points or nodes.*

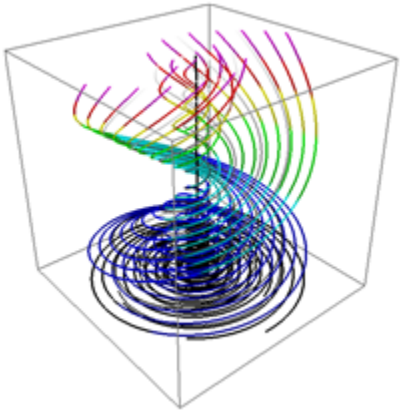
StreamLines

This module computes streamlines through a velocity field. Streamlines are lines within a volume of space that indicate flow direction and magnitude. The technique injects massless particles at specified seed points and traces their paths through the field. The particles stop when the new velocity is zero, the maximum stream length is exceeded, or when the stream intersects the bounds of the field.

Text

This module creates a two-dimensional text string aligned with the camera plane. The text has a three-dimensional anchor point that is transformed with the scene.

The text is not scaled according to the distance from the camera, nor is it influenced by rotation or scaling. It is, however, still obscured by graphics lying in front of it. The text is positioned according to the current transformation: the X origin is the first pixel of the leftmost character of text and the Y origin is the baseline of the first line of text with the baseline being the imaginary line on which all upper case characters are standing.

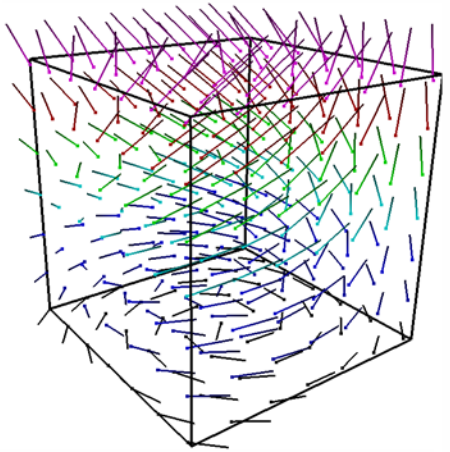


The StreamLines *module computes streamlines through a velocity field.*

Use the *Annotation* module to create text that is not anchored to the scene.

VectorPlot

This module displays vectors on a three-dimensional lattice or point set. It is often useful to combine the *VectorPlot* module with another module, such as *StreamLines*.

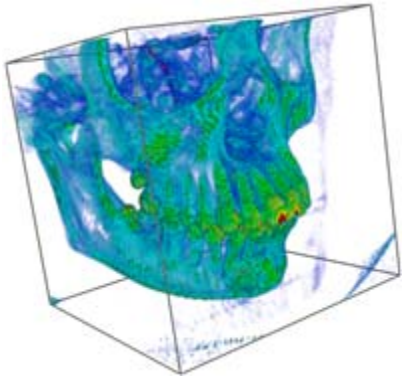


In this graphic, the TestLattice data are displayed as a VectorPlot *module.*

VolRender

Most visualization techniques convert volume data to surfaces. This module uses an alternative technique called *direct volume rendering* to render voxels directly. A voxel is short for **vo**lume pi**xel**, the smallest distinguishable box-shaped part of a three-dimensional image.

Volume rendering is a three-dimensional display of data that simulates the transmission and absorption of light through the points in the volume. Light rays are cast through the volume, where particles within the volume simultaneously emit and absorb light. The color of an individual pixel on the screen is computed by compositing the contributions from each particle that intersects the ray. This allows visualization of inhomogeneity inside objects with appropriate opacity adjustment.



In this graphic, a skull is visualized using the VolRender *module.*

Tutorial

The tutorial is designed to introduce you to some of **Voxler's** basic features. After you have completed the tutorial, you should be able to begin to use **Voxler** with your own data. We strongly encourage completion of the tutorial before proceeding with **Voxler**. The lessons should be completed in order; however, they do not need to be completed in one session. The tutorial should take approximately one hour to complete.

Now that you have an overview of **Voxler**, let's create some graphics. We will import data, link the data to modules, change properties, and save information.

Tutorial Lesson Overview

The following is an overview of lessons included in the tutorial.

- ***Lesson 1*** - ***Loading Data*** shows how to load data.
- ***Lesson 2 - Creating Graphics Output Modules*** shows how to create a scatter plot of the data and how to add a bounding box.
- ***Lesson 3 - Changing Properties*** shows how to change module properties and rotate the view.
- ***Lesson 4 - Using Computational Modules*** shows how to create a uniform lattice from the point set, display the lattice as an isosurface, and filter the data.
- ***Lesson 5 - Connecting Multiple Modules*** shows how to connect multiple output modules to a single input module.
- ***Lesson 6 - Saving Information*** shows how to save graphics, data, and a network.

Advanced tutorial lessons are available in the online help using the **Help | Tutorial** command.

Starting Voxler

To begin a **Voxler** session:

1. Click on the Windows Start button.
2. Navigate to **Programs | Golden Software Voxler 2** (in Windows XP) or **All Programs | Golden Software Voxler 2** (in Windows Vista and Windows 7) and click **Voxler 2**.
3. **Voxler 2** starts with a new blank **Viewer** window. The first time you open **Voxler** you are prompted for a serial number. Your serial number is located on the inside front cover of this getting started guide or in the email download instructions, depending on how you purchased **Voxler**.

If **Voxler** is already open, choose the **File | New** command or click the  button to open an empty visualization network before continuing with the tutorial.

Lesson 1 - Loading Data

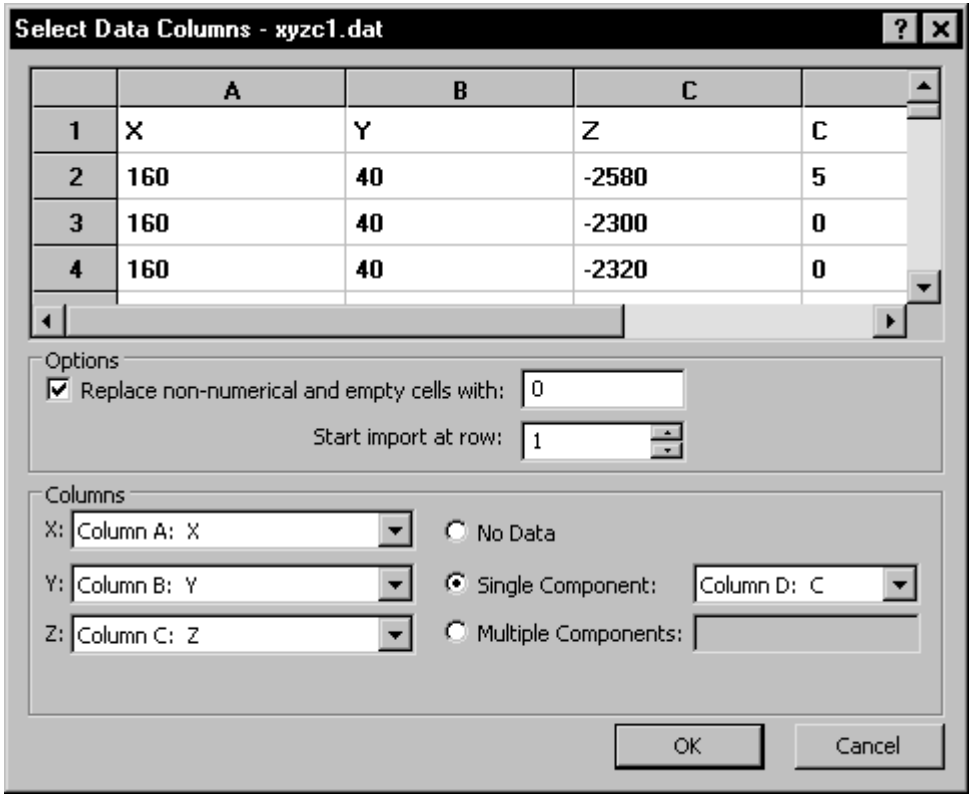
Many data types are used in **Voxler**. For a detailed list of supported formats, refer to the *File Format Chart* in the online help (**Help | Contents**). The three main categories of data are point sets, lattices, and geometry. Point sets contain X, Y, and Z data along with optional components. Lattices can be one-, two-, or three-dimensional data arrays. Geometry files include lines, triangles, and other shapes.

You can load data into **Voxler** by:

- choosing the **File | Load Data** command, or
- right-clicking in the **Network** window and selecting **Load Data**, or
- double-clicking the *Load Data* item in the **Module Library**.

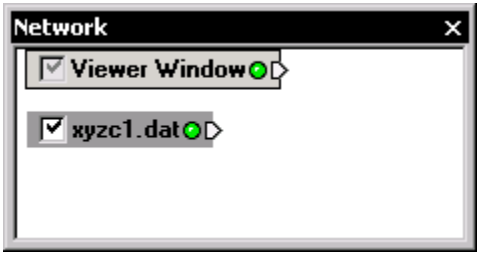
To load the tutorial data file:

1. Choose the **File | Load Data** command. The **Load Data** dialog opens.
2. In the **Load Data** dialog, change the *Look in* field to the Data folder inside the main **Voxler** directory. By default, this folder is located at C:\Program Files\Golden Software\Voxler 2\Data. Click on the XYZC1.DAT file and click the *Open* button.
3. In the **Data Import Options** dialog, check the *Comma* box to parse the data into separate columns as shown by the vertical lines in the *Preview* section. Click the *OK* button.
4. The **Select Data Columns** dialog offers options for changing the columns for the *X*, *Y*, *Z*, *Single Component*, and *Multiple Components*. The X, Y, and Z coordinates locate the points in space and the component data are the variables associated with each point. This dialog also allows you to specify the import starting row and a value to replace non-numerical and empty cells.
5. Uncheck the *Replace non-numerical and empty cells with* option so that empty rows are not imported.
6. Change the *Start import at row* to 2 so that the text in row 1 is not imported.
7. Click the *OK* button to finish importing the data.



*Select data for X, Y, Z, and component data in the **Select Data Columns** dialog.*

Once the **Select Data Columns** dialog closes, the **Network** window displays a module with the data file name. This module represents the raw data. To display graphics, you will add modules to the data set.



*After a data file is loaded, it appears as a module in the **Network** window.*

Lesson 2 - Creating Graphics Output Modules

Once the data are loaded, a data module appears in the **Network** window. The **Viewer** window is blank because no graphical modules have been added to the data module. In this lesson, we add a graphic.

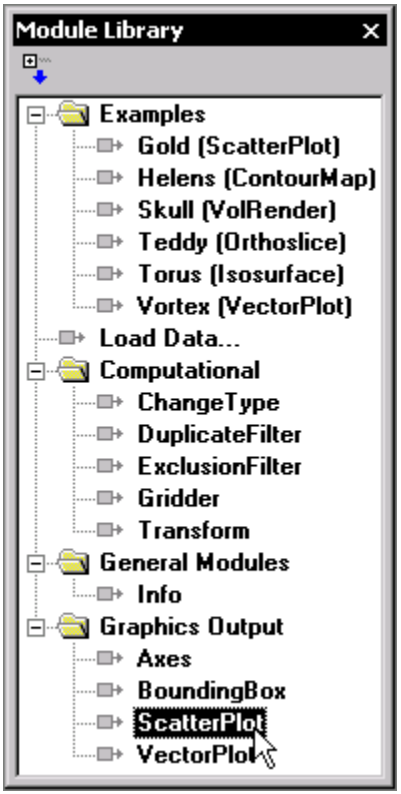
Creating a Scatter Plot

A scatter plot is a model of point data within a volume of space, optionally with colors representing data values.

To create a scatter plot:

1. Click on the *xyzc1.dat* module in the **Network** window to select it. The selected module is highlighted.
2. In the **Module Library**, double-click on the *ScatterPlot* module name in the *Graphics Output* section of the list. Alternatively, right-click on *xyzc1.dat* in the **Network** window and select **Graphics Output | ScatterPlot** from the context menu.

A scatter plot is displayed in the **Viewer** window and the *ScatterPlot* module appears connected to the *xyzc1.dat* module in the **Network** window.



Select ScatterPlot *in the* Graphics Output *folder to create a scatter plot of the data.*

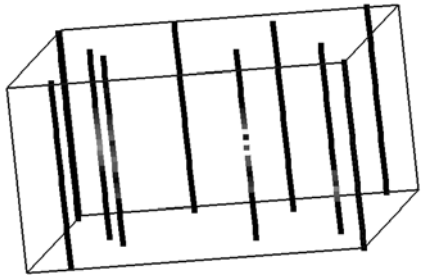
In the **Module Library**, only the modules that can be used with the data type are listed if the *Show All Modules* button is not selected. The button is selected if it looks like this  and not selected when it looks like this  . In this example, we used point data, so the main graphics outputs are axes, bounding box, scatter plots and vector plots. Other graphics, such as isosurfaces, need lattices as inputs so they are not listed when the button is not selected.

Adding a Bounding Box

Next, we will add a bounding box around the input data. A bounding box draws a

To draw a bounding box:

1. In the **Network** window, click the *xyzc1.dat* module.
2. In the **Module Library**, double-click *BoundingBox* under *Graphics Output*.



Your scatter plot should look similar to this after you add the bounding box.

Alternatively, you could have right-clicked the *xyzc1.dat* module in the **Network** window and selected **Graphics Output | BoundingBox** from the context menu.

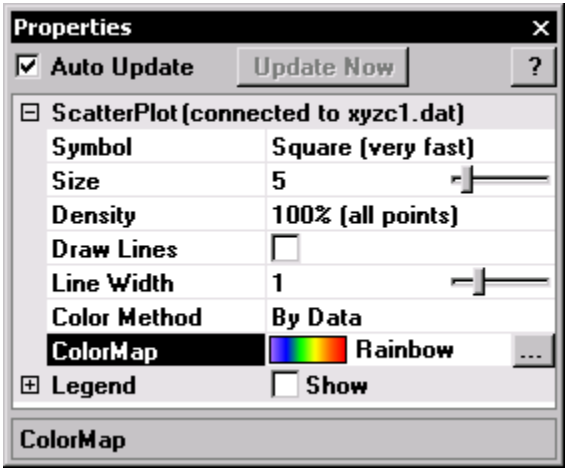
Lesson 3 - Changing Properties

Once modules have been created, their properties can be changed in the **Properties** window. Click on a module in the **Network** window to select it and display the module's properties in the **Properties** window. Note that some items in a module's property list are informational only and cannot be changed. These items appear gray in the list.

Changing Symbol Colors

One property that can be changed for a scatter plot is the symbol color. To change the symbol colors:

1. Click on *ScatterPlot* in the **Network** window. The *ScatterPlot* module properties open in the **Properties** window.
2. In the **Properties** window, locate the property named *ColorMap*. Click the sample color spectrum (*GrayScale*) to the left of the ... button to open the drop down list. Click *Rainbow* to change the colors. The colors are mapped to the data variable C, as selected when the data were loaded in *Lesson 1 - Loading Data*.



Click the color spectrum next to ColorMap *and select* Rainbow *from the list to change the symbol colors.*

Changing the Bounding Box Properties

Changing the bounding box properties is similar to changing the symbol colors.

To change the bounding box color:

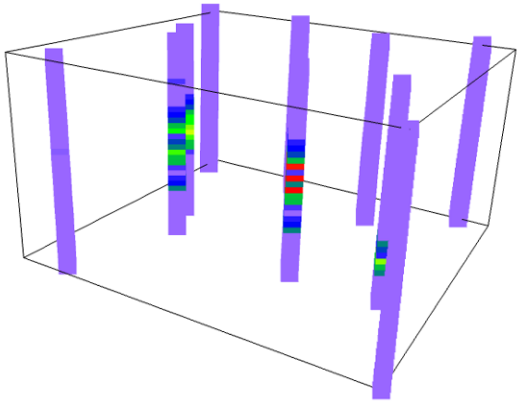
1. Click on *BoundingBox* in the **Network** window.
2. In the **Properties** window, click *Yellow* next to the *Color* property. The color palette opens.
3. Click on the color black to change the bounding box color to black.

Rotating Graphics

The **Viewer** window contents can be rotated and animated (spinning). Currently, we are viewing the scatter plot from the top. We can rotate the scatter plot to see the symbols more clearly.

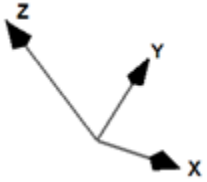
The view is rotated by clicking on the **Viewer** window and dragging the mouse. If you release the mouse button while the mouse is still moving, the **Viewer** window will enter spin mode.

Experiment with different rotations. If you spin the graphic, you can stop the spin by clicking anywhere in the **Viewer** window.



*The contents of the **Viewer** window can be rotated.*

A world axis triad is displayed in the lower right corner of the **Viewer** window. The axis triad is informational only. The triad is a depiction of the X, Y, and Z directions that shows the **Viewer** window camera orientation. This is useful when rotating graphics to see how the graphics have been rotated in space.



*The world axis triad shows the **Viewer** window camera orientation.*

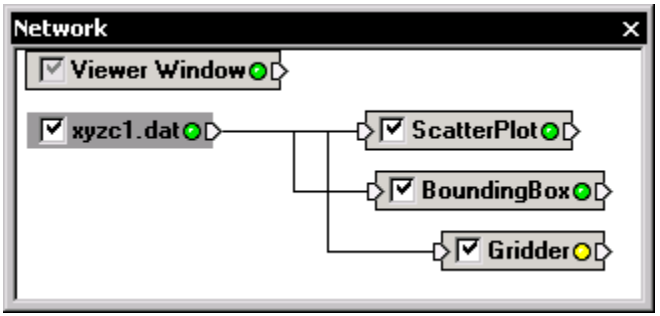
Lesson 4 - Using Computational Modules

Computational modules use data inputs to grid point sets, filter data, merge data, and perform other tasks. Several visually interesting graphics output modules require lattices, including contours, heightfields, isosurfaces, oblique images, orthogonal images, streamlines, and volrenders. The XYZC1.DAT file contains scattered point data. You can use the *Gridder* module to create a lattice from the scattered point data. After the point set is converted to a lattice, we can display it as an isosurface and experiment with filtering to see the effect on the network and isosurface.

Gridding Data

To create a lattice from XYZC1.DAT:

1. In the **Network** window, click the *xyzc1.dat* module.
2. In the **Module Library**, double-click the *Gridder* module in the *Computational* section. The *Gridder* module is loaded into the network. Alternatively, right-click on the *xyzc1.dat* module in the **Network** window and select **Computational | Gridder**.
3. In the **Network** window, the *Gridder* module displays with a yellow indicator LED, indicating that action is required. In this case, we need to initiate gridding in the **Properties** window. With the *Gridder* module selected in the **Network** window, click the *Begin Gridding* button in the **Properties** window to begin the gridding process. The *Gridder* module indicator LED changes to green when the gridding is complete.



The indicator LED is a small circle to the right of the module name. The Gridder *module indicator LED initially appears yellow, indicating you must make a property change.*

Creating an Isosurface

The *Gridder* module interpolated the scattered point data onto a uniform lattice. To display the lattice in the **Viewer** window, the *Gridder* module needs to be connected to a graphics output module. An isosurface, a surface of constant value in three dimensions, is one way a lattice can be displayed.

To create an isosurface:

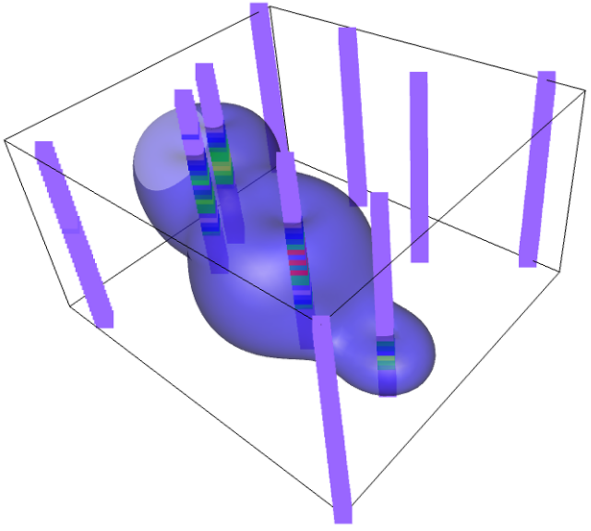
1. Click the *Gridder* module in the **Network** window to select it.
2. Double-click the *Isosurface* module in the **Module Library**. The *Isosurface* module appears connected to the *Gridder* module in the **Network** window and an isosurface displays in the **Viewer** window. Alternatively, right-click on the *Gridder* module and select **Graphics Output | Isosurface**.

Changing the Isosurface Properties

We can experiment with the isovalue (constant value) to change the isosurface appearance.

To change the isosurface properties:

1. Click on the *Isosurface* module in the **Network** window to select it.
2. In the **Properties** window, change the value next to *Isovalue* by double-clicking on the default value, typing 20, and pressing ENTER on your keyboard. Alternatively, the dragger next to the number can be moved to change the *Isovalue*. A new isosurface is calculated and displayed in the **Viewer** window.



An isosurface displays a constant value in three dimensions.

3. In the **Properties** window, click the *GrayScale* color spectrum next to *ColorMap* and select *Rainbow*.
4. In the **Properties** window, change the *Opacity* value by typing 0.5 and pressing ENTER or sliding the slider bar until the value is 0.5. Changing the *Opacity* to a lower value allows the isosurface to be partially transparent. The lower the value, the more transparent the isosurface.

Drag the slider bar to change a number, or type the number and press ENTER on the keyboard.

A Note About Transparency

An *Opacity* value of 0.0 is fully transparent. An *Opacity* value of 1.0 is fully opaque. Transparency can be very time consuming to get absolutely correct. As such, **Voxler** contains several different algorithms that trade off speed against correctness. See the **View | Transparency Type** options if the transparency does not look correct for your particular data. Usually the **Sorted object, sorted triangle add** and **Sorted object, sorted triangle blend** methods result in good output but these methods are significantly slower than the other methods. A quick method that produces good results in many cases is **Blend** or **Delayed Blend**.

Filtering Data

You can add additional computational modules between the data file module and the *Gridder* module to change the isosurface. There are many data filtering options in **Voxler**. Filtering modifies the data stream, which affects all downstream modules. Typically, the "downstream" modules are automatically changed when "upstream" modules are altered. The *Gridder* module is one exception due to the time required to grid the data.

As an example of filtering our data, assume the data contains points that are very close together and we would like to combine these duplicate points into a single representative value.

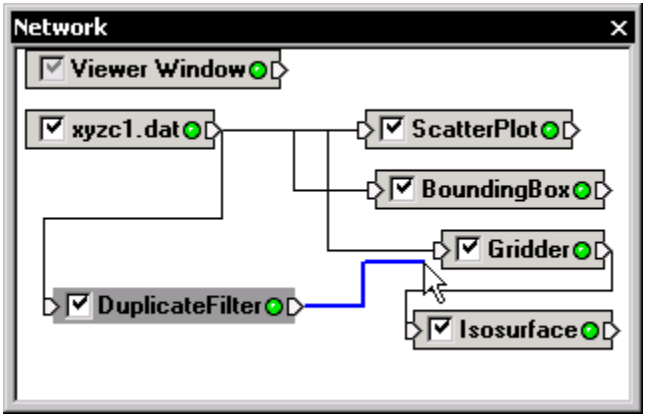
To average these duplicate points:

1. Click the *xyzc1.dat* module in the **Network** window.
2. In the **Module Library** *Computational* section, double-click the *DuplicateFilter* module to add it to the **Network** window. Alternatively, right-click on the *xyzc1.dat* module and select **Computational | DuplicateFilter**.
3. Click on the *DuplicateFilter* module in the **Network** window to select it.
4. In the **Properties** window, change the *Keep* option to *Median Z*.
5. In the **Properties** window, enter 20 for the *Z Tolerance* and press ENTER.

Since there are no output modules currently connected to the *DuplicateFilter* module, there are no visible changes in the **Viewer** window. We can make changes by connecting the *DuplicateFilter* module to the *Gridder* module.

To connect the *DuplicateFilter* module:

1. First, move the *DuplicateFilter* module icon to the left side of the **Network** window so the connections are easily seen. Click on the *DuplicateFilter* module rectangle and drag it to the left side of the **Network** window.
2. Click the output connection pad ▷ on the right side of the *DuplicateFilter* module in the **Network** window.
3. In the **Network** window, hold the mouse button down and drag the cursor to the left side of the *Gridder* module to connect the two modules. The connection line changes from blue to yellow when the cursor is over a module to which it can be connected. Release the mouse button when the line turns yellow. The connection line color changes to black when the connection is completed.

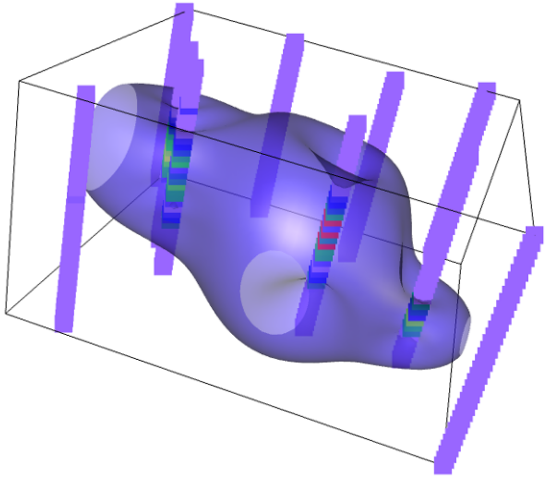


Click on the DuplicateFilter *output connection pad and drag a line to the* Gridder *module to connect the two modules.*

Since the *Gridder* module accepts only one input, connecting the *DuplicateFilter* module causes the *Gridder* module to automatically disconnect from the *xyzc1.dat* module. In addition, the *Gridder* module indicator LED turns yellow indicating that additional input is required. Once the gridding is complete, the *Isosurface* module automatically updates and the new graphical output is sent to the **Viewer** window.

To update the *Gridder* and *Isosurface* modules:

1. In the **Network** window, click the *Gridder* module to view its properties in the **Properties** window.
2. In the **Properties** window, click the *Reset* button next to *Data Dependent Parameters*. This recalculates the lattice limits and other parameters to use the new input coming from the *DuplicateFilter* module.
3. Click the *Begin Gridding* button at the bottom of the **Properties** window. The progress gauge displays the gridding progress and the *Gridder* module indicator LED turns green when the gridding is complete. The *Isosurface* module automatically updates with the new information and the results display in the **Viewer** window.



After the data are regridded, the isosurface automatically updates to reflect the changes since it is "downstream" from the Gridder *module.*

Lesson 5 - Connecting Multiple Modules

Modules can have multiple connections. For example, the output from a *Gridder* module can be connected to several graphics output modules to show multiple aspects of the data in one graphic.

Adding Contours

To add another graphics output module to the *Gridder* module:

1. Click the *Gridder* module in the **Network** window.
2. In the **Module Library**, double-click the *Contours* module in the *Graphics Output* section to connect it to the *Gridder* module. Alternatively, right-click on the *Gridder* module and select **Graphics Output | Contours**. The **Network** window now contains a connected *Contours* module and the **Viewer** window displays contours with the default settings.

Sometimes, the default settings are not exactly what we want to show in our **Viewer** window.

To change the contour properties:

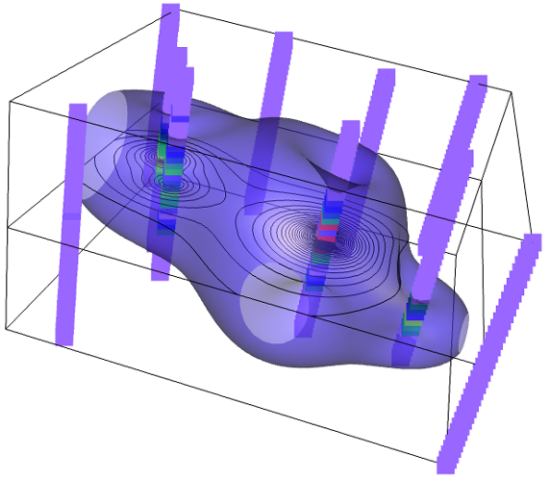
1. Click the *Contours* module in the **Network** window.
2. In the **Properties** window, click *Automatic* next to *Level Method* and choose *Min, Max, Interval*. This property selection allows you to set the minimum and maximum contour values, and the contour interval (number of units between contour lines).
3. In the **Properties** window, double-click on the default *Level Interval* value, type 5, and press ENTER.

With each contour property change, the contours update automatically in the **Viewer** window.

Changing the Transparency

All of the contours may not be visible in the **Viewer** window depending on the transparency settings.

To change the transparency mode, right-click in the **Viewer** window and select **Transparency Type | Delayed blend** from the context menu.



Experiment with the transparency options to see how it affects the scene. This example is using Delayed Blend *as the* Transparency Type.

Lesson 6 - Saving Information

There are several ways to save **Voxler** information:

- Select **File | Save Network** to save the data set and all of its associated modules as a **Voxler** network file.
- Select **File | Save Data** to save a selected module's data.
- Select **File | Export** to export graphic files such as bitmaps.
- Select **Edit | Copy Snapshot** to copy the **Viewer** window view to the clipboard.
- Select **Edit | Capture Video** to capture the screen rotation and save to an AVI.

If you are using the demo version of **Voxler**, you will not be able to save or export.

Saving a Network

To save a network:

1. Choose the **File | Save Network As** command. The **Save As** dialog opens.
2. Type TUTORIAL NETWORK into the *File name* field. Note there is only one option in the *Save as type* list, *Voxler Network Files (*.voxb)*.
3. Click the *Save* button and the dialog closes. The network is saved so that it can be reused in **Voxler**. The network file format includes all data, including the raw source modules, and everything else needed to reload the network in the future.

Saving Data

To save the selected module's output data:

1. Click the *DuplicateFilter* module in the **Network** window.
2. Choose the **File | Save Data** command. Alternatively, right-click on the module and select **Save Data**. The **Export** dialog appears.
3. Type TUTORIAL DATA into the *File name* field and select *DAT Golden Software Data (*.dat)* in the *Save as type* box. Click the *Save* button.
4. In the **Data Export Options** dialog, accept the defaults and click the *OK* button. The data are saved as a point set in the specified location.

Saving a Graphic

To save a graphic:

1. Choose the **File | Export** command. The **Export** dialog opens.
2. Type TUTORIAL GRAPHIC into the *File name* box. Keep *BMP Windows Bitmap (*.bmp)* in the *Save as type* field and click the *Save* button.
3. In the **Export Options** dialog, leave the default selections and click the *OK* button. The image is saved as a bitmap [.BMP] file in the specified location.

Congratulations, you have completed the **Voxler** tutorial. Advanced tutorial lessons are available in the program using the **Help | Tutorial** command. Training videos are available on the Golden Software website.

Getting Help

The getting started guide is a quick way to learn about the basics in **Voxler**. There are also other sources of help with **Voxler**.

Online Help

Extensive information about **Voxler** is located in the online help. To access the online help, choose **Help | Contents**. You can navigate help using the **Contents**, **Index**, **Search**, and **Favorites** pages in the navigation pane to the left of the topic page.

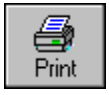
In the **Voxler2 Help** dialog, click the **Contents** tab to display and browse through the main help titles. Click the + button next to a book to display the topics contained in the book. Click a page to display a particular help topic. Click the **Index** tab to search for topics by keywords. Type a word in the box under *Type in the keyword to find* and a list of topics containing that keyword displays. Double-click the topic of interest to display the topic. Click the **Search** tab to search for a word or phrase in the help file. Type a keyword or phrase under *Type in the word(s) to search for*. Click the **Favorites** tab to bookmark help topics of interest for later reference.

Printing the Online Help

The online help topics may be printed. You can print a single topic, a section of the table of contents, or all topics in the table of contents. Open the online help by selecting the **Help | Contents** command in **Voxler**.

Printing One Topic

To print one topic:

1. Open the online help by selecting **Help | Contents** in the **Voxler** window.
2. Click the topic you want to print.
3. Click the 

button.
4. If the **Contents** page is open in the help navigation pane, the **Print Topics** dialog appears. Select *Print the selected topic* and click the *OK* button.

Printing One Book

To print one book, the tutorial for example:

1. Open the online help by selecting **Help | Contents** in the **Voxler** window.
2. Click the **Contents** page on the left side navigation pane.
3. Expand the *Voxler 2* book and click on the *Tutorial* book.
4. Click the 

button.

5. If the **Contents** page is open in the help navigation pane, the **Print Topics** dialog appears. Select *Print the selected heading and all subtopics* and click the *OK* button. All the topics included in the *Tutorial* book are printed.

Printing the Entire Help File

To print all of the topics in the help file table of contents:

1. Select the top-level book in the help file, *Voxler 2*.
2. Click the 

button.
3. The **Print Topics** dialog appears. Select *Print the selected heading and all subtopics* and click the *OK* button. All the topics included in the online help table of contents are printed. WARNING: Printing the entire help file takes hundreds of letter-sized sheets of paper and is very time consuming to print. There is no table of contents or index printed with the file.

Context-Sensitive Help

Voxler also contains context-sensitive help. Highlight a menu command, window region, or dialog box, and press the F1 key to display help for the highlighted item. You may also access context-sensitive help by pressing SHIFT+F1 or clicking on the  button. After clicking the  button, the cursor appears as  . Simply click the item for which help is required and a help window appears.

In addition, the dialogs and the **Properties** window contain a help button. Click the  button in the dialog title bar to obtain help for that dialog. Clicking the  button in the **Properties** window opens the help topic for the displayed properties.

Internet Resources

There are several Internet help resources.

- Click the *Forums* button in online help (**Help | Contents** command) to research a **Voxler** question or to post a question to the group.
- Use the **Help | Feedback** commands to send a problem report, suggestion, or information request by email directly to a technical support engineer.
- Search the FAQs on our web page at www.goldensoftware.com.
- Search the knowledge base on our website at www.goldensoftware.com.
- Direct links to the Golden Software home page, the **Voxler** product page, frequently asked questions, and the knowledge base are available by choosing the **Help | Golden Software on the Web** command in **Voxler**.

Technical Support

Golden Software's technical support is free to registered users of Golden Software products. Our technical support staff is trained to help you find answers to your questions quickly and accurately. We are happy to answer all of your questions about any of our products, both before and after your purchase. We also welcome suggestions for improvements to our software and encourage you to contact us with any ideas you may have for adding new features and capabilities to our programs.

Technical support is available Monday through Friday 8:00 AM to 5:00 PM Mountain Time, excluding major United States holidays. We respond to technical questions received by phone, email, and fax within one business day. When contacting us with your question, have the following information available:

- Your **Voxler** serial number, found in the **Help | About Voxler** dialog, in the original email you received with download instructions, or on the inside front cover of your printed getting started guide
- Your **Voxler** version number, found in the **Help | About Voxler** dialog
- The operating system you are using (Windows XP, Vista, 7, or higher)
- The steps you took prior to experiencing your problem
- The exact wording of the first error message (if any) that appeared

If you encounter problems with **Voxler**, you are welcome to send an email message to Golden Software using the **Help | Feedback | Problem Report** command or by sending an email to voxlersupport@goldensoftware.com. Report the steps you perform when the problem occurs and include the full text of any error messages that are displayed. You are welcome to attach a [.ZIP] file (10 MB maximum) containing the [.VOXB] file and other files that illustrate the problem. Contact technical support for other arrangements if you have very large zipped attachments to send.

Contact Information

Telephone: 303-279-1021

Fax: 303-279-0909

Email: voxlersupport@goldensoftware.com

Web: www.goldensoftware.com (includes FAQs, knowledge base, support forum, training videos, newsletters, downloads, and more!)

Mail: Golden Software, Inc., 809 14th Street, Golden, Colorado 80401-1866, USA

Index

Before calling, please check the following available resources as your question may already be answered.

Registration: Register online at www.goldensoftware.com or fax or mail in the card on the inside front cover of this getting started guide
Knowledge Base: www.goldensoftware.com/activekb or in the **Voxler** program using the **Help | Golden Software on the Web | Knowledge Base** command
Forums: www.goldensoftware.com/forum or in the **Voxler** program using the **Help | Contents** command and click on Forums
Frequently Asked Questions: In the **Voxler** program using the **Help | Golden Software on the Web | Frequently Asked Questions** command
Tutorial: Complete the tutorial section in this getting started guide or in the **Voxler** program using the **Help | Tutorial** command
Online Help: In the **Voxler** program using the **Help | Contents** command
Support Videos: www.goldensoftware.com

Business Hours

Technical Support:

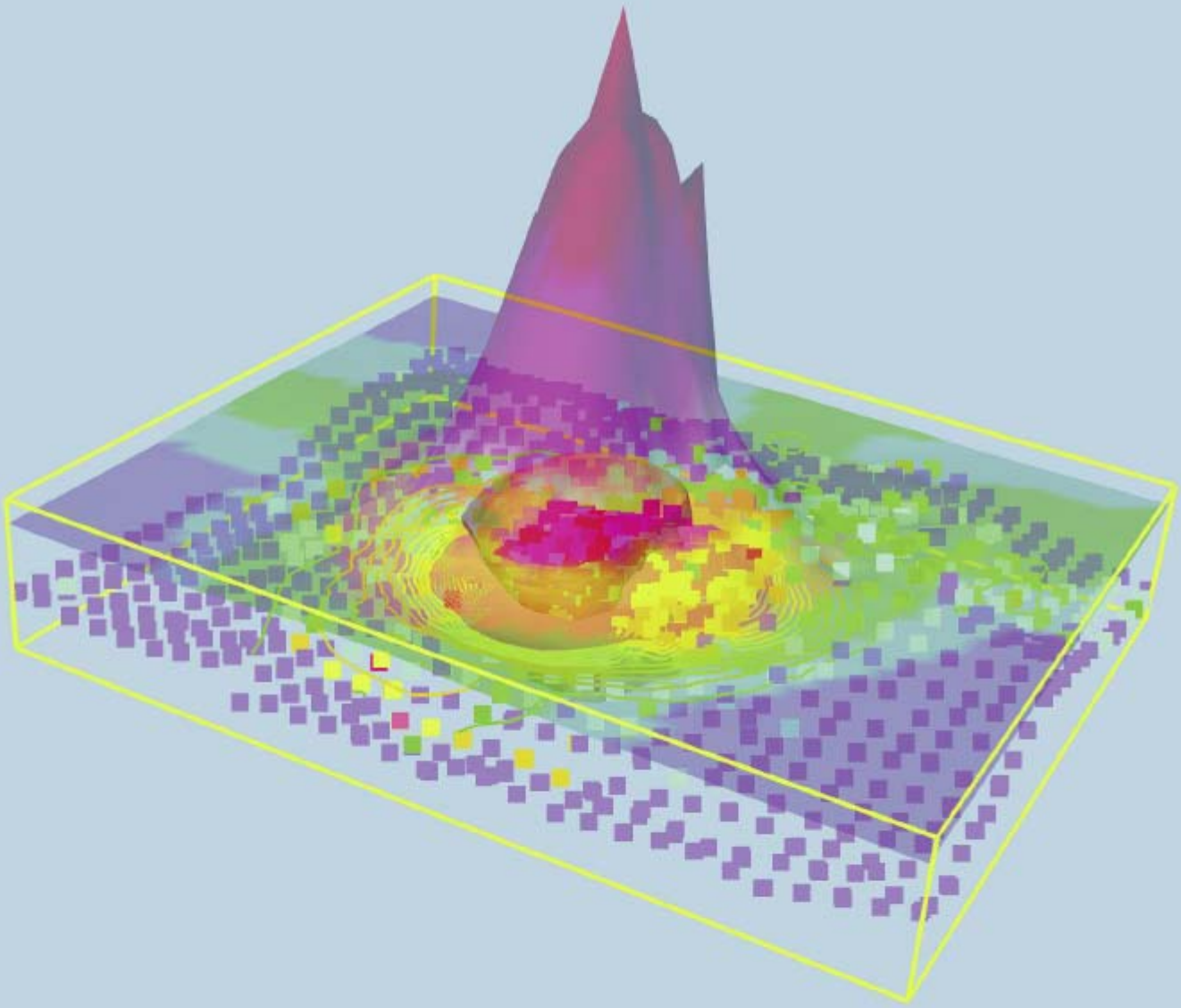
Monday through Friday, 8:00 AM - 5:00 PM, Mountain Time

Product Sales:

Online orders available 24 hours, 7 days a week with 2 business hour delivery

Golden Software Contact Information

www.goldensoftware.com
voxlersupport@goldensoftware.com
phone: 303-279-1021
fax: 303-279-0909



www.goldensoftware.com