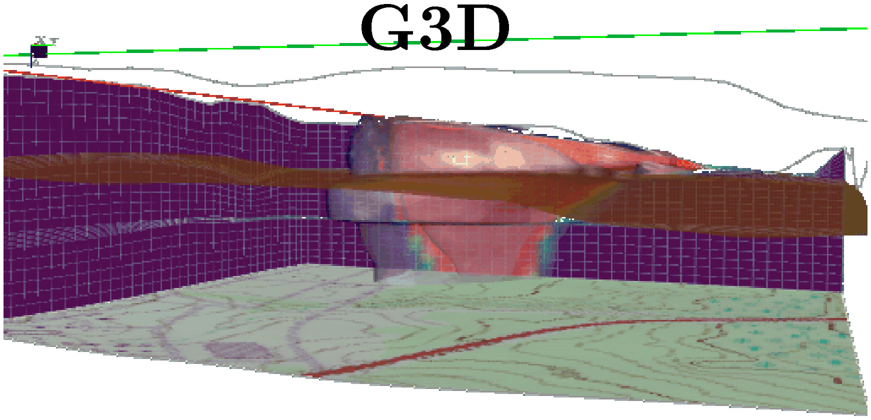
G3D



(A Contaminant Plume Infiltrating a Clay Layer)

A 3D User Environment for Partial Differential Equations

by

L. Scott Johnson, Alpana Kaulgud,
and Robert C. Sharpley

Version 1 (12/15/96)
Revised (5/8/97)

Contents

1 Introduction.

G3D is a graphical user interface designed to facilitate model development and understanding of complex dynamics governed by partial differential equations. **G3D** assists in the preparation of data for simulators based on three dimensional logically rectangular grids. The grid may be created by modifing an existing grid or by constructing new ones from a simple cube using 3D editing tools. **G3D** has the ability to simultaneously render multiple scalar and vector time-varying data fields.

1.1 G3D Development

G3D is written in OpenGL and Motif. By linking to the excellently designed Mesa 3-D graphics library, (http://www.ssec.wisc.edu/~brianp/Mesa.html) or to commercially available OpenGL products, **G3D** may be compiled on any UNIX machine running X-Windows.

The model type of the underlying simulator (**US3D**, for example) determines the type of equations used. The groups of coefficients expected by those equations are specified though ‘material types’ in **G3D**, but may be easily modified for other systems of partial differential equations (see Appendix III). Additional model parameters, such as initial conditions and boundary conditions, are also configured through **G3D**. Thoughout the specification process, **G3D** employs graphical tools and visual feedback to provide data-checking mechanisms for input errors.

1.2 Main Features

In this brief handbook, we describe the major features of the graphical user interface and of the interactive utilities. These include

- data preparation and pre-processing.
- interactive control on remote machines
- visualization tools and post-processing.

The tools and utilities developed for post-processing may be used separately or in tandem to visualize data as it is prepared for simulations or during the course of an interactive session. After a brief introduction to the primary features of the Main Window, we provide three tutorials (see Section 3) which illustrate the use of tools available through **G3D**. We encourage the user to review these tutorials, but emphasize that the steps described are only suggestive and do not necessarily have to be followed in a precise order. The Tutorials are provided to orient the user and quicky illustrate **G3D**’s code design and capabilities.

1.3 Distribution

The latest versions of **G3D** and accompanying materials may be obtained through the World Wide Web address

http://www.math.sc.edu/~sjohnson/g3d.html.

or by anonymous ftp at ftp://ftp.math.sc.edu/sharpley/outgoing/VIZ/g3d.tar.gz

1.4 Scheduled Development

Several features designed to add increased functionality are in various stages of development. Many of these are being pursued through joint efforts with our collaborators: heirarchial views for large data sets, additional import/export filters (Ron Peierls, BNL), cut-aways and censored views, switching view to different data sets, off-line generation of **G3D** rendered animations from an interpolated key frame file, bounds checking, a nonorthogonal slicer (Scott Goodrich, Texas A&M), stream and tube-lines (Brent Linquist, SUNY-SB).

1.5 Acknowledgements

We would like to acknowledge the work of several students at the University of South Carolina who have assisted in the code development over the last two years, including Ben Cohen, Parag Paranjpe, Seth Wandersman, and Jim Blair. Several of our colleagues have contributed in a variety of ways, particulary by helpful comments and suggestions on user perferences and added functionality. We would like to extend special appreciation to Dick Ewing, Larry Ewing, Michael Pilant, and Scott Goodrich of Texas A&M University, Michael Celia of Princeton University, Lin Ferrand of CUNY, John Haselow, Larry Hamm, and Greg Flach of Westinghouse Savannah River Technology Center, and Ron Peierls of Brookhaven National Laboratory for many helpful comments and suggestions.

We also take this opportunity to mention that **G3D** was formulated incorporating in part many of our earlier experiences in developing GUI prototypes (‘v3d’, ‘c3d’ for postprocessing, and ‘Pre’ for preprocessing) which were funded by the University of South Carolina, SCREF, University of Wyoming, Texas A&M University, and the Department of Energy through its Partnership in Computational Science program. The main SGI-GL code development of the early prototypes was done by Patrick O’Leary, Derek Mitchum, and Alan Perry at the University of Wyoming, University of South Carolina, and Texas A&M University.

2 Main Window.

To start **G3D**, type either '**g3d**' or '**g3d file**' where 'file' uses the formatting specified in the distribution file 'FILE_FMT.doc' (see also Appendix I). Once initialized the main draw window appears as in Figure 1. Notice that menu items run along the top of the window and may be 'torn-off' for easy selection of procedures (such as Redraw) and sub-menu items. Simply pull-down the menu item and select the dotted line. You may then place this widget in a convenient location.

2.1 Pull-down Menu Items:

1. **File**

   Creates models, Saves, and Exports (to specific formats). Also saves the Set-up for a future session and Exits the application.

2. **Edit**

   Provides tools to create and edit model data. Widgets provide editing of grid attributes (rescaling and centering), editing the grid in 2-D and 3-D, and 3D editing of initial conditions, time-varying boundary conditions, and material properties and their location. See Section 5 for a description and Tutorial #2 (Section 3.2) for their use.

3. **Draw**-

   Provides various drawing options which include: a new window with a different perspective, toggle controls for drawing objects (axes, wireframe, and the grid lattice), radio-buttons for coordinate system (right- or left-handed), colormaps, lighting model, stereo view, and other options (smoothing, depth-cuing, wide angle perspective, and draw area size). See Section 4 for a description of setting these selections.

4. **Tools**-

   Provides the primary drawing tools for scalar and vector data (multiple orthogonal slices with transparent overlay of an image, precise view control, cell contouring and multiple isosurfaces, with transparency). Section 4 has a description of each tool and the Tutorials in Section 3 show their use.

5. **Remote**-

   This tear-off widget provides for remote tracking and steering of simulators through a connection interface, a list of registered variables (for interactive tracking) and a list of registered parameters (for interactive steering). See Tutorial #3 in Section 3.3 for additional details.

2.2 3-D Select and View Control Icons:

Beneath the row of Menu items, appears a row of radio-button icons for point, line, plane (in each coordinate direction) and 3D region selections. By choosing an icon, one may use the left mouse button in the Draw area to select points, lines, planes, and regions in 3D. For Line and Region selection, one uses 'clicking and dragging': the mouse-down will select the first point and mouse-release will select the second vertex.

Following the 3D-Select icons, are three radio-buttons for View control (Rotate, Pan, and Zoom). Selecting one of these buttons provides a *left mouse button* view control with that attribute by 'clicking and dragging'. For example, selecting **Pan** allows one to translate the model data. Selecting **Zoom** and moving the mouse up and down zooms in and out, while selecting **Rotate** allows one to change the orientation of the model data (as if the user had a handle from the mouse position to the center of the data). More precise views (important for creating animated sequences) can be obtained using the **View** item under **T̲ools**, which brings up a tightly coupled **View** widget.

By selecting **Zoom** on a *three-button mouse*, the middle button can be used for rotation and the right button can be used for Pan. On a two-button mouse the *middle button* may be realized by simultaneously depressing both buttons.

2.3 Time Frame Control Widgets:

The next row of the Main **G3D** Window are, respectively, the time-selection control widget, auto-animate toggle-button, and time-interpolate and time-delete widgets. These last two widgets are used to interpolate and delete (3D) time frames to assist in producing animations and will be described under the **T̲ools** Section.

2.4 Main Draw Area:

The Main Draw Area is the main rendering window for **G3D**. In the upper left corner is drawn a small view of data and axes to remind the user of the current orientation of the data set. The x, y and z axes are colored red, green, and blue respectively. In the main draw area, objects such as grids, isosurfaces, rock layers, and so on will be rendered. The user will also use the mouse in this area to manipulate objects as we described, for example, in the previous paragraph concerning view control ('clicking and dragging' the mouse in the drawing area provides joy-stick control of the orientation).

2.5 Color Bar Information and Application:

At the bottom left of the draw area is an integer which indicates the time frame, followed by a color bar to indicate the color map being used. Beneath this are two text entry fields which allow the user to give (as floats) the minimum and maximum values for application of the color bar to the data. The numbers below each field are the min and max of the scalar field currently viewed.

3 Brief Tutorials.

In order to begin **G3D**, type '**g3d**' or '**g3d file**' where the file is formatted as described in the file 'FILE_FMT.doc' (see Appendix I). Once initialized, the main draw window appears (see Figure 1). In the main draw area will be a wireframe outline of the current grid, or a 2×2×2 cube if no file was specified on the command line. The three orthogonal x-y-z axes are colored red, green and blue, respectively. The icon in the upper left hand corner is a visual reminder of the data's orientation. In the next section we begin with a tutorial which illustrates vizualization features applied to two data sets. Familiarity with this exercise will be useful in all that follows. The purpose of Tutorial #2 is to create a simple grid, edit it and then specify (in independent order) the model's possibly heterogeneous material properties, initial conditions, and time-dependent boundary conditions. Tutorial #3 illustrates the **Remote** Tracking and Steering capability of **G3D** coupled with the Interactive Controller library.

3.1 Tutorial #1: Visualizing Data Sets.

Data Set 1

Give the command 'g3d DataSet1.g3d'. This data set (see Appendix II) has the format described in 'FILE_FMT.doc' (see also Appendix I) and is a good illustration of the simplicity of the data format.

Choose **Tools→Slice** to bring up the Slicer Widget. Slice through the data by using the forward/back arrows or by entering the slice index directly in the text field. Select x, y and z slices of 2, 3 and 3 respectively. Select the **Zoom** icon in the Main Drawing Window and use the mouse in the Drawing Area to change the view as described in Section 2.2.

Choose **Tools→Contour** to expose the **Contour** widget which is used to draw isosurfaces. Select **Cell Contour** and enter the range **From** .2 **to** .7 into the text fields. The cells with values in this range will be drawn. When done, toggle **Cell Contour** to 'off'.

Select **Contour**, enter a value of .2 into the text field and observe the Main Drawing area (see Figure 2). Changing the slider bars for opacity and contour value will produce different effects. To change the lighting model option, select **Draw→Options→Lighting→Front**.

Select **File→Quit** when done.

Data Set 2

For a more interesting data set, select 'DataSet2.g3d' by giving the command 'g3d DataSet2 .g3d'. You may choose various options to view and post-process the simulation. Begin, for example, by choosing a right-handed coordinate system from the widget (**Draw→Coord. System**).

To provide a translucent overlay wrapped to the top surface of the grid, use the **Image** button of the Slices widget to load the image 'topo.ppm' which is included in the downloaded tar-file. Do this by selecting the slice $z = 1$ from the **Slices** widget and toggle the radio-button **I** to the right of the $z = 1$ text field. The **Opacity** may be changed by entering appropriate values in the text field. Select other slices to show the griding (e.g. choose $x = 51$, $y = 41$) and change the view using the mouse buttons as described in Section 2.2.

Change the z slice to 31, and x and y slice text fields to 40 and 1, respectively. Select the menu option **Edit→Materials→Regions** and in this widget choose **Display: Selected**. Select the 'tan clay' region ((1,1,8)-(51,41,10)) from the list of material regions. Animate in time using the time control widget in the Main Drawing Window. Adjust the application of the color map, by use of the text fields for the minimum and maximum.

To add isosurfaces, choose **Tools→Contour** to expose the **Contour** widge. Select the first **Contour**, enter a value of .003 in the text field and set the **Opacity** to 60%. Changing the slider bars for opacity and contour value will produce different visual effects. Select the second **Contour**, enter a value of 1.*e*-4 in the text field and set the **Opacity** to 40%. Select the lighting model option **Draw→Options→Lighting→Front**.

A sample rendering of the actions produced by this tutorial is shown in Figure 3 and on the front cover. Select **File→Quit** when done.

3.2 Tutorial # 2: Simple Model Construction.

This tutorial will show how to create a simple grid and edit it. We will then specify (in independent order) the model's heterogeneous material properties, initial conditions, and time-dependent boundary conditions. Miscellaneous model parameters, such as simulation times and step size, error tolerances for the linear and nonlinear solvers, choices on solvers and quadrature method may be set through **G3D** sub-menu item **Misc** under the **Edit** menu item. The user may also read an existing **G3D** data set and edit it.

3.2.1 Creating and Editing a Simple Grid:

1. Under the **File** menu item, select **New Model**. In the widget that appears, change the number of vertices from a 2×2×2 model to an initial one with 5×4×2 vertices (i.e., 12 cells).

2. Under the **Draw** menu item, select **Objects** and click the **Lattice** selection box to toggle it on. In short, toggle on the **Draw→Objects→Lattice**, in order to draw the edges of the interior cells.

3. 'Tear off' the **Edit** Menu. The **Undo** button is used to correct or 'undo' editing mistakes. Select the **Edit→Grid** widget (see Figure 4) to edit and refine the grid. Also, 'tear off' the **Draw** menu, so that the **Redraw** button is available for use to view the resulting changes in the Main Window.

4. Select the **Edit Slice** button on the Grid Edit widget to bring up an initial slice editing window (see Figure 5(a)) which contains the currently selected slice (X-slice 1). The default editing mode (i.e., **2D** radio button='on') is to edit within this slice, holding the x-index constant and altering the y-z coordinates of the grid. Setting this to 'off' will allow editing out-of-slice (as described in Step 5 below when we edit z-slices). Adjust vertices in the $x = 1$ slice by selecting the **Pick** radio-button to pick vertices. Click and drag to change a vertex's position. Notice that the **Vertex** and **Coords** widgets below the Draw Area change accordingly. In fact, precise coordinate positions may be entered in the **Coords** text fields and vertices of large, dense data sets may be precisely selected by using the **Vertex** text fields.

   Note that changing the text field of **Vertex-x** changes to the corresponding x-slice in this edit window, since this window resulted from selecting EditSlice with **Axis:** x chosen in the Grid Edit widget. Also, note that the View controls for in-slice editing only permit rotations about the x-axis, and the drawing is a projected view of the main window in this mode.

   You may 'Dismiss' these edit-windows when done.

5. Using the EditGrid widget (Figure 4), choose **Axis:** $\boxed{z}$ and select the **Edit Slice** button to choose a z-slice to edit. We will move some of these vertices out-of-plane to illustrate adding ‘topography’ to the model. In the EditSlice Window toggle the **2d** option to ‘off’ and select the View buttons to **Pan**, **Zoom** and **Rotate** the object so that the azimuth is at aproximately a 45° angle with the horizon. *2-button* and *3-button* mice behave in View mode in the exactly the same manner as described in the Main Window section.

6. Choose the **Pick** radio-button, and select a vertex. To change the z-coordinate of the vertex, select and drag. By editing the **z-Coords** text field, you may precisely adjust the z-components of the vertices in that logically-rectangular z-slice as well. **Redraw**-ing the Main Window (Figure 1) shows the resulting changes. **Undo**’s also update and re-render both the Main Window and the EditSlice Window as corrections are made. Again, when done, you may ‘Dismiss’ this EditSlice Window.

7. Refine the current basic grid using the **Insert Slices** button of the EditGrid widget (Figure 4) to interpolate two additional slices between the top and bottom of our 5×4×2 vertex model. Select **Axis:**$\boxed{z}$ and choose the z=1 slice by entering 1 into the text field at the top of the widget. Just below the **Insert Slices** button, change the text field of the widget from the default value of 1 to 2. Select **Insert Slices** button to uniformly interpolate 2 slices from the currently selected slice (in this case, z slice = 1) to is successor.

   Additional x- and y- slices may be inserted by choosing the corresponding **Axis:**$\boxed{-}$, the number of slices to insert, and selecting the **Insert Slices** button.

8. Select **Edit→Attributes** and change the Z-Scale View text field from 50.00 to 8.00 and select **Redraw**.

A sample grid constructed in this way (see Figure 6) has corresponding x slice=1 as shown in Figure 5(b). Now that the grid is constructed, the other parameters may be set in arbitrary order. We first set the coefficients of the Partial Differential Equations (i.e. Material Properties), followed by Initial and Boundary Conditions. The grid may be altered, however, during or after any of the operations on the other parameters. Since each of these tools uses ‘patches’ determined by ranges of vertex indices, an insertion of additional slices may require an appropriate adjustment of the patch ranges for properly specifying the medium’s properties.

3.2.2 Editing Media Properties:

1. ’Tear-off’ the **Edit→Materials** widget and select **Properties** to expose the Material Properties Widget (see Figure 7) which produces a list of properties with corresponding parameters to specify coefficients to be entered into the Partial Differential Equations. The ‘Form’-button’s function and how to tailor the form to specify groups of coefficients for a particular model formulation is described in detail in Appendix III.

2. In the ‘Add Material’ text field, enter ‘sandy clay’. In the ‘Color’ text field, change ‘Black’ to ‘tan3’ and edit the parameter text fields in this row accordingly. Add a second material ‘clay’ with color ‘green3’. You may enter additional Materials now and change values for the various model parameters from their default values or dismiss the **Properties** widget and return at later time.

3. From the **Edit→Materials** widget, select **Regions** to expose the Material Regions Widget (See Figure 8) which contains an editable list of 3D regions (sub-grids) of material types

chosen from the Material Properties list. Choose option **Display - All**. The material properties are determined by the order in the list, with suceeding patches overwriting in their region the values determined by those patches that come before.

4. Select the **Add Region** button to expose the Region Specifier Widget (see Figure 9) and select **Material** **sandy clay**. Choose the '3-D Select-Region' icon of the Main Window (see Section 2.2 and Figure 1) and in the Draw Area select antipodal vertices by clicking on one vertex and dragging to the other. This determines a 3-D region outlined in 'Red' in the main Draw Area. Choose **Import Selection from 3-D Window** to import indices into the text fields. Select **lock** to 'lock' these text fields to the outlined region in the Main Draw area. The region may then be adjusted and simultaneously viewed in the Main Drawing area by adjusting the values in the index widget. Select **OK** after adjusting the bounding box to Min(1,1,1) and Max(17,15,2). (Be sure to select 'unlock' before dismissing this widget.) Add a layer region of **clay** with a bounding box from (1,1,2) to (17,15,8) and a **sandy clay** layer from (1,1,8) to (17,15,9).

5. Add 'green clay' material type to the list of Material Properties (see Figure 7) with color 'brown3'. Add a **green clay** region from (4,10,2) to (13,15,8). Change the **opacity** slider bar to 40% to view the layered model as in Figure 10.

3.2.3 Editing Initial Conditions:

Select **Edit→Initial Conditions** to expose the Initial Conditions 'patch list' Editor Widget which allows the user to add, edit, or remove 3D patches of initial conditions in much the same way as performed for material regions (see Section 3.3.3 (3) and Figure 8). The initial patch contains a default value of zero over the entire grid. For each patch one may enter a constant value or select the linear option for a linear variation over the patch based on the (x, y, z) components of each vertex (or cell-center) in the patch. The region to be overwritten must be specified by setting the text fields for the region's bounds. The user may wish to import a selected region from the Main Drawing area. By selecting **Lock** and adjusting the bounds, the user may view the selected region (outlined in red) in the Main Drawing area until a satisfactory region is selected. Press **Apply** to record the patch in the patch list and apply its action. The initial conditions are determined by over-writing the actions of the previous patches with the values specified in this patch. Select **Dismiss** when done editing the Initial Conditions patch list.

Render the initial conditions using **Tools→Slice** and select slices $x = 17$, $y = 15$ and $z = 1$. A typical completed data set is shown in Figure 12.

3.2.4 Editing Boundary Conditions:

1. Select **Edit→Boundary Conditions** to expose the Boundary Conditions Patch List (see Figure 13). The initial list begins with the 6 'faces' of the logically rectangular grid. This 'patch list' is edited in a similar manner as the Material Regions list. Additional 2D and 3D patches of Boundary Conditions may be added to this initial list.

2. From the list, select the patch with coordinate bounds (1,1,1)-(17,15,1), i.e. the 'top face' of the grid with z-index =1. The exposed Patch Editor widget (see Figure 14) is used to specify this boundary patch. The Patch Bounds may be edited, or imported a selected region in the

3D drawing area. The user may select **Lock** and adjust the index text fields while viewing the patch outline in the Main Draw Window to select the desired patch bounds.

Either Dirichlet, Neumann, or Robin (Mixed) boundary conditions may be chosen. Mixed boundary conditions require two values which are entered, separated by a colon. A linear variation option may also be selected and is based on the (x, y, z) components. Select **Type: Dirichlet** for the boundary condition. Enter a constant Dirichlet boundary condition value of -5 in the **Value** text field and select **OK**.

3. To apply time-varying Dirichlet conditions to a subregion of this or any other existing patch, enter the values and the end time for that value for each entry of the time series into the table (see Figure 15.a).

   To view the graph of the time series, select **Graph** (see Figure 15.b). To dismiss the graph, click within the Graph Window. Select **OK** when done editing or **Cancel**, as appropriate.

4. For new patch, select **Add Patch** to add a patch after the one currently selected in the list and proceed as above. The boundary conditions are determined by overwriting patches in space-time using the order in which the patches appear in the patch list.

3.3 Tutorial #3: A Remote Interactive Session.

This tutorial illustrates the coupling of **G3D** with the interactive controller for running applications on remote parallel machines. It suggested that the user review Tutorial #1 (Data Set 1) before proceeding with this exercise. This particular application uses **G3D** on the client workstations and a Paragon as the remote server. For a full description of the Controller Library (Documentation, FAQ, man pages, and example programs), the user may obtain the full package (SteerLib.9aug96.tar.gz) from the anonymous ftp site: ftp.math.sc.edu under the directory 'sharpley/outgoing/Controller'.

3.3.1 Setting up the Parallel Server for a Remote Session.

Download to the Paragon, the tar file (Controller_server.tar.gz) of the executable for the groundwater flow and transport simulator (**us3d***) and related files. The tar file is available at the URL (http://www.math.sc.edu/~sjohnson/g3d/Paragon_tar.html) or from the anonymous ftp site

ftp.math.sc.edu

A directory 'Celia_server/' is produced after uncompressing and untarring the file. Within this directory are the executable **us3d*** and related binary data files which are used for parallel I/O. In this example simulator, we have previously registered variables for tracking (pressures, concentrations, nodal data for solver behavior,...) and a parameter for steering.

3.3.2 Setting up the Local Client for a Remote Session.

On the local workstation (i.e. client), move to the corresponding directory 'Celia_client/' which was obtained from the **G3D** distribution file 'g3d.tar.gz'. Give the command 'g3d CeliaGrid.g3d' to load the grid for the interactive session. We need to choose options to view and render the simulation. Select **Draw→Coord. System** and choose the **Left Handed** option. Choose slices of x=8, y=18 and z=6 from the **Tools→Slice** widget. Choose an isosurface with a value of .003 from the **Tools→Contour** widget.

3.3.3 Running interactively.

Select the **Remote→Connection** widget and enter the (1) internet address of the remote server (i.e. Host), (2) 'cd Celia_server; pexec us3d -sz 4' in the Application text field, (3) user name and (4) password. Selecting **Start** initiates the remote commands on the server. (See Figure 16) In the text window, the direct output is displayed. Once registration of variables (Tracking) and Parameters (Steering) is complete, select the **Connection** button to register the corresponding variables on your client workstation.

Select the **Remote→Variables** widget which lists all registered tracking variables. You may enter any time step in the **Time Index** text field for variables that have been computed. An option for 3D data compression may be selected for transfer of large data sets. The value of -1 selects data for the current time step. You may select either variables 'pressure' or 'concentration', remembering that the Min-Max values of the color map need to be adjusted accordingly. Additional data from individual processors are also available for debugging and observing solver behavior.

First set the color map text fields in the Main Window to **Min**=-30 and **Max**=70 and select the current pressure data by entering -1 into the **Remote→Variables** widgets **Time Index** text field and selecting **pressure**. In the Main Window, advance the time control to view the current pressure field.

Next change the color map text field to **Min**=0. and **Max**=.01. Select the current concentration data by entering -1 into the **Remote→Variables** widgets **Time Index** text field and selecting **concentration**. Previous time-indexed concentration data sets may be retrieved and viewed at any time. Requesting them will automatically load them into **G3D** and overwrite previous data with that time index. You may use the **Delete** button in the Main Window to delete unwanted data with the current Time Index.

The results of using a 3D data compression option (using **Remote→Compression**) are illustrated in Figure 17 for a refined version (129× 137× 11) of the Celia data set which was used in Tutorial 1. Figure 17.a shows the results of rendering the original data set while Figure 17.d is a rendering of the data file reconstructed after lossy compression by a factor of 1200. 3D Data Compression is not included in all distributions. Please contact Bob Sharpley (sharpley@math.sc.edu) for more information.

To illustrate the capability of steering, select the **Start** button of the **Connection** widget. After variables and parameters are registered (observe the direct output text window in the widget), select **Connect**. Select **Remote→Parameters** to bring up the widget of all registered parameters. In this tutorial, only the concentration level of a specific cell is registered. Change its value to 1 and enter Return. Select an additional slice of y=2 in the **Tools→Slice** widget to observe the effect of introducing an additional contaminant at that location.

See the Frequently Asked Questions file ('FAQ'), which is included in the Interactive Controller library distribution, if there are problems with communication or security issues at a particular site. Several workarounds for facilties with firewalls are described.

4 Visualizaton Tools and the Main Draw Area.

The menu items **Draw** and **Tools** contain the main rendering options for scalar and vector data. However, the user may also combine rendering using these options with rendering of media properties was done with DataSet2 in Tutorial #1.

4.1 Draw Menu Options for Data Rendering.

The rendering options for scalar data are accessed through the **Draw** menu. These options include:

- **Redraw** -
  Re-renders the Main Drawing Area (with the current settings).
- **New Window** -
  Produces a new drawing area with view controls to provide a different perspective of the model data.
- **Objects** -
  Toggles the drawing of axes, wireframe, grid lattice, and vectors in the Main Drawing Area.
- **Coordinate System** -
  Offers a choice of a left- or right-handed coordinate system.
- **Palette** -
  Offers a choice of color maps or greyscale.
- **Lighting** -
  Offers a choice of lighting models through selection of one of the three radio-buttons. This affects, for example, the rendering of isosurfaces.
- **Options** - toggle buttons for changing the drawing modes:
  - **Smooth** -
    Toggles Gouraud-shading on and off for both Slices and Isosurfaces (see the **Tools** description). Smoothing-off colors exact cell values and renders isosurface contouring as triangulated surfaces.
  - **Depth Cuing** -
    Toggles the use of 'fog' for additional 3-D depth effect.
  - **Wide-Angle** -
    Toggles wide-angle perspective view for additional 3-D depth effect.
  - **Draw Size Area** -
    Exposes a widget to set the exact number of pixels of the Main Draw area for the recording of images. For rough sizing, one may resize the window in the standard way by selecting the lower right corner and dragging.
- **Stereo 3-D** -
  Toggles the monitor to stereo mode (on systems with stereo capability), and renders images accordingly.

The **Tools** menu includes options to render multiple slices and isosurfaces for the scalar data in addition to providing tools to record/playback a simulation session and to provide viewing controls.

4.2 VCR Tool:

By selecting **Tools→VCR**, a VCR control widget is exposed for recording and playback (see Figure 18). This widget has icons to open a tape (i.e. a directory), record, playback, frame advance, and continuous looping. Images are stored in 'rgb' or 'ppm' image format in the specified directory. The images are automatically sequenced as they are recorded and a file is kept in that directory of all key-frame information. For an example of an MPEG movie generated by **G3D** using DataSet2.g3d of Tutorial #1, see

'http://www.math.sc.edu/USC_PICS_4a.html'

or download the file

'ftp://ftp.math.sc.edu/sharpley/outgoing/Movies/ConNew.mpg'

4.3 Orthogonal Slices:

The **Tools→Slice** Widget renders 'slices' of scalar data and permits overlay images as described below. Currently 4 slices in each of the coordinate directions of the logically rectangular gridded scalar field are available.

4.4 Overlay Image:

Overlay images are useful to help the practitioner overlay sketches or contour maps for geographical orientation. Selecting the **Image** button on the widget, allows the user to overlay 'ppm' image format files over the appropriate slices by selecting the radio button **I** to the right of each of those slices (see Figure 19).

4.5 Isosurfaces:

Selecting **Tools→Contour** exposes a widget to allow specification of multiple isosurfaces. Currently the user can chose up to 2 isosurfaces to be simultaneously rendered by selecting the appropriate toggle boxes. Contour values are entered into the text fields and the opacity for each isosurface is set using the corresponding slider bars (see Figure 20).

4.6 View:

Selecting **Tools→View** exposes a widget to change the view in the Main Draw Area to an exact setting. Also use of the mouse in the Main Draw Area (with a View icon selected) provides View information in the text fields of the widget.

4.7 Vectors:

Selection of **Tools→Vectors** exposes a widget to draw vectors at specific locations.

5 Model Generation - Preprocessing.

The menu item **Edit** provides tools to create, edit, and export model data.

5.1 Undo Edit Operations.

Selection of **Edit→Undo** undoes the last operation on the edit work stack.

5.2 Attributes for Grid.

Selection of **Edit→Attributes** brings up a widget to rescale and recenter the view.

5.3 Grid Editor.

Selection of **Edit→Grid** provides a widget to edit grids in 2-D and 3-D. Tutorial #2 (see Section 3.2.1) provides a description of the grid editor's capability.

5.4 Initial Conditions Editing.

Selection of **Edit→Initial Conditions** provides a widget to edit initial condition data using 3-D patches. Tutorial #2 (see Section 3.2.3) illustrates its use.

5.5 Boundary Conditions Editing.

Selection of **Edit→Boundary Conditions** provides a widget to edit time-dependent general boundary condition data by a list of patches. Options include Dirichlet, Neumann, and Mixed boundary conditions allowing either a constant value or a linear variation over each patch. See Section 3.2.4 of Tutorial #2 for use of this widget.

5.6 Materials Editing.

Selection of **Edit→Materials** provides a widget to edit (1) material types (i.e., properties) and (2) regions where they are applied using a list of 3D patches. See Section 3.2.2 of Tutorial #2 for use of this widget.

6 Interactive Control of Remote Applications.

Widget for remote tracking and steering of simulators. A full description of the Controller Tracking/Steering library is included in its distribution which available at the anonymous ftp site

ftp://ftp.math.sc.edu

under the directory 'sharpley/outgoing/Controller'.

6.1 Connection

This widget allows the user to login, launch an application on a remote serial or parallel machine, observe (tracking) its progress, and modify (steer) parameters interactively. Alternatively, the user may start the application (in another window, or with NQS) on the remote machine and use the Connection widget to interact over a period of time. The library permits multiple workstation (client) access to the remote application for tracking purposes. Only one client has permission to steer the simulation. See Section 3.3.3 for an example of its use.

6.2 Variables

The Variables Widget provides a list of all registered variables (which may only be tracked and not modified) in the application code on the remote machine. One must select the **Connect** button on the Connection widget after registration is performed by the remote application (this may be checked in the 'direct out' window at the bottom of the Connection widget.

6.3 Parameters

The Parameters Widget provides a list of all registered parameters (which may be both modified and tracked) in the application code on the remote machine. One must select the **Connect** button on the Connection widget after registration is performed by the remote application (this may be checked in the 'direct out' window at the bottom of the Connection widget. Current implementation with **us3d** allows modification of a single cell for demonstration purposes. An interactive well placement and pumping rate option is being developed.

6.4 Compression

The Compression Widget enables a lossy compression of large data sets over small bandwidth or congested transmission lines for realistic interactive work at remote supercomputer facilities. The amount of compression is determined by the threshold value text field. These compression algorithms are based upon biorthogonal wavelets (regular and hyperbolic) and were designed for general multi-dimensional non-square, non-power of 2 data. Further information is included in "Elementary Compression Methods on Wavelet Coefficients," by Z. Gao, A. Andreev, and R.C. Sharpley.

7 Appendix 1: G3D Data File Format

```
---
 *   The file must have the following format. Anything in brackets is
 *   optional.  Case doesn't matter. Tokens can be separated by
 *   colons, commas, and/or whitespace.
 *     SIZE: nx, ny, nz
 *     SCALE: x, y, z
 *     CENTER: x, y, z
 *     TIME: [FROM] t1 [TO] t2 [ [STEP] dt ]
 *     GRID: <...>  END
 *     DATA: <...>  END
 *     MATERIAL: <...> END
 *     BOUNDARY: <...> END
 *
 *   The token '#INCLUDE filename' redirects reading to the named file.
 *   Any other '#' marks the rest of the line as a comment
 *
 *   SCALE and CENTER will be calculated for best fit by default -
 *   only specify them in the file if you want to override the default.
 *   TIME defaults to one timestep, at time index one.
 *   GRID defaults to a grid of unit cubes starting at (0,0,0) - that is,
 *     the x,y,z coordinates of each vertex is equal to the i,j,k index
 *     of the vertex.
 *   DATA defaults to a scalar dataset named "Default", uniformly zero.
 *   MATERIAL defaults to GCT1.2 material properties and no regions.
 *
 * Order doesn't matter, except that SIZE must come before DATA, GRID, and
 *   MATERIAL.
---
 *   Grid data must be formatted as follows:
 *     vertex(1,1,1) X-coor,  vertex(1,1,1) Y-coor, vertex(1,1,1) Z-coor,
 *     vertex(1,1,2) X-coor,  vertex(1,1,2) Y-coor, vertex(1,1,2) Z-coor,
 *     ...
 *       ... data(xmax,ymax,zmax) Z-coor,
 *     END
 *
 *   Note that the Z index varies fastest, the X index varies slowest
 *   (standard C-style row-major array).
---
 *   Material data is formatted as follows:
 *   MATERIAL
 *     PROPERTY
 *       "Prop Abbr"  "Long Description"  default_value
 *       ...
 *     END
 *     CLASS
 *       "Class Name"
 *         "Prop Abbr", "Prop Abbr", .... "Prop Abbr" END
 *     ...
 *     CLASS
 *       "Class Name"
 *         "Prop Abbr", "Prop Abbr", .... "Prop Abbr" END
 *     GROUP
```

```
 *       "Group Name" group_code
 *         "Class Name", "Class Name", .... "Class Name" END
 *     ...
 *     GROUP
 *       "Group Name" group_code
 *         "Class Name", "Class Name", .... "Class Name" END
 *     TYPE
 *       "Material Name"
 *         "Prop Abbr"  prop_value
 *         ...
 *       END
 *       ...
 *       "Material Name"
 *         "Prop Abbr"  prop_value
 *         ...
 *       END
 *     END
 *     REGION
 *       "Material Name" [FROM] x0, y0, z0 [TO] x1, y1, z1
 *       ...
 *     END
 *   END
---
 *   Boundary Conditions are formatted as follows:
 *   BOUNDARY
 *     type [FROM] x0, y0, z0 [TO] x1, y1, z1
 *       TIME time_index: value [value]
 *       TIME time_index: value [value]
 *       ...
 *     END
 *     type [FROM] x0, y0, z0 [TO] x1, y1, z1
 *       TIME time_index: value [value]
 *       TIME time_index: value [value]
 *       ...
 *     END
 *   END
---
 *   DataSet must be formatted as follows:
 *     SCALAR | VECTOR n
 *      "Name"
 *     [RANGE [FROM] min [TO] max]
 *
 *     TIME time_index1
 *     data(1,1,1)  data(1,1,2)  ... data(xmax,ymax,zmax)
 *     TIME time_index2
 *     data(1,1,1)  data(1,1,2)  ... data(xmax,ymax,zmax)
 *     ...
 *     END
 *
 *   Note that the Z index varies fastest, the X index varies slowest
 *   (standard C-style row-major array).
```

8 Appendix 2: A Sample G3D Data File

The small example problem below may be extracted from ‘FILE_FMT.doc’.

```
---
size 3 5 5
center 2 3 3
time from 1 to 1
grid
1 1 1  1 1 2  1 1 3  1 1 4  1 1 5
1 2 1  1 2 2  1 2 3  1 2 4  1 2 5
1 3 1  1 3 2  1 3 3  1 3 4  1 3 5
1 4 1  1 4 2  1 4 3  1 4 4  1 4 5
1 5 1  1 5 2  1 5 3  1 5 4  1 5 5

2 1 1  2 1 2  2 1 3  2 1 4  2 1 5
2 2 1  2 2 2  2 2 3  2 2 4  2 2 5
2 3 1  2 3 2  2 3 3  2 3 4  2 3 5
2 4 1  2 4 2  2 4 3  2 4 4  2 4 5
2 5 1  2 5 2  2 5 3  2 5 4  2 5 5

3 1 1  3 1 2  3 1 3  3 1 4  3 1 5
3 2 1  3 2 2  3 2 3  3 2 4  3 2 5
3 3 1  3 3 2  3 3 3  3 3 4  3 3 5
3 4 1  3 4 2  3 4 3  3 4 4  3 4 5
3 5 1  3 5 2  3 5 3  3 5 4  3 5 5
end
data
scalar "Concentration"
range from 0 to 1
time 1
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0
0 .1 .2 .1 0
0 .2 1 .2 0
0 .1 .2 .1 0
0 0 0 0 0

0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0
end
end
---
```

9 Appendix 3: Setting Model Formulation and Properties

In order to allow flexibility in prescribing model formulations for a wide variety of physical problems, a forms layout cabability is incorporated into **G3D**. By inclusion of the file "mat.cfg" (see the example for flow and transport models below) in the 'g3d' file, one may specify the types of coefficients and other parameters used for a particular formulation.

First the user provides a list of 'properties' or parameters, followed by grouping of these parameters into rows which will be displayed on a text-based form in which users may easily enter parameters in the appropriate fields. Each property (i.e. parameter) has a (i) label, which will be printed next to its text entry field, (ii) a brief explanation of the parameter, which will be displayed in a dialogue at the bottom of the form whenever the cursor is positioned in that field, and (iii) a default value for the parameter. The 'rows' of the form (i.e. groupings of parameters) which are to be presented to the user are determined by 'group' type. The group type is actually used to specify which model formulation will be set for simulation.

In the case presented below, "group 1" is used for a single-phase flow-only model formulation (neither transport parameters nor 2-phase coefficients are required), while 'group 2' (single-phase flow with transport) requires additional coefficients which must be entered to override the pre-assigned default values. In this way one may tailor the forms to simplify model specification for general finite element codes. **G3D** automatically layes out a form with the fields appropriate for the given model. The model type of the forms widget is selected from Option cascade button shown in Figure 21.a. Selecting model type "Unsaturated, Flow and Transport" in this Option will alter the form as in Figure 21.b to indicate to the user the additional parameters required for that model.

A Model Formulation Example

Saturated/Unsaturated Flow and Transport

```
---
property
"rkxsat", "saturated conductivity in x-direction", 0.0
"rkysat", "saturated conductivity in y-direction", 0.0
"rkzsat", "saturated conductivity in z-direction", 0.0
"poros", "porosity", 0.0
"sstor", "specific storativity", 0.0
"thsat", "saturated moisture content", 0.0
"thres", "residual moisture content", 0.0
"coef1", "coef 1 for nonlinear fit to moisture retention curve", 0.0
"coef2", "coef 2 for nonlinear fit to moisture retention curve", 0.0
"displ", "longitudinal dispertivity", 0.0
"dispth", "transverse horizontal dispersivity", 0.0
"disptv", "transverse vertical dispersivity", 0.0
"dmolec", "effective molecular diffusion coefficient", 0.0
"aret", "nonlinear retardation coefficient equation constant A", 1.0
"bret", "nonlinear retardation coefficient equation constant B", 0.0
"cret", "nonlinear retardation coefficient equation constant C", 0.0
"dret", "nonlinear retardation coefficient equation constant D", 0.0
"eret", "nonlinear retardation coefficient equation constant E", 0.0
"arxn", "nonlinear reaction term equation constant A", 0.0
```

```
"brxn", "nonlinear reaction term equation constant B", 0.0
"crxn", "nonlinear reaction term equation constant C", 0.0
"drxn", "nonlinear reaction term equation constant D", 0.0
"erxn", "nonlinear reaction term equation constant E", 0.0
end
class "row0"
"rkxsat" "rkysat" "rkzsat" "poros" "sstor"
END
class "row1"
"thsat" "thres" "coef1" "coef2"
END
class "row2"
"displ" "dispth" "disptv" "dmolec"
END
class "row3"
"aret" "bret" "cret" "dret" "eret"
END
class "row4"
"arxn" "brxn" "crxn" "drxn" "erxn"
END
group "Saturated, Flow-Only" 1
"row0"
END
group "Saturated, Transport-Only" 2
"row0" "row2" "row3" "row4"
END
group "Saturated, Flow and Transport" 3
"row0" "row2" "row3" "row4"
END
group "Unsaturated, Flow-Only" -1
"row0" "row1"
END
group "Unsaturated, Transport-Only" -2
"row0" "row1" "row2" "row3" "row4"
END
group "Unsaturated, Flow and Transport" -3
"row0" "row1" "row2" "row3" "row4"
END
---
```