UCRL-MA-115696 Rev. 2

GRIZ

Finite Element Analysis Results Visualization for Unstructured Grids

User Manual

Douglas E. Speck

Methods Development Group
Mechanical Engineering

(Release 4.05)

April, 14 2005

Lawrence Livermore National Laboratory

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

GRIZ

Finite Element Analysis Results Visualization for Unstructured Grids

User Manual

Douglas E. Speck

Methods Development Group
Mechanical Engineering

(Release 4.05)

Thomas E. Spelce - Original Contributor

Donald J. Dovey - Original Contributor

April 14, 2005

TABLE OF CONTENTS

iv

ACKNOWLEDGEMENTS

GRIZ is an interactive application for visualizing finite element analysis results on two- and three-dimensional unstructured grids. It was developed by the Methods Development Group at Lawrence Livermore National Laboratory (LLNL). The current authors of the software are Elsie Pierce, Larry Sanford, and Doug Speck. The original authors were Don Dovey and Tom Spelce. Mark Christon wrote the original video driver interface and contributed to the initial design of the program. The isosurface generation code was adapted from software written by Brian Cabral. Dan Badders, Tony DeGroot, Bob Ferencz, Carol Hoover, Greg Kay, Tony Lee, Jerry Lin, Ting Lo, Michael Loomis, Michael Puso, Peter Raboin, Dale Schauer, Bob Sherwood, and Edward Zywicz of LLNL provided many suggestions for improvements to the software and this documentation. John Benner and members of the Engineering Analysis Group at Los Alamos National Laboratory have been exceptionally helpful collaborators, providing financial support, source code, and concise user feedback.

1.0 INTRODUCTION

GRIZ Version 4 is a general-purpose post-processing application supporting interactive visualization of finite element analysis results on unstructured grids. GRIZ reads several simulation data formats, including the Mili and TAURUS formats used by Methods Development Group (MDG) analysis codes and the Exodus II format used by some Sandia National Laboratory finite element analysis applications. GRIZ is designed to permit the addition of other data file format "drivers" as needed.

In addition to basic pseudocolor plots of simulation state variables, GRIZ provides typical visualization techniques such as isocontours and isosurfaces, cutting planes, vector field display, and particle traces. GRIZ provides flexible control of mesh materials on an individual basis, allowing the user to concentrate analysis and visual focus on important subsets of the mesh. GRIZ incorporates the ability to animate all representations over time. GRIZ also provides flexible XY plotting capabilities for plotting result histories versus time or other results.

GRIZ uses the OpenGL graphics library for rendering and the Motif widget toolkit for its user interface. In order to compile and run GRIZ, both of these libraries are required. Both are accepted standards on all major UNIX workstation platforms. To date, GRIZ has been compiled and run on SGI, Sun, IBM, HP, Compaq Alpha (formerly DEC), Linux (Intel), and Cray computers. The distribution source has been developed and maintained on SGI platforms.

GRIZ is part of a set of public domain codes developed at MDG. Other codes include the analysis codes DYNA3D, DYNA2D, NIKE3D, NIKE2D, TOPAZ3D, TOPAZ2D, PALM2D, MONT3D, and the mesh generation and post-processing codes MAZE, ORION, INGRID, and TAURUS.

This User Manual contains four chapters and an appendix. This introduction is Chapter 1. Chapter 2, OPERATION, describes the visual appearance and basic operation of GRIZ. Chapter 3, COMMANDS, provides a complete description of the GRIZ command set. Chapter 4, COMMAND QUICK REFERENCE, provides a compact listing and description of the GRIZ command set. Appendix A contains a description of command set changes since the previous revision of this User Manual, including previously un-documented commands introduced with Version 2 and modified or new commands for Version 4 (GRIZ Version 3 is a specialized variant of Version 2 developed to improve post-processing of parallel databases. It has not been officially released, but its functionality is being ported to Version 4 for future release).

2.0 OPERATION

This chapter gives an overview of the GRIZ program and its operation.

GRIZ is an interactive application, with a batch mode available, that provides users with three interactive modes of control: graphical, command-line, and direct manipulation via the mouse and cursor. The most common operations (rendering simulation state data and modifying the mesh view) are accessible via the graphical user interface controls (pulldown menus and optional Material Manager and Utility Panel graphical interfaces) and through direct manipulation of the rendered mesh. All of those operations, plus the rest of GRIZ's many commands, are accessible through GRIZ's command-line interpreter. Although GRIZ has only one command entry window, commands may be typed while the cursor is in any active GRIZ window, and they will be "forwarded" to the command entry window.

To run GRIZ, type the command

griz -i [***path*/**]***dbname* **[*switch*...]**

where *dbname* (mandatory) identifies the database being post-processed. GRIZ will read several database formats. The form of *dbname* depends on the type of database being post-processed. For multi-file database formats (MDG's Mili and TAURUS databases), *dbname* is the root name of the file family. For single-file databases (Exodus II), *dbname* is the complete name and extension of the file.

GRIZ has several optional command-line switches, that may appear in any order. They are described in Table 1, below.

Table 1. GRIZ Command-line Switches

Switch	Effect
-s	Run GRIZ in single-buffer mode. This maximizes color resolution but does not hide the rendering activity as does double-buffer mode (the default).
-b <command file name>	Run GRIZ in batch mode with offscreen rendering. GRIZ reads input commands from the specified file. To enable batch mode operation, GRIZ must be compiled with the symbol "SERIAL_BATCH" defined to the pre-processor and linked with a Mesa OpenGL library which was itself compiled with the offscreen rendering interface enabled.

Table 1. GRIZ Command-line Switches

Switch	Effect
-f	Run GRIZ in the foreground. Default behavior is to run in the background of the user’s command shell.
-v	Set the initial size of the rendering window to an optimal size for video capture (720 by 486).
-w <width> <height>	Set the initial size of the rendering window to “width” pixels wide by “height” pixels tall.
-u	Include the Utility Panel as part of the control window.
-beta	Run the latest beta version of Griz.
-V	Displays detailed info about the version of Griz that you are running and how the code was built.

GRIZ provides an “init” file capability that permits users to configure GRIZ at startup via a command file (i.e., a text file with one GRIZ command per line). Upon initial execution, GRIZ looks in the current directory for a GRIZ command file named “grizinit”. If the file exists, the commands in it are read and executed sequentially. If the file “grizinit” does not exist in the current directory, GRIZ looks for a file specified by the environment variable “GRIZINIT”. For example, one might place a line similar to the following in the .cshrc file.

setenv GRIZINIT /usr/people/smith/grizstuff/grizinit

An init file is not required for GRIZ to run properly. Its use, however, may be very convenient when a user is routinely repeating a sequence of commands at startup to configure GRIZ in a certain way.

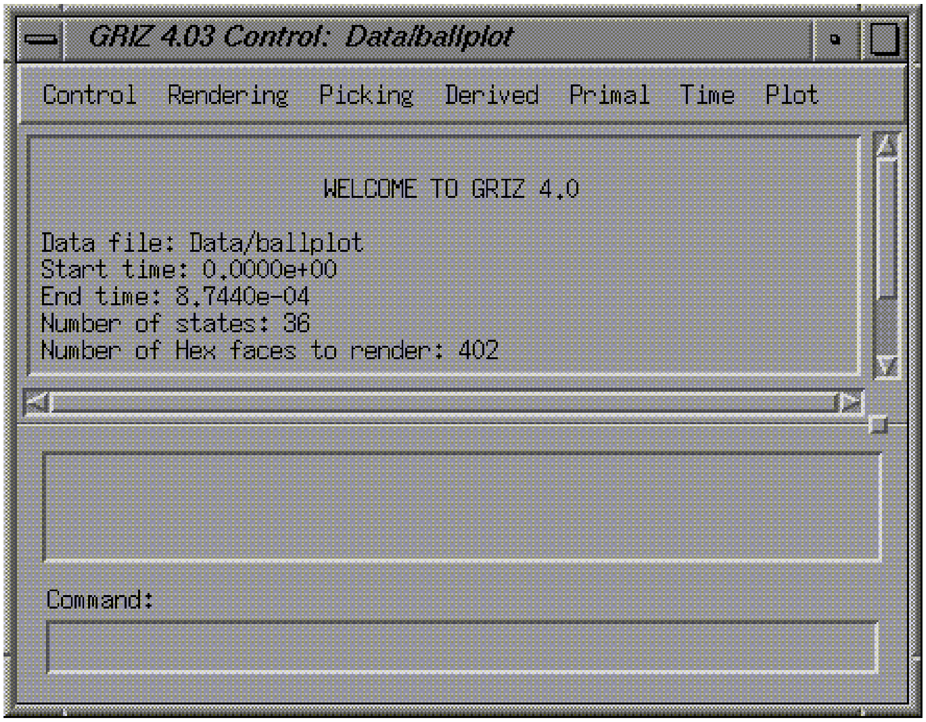
To terminate a GRIZ session, pick the **Quit** menu button at the bottom of the **Control** menu, type <Ctrl>-q while the cursor is in a GRIZ window, or type in “quit” in the command entry area.

GRIZ initially creates a rendering window and a control window. Visualization takes place in the rendering window; the control window provides primary access to the GRIZ user interface.

2.1 The Control Window

The control window, Figure 1, provides the main interface to GRIZ post-processing functions.

Figure 1. GRIZ Control Window



There are three primary elements in the control window. A pulldown menu bar provides mouse-driven access to a frequently-used command subset of the full command set. Below the menu bar is a scrollable text window containing textual information from a variety of sources. This information includes startup notices, information from the various interrogation commands, and info/warning dialog “overflow” messages. Below the feedback window is the command area consisting of a command history window and a command entry area. The full GRIZ command set is accessed by explicitly typing commands in the command entry area. Both commands typed in the command entry area as well as commands enacted via the pulldown menus are recorded in the command history window after execution. Any command recorded in the command history can be re-executed by double-clicking on its entry in the history window; previous commands can be modified by single-clicking on an entry in the history window to insert the command text into the command entry area where it can be edited and re-executed.

Optionally, the GRIZ Utility Panel (described below) will be included as an element of the control window if GRIZ is executed with the “-u” command-line switch. When included, the Utility Panel appears between the feedback window and the command history window.

Pulldown menu options are selected by depressing the left mouse button with the cursor positioned over the desired menu, then sliding the cursor over the options, releasing the mouse button when the cursor is over the desired option. A pulldown menu option with a small arrow to the right of the text indicates a “cascading” submenu. Positioning the cursor over such an option will cause the submenu to pop up.

The **Control** pulldown menu, leftmost on the menu bar, provides access to additional interface elements. These additional elements are the Material Manager and the Utility Panel, which are also accessed by typing <Ctrl>-m and <Ctrl>-u, respectively, while the cursor is in either the control window or the rendering window. The Material Manager provides a graphical interface for those GRIZ operations which affect mesh materials, specifically visibility/invisibility, enable/disable, and color. It is particularly useful when operating on a mesh with a large number of materials or when modifying color properties of materials. Its operation is described in detail in section 2.3 on page 11. The Utility Panel is an interface providing alternative access to a number of commonly used operations scattered among the pulldown menus. It is described in section 2.4 on page 16.

2.2 The Rendering Window

The rendering window (Figure 2 and Figure 3) serves two purposes. First, it is the destination for all of GRIZ’s visualization output, and second, it provides interactive manipulation and picking functions using the mouse and cursor.

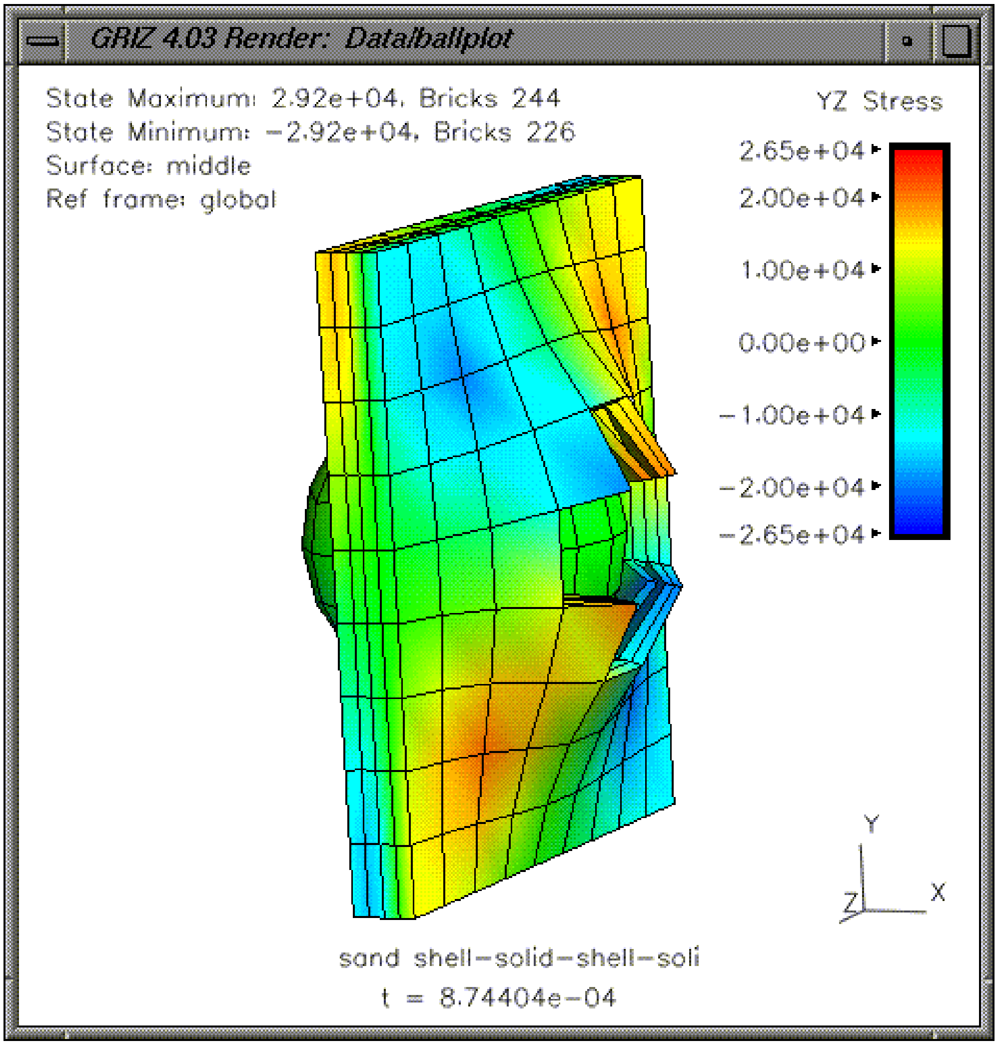
GRIZ renders mesh information in one of two modes - mesh view mode or plot mode. In mesh view mode, GRIZ renders the two- or three-dimensional mesh with state information conveyed via element shading and various text annotations. In plot mode, GRIZ renders time series information in a two-dimensional X-Y plot format with additional information provided by text annotations.

GRIZ transitions automatically between render modes based on the requirements of the most recently executed command or GUI function. In general, if the semantics of a command imply one mode, GRIZ will set the required mode. If a command applies to both modes or is unrelated to the mode (e.g. a “tell” query, page 46) then the current mode will be unaffected.

Mesh View Mode

Mesh view mode is GRIZ's initial rendering mode. This mode is utilized to investigate the state of the mesh (or a portion thereof) at a particular simulation time. All of GRIZ's mesh information visualization techniques, with the noted exception of time series plots, are viewed in this mode.

Figure 2. GRIZ Rendering Window - Mesh View Mode



These include pseudocolor or "fringe" plots (command "show" or pulldown menus **Derived** and **Primal**), vector plots, particle traces, isocontours and isosurfaces, and cutting planes. A variety of GRIZ commands are mapped to the mouse and cursor in this mode, enabling interactive picks (selections) of mesh objects and direct manipulation of the mesh view.

A mesh object pick is performed by maneuvering the cursor over a node, element, or particle and depressing whichever mouse button is configured to pick objects of that type (class). Picking objects executes or facilitates several GRIZ functions, depending on the current pick mode and which if any keyboard keys are depressed at the time of the pick. The pick mode is either "hilite" (default) or "select" and can be set via the **Picking** pulldown menu or the commands "switch pichil" and "switch picsel." See commands "hilite" (page 37) and "select" (page 37) for a description of the features associated with those states. Picking objects enables display of the exact result values associated with them, identifies them as default objects for time series plots, executes "tell position" queries when performed while the keyboard "shift" key is depressed, or executes material selections in the Material Manager when performed while the keyboard "control" key is depressed. Since there are a limited number of mouse buttons and an arbitrary number of object "classes" in a database that might contain pickable objects, GRIZ automatically assigns a pick class to each button when a database is loaded and provides facilities for letting the user interactively update the assignments. A prioritization scheme for the automatic assignments favors nodes for the first (left) mouse button, shell elements for the second button, and hexahedral volume elements for the third button. The user may update the pick class assignments via the "setpick" command, the "Set Btn ... Pick" submenus under the **Picking** pulldown menu, or via the "Btn Pick Class" option menus on the Utility Panel. The option menus on the Utility Panel have the added advantage of being labelled with the current assignments.

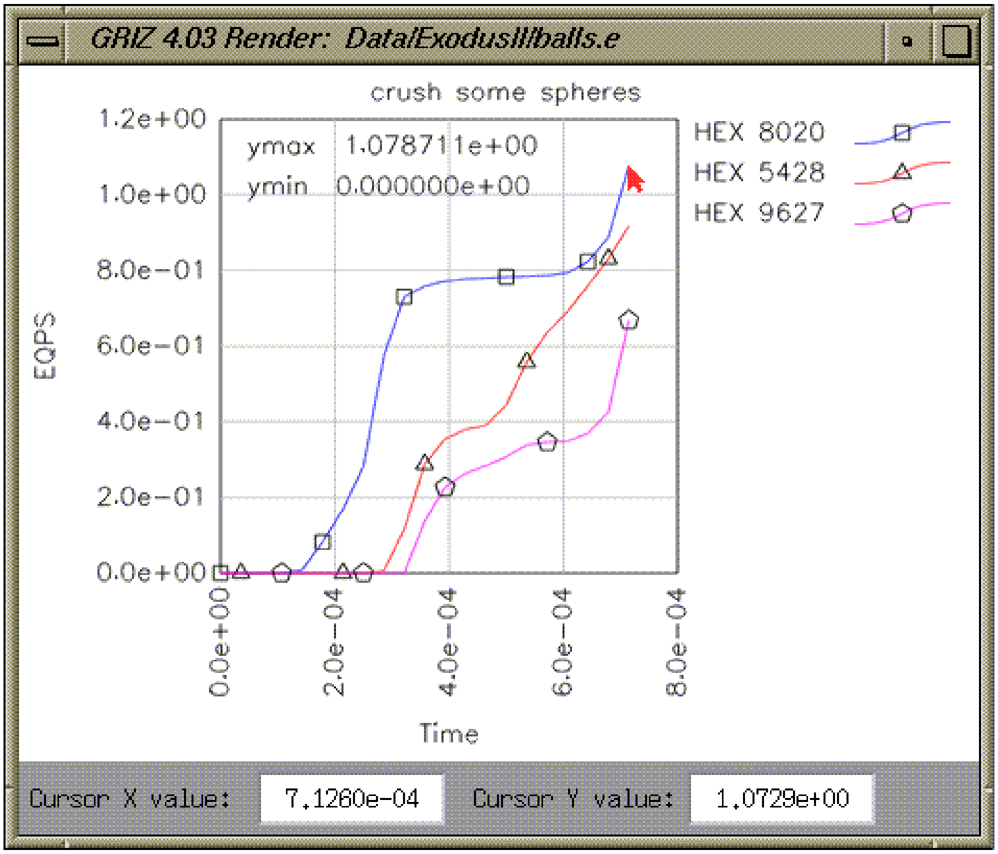
A pick occurs when a mouse button is depressed and released while the cursor is stationary. If the cursor is moved while a mouse button is depressed, GRIZ maps the motion into an interactive transformation of the current mesh view (Note - with some mouse designs, it can be a challenge to hold the cursor completely still during a pick. GRIZ provides the command "minmov," page 66, to set a minimum threshold for cursor motions to be mapped as view transformations). These transformations occur with respect to the screen (image) coordinates, i.e., X-direction parallel to the top and bottom edges of the window, Y-direction parallel to the sides, and Z-direction perpendicular to the screen.

Moving the mouse in the X or Y direction with the left mouse button held down causes rotation about the Y or X axes, respectively. Moving the mouse in the X direction with the middle mouse button held down implements Z rotation; Y movement with the middle button held down implements scaling. Mouse movement with the right mouse button depressed causes translation in the X and Y directions. The mesh display switches to a pseudo-wireframe mode while the view is being modified with the mouse and returns to the normal display when the mouse button is released.

Plot Mode

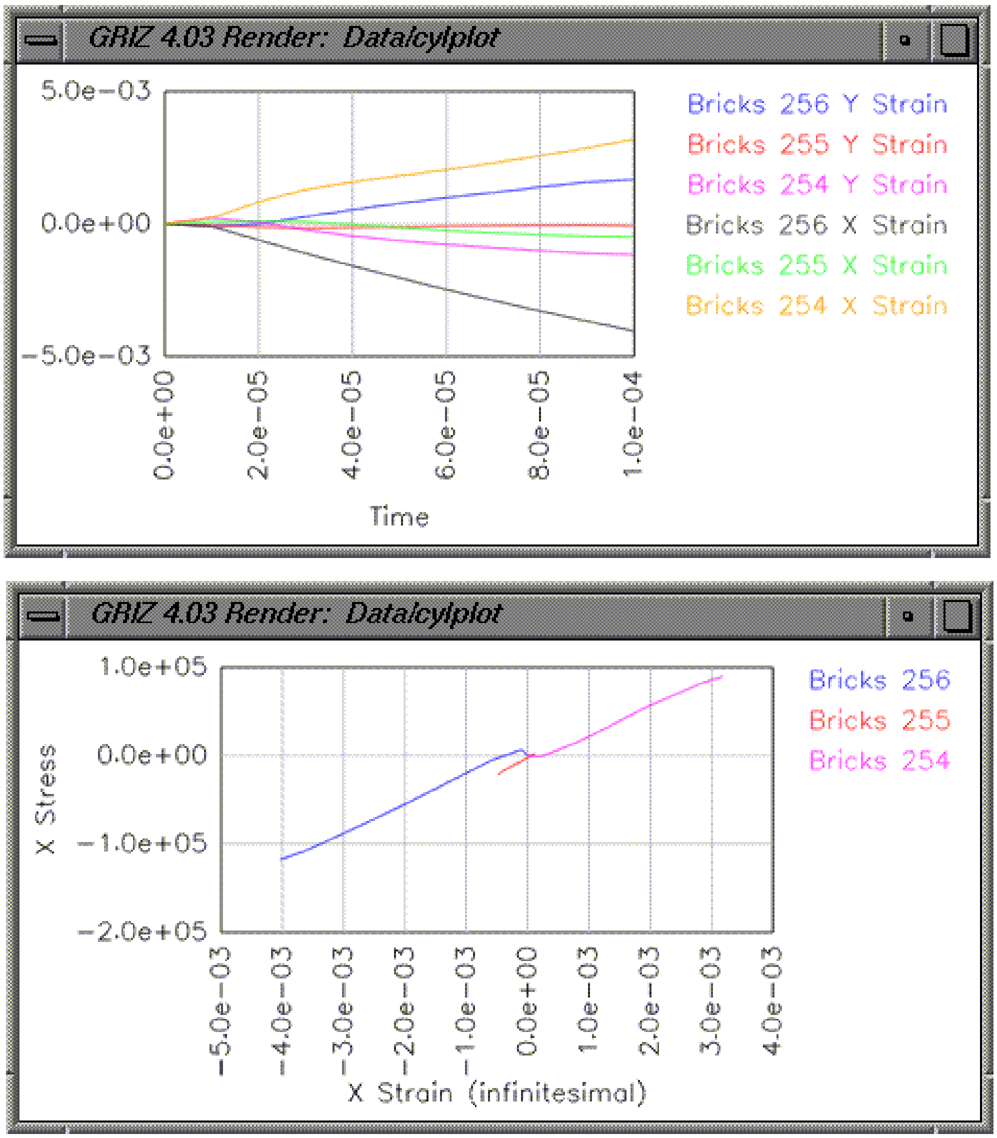
Plot mode is current when the user creates an XY plot (commands “plot” and “oplot”). Figure 3 shows the rendering window in plot mode with data from an Exodus II database. A number of visual elements of the plot mode display can be turned on or off by the user. All aspects of time series plotting are documented in Time History Commands on page 39. Figure 3 shows the default

Figure 3. GRIZ Rendering Window - Plot Mode



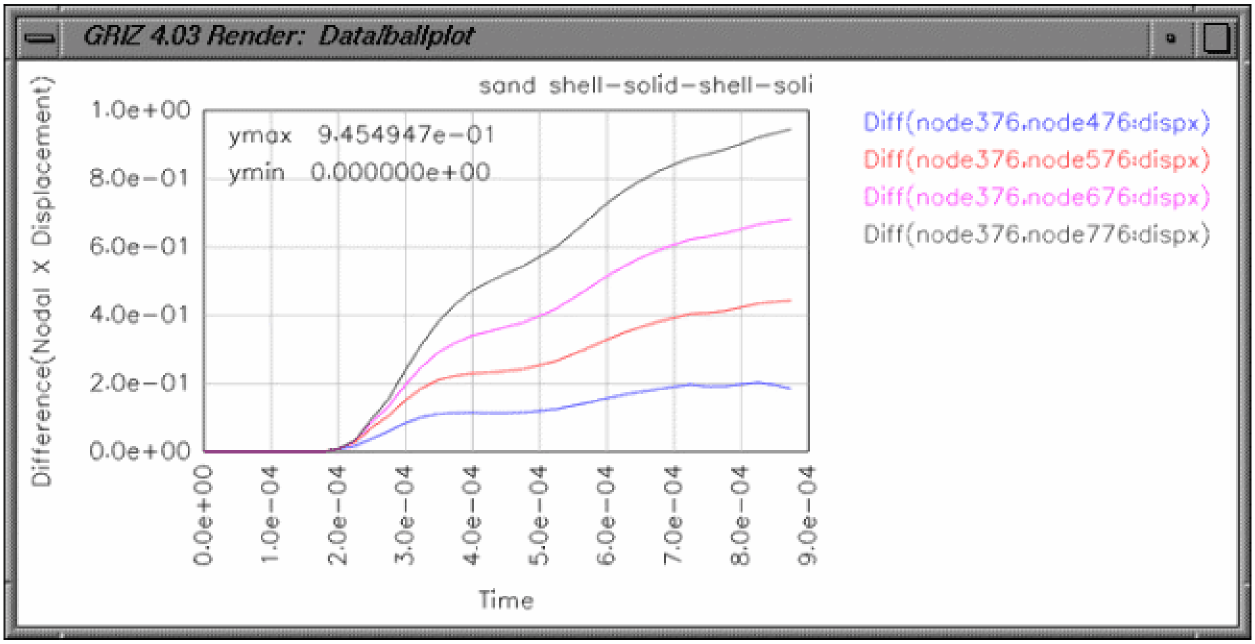
configuration augmented with “on all” (which, for plot mode, turns on the title display and the ymin/ymax display), “on glyphs,” and “on leglines.” Figure 3 also illustrates the cursor coordinates display. This provides a dynamic display of the cursor position, mapped into data units, while the cursor is within the plot box. The cursor Y-value in this example, with the cursor positioned at the maximum Y-data value, correlates well with the “ymax” value from the data rendered in the upper left corner of the plot box. Figure 4 illustrates other GRIZ plotting features. The top image of Figure 4 shows a plot of two results (Y Strain and X Strain) for each of three brick elements. Note that the Y-axis label has been eliminated, since there are multiple results, in favor of augmented legend annotations which provide both object and result labels. The bottom image of Figure 4 illus-

Figure 4. Additional Plot Examples



trates, for the same brick elements, a cross-plot of X Stress versus X Strain with time as an implicit parameter. Figure 5 illustrates GRIZ's "operation" plot capability. Operation plots permit the user to compute new time series by performing an algebraic operation on the samples of two input time series. The available operation resultants are difference, sum, product, and quotient. Operation plot logic is flexible, permitting time series to be computed between different results on objects of different classes (for example, comparing node velocity with material rigid-body velocity) or comparing results on a single object simultaneously with results on several other objects (as in Figure 5, which plots the difference in X displacement between node 376 and nodes 476, 576, 676, and 776).

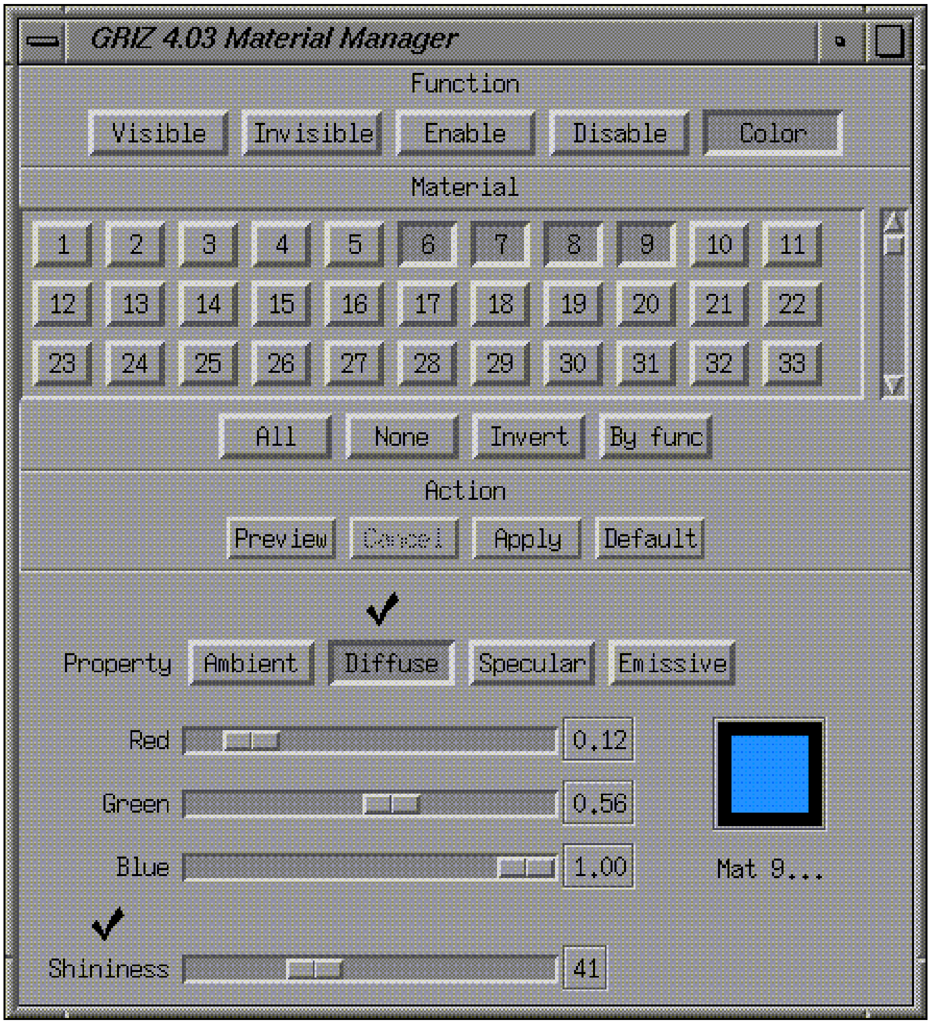
Figure 5. Operation Plot Example



2.3 The Material Manager

The Material Manager (Figure 6) is provided primarily to ease certain operations on meshes with many materials. Its graphical interface to color property editing is, for many users, preferable to the command-line “mat” command regardless of the quantity of materials being operated on. The Material Manager is invoked from the **Control** pulldown menu or by typing <Ctrl>-m while the cursor is in either the control or rendering window. It can be destroyed using whatever window-killing feature is provided by the user’s workstation window manager (typically via a pulldown menu in the upper left corner of the window frame). The collection of controls in the Material Manager is, to some degree, a function of the current mesh undergoing analysis. As such, GRIZ destroys the Material Manager when a "load" command is executed to open a new database.

Figure 6. GRIZ Material Manager



The Material Manager interface is divided vertically into four panels. These panels provide, from top to bottom, function selection, material selection, action (i.e., function execution), and color editing. The material selection panel may contain a scrollbar, depending upon the quantity of materials in the current mesh. The Material Manager window may be resized vertically and/or horizontally to eliminate the need for scrolling.

The toggles and pushbuttons of the Material Manager are managed such that only the controls meaningful given the current state of material and function selections are sensitive (i.e., able to receive input). For example, when first brought up the buttons on the action panel are all insensitive. This is logical since no actions are warranted until both a function is selected and one or more materials are selected.

Function Selection

The function selection panel allows the user to select one or more functions to be applied to selected materials. The functions are **Visible**, **Invisible**, **Enable**, **Disable**, and **Color**. The **Visible** function makes materials visible; the **Invisible** function hides them. The **Enable** function allows result computation on materials; the **Disable** function prevents it. The **Color** function allows modification, for the materials selected, of any of the color properties managed by the color editing panel.

It is possible to select more than one function at a time, subject to some constraints. The **Color** function is exclusive of all the other functions; **Visible** and **Invisible** are mutually exclusive, as are **Enable** and **Disable**. These constraints are incorporated in the logic controlling the function buttons, so functions which are incompatible with a new selection are automatically deselected. In practice, the most useful function combinations are **Invisible** + **Disable** and **Visible** + **Enable**.

Material Selection

The material selection panel contains one toggle for each material in the mesh plus a collection of "group select" buttons at the bottom. A material is selected for operation by placing the cursor over its toggle and depressing the left mouse button. The toggle remains "depressed," indicating selection. Pressing the left mouse button again deselects the toggle, returning it to its original state. Note that selecting a material does not cause the current function(s) to be applied, it merely adds the material to the set of materials that can be operated on.

The group select buttons at the bottom of the material selection panel provide additional convenience by allowing certain selection/deselection operations across the entire set of materials simultaneously. The **All** button selects all materials; **None** deselects all materials. **Invert** toggles the selection state of all materials, that is, materials selected prior to pushing **Invert** are deselected, and vice versa. The **By func** button provides a way to examine the current state of all materials with respect to a given function.

When **By func** is pressed, all materials whose state matches the current function are selected. If multiple functions are selected, then those materials that match all the current functions are selected. For example, if the current function is **Invisible**, pressing **By func** causes only those materials that are currently invisible to be selected. Similarly, if **Invisible** and **Disable** are the selected functions, **By func** causes all materials which are both invisible and disabled to be selected, deselecting all others.

It is recommended that users experiment with the group select buttons, as they can be very useful in particular circumstances. For example, if a user is post-processing a mesh with a large number of materials and wishes to select all but a few of the materials, it is convenient to use **All** to select all materials and then manually deselect the few unwanted materials. For a more complex example, suppose a user wants to reverse the visibility of all materials in a mesh, many of which are currently invisible. The user could first select the **Visible** function, then press **By func**. At this point, only those materials which are visible are selected. To toggle their visibility, the user would select the **Invisible** function (**Visible** is automatically deselected since it is logically exclusive of **Invisible**) and then press **Apply** on the action panel (described below). At this point, all mesh materials are invisible, but by pressing the **Invert** group selection button, the initially invisible materials (which must be made visible) are selected for operation. Finally, the **Visible** function is selected (deselecting the **Invisible** function) and the **Apply** action button is pressed. This completes the visibility toggling. Please note that reading this description actually takes much longer than performing the described sequence of operations.

Action

The Action panel contains four buttons: **Preview**, **Cancel**, **Apply**, and **Default**. The **Apply** button executes the selected function on the selected materials. **Preview**, **Cancel**, and **Default** are sensitive only at particular times when the **Color** function is selected. The **Preview** button temporarily applies a change to a color property of the selected materials, allowing the user to examine the change before permanently applying it. It is not necessary to preview a change before applying it. The **Cancel** button cancels a color preview, returning the color properties for the selected materials to their previous values. **Cancel** is sensitive only immediately following a preview action. The **Apply** button permanently applies a color property change. The **Default** button returns all color properties of the currently selected materials to their initial (startup) values.

Color Editing

The controls available on the color editing panel make it possible to edit all of the material color properties that GRIZ manages. These controls are sensitive only when the **Color** function is selected. A detailed discussion of each color property is beyond the scope of this document but is covered in the OpenGL Programming Guide (Addison-Wesley, ISBN 0-201-46138-2) and should be in any modern textbook on computer graphics. For the purposes of this manual, the ambient and diffuse properties determine the color of a material and should normally be set to (nearly) the same value in a particular material, shininess determines the size of highlights from light reflections, and specularity affects the color of highlights (normally white or grey). The emissive property allows a material to emit light of a given color. In most circumstances this property can be ignored. Materials that emit light are not treated as light sources with respect to other materials.

There are three sets of controls on the color editor. The color property toggles, **Ambient**, **Diffuse**, **Specular**, and **Emissive**, select among the four red/green/blue-valued (rgb) material color properties. The color component sliders, **Red**, **Green**, and **Blue**, allow modification of the rgb values for the chosen color property of the selected materials. Color component values range from 0.0 to 1.0. The values displayed on the sliders are determined by the most recently selected material. When materials are selected using the group select buttons, the highest-numbered material is indicated. The **Shininess** slider allows modification of the shininess property of the selected materials. Shininess values range from 0 to 128. Sliders are adjusted by selecting with the cursor and left mouse button and then moving the mouse laterally. They can also be moved with precision by selecting with the mouse and then using the left or right arrow keys on the keyboard while keeping the mouse button depressed.

There are two visual feedback elements on the color editor. A color swatch which appears to the right of the rgb sliders whenever the **Color** function is selected. This swatch shows the resultant color determined by the rgb slider settings. One or more materials must be selected, and one of the color property toggles (**Ambient**, **Diffuse**, **Specular**, or **Emissive**) must be selected before the sliders and the swatch actually show a material color. The swatch is labelled to indicate which material is providing the color displayed. When multiple materials are selected, an ellipsis appears in the label as a reminder that any color change applied will affect more than just the material number displayed. The second visual feedback element is a set of black check marks that appear over any of the five properties (**Ambient**, **Diffuse**, **Specular**, **Emissive**, or **Shininess**) when an associated slider is adjusted. This dictates a change in the current value if **Apply** or **Preview** is pushed. A check also appears over any rgb-valued component that is selected if more than one material is selected. This logic is based on the assumption that the component will be different in each of the selected materials. If the user presses **Apply**, this will force all of the selected materials to share the rgb-value shown, even if no slider adjustments have been made.

Selecting materials for which color properties will be edited must precede adjustment of any of the properties. Modifications to the set of selected materials are treated as an indication to start over. Therefore, any material selections or deselections will reset the sliders to the (new) most recently selected material, eliminating existing slider settings that have not been applied.

At startup, GRIZ initializes the ambient and diffuse properties of a material to the same color. The specular and emissive properties for all materials are initialized to black (red, green, and blue values of 0.0), and shininess for all materials is zero. As mentioned above, under normal circumstances one would want to keep the ambient and diffuse values the same.

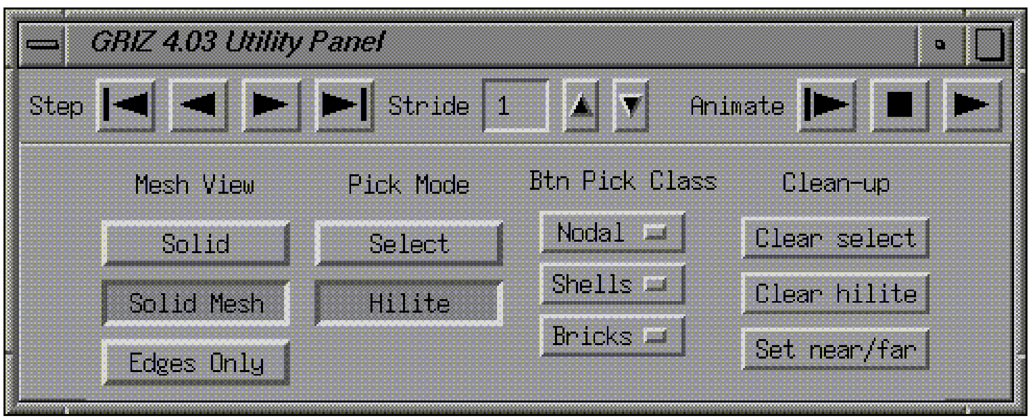
As with all other graphical GRIZ controls, Material Manager actions generate text commands which are recorded into a history file if one is active. These commands are all forms of GRIZ's "mtl" command (created specifically for the Material Manager and documented on page 67) and can alternatively be typed explicitly to the command interpreter. Typed command entry is supported in the Material Manager, as in other GRIZ windows, with the caveat that the color editing panel will only forward commands to the command entry window while the color editing panel is sensitive, i.e., the Color function is selected.

2.4 The Utility Panel

The Utility Panel (Figure 7) provides an alternative interface to selected commands, most of which are scattered throughout the GRIZ pulldown menus. It can be brought up in a standalone window by selecting the **Utility Panel** option in the **Control** pulldown menu or by typing <Ctrl>-u while the cursor is in either the control window or the rendering window. It can also be configured as part of the control window by executing GRIZ with the "-u" option. In the latter case, the Utility Panel is not available from the menus or keyboard.

The Utility Panel is divided into two parts. The top part contains controls for manipulating the current simulation state. The bottom part contains a collection of toggles and pushbuttons that control several different functions. The state controls allow the user to traverse states manually (**Step**) or traverse states automatically (**Animate**). The **Step** controls (top left) consist of four buttons labelled with direction icons that, from left to right, step to the first state, step backward, step forward, and step to the last state. The leftmost and rightmost **Step** buttons are simply graphical interfaces to the “f” (first state) and “l” (last state) commands, respectively. When the **Stride** control is set to 1, the second and third **Step** controls behave identically to the “p” (previous state) and “n” (next state) commands, respectively. The **Animate** controls (top right) consist of three buttons that, from left to right, animate starting at state one, stop animation, and continue animation from the current state. These functions are identical to those provided by the commands “anim”, “stopan”, and “animc.” The **Stride** control sets how many states are traversed with each step forward, each step backward, or each animation step. The stride setting applies only to animations which step over integral states, not animations which interpolate time (a feature available via command-line invocation of the “anim” command, described on page 24). The stride is initially 1 and is adjusted up or down with the associated arrow buttons. The stride is restricted to values on [1, maximum state number]. Note that if the current state number plus the stride exceeds the maximum state number, a step forward will fail, leaving the current state unchanged; similarly, a

Figure 7. GRIZ Utility Panel



backward step cannot step beyond state one.

The lower part of the Utility Panel contains toggles, pushbuttons, and option menus to execute several functions. All but one (**Edges Only**) are available through the pulldown menus as well. These functions are labelled similarly to their pulldown menu counterparts. The **Solid Mesh** toggle, however, renames the **Draw Hidden** option under the **Rendering** pulldown. The three option menus under the **Btn Pick Class** label correspond, from top to bottom, to the **Set Btn 1 Pick**, **Set Btn 2 Pick**, and **Set Btn 3 Pick** submenus under the **Picking** pulldown menu.

The **Edges Only** mesh view is essentially an alias for the command sequence “on edges” and “switch none.” It is a useful mode for large meshes when fine-tuning the mesh orientation to avoid lengthy redraws each time a mouse button is released. The meaning of the Utility Panel functions is explained in Chapter 3.0 COMMANDS, under the associated commands identified in Table 2.

Table 2. Command Equivalents for Utility Panel Functions

Button/Toggle/Option-menu	Command Equivalent
Solid	switch solid
Solid Mesh	switch hidden
Edges Only	on edges, switch none
Select	switch picsel
Hilite	switch pichil
Btn Pick Class top menu	setpick 1 <class name>
Btn Pick Class middle menu	setpick 2 <class name>
Btn Pick Class bottom menu	setpick 3 <class name>
Clear select	clrsel
Clear hilite	clrhil
Set near/far	rnf

2.5 Batch Mode

When appropriately compiled (see Table 1, “-b” switch), GRIZ permits a batch mode of operation with offscreen rendering. In this mode, GRIZ creates no onscreen windows. Batch mode is invoked when GRIZ is invoked with the “-b <*filename*>” execution arguments. The named file must be a GRIZ command file (i.e., text, with one GRIZ command per line). Textual feedback that in interactive mode appears in the feedback window or a dialog box is instead redirected to standard out or standard error.

The batch mode capability is provided by the Mesa OpenGL library. Since this library performs all rendering in software, performance in batch mode will typically be lower than in interactive mode on a workstation with hardware graphics acceleration. On the other hand, since Mesa uses computer memory and not the graphics hardware framebuffer as the destination for rendering operations, GRIZ in batch mode can render into windows whose size is ultimately limited only by the amount of computer memory available. This capability allows rendering at very high resolutions. The largest window available is set when Mesa is compiled. Some internal Mesa operations scale with the maximum window dimensions regardless of *actual* window dimensions, so there are practical reasons to set reasonable limits. The GRIZ execution arguments “-w <*width*> <*height*>” set the rendering window size. GRIZ will terminate with an error if a window size is requested that exceeds the limits set when Mesa was compiled.

3.0 COMMANDS

GRIZ commands are entered in the command entry area at the bottom of the control window. A command may have zero or more arguments. When a string that includes spaces is to be provided as a single argument, the string must be enclosed in quotes (except expressions for multi-dimensional primal results. See "show" command, page 27). Comment lines can be included in command sequences and are demarcated with the "#" character. Anything from a "#" to the end of the line is ignored. Comments are primarily of use in history files. Two special types of commands, "on/off" and "switch," appear throughout this section. The "on" and "off" commands are used to set and unset logical flags. For example, "on box" turns on display of the mesh bounding box; "off box" turns off this display option. Similarly, the "switch" command is used to set multiple-valued flags. For example, "switch hidden" switches to a hidden-line display of the mesh. The "switch" command can be abbreviated "sw". The command names are designed to be reasonably easy to remember and are short enough to be typed quickly. A command alias capability is provided for users who wish to specify an alternative command name for any command.

The following conventions apply in the command descriptions:

- Literal command text is **boldface**.
- Greek letters and italicized text are user-specified arguments.
- Optional arguments are enclosed in brackets ("[]").
- Alternative arguments are enclosed in braces ("{}") and separated with a vertical bar ("|").
- Ellipsis ... indicate indefinite repeat of the preceding argument type.

3.1 View Commands

View control commands allow the user to rotate, translate and scale the mesh view. Note that some of these commands operate in screen space and some operate in mesh space.

Rotation

The rotation commands operate in screen space.

rx ϕ	Rotate about the screen X axis by angle ϕ degrees. The screen X axis is parallel to the horizontal axis of the screen.

ry ϕ — Rotate about the screen Y axis by angle ϕ degrees. The screen Y axis is parallel to the vertical axis of the screen.

rz ϕ — Rotate about the screen Z axis by angle ϕ degrees. The positive screen Z axis comes out of the screen toward the viewer and is perpendicular to the screen.

Translation

The translation commands operate in screen space. Upon startup, the narrowest rendering window dimension has an extent of two units. In the default square rendering window, for example, an X-translation of 1.0 will move the mesh center to the right side of the window.

tx *v* — Translate along the screen X axis by value *v*.

ty *v* — Translate along the screen Y axis by value *v*.

tz *v* — Translate along the screen Z axis by value *v*.

Scaling

scale *v* — Set the mesh scale value to *v*. Unlike the rotate and translate commands, which are incremental, the scale command sets an absolute scale value.

scalax v_x v_y v_z — Set mesh scale values for each mesh axis individually.

Resetting

rview — Reset the view to the default view with no rotation, no translation and a scale of 1. When the program starts up, it uses an initial rotation about the X and Y axis in the view parameters. The view parameters can be reset to the defaults at startup or any other time with this command.

rnf — Reset the near and far planes. In a perspective view, the near and far planes clip parts of the scene that are in front of the near plane or behind the far plane. The near and far planes should be set just outside the mesh limits to obtain a good image. This accommodates the Z-buffering algorithm that the hardware uses. This command automatically resets the near and far planes to obtain a good image of the mesh. It will be necessary to execute this command if, after translating the mesh along the Z axis or scaling the mesh, parts of the mesh at its nearest or farthest extent (or both) are clipped.

View Centering

The following commands cause the view to be centered on a particular location. Note that turning on view centering overrides or modifies the effect of mesh rotation and translation commands.

vcent hi Center the view on the currently hilited node or element.

vcent n *node_number* Center the view on the specified node. The effect is the same as “vcent hi” with a node hilited, but without the hilite indicator drawn in the image.

vcent *x y z* Center the view on the specified mesh-space point. The point does not need to be contained within the mesh.

vcent off Turn off view centering.

3.2 State and Time Commands

minst *n* Set the minimum accessible data state to state *n*. States preceding state *n* will be ignored, but the states are not re-numbered.

maxst *n* Set the maximum accessible data state to state *n*. States after state *n* will be ignored.

stride *n* Set the state stride (used in the Utility Panel state step controls and in animations) to *n*. The value *n* must be a non-zero positive integer less than the quantity of states in the database. The stride has no effect on those animations which interpolate time to generate each frame.

The following commands are constrained by the current settings of “minst” and “maxst”.

state *n* Move to state number *n*. The total number of states in the data can be obtained with the “info” command, and the “lts” command lists all state times.

n Move to the next state.

p Move to the previous state (current state minus one).

f Move to the first state.

l Move to the last state.

time *t* Move to time *t*. If the specified time does not coincide with any of the states, the program interpolates the nodal positions for the specified time. If a result is displayed, the result will be interpolated as well. However, after moving to the specified time, changing the displayed result will not work and the new result will not be interpolated. Issuing the “time” command again will interpolate the new result.

lts List times for states (alias for “tell times”).

Animation

anim Animate the data by displaying it over time. This form of the “anim” command displays each state in the database in sequence.

anim *n* Animate the data by displaying it over time. This form of the “anim” command displays *n* frames between the start time and end time of the data. Interpolation is performed where necessary.

anim *n* t_s t_e Animate the data by displaying it over time. This form of the “anim” command displays *n* frames between the specified start time (t_s) and end time (t_e). Interpolation is performed where necessary.

stopan Stop an animation after the current frame is rendered.

animc Continue an animation from the current state instead of starting at the first state.

{**on** | **off**} **autoimg** Turn on (off) automatic output of rendered images to a sequence of raster image files during any animation command. The output files will share a common root name previously specified with the “autoimg” command and will have a suffix dictated by the image format utilized (see “autoimg” command below). For example, if a root file name were “bridge” and JPEG output were specified (i.e., “autoimg bridge jpeg”), the sequence of file names with “on autoimg” in effect would be “bridge0001.jpg”, “bridge0002.jpg”, “bridge0003.jpg”, etc. Note that neither turning “autoimg” on or off will reset the frame counter. Only the “autoimg” and “resetimg” commands reset the frame counter.

autoimg *rootfilename* [**jpeg** | **png**]
The “autoimg” command provides three functions: it lets the user specify the root file name for image files generated during an animation (see “on autoimg”, above), it lets the user select either RGB output format (default), JPEG output format (if the “jpeg” argument is supplied), or PNG output format (if the “png” argument is supplied), and it resets the image file frame counter used to generate the filenames. Note that availability of JPEG and PNG formats is set when GRIZ is compiled.

resetimg Reset the “autoimg” frame counter, used to generate image file names, to one.

3.3 Executive Commands

load *name* — Load in a new database. The argument is the plotfile root name for the new database.

{**quit** | **exit** | **end**} — Quit GRIZ.

on [**all**] [**box**] [**carpet**]...

Turn on the specified logical flags. Flags are described in pertinent sections of this chapter.

off [**all**] [**box**] [**carpet**]...

Turn off the specified logical flags. Flags are described in pertinent sections of this chapter.

switch {{**middle** | **inner** | **outer**} | {**hidden** | **solid** | **cloud** | **none**} | ... }

Set a multiple-valued flag. Flag groups are described in pertinent sections of this chapter. Command "switch" may be abbreviated "sw".

info — Print out information about the current state of the program. This command is documented in greater detail in Query Commands on page 46.

r — Repeat the last command. This command only works once after executing the "rdhis" command.

{**on** | **off**} **refresh** — Turn on (off) the display update while a series of commands is executed. This command can be used when the mesh image takes a long time to redraw and one wants to execute several commands before having the screen update.

alias *newcom* **"***comstring***"**

Create an alias for *comstring*. Whenever *newcom* appears as a separate word in a command, it is replaced by *comstring*. The quotes aren't needed if *comstring* consists of only one word.

exec "*command line***"**

Execute a UNIX command in the shell.

3.4 Results Commands

The results display commands allow the user to select the result variable to be displayed and to control various parameters associated with the results display. Results are color-mapped with regions of the highest result value drawn in the color at the top of the colormap and regions of the lowest result value drawn in the color at the bottom of the colormap.

GRIZ provides two sets of results - “primal” and “derived.” Primal results are the “raw” results stored in the database. GRIZ can render several categories of primal results that might occur in a database: nodal results, element results (truss, beam, triangle, quadrilateral, tetrahedral, and hexahedral, in two- or three-dimensional meshes as required), material results (all elements of a material effectively share the same result value), and global mesh results (all elements of the mesh share the same result value). Other categories of results that GRIZ does not provide explicit rendering support for can still be accessed and plotted as time histories for the objects populating the categories. Primal results need not be scalars. For example, the Mili I/O library, supported by GRIZ, allows for vector and array variables. GRIZ supports these primal results by making their individual components available as scalars. In menus, GRIZ adds another level of submenu for these variables in which the individual components are selectable. GRIZ puts array primal variables of up to two dimensions and up to 20 cells per dimension into menus. GRIZ implements a specific “show” command syntax for specifying a component of an array or vector variable from the GRIZ command line (below).

Derived results are calculated by GRIZ from primal results. Derived results may represent a transformation of a primal result into an alternative form, e.g., a global-to-local transformation of shell element strain components, or may be new quantities entirely. GRIZ makes few assumptions about the simulation state variables available in databases. It evaluates the contents of a database at startup to determine which derived result calculations are supported by the contents of the database. The GRIZ **Derived** menu adapts itself to the outcome of this determination, so its contents may vary with different databases.

There is some redundancy between the Primal and Derived menus. Previous versions of GRIZ made no distinction between derived and primal results and, for consistency’s sake, the current logic has been designed to provide a **Derived** menu that is a reasonable approximation of the **Result** menu in earlier versions of GRIZ. The code was previously structured such that all results passed through the derivation framework, even if in the end the quantities displayed were actually un-modified primal results. In this version of GRIZ, the distinction between derived and primal results must be made explicit because the new material- and simulation-specific state variables available from self-defining databases will not, in general, be the designated inputs to GRIZ’s existing derived result calculations. For these new variables, the **Primal** menu provides the natural menu reference.

It is important that the user keep in mind how the minimum and maximum result values of the colormap are set. A switch is provided so that either the state min/max or the global min/max can be used. The default is to use the global min/max. Because it is too time-consuming to calculate the min/max for all states at once, the global min/max is obtained incrementally. As each state is displayed, GRIZ calculates the min/max of the currently displayed result for that state and uses those values to update the global min/max. The global min/max for a specific result is not correct until all states have been visited with that particular result displayed. One can force an immediate traversal of all states to acquire the true global min/max by selecting the desired result and then entering the “globmm” command. Once the global min/max for a result variable has been obtained, it will be “remembered” when that result is shown unless explicitly destroyed with the “resetmm” command (page 35).

Result Display

show *result*

Display a result variable in a pseudocolor plot. Potentially available derived result values for *result* are listed in Table 3. Primal results are dependent upon the database being processed. GRIZ searches for *result* first among available derived results, then among available primal results. If a derived and primal result both have the same name, the derived result will be displayed. If a primal result is non-scalar, the following command-line syntax for *result* enables specification of a vector, array, or vector-array variable component to the “show” command:

To specify *result* as a component of a vector variable, use this syntax:
result_name[*component_name*]
Example - **show stress[sx]**

To specify *result* as a component of an array variable, use this syntax:
result_name[$index_1$ $index_2$...$index_n$]
where $index_1$, $index_2$,...,$index_n$ are in row-major order (i.e., in an in-order traversal of the array’s elements in memory, $index_1$ increments slowest and $index_n$ increments fastest). Indices may be blank- or comma-delimited, and all dimensions must be represented. Index values start with 1, not 0.
Example - **show gausstemp[1 2 2]**
where **gausstemp** is a 2x2x2 array variable.

To specify *result* as a component of a vector-array variable, use this syntax:
result_name[$index_1$ $index_2$...$index_n$ *component_name*]
where $index_1$, $index_2$,...,$index_n$ are in row-major order (i.e., in an in-order traversal of the array’s elements in memory, $index_1$ increments slowest and $index_n$ increments fastest). Indices may be blank- or comma-delimited, and all dimensions must be represented. Index values start with 1, not 0.
Example - **show composite1[6 sx]**
where **composite1** is a vector-array variable consisting of a one-di-

mensional array of vectors; the array size is at least 6, and one of the components of the vector is named **sx**.

Table 3. Derived Result Variables

Result	Description
mat	No result is displayed. The mesh materials are drawn in different colors. This is the default.
Results available for both Hexahedral and Shell Elements	
sx	X Stress
sy	Y Stress
sz	Z Stress
sxy	XY Stress
syz	YZ Stress
szx	ZX Stress
seff	Effective Stress
ex	X Strain
ey	Y Strain
ez	Z Strain
exy	XY Tensor Strain
eyz	YZ Tensor Strain
ezx	ZX Tensor Strain
eeff	Effective Strain
press	Pressure
Hexahedral Results	
pdev1	Principal Deviatoric Stress 1
pdev2	Principal Deviatoric Stress 2
pdev3	Principal Deviatoric Stress 3
maxshr	Maximum Shear Stress
prin1	Principal Stress 1
prin2	Principal Stress 2

Table 3. Derived Result Variables

Result	Description
damage	This command will hilite elements that meet a an evaluation metric that is a function of four response qualtities. Elements that have fully failed will display in RED while non-damaged elements willbe displayedin BLUE. Damage results requires the input of 4 variables as follows: `show damage vel_dir <vx,vy,vz> vel_cutoff relVol_cutoff eps_cutoff`
prin3	Principal Stress 3
pdstrn1	Principal Deviatoric Strain 1
pdstrn2	Principal Deviatoric Strain 2
pdstrn3	Principal Deviatoric Strain 3
pshrstr	Maximum Tensor Shear Strain
pstrn1	Principal Strain 1
pstrn2	Principal Strain 2
pstrn3	Principal Strain 3
relvol	Relative Volume
Shell Results	
res1	M_{xx} Bending Resultant
res2	M_{yy} Bending Resultant
res3	M_{xy} Bending Resultant
res4	Q_{xx} Shear Resultant
res5	Q_{yy} Shear Resultant
res6	N_{xx} Normal Resultant
res7	N_{yy} Normal Resultant
res8	N_{xy} Normal Resultant
thick	Shell Thickness
inteng	Internal Energy
surf1	Surface Stress 1

Table 3. Derived Result Variables

Result	Description
surf2	Surface Stress 2
surf3	Surface Stress 3
surf4	Surface Stress 4
surf5	Surface Stress 5
surf6	Surface Stress 6
eff1	Effective Upper Surface Stress
eff2	Effective Lower Surface Stress
effmax	Maximum Effective Surface Stress
Beam Results	
axfor	Axial Force
sshear	S Shear Resultant
tshear	T Shear Resultant
smom	S Moment
tmom	T Moment
tor	Torsional Resultant
saep	S Axial Strain (+)
saem	S Axial Strain (-)
taep	T Axial Strain (+)
taem	T Axial Strain (-)
Nodal Results	
dispx	X Displacement
dispy	Y Displacement
dispz	Z Displacement
dispmag	Displacement Magnitude
velx	X Velocity
vely	Y Velocity
velz	Z Velocity

Table 3. Derived Result Variables

Result	Description
velmag	Velocity Magnitude
accx	X Acceleration
accy	Y Acceleration
accz	Z Acceleration
accmag	Acceleration Magnitude
temp	Temperature
ens	Enstrophy (Hydra databases only)
hel	Helicity (Hydra databases only)
k	Turbulence Model k Component (Hydra databases only)
eps	Turbulence Model ε Component (Hydra databases only)
a2	Turbulence Model A2 Component (Hydra databases only)
pint	Pressure Intensity (Ping databases only)
pvmag	Projected Vector Magnitude - the dot product of the current vector quantity as defined by the “vec” command with the current direction vector as defined by either “dir3n”, “dir3p”, or “dirv”.

Result Color Interpolation Modes

These commands set how result color values are rendered on the polygons making up the mesh view.

switch interp Regular color interpolation from the polygon vertices. (default)

switch gterp Good, but "expensive," shading accomplished by dicing polygons into pixel-sized fragments.

switch noterp No color interpolation. Colors corresponding to values at element centers are drawn over the full polygons.

Colormap Thresholding

rzero *v* Set a result zero tolerance. Result values in the range (-*v*, *v*) are treated as zero values and are displayed in the default material color. Use "clrthr" to clear this value.

rmin *v* Set a minimum result value. Result values below or equal to this value are colormapped with the color that is in the first entry of the colormap. Command "clrthr" clears this value.

rmax *v* Set a maximum result value. Result values above or equal to this value are colormapped with the color that is in the last entry of the colormap. Command "clrthr" clears this value.

clrthr Clear the "rzero", "rmin" and "rmax" threshold values.

Strain Type

switch infin Set the strain type to Infinitesimal. (default)

switch alman Set the strain type to Almansi.

switch grn Set the strain type to Green.

switch rate Set the strain type to strain rate.

Shell Reference Frame

switch rglob Set the element (shell and hex) reference frame to global. (default)

switch rloc Set the element (shell and hex) reference frame to local.

Shell Result Surface

switch middle Calculate shell results at the middle surface. (default)

switch inner Calculate shell results at the inner surface.

switch outer Calculate shell results at the outer surface.

Nodal Displacement Reference State

refstate [*n*] Set the reference state for nodal displacement calculations (i.e., displacements are calculated as the difference between the current position and the position at the reference state). Argument *n* is an integer on [0, qty states]. When *n* is zero, the initial geometry from the mesh description is used as the reference state. Otherwise, *n* identifies a state to use as the reference state. If *n* is not supplied, a dialog is popped up giving the current reference state setting. The reference state setting impacts any explicit use of nodal displacements, including displacement results (dispx, dispy, dispz, dispmag) and displacement scaling. It does not affect strain calculations. Note that, with respect to the displacement results, each possible reference state setting implies a different result domain, so cached result min and max values are not shared across multiple reference state settings for the same result. (default: initial geometry)

Result Table Control

Griz maintains two tables of results: derived and primal. Primal results are links to “raw” data straight out of the database. Derived results are computed from primal results. The derived result framework, in addition to permitting the calculation of new quantities, also provides an avenue for tranformations on primal results which don’t change a result’s identity but may change the basis upon which it’s interpreted (for example, a global-to-local stress transformation). When Griz undertakes any activity that demands a result, it searches first in the derived result table for the named result and then in the primal result table if the derived result search fails. Depending on the data available in a database, it could occur that there are both derived and primal versions of a result with the same name. The commands described below allow the user to selectively turn off either table to ensure that results are only found in a particular table. By default, both tables are on.

{**on** | **off**} **derived** Turn on (off) searches of the derived result table on subsequent result requests. If only the derived table is active when an “off derived” command is parsed, the primal result table is automatically made active.

{**on** | **off**} **primal** Turn on (off) searches of the primal result table on subsequent result requests. If only the primal table is active when an “off primal” command is parsed, the derived result table is automatically made active.

Alternative Global Coordinate System

coordxf {$node_1$ $node_2$ $node_3$ | x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3}
Specify a coordinate system for stress/strain result transformations. The specification requires the definition of three points in the mesh global coordinate system. The two forms of the command arguments allow the user to specify the required points as the current coordinates of three mesh nodes (first form) or as three points explicitly defined (second form). The transformation coordinate system is derived from the three points as follows: The ray from point one through point two defines the positive X-axis; the ray perpendicular to the X-axis and containing point three defines the positive Y-axis; the positive Z-axis is defined by crossing the X-axis into the Y-axis (right-hand rule).

{**on** | **off**} **coordxf** Turn on (off) transformations of stress or strain results into an alternative global coordinate system. The derived results affected are sx, sy, sz, sxy, syz, szx, epsx, epsy, epsz, epsxy, epsyz, and epszx. The alternative global coordinate system is specified with the “coordxf” command. When both “on coord” and “on coordxf” are in effect, both the original and alternative coordinate triples are rendered with a common origin. The alternative coordinate system is rendered in brown, slightly smaller, and with lower-case “x”, “y”, and “z” axis labels to differentiate it from the original coordinate triple.

Global Direction Vector

These commands define a global direction vector, currently used by nodal derived result “pvmag”.

dirv *x y z* Specify a direction vector in the global mesh coordinate system.

dir3n $node_1$ $node_2$ $node_3$
Specify a direction vector in the global mesh coordinate system as the normal to the plane defined by the positions of three mesh nodes at the current state (right-hand rule determines the positive direction).

dir3p x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3
Specify a direction vector in the global mesh coordinate system as the normal to a plane defined by the three specified points (right-hand rule applies).

Result Min/Max

switch mglob Use the global min/max for the displayed result. (default)

switch mstat Use the state min/max for the displayed result.

extreme_min
extreme_max Find the extreme min or max cur current result across all states.

globmm Find the global result min/max for the currently displayed result.

resetmm Reset the global minimum and maximum for the current result to the current state minimum and maximum and delete any cached min/max generated from calculating the result at other states. This has the effect of nullifying a previous “globmm” command with the current result active and making it seem as if only the current state had been visited for the current result. The intent of this command is to prevent distortions of cached min/max values for results which can undergo transformations that might generate different mins/maxs at already-visited states. Specifically, a stress or strain that is transformed to a different coordinate system (“on coordxf”), or a “pvmag” result in which the driving vector definition is changed will create different value domains from which mins/maxs are taken, but the min/max caching facility does not take this into account. By issuing a “resetmm” command immediately after applying a new coordinate transformation or new vector definition, and again when a transformation is removed, one can ensure that cached mins/maxs originate from the same result value domain as the currently viewed result.

Result Units Conversion

{**on** | **off**} **conv** Turn on (off) units conversion. Conversion parameters must be specified with the “conv” command. Conversion parameters are displayed in the rendering window when conversion is on. Note that when units conversion is turned on, result values specified with the commands “rmin”, “rmax”, “rzero”, and “isov” are assumed to be in the converted units. If conversion is subsequently turned off, the stored values are effectively converted to the original data units.

conv [*scale* [*offset*]] Specify parameters for a linear data conversion of the form:

$$y_{new} = y_{old} * scale + offset$$

where “scale” defaults to 1.0 and “offset” defaults to 0. Note that you cannot specify “offset” without also specifying “scale”.

clrconv Reset conversion parameters to their default values and turn off conversion.

Miscellaneous Result Derivation Parameters

pref *v* Set the reference pressure for pressure intensity result calculations.

dia *v* Set the beam diameter for beam strain result calculations. (default: 1.0)

ym *v* — Set Young's Modulus for beam strain result calculations. (default: 1.0)

3.5 Optional Rendered Information Commands

Except as noted, these items are off by default.

{**on** | **off**} **all** — Turn on (off) "coord", "time", "title", "cmap", "minmax", and "dscal" simultaneously.

{**on** | **off**} **bgimage** — Turn on (off) display of the currently loaded rgb image (see "inrgb") in the background of the rendering window.

{**on** | **off**} **box** — Turn on (off) display of the bounding box of the mesh.

{**on** | **off**} **cmap** — Turn on (off) display of the colormap. Associated with colormap display are indicators (when appropriate for the displayed result) of strain type, strain basis, and shell reference surface.

{**on** | **off**} **coord** — Turn on (off) display of the mesh coordinate system.

{**on** | **off**} **cscale** — Turn on (off) display of title and result scale on the colormap. This is "on" by default, but is always off if "off cmap" is in effect.

{**on** | **off**} **dscal** — Turn on (off) display of the per axis nodal displacement scale factors.

{**on** | **off**} **edges** — Turn on (off) display of the mesh edges. An edge is defined to be any edge of an external element face that is not shared by another element or any edge that is shared by another element face in which the relative rotation between the adjoining faces exceeds a threshold (see command "crease"). The mesh edges are not automatically recomputed when changing states. Mesh edges are recomputed with the "getedg" command.

{**on** | **off**} **locref** — Turn on (off) display of the local reference frame on shell and hex elements that have been selected (see "select"). When "on locref" is in effect, the usual object label rendered for selected elements is turned off, i.e., the user can show the local reference frame or the selection label but not both.

{**on** | **off**} **minmax** — Turn on (off) display of min and max values. When in mesh view mode, the min and max are for the current result and the information includes the mesh object (node, element, material, etc.) where each extreme occurs. The domain from which the extremes are drawn is either all states visited or just the current state, as dictated by the "switch mglob/mstat" setting. When in plot view mode, the min and max are drawn from the ordinate domains of all curves plotted. In plot view mode, the min and max are rendered initially inside the upper left corner of the plotting window, but can be moved to any other corner via the “mmloc” command.

{**on** | **off**} **num** Turn on (off) display of node and element numbers for object classes previously specified with the “numclass” command.

numclass *class_short_name...*
Specify object classes (node or element) which will be numbered if “on num” is in effect.

{**on** | **off**} **time** Turn on (off) display of the current simulation time at the bottom of the mesh view window.

{**on** | **off**} **title** Turn on (off) display of the mesh data title (which comes from the plotfile or from the “title” command) at the bottom of the mesh view window.

title “*Title String***”** Change the data title. The data title is originally read from the plotfile, but may be changed with this command.

3.6 Mesh Control and Rendering Commands

Mesh Object Picking

hilite *class_short_name n*
Toggle a hilite on the specified node, element, or particle. If the object is currently hilited, the hilite will be removed. If not, the object will be hilited. For example, “hilite shell 538” turns on hilite of object number 538 of a class called “shell” (given that it is not already hilited) by drawing it with a colored border and putting a label near it. Only one object may be hilited at any time. Hiliting can be performed interactively by using the mouse to pick individual nodes or elements (See **Mouse Picking Commands**.) If a result variable is displayed on the object, the current value of the variable is appended to the label. GRIZ’s view centering function (command “vcent”, page 23) can be used to center the mesh view on the currently hilited object.

clrhil Clear the object hilite.

select *class_short_name n...*
Toggle selection of the specified nodes, elements or particles. For example, “select brick 245 28 192” would label and render wireframes on the three referenced elements of a class called “brick” (assuming they are not already selected). The specified object numbers are added to (or removed from) those currently selected. Selected objects are used by default when plotting, gathering, or writing out time histories. Selecting can be performed interactively by using the mouse to pick individual nodes, elements, or particles (See **Mouse Picking Commands**.). A range of objects can also be specified

such as “select brick 1-10”. If a result variable is displayed on an element or node, the value of the variable at the current state is included in the element label.

clrsel [*class_short_name n*]
Deselect (clear) the entire selected object list or just de-select the specified object.

Material Control

exclude {**all** | *n*...| *n-m*} Turn off the results evaluation on the specified materials and make them invisible. This combines the functionality of the commands “disable” and “invis” (below). The material specification can be a list of material numbers or the keyword “all”, representing all materials. A user can also specify a range of material numbers (material range) *n-m*, for example 2-5 or 1-10.

include{**all** | *n*...| *n-m*} Turn on the results evaluation on the specified materials and make them visible. This combines the functionality of the commands “enable” and “vis” (below). The material specification can be a list of material numbers or the keyword “all”, representing all materials.

disable {**all** | *n*...| *n-m*} Turn off the results evaluation on the specified materials. The material specification can be a list of material numbers or the keyword “all”, representing all materials.

enable {**all** | *n*...| *n-m*} Turn on the results evaluation on the specified materials. The material specification can be a list of material numbers or the keyword “all”, representing all materials.

invis {**all** | *n*...| *n-m*} Make the specified materials invisible. The material specification can be a list of material numbers or the keyword “all”, representing all materials.

vis {**all** | *n*...| *n-m*} Make the specified materials visible. The material specification can be a list of material numbers or the keyword “all”, representing all materials. All materials are visible by default.

Mesh Rendering Style

switch hidden Set the rendering style to filled polygons with their edges drawn (hidden-line drawing). (default)

switch solid Set the rendering style to filled polygons with their edges not drawn.

switch cloud Set the rendering style to render only nodes.

switch none Set the rendering style to render nothing. This can be useful in conjunction with "on edges" to obtain a simple representation of the mesh.

Nodal Displacement Scaling

dscal *v* Scale the nodal displacements by the specified scale value. The scaling is applied in all three axial directions. The command "dscal 1.0" will turn off displacement scaling.

dscalx *v* Scale the nodal displacements in X by the specified scale value.

dscaly *v* Scale the nodal displacements in Y by the specified scale value.

dscalz *v* Scale the nodal displacements in Z by the specified scale value.

Free Node Rendering

{on | off} fn Display all nodes that have seperated from their elements, or are free nodes.

on fn [scale 9.9] [res 5] [mass_scale on|off] [vol_scale on|off] [mat 1,2...]

where:

scale - Scaling factor to control nodes size.

res - Resolution to render nodes as spheres - higher number will make better looking nodes, but increase rendering time.

mass_scale - Nodes will be scaled by mass of contributing elements. Default is on.

vol_scale - Nodes will be scaled by volume of contributing elements. Default is off.

mat - Free nodes will be displayed for a selected list of materials.

3.7 Time History Commands

Time history commands allow users to plot and manage time history curves of results versus time or other results. Any time series gathered into memory remains available for subsequent use until explicitly deleted (command "delth"). Both data gathers and rendered plots are constrained by the current settings of "minst" and "maxst." If a time series identified for plotting or output was originally gathered under "minst" or "maxst" settings different from their current settings, then any missing states are automatically visited to fill in the missing data. Simple algebraic operations can be performed on the samples of two time series' to yield a new "operation" time series using the "oplot" command.

plot [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]

Draw one or more time history curves in the rendering window. Zero or more results may be specified. If no results are specified, the current result being displayed on the mesh is used. Zero or more objects over multiple mesh object classes may be specified from which the result time series' will be gathered. The object identifiers (*n*) may be individual numbers or hyphenated ranges of numbers. If no objects are specified, the currently selected objects ("select" command) are used. GRIZ generates the cross-product pairing of the results and mesh objects and draws curves for all combinations in which the result is available for the associated object. If curves for only one result are generated, then that result name labels the plot Y-axis. If curves for multiple results are generated, then the Y-axis label is eliminated, and result names are appended to the object labels in the legend. By default, time is the abscissa for each curve. If "**vs** *abscissa_result*" is specified, then that result is also gathered for all the objects, and curves of *result* versus *abscissa_result* are drawn for each object.

Examples:

plot

This command evaluates the currently shown result and the currently selected objects and generate curves for all valid combinations. If there were no valid combinations or if either default were empty a dialog would notify the user and the contents of the rendering window would remain unchanged.

plot sx sy brick 22-26 shell 3 8

Given that the results exist for all the objects, this command generates 14 curves - X stress for "brick" class elements 22, 23, 24, 25, and 26, X stress for "shell" class elements 3 and 8, and Y stress for the same five "brick" elements and two "shell" elements.

plot dispy node 10 15 20 vs dispx

This command generates three curves - result dispy (ordinate) versus result dispx (abscissa) for "node" class objects 10, 15, and 20.

oplot *operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]

Draw one or more time series curves that result from applying an algebraic operation between respective samples of two operand time series'. The operation specified may be the difference, sum, product, or quotient as follows. Both symbolic and textual operation specifiers are available:

Table 4

Operation Resultant	Operation Specifier	Alternative Specifier
difference	**-**	**diff**
sum	+	**sum**
product	*****	**prod**
quotient	/	**quot**

All forms of the “oplot” command specify one or more pairs of “operand” time series, each pair of which yields a single operation time series. The most general form of the command includes the operation specifier, a result for the first operand series, a class name and list of object identifiers for the set of first operand series’, a result for the second operand series, and a class name and list of object identifiers for the set of second operand series’.
Each operand pair is generated by taking, in order, one object from each of the two identifier lists (along with the associated result). The first operand pair is found by taking the first object from each identifier list, the second pair by taking the second object from each list, etc. If either object identifier list is shorter than the other, then its last object is repeated by the internal logic to effectively bring the the two identifier lists to the same length. This permits a single object to provide either the first or second operand time series in all of the computed operation time series.
Default processing permits more compact expressions. If the second operand identifier list is left out, then the first operand list is parsed pair-wise to generate the first/second operand pairs of time series’. If the first operand identifier list is left out, then the current “select” object list is used. If the second operand result is left out, it defaults to the first operand result. If the first operand result is left out, it defaults to the current result (note that you cannot leave out both the first operand result and identifier list and then specify the second operand result as the parser would interpret it as the first operand result).

Examples:

The first five examples below generate the same three operation time series and illustrate the various default substitutions. The computed time series are those representing the difference in result “dispx” between node pairs (100, 300), (150, 450), and (200, 500).

Example 1 - full syntax:

oplot - dispx node 100 150 200 dispx node 300 450 500

Example 2 - second operand result defaults to first operand result:
oplot - dispx node 100 150 200 node 300 450 500

Example 3 - second operand result defaults to first operand result and first operand identifer list is parsed to generate pairs:
oplot - dispx node 100 300 150 450 200 500

Example 4 - first operand identifier list defaults to current "select" list, second operand result defaults to first operand result, and first operand identifer list is parsed to generate pairs:
select node 100 300 150 450 200 500
oplot - dispx

Example 5 - first operand result defaults to current shown result, first operand identifier list defaults to current "select" list, second operand result defaults to first operand result, and first operand identifer list is parsed to generate pairs:
show dispx
select node 100 300 150 450 200 500
oplot -

The last example illustrates other features of "oplot" processing and capabilities, creating a comparison (difference) between four nodal X velocities and the rigid-body X velocity of a single material.

Example 6 - mixed results and object classes in operands and automatic identifer list extension. The four computed time series resulting from the following expression represent the difference between the nodal X velocity ("velx") at the specified nodes (100, 150, 200, and 250) and the material rigid body X velocity ("matv[matxv]") of a single material (3):
oplot - velx node 100 150 200 250 matv[matxv] mat 3

outth *filename* **cur**
outth *filename* [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]
outth *filename operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]

Generate time history data as needed and save it in the specified file as ASCII text. Three forms of the command are permitted; all begin with "outth *filename*". If keyword "cur" follows *filename*, the time series most recently generated by a "plot" or "oplot" command (whether currently displayed or not) will be output to the file. Otherwise, arguments following *filename* must match either the "plot" command argument list (second "outth" definition above) or the "oplot" command argument list (third "outth" definition above) and will be parsed accordingly.

gather [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]
gather *operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]

Gather time series data into memory. This command can be used to gather many time series in one traversal of the database. Two forms of the command argument list are permitted. The argument list must match either the “plot” command argument list (first form above) or the “oplot” command argument list (second form above) and will be parsed accordingly.

delth {**all** | {*index* | *result* | *class_short_name*}...}

Delete time series that have been gathered into memory. There are two forms of “delth.” One takes the single argument “all” and causes all gathered time series to be deleted. In the other form, the time series to be deleted are specified by one or more indices (these are the numeric indices provided in the output of the “tell th” command), one or more result names (in which all time series of each specified result are deleted), or one or more class names (in which all time series for objects belonging to the specified classes are deleted). Combinations of the three types of specification are permitted. If operation time series have been computed, a *result* or *class_short_name* argument to “delth” will cause all operation time series to be deleted for which the specified result or class represents either of the operand results or classes used to compute the operation time series. The “oplot” command operation specifiers are not permitted as *result* arguments to the “delth” command.

{**on** | **off**} **glyphs**

Turn on (off) display of glyphs (geometric symbols) over the plot curves. There are currently 15 unique glyphs available and 20 unique plot curve colors available, permitting 300 plot curves to be differentiated (default - off).

{**on** | **off**} **glyphcol**

Turn on (off) rendering of glyphs in their associated curve colors. When “off glyphcol” is in effect, glyphs are rendered in the current foreground color (default - off).

glyphqty *n*

Set the quantity of glyphs to render per curve (approximate). The data interval implied by the glyph quantity is actually mapped to a delta time to arrive at a quantity of states between glyphs. When time is the abscissa, this leads to a regular horizontal spacing of the glyphs (assuming the quantity of time samples is large relative to the quantity of glyphs). When a result value is the abscissa, glyphs are distributed over the curves, but the displacement along each curve between glyphs will only be regular to the degree that the abscissa and ordinate results vary regularly with respect to time (default - 5).

glyphscl *v*

Set the scale coefficient for glyphs to control their size. A negative value will invert the glyphs (default - 1.0).

{**on** | **off**} **minmax**

Turn on (off) display of min and max values. The min and max are drawn from the ordinate domains of all curves plotted. This directive also has a mesh view mode analog, described on page 36. The min and max are initially rendered inside the upper left corner of the plotting window, but can be moved to any other corner via the “mmloc” command.

mmloc {**ul | ll | lr | ur**} Set the corner of the plotting window in which the ordinate min and max values will be rendered. The arguments “ul,” “ll,” “lr,” and “ur” dictate the upper left, lower left, lower right, and upper right corners, respectively.

{**on | off**} **plotcoords** Turn on (off) the cursor coordinates display at the bottom of the rendering window. This display consists of two text fields giving the coordinates, in data units, of the current cursor position. It is active when the cursor is within the box defined by the plot axes, and blank otherwise (default - on).

{**on | off**} **plotcol** Turn on (off) rendering of plot curves in different colors. When “off plotcol” is in effect, plot curves are rendered in the current foreground color (default - on).

{**on | off**} **plotgrid** Turn on (off) the background grid in the plot area. When “off plotgrid” is in effect, tic marks annotate the x and y axes (default - on).

{**on | off**} **leglines** Turn on (off) rendering of short curve segments, appropriately colored, as part of the legend entries. If “on glyphs” is in effect, sample glyphs are drawn over the legend lines (default - off).

setcol plot*n r g b* Set the color for plot curve *n*, where *n* is a positive non-zero integer (note there is no space preceding *n*). Color components *r*, *g*, and *b* (for red, green, and blue, respectively) are each specified as floating point numbers on [0, 1]. GRIZ stores a limited number of plot colors, so *n* is internally recomputed as a modulo value of the quantity available (currently 20) to arrive at the plot color index actually modified.

switch glyphstag Set the vertical alignment of glyphs such that the *n*th glyph (*n* varying from one to the quantity of glyphs per curve; see “glyphqty”) on each curve is uniquely offset in the X direction. This keeps glyphs from overlapping when the curves have approximately the same Y value at points where glyphs are being drawn. This is the default glyph alignment.

switch glyphalign Set the vertical alignment of glyphs such that the *n*th glyph (*n* varying from one to the quantity of glyphs per curve; see “glyphqty”) on each curve is identically offset in the X direction.

timhis Plot the current result for the currently ‘select’ed objects. Although “timhis” has been superseded by the “plot” command (“plot” without arguments behaves identically to “timhis”), “timhis” is still supported.

3.8 Output Commands

Output commands create binary or text output files.

outps *filename* Save the image in the mesh view window to a PostScript file with the given name. The image is rotated if necessary to optimally match the aspect ratio of the window to the page, and it is scaled to use as much of the page as possible.

outrgb *filename* Save the image in the mesh view window to an SGI Image file (also referred to as an "rgb" file) with the given name.

outrgba *filename* Same as "outrgb" except the alpha channel is included in the output file.

outjpeg *filename* Save the image in the mesh view window to a jpeg-compressed image file. The amount of compression (and image quality) are controlled by the "jpegqual" command. [Note - this command is only available if the JPEG_SUPPORT macro is set during GRIZ compilation and the jpeg library is available for linking.]

jpegqual *n* Set the jpeg quality coefficient to trade-off compression vs. image quality. The valid range of values for *n* is 1-100, with maximum quality/minimum compression available at *n* = 100 (default - 75).

outpng *filename* Save the image in the mesh view window to a PNG format image file with the given name. [Note - this command is only available if the PNG_SUPPORT macro is set during GRIZ compilation and the png library is available for linking.]

outmm *filename* [*mat...*]
Create a text report of minimum and maximum values of the current node or element result at each state, broken out by material. One or more material numbers may be listed following the "filename" argument, in which case only those materials will be reported. If no material numbers are given, the report is generated for all materials that are not disabled or excluded. Note that specifying materials to the "outmm" command overrides any current disabled/excluded status for those materials. Materials whose elements or nodes don't support the current result are not reported. For nodal results, nodes on the boundaries between multiple materials are considered in the min/max searching for all referencing materials. Elements that have failed and nodes for which all referencing elements *among the specified materials* have failed are not considered in the min/max searching. The states evaluated for the report are constrained by the current settings of "minst" and "maxst".

outobj *filename* Save the current mesh and result data to a polygon object file for input to

Wavefront (tm). The filenames should be basename.0001.obj, basename.0002.obj, etc.

outview *filename* Output a text file containing a (minimal) set of GRIZ commands which will reproduce the current mesh view when read in (see “rdhis”).

outhid *filename* Save the current mesh polygon data to a file for input to the HIDDEN program. See discussion below for more information.

The GRIZ software package comes with a hidden-line program named HIDDEN for generating black-and-white vector PostScript images of initial and deformed mesh geometry. Polygon data for the HIDDEN program is output from GRIZ using the “outhid” command. HIDDEN reads from standard input and writes to standard output. The following command typed at the shell prompt will process output from the “outhid” command into a PostScript file (“.hid” and “.ps” suffixes are included only to distinguish the input and output files):

hidden < *filename*.hid > *filename*.ps

HIDDEN implements a very simple object-space hidden line algorithm which is quite slow but usually produces good results. It is suggested that the user run the above process in the background.

Normally, HIDDEN orients the image in either portrait mode or landscape mode based on the original window size in GRIZ. If the window was larger in the X direction, a landscape image would be produced. The default orientation can be overridden with the -p flag (portrait orientation) or -l flag (landscape orientation). Giving HIDDEN a -v flag (verbose) causes it to print status information while it is generating the hidden-line image.

The following output commands are documented in more detail in other sections as noted.

outth *filename* Save time history data in text file. See page 42.

outvec *filename* Save vector components in text file. See page 52.

outpt *filename* Save particle traces in text file. See page 53.

3.9 Query Commands

tell *report...* Print the specified reports to the feedback window. A sequence of one or more of the report specifiers described in the table below may be requested.

Table 5. "Tell" Command Reports

Report Specifier	Contents
info	Database name, quantity of states, start and end times; current state and time, result information, and bounding box.
times [*first_state* [*last_state*]]	Times for each state in the database plus delta times between adjacent states. If *first_state* is supplied, only the time for that state will be reported. If *first_state* and *last_state* are supplied, times for the range of states between those two states (inclusive) will be reported.
results	Available derived results, organized by superclass, and available primal results, organized by class. Both short and long names are provided. For non-scalar primal results, array dimension sizes and vector component short names are also reported, as appropriate.
class	Mesh object classes defined on the current mesh and their sizes.
pos *class_short_name ident*	Current coordinates of the specified object. The *class_short_name* should identify a node or element class. For elements, the coordinates of the nodes referenced by the element are reported. The report notes if an element is an inactive SAND element.
mm [*result* [*first_state* [*last_state*]]]	Result minimum and maximum values. With no arguments, "tell mm" reports the current result over all states in the database. If *result* is specified, the report is for that result. If *first_state* is specified (but not *last_state*) then the report contains only that state. If both *first_state* and *last_state* are specified then the report covers those states, inclusively. The report includes the mesh objects at which the extrema occur as well as an overall summary of the states traversed for the report.
th	Gathered time series currently in memory.
iso	Currently defined isosurface/isocontour percentages and, when a result is being displayed, the result values associated with each percentage.
view	Current view parameters and light locations.
em	Currently defined material/exploded view associations.

info Command alias for “tell info class view”.

lts Command alias for “tell times”.

tellmm [*result* [*first_state* [*last_state*]]]
Command alias for “tell mm”.

telliso Command alias for “tell iso”.

tellpos *class_short_name n*
Command alias for “tell pos”.

tellem Command alias for “tell em”.

3.10 Traction Commands

Traction commands provide the capability to compute traction forces and moments over a user-defined surface which overlays the mesh in a region of interest. The computed quantities are reported in the feedback window. The surface can be one of several pre-defined simple geometric shapes which the user positions, scales, and orients with respect to the mesh, or the surface can be an arbitrary shape defined by points read in from a text file.

Computing traction forces is a two-step process. First the geometric surface is defined via one of the several forms of the “surface” command. From this, GRIZ computes the surface definition points used to integrate the traction, a local coordinate system $U(i,j,k)$ for the surface, and the surface area. Second, the user issues the “traction” command to request calculation of traction forces and moments.

GRIZ populates each geometric surface with a set of points used to integrate the mesh stresses. The quantity of points is based upon specification(s) provided by the user. For pre-defined shapes an option allows GRIZ to automatically compute a set of points necessary to achieve a traction area convergence integral of less than 1% (bounded to a maximum of 10^6 points). For an arbitrary shape, the user explicitly defines the points, their local surface normals, and associated areas in a text file.

For most forms of the “surface” command, the user provides a direction vector $\bar{v}$ to orient the surface. GRIZ uses $\bar{v}$ to define the surface U_i axis. The U_j axis is derived from inspection of the components of $\bar{v}$, and, in general, is computed to lie in the global coordinate system plane perpendicular to the largest component of $\bar{v}$. For example, if v_x is largest, U_j will lie in the global *y*-*z* plane. The U_k axis is formed by crossing U_i into U_j. Users may wish to experiment with the "surface" commands, observing the calculated *U* axis components in order to acquire a feel for how *U* varies with $\bar{v}$.

surface rect $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ a\ b$

Define a rectangle of width *a* and length *b* centered at (p_x, p_y, p_z) with surface normal (v_x, v_y, v_z). The *a* dimension lies parallel to the U_j axis. The user may need to iterate with different values of $\bar{v}$, observing the computed U_j components, if it is important that the rectangle be oriented in a particular direction. Parameter *n* defines the minimum number of subdivisions of the rectangle in which traction integration points are placed. If *n* is zero, GRIZ will compute the necessary number of points.

surface spot $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta$

Define a circle of radius δ centered at (p_x, p_y, p_z) with surface normal (v_x, v_y, v_z). Parameter *n* defines the minimum number of "subdivision rings" of the circle in which traction integration points are placed. If *n* is zero, GRIZ will compute the necessary number of points.

surface ring $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta_a\ \delta_b$

Define a circular ring with inner radius δ_a and outer radius δ_b centered at (p_x, p_y, p_z) with surface normal (v_x, v_y, v_z). Parameter *n* defines the minimum number of "subdivision rings" of the circle in which traction integration points are placed. If *n* is zero, GRIZ will compute the necessary number of points.

surface tube $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta_a\ h$ [δ_b]

Define a circular tube with base radius δ_a, height *h*, and (optional) top radius δ_b, with the base centered at (p_x, p_y, p_z) and central axis parallel to (v_x, v_y, v_z). Parameter *n* defines the minimum number of "subdivision rings" along the tube height in which traction integration points are placed. If *n* is zero, GRIZ will compute the necessary number of points.

surface poly *filename*

Read a set of surface definition (i.e., traction integration) points from the specified text file. The first line in the file contains the number of points *n* being defined, and lines 2 through *n* + 1 contain point descriptions, one point per line. Each line has seven blank-separated fields - the three components of the point position (p_x, p_y, and p_z), the three components of the surface normal at the point (N_x, N_y, and N_z), and the area (*dA*) associated with the point. Since the user is providing the points, there is no provision for automatic point computation with this command.

traction {**all** | $n\ mat_1 ... mat_n$}

Compute traction force and moment vectors in both global and surface coordinate systems for the most recently defined surface and write them to the feedback window. Moments are calculated about the origins in both the global and surface coordinate systems. The results are calculated from the stress tensors at the surface definition points which fall within the materials specified. If “all” is specified, then surface points that lie in any material are considered. If “$n\ mat_1...mat_n$” is specified, where “n” is the quantity of materials and “$mat_1...mat_n$” are the material numbers, then only those surface definition points that lie within the specified materials are considered. If “switch noterp” is in effect, the stress tensor is treated as constant over each element, otherwise it is interpolated over the element nodes.

3.11 Visualization Commands

Visualization commands provide specialized 3D visualization functions such as cutting planes, isosurfaces, vector fields, and particle traces.

Cutting Planes

Cutting plane commands create cross-sectional views of volume elements.

{**on** | **off**} **cut** — Turn on (off) display of the currently defined cutting planes. See “cutpln,” below.

{**on** | **off**} **rough** — Turn on (off) the rough cutting plane display. The “rough cut” option deletes all elements that intersect the cutting plane or are on the side of the cutting plane in which the plane normal points. This command is useful in checking for degenerately shaped elements in the interior of a volume or for selecting nodes in the interior of the mesh.

cutpln $p_x\ p_y\ p_z\ n_x\ n_y\ n_z$
Define a cutting plane. The cutting plane is defined by a point on the plane and a vector normal to the plane. The cutting plane is added to any previously defined cutting planes.

clrcut — Clear all cutting planes.

Isocontours and Isosurfaces

{**on** | **off**} **con** — Turn on (off) display of contour curves on the surface of the mesh (contours are off by default). Contour curves are drawn at the isovalues in the isovalue list. By default, there are six evenly-spaced isovalues on the list specified as percentages of the difference between the result min and max values. The contour color can be changed using the “setcol con” command.

conwid *v* — Set the line width for isocontours. Argument *v* is a floating point number greater than zero. Fractional widths are allowable, but the actual resolution is platform-dependent (default - 1.0).

{**on** | **off**} **iso** — Turn on (off) display of isosurfaces (isosurfaces are off by default). Isosurfaces (surfaces of constant result value) are drawn at the isovalues in the isovalue list. By default, there are six evenly-spaced isovalues on the list specified as percentages of the difference between the result min and max values. Isosurfaces are currently displayed only in hexahedral elements.

ison *n* — Clear the current isovalues and create *n* isovalues which are evenly distributed between the result minimum and the result maximum.

isop *v* — Add an isovalue to the isovalue list that is *v* percent of the way between the result minimum and the result maximum. The value *v* should be in the range [0.0, 1.0].

isov *v* — Add an isovalue to the isovalue list that is at result value *v*. The value *v* should be in the range [result minimum, result maximum].

clriso — Clear all values from the isovalue list. Before displaying contours or isosurfaces, you must re-create a list of isovalues using “ison,” “isop,” or “isov.” At startup, the program has 6 evenly-distributed isovalues.

tell iso — Write a report to the feedback window of currently defined isosurface/isocontour percentages and, when a result is being displayed, the result values associated with each percentage (alias: telliso).

Vector Fields

The following commands control vector field displays. Vectors are displayed either at nodes or at regularly-spaced points defined by the vgrid commands. The “vgrid...” commands create one-, two-, or three-dimensional arrays (grids) of points at which to display the vector result. These vector grids are topologically independent of the finite element mesh, but occupy the same dimensional space. Two- and three-dimensional grids are axis-aligned. Multiple vgrid commands can be used to successively build up a list of points.

Vectors are drawn as small line segments. In 2D databases, the vectors are rendered with arrowheads. In 3D databases, the vectors are optionally rendered with spheres at their bases.

{**on** | **off**} **vec** — Turn on (off) display of vectors. Vectors are rendered at the vector source points as defined by the “nodvec/grdvec” switch setting (below). The vector quantity displayed must have been previously defined with the “vec” command. Vectors are off by default.

vec *rnx* [*rny* [*rnz*]] Define a vector quantity to be rendered. The arguments are normally result names from Table 3 and are used as the X, Y, and Z components of the vectors. Alternatively, any component can be specified as zero to eliminate its contribution to the vector. Components not specified are taken to be zero.

switch nodvec Render vectors at mesh nodes. This is the default.

switch grdvec Render vectors at vector grid points. Grid points must have been previously defined with the “vgrid1”, “vgrid2”, and/or “vgrid3” commands.

vgrid1 $n\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$
Define *n* evenly-spaced grid points on the line segment between point one and point two. These grid points are the points at which vector results will be displayed.

vgrid2 $n_i\ n_j\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$
Define a rectangular grid of grid points between point one (lower left) and point two (upper right). Point one and two should both lie in an axis-aligned plane (i.e., they should have the same X, Y, or Z value).

vgrid3 $n_i\ n_j\ n_k\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$
Define a rectangular volume of grid points with point one and point two at opposite corners of a diagonal.

clrvgr Clear the list of vector grid points.

setcol vec *r g b* Set the color of the vector line segments by specifying the red, green, and blue components, respectively. The component values are floating point numbers in the range [0.0, 1.0], where a zero value indicates no contribution from the specified component.

veccm Colormap the vectors using the vector magnitudes as the index.

{**on** | **off**} **sphere** Turn on (off) rendering of spheres at the bases of 3D vectors. Spheres are off by default.

vecscl *v* Scale all the vector line segment lengths by *v*.

vhdscl *v* Scale 2D vector arrowheads by *v*.

outvec *filename* Write an ASCII text file containing vector coordinates and vector component values. Data is written for vector source points as determined by the “nodvec”/“grdvec” switch setting.

Particle Traces

ptrace [t_0] [t_{end}] [Δt] Display a particle trace. Existing traces are removed. Particles begin at the initial positions specified with the “prake” command. Their paths over time are calculated by integrating the velocities. The first argument is the time at which to start the particles (defaults to 0.0). The second argument is the time at which to end the particle trace (defaults to the last state). The third argument is the time-step size for integration. All three arguments are optional.

aptrace [t_0] [t_{end}] [Δt] Behaves identically to “ptrace” except new traces are appended to the set of existing traces rather than replacing them. This command is useful for generating repeated “pulses” of traces at different times (in such cases users may find it preferable to limit the length of traces with “ptlim”).

ptstat t t_0 *duration* [Δt]
ptstat s *state_number duration* [Δt]

Generate instantaneous particle traces for a static flow field by integrating velocities at one time slice. The two forms of the command allow users to specify an arbitrary time or a specific state. The *duration* is the length of time over which traces are integrated, and Δt (optional) sets the integration step size, which defaults to 0.001 times *duration.*

prake $n\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$ [*r g b*]

Define *n* evenly-spaced initial particle positions on the line segment between point one and point two, and add them to the current list of initial particle positions. This is referred to as a “particle rake”. The last three arguments specify the color of the particle rake and are optional.

prake *filename* Same effect as the above form of the command, except rake specifications are read from the named file. The file may contain any number of specifications, one per line. The specifications are as described above without the leading word “prake.” Blank lines and lines with “#” in the first column are ignored.

clrpar Clear the current list of initial particle positions.

ptlim *n* Limit the length of particle traces to the most recent *n* particle velocity integration steps. This has the effect of turning the traces into particle “tails” whose length is a function of the particle’s velocity history over the *n* steps.

ptwid *width* Render particle traces with lines *width* pixels wide. *Width* may include a fraction, but the width granularity and maximum line width are platform-dependent.

ptdis *n...* Disable (hide) particle traces in the listed materials. When a material is trace-disabled, any integration step of a particle path which originates or terminates in the material will not be drawn.

outpt *filename* Save current particle traces in the named file as ASCII text.

inpt *filename* — Read the named file (previously created with the "outpt" command) and create particle traces. No checking is performed to verify that the current database is the same database used to create the traces in the file.

3.12 Independent Surface Commands

Independent surfaces are polygonal surfaces displayed independently of the mesh and read in from specially formatted files. There are two types of independent surfaces. "Reference surfaces" are composed of faces of volumetric mesh elements and form a spacial subset of the mesh. "External surfaces" are composed of polygons that are completely independent of mesh elements.

{**on** | **off**} **refsrf** — Turn on (off) display of the reference surfaces which are created by "inref". Reference surfaces are off by default.

{**on** | **off**} **refres** — Turn on (off) display of result values on reference surfaces.

{**on** | **off**} **extsrf** — Turn on (off) display of the external surfaces which are created by "inslp". External surfaces are off by default.

inref *filename* — Read in a reference surface file. This file contains a list of faces of hexahedral elements. The purpose is to treat these faces as a separate surface that can be displayed independently of the volume data. Display of the reference surfaces is turned on and off with **"**on/off refsrf". The file consists of the number of hex faces in the reference surface, an optional material number to associate with the surface, followed by the node numbers for each face. The material associated with the elements providing the faces for the surface sets the color and material translation of the surface by default. If *mat* is specified, it allows the surface to be associated with a different (arbitrary) material. This unlinks it from the color and material translation values of the default material. The file format is

> *num_faces* [*mat*]
> *n1 n2 n3 n4*
> n1 n2 n3 n4
> etc...

outref *filename* [{**x** | **y** | **z**} *v*]...
Write out the current hexahedral element face tables as a reference surface file. If optional constraints are specified, only faces are output for which <u>all</u> the nodes defining the face have an x, y, or z value within 0.001 of a constraint on that dimension. These constraints are useful mainly to create reference surfaces from flat collections of axis-aligned hex element faces (and assuming the overall mesh extents are large relative to a distance of 0.001).

clrref Clear (delete) all reference surfaces.

inslp *filename* Read in polygon data in SLP format from a file. The SLP format is an ASCII geometry file written by Pro/ENGINEER. This command provides a way to display surfaces which aren't part of the mesh. Display of the surfaces is turned on and off with **"**on/off extsrf".

3.13 Interaction History Commands

Command Histories

Command history files provide a general mechanism for running GRIZ in batch mode or for executing a series of commands at once. The history mechanism can be used to quickly customize the display for a particular dataset. A history file is an ASCII text file with one GRIZ command per line.

savhis *filename* Begin saving commands to a file designated by *filename*. All commands are saved, including menu picks and interactive control commands.

endhis Stop saving commands to a file.

rdhis *filename* Read in a command history file and execute the commands in it. (Alias: "h").

pause *n* Pause for approximately *n* seconds before executing the next command. This command is intended for demonstrations.

echo "*Some string***"** Print a string to the terminal. This command is useful when the user wishes to keep track of what command has just been executed.

loop *filename* Go into an endless loop, executing the commands in the specified file over and over. This command is useful for demonstrations.

Feedback Histories

savtxt *filename* Begin saving a copy of all text written to the feedback window into the named text file.

endtxt Stop saving feedback text and close the text file.

3.14 Colormap Commands

Colormap commands allow the user to modify the colormap. The colormap consists of 256 entries, each of which has *r*, *g*, and *b* values in the range [0.0, 1.0]. Low result values are colormapped to entries low in the colortable; high result values are colormapped to entries in the high end of the colortable. When a minimum result value or maximum result value has been set, the first and last entries of the colormap have a special role. Result values exceeding the min or max cutoffs are colormapped to the colors in those entries. When a min or max has been set, the first and last entries are drawn wider in the colormap display so that they can be seen easily.

GRIZ Version 1.0 accepted HDF color palettes using the “ldhmap” command to read them in. The “ldhmap” command and direct support for HDF palettes are no longer supported. However, a utility program, “h2g,” is provided which reads in an HDF palette file and writes out a GRIZ colormap specification to standard output. As an example, the shell command syntax to read an HDF palette file named “mypal.hdf” and save it into a GRIZ colormap file named “mypal.cmap” would be:

```
h2g mypal.hdf > mypal.cmap
```

ldmap *filename* — Load a GRIZ colormap. The specified text file should have 256 entries, one entry per line, with three color component values for each entry (red, green, and blue). Each component must have a value in the range [0.0, 1.0].

posmap p_x p_y s_x s_y — Position the colormap on the screen. The arguments are the position of the lower left corner of the colormap and the size of the colormap in the X and Y directions. The view window extends from approximately -1 to 1 in each direction, so those coordinates can be used as a guide. When the colormap has been repositioned, text annotations on the colormap are no longer displayed. This is mainly intended for animations.

hotmap — Load in the default hot-cold colormap.

grmap — Load in a grayscale colormap.

igrmap — Load in an inverse grayscale colormap.

invmap — Invert the current colormap. The high value becomes the low value and vice versa.

The following commands are documented in greater detail under Colormap Thresholding on page 32.

rzero *v*	Set a result zero tolerance.
rmin *v*	Set a minimum result value.
rmax *v*	Set a maximum result value.
clrthr	Clear all threshold values.

Colormap Contouring

These commands are intended to be used with "good" polygon color interpolation ("switch gterp").

conmap *n*	Contour the current colormap by sampling it at *n* places to create *n* equally-sized bands of colors from the sample points.
chmap *n*	Create a contoured hot-cold colormap with *n* bands.
cgmap *n*	Create a contoured grayscale colormap with *n* bands.

3.15 Additional View, Rendering, and Mesh Control Commands

These commands allow users to fine-tune various aspects of the visual display and are typically useful in specialized situations.

Reflection Planes

Reflection planes allow visualization of whole objects when, to take advantage of symmetry, meshing and computation were performed on only a subset of the object.

{**on** | **off**} **sym**	Turn on (off) reflection of the mesh geometry using reflection planes defined with the "sym" command. Reflection is off by default.
sym p_x p_y p_z n_x n_y n_z	Add a reflection plane to the list of reflection planes. The plane is specified by a point on the plane and the normal vector to the plane.
clrsym	Clear (delete) the list of reflection planes.
switch symcu	Reflection planes are cumulative. The initial geometry is reflected across the first reflection plane. The initial and reflected geometry are then reflected across the second reflection plane, etc. This is the default reflection mode.
switch symor	Reflection planes reflect only the original geometry and are not cumulative.

Material Translations

tmx *n v* — Translate material number *n* an absolute value *v* along the mesh X axis from its original position.

tmy *n v* — Translate material number *n* an absolute value *v* along the mesh Y axis from its original position.

tmz *n v* — Translate material number *n* an absolute value *v* along the mesh Z axis from its original position.

clrtm — Clear all material translations.

Particle Data Options

{**on** | **off**} **particles** — Turn on (off) display of particles when they are detected in a database and have a position vector variable among their data. These particles are unrelated to particle traces employed as a visualization technique (default - on).

partrad *v* — Set the radius for particles in global mesh displacement units (default - 0.25).

Exploded Views

These commands generate exploded views of objects by operating on multiple materials simultaneously. The commands “emsph”, “emcyl”, and “emax” set up spherical, cylindrical, or axial exploded views, respectively, by defining an association between a group of materials and a designated line or point. The materials are “exploded” by translating the center of each material in the direction dictated by the association. A material center is calculated by averaging the positions of the nodes referenced by all elements of the material.

em [*name*...] *distance*

Explode materials by translating them the specified distance in the direction dictated by their exploded view association (defined with the “emsph”, “emcyl”, or “emax” commands). A distance of 0.0 leaves materials untranslated. If name(s) given, only explode those associations.

emsc [*name*...] *scale*

Explode materials by translating each one by the product of “scale” and its distance from its reference point or line (defined with the “emsph”, “emcyl”, or “emax” commands). A scale of 1.0 leaves materials untranslated. A scale of 0.0 coalesces the material centers about the reference point, line, or midplane for spherical, cylindrical, or axial association types, respectively. If name(s) given, only explode those associations.

emsph *x y z* {**all |** *n...*} [*name*]
Associate the specified materials (indicated by the keyword “all” or a list of explicitly numbered materials) with a point in space for a spherical exploded view. The association may optionally be named.

emcyl $x_1\ y_1\ z_1\ x_2\ y_2\ z_2$ {**all |** *n...*} [*name*]
Associate the specified materials (indicated by the keyword “all” or a list of explicitly numbered materials) with a line in space, defined by two points, for a cylindrical exploded view (materials exploded radially about the line). The association may optionally be named.

emax $x_1\ y_1\ z_1\ x_2\ y_2\ z_2$ {**all |** *n...*} [*name*]
Associate the specified materials (indicated by the keyword “all” or a list of explicitly numbered materials) with a line in space, defined by two points, for an axial exploded view. A line direction is assumed from point one to point two. The line midpoint divides the mesh space in two. Materials in the half-space pointed at by the line direction will be translated in that direction; materials in the other half-space will be translated in the opposite direction. The association may optionally be named.

emrm {**all |** *name...*} Remove material/exploded view association(s). If names are specified, only remove the named associations.

clrem Clear all material translations. This command does not remove material/exploded view associations. “clrem” is an alias for “clrtm” and the two may be used interchangeably.

tellem Write a report to the feedback window of all material/exploded view associations currently defined.

Projection Type

switch persp Switch to a perspective view. The default eye position is at (0, 0, 20) in the view coordinate system. The viewport extends from -1 to 1 along the X and Y axes at the point 1 in Z. This is the default.

switch ortho Switch to an orthographic view. Parallel lines on the mesh will stay parallel on the screen in this view.

Mesh Edge Operations

crease ϕ Set the edge detection angle ϕ for surface normal smoothing. The default angle is approximately 44 degrees.

getedg Force edge re-calculation. This will reflect mesh distortion and material visibility changes since the last edge calculation.

edgwid *width* Set the width in pixels of mesh edges rendered when “on edges” is in effect. Available line width granularity and maximum width are OpenGL implementation-dependent.

edgbias *bias* Set the depth bias applied to edge line segments during rendering. This bias (initial value .005) translates all edge line segments closer to the user’s eyepoint. This is useful to avoid occlusion artifacts which can occur when element faces are coincident with edge segments. It has the undesirable side effect of making the tips of edge line segments that *should* be occluded (because they are formed from the edges of back facing external element faces) visible.

Color Specification

Color specifications require the input of three color component values: red, green, and blue. The combination of these components determines the resultant color. A component value of zero indicates no contribution to the resultant color from that component. When all three components have the same value, the resultant color is a shade of grey which varies from black (all zero) to white (all one). Any of the color specifications below can be reset to their initial values by replacing the “*r g b*” triad with the word “default”.

setcol text *r g b* Set the text color. This is the color in which text in the rendering window is drawn. The color components are floating point numbers in the range [0.0, 1.0]. The default color is black (0.0, 0.0, 0.0).

setcol mesh *r g b* Set the color in which mesh lines are drawn (i.e., when “switch hidden” is in effect). The color components are floating point numbers in the range [0.0, 1.0]. The default color is black (0.0, 0.0, 0.0).

setcol edges *r g b* Set the color in which edges are drawn (i.e., when “on edges” is in effect). The color components are floating point numbers in the range [0.0, 1.0]. The default color is black (0.0, 0.0, 0.0).

setcol fg *r g b* Set the foreground color. This is the color for rendered lines which aren’t controlled by any other “setcol” target, such as the mesh bounding box, the coordinate system triad, and the time history plot bounding box. The color components are floating point numbers in the range [0.0, 1.0]. The default color is black (0.0, 0.0, 0.0).

setcol bg *r g b* Set the background color. The color components are floating point numbers in the range [0.0, 1.0]. The default color is white (1.0, 1.0, 1.0).

setcol con *r g b* Set the contour color. This is the color in which contour lines are drawn. The color components are floating point numbers in the range [0.0, 1.0]. The default color is magenta (1.0, 0.0, 1.0).

setcol hilite *r g b* Set the hilite color. This is the color used to hilite picked nodes and elements. The color components are floating point numbers in the range [0.0, 1.0]. The default color is red (1.0, 0.0, 0.0).

setcol plot*n r g b* Set the color for plot curve *n*, where *n* is a positive non-zero integer (note there is no space preceding *n*). GRIZ stores a limited number of plot colors, so *n* is internally recomputed as a modulo value of the quantity available (currently 20) to arrive at the plot color index actually modified.

setcol rmax *r g b* Set the rmax threshold color. This color is used in fringe plots where the result value exceeds the rmax threshold (see “rmax” command). The color components are floating point numbers in the range [0.0, 1.0]. The default color is red (1.0, 0.0, 0.0).

setcol rmin *r g b* Set the rmin threshold color. This color is used in fringe plots where the result value is smaller or more negative than the rmin threshold (see “rmin” command). The color components are floating point numbers in the range [0.0, 1.0]. The default color is magenta (1.0, 0.0, 1.0).

setcol select *r g b* Set the select color. This color is used for selected nodes and elements. The color components are floating point numbers in the range [0.0, 1.0]. The default color is green (0.0, 1.0, 0.0).

setcol vec *r g b* Set the vector color. This is the color in which vectors are drawn. The color components are floating point numbers in the range [0.0, 1.0]. The default is black (0.0, 0.0, 0.0).

Any of the target color specifications above can be reset to their initial values by replacing the “*r g b*” triad with the word “default”, as below.

setcol *target* **default** Reset the target color specification to its initial value, where *target* is one of “text,” “mesh,” “edges,” “bg,” “fg,” “con,” “hilite,” “plot*n*,” “rmax,” “rmin,” “select,” or “vec.”

Image Loading and Display

inrgb [**bg**] *filename* Load the specified rgb image file into memory. If the “bg” argument is not given, the image is loaded into the framebuffer and freed from memory. The image will stay in the rendering window until any activity causes GRIZ to redraw the window. If “bg” is specified, the image is not loaded into the framebuffer but is retained indefinitely in memory for use as a background image in the rendering window.

{**on** | **off**} **bgimage** Turn on (off) display of the currently loaded rgb image in the background of the rendering window. The lower left corner of the rgb image is anchored to the lower left corner of the rendering window. If the window is larger than the image in either dimension, the window is filled with the current background color. If the image is larger in either dimension, it is clipped to the rendering window. If no rgb image is currently loaded into memory, "on bgimage" will have no effect. As soon as an image is loaded, however, it will appear in the rendering window.

Material Properties

mat *n* [**amb** *r g b*] [**diff** *r g b*] [**spec** *r g b*] [**shine** *v*] [**emis** *r g b*] [**alpha** *v*]

Sets the material display properties of material number *n*. Each of the arguments in brackets is optional. The arguments and their permitted ranges are outlined below.

amb: Ambient light color (0.0-1.0).
diff: Diffuse light color (0.0-1.0).
spec: Specular light color (0.0-1.0).
shine: Shininess (0-128; a value of 0 disables specularity). Higher values give smaller, more tightly focused specular highlights.
emis: Emissive light color, object gives off light (0.0-1.0).
alpha: Opacity (0.0-1.0, where 0.0 = transparent, 1.0 = opaque).

Normally, only the first two or possibly first four arguments are used. Only those properties that are specified in the command are modified. For practical purposes, the diffuse component sets the material color. The diffuse and ambient components should be set to the same color for intuitive results. To make material 3 bright green, for example, one could use the command "mat 3 amb 0 1 0 diff 0 1 0".

The Material Manager provides a superset of these capabilities and can operate on multiple materials simultaneously.

Lighting

{**on** | **off**} **lighting**

Turn on (off) the lighting contribution to the shading of 3D mesh element faces. When "off lighting" is in effect facets are colored independent of their orientation. This can allow more accurate interpretation of color fringes, but spatial orientation cues provided by lit facets are lost. Rendering time is slightly improved with lighting off (default - on).

light *n x y z w* [**amb** *r g b*] [**diff** *r g b*] [**spec** *r g b*] [**spotdir** v_x v_y v_z] [**spot** *exp* ϕ]

Defines a new light or modifies the definition of a previously-defined light. The light number *n* and homogeneous coordinates *x, y, z,* and *w* are required; the arguments in braces are optional. Coordinate *w* is normally set to 0.0 or 1.0. When *w* is 0.0, the light is treated as a directional light, i.e., the direction to the light (defined by *x, y,* and *z*) is constant for all objects in the scene. The light color is defined by “amb”, “diff”, and “spec”. The *r*, *g*, and *b* color components are values in the range (0.0-1.0). GRIZ’s default lights define all components for “diff” and “spec” to be 1.0 and all components for “amb” to be 0.0. The last two arguments are used to define spot lights. Spot lights will tend to slow the rendering performance. The “spotdir” is the direction in which the spotlight is aimed. The “spot” argument *exp* is the falloff exponent (0-128) where 128 gives the sharpest possible falloff and 0 gives a constant intensity across the cone of light. The “spot” argument ϕ is the spread angle (0-90 degrees) of the light cone. Setting the exponent to 0 and the spread angle to 180 will turn off the spotlight effect.

tlx *n v* Translate light number *n* in the X direction by a value *v*.

tly *n v* Translate light number *n* in the Y direction by a value *v*.

tlz *n v* Translate light number *n* in the Z direction by a value *v*.

dellit Deletes all lights.

Graphical Viewing Parameters

camang ϕ Set the perspective camera angle ϕ in either the X or Y direction dependent upon the aspect ratio of the rendering window.

lookfr p_x p_y p_z Set the location of the look from (eye) point. The default is (0, 0, 20).

lookat p_x p_y p_z Set the location of the look at point. The default is (0, 0, 0).

lookup v_x v_y v_z Set the look up vector. This vector controls the up (Y) direction of the current view. The up vector must not be parallel to the look direction. The default is (0, 1, 0).

tfx *v* Translate the look from point by value *v* in the X direction.

tfy *v* Translate the look from point by value *v* in the Y direction.

tfz *v* Translate the look from point by value *v* in the Z direction.

tax *v* Translate the look at point by value *v* in the X direction.

tay *v* Translate the look at point by value *v* in the Y direction.

taz *v* Translate the look at point by value *v* in the Z direction.

near *v* Set the near plane position to value *v*. The current position of the near plane can be obtained with the “info” command. Near plane positioning is tricky. This command should be avoided and “rnf” used instead when possible.

far *v* — Set the far plane position to value *v*. The current position of the far plane can be obtained with the "info" command. Far plane positioning is tricky. This command should be avoided and "rnf" used instead when possible.

Rendered Number Formatting

fracsz *n* — Set the number of digits to render after the decimal point on the colormap scale, mesh object result min/max values, result values on selected objects, and time history plot axis scales. This does not override the effect of "on autosz" (below) on time history plots (default - 2).

{**on** | **off**} **autosz** — Turn on (off) automatic determination of the quantity of digits to render after the decimal point on time history plot axis scale values. The quantity will be set so that adjacent scale values will be differentiated with a minimal number of digits (default - on).

Polygon Shading Type

switch smooth — Smooth-shade the polygons using averaged normals at the vertices. This is the default.

switch flat — Flat-shade the polygons. This may be faster than smooth-shading, but looks less realistic.

Bounding Box Operations

bbsrc {**nodes** | **elems** | **vis** | *class_short_name*...}

Set the source classes of mesh objects for mesh bounding box calculation and reset the view. The bounding box will be set to the maximum extents of the objects across all the classes specified. If "nodes" is specified, all node classes present are used. If "elems" is specified, all element classes in the mesh are used (i.e., the extents of the nodes referenced by all elements determine the bounding box). If "vis" is specified, the source is all element classes visible when "bbox" is invoked. Alternatively, if one or more node or element class short names are specified, then only those classes are used (default - all truss, beam, triangle, quadrilateral, and hexahedral element classes in the mesh).

bbox [x_{min} y_{min} z_{min} x_{max} y_{max} z_{max}]

Reset the mesh bounding box and reset the view. If coordinates are given, use them explicitly as the bounding box extents. Otherwise, recalculate the bounding box based on the current value of the bounding box source ("bbsrc") and mesh position at the current state.

{**on** | **off**} **bbmax** With "on bbmax" in effect, GRIZ defines the mesh bounding box by the largest displacement magnitude values experienced across all invocations of the "bbox" command. By invoking "bbox" at appropriate states, one can set the bounding box, and by derivation the scale applied in the default mesh view, to a size that will encompass the mesh regardless of displacement. Explicit box values given with the "bbox" command will be used regardless of "bbmax" setting. (default - off).

Miscellaneous

hidwid *width* Set the pixel-width of element-bounding lines in "hidden" lines mesh view mode. Parameter *width* must be a positive floating point number. In practice, since these lines are aliased, only whole-number widths are supported. The maximum line width is OpenGL implementation-dependent.

{**on** | **off**} **shrfac** OpenGL may cause polygons drawn on top of each other to be blended, yielding ugly artifacts. This is typically evident when shell elements share the same nodes as the faces of volume elements. This artifact can be avoided by turning on the "shrfac" flag at the cost of a slower display update.

{**on** | **off**} **hex_overlap**

Turn on (off) generation of redundant faces from coincident hexahedral elements of the same element class. In the unusual case where a class of hex elements contains multiple elements that share identical connectivities, Griz's normal face table generation logic eliminates all but one instance of each face. This may have undesirable consequences when the mesh is rendered. With "on hex_overlap" in effect, Griz builds the face table incrementally by separately considering groups of elements of the same material. If identically defined elements are of different materials and have faces on the material boundary, all their boundary faces will be generated into the face table, and each such element will be represented as many times as it exists with a different material. If in such an instance more than one material is visible simultaneously, the faces will be rendered for each visible material representation, meaning the last material rendered will color the element. This will be the highest numbered material of those that are visible. Since only the last rendering of a face is visible, all the preceding renderings are wasted effort, so this allowance comes with a performance penalty. Not only are the visible faces redundantly rendered, but the face table will contain (and render) all faces that were on a material boundary even if those faces are not true external faces of the element class. Rendering these faces is also wasted effort. Hex_overlap is off by default.

{**on** | **off**} **zlines** Set whether or not antialiased lines (edge lines, beam and truss elements, particle traces) are depth buffered with respect to each other. With "on zlines" in effect, lines are depth-buffered and rendered correctly in depth. However, there are some rendering artifacts that appear in this mode which can be distracting, particularly during animations. With “off zlines” in effect these artifacts disappear but, depth-buffering may be incorrect as line segments are not sorted by depth before rendering.

bufqty [*class_short_name*] *n*

Set the number of state I/O buffers used by the database I/O library. If a class is specified, only that class is affected. Otherwise, all classes are affected. Performance can be improved by utilizing multiple buffers and obviating multiple disk access at a state. This can be helpful, for example, when interpolating data between states. This feature may not be supported by all I/O libraries. In practice, *n* is limited by available memory.

copyrt Display the copyright/title screen. Any subsequent command that causes the display to update will erase this screen.

3.16 Mouse Picking Commands

The mouse can be used to interactively pick nodes or elements. The left mouse button is used for picking nodes, the middle mouse button is used for picking shell or beam elements (see below), and the right mouse button is used for picking volume elements. The commands in this section control the picking operation.

switch pichil Hilite mode. A mouse pick hilites the picked node, element, or particle. (see “hilite”.) This is the default.

switch picsel Select mode. A mouse pick selects the picked node, element, or particle. (see “select”.)

setpick {**1** | **2** | **3**} *class_short_name*

Set the pick class associated with the specified mouse button. The current settings can be viewed (and modified) in the Utility Panel. Since each database can have an arbitrary number of node and element classes, it may be necessary to modify the assignments GRIZ makes in order to select objects of a particular class. By default, GRIZ tries to assign the first mouse button to a node class, the second to a quadrilateral element class, and the third to a hexahedral element class.

minmov *n* Set the cursor motion threshold for mesh object picks. If the cursor moves *n* or fewer pixels while a mouse button is depressed, the button click (depress/release sequence) will be treated as a pick attempt. If the cursor moves more than *n* pixels, the motion will be interpreted as an interactive mesh manipulation (i.e., a rotation, scale, or translation input). This is useful on some workstations where it is difficult to keep the mouse completely still during pick attempts (default - 0).

3.17 Material Manager Commands

Although the Material Manager was added specifically to provide a graphical interface to performing certain operations on multiple materials, its actions are ultimately implemented as a result of commands being sent to GRIZ’s command interpreter. Material Manager commands can be quite complex and may contain many arguments. They are not normally issued interactively. The anticipated reason for requiring a Material Manager command explicitly is for inclusion in a command script. The most convenient method of preparing a command script is to execute the desired Material Manager operation while a command history file is active, then cut-and-paste the “mtl” command(s) from the history file into the script.

In a mesh with more than a few materials, the total number of command tokens in a Material Manager command can easily exceed the current token limit of 25. To bypass this restriction, a protocol has been implemented to permit the distribution of a Material Manager command’s tokens across several partial commands issued in sequence. The keyword “continue” placed at the end of a partial Material Manager command suspends command execution until the entire sequence is completed. To maintain clarity, the “continue” keyword is not shown in the syntax descriptions below, but it can follow any token in a command sequence as the last token of each partial command. The third bullet in the examples below illustrates the “continue” protocol.

There are several forms of Material Manager commands. The first form manages material visibility and enable-state. Its syntax is

mtl *function...* {**all** | *n...*}

where *function* is any of “**vis**”, “**invis**”, “**enable**”, or “**disable**”, “**all**” represents all materials, and *n...* is an arbitrary list of material numbers. Multiple *function*’s may be included, subject to the constraints that “vis” and “invis” are mutually exclusive and “enable” and “disable” are mutually exclusive.

The second form of Material Manager command manages material color properties. Its syntax is

mtl [**preview**] **mat** {**all** | *n...*} *color_property...*

where “**all**” and *n...* are defined as above. *color_property* is any of “**amb**”, “**diff**”, “**spec**”, “**emis**”, or “**shine**” with associated numeric arguments and meanings as described for the “**mat**” command arguments of the same names (page 62).

When the second form includes the optional “preview” argument, it must be followed by a “mtl” command of the third form to command GRIZ to either keep or reject the previewed changes. The syntax of the third form is:

mtl {**apply** | **cancel**}

The last form of the “mtl” command modifies material color properties by resetting them to their initial (default) values. Its syntax is

mtl default {**all** | *n...*}

Examples:

- mtl invis disable 1 2 3 4 5 6 7 23 24 25 52 57 89 100 101 102
- mtl mat 25 24 23 22 21 20 19 18 17 16 15 amb .8 0 .5 diff .8 0 .5 shine 25
- mtl mat 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 continue
mtl 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 amb 0 .9 .3 diff 0 .9 .3
- mtl preview mat 12 13 14 15 16 17 18 19 20 21 22 amb .5 .5 .8 diff .5 .5 .8
mtl apply
- mtl default 12 13 14 15 16 17 18 19 20 21 22

3.18 Video Commands

vidti *n* **“***Title line***”** Set a video title line, where *n* is in the range 1 to 4. These four lines are displayed with the “vidttl” command.

vidttl Draws a screen with the video title lines in large letters suitable for recording to videodisc. The title can be changed with the “vidti” command. Any action which causes the screen to refresh will erase this screen. This command provides a way of recording separator frames which label animation sequences on the video disc.

{**on** | **off**} **safe** Turn on (off) display of the “safe action” area border. The safe action area borders are five percent of the screen size. The safe title area borders are ten percent of the screen size. The “safe action” area is where all the image content should be contained -- the region outside this area may be off-screen on standard television monitors.

Image Aspect Ratio

switch norasp Set the aspect ratio to 1.0.

switch vidasp Set the aspect ratio for NTSC (x/y = 0.92).

4.0 COMMAND QUICK REFERENCE

View Commands (page 21)

rx ϕ, **ry** ϕ, **rz** ϕ	Rotate view.
tx *v*, **ty** *v*, **tz** *v*	Translate view.
scale *v*	Scale view.
scalax v_x v_y v_z	Scale axes independently.
rview	Reset view.
rnf	Reset near/far planes.
vcent hi	Center view on hilited object.
vcent n *node_number*	Center view on node.
vcent *x y z*	Center view on point in space.
vcent off	Turn off view centering.

State and Time Commands (page 23)

minst *n*	Ignore states before *n*.
maxst *n*	Ignore states after *n*.
stride *n*	Set state stride to *n*.
state *n*	Go to state *n*.
f, **p**, **n**, **l**	First, previous, next, last state.
time *t*	Go to time *t*.
lts	List state times.
anim	Animate states in database.
anim *n*	Animate between “f” and “l” with interpolation.
anim *n* t_s t_e	Animate between t_s and t_e with interpolation.
stopan	Stop animation at current frame.
animc	Continue animation from current state.
{**on** | **off**} **autoimg**	Turn on (off) automatic output of rendered images.
autoimg *rootfilename* [**jpeg** | **png**]	Specify file name root and non-default type for “autoimg” output files.
resetimg	Reset the “autoimg” frame counter to one.

Executive Commands (page 25)

load *name*	Load database.
quit	Exit GRIZ.
on *flag...*	Turn on the specified logical flags.
off *flag...*	Turn off the specified logical flags.
switch *flag*	Set a multiple-valued flag.
info	Tell current state, mesh, view, and light information.
r	Repeat previous command.
{**on** | **off**} **refresh**	Turn on (off) rendering window update.
alias *newcom* “*comstring*”	Create alias for *comstring*.
exec “*command line*”	Execute a UNIX command.

Results Commands (page 25)

show *result*	Display a result variable.
switch {**interp** | **gterp** | **noterp**}	Set result color interpolation mode.
rzero *v*	Set result zero tolerance.
rmin *v*	Set result minimum value.
rmax *v*	Set result maximum value.
clrthr	Clear “rzero”, “rmin”, “rmax” thresholds.

switch {**infin** | **alman** | **grn** | **rate**}	Set strain type.
switch {**rglob** | **rloc**}	Set element (shell, hex) reference frame.
switch {**middle** | **inner** | **outer**}	Set shell result surface.
refstate [*n*]	Set nodal displacement calculation reference state.
{**on** | **off**} **derived**	Turn on (off) searches of the derived result table.
{**on** | **off**} **primal**	Turn on (off) searches of the primal result table.
coordxf $node_1$ $node_2$ $node_3$	Specify alternative coordinate system via three nodes.
coordxf x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3	Specify alternative coordinate system via three points.
{**on** | **off**} **coordxf**	Turn on (off) stress/strain coordinate transformations.
dirv *x y z*	Specify a global direction vector.
dir3n $node_1$ $node_2$ $node_3$	Specify a global direction vector via three nodes.
dir3p x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3	Specify a global direction vector via three points.
switch {**mglob** | **mstat**}	Set result min/max scope.
globmm	Find global min/max for current result.
resetmm	Reset global min/max for current result and clear cache.
{**on** | **off**} **conv**	Turn on (off) result units conversion.
conv *scale offset*	Set units conversion parameters.
clrconv	Turn off units conversion; reset parameters to defaults.
pref *v*	Set reference pressure for “pint” result calculations.
dia *v*	Set beam diameter for beam strain calculation.
ym *v*	Set Young’s Modulus for beam strain calculation.

Optional Rendered Information Commands (page 36)

{**on** | **off**} **all**	Turn on (off) “coord”, “time”, “title”, “cmap”, “minmax”, and “dscal”.
{**on** | **off**} **bgimage**	Turn on (off) use of stored image as background.
{**on** | **off**} **box**	Turn on (off) mesh bounding box.
{**on** | **off**} **cmap**	Turn on (off) colormap and associated indicators.
{**on** | **off**} **coord**	Turn on (off) coordinate axes.
{**on** | **off**} **cscale**	Turn on (off) colormap title and value scale.
{**on** | **off**} **dscal**	Turn on (off) display of nodal displacement scale factors.
{**on** | **off**} **edges**	Turn on (off) mesh edges.
{**on** | **off**} **locref**	Turn on (off) display of local coords on “select”ed shell and hex elements.
{**on** | **off**} **minmax**	Turn on (off) result min/max.
{**on** | **off**} **num**	Turn on (off) node/element numbering.
numclass *class_short_name...*	Specify node/element class(es) for numbering.
{**on** | **off**} **time**	Turn on (off) current time.
{**on** | **off**} **title**	Turn on (off) mesh title.
title “*Title String*”	Update data title.

Mesh Control and Rendering Commands (page 37)

hilite *class_short_name n*	Toggle hilite on specified object.
clrhil	Clear hilite.
select *class_short_name n...*	Toggle selection state of specified objects.
clrsel [*class_short_name n*]	Clear object selection(s).
exclude {**all** | *n...*}	Combined “invis” and “disable” on specified materials.
include {**all** | *n...*}	Combined “vis” and “enable” on specified materials.
disable {**all** | *n...*}	Disable result display on specified materials.
enable {**all** | *n...*}	Enable result display on specified materials.
invis {**all** | *n...*}	Make specified materials invisible.
vis {**all** | *n...*}	Make specified materials visible.
switch {**hidden** | **solid**| **cloud** | **none**}	Set mesh rendering style.

dscal *v*	Scale nodal displacements.
dscalx *v*, **dscaly** *v*, **dscalz** *v*	Scale nodal displacements along axis directions.

Time History Commands (page 39)

plot [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]
Plot time series curves.
oplot *operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]
Plot operation time series curves.
outth *filename* **cur**
outth *filename* [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]
outth *filename operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]
Write “plot” or “oplot” time series data to text file.
gather [*result*...] [*class_short_name n*...]... [**vs** *abscissa_result*]
gather *operation* [[*result*] [*class_short_name n*...] [[*result*] [*class_short_name n*...]]]
Gather “plot” or “oplot” time series’ into memory.
delth {**all** | {*index* | *result* | *class_short_name*}...}
Delete time series’.

{**on** | **off**} **glyphs**	Turn on (off) display of geometric glyphs on plot curves.
{**on** | **off**} **glyphcol**	Turn on (off) rendering of glyphs in curve colors.
glyphqty *n*	Set approximate per-curve glyph qty.
glyphscl *v*	Set size scale coefficient for glyphs.
{**on** | **off**} **minmax**	Turn on (off) display of ymin/ymax values.
mmloc {**ul** | **ll** | **lr** | **ur**}	Set plot box ymin/ymax location.
{**on** | **off**} **plotcoords**	Turn on (off) cursor coordinates display.
{**on** | **off**} **plotcol**	Turn on (off) rendering of plot curves in individual colors.
{**on** | **off**} **plotgrid**	Turn on (off) background grid in plot box.
{**on** | **off**} **leglines**	Turn on (off) inclusion of curve samples in legend.
setcol plot*n r g b*	Set color for plot curve *n*.
switch {**glyphstag** | **glyphalign**}	Set vertical alignment of glyphs.
timhis	Plot time series curves using defaults.

Output Commands (page 45)

outps *filename*	Save rendered image to PostScript file.
outrgb *filename*	Save rendered image to SGI Image file.
outrgba *filename*	Save rendered image to SGI Image file with alpha data.
outjpeg *filename*	Save rendered image to JPEG file.
outpng *filename*	Save rendered image to PNG file.
jpegqual *n*	Set JPEG image quality.
outmm *filename* [*mat*...]	Create text report of result min/max values per state, broken out by material.
outobj *filename*	Save current mesh and result data to Wavefront file.
outview *filename*	Save current view transformation as GRIZ command file.
outhid *filename*	Save current mesh polygon data to HIDDEN file.
outth *filename*	Save time history as text.
outvec *filename*	Save vector components as text.
outpt *filename*	Save particle traces as text.

Query Commands (page 46)

tell *report*...	General information query interface.
info	Alias for “tell info class view”.
lts	Alias for “tell times.”

tellmm [*result* [*first_state* [*last_state*]]]
Alias for “tell mm...”

telliso	Alias for "tell iso."
tellpos *class_short_name n*	Alias for "tell pos..."
tellem	Alias for "tell em."

Traction Commands (page 48)

surface rect $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ a\ b$	Define rectangular surface for traction calculation.
surface spot $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta$	Define circular surface for traction calculation.
surface ring $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta_a\ \delta_b$	Define annular surface for traction calculation.
surface tube $n\ p_x\ p_y\ p_z\ v_x\ v_y\ v_z\ \delta_a\ h\ [\delta_b]$	Define (conical) tubular surface for traction calculation.
surface poly *filename*	Read surface definition from file for traction calculation.
traction {**all** | $n\ mat_1 ... mat_n$}	Compute traction force and moment vectors.

Visualization Commands (page 50)

{**on** | **off**} **cut**	Turn on (off) cutting planes.
{**on** | **off**} **rough**	Turn on (off) rough cutting planes.
cutpln $p_x\ p_y\ p_z\ n_x\ n_y\ n_z$	Define a cutting plane.
clrcut	Clear all cut planes.
{**on** | **off**} **con**	Turn on (off) contours.
conwid *v*	Set isocontour line width in pixels.
{**on** | **off**} **iso**	Turn on (off) isosurfaces.
ison *n*	Create *n* evenly-spaced isovalues.
isop *v*	Add an isovalue at *v* percent of result min/max delta.
isov *v*	Add an isovalue at result value *v*.
clriso	Clear all isovalues.
tell iso	Report isovalue percentages and result values.
{**on** | **off**} **vec**	Turn on (off) vector field display.
vec *rnx* [*rny* [*rnz*]]	Define vector result.
switch {**nodvec** | **grdvec**}	Set vector source.
vgrid1 $n\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$	Add vector grid points along a line.
vgrid2 $n_i\ n_j\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$	Add vector grid points on axis-aligned plane.
vgrid3 $n_i\ n_j\ n_k\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}$	Add vector grid points on axis-aligned volume.
clrvgr	Clear all vector grid points.
setcol vec *r g b*	Set vector color.
veccm	Colormap vectors by magnitude.
{**on** | **off**} **sphere**	Turn on (off) spheres on 3D vector bases.
vecscl *v*	Scale vector lengths.
vhdscl *v*	Scale 2D vector arrowheads.
outvec *filename*	Save vector positions and components as text.
ptrace $[t_0]\ [t_{end}]\ [\Delta t]$	Create particle traces, replacing existing traces.
aptrace $[t_0]\ [t_{end}]\ [\Delta t]$	Create particle traces, augmenting existing traces.
ptstat t t_0 *duration* $[\Delta t]$	Create instantaneous particle traces at time t_0.
ptstat s *state duration* $[\Delta t]$	Create instantaneous particle traces at state *state*.
prake $n\ p_{1x}\ p_{1y}\ p_{1z}\ p_{2x}\ p_{2y}\ p_{2z}\ [r\ g\ b]$	Create particle rake.
prake *filename*	Create particle rake(s) from specifications in file.
clrpar	Clear particle trace positions.
ptlim *n*	Limit length of particle traces to *n* most recent steps.
ptwid *width*	Set particle trace width.
ptdis *n...*	Disable particle traces in specified materials.
outpt *filename*	Save particle traces as text.
inpt *filename*	Read in particle traces from file.

Independent Surface Commands (page 54)

{**on** | **off**} **refsrf**	Turn on (off) reference surfaces.
{**on** | **off**} **refres**	Turn on (off) result display on reference surfaces.
{**on** | **off**} **extsrf**	Turn on (off) external surfaces.
inref *filename*	Read reference surface definition file.
outref *filename* [{**x**|**y**|**z**} *v*]...	Write current external hex faces as reference surface file.
clrref	Clear reference surfaces.
inslp *filename*	Read SLP-format external surface definition file.

Interaction History Commands (page 55)

savhis *filename*	Begin saving commands to file.
endhis	Close command history file.
rdhis *filename*	Read command history file and execute commands.
pause *n*	Pause for approximately *n* seconds.
echo "*Some string*"	Print string to history.
loop *filename*	Loop endlessly over commands in file.
savtxt *filename*	Begin saving feedback window text to file.
endtxt	Stop saving feedback text and close file.

Colormap Commands (page 56)

ldmap *filename*	Load a GRIZ colormap.
posmap p_x p_y s_x s_y	Position colormap in display.
hotmap	Load default hot-cold colormap.
grmap	Load greyscale colormap.
igrmap	Load inverse greyscale colormap.
invmap	Invert current colormap.
conmap *n*	Contour current colormap with *n* equal bands.
chmap *n*	Create contoured hot-cold colormap.
cgmap *n*	Create contoured greyscale colormap.

Additional View, Rendering, and Mesh Control Commands (page 57)

{**on** | **off**} **sym**	Turn on (off) reflections about symmetry planes.
sym p_x p_y p_z n_x n_y n_z	Add symmetry plane.
clrsym	Clear (delete) symmetry planes.
switch {**symcu** | **symor**}	Set reflections to cumulative or not.
tmx *n v*, **tmy** *n v*, **tmz** *n v*	Translate material *n* distance *v* along axis.
clrtm	Clear all material translations.
{**on** | **off**} **particles**	Turn on (off) display of particles from a database.
partrad *v*	Set particle radius.
em [*name*...] *distance*	Explode materials by absolute *distance*.
emsc [*name*...] *scale*	Explode materials by scaled distance.
emsph *x y z* {**all** | *n*...} [*name*]	Associate materials for spherical exploded views.
emcyl x_1 y_1 z_1 x_2 y_2 z_2 {**all** | *n*...} [*name*]	Associate materials for cylindrical exploded views.
emax x_1 y_1 z_1 x_2 y_2 z_2 {**all** | *n*...} [*name*]	Associate materials for axial exploded views.
emrm {**all** | *name*...}	Remove exploded view associations.
clrem	Clear all material translations.
tellem	Alias for "tell em."
switch {**persp** | **ortho**}	Set projection type.
crease ϕ	Set edge detection angle.
getedg	Re-calculate mesh edges.
edgwid *width*	Set line width (pixels) of edges.
edgbias *bias*	Set the depth bias applied to edge line segments.

setcol text *r g b*	Set text color.
setcol mesh *r g b*	Set mesh line color.
setcol edges *r g b*	Set edge line color.
setcol fg *r g b*	Set foreground color.
setcol bg *r g b*	Set background color.
setcol con *r g b*	Set contour color.
setcol hilite *r g b*	Set hilite color.
setcol plot*n r g b*	Set color for plot curve *n*.
setcol rmax *r g b*	Set rmax threshold color.
setcol rmin *r g b*	Set rmin threshold color.
setcol select *r g b*	Set select color.
setcol vec *r g b*	Set vector color.
setcol *target* **default**	Reset color for *target* to default.
inrgb bg *filename*	Load RGB image file.
{**on** | **off**} **bgimage**	Turn on (off) display of loaded image as background.
mat *n* [**amb** *r g b*] [**diff** *r g b*] [**spec** *r g b*] [**shine** *v*] [**emis** *r g b*] [**alpha** *v*]	Set material properties.
{**on** | **off**} **lighting**	Turn on (off) lighting contribution to shading.
light *n x y z w* [**amb** *r g b*] [**diff** *r g b*] [**spec** *r g b*] [**spotdir** v_x v_y v_z] [**spot** *exp* ϕ]	Set/modify light properties.
tlx *n v*, **tly** *n v*, **tlz** *n v*	Translate light *n* distance *v* along axis.
dellit	Delete all lights.
camang ϕ	Set perspective camera angle.
lookfr p_x p_y p_z	Set location of look-from point.
lookat p_x p_y p_z	Set location of look-at point.
lookup v_x v_y v_z	Set look-up vector.
tfx *v*, **tfy** *v*, **tfz** *v*	Translate look-from point distance *v* along axis.
tax *v*, **tay** *v*, **taz** *v*	Translate look-at point distance *v* along axis.
near *v*	Set near plane to position *v*.
far *v*	Set far plane to position *v*.
fracsz *n*	Set number of digits rendered in number fractions.
{**on** | **off**} **autosz**	Turn on (off) automatic sizing of plot axis numbers.
switch {**smooth** | **flat**}	Set polygon shading type.
bbsrc {**nodes** | **elems** | **vis** | *class_short_name*...}	Set bounding box source object class(es).
bbox [x_{min} y_{min} z_{min} x_{max} y_{max} z_{max}]	Recalculate or set bounding box.
{**on** | **off**} **bbmax**	Turn on (off) accumulation of max bounding box extents.
hidwid *width*	Set line width (pixels) of mesh lines.
{**on** | **off**} **shrfac**	Fix coincident polygon rendering bug.
{**on** | **off**} **hex_overlap**	Turn on (off) generation of redundant faces from coincident hexahedral elements of the same element class but different materials.
{**on** | **off**} **zlines**	Turn on (off) depth-buffering of anti-aliased lines with respect to each other.
bufqty [*class_short_name*] *n*	Set quantity of I/O buffers used.
copyrt	Display copyright screen.

Mouse Picking Commands (page 66)

switch {**pichil** | **picsel**}	Set pick mode.
setpick {**1** | **2** | **3**} *class_short_name*	Set mesh object pick class associated with mouse buttons.
minmov *n*	Set cursor motion threshold (pixels).

Material Manager Commands (page 67)

mtl *function*... {**all** | *n*...} [**continue**]	Modify visibility and/or enable-state of multiple materials.
mtl [**preview**] **mat** {**all** | *n*...} *color_property*... [**continue**]	Modify, or preview modification of, material color properties for multiple materials.
mtl {**apply** | **cancel**}	Keep or reject previewed material color property change.
mtl default {**all** | *n*...} [**continue**]	Reset material color properties for multiple materials to their default values.

Video Commands (page 68)

vidti *n* “*Title line*”	Set video title line.
vidttl	Draw video title screen.
{**on** | **off**} **safe**	Show “safe action” area border.
switch {**norasp** | **vidasp**}	Set image aspect ratio.

APPENDIX A. SUMMARY OF VERSION 4 COMMAND CHANGES

This appendix summarizes four categories of GRIZ command changes: (1) eliminated commands; (2) commands introduced in Version 2 that have not been documented previously in this User Manual; (3) existing commands modified in Version 4; and (4) new commands in Version 4. Commands that fall in both categories (2) and (3) are listed with category (3).

The primary driver for modifications to *existing* commands derives from the evolution of GRIZ in Version 4 to handle Mili self-defining databases. The older TAURUS database allowed for a single set of nodes, an optional set of hex volume elements, an optional set of shell elements, and an optional set of beam elements. Under Mili, there can be an arbitrary number of these sets, called *classes*, of those objects as well as other element types and other non-element mesh objects. Thus, those Version 2 commands requiring a string argument such as “s” or “shell” to indicate that type of element now require a class name (technically, a class *short* name, since Mili provides for both a *long* name, or title, and a short name, which is meant to be brief and easy to type). For the user, this means learning to use class names in place of the old node or element-type specifiers (“n,” “node,” “h,” “brick,” “s,” “shell,” “b,” or “beam) and learning where to find class names. The response to the “tell class” query is a listing of class long and short names in the database written to the feedback window. The “Primal” pulldown result menu contains a submenu for each class in the database and exposes both the long and short class names. The submenu labels consist of the long name followed by the short name in parentheses. TAURUS databases are made to look like Mili databases, in this regard, by pre-defining the long and short names for the existing sets of data as if they were Mili classes.

The existence of two new result menus, **Derived** and **Primal**, in place of GRIZ Version 2’s single **Result** menu, is another manifestation of Version 4’s support for self-defining databases. All of Version 2’s results were derived, i.e., all result data passed through a framework for deriving results from raw or “primal” data even if in some instances no computation was performed and the rendered data was in fact primal. With self-defining databases, GRIZ can no longer be assured that the required inputs for its result derivations will be available, so new logic has been added to determine which results can be derived on a given database. Those derivations that can occur are made available under Version 4’s **Derived** menu. For many databases the contents of the **Derived** menu closely match Version 2’s **Result** menu.

Version 4's **Primal** menu exists to make all of the raw simulation results in a database available for post-processing. As analysis codes grow to make material-model specific state variables available for post-processing, the **Primal** menu will provide GUI-based access to them. As with class names, the GRIZ Version 4 user can see what primal and derived results are available by popping the menus or by executing a "tell results" query, in which a listing of all derived and primal results is written to the feedback window.

A.1 Eliminated commands

These commands are not available in Version 4. Alternatives are included in parentheses if available.

- **autorgb (autoimg)**
- **carpet**
- **mth**
- **on/off autorgb (on/off autoimg)**
- **on/off carpet**
- **on/off elnum** (**on/off num**)
- **on/off ndnum** (**on/off num**)
- **on/off thsm**
- **switch picsh/picbm** (**setpick**)
- **thsm**
- **vecjf**
- **veclf**

A.2 Previously Undocumented Commands

The following commands were introduced in GRIZ Version 2 and are new to the User Manual:

- **conwid** (page 51). Set width of isocontour lines.
- **coordxf** (page 34). Define alternative global coordinate system.
- **dir3n** (page 34). Define direction vector using three nodes.
- **dir3p** (page 34). Define direction vector using three points in space.
- **dirv** (page 34). Define direction vector components.
- **edgbias** (page 60). Set Z-direction position bias for rendering edges.

- **fracsz** (page 64). Set number of digits to render in some numeric quantities.
- **inrgb** (page 61). Read in an RGB image file.
- **minmov** (page 66). Set cursor motion threshold to discriminate picks from view transformations.
- **on/off autosz** (page 64). Toggle automatic determination of fraction sizes on plot axis annotations.
- **on/off bbmax** (page 65). Toggle accumulation of maximal bounding box extents.
- **on/off bgimage** (page 61). Toggle use of stored image as background in rendering window.
- **on/off coordxf** (page 34). Toggle use of transformation of some results in alternative global coordinate system.
- **on/off lighting** (page 62). Toggle use of lighting contribution to shading calculation.
- **on/off zlines** (page 65). Toggle depth-buffering of anti-aliased lines with respect to each other during rendering.
- **outref** (page 54). Save a reference surface definition.
- **resetmm** (page 35). Delete cached min/max for current result.
- **show pvmag** (page 31). New "Projected Vector Magnitude" result.
- **tellmm** (page 48). Report result min/max values over states.

A.3 Changed Commands

The following commands existed in Version 2 but have changed in either their syntax or semantics (or both) in Version 4.

- **bbsrc** (page 64). New syntax replaces "n," "h," "s," "b" mnemonics.
- **bufqty** (page 66). New syntax replaces "n," "h," "s," "b" mnemonics.
- **clrsel** (page 38). Optionally clears a single selection instead of all selections.
- **conv** (page 35). Minor change in that the syntax actually supports the idea of defaults as documented in the User Manual.
- **gather** (page 43). New syntax matches that of new command "plot."
- **hilite** (page 37). Takes class short name instead of previous node, brick, shell, or beam mnemonics.
- **on/off minmax** (page 43). No syntax change, but now has meaning in plot mode as well as mesh view mode.
- **outmm** (page 45). No syntax change, but new logic also breaks out nodal results by material.

•**outth** (page 42). New syntax matches that of new command “plot.”

•**select** (page 37). Takes class short name instead of previous node, brick, shell, or beam mnemonics.

•**setcol** (page 60). Several new targets for which color can be set, plus new syntax to return a target to its initial color.

•**show** (page 27). Syntax of result specifications has been expanded to accommodate vector and array primal results.

A.4 New Commands

•**autoimg** (page 24). Set file name root for automatic image saves during animations.

•**delth** (page 43). Delete time histories.

•**edgwid** (page 60). Set width of edges rendered under “on edges.”

•**exclude** (page 38). Combined “disable” and “invis” for materials.

•**glyphqty** (page 43). Set quantity of glyphs displayed on time series plots.

•**glyphscl** (page 43). Set size of plot glyphs.

•**include** (page 38). Combined “enable” and “vis” for materials.

•**jpegqual** (page 45). Set image quality of JPEG images.

•**minst** (page 23). Set minimum accessible state.

•**mmloc** (page 44). Set location of ymin/ymax values in plot window.

•**numclass** (page 37). Specify object classes for numbering.

•**on/off autoimg** (page 24). Turn on automatic image saves during animations.

•**on/off derived** (page 33). Toggle search of derived result table in result requests.

•**on/off glyphcol** (page 43). Toggle glyph colors between curve colors and foreground color.

•**on/off glyphs** (page 43). Toggle rendering of glyphs on plots.

•**on/off hex_overlap** (page 65). Toggle generation of redundant hex faces in face table.

•**on/off leglines** (page 44). Toggle inclusion of plot curve samples in legend.

•**on/off locref** (page 36). Toggle display of local reference frame on ‘select’ed shell elements.

•**on/off num** (page 37). Toggle display of object numbers on nodes and/or elements.

•**on/off particles** (page 58). Toggle display of particles.

•**on/off plotcol** (page 44). Toggle plot colors between individual curve colors and foreground color.

•**on/off plotcoords** (page 44). Toggle inclusion of cursor coordinates display in plot mode.

•**on/off plotgrid** (page 44). Toggle display of background grid in plot window.

•**on/off primal** (page 33). Toggle search of primal result table in result requests.

•**outjpeg** (page 45). Save current image as JPEG file.

•**outview** (page 46). Save current view transformation.

•**partrad** (page 58). Set particle radius.

•**plot** (page 40). Generate time series plot(s); supersedes “timhis” command.

•**resetimg** (page 24). Reset file counter for automatic image saves.

•**setpick** (page 66). Assign pick class to a mouse button.

•**stride** (page 23). Set step and animation stride.

•**surface** (page 49). Specify surface for generating traction forces.

•**switch glyphstag/glyphalign** (page 44). Toggle glyph positions as staggered or aligned.

•**tell** (page 46). Information queries.

•**traction** (page 49). Compute traction forces.

APPENDIX B Release Notes

Release 4.04

This new release of GRIZ contains a variety of bug fixes and ehnancements. Following is a summary of changes for this release:

New Commands:

(1) show damage: This command will hilite elements that meet an evaluation metric that is a function of four response qualtities. Elements that have fully failed will display in RED while non-damaged elements will be colored in BLUE.

show damage vel_dir <vx,vy,vz> vel_cutoff relVol_cutoff eps_cutoff

SCR#: 284 - Related SCR: 285

(2) on damage_hide: All elements that are damaged will not be be displayed - They will be removed from the display upon failure.

off damage_hide: All damaged elements will be re-displayed.

SCR#: 285

(3) Material Range specification: When entering a list of materials, it is now valid to list a range as ’n-n’ : Example: mat 1 2 3-10 12

SCR#: 279

Bug Fixes:

(1) Elements with zero width were causing GRIZ to crash. This error has been corrected.

SCR#: 280

(2) When running from a history file using the ’rdhis’ command, any embedded animate command would exit immediately. This has been corrected so that animate can be included in a history file.

SCR#: 280

Release 4.05

This new release of GRIZ, version 4.05, contains a variety of bug fixes and ehnancements. Following is a summary of changes for this release:

New Commands:

(1) Extreme min/max: This command will display the current results extreme min or max for all time steps. Example: 'extreme_min' OR 'extreme_max'.

SCR#: 292

(2) Free Nodes: A new option has been added to Griz to display all nodes that have seperated from their elements, or are free nodes.

Example: on fn [scale 9.9] [res 5] [mass_scale on|off] [vol_scale on|off] [mat 1,2...]

where: scale - Scaling factor to control nodes size.

res - Resolution to render nodes as spheres - higher number will make better looking nodes, but increase rendering time.

mass_scale - Nodes will be scaled by mass of contributing elements. Default is on.

vol_scale - Nodes will be scaled by volume of contributing elements. Default is off.

mat - Free nodes will be displayed for a selected list of materials.

(3) Griz Version: A new command line option has been added to Griz to display detailed info about the version of Griz that you are running:

Example: griz4s -V

SCR#: 299

(4) Hidden Line Removal: The capability to generate high qualitypostscript files will hidden line removal has been moved directly into Griz - User no longer have to run the hidden utility. When the outhid command is executed, a postscript file is produced.

Example: outhid

SCR#: 289

(5) Running Beta Version of Griz: A feature has been added to Griz that will allow a user to run the latest version of Griz or the Beta Version.

Example: griz4s -beta -i input

SCR#: 298

New Features:

(1) Griz will now compute strain results for time-history databases. The list of selected objects will now appear in the startup window. For example:

Class: node

block[1]: 1-19

Class: brick

block[1]: 1-10

block[2]: 150-180

SCR#: 291

(2) Griz will now allow input of a range of objects for the select command. For example:

instead of: select brick 1 2 3 4 5 6 7 8 9 10

you may enter: select brick 1-10

SCR#: 308

APPENDIX C Change Pages

Release 4.04

(1) show damage -	Page (29)
(2) damage hide -	Page (36)
(3) Material range specification -	Page (38)

Release 4.05

(1) Extreme Min/Extreme Max-	Page (35)
(2) Select range specification	Page (38)
(3) FreeNodes-	Page (39)

INDEX