GRAFLAB 2.3 for UNIX
A MATLAB Database, Plotting, and Analysis Tool

User's Guide

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

William N. Dunn

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories
DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.
Abstract

This report is a user's manual for GRAFLAB, which is a new database, analysis, and plotting package that has been written entirely in the MATLAB programming language. GRAFLAB is currently used for data reduction, analysis, and archival. GRAFLAB currently runs on UNIX platforms supported by MATLAB. GRAFLAB was written to replace GRAFAID, which is a FORTRAN database, analysis, and plotting package that runs on VAX/VMS.
## Contents

1.0 Introduction .................................................................................................................. 7  
   1.1 Prepare to Use GRAFLAB ......................................................................................... 7  
1.2 Update the Database ..................................................................................................... 8  
   1.2a Update GRAFLAB database from version 1.0 to 2.3 ............................................ 8  
   1.2b Update GRAFLAB database from version 2.0 to 2.3 ............................................ 8  
1.3 Improvements in GRAFLAB Version 2.3 ................................................................... 9  
2.0 Basic GRAFLAB Usage ................................................................................................. 10  
   2.1 Create and save a curve from within GRAFLAB .................................................... 10  
   2.2 Load external files and save them into GRAFLAB .................................................. 10  
   2.3 Load a Neutral File into GRAFLAB ......................................................................... 12  
   2.4 Write a Neutral File from GRAFLAB ....................................................................... 12  
   2.5 Generic ASCII File (with Headers/Text) Input ....................................................... 12  
   2.6 GRAFLAB Curve Input ............................................................................................ 12  
   2.7 GRAFLAB Utilities ................................................................................................. 13  
3.0 The GRAFLAB Database .............................................................................................. 14  
   3.1 Database Design ...................................................................................................... 14  
   3.2 Database Structure ................................................................................................. 14  
   3.3 Database m-files ..................................................................................................... 15  
   3.4 CURVPARM – The System Dependent Function ..................................................... 16  
4.0 Plotting & Curve Manipulation ..................................................................................... 17  
   4.1 Useful SETD and SHOWD commands ................................................................... 17  
   4.2 Comprehensive SETD and SHOWD command listing .......................................... 23  
      4.2.1 Plotting Commands ....................................................................................... 23  
      4.2.2 Curve Header Commands ............................................................................. 26  
      4.2.3 QA Commands ............................................................................................ 26  
   4.3 GRAFLAB Globals ................................................................................................. 27  
      4.3.1 Useful Commands ....................................................................................... 27  
      4.3.2 Plotting Globals ......................................................................................... 27  
      4.3.3 Calculation Globals .................................................................................... 28  
      4.3.4 Other Globals ............................................................................................ 29  
5.0 GRAFLAB m-files ........................................................................................................ 30  
   5.1 General Notes on GRAFLAB m-files .................................................................... 30  
   5.2 Analysis m-files ...................................................................................................... 30  
       5.2.1 Mechanical Shock & Vibration (gmechanical) ............................................. 31  
       5.2.2 Thermal Data Reduction Utilities (gthermal) ............................................ 34  
   5.3 Database m-files ..................................................................................................... 34  
       5.3.1 Database – User m-files ............................................................................ 34  
       5.3.2 Database – GRAFLAB m-files for programmers ..................................... 36  
   5.4 File I/O m-files ....................................................................................................... 37  
   5.5 Plotting m-files ...................................................................................................... 37  
   5.6 String Manipulation m-files ................................................................................... 37  
   5.7 GLOBAL m-files .................................................................................................... 39  
   5.8 Other m-files ......................................................................................................... 39  
6.0 Programmer’s Guide ..................................................................................................... 41  
   6.1 Functions that Call GRAFLAB System Routines .................................................... 41
6.2 Add/Remove a Global Variable to GRAFLAB .......................................................... 43
6.3 GRAFLAB 23 Test Area ......................................................................................... 44
6.4 Create a New GRAFLAB function ......................................................................... 44
7.0 MATLAB Memory Management ........................................................................... 45
8.0 GRAFLAB Tips ...................................................................................................... 45
Appendix A .................................................................................................................. 47

Illustrations

Figure 1. GRAFLAB Database Structure ........................................................................ 15
Figure 2. Plot Commands by Location ......................................................................... 21
Figure 3. Plot Commands by Location ......................................................................... 22
Figure 4. GRAFLAB Utility Directory Tree .................................................................... 31

Tables

Table 1. GRAFLAB Routines calling SYS.M ................................................................. 41
1.0 Introduction

GRAFLAB is a database, analysis, and plotting package written entirely in MATLAB, designed to replace the outdated GRAFAID code. GRAFLAB is a superset of commands that works on top of the MATLAB matrix manipulation software. The user of GRAFLAB has the option of using either MATLAB routines or the added GRAFLAB routines to manipulate data.

The GRAFLAB routines are stored in subdirectories in the MATLAB path, and the GRAFLAB database is located under the current directory in a subdirectory called alldata. The database, analysis, and plotting routines are all located in several common directories that exist in the MATLAB path statement.

This report is a condensed tutorial of those commands that perform 80% of the GRAFLAB tasks. To get more extensive help concerning a specific command.m file, simply type in

```
help command
```

All GRAFLAB commands that are typed by the user, or output by the computer, are displayed in this manual in the Courier font as follows:

```
command(input);
```

1.1 Prepare to Use GRAFLAB

First, to start a MATLAB session that runs GRAFLAB 2.3, the startup.m file, which can be obtained from the code sponsor, must be placed in your /home/username/matlab directory. A sample startup.m file is shown in Appendix A.

Next, open the directory in which you want to run GRAFLAB and create a new subdirectory called alldata. It is in this subdirectory that the GRAFLAB database will store the curves and the information related to them.

Finally, type MATLAB at the UNIX prompt. On a UNIX platform, if the following output is displayed, the database is open.

```
..........................................
Welcome to GRAFLAB version 2.3U
     for UNIX
..........................................
```

However, if the following warning is displayed, the ./alldata directory may have been entered improperly.

---


GRAFLAB 2.3
At the MATLAB prompt, type

```matlab
>>!mkdir ./alldata
```

### 1.2 Update the Database

If you start up GRAFLAB in a directory that contains an old database (GRAFLAB 1.0 or GRAFLAB 2.0), the startup.m file for GRAFLAB 2.3 will call gl2update to update the Header.mat file for each curve to the current GRAFLAB format. This is an irreversible process that will cause you problems if you try to go back to running an older version of GRAFLAB on this new database format. The most recent versions of GRAFLAB (2.1, 2.2 and 2.3) all use the same database format.

#### 1.2a Update GRAFLAB database from version 1.0 to 2.3

When a version 1.0 database is encountered, the user is given a choice to update from version 1.0 to version 2.3 of the database and is warned that the process is irreversible.

```
***** WARNING: GRAFLAB 1.0 DATABASE ENCOUNTERED **********
```

Updating the database is irreversible!!
Do you wish to update the database to version 2.3a ?
1. Yes
2. No
Enter 1 or 2

>1

Updating ./alldata/PPCYTC1/Header.mat
Updating ./alldata/PPCZTC1/Header.mat
Updating ./alldata/PPGZTC1/Header.mat
Updating ./alldata/PRDXTC1/Header.mat
Updating ./alldata/PRDYTC1/Header.mat
Updating ./alldata/PRDZTC1/Header.mat
Updating ./alldata/PTSXTC1/Header.mat
Updating ./alldata/PTSYTC1/Header.mat

Entire GRAFLAB database updated to GRAFLAB 2.3a

#### 1.2b Update GRAFLAB database from version 2.0 to 2.3

When a version 2.0 database is encountered, the user is given no choice, and the update is automatic (and minor). The curvename is removed from the Header.mat, and the curve legend is updated to include only the legend, not the curvename.

```
**** Minor header update necessary for version 2.3a
```

Updating ./alldata/ah25_10/Header.mat
Updating ./alldata/dh25_1/Header.mat
Updating ./alldata/sh25_1/Header.mat
Updating ./alldata/shghav25/Header.mat
Updating ./alldata/vh25_1/Header.mat
Updating ./alldata/y/Header.mat

Entire GRAFLAB database updated to GRAFLAB 2.3

1.3 Improvements in GRAFLAB Version 2.3

GRAFLAB version 2.3 retains the functionality of previous versions, with many added enhancements. Your m-files should run as they did previously with only minor modifications.

The major differences between version 2.3 and 1.0 follow:

1. GRAFLAB 2.3 uses only MATLAB; that is, no MEX files (C code).
2. GRAFLAB 2.3 uses no master list of the database files. The database is now simply those subdirectories that appear under ./alldata. Each curvename is the subdirectory name and each subdirectory must contain a Header.mat, which contains header information, and a Data.mat, which contains the data.
3. GRAFLAB 2.3 now allows 31 character curvenames.
4. The complexity of the database has been reduced:
   a. While GRAFLAB 1.0 used 140 m file (many of which were written in C, version 2.3 uses less than 40, all of which are written in MATLAB.
   b. Several GRAFLAB 1.0 routines were rewritten for speed and simplicity.
5. The utility function, showd, now returns an output argument (nargout).
6. The utility function, gcopy, enables copying of curvename subdirectories from remote databases.
7. A faster, more powerful gdir was developed for parsing curvenames.
8. Plot legends (and grids) now work in bplot using new globals.
9. Code is more easily portable to other platforms.
10. Plotting of complex numbers is enabled.
2.0 Basic GRAFLAB Usage

This section provides a basic tutorial on the use of GRAFLAB. However, the user is encouraged to become familiar with the basic MATLAB syntax and command structure before attempting to use GRAFLAB. Then, having mastered the basics of MATLAB, GRAFLAB will seem much more intuitive.

GRAFLAB makes extensive use of the MATLAB concepts of global variables, a local workspace, and the ability to save and load data from binary .mat files stored on disk. Therefore, it is important for the user to understand that while most analyses will be conducted exclusively in the MATLAB local workspace, GRAFLAB stores numerous parameters needed to define the user environment (Section 4.0) as global variables. In addition, when the user retrieves curves from GRAFLAB's alldata database (using the setd('act') command), these curves are maintained within GRAFLAB as global variables.

2.1 Create and save a curve from within GRAFLAB

To create and save a GRAFLAB curve, first assign values to a MATLAB variable, zzz,

```matlab
zzz=[1 2 ; 3 4 ; 5 6]
```

Define the following plot parameters: plot title (pt), the y label (pyl), the x label (pxl), the plot extremes (pe), and the axis type (at), using the GRAFLAB command setd. GRAFLAB assigns default values for all the plot parameters except the plot title. These can be checked using the showd command.

```matlab
setd('pt','A sample plot/Includes 5 lines of info/line 3/line4/line5')
setd('pyl','Y axis description here')
setd('pxl','X axis description here')
setd('pe','0/10/0/10')
setd('at','nolog')
```

Since GRAFLAB curves are stored as global variables, the user must declare a new curve to be a global variable before it can be saved.

```matlab
global zzz
savecurf('zzz',0,6)
```

The curve zzz is now a part of your GRAFLAB database, containing both a header and data. Use gdir to view the database contents.

2.2 Load external files and save them into GRAFLAB

ASCII files can be loaded easily into GRAFLAB. If the files are columns of comma or tab delimited numbers in ASCII format, MATLAB has its own load command. First, load in the data. For example, an ASCII data set (CHAN1.DAT) with 2 columns (column 1 is the time, and column 2 is the acceleration in g's) will be loaded.
load CHAN1.DAT -ascii;

Then, when the data is loaded, perform a 

whos

The variable CHAN1 and its dimensions (127,998 rows and 2 columns, in this case) will display.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Elements</th>
<th>Bytes</th>
<th>Density</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAN1</td>
<td>127998 by 2</td>
<td>255996</td>
<td>2047968</td>
<td>Full</td>
<td>No</td>
</tr>
</tbody>
</table>

Now, plot column 2 versus column 1.

plot(CHAN1(:,1),CHAN1(:,2))

The data can be truncated, taking the first 100,000 pairs.

CHAN1=CHAN1(1:100000,:);

**Note.** Do not omit the semicolon (;) that is located at the end of the path, or the data (100,000 pairs of points!) will scroll to the screen.

Now, save the data in CHAN1 to curve name "chan1," load and save CHAN2.

chan1=CHAN1;
global chan1
setd('pt','Channel 1')
setd('pyl','G's')
setd('pxl','time (sec)')
setd('pe','0/.5/-2500/2500')
setd('at','nolog')
savecurf('chan1',0,6)
load CHAN2.DAT -ascii;
chan2=CHAN2;
global chan2
setd('pt','CHANNEL 2')
savecurf('chan2',0,6)

Now there are three curves in the database. Look at the names using gdir

gdir
  1. chan1
  2. chan2
  3. zzz
Now we can use

\begin{verbatim}
showd('ch','chan1')
end('act','chan*')
setd('pld','on')
setd('pg','on')
setd('plco','red/blue')
\end{verbatim}

To show the chan1 header values.
To put the curves into the active set.
To display the legends of the two curves.
To set the grid on.
To set the colors.
To plot the curves.

The database is saved to disk in ./alldata as subdirectories. The utility gdir is the key to accessing the database. Perform a help gdir, and carefully read the help in this important utility.

2.3 Load a Neutral File into GRAFLAB

Neutral files are formatted ASCII text data files. To load a neutral file into GRAFLAB, use the following command:

\begin{verbatim}
ninput('./directory/filename,neu',overwrite,grafaidflag)
\end{verbatim}

where overwrite = 1 if you want to overwrite existing curvenames in your current database with the curves read from filename.neu.

The grafaidflag identifies when a neutral file was written by GRAFAID (as compared to one written by GRAFLAB or some other source). This is important to know because, for some reason, GRAFAID neutral files omit the 'e' when writing out data in exponential format (for example 2.0e-3 is written 2.0-3). Unfortunately, GRAFLAB does not interpret this properly. The grafaidflag invokes a routine that accounts for the missing 'e' when loading in the data.

2.4 Write a Neutral File from GRAFLAB

To write a Neutral file from GRAFLAB, use the following command:

\begin{verbatim}
noutput('./directory/filename.neu','curvename')
\end{verbatim}

GRAFLAB will prompt you to select the option to dump the curves that match gdir('curvename') from the database, and then to enter a neutral file title.

2.5 Generic ASCII File (with Headers/Text) Input

Use the GRAFLAB gfile utility to import ASCII files that contain both data and headers in formats other than neutral files.

2.6 GRAFLAB Curve Input

Use the GRAFLAB gcopy utility to copy GRAFLAB curves from one GRAFLAB database to another.
2.7 GRAFLAB Utilities

GRAFLAB utilities perform useful transforms on the input arguments to generate output. For example, the utility `gpsd.m` generates a power (or acceleration) spectral density, `pxx`, from a user specified acceleration history, `x`. For more information, type `help gpsd`.

**syntax:**

```
pxx=gpsd(x,blocksize,overlap,timerange,window)
```

The variable `x` is the input acceleration history. As is typical of most GRAFLAB utilities, if 'x' is missing, or null, `gpsd` will use the first curve in the active set. Similarly, any other argument will either have a designated default value or will require the user to provide values before the analysis can proceed. Arguments can be either numeric or text as dictated by the utility in question. Empty brackets `[]`, denote that the user wishes to use the designated GRAFLAB default value.

An example input is shown below:

```
[yenv]=gpsd([],1024,50,[],2);
```
3.0 The GRAFLAB Database

3.1 Database Design

The database design is simple yet functional. All of the curves are stored as subdirectories in the alldata subdirectory. The name of each subdirectory is the name of the curve. The header information, the data, and additional auxiliary information for a given curve is located under that subdirectory. Data is accessed by setting it active using the setd('act','curvename') command.

3.2 Database Structure

The GRAFLAB 2.3 database is shown in Figure 1. Each curve has a curvename, which is merely a subdirectory under the ./alldata directory. The curvename subdirectory contains two or three MATLAB binary *.mat files: Header.mat and Data.mat, and optionally, an Aux.mat file. Header.mat contains 16 variables, which define primarily the plot parameters of curvename. The curvename length (number of characters) is a system dependent value equal to the variable CL. This length is a minimum of 39 characters. Data.mat contains the data in a variable that has the corresponding curvename. Aux.mat, although not fully implemented, contains auxiliary data for special uses.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>showd('ch','curvel')</td>
<td>Displays the relevant Header.mat variables for curvel.</td>
</tr>
<tr>
<td>showd('cd','curvel')</td>
<td>Displays the data stored in Data.mat for curvel.</td>
</tr>
</tbody>
</table>
Figure 1. GRAFLAB Database Structure

3.3 Database m-files

The GRAFLAB database m-files are located in the directory:
/disk1/graflab23

These files are informally divided into two categories: user m-files and programmer m-files. For a brief description of each of these m-files, refer to sections 5.3.1 (user m-files) and 5.3.2 (programmer m-files).
3.4 CURVPARM – The System Dependent Function

The function curvparm.m defines the globals that control the database filenames, the system dependent delimiters, and max curve name length, etc. These globals can be accessed using the curvparm command. For example,

\[
\text{curvparm('DIR\_PATH')} \rightarrow \text{returns 'alldata/' for UNIX.}
\]

<table>
<thead>
<tr>
<th>SYSTEM Globals</th>
<th>(Defined in curvparm.m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'NL'</td>
<td>System defined Newline Character.</td>
</tr>
<tr>
<td>'TRUE'</td>
<td>= 1</td>
</tr>
<tr>
<td>'FALSE'</td>
<td>= 0</td>
</tr>
<tr>
<td>'DIR_PATH'</td>
<td>Default Curve Directory Path.</td>
</tr>
<tr>
<td>'DIR_SEP'</td>
<td>System defined Directory Separator.</td>
</tr>
<tr>
<td>'NAME_MAX'</td>
<td>Maximum File Name Length.</td>
</tr>
<tr>
<td>'PATH_MAX'</td>
<td>Maximum Path Name Length.</td>
</tr>
<tr>
<td>'MACHINE_ID'</td>
<td>System Machine Identification.</td>
</tr>
<tr>
<td>'DIR_NAME'</td>
<td>Default Curve Names File.</td>
</tr>
<tr>
<td>'NEU_NAME'</td>
<td>Default Neutral Names File.</td>
</tr>
<tr>
<td>'HEAD_NAME'</td>
<td>Default Header File Name.</td>
</tr>
<tr>
<td>'AUX_NAME'</td>
<td>Default Auxiliary File Name.</td>
</tr>
<tr>
<td>'DATA_NAME'</td>
<td>Default Data File Name.</td>
</tr>
<tr>
<td>'SATADEF_NAME'</td>
<td>Default Satadef File Name.</td>
</tr>
<tr>
<td>'CNAME_LENGTH'</td>
<td>Curve Name Maximum Length.</td>
</tr>
<tr>
<td>'ENTRY_LENGTH'</td>
<td>Maximum Entry String Length.</td>
</tr>
<tr>
<td>'START_ENTRIES'</td>
<td>Initial # of Curve Entries in Curve File.</td>
</tr>
<tr>
<td>'EXT_ENTRIES'</td>
<td>Extend Curve Entry Increment.</td>
</tr>
<tr>
<td>'FILE_ID'</td>
<td>Standard File ID &amp; Version.</td>
</tr>
<tr>
<td>'HAUX'</td>
<td>Haversine Auxiliary File Type.</td>
</tr>
<tr>
<td>'SAUX'</td>
<td>Decayed Sine Auxiliary File Type.</td>
</tr>
</tbody>
</table>
4.0 Plotting & Curve Manipulation

Plotting and curve manipulation parameters are controlled by the \texttt{setd} and \texttt{showd} commands, which modify the GRAFLAB globals. Because most of the GL globals deal with plotting, globals are discussed in this section.

4.1 Useful SETD and SHOWD commands

The \texttt{setd} and \texttt{showd} commands are used to assign values to global variables and to echo those values, respectively. SETD and SHOWD commands fall into two main categories: active set environment, and curve environment.

The first category includes those commands associated with the environment for the active (or plot) set. These commands are generally intended to define the active set and to tailor the look of the active set plot (\texttt{bplot}). These commands have the form \texttt{setd('p*','option')}, where the '*' represents one or two additional characters which, along with the 'p', form the acronym for the desired command.

The second category includes those commands associated with the environment for specific curves. These commands have the form \texttt{setd('c*','curvename','option')}, where the '*' represents one or two additional characters, which along with the 'c', form the acronym for the desired command. Many curve set commands have analogous plot set commands.

Some of the most common setd and showd commands, followed by a comprehensive listing, are shown below. Figures 2 and 3 summarize many of the plotting commands in graphical form.

\textbf{SETD} – Defines the GRAFLAB working environment. Screen output is often provided as an echo of the parameter that has just been set. Some useful commands follow:

\begin{itemize}
  \item \texttt{setd('act',curvename)} – Used to put a curve in the active set.
    \texttt{setd('act','PA*')} will put all the curves matching the pattern into the active set. \texttt{setd('act','PAGPSX','add')} will add the curvename to the existing active set.
  \item \texttt{setd('af',option)} – Sets the axis format where option is as follows:
    \begin{itemize}
      \item \texttt{TICAXES} Produces tics only on X and Y axes.
      \item \texttt{TDASH} Produces tics on all sides at minor tic divisions and dashed grids at the major tic divisions.
      \item \texttt{TDOT} Produces tics on all sides at minor tic divisions and dotted grids at the major tic divisions.
      \item \texttt{TLINE} (DEFAULT) produces tics on all sides at minor tic divisions and solid grids at the major tic divisions.
      \item \texttt{GRID} Produces solid grid lines.
      \item \texttt{DASH} Produces dashed grid lines.
      \item \texttt{PLAIN} Produces no tics or grids.
    \end{itemize}
\end{itemize}
setd('at', option) - Sets the axis type. This can also be set permanently for individual curves using

setd('cat', curvename, option) where option is

- NOLOG  Linear/Linear Axes
- XLOG   X Log/Y Linear Axes
- YLOG   X Linear/Y Log Axes
- XYLOG  Log/Log Axes
- POLAR  Polar Axes

setd('cl', curvename, newlegendtext) - Sets the curve legend.

setd('cn', curvenamel, curvename2) - Renames curve.

setd('ct', curvename, titletext) - Sets the curve title.

setd('cyl', curvename, text) - Sets the curve y label.

setd('cxl', curvename, text) - Sets the curve x label.

setd('pe', text) - Sets the plot extremes, which can be a text string 'xmin/xmax/ymin/ymax', or they can be a vector.

[xmin xmax ymin ymax]

setd('pld', text) - Sets the plot legend display.

Text can be 'on', 'onl', or 'off'

setd('pld', 'on') - Displays the legend if numcurves active > 1.

setd('pld', 'onl') - Displays the legend if numcurves active >= 1.

setd('pld', 'off') - Does not display the legend.

setd('pll', option) - Sets the plot legend location in percent of the screen. The variable option can be a string '0.3/0.6', or a vector [0.3 0.6]. The variable option can also be 'AUTO', in which case MATLAB will reliably auto locate the legend box to avoid covering the data.

Caution: This procedure may require 15 to 20 seconds on an HP735 for MATLAB to calculate a good position if you are plotting several hundred thousand points.

setd('plco', text) - Sets the plot line colors. The variable text can be 'YELLOW/MAGENTA/CYAN/RED/GREEN/BLUE/WHITE/BLUE'.

setd('plto', text) - This sets the sequence of the dashes used to plot each curve. Text can be 'solid/dashed/dashdot/dotted/none'.

setd('plto', 'solid/dashed/dashdot/dotted/none')
setd('pls',text) – This command does not appear to work, however, it is supposed to allow you change the symbols to the Lucky Charms marshmallow shapes. The allowable symbols are ‘.ox+*’

setd('plt',text) – This allows you to set the permutations of the plot lines:

'\text{DASH}'/'\text{DSYM}'/'\text{LINE}'/'\text{LSYM}'/'\text{SYMB}'/'\text{NSYM}'
/'\text{NUL}'/'\text{SYM}'/'\text{D:NUL}'/'\text{D:SYM}'

setd('plw',text or num) – Sets the line width. Seems to have a "step" effect for bitmap and screen display, but has a nice gradual effect for vector rendering formats such as POSTSCRIPT. This feature is nice for viewgraphs.

setd('pn',text) – Sets the plot name over the top of the figure window.

setd('pp',text) - Sets the figure position.

setd('pt',text) – Sets the plot title.

setd('pxl',text) – Sets the plot x label.

setd('pyl',text) – Sets the plot y label.

setd('qa',text) – Sets the qa display ('on' or 'off').

setd('qad',text) – Sets the qad display ('on' or 'off').

SHOWD – Displays current GRAFLAB working environment. The showd command provides output if an output argument is included. Data input that can be set using setd, can be shown using showd. Some useful commands follow:

SHOWD permits the use of an output argument, nargout, having the following form:

nargout = showd('*',option)

In each case, the output argument provides useful data to the user for the particular SHOWD command being invoked. For example:

x=showd('cd',curvel);

populates ‘x’ with a copy of the curve data contained in curvel).

showd('act') – Shows the curvenames in the active set.

showd('cd',curvename,range) - Echoes to the screen the curve data contained in the string curvename over the span of points range. If you have an acceleration history over curve called 'ACC' and you want to put the data from time = 0.2 to 3.2 seconds in a variable named y, the following command would do it:

y=showd('cd','ACC','0.2/3.2')
showd('ct', curvename) – Echoes the curve title. For wildcard viewing of multiple curve titles using wildcards use
showd('ct', gdir('PTSZ*'))

showd('ch', curvename) – Echoes the curve header for curvename. This is useful if you need more than one piece of information about a curve.

showd('users') – Shows the users, their process id, and the CPU time. Also can use who_mat. This command is useful because MATLAB occasionally fails to relinquish licenses after a process is inadvertently killed.
Figure 2. Plot Commands by Location
**PLOT CONTROLS**

- **Background color** - `setd('BGC','BLACK')` / 'WHITE'
- **Plot extremes** - `setd('pe',0/10/1e-5/0.1)`
- **Line colors** - `setd('plco','RED/BLUE')` 'YELLOW/MAGENTA/CYAN/RED/GREEN/BLUE/WHITE/BLACK'
- **Line style** - `setd('plto','SOLID/DASHED')` 'DOTTED/DASHDOT'
- **Symbols** - `setd('pls', 'o', 'x', '+', '*')`
- **Line type** - `setd('plt','LSYM')`
- **Line width** - `setd('lw',1.5)`
- **Plot name** - `setd('pn','Plot name')`
- **Plot location** - `setd('pp',[49 99 20 70])`
- **X label text** - `setd('pxl','Plot x label')`
- **Y label text** - `setd('pyl','Plot y label')`

---

**Thermocouple Channel 15 raw data**

**Horizontal Calorimeter West Tower Flame Temperature**

**Half Pack SNL Area III Burn Test March 4, 1997**

![Plot Commands by Location](image)

Figure 3. Plot Commands by Location
Note that showd returns an nargout value that can be used.

To get the data stored in curve1 and store it in y,

\[ y = \text{showd('cd','curve1');} \]

To set the curve x label of curve2 to the same as curve1, type

\[ \text{setd('cxl','curve2',showd('cxl','curve1'));} \]

There is a considerable amount of data processing power using gdir and a MATLAB for loop. To set the curve x label of all the curves named Curve* to the same as curve1,

\[ a = \text{gdir('Curve*');} \]

\[ \text{for } i=1: \text{size(a,1)} \]

\[ \text{setd('cxl',clearaws(a(i,:)),showd('cxl','curve1'));} \]

\[ \text{end} \]

**4.2 Comprehensive SETD and SHOWD command listing**

### 4.2.1 Plotting Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbrev.</th>
<th>Description / Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE SET</td>
<td>(ACT)</td>
<td>Puts a list of curves into the active set and puts their names into the global activecurves. Use setdloc to bring the data into the local workspace named according to the curvename.</td>
</tr>
<tr>
<td>AXIS FORMAT</td>
<td>(AF)</td>
<td>ticaxes – dotted grid lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tdash – dashed grid lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tline – solid grid lines</td>
</tr>
<tr>
<td>AXIS LABEL ANGLE</td>
<td>(ALCA)</td>
<td>Normal –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Italic –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oblique –</td>
</tr>
<tr>
<td>AXIS LABEL SIZE</td>
<td>(ALCS)</td>
<td>Size in Points (1/72 inch)</td>
</tr>
<tr>
<td>AXIS LABEL WEIGHT</td>
<td>(ALCW)</td>
<td>Light –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Normal –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Demi –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bold –</td>
</tr>
<tr>
<td>AXIS SCALE</td>
<td>(AS)</td>
<td>own –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>auto –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>full –</td>
</tr>
<tr>
<td><strong>AXIS TYPE (AT)</strong></td>
<td>NOLOG</td>
<td>XYLOG</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>AXIS X LABEL TYPE (AXL)</strong></td>
<td>XBOOTH – Displays both alpha &amp; numeric.</td>
<td>XNUMERIC – Displays only numbers.</td>
</tr>
<tr>
<td><strong>AXIS Y LABEL TYPE (AYL)</strong></td>
<td>XBOOTH – Displays both alpha &amp; numeric.</td>
<td>XNUMERIC – Displays only numbers.</td>
</tr>
<tr>
<td><strong>BACKGROUND COLOR (BGC)</strong></td>
<td>Black</td>
<td>White</td>
</tr>
<tr>
<td><strong>FONT TYPE (FT)</strong></td>
<td>Helvetica</td>
<td>Courier</td>
</tr>
<tr>
<td><strong>PLOT EXTREMES (PE)</strong></td>
<td>Sets the <code>xmin/xmax/ymin/ymax</code> for plotting.</td>
<td></td>
</tr>
<tr>
<td><strong>PLOT GRID (PG)</strong></td>
<td>Sets the grid display on or off.</td>
<td>ON</td>
</tr>
<tr>
<td><strong>PLOT LEGEND DISPLAY (PLD)</strong></td>
<td>Sets the display of the Plot Legend on or off.</td>
<td>ON - Will display if more than one curve.</td>
</tr>
<tr>
<td><strong>PLOT LEGEND FONT SIZE (PLFS)</strong></td>
<td>Sets the plot legend font size.</td>
<td>Acceptable values are from 8 to 24</td>
</tr>
<tr>
<td><strong>PLOT LEGEND LOCATION (PLL)</strong></td>
<td>Sets the location of the plot legend in percent of the plot screen.</td>
<td><code>[.6 .3]</code> – 60% along x, 30% along y.</td>
</tr>
<tr>
<td><strong>PLOT LINE COLORS (PLCO)</strong></td>
<td><code>y = 'YELLOW';</code></td>
<td><code>m = 'MAGENTA';</code></td>
</tr>
<tr>
<td><strong>PLOT LINE ORDER (PLTO)</strong></td>
<td>Determines whether the plots will be dashed, solid, dot-dash, etc.</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setd('plto','solid/solid')</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT LINE SYMBOLS (PLS)</strong></th>
<th>'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'o'</td>
</tr>
<tr>
<td></td>
<td>'x'</td>
</tr>
<tr>
<td></td>
<td>'+'</td>
</tr>
<tr>
<td></td>
<td>'*'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT LINE TYPE (PLT)</strong></th>
<th>DASH—Dashed no symbols.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSYM—Dashed with symbols.</td>
</tr>
<tr>
<td></td>
<td>LINE—Solid lines with no symbols.</td>
</tr>
<tr>
<td></td>
<td>LSYM—Solid lines with symbols.</td>
</tr>
<tr>
<td></td>
<td>SYMB—Same as LSYM.</td>
</tr>
<tr>
<td></td>
<td>NSYM—Symbols only.</td>
</tr>
<tr>
<td></td>
<td>:NUL—Dotted plot lines nosymbols.</td>
</tr>
<tr>
<td></td>
<td>:SYM—Dotted plot lines with symbols.</td>
</tr>
<tr>
<td></td>
<td>D:NUL—Dashdot Plot Lines no symbols.</td>
</tr>
<tr>
<td></td>
<td>D:SYM—Dashdot Plot lines with symbols.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT LINE WIDTH (PLW)</strong></th>
<th>Sets the line width of the curves in Numeric Points (1/72 inch).</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT NAME (PN)</strong></th>
<th>Sets the name of the Figure Window.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT POSITION (FIGURE) (PP)</strong></th>
<th>Sets the location of the Figure Window.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT TITLE (PT)</strong></th>
<th>Sets the plot title.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE1/LINE2/LINE3/LINE4/LINES</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT X LABEL (PXL)</strong></th>
<th>Sets the plot x axis label.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>PLOT Y LABEL (PYL)</strong></th>
<th>Sets the plot y axis label.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE CHAR ANGLE (TCA)</strong></th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Italic</td>
</tr>
<tr>
<td></td>
<td>Oblique</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE CHAR SIZE (TCS)</strong></th>
<th>Size in Points (1/72 inch)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE CHAR WEIGHT (TCW)</strong></th>
<th>Light</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Demi</td>
</tr>
<tr>
<td></td>
<td>Bold</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE HORIZONTAL LOCATION (THL)</strong></th>
<th>Left</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Center</td>
</tr>
<tr>
<td></td>
<td>Right</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE JUSTIFICATION (TJ)</strong></th>
<th>Left</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Center</td>
</tr>
<tr>
<td></td>
<td>Right</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TITLE VERTICAL LOCATION (TVL)</strong></th>
<th>Sets the plot title vertical location.</th>
</tr>
</thead>
</table>
### 4.2.2 Curve Header Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbrev.</th>
<th>Description / Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURVE AXIS TYPE</td>
<td>(CAT)</td>
<td>NOLOG – XYLOG – YLOG – XLOG – POLAR –</td>
</tr>
<tr>
<td>CURVE DATA</td>
<td>(CD)</td>
<td>Echoes the curve data to the screen or to nargout.</td>
</tr>
<tr>
<td>CURVE EXTREMES</td>
<td>(CE)</td>
<td>Sets the 'xmin/xmax/ymin/ymax' for plotting.</td>
</tr>
<tr>
<td>CURVE LEGEND</td>
<td>(CL)</td>
<td>Sets the curve legend to a string to be used in plotting.</td>
</tr>
<tr>
<td>CURVE NAME</td>
<td>(CN)</td>
<td>Renames the curve. setd('cn','curve1','curve2')</td>
</tr>
<tr>
<td>CURVE TITLE</td>
<td>(CT)</td>
<td>Sets the curve title setd('ct','curvename','LINE1/LINE2/LINE3/LINE4/LINE5')</td>
</tr>
<tr>
<td>CURVE X LABEL</td>
<td>(CXL)</td>
<td>Sets the x label of the curve.</td>
</tr>
<tr>
<td>CURVE X TYPE</td>
<td>(CXT)</td>
<td>X axis data type. MONO - Monotonically increasing data. NONM - Non monotonic data.</td>
</tr>
<tr>
<td>CURVE Y LABEL</td>
<td>(CYL)</td>
<td>Sets the y label of the curve.</td>
</tr>
</tbody>
</table>

### 4.2.3 QA Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbrev.</th>
<th>Description / Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA CHAR ANGLE</td>
<td>(QACA)</td>
<td>Normal</td>
</tr>
<tr>
<td>QA CHAR SIZE</td>
<td>(QACS)</td>
<td>Size in Points (1/72 inch)</td>
</tr>
<tr>
<td>QA CHAR WEIGHT</td>
<td>(QACW)</td>
<td>Light</td>
</tr>
<tr>
<td>QA DATABASE</td>
<td>(QAD)</td>
<td>Toggles the display of the database title. Currently does not work in GRAFLAB 2.3. ON OFF</td>
</tr>
<tr>
<td>QA DISPLAY</td>
<td>(QA)</td>
<td>Toggles the display of the QA database. ON OFF</td>
</tr>
<tr>
<td>QA HORIZONTAL LOCATION</td>
<td>(QHL)</td>
<td>Left Right</td>
</tr>
</tbody>
</table>
4.3 GRAFLAB Globals

Shown below is the GRAFLAB 2.3 list of globals, and a description of each one. Typically, the user will employ `setd` to set the globals and `showd` to display their values. This section is included primarily as reference for programmers, or for those who want to know how GRAFLAB works in more detail.

**WARNING:** If a user performs a `clear globals` in MATLAB, then all of the globals shown below are cleared, and GRAFLAB will not function properly until MATLAB is restarted.

4.3.1 Useful Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>whos global</code></td>
<td>To peek at all the hidden globals.</td>
</tr>
<tr>
<td><code>shoglob('LINE_WIDTH')</code></td>
<td>To echo the value of the global.</td>
</tr>
<tr>
<td><code>global BACKGROUND_COLOR</code></td>
<td>To make the globals accessible to the local space.</td>
</tr>
</tbody>
</table>

4.3.2 Plotting Globals

Use `whos global` to list plotting globals in MATLAB. To add a new global, see the programmer's section.

<table>
<thead>
<tr>
<th>Global</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>'activecurves'</code></td>
<td>A list of the active curves.</td>
</tr>
<tr>
<td><code>'ACTIVE_FLAG'</code></td>
<td>Active Set Plot.</td>
</tr>
<tr>
<td><code>'AXIS_AREA'</code></td>
<td>Screen Plot Area.</td>
</tr>
<tr>
<td><code>'AXIS_FORMAT'</code></td>
<td>Type of X Y Grid System.</td>
</tr>
<tr>
<td><code>'AXIS_LABEL_ANGLE'</code></td>
<td>Plot Label Character Angle.</td>
</tr>
<tr>
<td><code>'AXIS_LABEL_SIZE'</code></td>
<td>Plot Label Character Size.</td>
</tr>
<tr>
<td><code>'AXIS_LABEL_WEIGHT'</code></td>
<td>Plot Label Character Weight.</td>
</tr>
<tr>
<td><code>'AXIS_SCALE'</code></td>
<td>Type of Axes Scaling.</td>
</tr>
<tr>
<td><code>'AXIS_TYPE'</code></td>
<td>Plot Axis Type.</td>
</tr>
<tr>
<td><code>'AXIS_X_LABEL'</code></td>
<td>Plot X Labels Display.</td>
</tr>
<tr>
<td><code>'AXIS_Y_LABEL'</code></td>
<td>Plot Y Labels Display.</td>
</tr>
<tr>
<td><code>'BACKGROUND_COLOR'</code></td>
<td>Plot Background Color.</td>
</tr>
<tr>
<td><code>'COMPLEX_DATA_TYPE'</code></td>
<td>Flag for Real vs Imag, Mag, or Phase.</td>
</tr>
<tr>
<td><code>'DECIOLOG'</code></td>
<td>Decimate Log Status (ON/OFF).</td>
</tr>
<tr>
<td><code>'DECIMATE'</code></td>
<td>Decimate Bins/Factor.</td>
</tr>
<tr>
<td><code>'DECITEST'</code></td>
<td>Decimate Test Status (ON/OFF).</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>'DEVICE_GRID'</td>
<td>Plot Grid Status (ON/OFF).</td>
</tr>
<tr>
<td>'FONT_TYPE'</td>
<td>Plot Character Font Type.</td>
</tr>
<tr>
<td>'LINE_WIDTH'</td>
<td>Plot Line Width.</td>
</tr>
<tr>
<td>'PLOT_EXTREMES'</td>
<td>Plot Axes Limits.</td>
</tr>
<tr>
<td>'PLOT_LEGEND'</td>
<td>Legend Text.</td>
</tr>
<tr>
<td>'PLOT_LEGEND_BOX'</td>
<td>Scale factor for Plot legend.</td>
</tr>
<tr>
<td>'PLOT_LEGEND_FLAG'</td>
<td>Legend Text Display ('ON' 'OFF').</td>
</tr>
<tr>
<td>'PLOT_LEGEND_FS'</td>
<td>Plot legend font size.</td>
</tr>
<tr>
<td>'PLOT_LEGEND_LOC'</td>
<td>Legend Location in % [0.25 0.25].</td>
</tr>
<tr>
<td>'PLOT_LINE_COLORS'</td>
<td>Plot Colors.</td>
</tr>
<tr>
<td>'PLOT_LINE_ORDER'</td>
<td>Plot Line Types.</td>
</tr>
<tr>
<td>'PLOT_LINE_SYMBOLS'</td>
<td>Plot Symbols.</td>
</tr>
<tr>
<td>'PLOT_LINE_TYPE'</td>
<td>Plot Line Type.</td>
</tr>
<tr>
<td>'PLOT_NAME'</td>
<td>Plot (Figure) Name.</td>
</tr>
<tr>
<td>'PLOT_TITLE'</td>
<td>Plot Title Lines.</td>
</tr>
<tr>
<td>'PLOT_X_LABEL'</td>
<td>Plot X Label.</td>
</tr>
<tr>
<td>'PLOT_Y_LABEL'</td>
<td>Plot Y Label.</td>
</tr>
<tr>
<td>'QA_CHAR_ANGLE'</td>
<td>QA Character Display Angle.</td>
</tr>
<tr>
<td>'QA_CHAR_SIZE'</td>
<td>QA Character Display Size.</td>
</tr>
<tr>
<td>'QA_CHAR_WEIGHT'</td>
<td>QA Character Display Weight.</td>
</tr>
<tr>
<td>'QA_DATABASE'</td>
<td>QA DataBase Name Switch(ON/OFF).</td>
</tr>
<tr>
<td>'QA_DISPLAY'</td>
<td>QA Display Switch(ON/OFF).</td>
</tr>
<tr>
<td>'QA_HORZ_LOC'</td>
<td>QA Horizontal Location</td>
</tr>
<tr>
<td>'QA_VERT_LOC'</td>
<td>QA Vertical Location</td>
</tr>
<tr>
<td>'TITLE_CHAR_ANGLE'</td>
<td>Title Line Character Angle.</td>
</tr>
<tr>
<td>'TITLE_CHAR_SIZE'</td>
<td>Title Line Character Size.</td>
</tr>
<tr>
<td>'TITLE_CHAR_WEIGHT'</td>
<td>Title Line Character Weight.</td>
</tr>
<tr>
<td>'TITLE_HORZ_LOC'</td>
<td>Horizontal Title Location</td>
</tr>
<tr>
<td>'TITLE_VERT_LOC'</td>
<td>Vertical Title Location</td>
</tr>
</tbody>
</table>

4.3.3 Calculation Globals

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'AUX_TYPE'</td>
<td>Shock Spectra Type.</td>
</tr>
<tr>
<td>'CURVE_DEFAULT'</td>
<td>Curve Definition.</td>
</tr>
<tr>
<td>'DAMPING'</td>
<td>Damping (Decay Rate) Value.</td>
</tr>
</tbody>
</table>
4.3.4 Other Globals

| 'GRAVITY_CONSTANT' | Value to use for 'G'. |
| 'INTERPOLATION_TYPE' | Interpolation Parameter. |
| 'MATCH_CHARACTER' | Match Character. |
| 'NUMBER_SAMPLES' | Number of Samples Value. |
| 'QUERY' | Query flag. |
| 'ROUND' | X/Y Rounding Places. |
| 'SAMPLE_RATE' | Sample Rate. |
| 'SS_TYPE' | Shock Spectra Parameters. |
| 'TOLERANCE' | Tolerance. |
| 'WILD_CHARACTER' | Wild Card Character. |
5.0 GRAFLAB m-files

This section documents all the GRAFLAB m-files and many useful MATLAB m-files. GRAFLAB analysis functions that start with a "g" imply that they have been accepted as general "quality assured" m-files. Those that start with other letters (by convention, the first letter of the programmer's given name) are subject to modification and/or have not been accepted as a standard GRAFLAB utility.

5.1 General Notes on GRAFLAB m-files

1. GRAFLAB performs much of its housekeeping tasks in the MATLAB global workspace. However, manipulation of data must be done in the MATLAB local workspace.
2. GRAFLAB utilities are written for the most part as MATLAB functions and therefore, follow the corresponding Matlab conventions.
3. GRAFLAB uses the concept of an active set (just as with GRAFAID). Unless otherwise noted, GRAFLAB utilities will operate on either GRAFLAB curves in the active set or data in the local Matlab workspace.
4. Unless otherwise noted, GRAFLAB expects an nx2 matrix: the first column is the independent variable (usually assumed to be time in seconds or frequency in Hertz), the second column is the dependent variable (often assumed to be acceleration in G's for purposes of automatic curve labeling).
5. MATLAB works on row indices when working with the dependent variable (time). To permit the user to work with time directly, several utilities have been developed to make the conversion in the background.
6. A great strength of MATLAB is its ability to use FOR loops to perform repetitive tasks.
7. Curve data is edited using standard MATLAB matrix manipulation (refer to section 8, example 4 for a sample editing session).
8. GRAFLAB takes advantage of the Matlab HELP file syntax. Therefore, the user may display a help file for any specific m file by typing help mfilename. Similarly, the user may display a listing of all of the available m files within a GRAFLAB toolbox by typing help toolboxname (refer to Section 5.2 below for an explanation of available toolboxes).

5.2 Analysis m-files

GRAFLAB utilities are partitioned in a manner that is analogous to Matlab's toolboxes in order to provide the reader with a useful grouping of like routines. Figure 4 presents the preferred grouping of utilities. Only the utilities in the gmechanical and gthermal toolboxes will be discussed in this manual.

The gshaker toolbox contains the m-files needed to synthesize shock pulses for use in shaker testing. The gmechanical utility gdsine was written to act as a front end to one of the main sets of gshaker synthesis routines.

The caputil and dosutil toolboxes contain many useful routines not considered suitable for inclusion in GRAFLAB at this time and are therefore left under the control of a specific GRAFLAB user. The m files contained in these toolboxes are either considered to be work in progress (i.e., GRAFLAB utilities that have not been completely checked out) or Matlab m files that have been completely checked out but do not adhere to the GRAFLAB
syntax. The testcases toolbox contains m files that serve as quality assurance checks for the various utilities as well as useful examples of syntax and intended usage.

![Figure 4. GRAFLAB Utility Directory Tree](image)

Other toolboxes (both formal and user) will be added to GRAFLAB as the need arises.

In order to achieve the full functionality described in this manual, the GRAFLAB 2.3 STARTUP.M file must point to the following directories:

1. disk1/glutil/gmechanical
2. disk1/glutil/gthermal
3. disk1/glutil/gshaker
4. disk1/glutil/testcases (m-files to help quality assure utility output, and to serve as examples of intended usage).

Only the first two directories will be discussed in detail.

### 5.2.1 Mechanical Shock & Vibration (gmechanical)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>avdconv</td>
<td>Routine to support avdplot.</td>
</tr>
<tr>
<td>avdgrid</td>
<td>Routine to support avdplot.</td>
</tr>
<tr>
<td>avdplot</td>
<td>Plots Shock Response Spectra on tricordinate paper</td>
</tr>
<tr>
<td>dsd</td>
<td>Computes the displacement spectral density (DSD) from the corresponding acceleration spectral density (ASD/PSD).</td>
</tr>
<tr>
<td>gtempmom</td>
<td>Computes the temporal moments for a time history.</td>
</tr>
<tr>
<td>gbutterfilt</td>
<td>Digital Butterworth lowpass filter.</td>
</tr>
<tr>
<td>gclip</td>
<td>Computes +/-n sigma envelopes based on a running rms for a time history and returns a time history that is the minimum of the original curve and the envelopes. The utility is useful for mitigating the effects of numerous data dropouts (see also gspike).</td>
</tr>
</tbody>
</table>

GRAFLAB 2.3 31 GRAFLAB m-files
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gconvolve</td>
<td>Computes an output time history from an input time history and frequency response function using the overlap and add method.</td>
</tr>
<tr>
<td>gcse</td>
<td>Computes the curve set data extremes (xmin, xmax, ymin, ymax) for the active set curves.</td>
</tr>
<tr>
<td>gdsine</td>
<td>Computes a decayed sine acceleration history, whose shock response spectra approximates a user defined reference shock response spectra. An auxiliary parameter file containing the associated input parameters is also generated.</td>
</tr>
<tr>
<td>gdsine_rpt</td>
<td>Generates a formatted report for a decayed sine parameter file (see gdsine).</td>
</tr>
<tr>
<td>genvelope</td>
<td>Produces a max or min envelope from a set of data curves.</td>
</tr>
<tr>
<td>genvelope_fatigue</td>
<td>Computes a PSD envelope and associated test duration that will produce the same amount of fatigue damage as a set of user defined PSDs and their associated test durations. The equivalencing procedure is based on a Minor's rule scaling law.</td>
</tr>
<tr>
<td>ghaversine</td>
<td>Computes the shock response spectra for a user defined haversine pulse.</td>
</tr>
<tr>
<td>gintegrate</td>
<td>Computes the integral of a time history using one of two techniques: a fourier transform technique or MATLAB cumsum.</td>
</tr>
<tr>
<td>gpsd</td>
<td>Computes the acceleration or power spectral density for an input acceleration history.</td>
</tr>
<tr>
<td>gpsd_bp</td>
<td>Computes the minimum number of break points needed to uniquely define a uniformly oversampled reference PSD assuming straight line on a log-log plot.</td>
</tr>
<tr>
<td>gpsd_rpt</td>
<td>Generates a formatted report containing the break points for a PSD. This file is intended primarily for use with straight line envelopes.</td>
</tr>
<tr>
<td>gpsdoct</td>
<td>Computes an octal bandwidth PSD from a linear bandwidth PSD.</td>
</tr>
<tr>
<td>grmsavd</td>
<td>Computes the acceleration, velocity, and displacement rms values for a PSD.</td>
</tr>
<tr>
<td>gsample</td>
<td>Produces a uniformly sampled matrix of points for a set of input curves.</td>
</tr>
<tr>
<td>gsmooth</td>
<td>Computes the smoothed version of a curve using a running average.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gspike</td>
<td>Identifies the highest amplitude points in a time history and scales them using a user defined scale factor. This utility is designed to assist the user in identifying a finite set of data dropouts and noise spikes (see also gclip).</td>
</tr>
<tr>
<td>gsrsp</td>
<td>Computes the shock response spectra for a time history.</td>
</tr>
<tr>
<td>gsrs_rpt</td>
<td>Generates a formatted report containing the break points for a shock response spectra. This file is intended primarily for use with straight line envelopes.</td>
</tr>
<tr>
<td>gtol_band</td>
<td>Computes the +/-N dB tolerance bands for a user defined reference curve.</td>
</tr>
<tr>
<td>gtranspose</td>
<td>Generates the two-sided (transposed or 0-2π) frequency response function from a one-sided (0-p) frequency response function.</td>
</tr>
<tr>
<td>gxspac</td>
<td>Computes the delta x value (and corresponding sample rate) and evaluates whether the input vector x is spaced evenly.</td>
</tr>
<tr>
<td>gxval2nrow</td>
<td>Identifies the row indices based on a set of user defined time (dependent variable) values. This utility is used to define a desired time range in terms of the row indices for use in Matlab.</td>
</tr>
<tr>
<td>vsd</td>
<td>Computes the velocity spectral density (VSD) from the corresponding acceleration spectral density (ASD/PSD).</td>
</tr>
<tr>
<td>gpsdbin</td>
<td>Computes a coarser linear bandwidth averaged version of a linear bandwidth PSD.</td>
</tr>
<tr>
<td>greconstruct</td>
<td>Generates a higher resolution curve from an input curve using reconstruction techniques</td>
</tr>
<tr>
<td>gwaterfall</td>
<td>Generates a waterfall plot (Matlab Specgram).</td>
</tr>
<tr>
<td>gwavsyn</td>
<td>A function to synthesize a transient composed of sums of WAVSYN wavelets that will match a specified shock response spectrum.</td>
</tr>
<tr>
<td>rfft</td>
<td>Finds the fft of a real time history.</td>
</tr>
<tr>
<td>rffti</td>
<td>Finds the inverse fft of a spectrum for a real time history.</td>
</tr>
</tbody>
</table>
5.2.2 Thermal Data Reduction Utilities (gthermal)

<table>
<thead>
<tr>
<th>sodDFT</th>
<th>Thermal DR file to calc qi from DFT thermocouples.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sodqvt</td>
<td>Thermal DR file to combine q vs surf temp.</td>
</tr>
<tr>
<td>sodreadplt</td>
<td>Thermal DR file to read a SODDIT .plt file.</td>
</tr>
<tr>
<td>sodsaveall</td>
<td>Thermal DR file to save the input TC matrix.</td>
</tr>
<tr>
<td>sodwriteinp</td>
<td>Thermal DR file to write a SODDIT .inp file.</td>
</tr>
</tbody>
</table>

5.3 Database m-files

The "m-files," which are described below make up the GRAFLAB23 database structure. They are grouped by categories. They are found in the following directory on the Department 9735 HP735 server:

/disk1/graflab23

5.3.1 Database - User m-files

<table>
<thead>
<tr>
<th>delcurf</th>
<th>Deletes a curve from the GRAFLAB 2.x database.</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcipy</td>
<td>Copies curve(s) from another GRAFLAB database to the current database.</td>
</tr>
<tr>
<td>gdir</td>
<td>Provides a selectable directory of the curves in the GRAFLAB database.</td>
</tr>
<tr>
<td>gfile</td>
<td>General file input with headers and various delimiters.</td>
</tr>
<tr>
<td>global</td>
<td>Useful for making GRAFLAB curves visible both to the local Matlab workspace and other functions.</td>
</tr>
<tr>
<td>gnargin</td>
<td>Checks the size and data type for any matlab variable. This m-file is intended to be used by programmers to verify that the user has passed a valid input argument to a GRAFLAB utility.</td>
</tr>
<tr>
<td>ninput</td>
<td>Inputs a neutral file into the GRAFLAB 2.x database.</td>
</tr>
<tr>
<td>noutput</td>
<td>Writes out a neutral for selected curves.</td>
</tr>
<tr>
<td>saveaux</td>
<td>Moves an auxiliary file (stored in MATLAB .mat format) from the user's present working directory (pwd) to the corresponding GRAFLAB database (/alldata/cname/Aux.mat).</td>
</tr>
<tr>
<td>savecurf</td>
<td>Saves header, data, and/or Aux data for a curve.</td>
</tr>
<tr>
<td>setd</td>
<td>Sets GRAFLAB globals, and curve attributes.</td>
</tr>
<tr>
<td>setdloc</td>
<td>Makes the active set curves visible to the Matlab local workspace.</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>showd</td>
<td>Shows the GRAFLAB globals, and curve attributes.</td>
</tr>
</tbody>
</table>
5.3.2 Database – GRAFLAB m-files for programmers

These files will never be called by the average user, but they are documented here for completeness.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixPLOTparm</td>
<td>Fixes the plot parameters upon setactive.</td>
</tr>
<tr>
<td>gl2update</td>
<td>Updates the database or a curve to the current version.</td>
</tr>
<tr>
<td>loaddata</td>
<td>Loads the data from a curve.</td>
</tr>
<tr>
<td>loadheader</td>
<td>Loads the header from a curve.</td>
</tr>
<tr>
<td>load_save.txt</td>
<td>A text file describing the programmer's use of load &amp; save.</td>
</tr>
<tr>
<td>saveheader</td>
<td>Saves the header of a curve.</td>
</tr>
<tr>
<td>setactive</td>
<td>Called by setd('act','...').</td>
</tr>
<tr>
<td>setcvant</td>
<td>Sets the curve extremes.</td>
</tr>
<tr>
<td>setplext</td>
<td>Sets the plot extremes.</td>
</tr>
<tr>
<td>str2array</td>
<td>Parses an input string into a real array.</td>
</tr>
<tr>
<td>str2mstr</td>
<td>Parses an input string into a string matrix.</td>
</tr>
<tr>
<td>sys</td>
<td>Provides platform independent system calls, such as list files, list directories, copy, move, make directories, or delete directories.</td>
</tr>
<tr>
<td>template</td>
<td>A template used for writing new m files.</td>
</tr>
</tbody>
</table>
5.4 File I/O m-files

The "m-files," which are described below make up the GRAFLAB23 database structure. They are grouped by categories. They are found in the following directory on the Department 9735 HP735 server:

/disk1/graflab23

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gfile</code></td>
<td>Loads ASCII files with headers and data.</td>
</tr>
<tr>
<td><code>load</code></td>
<td>Loads Matlab mat files or ASCII files (tab or space delimited).</td>
</tr>
<tr>
<td><code>ninput</code></td>
<td>Reads in Neutral files.</td>
</tr>
<tr>
<td><code>noutput</code></td>
<td>Writes out Neutral files.</td>
</tr>
<tr>
<td><code>save</code></td>
<td>Writes out Matlab mat files or ASCII files (space delimited).</td>
</tr>
<tr>
<td><code>savecurf</code></td>
<td>Saves header, data, and/or Aux data for a curve.</td>
</tr>
<tr>
<td><code>readline</code></td>
<td>Read in one line of an open file as text.</td>
</tr>
<tr>
<td><code>readtext</code></td>
<td>Puts the contents of a file into a text matrix.</td>
</tr>
<tr>
<td><code>writetext</code></td>
<td>Writes out a real or string matrix to a ASCII file.</td>
</tr>
<tr>
<td><code>writeuff58</code></td>
<td>Writes out a universal file format 58.</td>
</tr>
</tbody>
</table>

5.5 Plotting m-files

The "m-files," which are described below make up the GRAFLAB23 database structure. They are grouped by categories. They are found in the following directory on the Department 9735 HP735 server:

/disk1/graflab23

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bplot</code></td>
<td>Plots GRAFLAB active set using GRAFLAB globals that define the plot. See Matlab plot command to plot local working variables.</td>
</tr>
<tr>
<td><code>glegend</code></td>
<td>Called from bplot to a curve legend string for each curve in the active set. Use setd('pld','on') to set bplot's legend display on.</td>
</tr>
<tr>
<td><code>ggrid</code></td>
<td>Creates a major grid pattern on a bplot. This was created as an alternative to the Matlab grid option (which plots both major and minor grids). Use setd('pg','on') to turn on grid display.</td>
</tr>
<tr>
<td><code>plot2</code></td>
<td>Plots an nx2 matrix, second column vs. first.</td>
</tr>
</tbody>
</table>

5.6 String Manipulation m-files

The "m-files," which are described below make up the GRAFLAB23 database structure. They are grouped by categories. They are found in the following directory on the Department 9735 HP735 server: GRAFLAB 2.3

GRAFLAB m-files
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(text)</td>
<td>Echoes ASCII values for each character.</td>
</tr>
<tr>
<td>blanks(num)</td>
<td>Returns a string num spaces long.</td>
</tr>
<tr>
<td>clearws(string)</td>
<td>Clears leading &amp; trailing spaces of string (not internal spaces).</td>
</tr>
<tr>
<td>clearws</td>
<td>Clears leading white space (ASCII char 32) in string.</td>
</tr>
<tr>
<td>deblank(string)</td>
<td>Removes spaces at beginning.</td>
</tr>
<tr>
<td>findstr('/',string)</td>
<td>Returns a vector containing all locations of '/' in string.</td>
</tr>
<tr>
<td>isempty(string)</td>
<td>Returns 1 if string is [].</td>
</tr>
<tr>
<td>~isempty(string)</td>
<td>Returns 1 if string is not [].</td>
</tr>
<tr>
<td>isletter(string)</td>
<td>Returns 1 if is an alphabetic letter.</td>
</tr>
<tr>
<td>isspace</td>
<td>Returns 1 for space, newline, carriage return, tab, vert tab or formfeed.</td>
</tr>
<tr>
<td>item(num,instr,delim)</td>
<td>Parses a string by items and returns item n.</td>
</tr>
<tr>
<td>readtext(filename,delim)</td>
<td>Reads in a file as a text matrix.</td>
</tr>
<tr>
<td>setstr(textstring)</td>
<td>Restores a string to text echo.</td>
</tr>
<tr>
<td>str2mat</td>
<td>Concatenates two strings (or text matrices).</td>
</tr>
<tr>
<td>word(n,string)</td>
<td>Parses a string by words and returns word n.</td>
</tr>
<tr>
<td>wildcard</td>
<td>Finds the location of a string in a text Matrix.</td>
</tr>
<tr>
<td>writetext(filename,TextMat)</td>
<td>Writes out a real or string matrix as a text file.</td>
</tr>
</tbody>
</table>

Example: Suppose you want to read from the text file named "input.dat" containing a filename, and the time range of the data, and use that time range to choose a portion of the data.

Text File: "input.dat"

```
TIMEFILE.DAT,10:05:05-15
```

The MATLAB syntax for this would be

```matlab
theinput=readtext('input.dat',10);
% Note: theinput now is 'TIMEFILE.DAT,10:05:05-15'
% The first item of theinput (by comma delim) is 'TIMEFILE.DAT'
datafile=item(1,theinput,',');
eval(['load ',datafile]) % load the data in from 'TIMEFILE.DAT'
% Get the first item (by period delim) which is 'TIMEFILE'
dataname=item(1,datafile,'.');?></code>

---

GRAFLAB 2.3

GRAFLAB m-files
eval(['thedata= ' dataname ';']) % put the data into thedata

% Now we need to truncate the data according to the time.
timerange=item(2,thetimeinput,'');
% timerange is now '10:05:05-15'
secondrange=item(3,timerange,':');
% second range is now '05-15'
begintime=item(1,secondrange,'-');
endtime = item(2,secondrange,'-');
% begintime is now '05' and endtime is now '15'
% NOW TRUNCATE THE DATA USING GXVAL2NROW.
thedata=gxval2nrow(thedata,[ begintime,endtime ]);
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>format</strong></td>
<td>Controls the display format of numbers. Example: Try <code>format short e</code>.</td>
</tr>
<tr>
<td><strong>prepepsf</strong></td>
<td>Prepares and prints an epsf file for import into word processor.</td>
</tr>
<tr>
<td><strong>startup</strong></td>
<td>Defines the path to find these GRAFLAB files.</td>
</tr>
<tr>
<td><strong>template</strong></td>
<td>A template file for writing new m files.</td>
</tr>
</tbody>
</table>
Always include a "modified/date/who did it" comment when updating a GRAFLAB m-file to help with understanding future bugs caused by current features and fixes.

6.1 Functions that Call GRAFLAB System Routines

The routine SYS.M is designed to isolate the system calls to one routine, simplifying the process of porting the code to other operating systems. Six system calls are necessary in order to port the code.

1. LSD - List directories
2. LSF - List files
3. DELR - Recursive delete of directory and subdirectories underneath
4. CPR - Recursive copy of directory and contents
5. MV - Move a directory
6. MKDIR - Create a new directory

If this routine can be successfully ported to another operating system where MATLAB resides, then the rest of GRAFLAB can be transparently ported. Table 1 provides a list of the GRAFLAB routines that use SYS.M to make system calls, such as list files, delete, copy, move, or make directories.

<table>
<thead>
<tr>
<th>Routine Name</th>
<th>LSD</th>
<th>LSF</th>
<th>DELR</th>
<th>CPR</th>
<th>MV</th>
<th>MKDIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>delcurf</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gcropy</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gdir</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gl2update</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ninput</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>savecurf</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>sttd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Use the SYS command to do system commands

GRAFLAB has been written so that all the system commands are located in sys.m. For example, to get a list (a dir in DOS) of all the *.inp files and read the text of each one, perform a task with the text from each file, execute the following commands.

LIST FILES

```matlab
thelist=sys('lsf','*.inp');
```
if ~isempty(thelist)
    for j=1:size(thelist,1)
        theText=readtext(clearaws(thelist(j,:)));
        % do something here with the text
        % of each of the input files.
    end
end

• Note that `sys('lsf')` will list the files in `./alldata`.
• To check for the file existence, for example prior to opening it, use the following:
if ~isempty(sys('lsf','pathname/subdir/filename.txt'))
    fid=fopen('pathname/subdir/filename.txt','rt');
else
    disp('WARNING: file does not exist!')
end

LIST DIRECTORIES
Provides a directory of the directories.
theDirs=sys('lsd','pathname/filename');

• Note that `sys('lsd')` will list the directories in `./alldata`.

RECURSIVE DELETE
`sys('delr','./alldata/curvename')` will delete the
`./alldata/curvename` directory and all subdirectories underneath it.

RECURSIVE COPY
`sys('cpr','./alldata/curvename','/disk1/username/project/alldata')` will copy the directory curvename, and all of its contents, under the second path.

RECURSIVE MOVE
`sys('mv','./alldata/curvename','/disk1/username/project/alldata')` will move the directory curvename, and all of its contents, under the second path.
MAKE DIRECTORY

sys('mkdir', './alldata', 'curvename') will make the directory curvename under ./alldata.

6.2 Add/Remove a Global Variable to GRAFLAB

The following four routines must be updated when adding a new global variable, and then documented in each routine's Help file to show that it has been added.

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actparm</td>
<td>Returns the default value of the global when called.</td>
</tr>
<tr>
<td>setdglob</td>
<td>Calls actparm.m to set each of the globals (usually called at startup).</td>
</tr>
<tr>
<td>setd</td>
<td>Sets the new value of the global to an allowable value.</td>
</tr>
<tr>
<td>showd</td>
<td>Shows (and interprets in plain language) the value of the global.</td>
</tr>
</tbody>
</table>

Finally, curvparm.m is the routine that controls system variables.
6.3 GRAFLAB 23 Test Area

Located under the GRAFLAB directory is a subdirectory called gl2test. This directory contains sample data for testing GRAFLAB:

<table>
<thead>
<tr>
<th>Complex data</th>
<th>A sample database to test GRAFLAB’s handling of complex data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral files</td>
<td>From GRAFALID, 3 column, 2 column, bad curvenames (containing ‘,’ or ‘-’ or first char is not alpha).</td>
</tr>
<tr>
<td>Universal files</td>
<td>Universal files for input using uffinput.m.</td>
</tr>
<tr>
<td>29E data</td>
<td>Sample optical data files for input using input29ea.m and input29ed.m.</td>
</tr>
<tr>
<td>Old Databases</td>
<td>There is a database from version 1.0, 2.0 and 2.1. These databases should be copied to a new area, and then used. DO NOT WORK DIRECTLY ON THESE DATABASES using the current version of GRAFLAB or you will update them and lose them.</td>
</tr>
</tbody>
</table>

6.4 Create a New GRAFLAB function

Located under the GRAFLAB directory is an m-file called template.m, which is the recommended best practice m-file header documentation.

```matlab
function y=templdate(inputarg)
%     template.m
% SYNTAX:
% DESCRIPTION:
% EXAMPLES:
% SEE ALSO:
% AUTHOR:       Date:
% Modified:     Date: Description:
```

GRAFLAB 2.3

Programmer’s Guide
7.0 MATLAB Memory Management

When processing large volumes of data, it is good practice to use the `clear` utility to free memory stored in variable `xxx` that is no longer needed. This releases memory that MATLAB has allocated. However, for a variable that has been declared global, `(global xxx)`, `clear xxx` clears only the active process (working space) variable and does NOT clear the global variable that was passed up to the parent MATLAB process. Therefore, MATLAB does not release the memory, it only makes it invisible to the active process.

To release memory from a global variable, perform a `clear global xxx`.

An additional useful command is `pack`, which defragments the memory that MATLAB occupies.

8.0 GRAFLAB Tips

This section contains a few handy "how to's" with examples.

1. **Wildcard looping on SETD & SHOWD**, for example, show curve titles of PA* curves:
   
   ```matlab
   a=gdir('PA*');
   for i=1:size(a,1);showd('ct',clearaws(a(i,:)));end
   ```

2. **Check to determine whether vector A contains a scalar B**
   
   This takes advantage of the vectorized if statement in MATLAB.
   
   ```matlab
   A=[1 2 3 4 5];
   B=[2];
   if ~all(A-B) % Uses vectorized if. If not all non-zero then
                B is in A!
       disp('Yes, vector A contains at least one of scalar B!')
    end
   ```

3. **Using the eval command**

   You can evaluate string expressions as though they are GRAFLAB commands:
   
   ```matlab
   thecurve='PAGY';
   a=['setd('act',' thecurve ')'];
   eval(a)
   ```

   The above three lines are the same as the following:
setd('act','PAGY')
Another example:
clist='curve';
setd('act',clist)
setdloc
x=eval(clist);

4. Multiply the y values of curve1 by a factor of 2 (This is an example of editing a GRAFLAB curve).

Simply set it active, make it available to the local space, then do the math and save only the data back out.

setd('act','curve1')
setdloc
curve1(:,2)=2*curve1(:,2);
savecurf('curve1',1,2)

5. Write select CURVE1 data to file named FILE1

Put the output of showd('cd' ... ) into a file using writetext. The data can be truncated according to the independent variable, for example from TIME1 to TIME2.

writetext('FILE1',showd('cd','CURVE1',[TIME1 TIME2]));

6. Use graphical cursor input to create curves.

Create a global variable called test. Use ginput to digitize data points from the active figure window, scaling according to the axis scaling. Use help ginput for more complete instructions.

% Need to have a plot figure displayed.
global test
[test(:,1) test(:,2)]=ginput(5);
% digitize the data using the cursor
savecurf('test',0,6)
Appendix A

Example Startup.m file

This shows an example startup.m file. The startup.m file is executed automatically upon startup by MATLAB. Five key functions are accomplished by this file. Items 1, 3, and 4 are necessary for GRAFLAB to operate properly.

1. Set the appropriate paths.
2. Echo the version of GRAFLAB.
3. Set the GRAFLAB globals (setdglob).
4. Check to determine whether the database needs updating; if so, perform a g12update.
5. Customize the user's local GRAFLAB environment.

```matlab
% EXAMPLE STARTUP.M FILE

a=['/matlab_42a/toolbox/matlab/general:';
a=['/matlab_42a/toolbox/matlab/ops:';
a=['/matlab_42a/toolbox/matlab/lang:';
a=['/matlab_42a/toolbox/matlab/elmat:';
a=['/matlab_42a/toolbox/matlab/specmat:';
a=['/matlab_42a/toolbox/matlab/elfun:';
a=['/matlab_42a/toolbox/matlab/specfun:';
a=['/matlab_42a/toolbox/matlab/matfun:';
a=['/matlab_42a/toolbox/matlab/datafun:';
a=['/matlab_42a/toolbox/matlab/polyfun:';
a=['/matlab_42a/toolbox/matlab/funfun:';
a=['/matlab_42a/toolbox/matlab/sparfun:';
a=['/matlab_42a/toolbox/matlab/plotxy:';
a=['/matlab_42a/toolbox/matlab/plotxyz:';
a=['/matlab_42a/toolbox/matlab/graphics:';
a=['/matlab_42a/toolbox/matlab/color:';
a=['/matlab_42a/toolbox/matlab/sounds:';
a=['/matlab_42a/toolbox/matlab/strfun:';
a=['/matlab_42a/toolbox/matlab/iofun:';
a=['/matlab_42a/toolbox/matlab/demos:';
a=['/matlab_42a/toolbox/local:';
a=['/matlab_42a/toolbox/signal:';
a=['/disk1/graflab23:';
a=['/disk1/glutil/gmechanical:';
a=['/disk1/glutil/gthermal:';
a=['/disk1/glutil/gshaker:';
a=['/disk1/glutil/testcases';
path(a);
clear a
disp(['*******************************'])
```

GRAFLAB 2.3

Appendix A
disp(['Welcome to GRAFLAB version ' curvparm('GL_VER')])
disp([' for ' curvparm('MACHINE_ID')])
disp(['********************************************************************'])
disp(' ')
disp(' Type buglist at the prompt to get a current bug listing')

% 
% SET THE GRAFLAB GLOBALS
    setdglob
% UPDATE THE DATABASE IF NECESSARY
    gl2update
% OPTIONAL, CUSTOMIZED STUFF
    setd('dg','on')  % TURN ON GRIDS
    setd('tj','center')  % CENTER THE TITLE
Index

abs 38
acceleration 32
active curves 27
active set 17, 19, 23, 30
actparm 39, 43
alldata 7
Angle 28
ASCII 37, 38
ASCII files 10
AUTO 18
Aux.mat 14
auxillary data 14
auxilliary file 34
Axes Limits 28
Axes Scaling 27
axis format 17, 23
AXIS LABEL ANGLE 23
AXIS LABEL SIZE 23
AXIS LABEL WEIGHT 23
AXIS SCALE 23
axis type 10, 18, 24, 26, 27
AXIS X LABEL TYPE 24
AXIS Y LABEL TYPE 24
BACKGROUND COLOR 24, 27
basics 10
bitmap 19
blanks 38
bplot 37
break points 32, 33
carriage return 38
Character Angle 27
Character Size 27, 28
Character Weight 27
clear 45
clear global 45
clearaws 38
clearws 38
Colors 28
complex numbers 9
concatenate 38
Copies curve(s) 34
copy 12, 36, 42
cpr 42
CPU time 20
cursor input 46
curve data 19, 26
curve header 20
curve legend 18, 37
curve title 18, 20
curvparm 16, 39
curvparm.m 43
Damping 28
dashed 25
dashes 18
data 37
Data.mat 14
database design 14
DataBase Name 28
deblank 38
decayed sine 32
Decimate Bins 27
Decimate Log 27
Decimate Test 27
defragment 45
delcurf 34
delete 42
delete directories 36
Deletes a curve 34
delr 42
delta x 33
dependent variable 30
digitize data 46
dir 41
directories 36, 42
directory 34, 42, 43
directory Separator 16
DIR_PATH 16
DIR_SEP 16
displacement 32
displacement spectral density 31
Display Angle 28
display format 40
Display Switch 28
dot-dash 25
dropouts 33
decayed sine 32
enhancements 9
envelope 32
envelopes 33
epsf 40
eval 45
evaluate 45
evenly spaced 33
example startup.m 47
EXTREMES 26, 32
fft 33
figure position 19
file existence 42
File Name Length 16
Finds 38
findstr 38
fixPLOTparm 36
FONT TYPE 24, 28
fopen 42
for 23
FOR loops 30
formfeed 38
frequency response function 33
gclip 33
gcopy 12, 34
gdir 11, 23, 34
General file input 34
gfile 34, 37
ggrid 37
ginput 46
gl2test 44
gl2update 8, 36
glegend 37
global 10, 34, 45
global variable 39
global variables 17
gpsd.m 13
GRAFLAB 7
GRAFLAB 1.0 8, 9
GRAFLAB routines 7
grid 37
grid display 37
grids 17
gspike 31
haversine 32
Header.mat 14
headers 37
help 7, 13
Horizontal Location 28
import 12
independent variable 30, 46
Initialize 39
integral 32
inverse fft 33
invisible 45
isempty 38
isletter 38
item 38
LEGEND 26
legend display 18, 37
legend font 28
legend location 18, 28
Legend Text 28
line colors 18
Line Type 28
Line Types 28
line width 19, 25, 28
Linear Axes 18
list files 36, 41
load 10, 37
loaddata 36
loadheader 36
load_save 36
local space 39, 46
Log Axes 18
log file 39
looping 45
MACHINE_ID 16
make directories 36
Match Character 29
memory 45
mkdir 8, 43
MONO 26
Monotonically 26
move 36, 42
mv 42
NAME 26
nargout 23
neutral file 12, 34, 37
new global 43
newline 38
Newline Character 16
ninput 12, 34, 37
noise spikes 33
nolog 24
Non monotonic 26
noutput 12, 34, 37
nterpolation 29
Number of Samples 29
octal bandwidth 32
optical data 44
pack 45
Parse 36, 38
path 40
Path Name Length 16
PLFS 24
plot 11, 27, 37
Plot Area 27
plot extremes 10, 18, 24, 36
PLOT GRID 24, 28
PLOT LEGEND DISPLAY 24
PLOT LEGEND FONT SIZE 24
PLOT LEGEND LOCATION 24
PLOT LINE COLORS 24
PLOT LINE ORDER 25
PLOT LINE SYMBOLS 25
PLOT LINE TYPE 25
PLOT LINE WIDTH 25
plot lines 19
plot name 19, 25
plot parameters 14, 36
PLOT POSITION 25
plot title 10, 19, 25
PLOT X LABEL 25
PLOT Y LABEL 25
plot2 37
Plotting 17
Points 25
polar 24
Polar Axes 18
POSTSCRIPT 19
power spectral density 32
prepepsf 40
previous versions 9
prints 40
process id 20
programmers 27
PSD 32
QA 26, 28
qa display 19
QA Vertical Location 28
qad display 19
Query flag 29
readline 37
readtext 37, 38
reconstruction 33
Rename 26
rename curve 18
rms 31, 32
Rounding 29
row indices 33
Sample Rate 29
Save 34, 36, 37
saveaux 34
savecurf 11, 34, 37, 46
saveheader 36
setactive 36
setcvext 36
setd 12, 17, 19, 34, 43
setdglob 39, 43
setdloc 23, 35, 39, 46
setplext 36
setstr 38
shock response spectra 32, 33
shock response spectrum 33
Shock Spectra 28, 29
shoglob 39
showd 12, 19, 23, 35, 43
Size 25, 28
space 38
spaces 38
Specgram 33
startup 40
startup.m 7, 47
str2arry 36
str2mstr 36
straight line envelopes 32
string matrices 39
subdirectories 7
subdirectory 14
symbols 19, 28
sys 36, 41
sys.m 41
system calls 36
system commands 41
system parameter 39
system variables 43
template 36, 40
template.m 44
testing GRAFLAB 44
Text Display 28
text file 38
Thermal 34
tics 17
time range 33
TITLE CHAR ANGLE 25
TITLE CHAR SIZE 25
TITLE CHAR WEIGHT 25
TITLE HORIZONTAL LOCATION 25
Title Justification 28
TITLE JUSTIFICATION 25
Title Lines 28
Title Location 28
TITLE VERTICAL LOCATION 25
Tolerance 29
transient 33
truncete 11
truncated 46
tutorial 10
uniformly sampled 32
universal file 37
update 8, 36
users 20
vector rendering 19
vectorized if 45
velocity 32
velocity spectral density 33
version 8, 36, 44
Version 2.3 9
waterfall plot 33
wavelets 33
Weight 28
white space 38
whos 11, 27
who_mat 20
Wild Card Character 29
wildcard 20, 38
word 38
writetext 37, 38, 46
writeuff58 37
x label 10, 18, 19, 23, 26
X Labels 27
X.TYPE 26
X-Y Grid 27
xylog 24
y label 10, 18, 19, 26, 27
<table>
<thead>
<tr>
<th>No.</th>
<th>MS</th>
<th>Name</th>
<th>Copy</th>
<th>MS</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MS 0415</td>
<td>W. N. Dunn Jr.</td>
<td>1</td>
<td>MS 0439</td>
<td>R. V. Field Jr.</td>
</tr>
<tr>
<td>5</td>
<td>MS 0436</td>
<td>G. L. Maxam</td>
<td>1</td>
<td>MS 0439</td>
<td>C. W. G. Fulcher</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>H. S. Morgan</td>
<td>1</td>
<td>MS 0439</td>
<td>D. W. Lobitz</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>C. R. Adams</td>
<td>1</td>
<td>MS 0439</td>
<td>G. M. Reese</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. B. Aidun</td>
<td>1</td>
<td>MS 0439</td>
<td>D. J. Segalman</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. G. Arguello Jr.</td>
<td>1</td>
<td>MS 0439</td>
<td>H. P. Walther</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>S. W. Attaway</td>
<td>1</td>
<td>MS 0439</td>
<td>W. R. Witkowski</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>M. L. Blanford</td>
<td>1</td>
<td>MS 0555</td>
<td>D. B. Davis</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>K. H. Brown</td>
<td>1</td>
<td>MS 0555</td>
<td>M. S. Garrett</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>S. N. Burchett</td>
<td>1</td>
<td>MS 0555</td>
<td>V. I. Bateman</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>R. S. Chambers</td>
<td>1</td>
<td>MS 0555</td>
<td>D. L. Gregory</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>A. F. Fossum</td>
<td>1</td>
<td>MS 0557</td>
<td>T. J. Baca</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. D. Gruda</td>
<td>1</td>
<td>MS 0557</td>
<td>P. S. Barney</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>M. W. Heinstein</td>
<td>1</td>
<td>MS 0557</td>
<td>T. G. Carne</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>E. L. Hoffman</td>
<td>1</td>
<td>MS 0557</td>
<td>S. E. Klenke</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. Jung</td>
<td>1</td>
<td>MS 0557</td>
<td>J. P. Lauffer</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>S. W. Key</td>
<td>1</td>
<td>MS 0557</td>
<td>R. L. Mayes</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. R. Koteras</td>
<td>1</td>
<td>MS 0557</td>
<td>C. C. O'Gorman</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>C. S. Lo</td>
<td>1</td>
<td>MS 0557</td>
<td>T. L. Paez</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>F. J. Mello</td>
<td>1</td>
<td>MS 0557</td>
<td>M. J. Sagartz</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>K. E. Metzinger</td>
<td>1</td>
<td>MS 0615</td>
<td>R. L. Perry</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. A. Mitchell</td>
<td>1</td>
<td>MS 0865</td>
<td>J. L. Moya</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>M. K. Neilson</td>
<td>1</td>
<td>MS 0865</td>
<td>J. R. Barnum</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. Pott</td>
<td>15</td>
<td>MS 0865</td>
<td>J. S. Cap</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>E. D. Reedy Jr.</td>
<td>2</td>
<td>MS 0865</td>
<td>J. E. C de Baca</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>L. A. Schoof</td>
<td>1</td>
<td>MS 0865</td>
<td>T. Y. Chu</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>G. D. Sjaardema</td>
<td>2</td>
<td>MS 0865</td>
<td>R. D. Foral</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. R. Stewart</td>
<td>2</td>
<td>MS 0865</td>
<td>W. Gill</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>C. M. Stone</td>
<td>1</td>
<td>MS 0865</td>
<td>H. G. Hudson</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>J. W. Swegle</td>
<td>2</td>
<td>MS 0865</td>
<td>D. O. Smallwood</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>M. R. Tabbara</td>
<td>1</td>
<td>MS 0865</td>
<td>J. E. Solberg</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>L. M. Taylor</td>
<td>1</td>
<td>MS 1135</td>
<td>J. R. Garcia</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>B. J. Thorne</td>
<td>1</td>
<td>MS 1135</td>
<td>R. G. Coleman</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>G. W. Wellman</td>
<td>1</td>
<td>MS 1135</td>
<td>N. T. Davie</td>
</tr>
<tr>
<td>1</td>
<td>MS 0437</td>
<td>D. R. Martinez</td>
<td>1</td>
<td>MS 1135</td>
<td>T. C. Togami</td>
</tr>
<tr>
<td>1</td>
<td>MS 0439</td>
<td>K. F. Alvin</td>
<td>1</td>
<td>MS 1392</td>
<td>V. Gabbard</td>
</tr>
<tr>
<td>1</td>
<td>MS 0439</td>
<td>J. L. Dohner</td>
<td>1</td>
<td>MS 0439</td>
<td>D. B. Longcope</td>
</tr>
<tr>
<td>1</td>
<td>MS 0439</td>
<td>C. R. Dohrmann</td>
<td>1</td>
<td>MS 0439</td>
<td>J. R. Red-Horse</td>
</tr>
<tr>
<td>1</td>
<td>MS 0439</td>
<td>B. Driessenn</td>
<td>1</td>
<td>MS 0439</td>
<td>J. M. Redmond</td>
</tr>
<tr>
<td>1</td>
<td>MS 0557</td>
<td>T. W. Simmermannacher</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MS 9018</td>
<td>Central Technical Files 8940-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MS 0899</td>
<td>Technical Library 4916</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MS 0619</td>
<td>Review and Approval Desk 12690 For DOE/OSTI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>