Quick Reference for gnuplot-mode

This card describes the features of gnuplot-mode for Emacs. Gnuplot-mode is intended for composing scripts for the GNUPLOT plotting program. It offers functions for sending commands or entire scripts to the GNUPLOT program as well as various functions to aid in composing scripts. It works with any version of GNUPLOT above 3.5. See the comments in the file `gnuplot.el' for instructions on installing gnuplot-mode.

gnuplot-mode key sequences

Gnuplot-mode buffer

key	description
`C-c C-l`	send a line to gnuplot
`C-c C-v`	send a line and move forward 1 line
`C-c C-r`	send the region to gnuplot
`C-c C-b`	send the buffer to gnuplot
`C-c C-f`	send a file to gnuplot
`M-⟨tab⟩`	complete keyword at point
`C-c C-i`	insert filename at point
`C-c C-j`	jump to next statement
`C-c C-n`	negate set option at point
`C-c C-c`	comment region
`C-c C-o`	set arguments of command at point
`S-mouse-2`	set arguments of command under mouse
`C-c C-d`	get help from the gnuplot info file
`C-c M-h`	get quick syntax help in context-sensitive mode
`C-c C-e`	look at the gnuplot process buffer
`C-c M-i`	toggle inline plot display
`C-c C-k`	kill the gnuplot process
`C-c C-u`	submit a bug report about gnuplot-mode
`C-c C-z`	customize gnuplot-mode

Gnuplot process buffer

key	description
`M-C-p`	plot script
`M-C-f`	load file containing script
`C-c C-e`	pop back to script

Starting gnuplot-mode

M-x gnuplot-mode

Start gnuplot-mode in the current buffer.

M-x gnuplot-make-buffer

Open a new buffer in gnuplot-mode

Gnuplot-mode homepage
`http://github.com/bruceravel/gnuplot-mode/`

Setting up gnuplot-mode

Put the lines in the box below in your `.emacs' file or in the system wide start-up file to enable gnuplot-mode. The first two lines make Emacs recognize the functions described in the "Starting gnuplot-mode" section on this page. The third line causes Emacs to put all files ending in `.gp' into gnuplot-mode. The final line defines a hotkey – in this case ⟨F9⟩ – for starting gnuplot-mode.

```
(autoload 'gnuplot-mode "gnuplot"
          "gnuplot major mode" t)
(autoload 'gnuplot-make-buffer "gnuplot"
          "open a buffer in gnuplot mode" t)
(setq auto-mode-alist
      (append '(("\\.gp$" . gnuplot-mode))
              auto-mode-alist))
(global-set-key [(f9)] 'gnuplot-make-buffer)
```

Using the gnuplot-process buffer

The process buffer contains an active GNUPLOT command line for interacting with GNUPLOT directly. The `M-C-p` and `M-C-f` key sequences will plot using the contents of the gnuplot script buffer.

Using the GUI to set command arguments

`C-c C-c` and `S-mouse-2` are used to invoke the graphical tool for setting command arguments. Use text fields and option menus to choose appropriate values. Menus and buttons are activated with the middle mouse button. A few plot options may not be fully supported.

Customizing variables

The graphical customization tool for variables can be invoked using `C-c C-z`. Descriptions of the variables relevant to gnuplot-mode can be obtained by using `gnuplot` as the regular expression for `M-x apropos`.

On-line help

Keyword completion and on-line help require that the GNUPLOT info file be available and that the info-look package be installed. The info file can be made from the documentation supplied with the GNUPLOT distribution and the info-look package is a standard part of Emacs 20. Users of XEmacs or Emacs 19 should download `info-look.el' from the gnuplot-mode homepage.

Using pm3d

All features of the pm3d patch to GNUPLOT should be available when using gnuplot-mode. One particularly useful feature of pm3d is the ability to push a cursor position into the clipboard. This is done by double-clicking `mouse-1` in the plot window, then doing `M-x yank-clipboard-selection` (usually bound to `mouse-2`) in the gnuplot script buffer.

User configurable variables

gnuplot-program [`gnuplot`]

The name of the gnuplot executable.

gnuplot-process-name [`*gnuplot*`]

The name of the gnuplot process and process buffer.

gnuplot-gnuplot-buffer [`plot.gp`]

The name of the gnuplot scratch buffer opened by `gnuplot-make-buffer`.

gnuplot-display-process [`'window`]

Determines how to display the gnuplot process buffer, either 'frame, 'window, or nil

gnuplot-info-display [`'window`]

Determines how 'gnuplot-get-help' displays the info file, either 'frame, 'window, or nil

gnuplot-echo-command-line-flag [`t`]

If lines that you send to gnuplot from the gnuplot-mode buffer are not appearing at the gnuplot prompt in the process buffer, set this to nil and restart emacs.

gnuplot-delay [`0.01`]

Time in seconds to allow the gnuplot display to update. Increase this number if the prompts and lines are displayed out of order.

gnuplot-quote-character [`'`]

Quotation character used when inserting a filename into the script (single, double, or no quote).

gnuplot-buffer-max-size [`1000`]

The maximum size in lines of the gnuplot buffer. Excess lines are trimmed. 0 means to never trim.

gnuplot-context-sensitive-mode [`t`]

Whether to try to do intelligent keyword completion by parsing the current script line.

gnuplot-tab-completion [`nil`]

Whether the TAB key should try to complete the symbol at point, on lines that are already indented.

gnuplot-inline-image-mode [`nil`]

Whether to display Gnuplot plot output inline in the comint process buffer as images.

gnuplot-eldoc-mode [`nil`]

Whether to enable eldoc-mode automatically in gnuplot buffers. Requires both context-sensitive-mode and some additional configuration steps.

gnuplot-basic-offset [`4`]

How many spaces to indent Gnuplot 4's block-structured `do` and `if` statements

Hook variables

gnuplot-mode-hook [`nil`]

Functions run when gnuplot minor mode is entered.

gnuplot-load-hook [`nil`]

Functions run when gnuplot.el is first loaded.

gnuplot-after-plot-hook [`nil`]

Functions run after gnuplot plots an entire buffer. See the doc string for `gnuplot-recently-sent`.

gnuplot-comint-setup-hook [`nil`]

Functions run after setting up the gnuplot process buffer in comint mode.

gnuplot-info-hook [`nil`]

Functions run before setting up info-look in the gnuplot-mode buffer.

Insertion variables

These variables control the `Insertions` pull-down menu, which can be used to insert GNUPLOT commands into the script. The various sub-menu variables can be used to customize which commands appear in the `Insertions` menu.

gnuplot-insertions-menu-flag [`t`]

Non-nil means to display the `Insertions` menu in the menubar.

gnuplot-insertions-show-help-flag [`nil`]

Non-nil means to display help from info file when using the `Insertions` menu.

gnuplot-insertions-adornments [⋆]

Contents of the `adornments` sub-menu.

gnuplot-insertions-plot-options [⋆]

Contents of the `plot-options` sub-menu.

gnuplot-insertions-terminal [⋆]

Contents of the `terminal` sub-menu.

gnuplot-insertions-x-axis [⋆]

Contents of the `x axis` sub-menu.

gnuplot-insertions-x2-axis [⋆]

Contents of the `x2 axis` sub-menu.

gnuplot-insertions-y-axis [⋆]

Contents of the `y axis` sub-menu.

gnuplot-insertions-y2-axis [⋆]

Contents of the `y2 axis` sub-menu.

gnuplot-insertions-z-axis [⋆]

Contents of the `z axis` sub-menu.

gnuplot-insertions-parametric-plots [⋆]

Contents of the `parametric plots` sub-menu.

gnuplot-insertions-polar-plots [⋆]

Contents of the `polar plots` sub-menu.

gnuplot-insertions-surface-plots [⋆]

Contents of the `surface plots` sub-menu.

Toolbar variables

These variables control the use and location of the toolbar in XEmacs. The toolbar has buttons equivalent to the key sequences `C-c C-l`, `C-c C-r`, `C-c C-b`, `C-c C-e`, and `C-c C-h`.

gnuplot-display-toolbar-flag [`nil`]

Non-nil means to display a toolbar if using XEmacs.

gnuplot-use-toolbar [`left-toolbar`]

Location of XEmacs toolbar. Valid values are `left-toolbar`, `right-toolbar`, `top-toolbar`, `bottom-toolbar`, `default-toolbar` and nil.

Set Arguments

These variables control the behavior of the graphical interface to setting command arguments. `C-c C-c` with point over a command or `S-mouse-2` with the mouse cursor over a command will cause a small frame to pop open with which you can set command arguments. Green button with bold text are bound to pup-up menus — use the mouse-2 to select an item from the menu. Grey fields are for filling in strings or numbers. Hit the [**Set Options**] button with `mouse-2` to insert command arguments into the script. You can also use the ⟨tab⟩ key to move among the widgets and ⟨ret⟩ to push the buttons.

gnuplot-gui-popup-flag [`nil`]

When non-nil an argument setting frame will pop open whenever the `Insertions` menu is used.

gnuplot-gui-plot-splot-fit-style [`'simple`]

`'simple` or `'complete` – describes the extent of the list of properties of for plot, splot, and fit in the GUI.

gnuplot-gui-frame-plist [⋆]

Property list of parameters controlling the argument setting frame. Used by XEmacs.

gnuplot-gui-frame-parameters [⋆]

List of parameters controlling the argument setting frame. Used by Emacs.

gnuplot-gui-fontname-list [⋆]

List of font available on your computer to the terminal drivers.

Faces

These are various faces defined for use with gnuplot-mode.

gnuplot-prompt-face [`firebrick`]

Color of gnuplot prompt (on a light background) in process buffer. Bold and underlined on a monochrome display.

gnuplot-gui-menu-face [`dark olive green`]

Color of menu buttons (on a light background) in the argument setting frame. Italic on a monochrome display.

gnuplot-gui-button-face [`sienna`]

Color of push buttons (on a light background) in the argument setting frame. Italic on a monochrome display.

gnuplot-gui-label-face [`dark slate blue`]

Color of buttons (on a light background) used to set label lists in the argument setting frame. Italic on a monochrome display.

Variables marked with ⋆ have default values that are too long to print here.