

Gnuplotfortran

A gnuplot API for Fortran 95

Madhusudan Singh

This is a manual for gnuplotfortran, version 0.2.2.

Copyright © 2005 Madhusudan Singh.

Copyright (C) 2005 Madhusudan Singh.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-CoverTexts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Published by Madhusudan Singh.

Table of Contents

1	Introduction	1
2	Requirements	2
3	Supported Platforms	3
3.1	A Microsoft Windows port ?	3
4	Usage	4
4.1	Installation	4
4.2	Compiling and Linking	4
4.3	Execution	4
4.4	Data Types	5
4.5	Plots for Publications	5
4.6	Global Variables of Significance	5
4.7	Debugging Parameters	6
5	Function Reference	7
5.1	Atomic Routines	7
5.1.1	GNUPLOT_INIT	7
5.1.2	GNUPLOT_CLOSE	7
5.1.3	GNUPLOT_CMD	8
5.1.4	GNUPLOT_SETSTYLE	8
5.1.5	GNUPLOT_SETSCALE	9
5.1.6	GNUPLOT_SETAXISLABEL	9
5.1.7	GNUPLOT_SETTITLE	10
5.1.8	GNUPLOT_RESETSESSION	11
5.1.9	GNUPLOT_SET	11
5.1.10	GNUPLOT_UNSET	12
5.1.11	GNUPLOT_RESET	12
5.1.12	GNUPLOT_LOAD	13
5.1.13	GNUPLOT_REPLOT	14
5.1.14	GNUPLOT_SETAXISFORMAT	14
5.1.15	GNUPLOT_SETCOORDINATESYSTEM	15
5.1.16	GNUPLOT_SETANGLEUNITS	15
5.1.17	GNUPLOT_SETRANGE	16
5.1.18	GNUPLOT_PAUSE	17
5.2	Canned Routines	17
5.2.1	GNUPLOT_PLOT2D	17
5.2.2	GNUPLOT_FITCURVE	19
5.2.3	GNUPLOT_PLOT3D	20
5.2.4	GNUPLOT_FITSURFACE	21
5.2.5	GNUPLOT_PLOTSLOPE	23

5.2.6	GNUPLOT_PLOTEQUATION2D	24
5.2.7	GNUPLOT_PLOTEQUATION3D	25
5.2.8	GNUPLOT_PLOTPOLAREQUATION	25
5.2.9	GNUPLOT_PLOTCOMPLEX	26
5.2.10	GNUPLOT_HARDCOPY	27
5.2.11	GNUPLOT_GENPDF	29
5.3	Super-canned Routines	29
5.3.1	GNUPLOT_PLOT2DSET	29
5.3.2	GNUPLOT_PLOT3DPANEL	31
6	Miscellany	33
6.1	Homepage	33
6.2	Reporting bugs	33
6.3	Feature Requests	33
6.4	Mailing List	33
6.5	Usenet	33
6.6	Contact information	33
Appendix A GNU Free Documentation License		
.....		34
A.0.1	ADDENDUM: How to use this License for your documents	40
Global Variables		41
Function Index		42
Concept Index		43

1 Introduction

Gnuplot is a freely available, command-driven graphical display tool for Unix. It compiles and works quite well on a number of Unix flavours as well as other operating systems. This library enables sending display requests to Gnuplot through simple f95 calls.

I believe that the best way to ensure prompt fixing of bugs in code and feature enhancements is to make stuff open source. I chose to release `fortranposix` and `gnuplotfortran` under the LGPL (Lesser GNU Public License). The terms of this license, unlike commercial licenses, permit you to use this library freely and unlike full GPL, permit commercial programs to link against it (please read the exact terms of the license to see what is allowed / not allowed).

In discussion that follows, reference will be made to three kinds of routines. The nomenclature adopted is by no means the best possible choice, but it serves its purpose.

1. Atomic routines

Present since ver. 0.1.

As the name suggests, these are routines that correspond in an injective fashion to single gnuplot commands.

2. Canned routines

Present since ver. 0.1.

These routines are composed of more than one gnuplot command, and collectively accomplish one logical task.

3. Super-canned routines

New in ver. 0.2.2.

These routines are bigger in scope than the canned routines, and attempt to hide the entire interaction with gnuplot away from the user, presenting a single shot function / subroutine call to the user. These are created with a view towards reducing the amount of clutter induced in a user's program by having to declare control pointers or dictating fine points of gnuplot output. The trade off here is for the user to give up some control over the exact appearance of the plots in exchange for coding simplicity.

2 Requirements

gnuplotfortran (and fortranposix), should work on any Unix like system that supports a runtime C library like glibc. User reports from a person using gnuplotfortran on IBM's XL Fortran system indicate that the AIX version of ld has some issues with it. At the time of this writing, I am not aware if that issue has been sorted out.

To summarize what we need :

- The operating system must provide popen, pclose, fflush, and some other functions.
- C object files append an underscore to the end of C library functions.
- A copy of the fortranposix library (released under LGPL - © 2004 Madhusudan Singh).
- Gnuplot of course. This library has been tested with Gnuplot versions 3.7.3 and 4.0. It may not support all the extensions in version 4.0 yet.

If you succeeded in porting gnuplotfortran and fortranposix to a new platform and/or compiler, let me know.

3 Supported Platforms

The title of this part is a misnomer. It could have been better called *Platforms where we know this works*. gnuplotfortran (and fortranposix) is known to work on the following platform / compiler combinations :

- Intel Fortran Compiler version ≥ 7.1 on Linux with gcc.
- Sun f95 compiler on Solaris 5.8 with gcc.
- gfortran on Linux with gcc.
- g95 on Linux with gcc.

3.1 A Microsoft Windows port ?

A few words are in order regarding a possible MS Windows port :

Please do **not** request a MS Windows port. I neither use MS Windows, nor do I own a MS Windows infected machine (and never will, if you get my drift). If you are interested in working on a MS Windows port (or have somehow managed to make it work with cygwin), please get in contact with me.

That out of the way, I believe that I should include some information that could help that poor fellow starting on a Windows port. Recently, I tried to get gnuplotfortran to work for a friend of mine¹. I first found that Windows XP Home apparently does not support POSIX system calls. Then we downloaded something called Windows Services for Unix². Then we discovered that it needed Windows XP Professional³ ! So, the upshot was that my friend would have needed to “upgrade” his windows computer so that he could use some functions that the sane part of the world takes for granted (all assuming that the package we downloaded included the functions we needed). This was a needless, but an expected headache (ok, my opinion of the MS products is a little jaded, in case you have not noticed). Then we looked at cygwin, which apparently does not have (I might be mistaken in this) popen and pclose functions (which gnuplotfortran depends on). Too much trouble all told. My friend then made the logical decision to shift his OS from Windows to Debian GNU Linux. I would like to flatter myself and think that gnuplotfortran brought one person over from the darkness of the Windows world, but the fact is that it was merely the straw that broke the camel’s back.

Enough of this digression.

¹ Who is now an erstwhile Windows user.

² IMO, it should be named the other way round - but I guess the folks at Microsoft are unlikely to want to imply that Unix does something that their buggy OS does not.

³ The implicit irony in that statement should be apparent to most people if they pause to think, slowly repeating that to themselves

4 Usage

First download and install fortranposix. Gnuplotfortran will **not** work without that.

4.1 Installation

1. Edit the supplied Makefile if necessary (I am always on the lookout for people who test fortranposix and gnuplotfortran out with new compilers and on new platforms - so if you are porting the libraries to *BSD, HP-UX, VMS, Mac OSX Darwin, etc. let me know so that we can spread the joy :)).
2. Run make.
3. This will generate a shared library : libgnuplotfortran.so and a few .mod files.
4. Move the shared library to some place within your LD_LIBRARY_PATH. You may want to make it available to everyone, so place it in a system wide path.
5. Move the .mod files to some standard location on your system. You may want to make these available to everyone, so place them in a system wide path.
6. Read the file gnuplot_fortran95_interfaces.f90 to learn about the usage of each function and to obtain the interface - superseded by this manual since ver. 0.2.2..
7. Append the -lgnuplotfortran -lfortranposix options to your linker and three lines to the code where you want to use the library : use datatypes, use gnuplot_module_data, use gnuplot_module. You may also need to specify an addition search path for library files (usually -L), depending upon your fortran compiler.

Done

4.2 Compiling and Linking

```
$ f95 (extraoptions) -o gnuplot_fortran95_testbench \
gnuplot_fortran95_testbench.f90 -lfortranposix -lgnuplotfortran
$ ./gnuplot_fortran95_testbench
```

If things worked out well (in other words, Waheguru is smiling on you today), the above should start off a series of semi-interactive plot directives and leave a hardcopy of one of the plots in the working directory.

4.3 Execution

During execution, gnuplotfortran will create a temporary directory in your working directory. The name of this directory (hidden - as it starts with a .) is long and complicated and formed from the hostname, timestamp, etc. The probability that it will overwrite anything in your working directory is vanishingly small. This directory is removed as soon as the session is over. However, if there is an error in the execution (whether inside a gnuplotfortran library function) or outside, this could be left behind. In that case, you will have to remove it manually if you think it is important.

Each gnuplotfortran function returns a status value, which is an integer. In general, if there is no error in the execution, the returned status value would be either zero (flawless

execution) or positive (warnings). Gnuplotfortran issues error messages with error codes which can help you locate the reason for the error.

4.4 Data Types

The basic types used for the integers, reals and complex numbers are defined as :

```
integer, parameter :: i4b = SELECTED_INT_KIND(9)
integer, parameter :: dp = KIND(1.0d0)
```

The fundamental data type used to control the gnuplot session is a derived type named `gnuplot_ctrl`. Unless you are using a super-canned routine, you must define a pointer to this type, and invoke the gnuplot session using `gnuplot_init` :

```
ptr_gctrl=gnuplot_init(argument)
```

where `argument` is the X option you wish to use with `gnuplot_init`[\[gnuplot-init\]](#), [page 7](#). Then use this pointer in the function invocations that follow.

4.5 Plots for Publications

By default, plots using gnuplotfortran include an acknowledgement string in the plots indicating the author (you), the creator (yours truly), hostname, etc. For plots that will be used for publications or any other instance where this might not be appropriate, you may suppress the output of the acknowledgment string by using the `PUB` argument in `gnuplot_hardcopy`[\[gnuplot-hardcopy\]](#), [page 27](#).

4.6 Global Variables of Significance

- **BAD_SIGNAL**

The module `gnuplot_module_data` defines an integer named `BAD_SIGNAL` that denotes certain error conditions on return from many of the functions in the library. To do error checking in your code, use the module's definition of this integer constant.

- **VALIDPLOTSTYLE**

The module `gnuplot_module_data` defines a `character(len=PLOTSTYLELENGTH)`¹ array containing currently supported plotting styles. These (as of ver. 0.2.2) include : `lines`, `points`, `linespoints`, `impulses`, `dots`, `steps`, `errorbars`, `boxes`, `boxerrorbars`, `financebars`, `candlesticks`, `vector`, and `pm3d`.

- **GNUPLOT_EXECUTABLE**

The value of this variable determines the name of the gnuplot executable. The default value set in the `gnuplot_module_data` to `gnuplot`. This value should be adequate for most standard installations, but in case someone renamed yours (and you cannot or will not put a symlink named `gnuplot` to the actual executable), you may change this parameter definition, and reinstall your library.

¹ PLOTSTYLELENGTH=12

- ACKTAG

This parameter is set to 1024. You need pay attention to this only if you are using a ridiculously large number of gnuplot tags and somehow get into conflict. This is unlikely, but I mention this for completeness.

- GP_MAX_TMP_FILES

This is a parameter that sets the maximum number of temporary files that may be open at any given time. In case of real time plotting, this might pose a constraint. In that case, consider using the routine `gnuplot_reset_session`[\[gnuplot-reset-session\]](#), [page 11](#).

- GP_CMD_SIZE

The maximum length of a gnuplot command is determined by :

$\text{length} = \min(\text{GP_CMD_SIZE}, \text{pipebuffersize})$

where *pipebuffersize* is determined by the C runtime library / operating system.

4.7 Debugging Parameters

Ordinarily, a `gnuplotfortran` function call will not return any diagnostic information about any warnings, or even echo the gnuplot session commands as they are executed. However, the module `gnuplot_module_data` contains two logical variables that can be modified by user to debug any possible problems :

- GNUPLOT_SHOWWARNINGS

If set to `.true.`, it will echo any warnings generated at run time.

- GNUPLOT_SHOWDEBUG

If set to `.true.`, it will echo all the gnuplot commands as they are piped for execution. This can be useful in tracking down some obscure errors.

5 Function Reference

5.1 Atomic Routines

5.1.1 GNUPLOT_INIT

- *INPUTS* :
 - **xoption** :
Character string that passes any X11 options to the GNUPlot session, such as -persist,-background,-clear,-raise,-noraise,-vtwm,-gray,or -mono.
- *OUTPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
- *PURPOSE* :
Initiates a `gnuplot` session.
- *DIAGNOSTICS* :
 - **ptr_gctrl** = associated : Success.
 - **ptr_gctrl** = NULL : Failure.
- *INTERFACE* :

```
interface
  function gnuplot_init(xoption) result(ptr_gctrl)
    character(len=*), intent(in) :: xoption
  end function gnuplot_init
end interface
```

5.1.2 GNUPLOT_CLOSE

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Concludes a `gnuplot` session
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -8888 : PDF conversion requested and failed.
 - **status** = -7777 : Cannot close communication pipe to `gnuplot`.
 - **status** = -1 : Could not remove temporary directory.

– *INTERFACE* :

```
interface
  integer function gnuplot_close(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
  end function gnuplot_close
end interface
```

5.1.3 GNUPLOT_CMD

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **cmd** : Character string (command to be executed in the gnuplot session).

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Executes a single line of commands in the gnuplot session via a pipe. Its use is generally *not* recommended. Use it only as a last resort if none of the other functions in the library work for you.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -8888 : Failed to write to pipe.
- **status** = -7777 : Command too long for pipe buffer.
- **status** = -6666 : Failed to flush the pipe.

– *INTERFACE* :

```
interface
  integer function gnuplot_cmd(ptr_gctrl,cmd) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: cmd
  end function gnuplot_cmd
end interface
```

5.1.4 GNUPLOT_SETSTYLE

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **plotstyle** : Character string (plot style - see [\[VALIDPLOTSTYLE\]](#), page 5).

– *OUTPUTS* :

- **status** : Integer (error flag).

- *PURPOSE* :
Sets the plotting style of a gnuplot session.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = 22 : Invalid plotting style requested, replaced with points.
- *INTERFACE* :

```
interface
  integer function gnuplot_setstyle(ptr_gctrl,plotstyle) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: plotstyle
  end function gnuplot_setstyle
end interface
```

5.1.5 GNUPLOT_SETSCALE

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
 - **axis** : Character string indicating the axis, such as x,x2,y,y2,z,z2,v,t,u, etc.
 - **scale** : 3-character string (choices - 'LOG' or 'NLG').
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Sets scaling along different axes.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -1 : Cannot set requested scaling for the specified axis.
- *INTERFACE* :

```
interface
  integer function gnuplot_setscale(ptr_gctrl,axis,scale) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: axis
    character(len=3), intent(in) :: scale
  end function gnuplot_setscale
end interface
```

5.1.6 GNUPLOT_SETAXISLABEL

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
 - **axis** : Character string indicating the axis.
 - **axislabel** : Character string indicating the label.
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Sets axis label.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = `BAD_SIGNAL` : Called with a dissociated pointer.
 - **status** = -1 : Cannot set requested label for the axis.
- *INTERFACE* :

```
interface
  integer function gnuplot_setaxislabel(ptr_gctrl,axis,axislabel) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: axis,axislabel
  end function gnuplot_setaxislabel
end interface
```

5.1.7 GNUPLOT_SETTITLE

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
 - **title** : Character string (title for the plot).
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Sets the plot title
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = `BAD_SIGNAL` : Called with a dissociated pointer.
 - **status** = -1 : Cannot set title.

– *INTERFACE* :

```
interface
  integer function gnuplot_settitle(ptr_gctrl,title) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: title
  end function gnuplot_settitle
end interface
```

5.1.8 GNUPLOT_RESESSSION

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Resets the session¹. Gnuplotfortran internally keeps a count of number of plots currently active. This function sets that count to zero. This might be useful in a situation where real time data is being plotted, and the user is certain that a large number of user plots do not correspond to a calling program that is in a runaway state. If you are looking for a way to issue the **reset** command, see [\[gnuplot-reset\]](#), page 13.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = 22 : Failed to delete temporary file.

– *INTERFACE* :

```
interface
  integer function gnuplot_resetsession(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
  end function gnuplot_resetsession
end interface
```

5.1.9 GNUPLOT_SET

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **setstring** : Character string.

– *OUTPUTS* :

¹ This function is not truly an atomic function, as it does not correspond to any gnuplot session action.

- **status** : Integer (error flag)
- *PURPOSE* :
Sets a gnuplot variable. Can be used for any character string that needs to be prepended with a **set** command.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -1 : Cannot set with requested string.
- *INTERFACE* :

```
interface
  integer function gnuplot_set(ptr_gctrl,setstring) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: setstring
  end function gnuplot_set
end interface
```

5.1.10 GNUPLOT_UNSET

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
 - **setstring** : Character string.
- *OUTPUTS* :
 - **status** : Integer (error flag)
- *PURPOSE* :
Unsets a gnuplot variable. Can be used for any character string that needs to be prepended with a **unset** command.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -1 : Cannot unset with requested string.
- *INTERFACE* :

```
interface
  integer function gnuplot_unset(ptr_gctrl,unsetstring) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: unsetstring
  end function gnuplot_unset
end interface
```


5.1.11 GNUPLOT_RESET

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.

– *OUTPUTS* :

- **status** : Integer (error flag)

– *PURPOSE* :

Resets the parameters for the current session. Issues the `gnuplot reset` command.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -1 : Failed to reset parameters.

– *INTERFACE* :

```
interface
  integer function gnuplot_reset(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
  end function gnuplot_reset
end interface
```

5.1.12 GNUPLOT_LOAD

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
- **loadfile** : Character string : name of the load file

– *OUTPUTS* :

- **status** : Integer (error flag)

– *PURPOSE* :

Runs a `gnuplot` load file.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -1 : Cannot find the specified load file.
- **status** = -2 : You do not have read permissions for the load file.
- **status** = -3 : Cannot run the specified load file.

- *INTERFACE* :

```
interface
  integer function gnuplot_load(ptr_gctrl,loadfile) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: loadfile
  end function gnuplot_load
end interface
```

5.1.13 GNUPLOT_REPLOT

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- *OUTPUTS* :
 - **status** : Integer (error flag)
- *PURPOSE* :
Sends a replot command to the session.
- *DIAGNOSTICS* :
 - **status** = 0 : Success
 - **status** = BAD_SIGNAL : Called with a dissociated pointer
 - **status** = -1 : Failed to replot
- *INTERFACE* :

```
interface
  integer function gnuplot_replot(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
  end function gnuplot_replot
end interface
```

5.1.14 GNUPLOT_SETAXISFORMAT

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
 - **axis** : Character string indicating the axis
 - **axisformat** : Character string indicating the format
- *OUTPUTS* :
 - **status** : Integer (error flag)
- *PURPOSE* :
Sets the format for an axis.
- *DIAGNOSTICS* :

- **status** = 0 : Success
- **status** = BAD_SIGNAL : Called with a dissociated pointer
- **status** = -1 : Failed to set format for the specified axis

– *INTERFACE* :

```
interface
  integer function gnuplot_setaxisformat(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: axis,axisformat
  end function gnuplot_setaxisformat
end interface
```

5.1.15 GNUPLOT_SETCOORDINATESYSTEM

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **coordsystem** : Character string indicating the kind of coordinate system (polar or nopolar)

– *OUTPUTS* :

- **status** : Integer (error flag)

– *PURPOSE* :

Sets the coordinate system.

– *DIAGNOSTICS* :

- **status** = 0 : Success
- **status** = BAD_SIGNAL : Called with a dissociated pointer
- **status** = -1 : Cannot set requested coordinate system

– *INTERFACE* :

```
interface
  integer function gnuplot_setcoordsystem(ptr_gctrl,coordsystem) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: coordsystem
  end function gnuplot_setcoordsystem
end interface
```

5.1.16 GNUPLOT_SETANGLEUNITS

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **angleunits** : Character string indicating the units of angle measurement (radians or degrees) - this may effect some transcendental functions - consult gnuplot help.

- *OUTPUTS* :
 - **status** : Integer (error flag)
- *PURPOSE* :
Sets the angle units.
- *DIAGNOSTICS* :
 - **status** = 0 : Success
 - **status** = BAD_SIGNAL : Called with a dissociated pointer
 - **status** = -1 : Cannot set requested angle units
- *INTERFACE* :

```
interface
  integer function gnuplot_setangleunits(ptr_gctrl,angles) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: angleunits
  end function gnuplot_setangleunits
end interface
```

5.1.17 GNUPLOT_SETRANGE

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
 - **axis** : Character string indicating the axis.
 - **lo** : Real (lower limit of the range).
 - **hi** : Real (upper limit of the range).
- *OUTPUTS* :
 - **status** : Integer (error flag)
- *PURPOSE* :
Sets axes ranges.
- *DIAGNOSTICS* :
 - **status** = 0 : Success
 - **status** = BAD_SIGNAL : Called with a dissociated pointer
 - **status** = -1 : Cannot set requested range for the axis
 - **status** = -8888 : Invalid range specification
- *INTERFACE* :

```

interface
  integer function gnuplot_setrange(ptr_gctrl,axis,lo,hi) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: axis
    real(dp), intent(in) :: lo,hi
  end function gnuplot_setrange
end interface

```

5.1.18 GNUPLOT_PAUSE

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **t** : Integer time for which to pause.
- **displaystring** : Character string to be displayed (optional).

– *OUTPUTS* :

- **status** : Integer (error flag)

– *PURPOSE* :

Pauses the session. Unlikely to be of much use in a typical program, but included for completeness.

– *DIAGNOSTICS* :

- **status** = 0 : Success
- **status** = BAD_SIGNAL : Called with a dissociated pointer
- **status** = -1 : Cannot pause

– *INTERFACE* :

```

interface
  integer function gnuplot_pause(ptr_gctrl,t,displaystring) &
  result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: t
    character(len=*), intent(in), optional :: displaystring
  end function gnuplot_pause
end interface

```

5.2 Canned Routines

5.2.1 GNUPLOT_PLOT2D

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **n** : Integer (number of points to be plotted).

- **x** : Real, dimension(n) (abscissae).
 - **y1** : Real, dimension(n) (first ordinate).
 - **y1title** : Character string (title for y1) (optional).
 - **y1axis** : Character string (axis for y1) (optional).
 - **y2** : Real, dimension(n) (second ordinate) (optional).
 - **y2title** : Character string (title for y2) (optional).
 - **y2axis** : Character string (axis for y2) (optional).
 - **y3** : Real, dimension(n) (third ordinate) (optional).
 - **y3title** : Character string (title for y3) (optional).
 - **y3axis** : Character string (axis for y3) (optional).
 - **y4** : Real, dimension(n) (fourth ordinate) (optional).
 - **y4title** : Character string (title for y4) (optional).
 - **y4axis** : Character string (axis for y4) (optional).
- *OUTPUTS* :
- **status** : Integer (error flag)
- *PURPOSE* :
- Plots x,y1,y2,y3,y4 together (max. of 4 plots possible).
- *DIAGNOSTICS* :
- **status** = 0 : Success
 - **status** = BAD_SIGNAL : Called with a dissociated pointer
 - **status** = -3 : Called with less than one point
 - **status** = -1 : Impossible error
 - **status** = -2 : Cannot create temporary file
 - **status** = 22 : Failed to delete temporary file
 - **status** = -4 : Failed to write to temporary file
 - **status** = -5 : Failed to open temporary file in temporary directory
 - **status** = -6 : Failed to execute the plot / replot command
 - **status** = -7 : Invalid y axis specification supplied
- *INTERFACE* :

```

interface
  integer function gnuplot_plot2d(ptr_gctrl,n,x,y1,y1title,&
    y1axis,y2,y2title,y2axis,y3,y3title,y3axis,y4,y4title,y4axis) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    real(dp), dimension(:), intent(in) :: x,y1
    real(dp), dimension(:), intent(in), optional :: y2,y3,y4
    integer(i4b), intent(in) :: n
    character(len=*), intent(in), optional :: y1title,y2title,&
      y3title,y4title
    character(len=*), intent(in), optional :: y1axis,y2axis,&
      y3axis,y4axis
  end function gnuplot_plot2d
end interface

```

5.2.2 GNUPLOT_FITCURVE

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **n** : Integer (number of points to be plotted).
- **np** : Integer (number of adjustable parameters).
- **x** : Real, dimension(n) (abscissae).
- **y** : Real, dimension(n) (ordinates).
- **ffit** : Character string : function to be fit (must be the RHS of the expression $f(x)=\dots$).
- **vialist** : Character(len=1), dimension(np) : the via list (like 'a,b,c' (constants used in ffit)
- **paraminit** : Real, dimension(np) : array containing initial values for the parameters (optional).
- **fitlimit** : Real : FIT_LIMIT , default is usually 1e-6 (depends upon your gnuplot installation) (optional).
- **fitlog** : Character string : filename of the fitting log (optional)
- **plotresult** : Integer : whether to plot the result (=1) or not (=0) with the data (optional).
- **xlo** : Real xrange low (optional).
- **xhi** : Real xrange high (optional).

– *OUTPUTS* :

- **status** : Integer (error flag)

– *PURPOSE* :

Fits $f(x)$ to supplied y vs x and plots the result if desired.

– *DIAGNOSTICS* :

- **status** = 0 : Success
- **status** = BAD_SIGNAL : Called with a dissociated pointer.

- **status** = -3 : Called with less than one point.
- **status** = -1 : Impossible error.
- **status** = -2 : Cannot create temporary file.
- **status** = 22 : Failed to delete temporary file.
- **status** = -4 : Failed to write to temporary file.
- **status** = -5 : No function specified.
- **status** = -7 : Invalid function specification.
- **status** = -8 : Invalid plot directive.
- **status** = 23 : Specified fit log file will overwrite an existing file.
- **status** = -9 : Failed to open temporary file in temporary directory.
- **status** = -10 : Could not execute the fit.
- **status** = -11 : Could not plot the result.
- **status** = -12 : Wrong initial paraminit array size.
- **status** = -13 : Wrong vialist size.
- **status** = -14 : Invalid input xrange.

– *INTERFACE* :

```
interface
  integer function gnuplot_fitcurve(ptr_gctrl,n,np,x,y,ffit,&
    vialist,paraminit,xlo,xhi,fitlimit,fitlog,plotresult) &
    result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    real(dp), dimension(:), intent(in) :: x,y
    character(len=*), intent(in) :: ffit
    character(len=1), dimension(:) :: vialist
    real(dp), intent(in), optional :: fitlimit
    integer(i4b), intent(in) :: n,np
    integer(i4b), intent(in), optional :: plotresult
    real(dp), dimension(:), intent(in), optional :: paraminit
    character(len=*), intent(in), optional :: fitlog
  end function gnuplot_fitcurve
end interface
```

5.2.3 GNUPLOT_PLOT3D

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
- **n** : Integer (number of points along x axis to be plotted).
- **m** : Integer (number of points along y axis to be plotted).
- **x** : Real, dimension(n) (abscissae).
- **y** : Real, dimension(m) (abscissae).
- **z1** : Real, dimension(n X m) (first surface).

- **z1title** : Character string (title for z1) (optional).
 - **z2** : Real, dimension(n X m) (second surface) (optional).
 - **z2title** : Character string (title for z2) (optional).
 - **z3** : Real, dimension(n X m) (third surface) (optional).
 - **z3title** : Character string (title for z3) (optional).
 - **z4** : Real, dimension(n X m) (fourth surface) (optional).
 - **z4title** : Character string (title for z4) (optional).
- *OUTPUTS* :
- **status** : Integer (error flag)
- *PURPOSE* :
- Plots x,y,z1,z2,z3,z4 together(max. of 4 surface plots possible).
- *DIAGNOSTICS* :
- **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -3 : Called with less than one point.
 - **status** = -1 : Impossible error.
 - **status** = -2 : Cannot create temporary file.
 - **status** = 22 : Failed to delete temporary file.
 - **status** = -4 : Failed to write to temporary file.
 - **status** = 24 : Failed to unset previously set acknowledgement string.
 - **status** = 23 : Failed to set acknowledgement string.
 - **status** = -5 : Failed to open temporary file in temporary directory.
 - **status** = -6 : Failed to execute the splot / replot command.
- *INTERFACE* :

```

interface
  integer function gnuplot_plot3d(ptr_gctrl,n,m,x,y,z1,z1title,&
    z2,z2title,z3,z3title,z4,z4title) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    real(dp), dimension(:), intent(in) :: x,y
    real(dp), dimension(:,,:), intent(in) :: z1
    real(dp), dimension(:,,:), intent(in), optional :: z2,z3,z4
    integer(i4b), intent(in) :: n,m
    character(len=*), intent(in), optional :: z1title,z2title,&
      z3title,z4title
  end function gnuplot_plot3d
end interface

```

5.2.4 GNUPLOT_FITSURFACE

- *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
 - **n** : Integer (number of points to be plotted along x).
 - **m** : Integer (number of points to be plotted along y).
 - **np** : Integer (number of adjustable parameters).
 - **x** : Real, dimension(n) (abscissae).
 - **y** : Real, dimension(m) (abscissae).
 - **z** : Real, dimension(n X m) (ordinate).
 - **gfit** : Character string : function to be fit (must be the RHS of the expression $g(x,y)=\dots$).
 - **vialist** : Character(len=1), dimension(np) : the via list (like 'a,b,c' (constants used in fit)).
 - **paraminit** : Real, dimension(np) : array containing initial values for the parameters (optional).
 - **fitlimit** : Real : FIT_LIMIT , default is usually 1e-6 (depends upon your gnuplot installation) (optional).
 - **fitlog** : Character string : filename of the fitting log (optional).
 - **plotresult** : Integer whether to plot the result (=1) or not (=0) with the data (optional).
 - **xlo** : Real : xrange low (optional).
 - **xhi** : Real : xrange high (optional).
 - **ylo** : Real : yrange low (optional).
 - **yhi** : Real : yrange high (optional).
- *OUTPUTS* :
- **status** : Integer (error flag).
- *PURPOSE* :
- Fits $g(x,y)$ to supplied z vs x,y and plots the result if desired.
- *DIAGNOSTICS* :
- **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -3 : Called with less than one point.
 - **status** = -1 : Impossible error.
 - **status** = -2 : Cannot create temporary file.
 - **status** = 22 : Failed to delete temporary file.
 - **status** = -4 : Failed to write to temporary file.
 - **status** = -5 : No function specified.
 - **status** = -7 : Invalid function specification.
 - **status** = -8 : Invalid plot directive.
 - **status** = 23 : Specified fit log file will overwrite an existing file.
 - **status** = -9 : Failed to open temporary file temporary directory.

- **status** = -10 : Could not execute the fit.
- **status** = -11 : Could not plot the result.
- **status** = -12 : Wrong initial paraminit array size.
- **status** = -13 : Wrong vialist size.
- **status** = -14 : Invalid input x/y range.
- **status** = 24 : Failed to unset previously set acknowledgement string.
- **status** = 25 : Failed to set acknowledgement string.

– *INTERFACE* :

```
interface
  integer function gnuplot_fitsurface(ptr_gctrl,n,m,np,x,y,z,&
    gfit,vialist,paraminit,xlo,xhi,ylo,yhi,fitlimit,fitlog,&
    plotresult) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    real(dp), dimension(:), intent(in) :: x,y
    real(dp), dimension(:,,:), intent(in) :: z
    character(len=*), intent(in) :: gfit
    character(len=1), dimension(:) :: vialist
    real(dp), intent(in), optional :: fitlimit
    integer(i4b), intent(in) :: n,m,np
    integer(i4b), intent(in), optional :: plotresult
    real(dp), dimension(:), intent(in), optional :: paraminit
    character(len=*), intent(in), optional :: fitlog
  end function gnuplot_fitsurface
end interface
```

5.2.5 GNUPLOT_PLOTSLOPE

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **a** : Real (in $a*x+b$).
- **b** : Real (in $a*x+b$).
- **xlo** : Real (lower limit of xrange) (optional) ignored if a previous plot exists.
- **xhi** : Real (upper limit of xrange) (optional) ignored if a previous plot exists.
- **slopetitle** : Character string (title for slope).

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Plots the specified slope line.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.

- **status** = -1 : Called with invalid range (if present).
- **status** = -2 : Failed to plot slope.
- **status** = -4 : Missing one of the two xrange specifications (if present).

– *INTERFACE* :

```
interface
  integer function gnuplot_plotslope(ptr_gctrl,a,b,slopetitle,&
  xlo,xhi) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    real(dp), intent(in) :: a,b
    real(dp), intent(in), optional :: xlo,xhi
    character(len=*), intent(in) :: slopetitle
  end function gnuplot_plotslope
end interface
```

5.2.6 GNUPLOT_PLOTEQUATION2D

– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- **equationstring** : Character string (string defining the function to be plotted).
- **equationtitle** : Character string (title for equation).
- **xlo** : Real (lower limit of xrange) (optional) ignored if a previous plot exists.
- **xhi** : Real (upper limit of xrange) (optional) ignored if a previous plot exists.

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Plots a specified equation in two-dimensions.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -1 : Called with invalid range (if present).
- **status** = -2 : Failed to plot equation.
- **status** = -4 : Missing one of the two xrange specifications (if present).

– *INTERFACE* :

```

interface
  integer function gnuplot_plotequation2d(ptr_gctrl,&
    equationstring,equationtitle,xlo,xhi) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*) :: equationstring
    real(dp), intent(in), optional :: xlo,xhi
    character(len=*), intent(in) :: equationtitle
  end function gnuplot_plotequation2d
end interface

```

5.2.7 GNUPLOT_PLOTEQUATION3D

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
 - **equationstring** : Character string (string defining the function to be plotted).
 - **equationtitle** : Character string (title for equation).
 - **xlo** : Real (lower limit of xrange) (optional) ignored if a previous plot exists.
 - **xhi** : Real (upper limit of xrange) (optional) ignored if a previous plot exists.
 - **ylo** : Real (lower limit of yrange) (optional) ignored if a previous plot exists.
 - **yhi** : Real (upper limit of yrange) (optional) ignored if a previous plot exists.
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Plots a specified equation in three dimensions.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -1 : Called with invalid range (if present).
 - **status** = -2 : Failed to plot equation.
 - **status** = -4 : Missing one of the two x/y range specifications (if present).
 - **status** = 24 : Failed to unset previously set acknowledgement string.
 - **status** = 23 : Failed to set acknowledgement string.
- *INTERFACE* :

```

interface
  integer function gnuplot_plotequation3d(ptr_gctrl,&
    equationstring,equationtitle,xlo,xhi,ylo,yhi) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*) :: equationstring
    real(dp), intent(in), optional :: xlo,xhi,ylo,yhi
    character(len=*), intent(in) :: equationtitle
  end function gnuplot_plotequation3d
end interface

```

5.2.8 GNUPLOT_PLOTPOLAREQUATION

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
 - **equationstring** : Character string (string defining the function to be plotted).
 - **equationtitle** : Character string (title for equation).
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :
Plots a specified equation $r=f(t)$ in polar coordinates - invoke `gnuplot_setcoordinatesystem` first to set the polar coordinate system.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -2 : Failed to plot equation.
- *INTERFACE* :

```
interface
  integer function gnuplot_plotpolarequation(ptr_gctrl,&
    equationstring,equationtitle) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*) :: equationstring
    character(len=*), intent(in) :: equationtitle
  end function gnuplot_plotpolarequation
end interface
```

5.2.9 GNUPLOT_PLOTCOMPLEX

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
 - **n** : Integer (number of points to be plotted).
 - **c1** : Complex, dimension(n).
 - **c1title** : Character string (title for c1) (optional).
 - **c2** : Complex, dimension(n) (optional).
 - **c2title** : Character string (title for c2) (optional).
 - **c3** : Complex, dimension(n) (optional).
 - **c3title** : Character string (title for c3) (optional).
 - **c4** : Complex, dimension(n) (optional).
 - **c4title** : Character string (title for c4) (optional).
- *OUTPUTS* :
 - **status** : Integer (error flag).

– *PURPOSE* :

Polar complex plots c_1, c_2, c_3, c_4 plot (max of 4 plots possible) in the z plane

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -3 : Called with less than one point.
- **status** = -1 : Impossible error.
- **status** = -2 : Cannot create temporary file.
- **status** = -4 : Cannot allocate an internal array.
- **status** = -5 : Cannot deallocate an internal array.
- **status** = 22 : Failed to delete temporary file.
- **status** = -6 : Failed to write to temporary file.
- **status** = -7 : Failed to open temporary file in temporary directory.
- **status** = -8 : Failed to run plot / replot command.

– *INTERFACE* :

```
interface
  integer function gnuplot_plotcomplex(ptr_gctrl,n,c1,c1title,&
    c2,c2title,c3,c3title,c4,c4title) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    complex(dp), dimension(:), intent(in) :: c1
    complex(dp), dimension(:), intent(in), optional :: c2,c3,c4
    integer(i4b), intent(in) :: n
    character(len=*), intent(in), optional :: c1title,c2title,&
      c3title,c4title
  end function gnuplot_plotcomplex
end interface
```

5.2.10 GNUPLOT_HARDCOPY– *INPUTS* :

- **ptr_gctrl** : Pointer to structure of type `gnuplot_ctrl`.
- **plotfileformat** : Character string (format of the requested file).
- **plotfilename** : Character string (file name for the plot file).
- **extraarguments** : Character string (any user specified commands that come after "set terminal <terminal-type>", such as the enhanced mode for PostScript) (optional).
- **pubflag** : 3 character string (optional) (if set to 'PUB', the acknowledgement and date string will be suppressed).

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Produces a hard(soft? :))copy of the plot in a user specified format. Valid file formats are :

- a 'PS' : PostScript.
- a 'EPS' : Encapsulated PostScript.
- a 'PDF' : Portable Document Format.
- a 'PNG' : Portable Network Graphics.
- a 'LATEX' : LaTeX format.
- a 'PSLATEX' : PS LaTeX format.
- a 'EPSLATEX' : EPS LaTeX format.
- a 'PSTRICKS' : Output suitable for use with Timothy Von Zandt's PStricks package.
- a 'HPGL' : HP Graphics Language.
- a 'HPDJ' : HP Desk Jet format.
- a 'FIG' : Fig format.
- a 'DXF' : AutoCad format.
- a 'EPPIC' : LaTeX extension EEPIC format.

in addition to x11, which of course does not generate any "hardcopy" :)

This function must be called immediately after `gnuplot_init` and before any plotting or set commands. In case an unknown format is chosen, the function reverts to PostScript. If you choose the PDF format, you must have the `epstopdf` command available in your `$PATH` variable. The actual conversion from intermediary ps to pdf takes place in `gnuplot_genpdf`². So `gnuplot_genpdf` must be called after this to ensure that the PDF file is generated. The function does no file extension checking, so make sure that you supply the correct output filename.

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = 22 : Unknown format requested.
- **status** = -2 : Failed to set requested (valid) terminal type.
- **status** = -3 : Failed to set output file.
- **status** = 23 : Failed to set acknowledgement string.
- **status** = 24 : Failed to unset prior acknowledgement label.

– *INTERFACE* :

² Gnuplot ver. 4.0 introduced the `pdf` terminal, so this might change in a future version of `gnuplotfortran`.


```

interface
  integer function gnuplot_hardcopy(ptr_gctrl,plotfileformat,&
    plotfilename,extraarguments,pubflag) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
    character(len=*), intent(in) :: plotfileformat,plotfilename
    character(len=*), intent(in), optional :: extraarguments
    character(len=3), intent(in), optional :: pubflag
  end function gnuplot_hardcopy
end interface

```

5.2.11 GNUPLOT_GENPDF

- *INPUTS* :
 - **ptr_gctrl** : Pointer to structure of type gnuplot_ctrl.
- *OUTPUTS* :
 - **status** : Integer (error flag).
- *PURPOSE* :

Produces a pdf - must be called after all the plot commands you want included have been executed.
- *DIAGNOSTICS* :
 - **status** = 0 : Success.
 - **status** = BAD_SIGNAL : Called with a dissociated pointer.
 - **status** = -1 : Cannot generate PDF from the PS file..
 - **status** = 22 : Called when the o/p format was not PDF.
- *INTERFACE* :

```

interface
  integer function gnuplot_genpdf(ptr_gctrl) result(status)
    type(gnuplot_ctrl), pointer :: ptr_gctrl
  end function gnuplot_genpdf
end interface

```

5.3 Super-canned Routines

New in ver. 0.2.2, these routines are different in character from the functions discussed earlier. These require only single shot interaction with the calling program, thereby considerably simplifying the use of gnuplotfortran³. There is some loss in control over the exact appearance of the plot, but the formatting is chosen with some common sense to permit as much conventional look as well as flexibility as is possible.

³ It is not unusual for a fortran 95 to contain 10-15 lines of gnuplotfortran specific code for each plot invocation.

5.3.1 GNUPLOT_PLOT2DSET

– *INPUTS* :

- **pf** : Character string (plot file name).
- **pformat** : Character string (format of hardcopy).
- **commontitle** : Character string (title used for the whole plot).
- **xarray** : Real, dimension(n) (x array).
- **xlabel** : Character string (x label).
- **ylabel** : Character string (label for the y1 axis).
- **y2label** : Character string (label for the y2 axis) (optional).
- **yarray** : Real, dimension(n) (y array).
- **yaxis** : Character string (len=2) (axis used - y1 or y2) (optional).
- **ykey** : Character string (ykey).
- **yarray1** : Real, dimension(n) (yarray1) (optional).
- **yaxis1** : Character string (len=2) (axis used - y1 or y2) (optional).
- **ykey1** : Character string (ykey1) (optional).
- **yarray2** : Real, dimension(n) (yarray2) (optional).
- **yaxis2** : Character string (len=2) (axis used - y1 or y2) (optional).
- **ykey2** : Character string (ykey2) (optional).
- **yarray3** : Real, dimension(n) (yarray3) (optional).
- **yaxis3** : Character string (len=2) (axis used - y1 or y2) (optional).
- **ykey3** : Character string (ykey3) (optional).
- **logy1** : Character string (len=3) (log scale or not – for y1 axis) (optional).
- **logy2** : Character string (len=3) (log scale or not – for y2 axis) (optional).

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Plots x,y,y1,y2,y3 together (max. of 4 line plots possible).

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -1 : Array size mismatch.

– *INTERFACE* :

```

interface
  integer function gnuplot_plot2dset(pf,pformat,commontitle,&
    xarray,xlabel,ylabel,y2label,yarray,yaxis,ykey,yarray1,&
    yaxis1,ykey1,yarray2,yaxis2,ykey2,yarray3,yaxis3,ykey3,&
    logy1,logy2) result(status)
    character(len=*), intent(in) :: pf,pformat,commontitle
    real(dp), dimension(:), intent(in) :: xarray,yarray
    character(len=*), intent(in) :: xlabel,ylabel,ykey
    real(dp), dimension(:), intent(in), optional :: yarray1,&
    yarray2,yarray3
    character(len=*), intent(in), optional :: y2label,ykey1,&
    ykey2,ykey3
    character(len=*), intent(in), optional :: yaxis,yaxis1,&
    yaxis2,yaxis3
    character(len=*), intent(in), optional :: logy1,logy2
  end function gnuplot_plot2dset
end interface

```

5.3.2 GNUPLOT_PLOT3DPANEL

– *INPUTS* :

- **pf** : Character string (plot file name).
- **pformat** : Character string (format of hardcopy).
- **commontitle** : Character string (used for the whole plot).
- **xarray** : Real, dimension(n) (x array).
- **xlabel** : Character string (x label).
- **yarray** : Real, dimension(m) (y array).
- **ylabel** : Character string (y label).
- **zarray** : Real, dimension (n X m) (z array).
- **zlabel** : Character string (z label).
- **zarray1** : Real, dimension (n X m) (z1 array) (optional).
- **zlabel1** : Character string (z1 label) (optional).
- **zarray2** : Real, dimension (n X m) (z2 array) (optional).
- **zlabel2** : Character string (z2 label) (optional).
- **zarray3** : Real, dimension (n X m) (z3 array) (optional).
- **zlabel3** : Character string (z3 label) (optional).
- **logz** : Character string (len=3) (log scale or not) (optional).

– *OUTPUTS* :

- **status** : Integer (error flag).

– *PURPOSE* :

Plots x,y,z,z1,z2,z3 in a panel (max. of 4 surface plots possible).

– *DIAGNOSTICS* :

- **status** = 0 : Success.
- **status** = BAD_SIGNAL : Called with a dissociated pointer.
- **status** = -1 : Array size mismatch.

– *INTERFACE* :

```
interface
  integer function gnuplot_plot3dpanel(pf,pformat,commontitle,&
    xarray,xlabel,yarray,ylabel,zarray,zlabel,zarray1,zlabel1,&
    zarray2,zlabel2,zarray3,zlabel3,logz) result(status)
    character(len=*), intent(in) :: pf,pformat,commontitle
    real(dp), dimension(:), intent(in) :: xarray,yarray
    character(len=*), intent(in) :: xlabel,ylabel,zlabel
    real(dp), dimension(:,:), intent(in) :: zarray
    real(dp), dimension(:,:), intent(in), optional :: zarray1,&
    zarray2,zarray3
    character(len=*), intent(in), optional :: zlabel1,zlabel2,&
    zlabel3
    character(len=*), intent(in), optional :: logz
  end function gnuplot_plot3dpanel
end interface
```

6 Miscellany

6.1 Homepage

The project homepage is located at <http://gnuplotfortran.sourceforge.net>.

6.2 Reporting bugs

You can use the bug reporting interface at sourceforge (see homepage). Or informally inform me via a posting on [comp.lang.fortran](#).

6.3 Feature Requests

I am certain that more routines can be added to gnuplotfortran (especially the super-canned kind). So, you may use the feature request interface at sourceforge. Or use the fortran newsgroup.

6.4 Mailing List

There is a mailing list at sourceforge, but in my experience, there is **never** a reason to use a mailing list when the Usenet exists. Spamming a thousand people with a letter in their mailbox when you can pose a question non-intrusively on any one of the newsgroups and invite an informed response is one of those “bright” ideas, whose origin is a little obscure to me.

6.5 Usenet

- [comp.lang.fortran](#)
- [comp.graphics.apps.gnuplot](#)

6.6 Contact information

I cannot imagine a circumstance in which you will need to contact me directly via email instead of using the Usenet or any of the other methods suggested above. But for the record, here it is :

msc@ieee.org

Appendix A GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none. The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and

that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called

an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.0.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Global Variables

A		GNUPLOT_EXECUTABLE	5
ACKTAG	6	GNUPLOT_SHOWDEBUG	6
		GNUPLOT_SHOWWARNINGS	6
B		GP_CMD_SIZE	6
BAD_SIGNAL	5	GP_MAX_TMP_FILES	6
		V	
G		VALIDPLOTSTYLE	5

Function Index

<code>gnuplot_close</code>	7	<code>gnuplot_plotpolarequation</code>	26
<code>gnuplot_cmd</code>	8	<code>gnuplot_plotslope</code>	23
<code>gnuplot_fitcurve</code>	19	<code>gnuplot_replot</code>	14
<code>gnuplot_fitsurface</code>	21	<code>gnuplot_reset</code>	13
<code>gnuplot_genpdf</code>	29	<code>gnuplot_resetsession</code>	11
<code>gnuplot_hardcopy</code>	27	<code>gnuplot_set</code>	11
<code>gnuplot_init</code>	7	<code>gnuplot_setangleunits</code>	15
<code>gnuplot_load</code>	13	<code>gnuplot_setaxisformat</code>	14
<code>gnuplot_pause</code>	17	<code>gnuplot_setaxislabel</code>	10
<code>gnuplot_plot2d</code>	17	<code>gnuplot_setcoordinatesystem</code>	15
<code>gnuplot_plot2dset</code>	30	<code>gnuplot_setrange</code>	16
<code>gnuplot_plot3d</code>	20	<code>gnuplot_setscale</code>	9
<code>gnuplot_plot3dpanel</code>	31	<code>gnuplot_setstyle</code>	8
<code>gnuplot_plotcomplex</code>	26	<code>gnuplot_settitle</code>	10
<code>gnuplot_plotequation2d</code>	24	<code>gnuplot_unset</code>	12
<code>gnuplot_plotequation3d</code>	25		

Concept Index

A

Acknowledgment tag	6
Angle units, setting	16
Axis label, setting	10

C

Coordinate system, setting	15
----------------------------------	----

D

data types	5
Debugging control	6

E

execution	4
-----------------	---

F

FDL, GNU Free Documentation License	34
Fitting, three-dimensional	22
Fitting, two-dimensional	19

G

global variables, significant	5
Gnuplot executable	5
gnuplot load file, running	13
Gnuplot session, concluding	7
Gnuplot session, initiating	7
gnuplot variables, setting	12
gnuplot variables, unsetting	12

I

Introduction	1
--------------------	---

M

Maximum length of a gnuplot command	6
Maximum temporary files	6

P

platforms, supported	3
Plot title, setting	10
Plotting a panel, three-dimensional	31
Plotting a set, two-dimensional	30
Plotting equations, polar	26
Plotting equations, three-dimensions	25
Plotting equations, two-dimensions	24
Plotting style, setting	9
Plotting styles, supported	5
Plotting, complex numbers	27
Plotting, slope	23
Plotting, three-dimensional	21
Plotting, two-dimensional	18
port, Microsoft Windows	3
Producing a hard copy	28
Producing a hard copy, PDF	28, 29
Publications	5

R

Ranges, setting	16
reference, function	7
replotting	14
Requirements	2
Resetting, number of plots	11

S

session reset	13
Session, pausing	17
Setting scaling	9
Single command, execution; direct control	8

U

Usage	4
-------------	---