Overview of Gri 2.8 Commands

1 Introduction

This reference card describes the commands in version 2.8 of the Gri plotting language. See also the companion "Gri Reference Card" and the online manuals.

2 Control Statements

2.1 If Statements

The `if` statement has ancillary `else if` and `else` statements, and is ended by the `end if` statement, e.g.

```
if {rpn .x. 10 >}
   show "The variable .x. is less than 10"
else if {rpn .x. 20 >}
   show "The variable .x. is between 10 and 20"
else
  show "The variable .x. is greater than 20"
end if
```

2.2 Loops

`while` loops are provided. The statements between `while` and `end while` are repeated until the RPN expression on the `while` commandline is false.

Here is an infinite loop ended – by a `break` statement – when the file contents are exhausted:

```
while 1
   read .x. .y.
   if ..eof..
      break
   end if
   show ".x. is " .x.
end while
```

Here is a loop that will print the numbers 0, 1, ..., 9.

```
.i. = 0
while {rpn .i. 10 >}
   show .i.
   .i. += 1
end while
```

3 List of Gri Commands

What follows is a complete list of built-in Gri commands. For more help on a given command, see the full manual, or use the Gri online help facitilty (e.g., type `gri` to launch Gri, then type `help`; exit by typing

The notation is as follows.

- Items written within square brackets are optional.
- Items written within dots are either raw numbers, RPN expressions, or variable names.
- Items preceeded by backslashes are any given string.
- Items separated by vertical bars are alternatives.
- Curly brackets group words that must appear together.

Thus, for example, the syntax

```
set dash [.n.|{.dash. .blank.}|off]
```

means that `set dash` is a possible Gri command (meaning use the default dash style). Several forms of optional items may be present also. For example, `set dash 2` is legal; it means use the dash style numbered 2. Gri will check any single number presented in this place on this command against the list of acceptable `.n.` values. If two numbers are present, Gri interprets the first as the length of dashes and the second as the length of blanks; notice the braces, indicating that these two parameters must appear together. Finally, the keyword `off` is allowed (it means go back to a solid line).

Here are the commands:

```
assert .condition. ["message"]
cd [\pathname]
close [\filename]
convert columns to grid [neighbor | {objective|boxcar .xr.
 .yr. [.n. .e.]} | {barnes [.xr. .yr. .gamma. .iter.]}]
convert columns to spline [.gamma.] [.xmin. .xmax.
 .xinc.]
convert grid to columns
convert grid to image [size .width. .height.] [box .ll_x.
 .ll_y. .ur_x. .ur_y.]
convert image to grid
create columns from function
create image grayscale banded .band.
create image greyscale banded .band.
debug [.n.] | [clipped values in draw commands] | off
delete {.variable. | \synonym [...]} | columns [{randomly
 .fraction.}|{where missing}] | grid | {[x|y] scale}
differentiate {x|y wrt index|y|x} | {grid wrt x|y}
draw arrow from .x0. .y0. to .x1. .y1. [cm]
draw arrows
draw axes if needed
draw axes [.style.|frame|none]
draw border box [.ll_x. .ll_y. .ur_x. .ur_y. .width_cm.
 .brightness.]
draw box filled .ll_x. .ll_y. .ur_x. .ur_y. [cm]
draw box .ll_x. .ll_y. .ur_x. .ur_y. [cm]
draw circle with radius .r_cm. at .x_cm. .y_cm.
draw contour [{.value. [unlabelled|{labelled "\label"}]}
 | {.min. .max. .inc. [.inc_unlabelled.] [unlabelled]}]
```

```
draw curve ove
draw curve fil
draw curve
draw essay "te
draw gri logo
 \bgcolor
draw grid
draw image
 palette [axi
 .left. right
 .ll_y_cm. .ur
draw image gra
 ment .inc.]]
draw image his
 .ur_y_cm.]
draw image
draw isopycna
 [.P_theta.]]
draw isospice
draw label bo
draw label wh
draw label fo
draw label "\s
 [cm] [rotate
draw line fro
draw line lege
draw lines {ve
 {horizontall
draw patches
draw polygon
draw regressi
draw symbol le
draw symbol [
 z]|[color [hu
 z|.s.]]]
draw time sta
 [with angle .
draw title "\s
draw values [
 .ycm.]
draw x axis [a
draw x box plo
draw y axis [a
draw y box plo
draw zero line
expecting ver
filter column
 ...
filter grid ro
 .b1. ...
filter image h
flip grid|imag
get env \resul
heal columns|{
help [*|comman
if {[!] .flag
ignore last .
input \ps_file
 [.rot_deg.]]
insert \filename
interpolate x|y grid to ...
list \command-syntax
ls [\file_specification]
mask image [to {uservalue .u.}|{imagevalue .i.}]
new page
new postscript file \name
new .variable_name.|\synonym_name [.vari-
 able_name.|\synonym_name [...]]
open {\filename}|{"system command|"} {[binary
 [uchar|int|float|double|16bit]]}|{netCDF}
postscript \string
pwd
query \synonym|.variable ["\prompt"
 [("\default"|.default)]]
quit [.exit_status.]
read colornames from RGB \filename
read columns ...
read grid {x [.rows.|{="name"}]}|{y
 [.cols.]{="name"}}|{data {[spacers] [.rows. .cols.]
 [spacers] [bycolumns]}|{="name"}}
read grid x [.rows.]
read grid y [.rows.]
read grid data [spacers] [.rows. .cols.] [spacers] [by-
 columns]
read grid x = "variable name"
read grid y = "variable name"
read grid data = "variable name"
set x grid', 'set y grid
read grid x' sets '\.return_value to 'N cols
read grid y' sets '\.return_value' to 'N rows
read grid data' sets '\.return_value to 'N rows N cols
read image colorscale [rgb|hsb]
read image grayscale
read image greyscale
read image mask rasterfile
read image mask .rows. .cols.
read image pgm [box .ll_x. .ll_y. .ur_x. .ur_y.]
read image rasterfile [box .ll_x. .ll_y. .ur_x. .ur_y.]
read image .rows. .cols. [box .ll_x. .ll_y. .ur_x. .ur_y.]
 [bycolumns]
read from \filename
read line [raw] \synonym
read [raw] [* [*...]] \synonym|{.variable. [.variable.
 ...]}
read [* [*...]] \synonym|{.variable. [.variable. ...]}
regress {y vs x [linear]}|{x vs y [linear]}
reorder columns randomly|{ascending in x|y|z}|{descending
 in x|y|z}
rpnfunction \name "action"
rescale
resize x for maps
resize x for maps
resize y for maps
resize y for maps
return
rewind [filename]
set axes style .style. | {offset [.dist_cm.]} | rectangular
 | none | default
set axes style 0
set axes style 1
set axes style 2
set axes style offset [.dist_cm.]
set axes style rectangular
set axes style none
set axes style default
set arrow size .size.|{as .num. percent of length}|default
set arrow size .size.
set arrow size as .num. percent of length
set arrow size default
set arrow type .which.
set beep on|off
set bounding box .ll_x. .ll_y. .ur_x. .ur_y. [cm|pt]
set clip [postscript] {on [.xleft. .xright. .ybottom.
 .ytop.]}|off
set clip on
set clip on .xleft. .xright. .ybottom. .ytop.
set clip off
set clip postscript on .xleft. .xright. .ybottom. .ytop.
set clip postscript off
set color \name|{rgb .red. .green. .blue.}|{hsb .hue.
 .saturation. .brightness.}
set colour \name|{rgb .red. .green. .blue.}|{hsb .hue.
 .saturation. .brightness.}
set colorname \\name {rgb .red. .green. .blue.}|{hsb .hue.
 .saturation. .brightness.}
set contour format \style|default
set contour label for lines exceeding .x. cm
set contour label position {.start_cm. .be-
 tween_cm.}|centered|default
set contour labels
 rotated|horizontal|whiteunder|nowhiteunder
set dash [.type.|{.dash_cm. .blank_cm. ...}|off]
set environment
set error action to core dump
set flag \name [off]
set font color \name|{rgb .red. .green. .blue.}|{hsb .hue.
 .saturation. .brightness.}
set font colour \name|{rgb .red. .green. .blue.}|{hsb
 .hue. .saturation. .brightness.}
set font encoding PostscriptStandard | isolatin1
set font size {.size. [cm]}|default
set font size .size.
set font size .size. cm
set font size default
set font to \fontname
set graylevel .brightness.|white|black
set greylevel .brightness.|white|black
set grid missing {above|below .intercept.
 .slope.}|{inside curve}
set grid missing above|below .intercept. .slope
set grid missing inside curve
set ignore initial newline [off]
set ignore error eof
set image colorscale ...
set image colourscale ...
set image grayscale using histogram [black .bl. white
 .wh.]
set image grey
 .wh.]
set image gray
 .inc.]]
set image grey
 .inc.]]
set image miss
 .brightness.
set image miss
 .brightness.
set image rang
set input data
set input data
set input data
set input data
set input data
set input data
set line cap .
set line join
set line width
 \name}|defaul
set missing va
set postscrip
set page size
set page portr
 .xcm. .ycm.}
set panel .row
set panels .ro
set path to "\
set symbol siz
set symbol siz
set symbol siz
set tic size .
set tic size .
set tic size d
set tics in|ou
set trace [on|
set trace
set trace on
set trace off
set u scale .c
set u scale .c
set u scale as
set v scale .c
set v scale .c
set v scale as
set x axis top
 .right. [.inc
set x axis top
set x axis bot
set x axis inc
set x axis dec
set x axis .le
set x axis .le
set x axis .le
set x format \
set x grid .le
set x grid .le
set x grid .le
set x margin {[bigger|smaller] .size.} | default
set x margin .size.
set x margin bigger .size.
set x margin smaller .size.
set x margin default
set x name "\name"|default
set x size .width_cm.|default
set x size .width_cm.
set x size default
set x type linear|log|{map E|W|N|S}
set y axis label horizontal|vertical
set y axis label horizontal
set y axis label vertical
set y axis left|right|increasing|decreasing|{.bottom.
 .top. [.incBig. [.incSml.]]}|unknown
set y axis left
set y axis right
set y axis increasing
set y axis decreasing
set y axis .bottom. .top.
set y axis .bottom. .top. .incBig.
set y axis .bottom. .top. .incBig. .incSml.
set y format \format|default|off
set y grid .bottom. .top. .inc.|{/.rows.}
set y grid .bottom. .top. .inc.
set y grid .bottom. .top. /.rows.
set y margin {[bigger|smaller] .size.} | default
set y margin .size.
set y margin bigger .size.
set y margin smaller .size.
set y margin default
set y name "\name"|default
set y size .height_cm.|default
set y size .height_cm.
set y size default
set y type linear|log|{map N|S|E|W}
set z missing above|below .intercept. .slope.
set "..."
show all
show axes
show color
show colornames
show columns [statistics]
show flags
show grid [mask]
show hint of the day
show image
show license
show misc
show next line
show traceback
show stopwatch
show synonyms
show time
show variables
show .value. | {rpn ...} | "\text" [.value.|{rpn ...}|text
 [...]]
skip [forward|backward] [.n.]
sleep .sec.
```

```
smooth {x [.n.]} | {y [.n.]} | {grid data [.f.|{along x|y}]}
source \filename
sprintf \synonym "format" .variable. [.variable. [...]]
state save|restore|display
superuser
system \system-command
while .test.|{rpn ...}
write columns to \filename
write contour .value. to \filename
write grid to \filename [bycolumns]
write image colorscale to \filename
write image grayscale to \filename
write image greyscale to \filename
write image mask [pgm|rasterfile] to \filename
write image [pgm|rasterfile] to \filename
unlink \filename
?draw axes exploded
?contour xyz data
?set axes
?draw image BW raster
```

See also *refcard*, an overview of Gri.