

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 facility (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 preceded 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 overlying
draw curve filled [to {.y. y} | {.x. x}]
draw curve
draw essay "text"|reset
draw gri logo .x.cm. .y.cm. .height.cm. .style. \fgcolor
\bgcolor
draw grid
draw image
  palette [axisleft|axisright|axistop|axisbottom] [left
.left. right .right. [increment .inc.]] [box .ll.x.cm.
.ll.y.cm. .ur.x.cm. .ur.y.cm.]
draw image grayscale [left .left. right .right. [incre-
ment .inc.]] [box .ll.x.cm. .ll.y.cm. .ur.x.cm. .ur.y.cm.]
draw image histogram [box .ll.x.cm. .ll.y.cm. .ur.x.cm.
.ur.y.cm.]
draw image
draw isopycnal [unlabelled] .density. [.P.sigma.
.P.theta.]]
draw isospice .spice. [unlabelled]
draw label boxed "\string" at .ll.x. .ll.y. [cm]
draw label whiteunder "\string" at .ll.x. .ll.y. [cm]
draw label for last curve "label"
draw label "\string" [centered|rightjustified] at .x. .y.
[cm] [rotated .deg.]
draw line from .x0. .y0. to .x1. .y1. [cm]
draw line legend "label" at .x. .y. [cm] [length .cm.]
draw lines {vertically .left. .right. .inc.} |
{horizontally .bottom. .top. .inc.}
draw patches .width. .height. [cm]
draw polygon [filled] .x0. .y0. .x1. .y1. .x2. .y2. [...]
draw regression line [clipped]
draw symbol legend \symbol_name "label" at .x. .y. [cm]
draw symbol [.code.]\name [at .x. .y. [cm]] [graylevel
z]|[color [hue z|h.] [brightness z|b.] [saturation
z|.s.]]]
draw time stamp [fontsize .points. [at .x.cm. .y.cm. cm
[with angle .deg.]]]
draw title "\string"
draw values [.dx. .dy.] [\format] [separation .x.cm.
.ycm.]
draw x axis [at bottom|top]{.y. [cm]} [lower|upper]]
draw x box plot at .y. [size .cm.]
draw y axis [at left|right]{.x. cm} [left|right]]
draw y box plot at .x. [size .cm]
draw zero line [horizontally|vertically]
expecting version .n.
filter column x|y|z|u|v recursively .a0. .a1. ... .b0. .b1.
...
filter grid rows|columns recursively .a0. .a1. ... .b0.
.b1. ...
filter image highpass|lowpass
flip grid|image x|y
get env \result \environment_variable
heal columns{|grid along x|y}
help [*|command_name]{- topic}
if {[!] .flag.}|\flag[{"string1" == "string2"}]
ignore last .n.
input \ps_filename [.x.cm. .y.cm. .xmag. .ymag.
.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 colnames 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 greyscale using histogram [black .bl. white
.wh.]
set image grayscale [black .bl. white .wh. [increment
.inc.]]
set image greyscale [black .bl. white .wh. [increment
.inc.]]
set image missing value color to white|black|{graylevel
.brightness.}|{rgb .red. .green. .blue.}
set image missing value colour to white|black|{graylevel
.brightness.}
set image range .min_value. .max_value.
set input data window x|y {min. .max.}|off
set input data window x .min. .max.
set input data window x off
set input data window y .min. .max.
set input data window y off
set input data separator TAB|default
set line cap .type.
set line join .type.
set line width [axis|symbol|all] .width_pt. |{rapidograph
.name}|default
set missing value .value.
set postscript filename \"string\"
set page size letter|legal|folio|tabloid|A0|A1|A2|A3|A4|A5
set page portrait|landscape|{factor .mag.}|{translate
.xcm. .ycm.}
set panel .row. .col.
set panels .rows. .cols. [.dx_cm. .dy_cm.]
set path to \"path\"|default for data|commands
set symbol size .diameter_cm. |default
set symbol size .diameter_cm.
set symbol size default
set tic size .size. |default
set tic size .size.
set tic size default
set tics in|out
set trace [on|off]
set trace
set trace on
set trace off
set u scale .cm_per_unit. |{as x}
set u scale .cm_per_unit.
set u scale as x
set v scale .cm_per_unit. |{as y}
set v scale .cm_per_unit.
set v scale as y
set x axis top|bottom|increasing|decreasing|{left.
.right. [.incBig. [.incSml.]]}|unknown
set x axis top
set x axis bottom
set x axis increasing
set x axis decreasing
set x axis .left. .right.
set x axis .left. .right. .incBig.
set x axis .left. .right. .incBig. .incSml.
set x format \format|default|off
set x grid .left. .right. .inc.|{/.cols.}
set x grid .left. .right. .inc.
set x grid .left. .right. /.cols.

```

```

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

```

(c) 2001, Dan E. Kelley

See also *refcard*, an overview of Gri.