

**NAME**

**ode** – numerical solution of ordinary differential equations

**SYNOPSIS**

**ode** [ *options* ] [ *file* ]

**DESCRIPTION**

**ode** is a tool that solves, by numerical integration, the initial value problem for a specified system of first-order ordinary differential equations. Three distinct numerical integration schemes are available: Runge–Kutta–Fehlberg (the default), Adams–Moulton, and Euler. The Adams–Moulton and Runge–Kutta schemes are available with adaptive step size.

The operation of **ode** is specified by a program, written in its input language. The program is simply a list of expressions for the derivatives of the variables to be integrated, together with some control statements. Some examples are given in the **EXAMPLES** section.

**ode** reads the program from the specified file, or from standard input if no file name is given. If reading from standard input, **ode** will stop reading and exit when it sees a single period on a line by itself.

At each time step, the values of variables specified in the program are written to standard output. So a table of values will be produced, with each column showing the evolution of a variable. If there are only two columns, the output can be piped to **graph**(1) or a similar plotting program.

**OPTIONS****Input Options**

**-f** *file*

**--input-file** *file*

Read input from *file* before reading from standard input. This option makes it possible to work interactively, after reading a program fragment that defines the system of differential equations.

**Output Options**

**-p** *prec*

**--precision** *prec*

When printing numerical results, use *prec* significant digits (the default is 6). If this option is given, the print format will be scientific notation.

**-t**

**--title** Print a title line at the head of the output, naming the variables in each column. If this option is given, the print format will be scientific notation.

**Integration Scheme Options**

The following options specify the numerical integration scheme. Only one of the three basic options **-R**, **-A**, **-E** may be specified. The default is **-R** (Runge–Kutta–Fehlberg).

**-R** [*stepsize*]

**--runge-kutta** [*stepsize*]

Use a fifth-order Runge–Kutta–Fehlberg algorithm, with an adaptive stepsize unless a constant stepsize is specified. When a constant stepsize is specified and no error analysis is requested, then a classical fourth-order Runge–Kutta scheme is used.

**-A** [*stepsize*]

**--adams-moulton** [*stepsize*]

Use a fourth-order Adams–Moulton predictor-corrector scheme, with an adaptive stepsize unless a constant stepsize, *stepsize*, is specified. The Runge–Kutta–Fehlberg algorithm is used to get past ‘bad’ points (if any).

**-E** [*stepsize*]

**--euler** [*stepsize*]

Use a ‘quick and dirty’ Euler scheme, with a constant stepsize. The default value of *stepsize* is 0.1. Not recommended for serious applications.

The error bound options **-r** and **-e** (see below) may not be used if **-E** is specified.

**-h** *hmin* [*hmax*]

**--step-size-bound** *hmin* [*hmax*]

Use a lower bound *hmin* on the stepsize. The numerical scheme will not let the stepsize go below *hmin*. The default is to allow the stepsize to shrink to the machine limit, i.e., the minimum nonzero double-precision floating point number.

The optional argument *hmax*, if included, specifies a maximum value for the stepsize. It is useful in preventing the numerical routine from skipping quickly over an interesting region.

### Error Bound Options

**-r** *rmax* [*rmin*]

**--relative-error-bound** *rmax* [*rmin*]

The **-r** option sets an upper bound on the relative single-step error. If the **-r** option is used, the relative single-step error in any dependent variable will never exceed *rmax* (the default for which is  $10^{-9}$ ). If this should occur, the solution will be abandoned and an error message will be printed. If the stepsize is not constant, the stepsize will be decreased 'adaptively', so that the upper bound on the single-step error is not violated. Thus, choosing a smaller upper bound on the single-step error will cause smaller stepsizes to be chosen. A lower bound *rmin* may optionally be specified, to suggest when the stepsize should be increased (the default for *rmin* is *rmax*/1000).

**-e** *emax* [*emin*]

**--absolute-error-bound** *emax* [*emin*]

Similar to **-r**, but bounds the absolute rather than the relative single-step error.

**-s**

**--suppress-error-bound**

Suppress the ceiling on single-step error, allowing **ode** to continue even if this ceiling is exceeded. This may result in large numerical errors.

### Informational Options

**--help** Print a list of command-line options, and exit.

**--version**

Print the version number of **ode** and the plotting utilities package, and exit.

### DIAGNOSTICS

Mostly self-explanatory. The biggest exception is 'syntax error', meaning there is a grammatical error. Language error messages are of the form

**ode: nnn: message...**

where 'nnn' is the number of the input line containing the error. If the **-f** option is used, the phrase "(file)" follows the 'nnn' for errors encountered inside the file. Subsequently, when **ode** begins reading the standard input, line numbers start over from 1.

No effort is made to recover successfully from syntactic errors in the input. However, there is a meager effort to resynchronize so more than one error can be found in one scan.

Run-time errors elicit a message describing the problem, and the solution is abandoned.

### EXAMPLES

The program

```
y' = y
y = 1
print t, y
step 0, 1
```

solves an initial value problem whose solution is  $y=e^t$ . When **ode** runs this program, it will write two columns of numbers to standard output. Each line will show the value of the independent variable *t*, and the variable *y*, as *t* is stepped from 0 to 1.

A more sophisticated example would be

```
sine' = cosine
cosine' = -sine
sine = 0
```

```
cosine = 1  
print t, sine  
step 0, 2*PI
```

This program solves an initial value problem for a system of two differential equations. The initial value problem turns out to define the sine and cosine functions. The program steps the system over a full period.

#### **AUTHORS**

**ode** was written by Nicholas B. Tufillaro ([nbt@reed.edu](mailto:nbt@reed.edu)), and slightly enhanced by Robert S. Maier ([rsm@math.arizona.edu](mailto:rsm@math.arizona.edu)) to merge it into the GNU plotting utilities.

#### **SEE ALSO**

"The GNU Plotting Utilities Manual".

#### **BUGS**

Email bug reports to [bug-gnu-utils@gnu.org](mailto:bug-gnu-utils@gnu.org).

**NAME**

**plot** – translate GNU metafiles to other graphics formats

**SYNOPSIS**

**plot** [ *options* ] [ *files* ]

**DESCRIPTION**

**plot** translates files in GNU metafile format to other graphics formats, or displays them on an X Window System display. GNU metafile format is a device-independent format for the storage of graphic data. It is the default output format of the programs **graph**(1), **pic2plot**(1), **tek2plot**(1), and **plotfont**(1), and is further documented in **plot**(5), since it is an enhanced version of the traditional **plot**(5) format found on non-GNU systems. It can also be produced by the GNU libplot 2-D graphics export library (see **plot**(3)).

The output format or display type is specified with the **-T** option. The possible output formats and display types are the same as those supported by **graph**(1), **plotfont**(1), **pic2plot**(1), and **tek2plot**(1). If an output file is produced, it is written to standard output.

Options and file names may be interspersed on the command line, but the options are processed before the file names are read. If **---** is seen, it is interpreted as the end of the options. If no file names are specified, or the file name **-** is encountered, the standard input is read.

**OPTIONS****General Options**

**-T** *type*

**---display-type** *type*

Select *type* as the output format or display type. It may be "X", "png", "pnm", "gif", "svg", "ai", "ps", "cgm", "fig", "pcl", "hpgl", "regis", "tek", or "meta" (the default). These refer respectively to the X Window System, PNG (Portable Network Graphics) format, portable anymap format (PBM/PGM/PPM), a pseudo-GIF format that does not use LZW encoding, the new XML-based Scalable Vector Graphics format, the format used by Adobe Illustrator, Postscript or Encapsulated Postscript (EPS) that can be edited with **idraw**(1), CGM format (by default, conforming to the WebCGM profile), the format used by the **xfig**(1) drawing editor, the Hewlett-Packard PCL 5 printer language, the Hewlett-Packard Graphics Language, ReGIS graphics format (which can be displayed by the **dxterm**(1) terminal emulator or by a VT330 or VT340 terminal), Tektronix format (which can be displayed by the **xterm**(1) terminal emulator), and device-independent GNU metafile format itself. Unless *type* is "X", an output file is produced and written to standard output.

Omitting the **-T** option is equivalent to specifying **-T meta**. Translating from metafile format to itself is occasionally useful, since there are two versions of metafile format (see the **-O** option below).

A listing of the fonts available in any specified output format may be obtained with the **---help-fonts** option (see below). If a requested font is unavailable, a default font will be substituted. The default font is "Helvetica" for "X", "svg", "ai", "ps", "cgm", and "fig", "Univers" for "pcl", and "HersheySerif" for "png", "pnm", "gif", "hpgl", "regis", "tek", and "meta".

**-p** *n*

**---page-number** *n*

Output only page number *n*, within the metafile or sequence of metafiles that is being translated.

Metafiles may consist of one or more pages, numbered beginning with 1. Also, each page may contain multiple 'frames'. **plot -T X**, **plot -T regis**, and **plot -T tek**, which plot in real time, will separate successive frames by screen erasures. **plot -T png**, **plot -T pnm**, **plot -T gif**, **plot -T svg**, **plot -T ai**, **plot -T ps**, **plot -T cgm**, **plot -T fig**, **plot -T pcl**, and **plot -T hpgl**, which do not plot in real time, will output only the last frame of any multi-frame page.

The default behavior, if **-p** is not used, is to output all pages. For example, **plot -T X** displays each page in its own X window. If the **-T png**, **-T pnm**, **-T gif**, **-T ai**, or **-T fig** option is used, the default behavior is to output only the first nonempty page, since files in those output formats contain only a single page of graphics.

Metafiles produced by **graph**(1) and **plotfont**(1) contain only a single page (page #1), which consists of two frames: an empty frame to clear the display, and a second frame that contains the graphics.

—s

—merge—pages

Merge all displayed pages into a single page, and also merge all ‘frames’.

This option is useful when merging together single-page plots from different sources. For example, it can be used to merge together plots obtained from separate invocations of **graph**(1).

—bitmap—size *bitmap\_size*

Set the size of the graphics display in which the plot will be drawn, in terms of pixels, to be *bitmap\_size*. The default is "570x570". This is relevant only to **plot -T X**, **plot -T png**, **plot -T pnm**, and **plot -T gif**, all of which produce bitmaps. If you choose a rectangular (non-square) window size, the fonts in the plot will be scaled anisotropically, i.e., by different factors in the horizontal and vertical directions. For **plot -T X**, this requires an X11R6 display. Any font that cannot be scaled in this way will be replaced by a default scalable font, such as the vector font "HersheySerif".

The environment variable **BITMAPSIZE** can equally well be used to specify the window size. For backward compatibility, the X resource **Xplot.geometry** may be used instead.

—emulate—color *option*

If *option* is *yes*, replace each color in the output by an appropriate shade of gray. This is seldom useful, except when using **plot -T pcl** to prepare output for a PCL 5 device. (Many monochrome PCL 5 devices, such as monochrome LaserJets, do a poor job of emulating color on their own.) You may equally well request color emulation by setting the environment variable **EMULATE\_COLOR** to "yes".

—max—line—length *max\_line\_length*

Set the maximum number of points that a polygonal line may contain, before it is flushed out, to be *max\_line\_length*. If this flushing occurs, the polygonal line will be split into two or more sub-lines, though the splitting should not be noticeable. The default value of *max\_line\_length* is 500.

The reason for splitting long polygonal lines is that some display devices (e.g., old Postscript printers and pen HP-GL plotters) have limited buffer sizes. The environment variable **MAX\_LINE\_LENGTH** can also be used to specify the maximum line length.

—page—size *pagesize*

Set the size of the page on which the plot will be positioned. This is relevant only to **plot -T svg**, **plot -T ai**, **plot -T ps**, **plot -T cgm**, **plot -T fig**, **plot -T pcl**, and **plot -T hpgl**. The default is "letter", which means an 8.5 inch by 11 inch page. Any ISO page size in the range "a0"... "a4" or ANSI page size in the range "a"... "e" may be specified ("letter" is an alias for "a" and "tabloid" is an alias for "b"). "legal" and "ledger" are recognized page sizes also. The environment variable **PAGESIZE** can equally well be used to specify the page size.

The graphics display in which the plot is drawn will, by default, be a square region that occupies nearly the full width of the specified page. An alternative size for the graphics display can be specified. For example, the page size could be specified as "letter,xsize=4in,ysize=6in", or "a4,xsize=5.0cm,ysize=100mm". For all of the above except **plot -T hpgl**, the graphics display will, by default, be centered on the page. For all of the above except **plot -T svg** and **plot -T cgm**, the graphics display may be repositioned manually, by specifying the location of its lower left corner, relative to the lower left corner of the page. For example, the page size could be specified as "letter,xorigin=2in,yorigin=3in", or "a4,xorigin=0.5cm,yorigin=0.5cm". It is also possible to specify an offset vector. For example, the page size could be specified as "letter,xoffset=1in", or "letter,xoffset=1in,yoffset=1.2in", or "a4,yoffset=-1cm". In SVG format and WebCGM format it is possible to specify the size of the graphics display, but not its position.

—rotation *angle*

Rotate the graphics display by *angle* degrees. Recognized values are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. The environment variable

**ROTATION** can also be used to specify a rotation angle.

### Parameter Initialization Options

The following options set the initial values of drawing parameters. However, all of these may be overridden by directives in a metafile. In fact, these options are useful primarily when plotting old metafiles in the traditional (pre-GNU) **plot(5)** format, which did not support such directives.

#### **--bg-color** *name*

Set the color initially used for the background to be *name*. This is relevant only to **plot -T X**, **plot -T png**, **plot -T pnm**, **plot -T gif**, **plot -T svg**, **plot -T cgm**, and **plot -T regis**. An unrecognized name sets the color to the default, which is "white". The environment variable **BG\_COLOR** can equally well be used to specify the background color.

If the **-T png** or **-T gif** option is used, a transparent PNG file or a transparent pseudo-GIF, respectively, may be produced by setting the **TRANSPARENT\_COLOR** environment variable to the name of the background color. If the **-T svg** or **-T cgm** option is used, an output file without a background may be produced by setting the background color to "none".

#### **-f** *size*

#### **--font-size** *size*

Set the size of the font initially used for rendering text, as a fraction of the width of the graphics display, to be *size*. The default is 0.0525.

#### **-F** *name*

#### **--font-name** *name*

Set the font initially used for text to be *name*. Font names are case-insensitive. If the specified font is not available, the default font will be used. Which fonts are available, and the default font, depend on which **-T** option is specified (see above). A list of available fonts can be obtained with the **--help-fonts** option (see below).

#### **-W** *line\_width*

#### **--line-width** *line\_width*

Set the initial width of lines, as a fraction of the width of the display, to be *line\_width*. A negative value means that a default value should be used. This value is format-dependent. The interpretation of zero line width is also format-dependent (in some output formats, a zero-width line is the thinnest line that can be drawn; in others, a zero-width line is invisible).

#### **--pen-color** *name*

Set the initial pen color to be *name*. An unrecognized name sets the pen color to the default, which is "black".

### Options for Metafile Output

The following option is relevant only if the **-T** option is omitted or if **-T meta** is used. In this case the output of **plot**, like the input, will be in GNU graphics metafile format.

#### **-O**

#### **--portable-output**

Output the portable (human-readable) version of GNU metafile format, rather than the binary version (the default). The format of the binary version is machine-dependent.

### Options for Backward Compatibility

By default, **plot** assumes that its input file(s) are in either the binary version or the portable version of GNU metafile format. You may specify that the input is, instead, in the traditional Unix (pre-GNU) graphics metafile format, which is documented in **plot(5)**. The traditional graphics metafile format was produced by pre-GNU versions of **graph(1)**.

#### **-h**

#### **--high-byte-first-input**

Input file(s) are assumed to be in the binary, 'high byte first' version of traditional metafile format. This variant is uncommon.

#### **-l**

#### **--low-byte-first-input**

Input file(s) are assumed to be in the binary, 'low byte first' version of traditional metafile format. This variant is the most common.

**-A**

**--ascii-input**

Input file(s) are assumed to be in the ASCII (human-readable) variant of traditional metafile format. On some older Unix systems, this variant was produced by **plottoa(1)**.

### Informational Options

**--help** Print a list of command-line options, and exit.

**--help-fonts**

Print a table of available fonts, and exit. The table will depend on which output format or display type is specified with the **-T** option. **plot -T X**, **plot -T svg**, **plot -T ai**, **plot -T ps**, **plot -T cgm**, and **plot -T fig** each support the 35 standard Postscript fonts. **plot -T svg**, **plot -T pcl**, and **plot -T hpgl** support the 45 standard PCL 5 fonts, and the latter two support a number of Hewlett-Packard vector fonts. All seven support a set of 22 Hershey vector fonts, as do **plot -T png**, **plot -T pnm**, **plot -T gif**, **plot -T regis**, and **plot -T tek**. **plot** without a **-T** option in principle supports any of these fonts, since its output must be translated to other formats by a further invocation of **plot**.

The **plotfont(1)** utility may be used to obtain a character map of any supported font.

**--list-fonts**

Like **--help-fonts**, but lists the fonts in a single column to facilitate piping to other programs. If no output format is specified with the **-T** option, the full set of supported fonts is listed.

**--version**

Print the version number of **plot** and the plotting utilities package, and exit.

### ENVIRONMENT

The environment variables **BITMAPSIZE**, **PAGESIZE**, **BG\_COLOR**, **EMULATE\_COLOR**, **MAX\_LINE\_LENGTH** and **ROTATION** serve as backups for the options **--bitmap-size**, **--page-size**, **--bg-color**, **--emulate-color**, **--max-line-length**, and **--rotation**, respectively. The remaining environment variables are specific to individual output formats.

**plot -T X**, which pops up a window on an X Window System display and draws graphics in it, checks the **DISPLAY** environment variable. Its value determines the display that will be used.

**plot -T png** and **plot -T gif**, which produce output in PNG format and pseudo-GIF format respectively, are affected by the **INTERLACE** environment variable. If its value is "yes", the output will be interlaced. Also, if the **TRANSPARENT\_COLOR** environment variable is set to the name of a color, that color will be treated as transparent in the output.

**plot -T pnm**, which produces output in portable anymap (PBM/PGM/PPM) format, is affected by the **PNM\_PORTABLE** environment variable. If its value is "yes", the output will be in a human-readable format rather than binary (the default).

**plot -T cgm**, which produces output in CGM (Computer Graphics Metafile) format, is affected by the **CGM\_MAX\_VERSION** and **CGM\_ENCODING** environment variables. By default, it produces a binary-encoded version of CGM version 3 format. For backward compatibility, the version number may be reduced by setting **CGM\_MAX\_VERSION** to "2" or "1". Irrespective of version, the output CGM file will use the human-readable clear text encoding if **CGM\_ENCODING** is set to "clear\_text". However, only binary-encoded CGM files conform to the WebCGM profile.

**plot -T pcl**, which produces PCL 5 output for Hewlett-Packard printers and plotters, is affected by the environment variable **PCL\_ASSIGN\_COLORS**. It should be set to "yes" when producing PCL 5 output for a color printer or other color device. This will ensure accurate color reproduction by giving the output device complete freedom in assigning colors, internally, to its "logical pens". If it is "no" then the device will use a fixed set of colored pens, and will emulate other colors by shading. The default is "no" because monochrome PCL 5 devices, which are much more common than colored ones, must use shading to emulate color.

**plot -T hpgl**, which produces Hewlett-Packard Graphics Language output, is affected by several environment variables. The most important is **HPGL\_VERSION**, which may be set to "1", "1.5", or "2" (the default). "1" means that the output should be generic HP-GL, "1.5" means that the output should be suitable for the HP7550A graphics plotter and the HP758x, HP7595A and HP7596A drafting plotters (HP-GL with some HP-GL/2 extensions), and "2" means that the output should be modern HP-GL/2. If the version is "1" or "1.5" then the only available fonts will be vector fonts, and all lines will be

drawn with a default width (the **-W** option will not work). Additionally, if the version is "1" then the filling of arbitrary curves with solid color will not be supported (circles and rectangles aligned with the coordinate axes may be filled, though).

The position of the **plot -T hpgl** graphics display on the page can be rotated 90 degrees counterclockwise by setting the **HPGL\_ROTATE** environment variable to "yes". This is not the same as the rotation obtained with the **--rotation** option, since it both rotates the graphics display and repositions its lower left corner toward another corner of the page. Besides "no" and "yes", recognized values for **HPGL\_ROTATE** are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. "180" and "270" are supported only if **HPGL\_VERSION** is "2" (the default).

By default, **plot -T hpgl** will draw with a fixed set of pens. Which pens are present may be specified by setting the **HPGL\_PENS** environment variable. If **HPGL\_VERSION** is "1", the default value of **HPGL\_PENS** is "1=black"; if **HPGL\_VERSION** is "1.5" or "2", the default value of **HPGL\_PENS** is "1=black:2=red:3=green:4=yellow:5=blue:6=magenta:7=cyan". The format should be self-explanatory. By setting **HPGL\_PENS** you may specify a color for any pen in the range #1..#31. All color names recognized by the X Window System may be used. Pen #1 must always be present, though it need not be black. Any other pen in the range #1..#31 may be omitted.

If **HPGL\_VERSION** is "2" then **plot -T hpgl** will also be affected by the environment variable **HPGL\_ASSIGN\_COLORS**. If its value is "yes", then **plot -T hpgl** will not be restricted to the palette specified in **HPGL\_PENS**: it will assign colors to "logical pens" in the range #1..#31, as needed. The default value is "no" because other than color LaserJet printers and DesignJet plotters, not many HP-GL/2 devices allow the assignment of colors to logical pens.

Opaque filling and the drawing of visible white lines are supported only if **HPGL\_VERSION** is "2" and the environment variable **HPGL\_OPAQUE\_MODE** is "yes" (the default). If its value is "no" then white lines (if any), which are normally drawn with pen #0, will not be drawn. This feature is to accommodate older HP-GL/2 devices. HP-GL/2 pen plotters, for example, do not support opacity or the use of pen #0 to draw visible white lines. Some older HP-GL/2 devices may, in fact, malfunction if asked to draw opaque objects.

**plot -T tek**, which produces output for a Tektronix terminal or emulator, checks the **TERM** environment variable. If the value of **TERM** is a string beginning with "xterm", "nxterm", or "kterm", it is taken as a sign that **plot** is running in an X Window System VT100 terminal emulator: a copy of **xterm(1)**, **nxterm(1)**, or **kterm(1)**. Before drawing graphics, **plot -T tek** will emit an escape sequence that causes the terminal emulator's auxiliary Tektronix window, which is normally hidden, to pop up. After the graphics are drawn, an escape sequence that returns control to the original VT100 window will be emitted. The Tektronix window will remain on the screen.

If the value of **TERM** is a string beginning with "kermit", "ansi.sys", or "nansi.sys", it is taken as a sign that **plot** is running in the VT100 terminal emulator provided by the MS-DOS version of **kermit(1)**. Before drawing graphics, **plot -T tek** will emit an escape sequence that switches the terminal emulator to Tektronix mode. Also, some of the Tektronix control codes emitted by **plot -T tek** will be **kermit**-specific. There will be a limited amount of color support, which is not normally the case (the 16 'ansi.sys' colors will be supported). After drawing graphics, **plot -T tek** will emit an escape sequence that returns the emulator to VT100 mode. The key sequence 'ALT minus' can be employed manually within **kermit** to switch between the two modes.

## SEE ALSO

**graph(1)**, **pic2plot(1)**, **tek2plot(1)**, **plotfont(1)**, **plot(3)**, **plot(5)**, and "The GNU Plotting Utilities Manual".

## AUTHORS

**plot** was written by Robert S. Maier ([rsm@math.arizona.edu](mailto:rsm@math.arizona.edu)).

## BUGS

Email bug reports to [bug-gnu-utils@gnu.org](mailto:bug-gnu-utils@gnu.org).

**NAME**

plotfont – produce character maps of fonts supported by the plotting utilities

**SYNOPSIS**

**plotfont** [ *options* ] *fonts*

**DESCRIPTION**

**plotfont** produces a character map for any font that is supported by the plotting utilities, which include **graph**(1), **plot**(1), **pic2plot**(1), **tek2plot**(1), and the GNU libplot 2-D graphics export library (see **plot**(3)). Which fonts are supported depends on the output format or display type, which is specified by the **-T** option. A listing of the fonts available in any specified output format may be obtained with the **--help-fonts** option (see below).

The character map, or maps, will be written to standard output in the specified format. For example, the Times-Roman font is available when producing Postscript output. The command **plotfont -T ps Times-Roman > charmap.ps** will yield a character map of the Times-Roman font, in a Postscript format that can be viewed or edited with the **idraw**(1) drawing editor. The Times-Roman font is also available when producing Fig output, which can be viewed or edited with the **xfig**(1) drawing editor. The command **plotfont -T fig Times-Roman > charmap.fig** will yield the same character map, but in Fig format rather than in Postscript format.

As another example, the Univers font is available when producing PCL 5 output. The command **plotfont -T pcl Univers > charmap.pcl** will produce a character map of the Univers font, in PCL 5 format.

When producing output for the X Window System, i.e., for a popped-up window, any scalable X Window System font that has an XLFD (i.e., X Logical Font Description) name is supported. For example, the command **plotfont -T X utopia-medium-r-normal** will pop up a window, and draw a character map of the Utopia-Regular font. "utopia-medium-r-normal" is a truncated version of the Utopia-Regular font's XLFD name. The Utopia-Regular font is available on most X Window System displays.

**OPTIONS****General Options**

**-T** *type*

**--display-type** *type*

Select *type* as the output format or display type. It may be "X", "png", "pnm", "gif", "svg", "ai", "ps", "cgm", "fig", "pcl", "hpgl", "regis", "tek", or "meta" (the default). These refer respectively to the X Window System, PNG (Portable Network Graphics) format, portable anymap format (PBM/PGM/PPM), a pseudo-GIF format that does not use LZW encoding, the new XML-based Scalable Vector Graphics format, the format used by Adobe Illustrator, Postscript or Encapsulated Postscript (EPS) that can be edited with **idraw**(1), CGM format (by default, conforming to the WebCGM profile), the format used by the **xfig**(1) drawing editor, the Hewlett-Packard PCL 5 printer language, the Hewlett-Packard Graphics Language, ReGIS graphics format (which can be displayed by the **dxterm**(1) terminal emulator or by a VT330 or VT340 terminal), Tektronix format (which can be displayed by the **xterm**(1) terminal emulator), and device-independent GNU metafile format itself. Unless *type* is "X", an output file is produced and written to standard output.

Files in PNG, PNM, pseudo-GIF, AI, or Fig format contain only a single page of graphics. So if the **-T png** option, the **-T pnm** option, the **-T gif** option, the **-T ai** option, or the **-T fig** option is used, the output file will contain a character map for only the first-specified font.

A listing of the fonts available in any specified output format may be obtained with the **--help-fonts** option (see below). If a requested font is unavailable, a default font will be substituted. The default font is "Helvetica" for "X", "svg", "ai", "ps", "cgm", and "fig", "Univers" for "pcl", and "HersheySerif" for "png", "pnm", "gif", "hpgl", "regis", "tek", and "meta".

**-1**

**--lower-half**

Generate a character map for the lower half of each specified font. This is the default.

**-2**

- upper-half**  
Generate a character map for the upper half of each specified font.
- o**
- octal**  
Number the characters in octal rather than in decimal (the default).
- x**
- hexadecimal**  
Number the characters in hexadecimal rather than in decimal (the default).
- box** Surround each character with a box, showing its extent to left and right. The default is not to do this.
- j row**
- jis-row row**  
Generate a character map for row *row* of a Japanese font arranged according to JIS [Japanese Industrial Standard] X0208. The only such font currently available is the HersheyEUC [Extended Unix Code] font. If used, this option overrides the **-1** and **-2** options. The valid rows are 1..94. In the JIS X0208 standard, Roman characters are located in row 3, and Japanese syllabic characters (Hiragana and Katakana) are located in rows 4 and 5. Greek and Cyrillic characters are located in rows 6 and 7. Japanese ideographic characters (Kanji) are located in rows 16..84.
- bg-color name**  
Set the color used for the background to be *name*. This is relevant only to **plotfont -T X**, **plotfont -T png**, **plotfont -T pnm**, **plotfont -T gif**, **plotfont -T svg**, **plotfont -T cgm**, and **plotfont -T regis**. An unrecognized name sets the color to the default, which is "white". The environment variable **BG\_COLOR** can equally well be used to specify the background color.  
  
If the **-T png** or **-T gif** option is used, a transparent PNG file or a transparent pseudo-GIF, respectively, may be produced by setting the **TRANSPARENT\_COLOR** environment variable to the name of the background color. If the **-T svg** or **-T cgm** option is used, an output file without a background may be produced by setting the background color to "none".
- bitmap-size bitmap\_size**  
Set the size of the graphics display in which the character map(s) will be drawn, in terms of pixels, to be *bitmap\_size*. The default is "570x570". This is relevant only to **plotfont -T X**, **plotfont -T png**, **plotfont -T pnm**, and **plotfont -T gif**, all of which produce bitmaps. If you choose a rectangular (non-square) window size, the fonts in the character map(s) will be scaled anisotropically, i.e., by different factors in the horizontal and vertical directions. For **plotfont -T X**, this requires an X11R6 display. Any font that cannot be scaled in this way will be replaced by a default scalable font, such as the vector font "HersheySerif".  
  
The environment variable **BITMAPSIZE** can equally well be used to specify the window size. For backward compatibility, the X resource **Xplot.geometry** may be used instead.
- emulate-color option**  
If *option* is *yes*, replace each color in the output by an appropriate shade of gray. This is seldom useful, except when using **plotfont -T pcl** to prepare output for a PCL 5 device. (Many monochrome PCL 5 devices, such as monochrome LaserJets, do a poor job of emulating color on their own.) You may equally well request color emulation by setting the environment variable **EMULATE\_COLOR** to "yes".
- numbering-font name**  
Set the font used for the numbering of the characters in the character map(s) to be *name*, rather than the default.
- page-size pagesize**  
Set the size of size of the page on which the character map(s) will be positioned. This is relevant only to **plotfont -T svg**, **plotfont -T ai**, **plotfont -T ps**, **plotfont -T cgm**, **plotfont -T fig**, **plotfont -T pcl**, and **plotfont -T hpgl**. The default is "letter", which means an 8.5 inch by 11 inch page. Any ISO page size in the range "a0".."a4" or ANSI page size in the range "a".."e" may be specified ("letter" is an alias for "a" and "tabloid" is an alias for "b"). "legal" and "ledger" are recognized page sizes also. The environment variable **PAGESIZE** can equally

well be used to specify the page size.

The graphics display in which each character map is drawn will be a square region that would occupy nearly the full width of the specified page. An alternative size for the graphics display can be specified. For example, the page size could be specified as "letter,xsize=4in,ysize=6in", or "a4,xsize=5.0cm,ysize=100mm". For all of the above except **plotfont -T hpgl**, the graphics display will, by default, be centered on the page. For all of the above except **plotfont -T svg** and **plotfont -T cgm**, the graphics display may be repositioned manually, by specifying the location of its lower left corner, relative to the lower left corner of the page. For example, the page size could be specified as "letter,xorigin=2in,yorigin=3in", or "a4,xorigin=0.5cm,yorigin=0.5cm". It is also possible to specify an offset vector. For example, the page size could be specified as "letter,xoffset=1in", or "letter,xoffset=1in,yoffset=1.2in", or "a4,yoffset=-1cm". In SVG format and WebCGM format it is possible to specify the size of the graphics display, but not its position.

**—rotation *angle***

Rotate the graphics display by *angle* degrees. Recognized values are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. The environment variable **ROTATION** can also be used to specify a rotation angle.

**—pen—color *name***

Set the pen color to be *name*. An unrecognized name sets the pen color to the default, which is "black".

### Options for Metafile Output

The following option is relevant only if the **-T** option is omitted or if **-T meta** is used. In this case the output of **plotfont** will be in GNU graphics metafile format. It may be translated to other formats by invoking **plot(1)**.

**-O**

**—portable—output**

Output the portable (human-readable) version of GNU metafile format, rather than the binary version (the default). The format of the binary version is machine-dependent.

### Informational Options

**—help** Print a list of command-line options, and exit.

**—help—fonts**

Print a table of available fonts, and exit. The table will depend on which output format or display type is specified with the **-T** option. **plotfont -T X**, **plotfont -T svg**, **plotfont -T ai**, **plotfont -T ps**, **plotfont -T cgm**, and **plotfont -T fig** each support the 35 standard Postscript fonts. **plotfont -T svg**, **plotfont -T pcl**, and **plotfont -T hpgl** support the 45 standard PCL 5 fonts, and the latter two support a number of Hewlett-Packard vector fonts. All seven support a set of 22 Hershey vector fonts, as do **plotfont -T png**, **plotfont -T pnm**, **plotfont -T gif**, **plotfont -T regis**, and **plotfont -T tek**. **plotfont** without a **-T** option in principle supports any of these fonts, since its output must be translated to other formats by invoking **plot(1)**.

**—list—fonts**

Like **—help—fonts**, but lists the fonts in a single column to facilitate piping to other programs. If no output format is specified with the **-T** option, the full set of supported fonts is listed.

**—version**

Print the version number of **plotfont** and the plotting utilities package, and exit.

### ENVIRONMENT

The environment variables **BITMAPSIZE**, **PAGESIZE**, **BG\_COLOR**, **EMULATE\_COLOR**, and **ROTATION** serve as backups for the options **—bitmap—size**, **—page—size**, **—bg—color**, **—emulate—color**, and **—rotation**, respectively. The remaining environment variables are specific to individual output formats.

**plotfont -T X**, which pops up a window on an X Window System display for each character map, checks the **DISPLAY** environment variable. Its value determines the display that will be used.

**plotfont -T png** and **plotfont -T gif**, which produce output in PNG format and pseudo-GIF format respectively, are affected by the **INTERLACE** environment variable. If its value is "yes", the output will be interlaced. Also, if the **TRANSPARENT\_COLOR** environment variable is set to the name of a

color, that color will be treated as transparent in the output.

**plotfont -T pnm**, which produces output in portable anymap (PBM/PGM/PPM) format, is affected by the **PNM\_PORTABLE** environment variable. If its value is "yes", the output will be in a human-readable format rather than binary (the default).

**plotfont -T cgm**, which produces output in CGM (Computer Graphics Metafile) format, is affected by the **CGM\_MAX\_VERSION** and **CGM\_ENCODING** environment variables. By default, it produces a binary-encoded version of CGM version 3 format. For backward compatibility, the version number may be reduced by setting **CGM\_MAX\_VERSION** to "2" or "1". Irrespective of version, the output CGM file will use the human-readable clear text encoding if **CGM\_ENCODING** is set to "clear\_text". However, only binary-encoded CGM files conform to the WebCGM profile.

**plotfont -T pcl**, which produces PCL 5 output for Hewlett-Packard printers and plotters, is affected by the environment variable **PCL\_ASSIGN\_COLORS**. It should be set to "yes" when producing PCL 5 output for a color printer or other color device. This will ensure accurate color reproduction by giving the output device complete freedom in assigning colors, internally, to its "logical pens". If it is "no" then the device will use a fixed set of colored pens, and will emulate other colors by shading. The default is "no" because monochrome PCL 5 devices, which are much more common than colored ones, must use shading to emulate color.

**plotfont -T hpgl**, which produces Hewlett-Packard Graphics Language output, is affected by several environment variables. The most important is **HPGL\_VERSION**, which may be set to "1", "1.5", or "2" (the default). "1" means that the output should be generic HP-GL, "1.5" means that the output should be suitable for the HP7550A graphics plotter and the HP758x, HP7595A and HP7596A drafting plotters (HP-GL with some HP-GL/2 extensions), and "2" means that the output should be modern HP-GL/2. If the version is "1" or "1.5" then the only available fonts will be vector fonts, and all lines will be drawn with a default width. Additionally, if the version is "1" then the filling of arbitrary curves with solid color will not be supported (circles and rectangles aligned with the coordinate axes may be filled, though).

The position of the **plotfont -T hpgl** graphics display on the page can be rotated 90 degrees counter-clockwise by setting the **HPGL\_ROTATE** environment variable to "yes". This is not the same as the rotation obtained with the **—rotation** option, since it both rotates the graphics display and repositions its lower left corner toward another corner of the page. Besides "no" and "yes", recognized values for **HPGL\_ROTATE** are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. "180" and "270" are supported only if **HPGL\_VERSION** is "2" (the default).

By default, **plotfont -T hpgl** will draw with a fixed set of pens. Which pens are present may be specified by setting the **HPGL\_PENS** environment variable. If **HPGL\_VERSION** is "1", the default value of **HPGL\_PENS** is "1=black"; if **HPGL\_VERSION** is "1.5" or "2", the default value of **HPGL\_PENS** is "1=black:2=red:3=green:4=yellow:5=blue:6=magenta:7=cyan". The format should be self-explanatory. By setting **HPGL\_PENS** you may specify a color for any pen in the range #1...#31. All color names recognized by the X Window System may be used. Pen #1 must always be present, though it need not be black. Any other pen in the range #1...#31 may be omitted.

If **HPGL\_VERSION** is "2" then **plotfont -T hpgl** will also be affected by the environment variable **HPGL\_ASSIGN\_COLORS**. If its value is "yes", then **plotfont -T hpgl** will not be restricted to the palette specified in **HPGL\_PENS**: it will assign colors to "logical pens" in the range #1...#31, as needed. The default value is "no" because other than color LaserJet printers and DesignJet plotters, not many HP-GL/2 devices allow the assignment of colors to logical pens.

Opaque filling and the drawing of visible white lines are supported only if **HPGL\_VERSION** is "2" and the environment variable **HPGL\_OPAQUE\_MODE** is "yes" (the default). If its value is "no" then white lines (if any), which are normally drawn with pen #0, will not be drawn. This feature is to accommodate older HP-GL/2 devices. HP-GL/2 pen plotters, for example, do not support opacity or the use of pen #0 to draw visible white lines. Some older HP-GL/2 devices may, in fact, malfunction if asked to draw opaque objects.

**plotfont -T tek**, which produces output for a Tektronix terminal or emulator, checks the **TERM** environment variable. If the value of **TERM** is a string beginning with "xterm", "nxterm", or "kterm", it is taken as a sign that **plotfont** is running in an X Window System VT100 terminal emulator: a copy of **xterm(1)**, **nxterm(1)**, or **kterm(1)**. Before drawing graphics, **plotfont -T tek** will emit an escape sequence that causes the terminal emulator's auxiliary Tektronix window, which is normally hidden, to

pop up. After the graphics are drawn, an escape sequence that returns control to the original VT100 window will be emitted. The Tektronix window will remain on the screen.

If the value of **TERM** is a string beginning with "kermit", "ansi.sys", or "nansi.sys", it is taken as a sign that **plotfont** is running in the VT100 terminal emulator provided by the MS-DOS version of **kermit**(1). Before drawing graphics, **plotfont -T tek** will emit an escape sequence that switches the terminal emulator to Tektronix mode. Also, some of the Tektronix control codes emitted by **plotfont -T tek** will be **kermit**-specific. There will be a limited amount of color support, which is not normally the case (the 16 'ansi.sys' colors will be supported). After drawing graphics, **plotfont -T tek** will emit an escape sequence that returns the emulator to VT100 mode. The key sequence 'ALT minus' can be employed manually within **kermit** to switch between the two modes.

#### SEE ALSO

**graph**(1), **pic2plot**(1), **tek2plot**(1), **plot**(1), **plot**(3), and "The GNU Plotting Utilities Manual".

#### AUTHORS

**plotfont** was written by Robert S. Maier ([rsm@math.arizona.edu](mailto:rsm@math.arizona.edu)).

#### BUGS

Email bug reports to [bug-gnu-utils@gnu.org](mailto:bug-gnu-utils@gnu.org).

**NAME**

spline – interpolate datasets using splines under tension

**SYNOPSIS**

**spline** [ *options* ] [ *files* ]

**DESCRIPTION**

**spline** reads datasets from standard input or from one or more files, and fits a smooth curve (a "spline") through each dataset. An interpolated version of each dataset, consisting of points from the smooth curve, is written to standard output.

Unless the **-a** or **-A** options are used (see below), each dataset should be a sequence of values for a vector-valued function of a single scalar variable. That is, each dataset should be a sequence of data points, given as alternating *t* and *y* values. *t* is a scalar independent variable, and *y* is a vector-valued dependent variable. The dimensionality of *y* is specified with the **-d** option (the default dimensionality is 1). Between each data point and the next, *t* should increase.

An input file may contain more than a single dataset. If an input file is in ASCII format (the default), its datasets should be separated by blank lines. The *t* and *y* values of the data points in each dataset may be arranged arbitrarily, so long as they are separated by white space. Besides datasets, an input file may contain any number of comment lines, which should begin with the comment character '#'. Comment lines are ignored. They are not treated as blank, i.e., they do not interrupt a dataset in progress.

Options and file names may be interspersed on the command line, but the options are processed before the file names are read. If **--** is seen, it is interpreted as the end of the options. If no file names are specified, or the file name **-** is encountered, the standard input is read.

The type of interpolation, and the format of the input and output files, may be selected by command-line options.

**OPTIONS****Interpolation-Related Options**

**-f**

**--filter**

Use a local interpolation algorithm (the cubic Bessel algorithm), so that **spline** can be used as a real-time filter. The slope of the interpolating curve at each point in a dataset will be chosen by fitting a quadratic function through that point and the two adjacent points in the dataset. If **-f** is specified then the **-t** option, otherwise optional, must be used as well. Also, if **-f** is specified then the **-k**, **-p**, and **-T** options may not be used.

If **-f** is *not* specified, then the default (global) interpolation algorithm will be used.

**-k** *k*

**--boundary-condition** *k*

Set the boundary condition parameter for each constructed spline to be *k*. (The default value is 1.0.) In each of its components, the spline will satisfy the two boundary conditions  $y''[0]=ky''[1]$  and  $y''[n]=ky''[n-1]$ . Here  $y[0]$  and  $y[1]$  signify the values of a specified component of the vector-valued dependent variable *y* at the first two points of a dataset, and  $y[n-1]$  and  $y[n]$  the values at the last two points. Setting *k* to zero will yield a "natural" spline, i.e., one that has zero curvature at the two ends of the dataset. The **-k** option may not be used if **-f** or **-p** is specified.

**-n** *n*

**--number-of-intervals** *n*

Subdivide the interval over which interpolation occurs into *n* subintervals. The number of data points computed, and written to the output, will be  $n+1$ . The default value for *n* is 100.

**-p**

**--periodic**

Construct a periodic spline. If this option is specified, the *y* values for the first and last points in each dataset must be equal. The **-f** and **-k** options may not be used if **-p** is specified.

**-T** *tension*

**--tension** *tension*

Each interpolating curve will be a spline under tension. This option sets the tension value (the default is 0.0).

If *tension* equals zero, the curve will be a piecewise cubic spline. Increasing the tension above zero makes the curve "tighter", and reduces the likelihood of spurious inflection points. That is because between each pair of successive points in a dataset, the curve will satisfy the fourth-order differential equation  $y'''' = \text{sgn}(\textit{tension}) * (\textit{tension}^2)y''$  in each of its components. As *tension* increases to positive infinity, it will converge to a polygonal line. The **-T** option may not be used if **-f** is specified.

**-t** *tmin tmax [tspacing]*

**--t-spacing** *tmin tmax [tspacing]*

For each dataset, set the interval over which interpolation occurs to be the interval between *tmin* and *tmax*. If *tspacing* is not specified, the interval will be divided into the number of subintervals specified by the **-n** option.

If the **-t** option is not used, the interval over which interpolation occurs will be the entire range of the independent variable in the dataset. The **-t** option must always be used if the **-f** option is used to request filter-like behavior (see above).

### Format-Related Options

**-d** *dimension*

**--y-dimension** *dimension*

Set the dimensionality of the dependent variable *y* in the input and output files to be *dimension*. The default dimension is 1.

**-I** *data-format*

**--input-format** *data-format*

Set the data format for the input file(s) to be *data-format*, which may be one of the following.

- a** ASCII format (the default). Each file is a sequence of floating point numbers, interpreted as the *t* and *y* coordinates of the successive data points in a dataset. If *y* is *d*-dimensional, there will be *d+1* numbers for each point. The *t* and *y* coordinates of a point need not appear on the same line, and points need not appear on different lines. But if a blank line occurs (i.e., two newlines in succession are seen), it is interpreted as the end of a dataset, and the beginning of the next.
- f** Single precision binary format. Each file is a sequence of floating point numbers, interpreted as the *t* and *y* coordinates of the successive data points in a dataset. If *y* is *d*-dimensional, there will be *d+1* numbers for each point. Successive datasets are separated by a single occurrence of the quantity FLT\_MAX, which is the largest possible single precision floating point number. On most machines this is approximately  $3.4 \times 10^{38}$ .
- d** Double precision binary format. Each file is a sequence of double precision floating point numbers, interpreted as the *t* and *y* coordinates of the successive data points in a dataset. If *y* is *d*-dimensional, there will be *d+1* numbers for each point. Successive datasets are separated by a single occurrence of the quantity DBL\_MAX, which is the largest possible double precision floating point number. On most machines this is approximately  $1.8 \times 10^{308}$ .
- i** Integer binary format. Each file is a sequence of integers, interpreted as the *t* and *y* coordinates of the successive data points in a dataset. If *y* is *d*-dimensional, there will be *d+1* numbers for each point. Successive datasets are separated by a single occurrence of the quantity INT\_MAX, which is the largest possible integer. On most machines this is  $2^{31}-1$ .

**-a** [*step\_size* [*lower\_limit*]]

**--auto-abscissa** [*step\_size* [*lower\_limit*]]

Automatically generate values for *t*, the independent variable (the default values of *step\_size* and *lower\_limit* are 1.0 and 0.0, respectively).

Irrespective of data format ('a', 'f', 'd', or 'i'), this option specifies that the values of  $t$  are missing from the input file: the dataset(s) to be read contain only values of  $y$ , the dependent variable. So if  $y$  is  $d$ -dimensional, there will be only  $d$  numbers for each point. The increment from each  $t$  value to the next will be *step\_size*, and the first  $t$  value will be *lower\_limit*. This option is useful, e.g., when interpolating curves rather than functions.

**-A**

**--auto-dist-abscissa**

Automatically generate values for  $t$ , the independent variable. This is a variant form of the **-a** option. The increment from each  $t$  value to the next will be the distance in  $d$ -dimensional space between the corresponding  $y$  values, and the first  $t$  value will be 0.0. That is,  $t$  will be "polygonal arclength". This option is useful when interpolating curves rather than functions.

**-O** *data-format*

**--output-format** *data-format*

Set the data format for the output file to be *data-format*. The interpretation of *data-format* is the same as for the **-I** option. The default is 'a', i.e., ASCII format.

**-P** *significant-digits*

**--precision** *significant-digits*

Set the numerical precision for the  $t$  and  $y$  values in the output file to be *significant-digits*. This takes effect only if the output file is written in 'a' format, i.e., in ASCII. *significant-digits* must be a positive integer (the default is 6).

**-s**

**--suppress-abscissa**

Omit the independent variable  $t$  from the output file; for each point, supply only the dependent variable  $y$ . If  $y$  is  $d$ -dimensional, there will be only  $d$  numbers for each point, not  $d+1$ . This option is useful when interpolating curves rather than functions.

### Informational Options

**--help** Print a list of command-line options, and exit.

**--version**

Print the version number of **spline** and the plotting utilities package, and exit.

### EXAMPLES

Typing

```
echo 0 0 1 1 2 0 | spline
```

will produce on standard output an interpolated dataset consisting of 101 data points. If graphed, this interpolated dataset will yield a parabola.

It is sometimes useful to interpolate between a sequence of arbitrarily placed points in  $d$ -dimensional space, i.e., to "spline a curve" rather than a function. The **-a** and **-s** options are used for this. For example,

```
echo 0 0 1 0 1 1 0 1 | spline -d 2 -a -s
```

will produce on standard output a 101-point dataset that interpolates between the four points (0,0), (1,0), (1,1), and (0,1). The **-d 2** option specifies that the dependent variable  $y$  is two-dimensional. The **-a** option specifies that the  $t$  values are missing from the input and should be automatically generated. The **-s** option specifies that the  $t$  values should be stripped from the output.

### AUTHORS

**spline** was written by Robert S. Maier ([rsm@math.arizona.edu](mailto:rsm@math.arizona.edu)), starting with an earlier version by Rich Murphey ([rich@freebsd.org](mailto:rich@freebsd.org)). The algorithms for constructing splines under tension are similar to those used in the FITPACK subroutine library, and are ultimately due to Alan K. Cline ([cline@cs.utexas.edu](mailto:cline@cs.utexas.edu)).

### SEE ALSO

"The GNU Plotting Utilities Manual".

### BUGS

Email bug reports to [bug-gnu-utils@gnu.org](mailto:bug-gnu-utils@gnu.org).

**NAME**

tek2plot – translate Tektronix files to other graphics formats

**SYNOPSIS**

**tek2plot** [ *options* ] [ *files* ]

**DESCRIPTION**

**tek2plot** translates Tektronix graphics files to other formats, or displays them on an X Window System display. The output format or display type is specified with the **-T** option. The possible output formats and display types are the same as those supported by **graph**(1), **plot**(1), **pic2plot**(1), and **plotfont**(1). If an output file is produced, it is written to standard output.

Options and file names may be interspersed on the command line, but the options are processed before the file names are read. If **--** is seen, it is interpreted as the end of the options. If no file names are specified, or the file name **-** is encountered, the standard input is read.

**OPTIONS****General Options**

**-T** *type*

**--display=***type type*

Select *type* as the output format or display type. It may be "X", "png", "pnm", "gif", "svg", "ai", "ps", "cgm", "fig", "pcl", "hpgl", "regis", "tek", or "meta" (the default). These refer respectively to the X Window System, PNG (Portable Network Graphics) format, portable anymap format (PBM/PGM/PPM), a pseudo-GIF format that does not use LZW encoding, the new XML-based Scalable Vector Graphics format, the format used by Adobe Illustrator, Postscript or Encapsulated Postscript (EPS) that can be edited with **idraw**(1), CGM format (by default, conforming to the WebCGM profile), the format used by the **xfig**(1) drawing editor, the Hewlett-Packard PCL 5 printer language, the Hewlett-Packard Graphics Language, ReGIS graphics format (which can be displayed by the **dxterm**(1) terminal emulator or by a VT330 or VT340 terminal), Tektronix format itself, and device-independent GNU metafile format. Unless *type* is "X", an output file is produced and written to standard output.

Omitting the **-T** option is equivalent to specifying **-T meta**. GNU metafile format may be translated to other formats with **plot**(1).

**-p** *n*

**--page=***number n*

Output only page number *n*, within the Tektronix file or sequence of Tektronix files that is being translated. *n* must be a non-negative integer, since a Tektronix file may consist of one or more pages, numbered beginning with zero.

The default behavior if the **-p** option is not used is to output all nonempty pages in succession. For example, **tek2plot -T X** displays each Tektronix page in its own X window. If the **-T png**, **-T pnm**, **-T gif**, **-T ai**, or **-T fig** option is used, the default behavior is to output only the first nonempty Tektronix page, since files in those output formats contain only a single page of graphics.

Most Tektronix files consist of either one page (page #0) or two pages (an empty page #0, and page #1). Tektronix files produced by the GNU plotting utilities (e.g., by **graph -T tek**) are normally of the latter sort.

**-F** *name*

**--font=***name name*

Use the font *name* for rendering the native Textronix fonts, if it is available. The default font is "Courier" except for **tek2plot -T png**, **tek2plot -T pnm**, **tek2plot -T gif**, **tek2plot -T hpgl**, **tek2plot -T regis**, and **tek2plot -T tek**, for which it is "HersheySerif". A list of available fonts can be obtained with the **--help-fonts** option (see below). If a font outside the Courier family is used, the **--position-chars** option (see below) should probably be specified.

The **-F** option is useful only if you have a Tektronix file that draws text using native Tektronix fonts. Tektronix files produced by the GNU plotting utilities (e.g., by **graph -T tek**) do not use native Tektronix fonts: they use Hershey vector fonts instead.

**-W** *line\_width*

**--line-width** *line\_width*

Set the width of lines, as a fraction of the width of the display, to be *line\_width*. A negative value means that a default value should be used. This value is format-dependent. The interpretation of zero line width is also format-dependent (in some output formats, a zero-width line is the thinnest line that can be drawn; in others, a zero-width line is invisible).

**--bg-color** *name*

Set the color used for the background to be *name*. This is relevant only to **tek2plot -T X**, **tek2plot -T png**, **tek2plot -T pnm**, **tek2plot -T gif**, **tek2plot -T svg**, **tek2plot -T cgm**, and **tek2plot -T regis**. An unrecognized name sets the color to the default, which is "white". The environment variable **BG\_COLOR** can equally well be used to specify the background color. If the **-T svg** or **-T cgm** option is used, an output file without a background may be produced by setting the background color to "none".

If the **-T png** or **-T gif** option is used, a transparent PNG file or a transparent pseudo-GIF, respectively, may be produced by setting the **TRANSPARENT\_COLOR** environment variable to the name of the background color.

**--bitmap-size** *bitmap\_size*

Set the size of the graphics display in which the plot will be drawn, in terms of pixels, to be *bitmap\_size*. The default is "570x570". This is relevant only to **plot -T X**, **plot -T png**, **plot -T pnm**, and **plot -T gif**. If you choose a rectangular (non-square) window size, the fonts in the plot will be scaled anisotropically, i.e., by different factors in the horizontal and vertical directions. For **plot -T X**, this requires an X11R6 display. Any font that cannot be scaled in this way will be replaced by a default scalable font, such as the vector font "HersheySerif".

The environment variable **BITMAPSIZE** can equally well be used to specify the window size. For backward compatibility, the X resource **Xplot.geometry** may be used instead.

**--emulate-color** *option*

If *option* is *yes*, replace each color in the output by an appropriate shade of gray. This is seldom useful, except when using **tek2plot -T pcl** to prepare output for a PCL 5 device. (Many monochrome PCL 5 devices, such as monochrome LaserJets, do a poor job of emulating color on their own.) You may equally well request color emulation by setting the environment variable **EMULATE\_COLOR** to "yes".

**--max-line-length** *max\_line\_length*

Set the maximum number of points that a polygonal line may contain, before it is flushed out, to be *max\_line\_length*. If this flushing occurs, the polygonal line will be split into two or more sub-lines, though the splitting should not be noticeable. The default value of *max\_line\_length* is 500.

The reason for splitting long polygonal lines is that some display devices (e.g., old Postscript printers and HP-GL pen plotters) have limited buffer sizes. The environment variable **MAX\_LINE\_LENGTH** can also be used to specify the maximum line length.

**--page-size** *pagesize*

Set the size of the page on which the plot will be positioned. This is relevant only to **tek2plot -T svg**, **tek2plot -T ai**, **tek2plot -T ps**, **tek2plot -T cgm**, **tek2plot -T fig**, **tek2plot -T pcl**, and **tek2plot -T hpgl**. The default is "letter", which means an 8.5 inch by 11 inch page. Any ISO page size in the range "a0"..."a4" or ANSI page size in the range "a"..."e" may be specified ("letter" is an alias for "a" and "tabloid" is an alias for "b"). "legal" and "ledger" are recognized page sizes also. The environment variable **PAGESIZE** can equally well be used to specify the page size.

The graphics display in which the plot is drawn will be a square region that would occupy nearly the full width of the specified page. An alternative size for the graphics display can be specified. For example, the page size could be specified as "letter,xsize=4in,ysize=6in", or "a4,xsize=5.0cm,ysize=100mm". For all of the above except **tek2plot -T hpgl**, the graphics display will, by default, be centered on the page. For all of the above except **tek2plot -T svg** and **tek2plot -T cgm**, the graphics display may be repositioned manually, by specifying the location of its lower left corner, relative to the lower left corner of the page. For example, the page size could be specified as "letter,xorigin=2in,yorigin=3in", or

"a4,xorigin=0.5cm,yorigin=0.5cm". It is also possible to specify an offset vector. For example, the page size could be specified as "letter,xoffset=1in", or "letter,xoffset=1in,yoffset=1.2in", or "a4,yoffset=-1cm". In SVG format and WebCGM format it is possible to specify the size of the graphics display, but not its position.

**--pen-color *name***

Set the pen color to be *name*. An unrecognized name sets the pen color to the default, which is "black".

**--position-chars**

Position the characters in each text string individually. If the text font is not a member of the Courier family, and especially if it is not a fixed-width font, this option is recommended. It will improve the appearance of text strings, at the price of making it difficult to edit the output file with **xfig(1)**, **idraw(1)**, or Illustrator.

**--rotation *angle***

Rotate the graphics display by *angle* degrees. Recognized values are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. The environment variable **ROTATION** can also be used to specify a rotation angle.

**--use-tek-fonts**

Use the bitmap fonts that were used on the original Tektronix 4010/4014 terminal. This option is relevant only to **tek2plot -T X**. The four relevant bitmap fonts are distributed with most versions of the GNU plotting utilities, under the names "tekfont0"..."tekfont3". They can easily be installed on any modern X Window System display. For this option to work properly, you must also select a window size of 1024×1024 pixels, either by using the **--bitmap-size 1024x1024** option or by setting the value of the **Xplot.geometry** resource. This is because bitmap fonts, unlike the scalable fonts that **tek2plot** normally uses, cannot be rescaled.

This option is useful only if you have a file in Tektronix format that draws text using native Tektronix fonts. Tektronix files produced by the GNU plotting utilities (e.g., by **graph -T tek**) do not use native Tektronix fonts: they use Hershey vector fonts instead.

### Options for Metafile Output

The following option is relevant only if the **-T** option is omitted or if **-T meta** is used. In this case **tek2plot** outputs a GNU graphics metafile, which must be translated to other formats with **plot(1)**.

**-O**

**--portable-output**

Output the portable (human-readable) version of GNU metafile format, rather than a binary version (the default). The format of the binary version is machine-dependent.

### Informational Options

**--help** Print a list of command-line options, and exit.

**--help-fonts**

Print a table of available fonts, and exit. The table will depend on which output format or display type is specified with the **-T** option. **tek2plot -T X**, **tek2plot -T svg**, **tek2plot -T ai**, **tek2plot -T ps**, **tek2plot -T cgm**, and **tek2plot -T fig** each support the 35 standard Postscript fonts. **tek2plot -T svg**, **tek2plot -T pcl**, and **tek2plot -T hppl** support the 45 standard PCL 5 fonts, and the latter two support a number of Hewlett-Packard vector fonts. All seven support a set of 22 Hershey vector fonts, as do **tek2plot -T png**, **tek2plot -T pnm**, **tek2plot -T gif**, **tek2plot -T regis**, and **tek2plot -T tek**. **tek2plot** without a **-T** option in principle supports any of these fonts, since its output must be translated to other formats with **plot(1)**.

The **plotfont(1)** utility may be used to obtain a character map of any supported font.

**--list-fonts**

Like **--help-fonts**, but lists the fonts in a single column to facilitate piping to other programs. If no output format is specified with the **-T** option, the full set of supported fonts is listed.

**--version**

Print the version number of **tek2plot** and the plotting utilities package, and exit.

## ENVIRONMENT

The environment variables **BITMAPSIZE**, **PAGESIZE**, **BG\_COLOR**, **EMULATE\_COLOR**, **MAX\_LINE\_LENGTH** and **ROTATION** serve as backups for the options **--bitmap-size**, **--page-size**, **--bg-color**, **--emulate-color**, **--max-line-length**, and **--rotation**, respectively. The remaining environment variables are specific to individual output formats.

**tek2plot -T X**, which pops up a window on an X Window System display and draws graphics in it, checks the **DISPLAY** environment variable. Its value determines the display that will be used.

**tek2plot -T png** and **tek2plot -T gif**, which produce output in PNG format and pseudo-GIF format respectively, are affected by the **INTERLACE** environment variable. If its value is "yes", the output will be interlaced. Also, if the **TRANSPARENT\_COLOR** environment variable is set to the name of a color, that color will be treated as transparent in the output.

**tek2plot -T pnm**, which produces output in portable anymap (PBM/PGM/PPM) format, is affected by the **PNM\_PORTABLE** environment variable. If its value is "yes", the output will be in a human-readable format rather than binary (the default).

**tek2plot -T cgm**, which produces output in CGM (Computer Graphics Metafile) format, is affected by the **CGM\_MAX\_VERSION** and **CGM\_ENCODING** environment variables. By default, it produces a binary-encoded version of CGM version 3 format. For backward compatibility, the version number may be reduced by setting **CGM\_MAX\_VERSION** to "2" or "1". Irrespective of version, the output CGM file will use the human-readable clear text encoding if **CGM\_ENCODING** is set to "clear\_text". However, only binary-encoded CGM files conform to the WebCGM profile.

**tek2plot -T pcl**, which produces PCL 5 output for Hewlett-Packard printers and plotters, is affected by the environment variable **PCL\_ASSIGN\_COLORS**. It should be set to "yes" when producing PCL 5 output for a color printer or other color device. This will ensure accurate color reproduction by giving the output device complete freedom in assigning colors, internally, to its "logical pens". If it is "no" then the device will use a fixed set of colored pens, and will emulate other colors by shading. The default is "no" because monochrome PCL 5 devices, which are much more common than colored ones, must use shading to emulate color.

**tek2plot -T hppl**, which produces Hewlett-Packard Graphics Language output, is affected by several environment variables. The most important is **HPGL\_VERSION**, which may be set to "1", "1.5", or "2" (the default). "1" means that the output should be generic HP-GL, "1.5" means that the output should be suitable for the HP7550A graphics plotter and the HP758x, HP7595A and HP7596A drafting plotters (HP-GL with some HP-GL/2 extensions), and "2" means that the output should be modern HP-GL/2. If the version is "1" or "1.5" then the only available fonts will be vector fonts, and all lines will be drawn with a default width (the **-W** option will not work).

The position of the **tek2plot -T hppl** graphics display on the page can be rotated 90 degrees counter-clockwise by setting the **HPGL\_ROTATE** environment variable to "yes". This is not the same as the rotation obtained with the **--rotation** option, since it both rotates the graphics display and repositions its lower left corner toward another corner of the page. Besides "no" and "yes", recognized values for **HPGL\_ROTATE** are "0", "90", "180", and "270". "no" and "yes" are equivalent to "0" and "90", respectively. "180" and "270" are supported only if **HPGL\_VERSION** is "2" (the default).

By default, **tek2plot -T hppl** will draw with a fixed set of pens. Which pens are present may be specified by setting the **HPGL\_PENS** environment variable. If **HPGL\_VERSION** is "1", the default value of **HPGL\_PENS** is "1=black"; if **HPGL\_VERSION** is "1.5" or "2", the default value of **HPGL\_PENS** is "1=black:2=red:3=green:4=yellow:5=blue:6=magenta:7=cyan". The format should be self-explanatory. By setting **HPGL\_PENS** you may specify a color for any pen in the range #1..#31. All color names recognized by the X Window System may be used. Pen #1 must always be present, though it need not be black. Any other pen in the range #1..#31 may be omitted.

If **HPGL\_VERSION** is "2" then **tek2plot -T hppl** will also be affected by the environment variable **HPGL\_ASSIGN\_COLORS**. If its value is "yes", then **tek2plot -T hppl** will not be restricted to the palette specified in **HPGL\_PENS**: it will assign colors to "logical pens" in the range #1..#31, as needed. The default value is "no" because other than color LaserJet printers and DesignJet plotters, not many HP-GL/2 devices allow the assignment of colors to logical pens.

The drawing of visible white lines is supported only if **HPGL\_VERSION** is "2" and the environment variable **HPGL\_OPAQUE\_MODE** is "yes" (the default). If its value is "no" then white lines (if any), which are normally drawn with pen #0, will not be drawn. This feature is to accommodate older HP-

GL/2 devices. HP-GL/2 pen plotters, for example, do not support the use of pen #0 to draw visible white lines. Some older HP-GL/2 devices may, in fact, malfunction if asked to draw opaque objects.

**SEE ALSO**

**plot(1)**, **plotfont(1)**, and "The GNU Plotting Utilities Manual".

**AUTHORS**

**tek2plot** was written by Robert S. Maier ([rsm@math.arizona.edu](mailto:rsm@math.arizona.edu)). It incorporates a Tektronix parser written by Edward Moy ([moy@parc.xerox.com](mailto:moy@parc.xerox.com)).

**BUGS**

Email bug reports to [bug-gnu-utils@gnu.org](mailto:bug-gnu-utils@gnu.org).