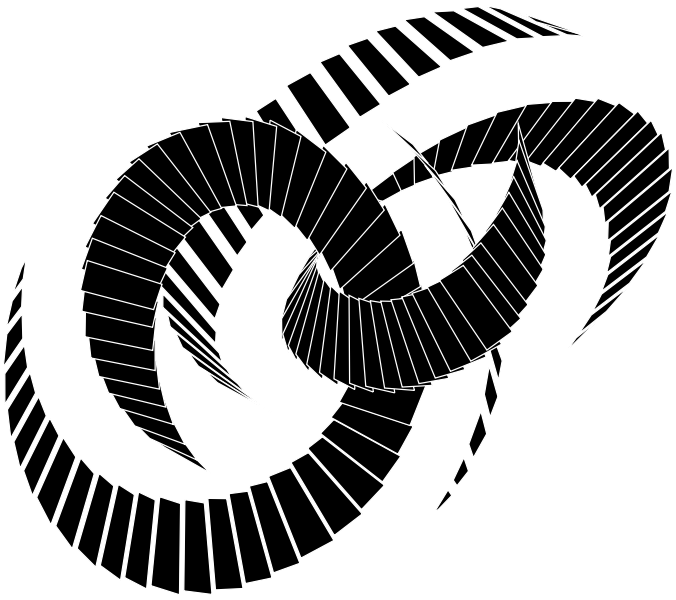
KnotPlot

A Program for Viewing Mathematical Knots

July 11, 2018



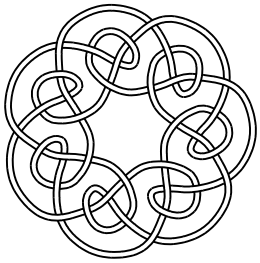
Rob Scharein

Hypnagogic Software
Vancouver, British Columbia, Canada

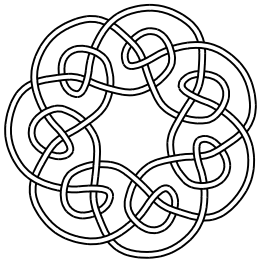
(email: rob@hypnagogic.net)
(WWW: knotplot.com)

Introduction

This highly inadequate manual vaguely describes KnotPlot, a program to display knots and links in three and four dimensions. KnotPlot is probably one of the largest programs ever written which serves no clearly defined purpose whatsoever. However, it is a great deal of fun, mostly because knots are such nice things to look at.



Section 1: About KnotPlot



KnotPlot is a program for visualizing and interacting with 3-D and 4-D knots. KnotPlot started as a course project several years ago under the guidance of Kellogg Booth and Dale Rolfsen. Since then the program has steadily accreted new features, of which some of the most interesting are the following:

Knot catalogue — The complete appendix C of Rolfsen's book *Knots and Links* [Rol76] is on-line (about 384 knots and links). Many other knots of a special nature are also included. This database makes possible many oppurtunities for experimentation.

Automatic construction — Knots and links may be constructed using the notation system developed by Conway [Con70]. The Conway notation is parsed into a string of commands that are interpreted by a *tangle calculator*. Also special knots and links such as knot chains, torus knots (and links), and Lissajous knots [BHJS94] may be constructed automatically.

Sketching — Knots may be sketched by hand in 3-D, marking the location of individual points along the knot (called *beads*) with the mouse and using the mouse buttons to indicate over and under crossings.

Knot transformations — Knots may be transformed into new knots via a number of procedures. This is used to create satellite knots, doubled knots, and cabled knots [Rol76, Kau87].

Knot relaxation — Once a knot is entered into KnotPlot it may be *relaxed* into a smoother configuration by applying a system of forces to it.

Interactive manipulation — Knots may be interactively modified by direct manipulation. This can take the form of "dragging" the knot around in three dimensions or by selectively applying forces to local regions of the knot.

Topological properties — The Alexander and HOMFLY polynomials, writhe, average crossing number, thickness [Sim96] and Dowker code [DT83, Thi85] of a knot can be computed. The polynomial calculations are an example of the use of KnotPlot in conjunction with other software.

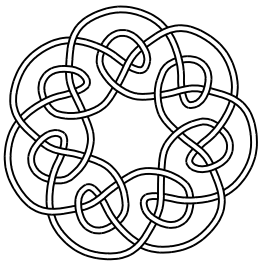
Experimental knot theory — KnotPlot has a simple but effective programming model that allows users to search for minimal stick conformations and interesting random knots [Mil96] easily.

Graphs, open strings, and braids — Knotted graphs [SW90, Han89] are handled (almost) as easily as normal knots. Also, knots or links don't have to be closed strings. Commands are available that allow for the easy generation of arbitrary braids.

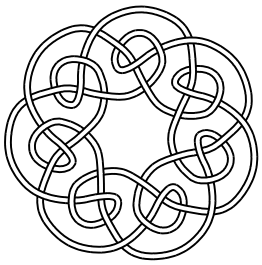
4D surfaces — Knotted and linked spheres [AC59, Rol76] in four dimensions can be interactively constructed.

Graphical display — A large number of options are available for changing the display, many are bizarre or abstract to the point where they are no longer recognizable as knots [HBK+94, New96].

External support — KnotPlot can export surface models to a number of external software platforms, including Blender, Unity 3D, Maya, AutoCAD, and METAFONT. Also, PostScript pictures in numerous flavours may be generated.



Section 2: Running KnotPlot



2.1 Starting it up

KnotPlot can be run in two ways, either by double clicking the application or from the command line in a terminal application of some sort. The latter is the preferred method for any research project. To run the program within a terminal, change to the appropriate directory and enter knotplot at the command line prompt. Please note that KnotPlot should generally be run from a directory where you have write access. KnotPlot will open three windows: the main view window, the control panel, and the command line window. KnotPlot commands may be entered when either the command line window or the control panel have keyboard focus. The keyboard has a different meaning when the main view window has keyboard focus. Once all three windows are opened, they may be repositioned or resized as desired.

When starting from the command line, the default behaviour of the program can be modified by supplying one of several command line options:

-nog Starts KnotPlot in non-graphics mode. This is the main technique for running KnotPlot scripts in batch mode.

-nopanel Will start the program up without the control panel.

-debug Turns on debugging information.

2.2 Entering commands

There are three modes of interaction with the program:

command line — Many of the features of KnotPlot are available only through this mode. Although the program echos commands in the command line window, the mouse can be positioned in either the command window or the control panel as previously mentioned.

control panel — Some commands are only available by using the control panel. Other operations, such as setting various parameters may be easier to do with the control panel.

menu — Clicking the right mouse button in either the control panel window or the command line window will bring up a pop-up menu. On a Mac, ignore the standard Apple menus (for now).

2.3 Loading knots

A complete catalogue of all the knots in appendix C of Dale Rolfsen's book *Knots and Links* [Rol76] is included on line. The coördinates for these knots were kindly provided by Professor Rolfsen and then relaxed using KnotPlot. The file naming convention is as follows:

- For single component knots, the file *C.n* contains the knot C_n.
- For multi-component links, the file *C.k.n* contains the link C_n^k.

So for example, to load the knot 10_{102} you would type the command "**load 10.102**"

For demo purposes, knots can also be selected at random from the data base by using the command "**knot**" (which can be abbreviated to the single letter **k**). To randomly select a knot with a specific number of components use the command "**knot** n" where n is the number of components you want (from 1 to 4). To randomly select a knot from the "special

collection", use the command "**knot s**". This collection of knots contains a number of unusual knots, or knots in non-standard configurations, as well as several composite[1] knots.

2.4 Changing the view

The view seen of the knot or link may be changed by using one of the mouse buttons in the view window. Note that these commands affect only the view seen in the view window. PostScript output is affected by only rotations. Changing the view window has no effect on any geometry exported in the form of surface models or knot files.

2.4.1 Rotations

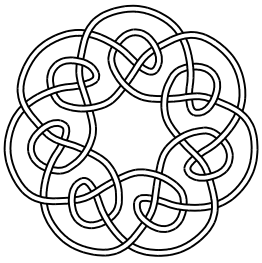
The knot may be rotated with the **rotate** command (see below). However it is much simpler to use the "virtual trackball" [CMS88] which can be thought of as enclosing the knot. The best way to learn how to use the trackball is to try it out. Once a knot is loaded, move the mouse to the view window and click down with the left mouse button. Now move the mouse and watch how the knot rotates.

2.4.2 Scaling

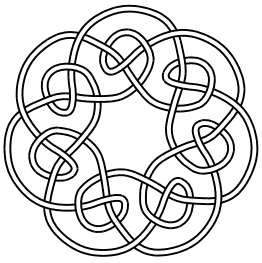
To scale the knot (actually the viewing transform), click down in the view window with the right mouse button. Move the mouse to the right to increase the scale and to the left to decrease. This action is identical to changing the value of the parameter **scale**.

[1] A composite knot is a knot that can be cut into several simpler knots (this can be defined rigorously). Tying two knots consecutively in the same piece of string corresponds roughly to the idea of a composite knot. If a knot (or link) is not composite, then it is called *prime*.

Figure 1: Top part of the main control panel.



Section 3: Control panels



3.1 Main control panel

The top part of the main control panel is shown in Figure 1.

- There are two rows of tabs at the top of all control panels. Use these to go to other control panels.
- To reset the display, click on the 'reset' button in the right of the display section. This reset button also resets pretty much everything else in the program, including 4D parameters, PostScript output, the dynamics, and what knot.
- Yes, it's true, the 'exit' button terminates KnotPlot. You can also do this by typing in the commands quit, exit, or stop, by hitting the ESC-key. Be warned that KnotPlot (currently) doesn't warn you about unsaved changes.
- Next is a row of buttons to load some knots, randomly selected, from the knot catalogue. The 'All' button chooses a knot from the entire catalogue, the numbered buttons randomly choose a link with that many components. The 'spec coll' button picks a knot selected at random from the Special Collection.[2]

[2]Knots and link that are "special" for some reason (mathematically interesting or knots with names).

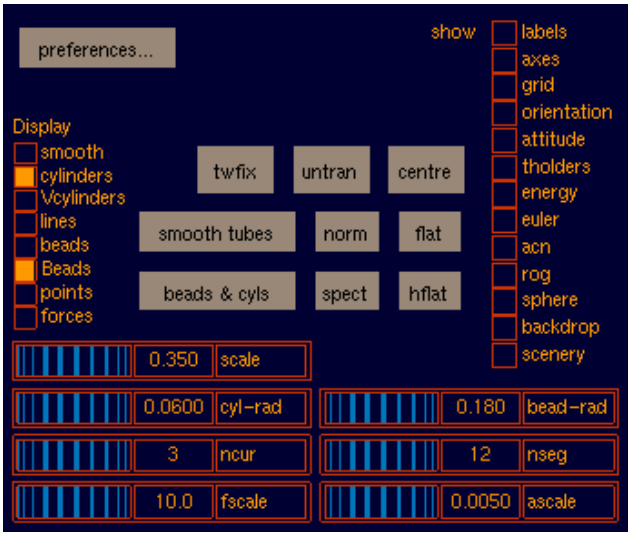


Figure 2: Display part of the main control panel.

- The next row of buttons bring up various knot "zoos": 'Rand', a zoo with randomly chosen knots, then 'A' thru 'F' are the Rolfsen catalog, then torus knots and randomly selected torus knots.

All the buttons and sliders shown in Figure 2 control how the knot is displayed. In addition, some of them will affect any surface geometry exported by KnotPlot. Nothing in this section affects the underlying mathematical object (that is, what is read and written with the load and save commands.[3]

To get a feel for how things work try loading a knot at random (see above) and then click on the two large buttons labelled 'smooth tubes' and 'beads & cyls'. This will show you the two principal methods of displaying knots. The beads and cylinders closely approximate the actual mathematical object (a PL-knot). KnotPlot starts out in a smooth tube mode, mostly because knots look prettier in this mode.

- You can rotate the view seen in the view window by using a virtual trackball (left mouse) and scale the view by using the right mouse in the view window. These transformations may be undone, and the view restored to the default orientation by clicking on the 'untran' button.

- Sometimes, especially if nseg is too small, you'll see a funny crimp in the knot or link's surface at the point at which the drawing begins and ends for each component. There are technical reasons for this that are explained elsewhere. For the present you can

[3]With the exception of the 'centre' button, see the centre command.

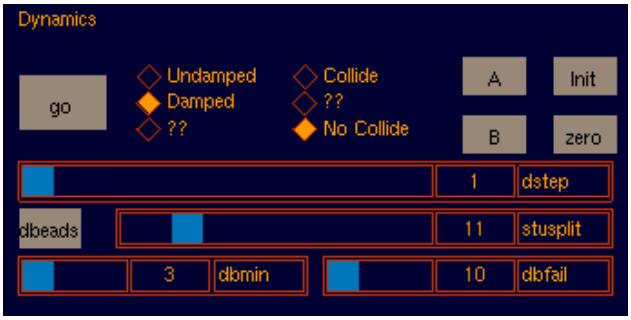


Figure 3: Dynamics part of the main control panel.

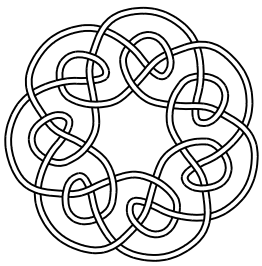
get rid of this deficit angle by clicking on the 'twfix' button (twist fix). There is also a twfix command that does the same job (and more).

- The 'norm', 'spec', 'flat', 'hflat' buttons need explaining, try them out when you're in 'smooth tubes' mode.

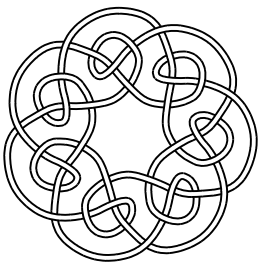
The bottom part of the main control panel (Figure 3) allows the dynamical model to be interactively modified.

- The default dynamical model is a damped force law with collision avoidance. This is what you get if you click the button 'A' in the dynamics section. The other main force law of interest is an undamped force law without collision checking. You can get this by clicking 'B'. Another interesting combination is 'Damped' with 'Collide'. This speeds up the relaxation but allows for possible change of knot type. The 'Init' button initializes the dynamical model to the default setting (including all the slider positions).
- Click on the 'go' button to watch the knot relax. The 'go' button will turn red while relaxation is taking place.
- The 'zero' button zeros the velocities of all the beads. This is only applicable to undamped relaxations.

All of the parameters for the dynamics may be changed while relaxation is taking place. The same thing applies to any display parameters.



Section 4: Commands



This is a description of some of the commands available in KnotPlot. Commands may be abbreviated to any length. However it is safer to never abbreviate a command to *less than four characters.* Commands are guaranteed to be unambiguous if this general rule is followed. Command arguments that take on numerical values may be set in the same way as parameters (except without an equal sign). See the explanation at the beginning of that section.

4.1 General purpose commands

< filename — Reads the file *filename* and executes the commands found there. Same as the read command.
Example: <myscript.txt

. shell-command — Sends the string *shell-command* to the UNIX or UNIX-like shell to be executed there.

exit — Same as the quit command.

function number action — With no arguments, prints the current binding of the function keys. With one argument, prints the binding for that key. Otherwise the function key *number* is bound to *action*. See the demo files and the knotplotrc file for examples of use.

information — Prints out some useful information about the current knot, such as the number of beads and components, and the extent in the three orthogonal directions.

nap number — Same as pause except on SGIs, where there is a difference but not a very interesting one.

nop — This command does *nothing at all.*

panel set which — Sets the control panel to *which*. The same effect can be achieved using the pop-up menus.

parameters — Prints the current settings of all parameters.

parameters name — Prints the value of any parameters starting with the string *name.*

path [directory] — If *directory* is omitted, this command prints the current read and write paths.

pause number — Pauses execution for *number* seconds. This is only useful for sessions run from a script.

quit — Same as the stop command.

seed value — Sets the random number seed to *value.*

stop — Same as the exit command.

version — Prints out some useful information, including how to find the KnotPlot home directory (the root of the KnotPlot distribution) and the resource directory.

4.2 Loading and saving knots and links

allocate nbeads — Allocates enough memory for *nbeads* number of beads.

knot — Loads a knot or link from the catalogue, selected at random. Very useful for demos.

knot n — Same as above, but loads a knot with n components.

knot number n — Loads the n^{th} knot or link from the tables. The valid range for n is from 1 to 384.

knot special — Loads a knot at random from the "special collection" The special collection contains several oddities as well as a few composite knots.

load filename — Loads a knot from the file *filename.* Note that *filename* may be in any of the formats that the save command can write.

load combine filename — Same as above except a new link is created by combining the current knot and the knot in the file *filename* (*i.e.* creates the union of the two knots or links).

load sum filename — Same as above except a new knot is formed by creating the direct sum of the current knot and the knot in file *filename.* This command uses the shift, revbeads, and translate to commands to move everything into the correct position. If you find you don't like the results it gives, you may have to do these operations as well as cut and join yourself. One limitation is that KnotPlot expects both summands to be proper knot*s.*

save filename [format] — Saves the current knot or link to the file *filename*. If *format* is omitted, the file is saved in the format given by the current value of sformat. If *format* is an integer, then that value is used as the save format and **sformat** is ignored. Otherwise, if *format* is **float** (or **LOCF**), **short** (or **LOCS**), **char** (or **LOCC**), **ascii** (or **raw**), **ked**, or **ming** then the file is saved with that format instead of the format specified by **sformat** (see the description of **sformat**). Including the second parameter is useful for temporarily overriding the default save format.

zoo start size — Generates a "knot zoo" of size *size*×*size* starting with the knot number *start* in the standard catalogue. If *size* is omitted, it will default to 5. If *start* is 0, a random selection of knots is loaded. This command uses singlebuffer mode so that the knots can be seen while they are being loaded. Clicking with the left mouse on an image in the zoo will load that knot or link.

4.3 Creating new knots and links

These commands will create something from scratch. In addition you could load a knot or link from the catalogues with the load command or sketch one yourself. See also the construct command below, which can be used to construct user-defined things. The commands **chain**, **lissajous**, **2lissajous**, **torus**, and **unknot** are affected by the new parameter.

chain ncomp [nbeads] — Creates a linked chain with *ncomp* components (this number should be even) of *nbeads* each. If *nbeads* is omitted, it defaults to 9. The parameters D-chain and d-chain control the geometry of the chain.

conway word — Constructs the knot with Conway notation *word*. This command will show you the equivalent tangle command. Examples:
conway 3
conway 252
conway 3,3,3

lissajous nx ny nz [phix phiy nbeads] — Constructs a Lissajous knot [BHJS94] of type *(nx, ny, nz)* and phase *(phix, phiy)* with *nbeads* number of beads. If *nbeads* is omitted, the number of beads defaults to N-torus. *phix* and *phiy* both default to 0 if omitted, and that's not too interesting. For best results they should have different values.

2lissajous nx ny nz nk [phix phiy phiz nbeads] — Similar to the lissajous command except constructs a 2-mode Lissajous knot.

random tangle — Creates a random tangle using the tangle calculator. This is one of several methods of creating random knots and links that will be implemented in KnotPlot in the near future.

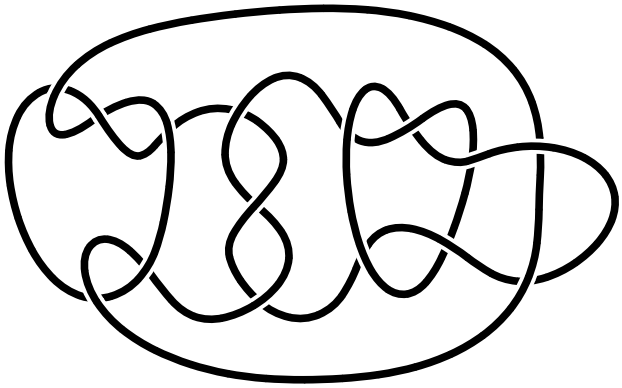
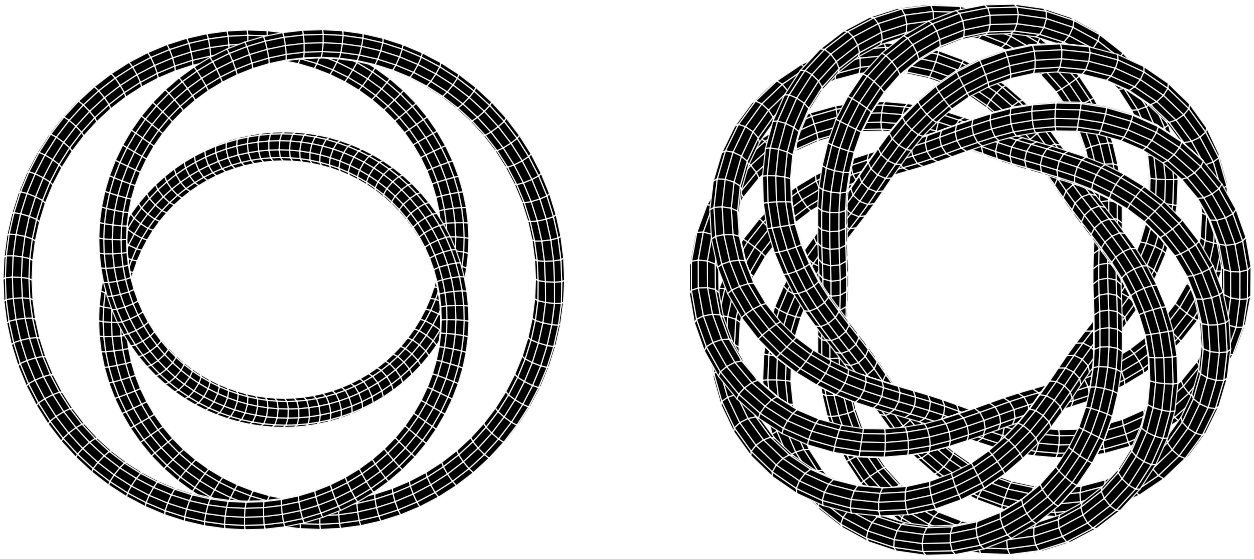


Figure 4: 15 crossing mutant of a 13 crossing knot created with the tangle calculator.

tangle tangle-string — Uses the tangle calculator to execute the commands in *tangle-string.* See Figure 4 for an example of a knot created with the tangle calculator (this example from a paper by Morwen Thistlethwaite [Thi85]). An interactive version of the tangle calculator can be accessed using the pop-up menus.



The torus knot $K_{3,2}$ (the trefoil) The torus link $K_{6,10}$

Figure 5: Some nice torus knots.

torus n m [beads] — Produces the torus knot $K_{n,m}$. If *beads* is omitted, the number of beads defaults to N-torus. Other parameters controlling how the knot is constructed are R-torus and d-torus. If two few beads are used, the results can be bizarre. If n and m are not relatively prime, then a torus link with gcd (n, m) components is created. You can also omit n and/or m to have values chosen randomly. Figure 5 shows a few nice torus knots.

unknot [B [R]] — Creates an unknot with B beads of radius R. in the xy-plane. If B is omitted, the number of beads defaults to N-torus. If R is omitted, the radius defaults to R-torus.

4.4 Transformation tools

Knots or links can be transformed into new knots or links using the commands in this section. Other commands that could belong in this section are the jitter, mangle, and split commands as some of these may change the knot type.

cable — Takes the current knot and *cables* it.

close compA [**to** compB] — With one argument, closes component *compA* if it is open, otherwise closes all components in the range *compA* to *compB*. Without any arguments, prints out information on all components. See the open command.

close all — Closes all components.

companion — Transforms the current knot (one component only) into its "companion" knot.

cut bead — Cuts a component at location *bead*. If the component is closed, it becomes open. Otherwise two open components result. Small components (less than four beads) are automatically reaped. See the join command.

delete n — Deletes component n of the current link. Components are numbered starting at 0. See the keep command.

delete all — Deletes the current knot or link.

double — Takes the current knot and *doubles* it.

join a b — Joins together beads a and b. Both a and b must be the endpoints of an open component. See the cut command.

identify a b — Identifies bead a and bead b. This is how knotted graphs are implemented.

identify init — Initializes (unidentifies) all bead identifications. Use this whenever specifying new identifications.

keep component — Deletes all knot components except *component*. Components are numbered starting at 0. Only makes sense for links. See the delete command.

nbeads value — If *value* begins with a '+' or a '-' sign then the number of beads is incremented or decremented by *value*, otherwise the number of beads is changed to *value*.

nbeads mult fraction — The number of beads is multiplied by *fraction*.

open compA [**to** compB] — With one argument, opens component *compA* if it is closed, otherwise opens all components in the range *compA* to *compB*. Without any arguments, prints out information on all components. See the close command.

open all — Opens all components.

revbeads [comp] — Reverses the beads in component *comp*. If *comp* is omitted all beads in each component are reversed. Useful in combination with the shift command.

shift distance [comp] — Shifts the bead numbering in component *comp* by *distance*, which may be positive or negative. If *comp* is omitted, all components are affected. Use the special command to turn on bead labels.

swap compa compb — Swaps components *compa* and *compb*.

4.5 Relaxing knots and links

If you're running in an interactive mode, you should almost never need to enter any of these commands, since everything can be controlled using the main control panel.

collision [**allow** |**none** |**fast**] — If the argument is omitted, prints the current collision constraint. If set to **allow** collision checking is disabled. Mode **none** is obsolete and should not be used. Defaults to **fast.**

go niter — Performs *niter* iterations of the knot relaxation. Relaxation is performed in the background so you rotate the knot or change display parameters whilst it relaxes. Omit the parameter if you want the knot to keep relaxation until instructed to stop. Use go 0 to stop relaxation.

ago niter — Performs *niter* iterations of the knot relaxation *atomically*.

4.6 Graphics display

These commands affect the display of the knot or link. It is possible that they will also affect any output geometry in the cases where the type of surface (broken, smooth) or quality is changed.

blendfunction sfactr dfactr — Sets the blendfunction (see the GL manual). The parameters must be one of **zero one dc mdc sa msa da mda min-sa-mda sc msc.** This command is an exception to most commands in that abbreviations are not allowed for the parameters. The default is **blendfunction sa msa.** This command may not work on all machines.

cheapo — Sets nsegments to chnseg and ncurve to chncurve. This is useful if a cheaper quality rendering is desired in exchange for a faster drawing time. See luxo command.

clrfun [what] — Sets the clear functions used each time something is drawn. The value supplied for *what* should be one of the following:

- **fz | zf** The frame buffer and z-buffer will be cleared (this is the default).
- **z** Only the z-buffer will be cleared.
- **f** Only the frame buffer will be cleared.
- **n** Neither the frame buffer nor the z-buffer will be cleared.

This function is useful for several neat tricks, include painting and getting beads to leave traces during relaxation. Enter the command without an argument to see the buffers which are currently being cleared.

display — Forces display to be updated, even if nothing has changed.

display true — Forces the display to be updated, even if the parameter Drawing is set to **off**.

draw what — Changes the way the knot is drawn. Currently only five settings are of interest:

- **normal** Sets the drawing to its normal setting.
- **spectrum** Draws the knot as a "spectrum".
- **flat** Draws the knot flat shaded.
- **broken** Draw the knot as a broken surface. Controlled by the broff parameter.
- **sbroken** Combines the **broken** and **spectrum** options above.

luxo — Similar to cheapo above except produces high quality rendering. Sets nsegments to lxnseg and ncurve to lxncurve. May be very slow on some machines.

mathsv material hue saturation value — Sets the colour of the indicated material. *material* must be one of the following:

s The knot material in smooth tubes mode.

b The bead material.

a The arrow material.

hue is in the range 0 to 360, *saturation,* and *value* must be in the range from 0 to 1.

matrgb material red green blue — Similar to mat-hsv but uses RGB space instead of HSV space.

mode mode — Sets the drawing mode. *mode* may be one of the following:

s Draws the knot as a smooth tube, interpolating between beads depending on the setting of ncurve.

c Draws cylinders between beads.

b Draws the beads.

f Draws the force vectors.

a Draws the anchors.

C Draws identification clusters (for knotted graphs).

These settings may be OR'd together, for example **mode cfb** will draw cylinders, beads, and force vectors. Use **mode sc** in conjunction with **draw spectrum** for bizarre effects.

nonprop xs ys zs — Modifies the current view matrix by a non-proportional scaling factor. Useful for special effects.

ortho x_s y_s z_s — Sets the view to an orthographic projection with clipping volume $x \in (-x_s, x_s)$, $y \in (-y_s, y_s)$, and $z \in (-z_s, z_s)$. Omit all the arguments to have KnotPlot choose reasonable defaults.

rotate [**x** | **y** | **z** | **i** | **j** | **k**] angle — Rotates the current view about the indicated axis by an amount of *angle* degrees. Note this is not the same as the about command. See the 'rotate' demo (first demo on 'DemoA' panel) to explore the difference between rotate and about and the difference between **x** and **i**.

rotate fix — Uses the current rotation matrix to modify the embedding of the knot and then resets the view matrix to the unit matrix.

rotate unit — Resets the rotation matrix to the unit matrix.

twfix — A command that will automatically adjust twist to eliminate the deficit angle seen when nsegments is too small. A demo script is available in the demos directory that illustrates the use of this command.

zbuffer [true |false] — Enables or disables z-buffering.

4.7 Geometry commands

These commands affect the embedding of the knot or link in space. They have no effect on the viewing transform but will affect any surface models exported or anything saved with the save command.

about [**x** | **y** | **z**] angle [comp] — Rotates component *comp* of the current link about the indicated axis by *angle* degrees. This may appear to be identical to the rotate command, but there is a difference. This command modifies the coordinate values of the link, thus affecting any geometric data that might be written out in a file. The **rotate** command simply modifies the viewing transformation. If *comp* is omitted, all components in the link are modified. If *comp* is the keyword **last** then only the last component is rotated.

centre — Translates the knot or link so its bounding is centered at the origin.

centre mass — Translates the knot or link so its centre of mass is at the origin.

fitto size — Fits the knot to a radius of *size.*

jitter distance — Displaces each bead by a random distance.

mangle niters α redraw centre — *Mangles* the knot (single component) *niters* times. Each time the knot is mangled two beads are chosen at random and one of the two arcs connecting the beads is rotated about the line through the beads by a random angle[4]. The angle rotated is chosen uniformly between $-\alpha$ and $+\alpha$. If *redraw* or *centre* are omitted then all mangling is done and then the mangled knot is displayed. Otherwise if *redraw* is non-zero the knot is redrawn every *redraw* manglings. The *centre* parameter is used in a similar fashion.

reflect axis [comp] — Reflects component *comp* of the current link about axis *axis* which must be one of **x, y,** or **z.** Reflections can be concatenated, for example **refl xyz** will perform an inversion through the origin. If *comp* is omitted, all components are reflected. If *comp* is the keyword **last** then only the last component is reflected.

scale xs ys zs [comp] — Scales component *comp* of the current link by the factors *xs*, *ys*, and *zs* in the obvious directions. Not to be confused with the scale parameter. If *comp* is omitted, all components are scaled. If *comp* is the keyword **last** then only the last component is scaled.

split — Splits every cylinder in two. Doubles the number of beads.

[4]This function was inspired by a talk given by K. C. Millett at the 1993 meeting of the American Mathematical Society in Vancouver.

split cylinder n — Splits the n^{th} cylinder in two.

split value — Splits any cylinder with a length greater than *value.*

swirl angle [**x** | **y** | **z** | **r**] — "Swirls" the knot by the *angle* degrees about the indictated axis. If the second parameter is omitted, it defaults to **y.** Use **r** to have a axis chosen at random. The best way to see how this command works is to try it. This is one of a (eventually) large number of *homeomorphisms* on three-space that will be available.

translate x y z [comp] — Translates component *comp* of the current link by the vector (x, y, z). This is a translation relative to the current location. If *comp* is omitted, all components are translated. If *comp* is the keyword **last** then only the last component is translated.

translate to x y z [comp] — Translates component *comp* of the current link so that its initial bead is located at the postion (x, y, z). Unlike the above command, this is an absolute translation. If *comp* is omitted, all components are translated so that bead 0 is at the indicated location.

4.8 Producing pictures

These commands produce pictures of one form or another.

ascii [xsize [ysize]] — Makes a cheapo ascii picture of the knot or link. Most useful when running without the command window or without graphics at all (for example over a modem on a VT100 terminal). Assumes that the knot fits into a bounding box of (-10, 10). You can use the fitto command to ensure this. If *ysize* is omitted, it defaults to 11/25 of *xsize.* If *xsize* is omitted, it defaults to 50.

imgout filename — Outputs the current view window as an image file.

psoption keyword setting — Sets several options for PostScript output via the psout command. Currently, *keyword* can be one of the following:

- **bbox** Changes the kind of bounding box written to the PostScript file. Here, *setting* can be one of
  - **fullpage** Bounding box will be the full page.
  - **smallest** Bounding box will just enclose the figure.
  - **square** Bounding box will be square (this is the default).
- **extension** Sets the file name extension (defaults to .eps)
- **font** Sets the font used for PostScript output. Default is ¡kbd¿/Times-Roman¡/kbd¿.

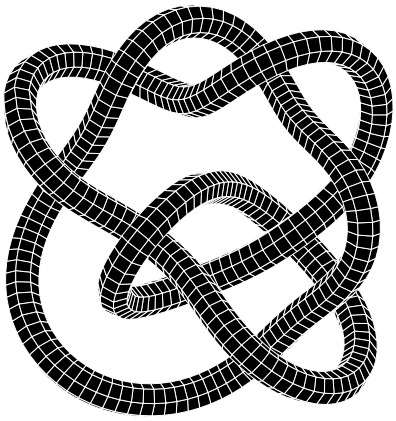


Figure 6: Example picture with **psout** command and psmode = 2.

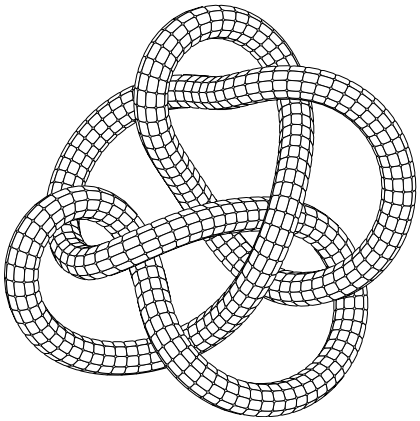


Figure 7: Example picture with **psout** command and psmode = 3.

- **format**
- **texture** Sets the file used for PostScript textures (applies to psmode = 30)

Enter the command without any arguments to see the current settings.

psout name — Outputs the current knot as a Encapsulated PostScript file (EPSF). This command may be used to produce pictures of the of the type seen in Figure **??**. For this command to work properly, the program should be set to **mode s** and cyl-rad should be around 1.0. Also, nsegments and ncurve should not be too high or enormous output files will be produced. Different effects may be achieved by changing the value of psmode. Figures **??** and 6 show the results with psmode equal to 1 and 2 respectively. The command twfix should be used first to eliminate any deficit angle.

4.9 Exporting surface models

KnotPlot can export surface descriptions to several other software platforms. **Note that all these commands assume that drawing mode 's' is currently in effect (see the mode command).** Bizarre effects will occur if this is not the case. Also they work only for one component knots. This is a minor limitation because separate models can be can be exported for each component of a multi-component link (just use the save, load, and keep commands in the right order).

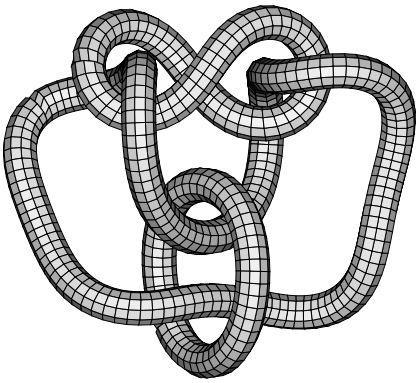


Figure 8: Example picture with **psout** command and psmode = 4.

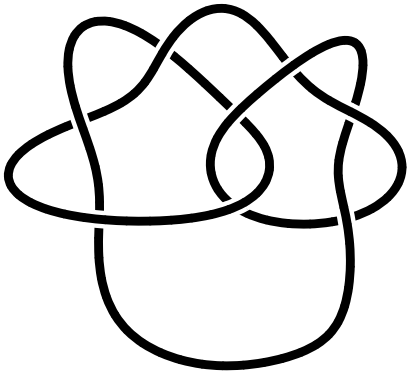


Figure 9: Example picture with **psout** command and psmode = 40.

dxf filename — Outputs a DXF NURBS[5] description to the file *filename.dxf* suitable for input into the ALIAS™ modeller.

pfdxf filename [option] — Outputs a DXF polygonal face description of the current knot to the file *filename.dxf*. If *option* is present and equal to **broken** or if the current drawing function (see the draw command) is **broken** the knot is output as a broken surface.

psdl — Outputs a SDL polygonal face description and model of the current knot to the files *knotD.sdl* and *knotM.sdl*.

rawout name — Outputs the current knot as a BPF[6] file. BPF files may be passed through a number of filters to input knots into other software packages, such as animation and raytracing programs.

sdl filename patchname shadername — Outputs a NURBS representation of the knot in a format suitable for input into the Alias renderer or raytracer. The NURBS patch is given the name *patchname* and is assigned to the shader *shadername* (defaults are **Knot** and **KnotShader** respectively). The output is written to the file *filename.sdl*. The parameter atw should be set to **on** before this command is used.

objout filename — Outputs a Alias / Wavefront polygon description of the current knot to the file *filename.obj*.

[5] **N**on-**U**niform **R**ational **B**-**S**pline. The surfaces output are actually uniform and non-rational, but that would make them BS surfaces.

[6] **B**asic **P**olygon **F**ormat.

4.10 Topology commands

Most of the commands in this section will be of interest to mathematicians. They may or may not be interesting to other people.

alexander — Computes the Alexander polynomial for the knot (single component only). Needs Mathematica somewhere on the system. Sometimes fails to give any answer, but if it does give one it is correct (I think!).

gauss — Computes the Gauss code of the current knot or link.

dowker — Computes the Dowker code of the current knot (single component only).

energy — Computes Simon's minimum distance energy [Sim94] of the current knot or link.

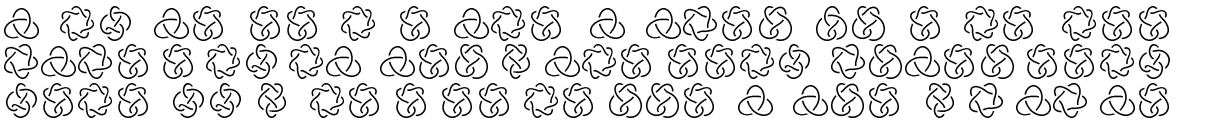
homfly — Computes the HOMFLY polynomial.

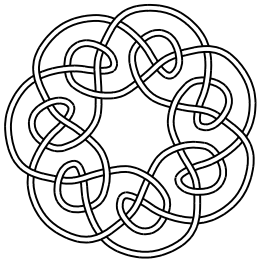
4.11 Other commands

construct name [par1 [par2 [par2 ... [parN]]]] — Invokes the constructor **KP-cons-name** with the given paramters. Examples of constructors, KP-cons-cable, KP-cons-symm, and others are in the demos directory.

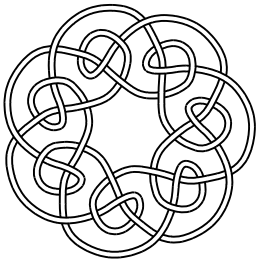
Fun — This command is not documented and never will be.

mf — Knots and (soon?) links can be exported as Metafont programs. This feature is under development, but is available for test purposes and examples can be found.





Section 5: Parameters



Parameters may be set by typing **parameter = value**. Like commands, parameters may be abbreviated to any length. However it is safer to never abbreviate a parameter to *less than four characters.* Parameters are guaranteed to be unambiguous if this general rule is followed.

5.1 Setting parameters

Parameters come in several varieties, real numbers, integers, and boolean numbers. Real numbers will be rounded off if an integer parameter is being set. Boolean parameters may be set to one of **true**, **on**, **yes**, or **1** to indicate true, or **false**, **off**, **no**, or **0** to indicate false. Some integer parameters are naturally expressed as hexadecimal values. An example is background. These parameters are also displayed in hexadecimal when using the parameters command. To set a parameter (or command argument) to a hexidecimal integer, preface the value with the characters **0x** (for example **back = 0xff7648** will set the background to a nice royal blue).

5.1.1 Random values

Parameters and all numerical arguments to commands may be set to values which are chosen randomly in several different ways. A random value is indicated by **\r** as in the following:

- **\rr**[a/b] A real number chosen uniformly in the range a to b (inclusive).
- **\rr**a A real number chosen uniformly in the range 0 to a (inclusive). For example, **\rr1** will choose a number from 0 to 1.
- **\rr+-**a A real number chosen uniformly in the range $-a$ to a (inclusive).
- **\ri**[a/b] An integer chosen uniformly in the range a to b (inclusive).
- **\ri**a An integer chosen uniformly in the range 0 to a (inclusive).

- **\ri+-**a An integer chosen uniformly in the range $-a$ to a (inclusive).

An example using all of the above is in the demos directory.

5.2 Changing the display

All the parameters in this section affect only the display as seen on the screen. They have absolutely no effect on the actual surface geometry.

auto-colour — If **auto-colour** is **true**, the parameter **hincr** is automatically set when loading a knot or creating a new one, the value of **hincr** depending on the number of components. Set **auto-colour** to **false** to turn off this action (default is true).

background — The background colour of the view window. Normally set by specifying a hex value (see the note above, default is 0x444444).

Drawing — Turns on or off drawing (default is on).

drecaps — Controls whether or not end-caps are drawn for open components. End-caps are not properly drawn in KnotPlot currently, set this parameter to **false** to turn off the drawing. This problem should be fixed in upcoming versions of the software, eliminating the need for this parameter (default is true).

grmode — Grid mode, 1 is a rectangular grid, 2 is a pie grid. Use the 3doptions command to change the colours used to draw the grid.

grsize — Size of the grid (default is 10).

grspace — Minor spacing between lines on the grid (default is 1). Set to 0 to turn off minor spacing.

grSpace — Major spacing between lines on the grid (default is 5). Set to 0 to turn off major spacing.

hstart — 'Hue start', the hue of component 0 in smooth tubes mode. Valid range is from 0 to 360 (default is 250).

hincr — 'Hue increment', the difference in hue between consecutive components in smooth tubes mode (default depends on number of components).

saturation — The colour saturation of the surface while in smooth tubes mode. Valid range is from 0 (no saturation, colours are grey) to 1 (maximum saturation, this is the default).

scale — The drawing scale (default is 0.35). Not to be confused with the scale command.

trx try trz — The offset in the three directions of the displayed knot. Using the right mouse button in the view window is the same as changing **trx** and **try**.

value — The 'value' or brightness of the surface while in smooth tubes mode. Valid range is from 0 to 1 (default is 1).

5.3 Surface geometry

All of the parameters in this section affect both the display seen on the screen and any surface description output to another program (DXF, SDL, etc.).

cyl-rad — The radius of the cylinders or smooth tubes when drawn (0.35). This parameter may also be set using the 'cyl-rad' slider on the main control panel.

nsegments — The number of sides of the drawn cylinders. Increase this for better rendering, decrease for faster rendering (default is 12). This parameter is changed by the luxo and cheapo commands. This parameter may also be set using the 'nseg' slider on the main control panel.

ncurve — The number of interpolations (segments) drawn between each bead using **mode s** (default is 3). This parameter is changed by the luxo and cheapo commands. This parameter may also be set using the 'ncur' slider on the main control panel.

broff — The size of the 'broken offset' when in drawing modes 'brok' or 'sbro' (default is 0.1).

5.4 Creating new knots and links

These parameters control various settings related to the creation of new knots and links.

D-chain — The radius of the entire link created with the chain command (default is 10).

d-chain — The radius of individual components in a link created with the chain command (default is 0.75).

d-torus — The smaller radius used in constructing a knot with the torus command (default is 2.75).

new — If **true**, everything is deleted when new knots or links are created with commands such as **chain** or others (see the section on creating new knots and links to see which commands are affected by this parameter). Set to **false** if you want to keep the old knot or link while adding a new one (default is **true**).

N-torus — Number of beads in knots constructed with the lissajous, torus or unknot commands.

R-torus — The larger radius used in constructing a knot with the torus or unknot commands (default is 8.25).

5.5 Relaxation of knots

These parameters affect the dynamical system run on the knot or link during the relaxation simulation.

bencon — Parameter controlling the magnitude of bending forces (default is 1.0).

bendforce — Sets bending forces on or off (default is off).

close — The closest distance that the cylinders are allowed to approach in the dynamical simulation (default is 0.12).

dstep — Normally when KnotPlot is relaxing a knot, the knot is displayed at each iteration step. On some machines, especially those with slow graphics but fast CPUs, it is often preferable to draw after several iterations. Setting this parameter to a value greater than one will cause KnotPlot to draw every **dstep** iterations during knot relaxation.

elecforce — Sets the 'electrical' force on or off (default is on).

iteration — Current iteration count (the number of relaxation steps).

max-dr — The maximum distance beads are allowed to move each relaxation step. In order for the relaxation to preserve knot type, this parameter should be less than **close** above (default is 0.1).

mechforce — Sets the 'mechanical force' on or off (default is on).

5.6 Four dimensional knots

These are the relevant parameters for four dimensional knot construction and display.

5.6.1 Constructing the knot

These parameters are used to define the surface constructed in the spinning or suspension process.

4dss — the number of spin sections created (the number of vertices along a line of latitude) (default is 30).

4dspangle — the angle of the spin plane (can also be set with the splane command).

4dw — suspended knots are constructed "hanging" from the points $(0, 0, 0, w)$ and $(0, 0, 0, -w)$ where w is the value of this parameter.

5.6.2 Displaying the knot

These parameters control the display of the knot (or related objects).

4deye — this sets the distance of the 4D eye point. Only relevant to the perspective projection.

4dw — this controls the length of the coördinate axes and the size of the bounding hyper-cube.

4dda — (**true** or **false**) display the four coördinate axes (**false**).

4ddc — (**true** or **false**) display a bounding hyper-cube (**false**).

4ddk — (**true** or **false**) display the knot itself (**true**).

4dsm — the way the knot surface is displayed. The possible values are

- **0** — draw as a shaded surface.
- **1** — draw as a wire frame coloured according to latitude on the sphere.
- **2** — same as 1 except draw latitude lines only.
- **3** — same as 1 except draw longitude lines only.
- **11** — draw as a wire frame but colour according to the value of the dimension lost in the projection from 4D to 3D.

Portions of the knot's surface can be selectively enabled or disabled using the spactivate command (for example, to view the northern hemisphere only).

5.7 PostScript pictures

There are several parameters controlling how PostScript pictures are generated. Also, provision is made in the PostScript file for easy editing if something unusual is required.

psbackface — If **true**, backfacing polygons are removed when outputting a PostScript file. In certain situations, such as when using a 'broken' drawing function, you might want to see backfacing polygons. If so, set this parameter to **false** (default is **true**).

psmode — Will affect the type of PostScript output generated. Defaults to 1. Modes 1 and 2 generally produce smaller files and print faster than mode 0. Modes in the "forties" produce the smallest files of all, and print much faster too. Possible values for **psmode** are:

- **0** — smoothly shaded surface (see Figure **??**).
- **1** — white polygons with black outline (see Figure **??**).
- **2** — black polygons with white outline (see Figure 6).
- **3** — same as mode 1 except the polygons are slightly rounded (see Figure 7).
- **4** — flat shaded (see Figure 8).
- **5** — similar to mode 0 except a pattern can be specified.
- **10** — output colour PostScript with colours similar to those seen with the drawing mode set to **draw norm**.
- **11** — same as mode 10 except use colours as in drawing mode **draw spec**.
- **30** — Allows a texture map in the form of a PGM[7] file to be specified.
- **40** — Simple rendering with a black curve (see Figure 9).
- **41** — Same as mode 40 except black and white are swapped. This gives a similar rendering to the knots and links found in Appendix C of *Knots and Links* (the links in the section headings are in this format, as well as the "infinity" symbol at the end of this manual)
- **42** — Intended to be an improvement over mode 40 but I'm not quite sure. All these modes need some minor improvements. Note that you'll have to make cyl-rad smaller with this mode to get similar results to using mode 40.
- **43** — The mode corresponding to mode 42 with black and white swapped.
- **104** — Maps a random Truchet pattern on the surface.

psN —

[7] **P**ortable **G**rey **M**ap

psM —

pssort — Hidden surfaces are handled in the PostScript rendering by using the 'Painter's algorithm', that is by drawing the polygons back to front. In order to do this, the set of polygons is normally sorted. Set this parameter to **false** to override this default behaviour.

5.8 Miscellaneous parameters

5.8.1 That twisting stuff

A demo script is available in the demos directory that illustrates the use of the following.

aftwist — Usually **off**, set to **on** if you want knots to be automatically twfixed when they are loaded. Note that **twfix**ing is done when using the buttons on the control panel are used to load something at random.

atw — A flag normally set to **off**, but some commands such as dxf and sdl will set it to **on** because they require that the coordinate frame of the smooth tube match up exactly.

twdefault — The twisting value used upon loading if **aftwist** is **on**.

twist — The amount of twist in milli-radians per segment (best see me for an explanation). Default is 0. Basically, this is a way to eliminate the deficit angle seen at the tie-point in the knot. To eliminate the deficit angle, first set **twist** and then enter the command twfix (or just give the value as an argument to **twfix**). Interesting effects can be produced by setting **nseg = 3** and using a **twist** of around 100.

5.8.2 Other parameters

silent — KnotPlot will not print anything if this parameter is **true** (default is **false**).

sformat — Format used to save knots or links with the save command. See the documentation on file formats.

- **0** — KnotPlot LOCF **float** format.
- **1** — KnotPlot LOCS **short** format. Files saved with this format are about half the size of those saved with the LOCF format. However, a small amount of noise is introduced to to the limited precision of short integers. Users should beware of accumulated errors if files are repeatedly loaded and saved with this save format in effect. Most of the knot and link files distributed with KnotPlot are stored in this format.

2 — KnotPlot LOCC **char** format. Files saved with the LOCC are only half the size the equivalent LOCS file and a quarter the size of an equivalent LOCF file. However, the same warning applies to the use of this file format as was given for the LOCS format, only more so. This format is very useful, however, for large animation databases.

3 — Raw ascii (same format as used by the coords command).

4 — Ascii format suitable for input into Kenny Hunt's KED program. Cannot be used for multi-component links.

5 — Similar to KED format except suitable for input into Ming. Cannot be used for multi-component links.

References

[AC59] J. J. Andrews and M. L. Curtis. Knotted 2-spheres in the 4-sphere. *Annals of Mathematics*, 70(3):565–571, November 1959.

[BHJS94] M. G. V. Bogle, J. E. Hearst, V. F. R. Jones, and L. Stoilov. Lissajous knots. *Journal of Knot Theory and Its Ramifications*, 3(2):121–140, 1994.

[CMS88] Michael Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-D rotation using 2-D control devices. In *Computer Graphics Proceedings (SIGGRAPH 88)*, Annual Conference Series, pages 121–129. ACM SIGGRAPH, 1988.

[Con70] J. H. Conway. An enumeration of knots and links, and some of their algebraic properties. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 329–358, 1970.

[DT83] C. H. Dowker and Morwen B. Thistlethwaite. Classification of knot projections. *Topology and its Applications*, 16:19–31, 1983.

[Han89] Vagn Lundsgaard Hansen. *Braids and Coverings*, volume 18 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 1989.

[HBK+94] Roland Haynes, Tom Berryhill, Jeff Koftinoff, Rob Scharein, and Malcolm Warwick. Multi-media jazz improvisation, November 1994. Centre for Image and Sound Research (CI⋆SR), Vancouver B.C.

[Kau87] Louis H. Kauffman. *On Knots*. Princeton University Press, 1987.

[Mil96] Kenneth C. Millett. Random walks in polygonal knot space. In *Topology and Geometry in Polymer Science*, University of Minnesota, Minneapolis, June 1996. Institute for Mathematics and its Applications.

[New96] William H. New. *Science Lessons*. Oolichan Books, Lantzville, British Columbia, 1996.

[Rol76] Dale Rolfsen. *Knots and Links*. Publish or Perish, Inc., 1976.

[Sim94] Jonathan K. Simon. Energy functions for polygonal knots. *Journal of Knot Theory and Its Ramifications*, 3(3):299–320, 1994.

[Sim96] Jonathan Simon. Energy and thickness of knots. In *Topology and Geometry in Polymer Science*, University of Minnesota, Minneapolis, June 1996. Institute for Mathematics and its Applications.

[SW90] Jonathan K. Simon and Keith Wolcott. Minimally knotted graphs in S^3. *Topology and its Applications*, 37(2):163–180, November 1990.

[Thi85] Morwen B. Thistlethwaite. Knot tabulations and related topics. In I. M. James and E. H. Kronheimer, editors, *Aspects of Topology in Memory of Hugh Dowker*, volume 93 of *London Mathematical Society Lecture Note Series*, pages 1–76. Cambridge University Press, 1985.