

Curves in L<sup>A</sup>T<sub>E</sub>X Pictures  
A Manual for **CURVES.STY** and **CURVESLS.STY**  
Version 1.41

Ian Maclaine-cross\*  
School of Mechanical and Manufacturing Engineering  
The University of New South Wales  
Sydney Australia 2052

22nd August 1995

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>4</b>
<b>3</b>	<b>Curves</b>	<b>5</b>
<b>4</b>	<b>Scaling</b>	<b>7</b>
<b>5</b>	<b>Symbols</b>	<b>8</b>
<b>6</b>	<b>Dashes</b>	<b>10</b>
<b>7</b>	<b>Errors</b>	<b>11</b>
<b>8</b>	<b>Pleas for the Future</b>	<b>12</b>
<b>9</b>	<b>Curves Summary</b>	<b>13</b>
9.1	Arguments of Commands . . . . .	13
9.2	Lengths used by Commands . . . . .	13
9.3	Control Commands . . . . .	14
9.4	Parameter Setting Commands . . . . .	14
9.5	Curve Drawing Commands . . . . .	15

---




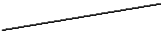

\*Current Internet address: [ian@ilm.mech.unsw.edu.au](mailto:ian@ilm.mech.unsw.edu.au)

## 1 Introduction

The picture environment in the  $\LaTeX$ <sup>1</sup> macro package for  $\TeX$ <sup>2</sup> allows simple line drawing using characters. These characters include quadrant circular arcs, solid disks with diameters from 1 to 15pt<sup>3</sup> and short lines with a limited range of slopes in two thicknesses. A `\begin{picture}` command defines an area where following commands place these characters to draw a  $\LaTeX$  picture.

$\LaTeX$  pictures save disk space for source descriptions and computer time in producing documents compared with printer commands or bit mapped graphics. From initial pencil sketch on squared graph paper to final printout, they take half the time for manual pen drawings. The labor savings are higher for revisions and rewrites. Unfortunately standard  $\LaTeX$  cannot yet draw curves like a pen, compass and French curves can.

Superimposing characters closely can draw any curve. Disks give directional independence of line thickness and visual smoothness at large pitch. Smoothness increases with output resolution but is almost independent of disk pitch below a critical maximum. The following table suggests this maximum varies from 0.34pt for an 0.5pt thick curve to 3pt for a 15pt curve.

Thickness	Magnified	Comments	Line at 1/6 slope
15pt	×1	1pt pitch disks	
15pt	×1	3pt pitch disks	
≈0.5pt	×1	$\LaTeX$ <code>\line(6,1)</code>	
≈0.5pt	×1	0.34pt pitch disks	
≈0.5pt	×30	0.34pt pitch disks resolution ×30	

$\LaTeX$  `\line`<sup>1</sup> or line drawing `\special` commands require a tenth the  $\TeX$  memory so are preferable to disks if available. A smooth curve may require thousands of disks or hundreds of lines.  $\LaTeX$  can load  $\TeX$ <sup>4</sup> macros from style files with a `.sty` extension. These macros can calculate the disk or line positions for a curve.

`curves.sty` provides macro commands for drawing curves using disks. Version 1.41 loads in about 2500 words<sup>5</sup> of  $\TeX$ 's main memory, which allows a small  $\TeX$ . Complex or numerous floating drawings still require a big  $\TeX$  with

<sup>1</sup>Leslie Lamport,  *$\LaTeX$  A Document Preparation System 2nd ed.*, Addison-Wesley, 1994.

<sup>2</sup>Donald E. Knuth, *The  $\TeX$ book*, Addison-Wesley, 1984.

<sup>3</sup>A printer's point, abbreviated pt, is approximately 0.351460 mm.

<sup>4</sup>Donald E. Knuth, *The  $\TeX$ book*, Addison-Wesley, 1984, Chapter 20.

<sup>5</sup>A character or token takes a  $\TeX$  word, 32 bits or larger.

`curves.sty`. `curvesls.sty` uses fewer disks and line drawing `\specials` for increased speed and a tenth the `TeX` memory.

The capabilities of these macros are:—

- A compatible replacement for `\bezier`<sup>6</sup> from `bezier.sty` or `LaTeX2e`.
- Work with `SLiTeX` for overhead transparencies and `LaTeX2e`<sup>7</sup>
- Curves have the minimum number of disks or straight lines for visual smoothness.
- Curve thickness adjustable from 0.5 to 15pt (0.17 to 5mm).
- Curves have continuous slope.
- Curves for any number of points greater than one using `\curve`.
- Control of end slopes using `\tagcurve`.
- Closed curves with continuous slope using `\closecurve`.
- Large circles `\bigcircle` and circular arcs `\arc`.
- Independent scaling of curve abscissa and ordinates to fit graphs.
- Affine scaling for making arcs or circles elliptical.
- Symbols and dash patterns combined without interference.
- Any dash length or spacing.
- Three methods for fitting dash patterns to curves.

Parabolas approximate the segments between the specified coordinate points. At an internal point, the slope of the parabolas joining from either side is the same as the straight line joining the points on either side. For an end segment, the inside point of the parabola is made a vertex which determines the slope of the end point. `\tagcurve` has hidden end segments which allows complete control of slope at the visible end point when desired.

The four points at the end and adjacent to a segment may make a parabola inappropriate. The points could be close to a straight line or consistent with an inflexion point. A straight line then replaces the parabola and optionally a warning is issued.

The parabolas are drawn as a series of short straight line segments. For `curvesls.sty`, the dvi driver draws the straight lines as instructed by `\special` commands. For `curves.sty`, the straight lines are drawn as overlapping disks

---

<sup>6</sup>See your Local Guide. Try the system command `latex local` to get a `LaTeXed` Guide.

<sup>7</sup>Michel Goossens, Frank Mittelbach and Alexander Samarin, *The LaTeX Companion*, Addison-Wesley, 1994.

at high speed using a simple tail chasing macro. This macro's arithmetic calculations are two fixed point additions per disk drawn.

The  $\TeX$  arithmetic in `curves.sty` for the initial differences between disk positions gives an error of less than 0.005% at the segment end. Even for a segment 1000pt long, this is less than 0.05pt and the resolution of most output devices. Errors are sometimes slightly larger between coordinates but you can reduce them if necessary by specifying more coordinates.

For circles and circular arcs, a parabola approximates an arc segment no more than  $23^\circ$  giving a radius increase between segment ends less than 0.02%. The error in computing and multiplying by sine and cosine is usually less than 0.01% of the radius at the far end of an arc.

Brief descriptions, simple examples and a command summary follow. They presume familiarity with relevant chapters of the  $\LaTeX$  manual<sup>1</sup>.

## 2 Installation

Name either `curves` or `curvesls` in the `\usepackage` command at the top of your main `.tex` file *e.g.*,

```
\documentclass[11pt]{article}
\usepackage{curves}
```

`curves.sty` or `curvesls.sty` should be in the default or a `texinput` directory. For the older  $\LaTeX$  2.09 name them in the `\documentstyle` command.

Do not combine `curves` or `curvesls` with `bezier` in this command. They contain a fast powerful replacement for `\bezier`. Drawings using `\bezier` should not change their appearance.

$\Pi\TeX$ <sup>8</sup> is a large set of  $\TeX$ ncial drawing macros for plain  $\TeX$  which  $\LaTeX$  pictures can also use.  $\Pi\TeX$  requires a big  $\TeX$ . I have not checked compatibility with `curves` or `curvesls` because I do not use  $\Pi\TeX$ .

`curvesls.sty` uses line drawing `\specials` to reduce required memory and increase speed. As originally supplied it uses the `\specials` of  $\em\TeX$  by Eberhard Mattes for IBM PC compatibles.  $\em\TeX$  is available from CTAN<sup>9</sup> and the  $\TeX$  Users' Group. The  $\em\TeX$  `\specials` are also available for Tomas Rokicki's `dvips`. If you use `dvips`, ask if they have been compiled in. To use other line drawing `\specials` substitute them at `curvesls.sty`'s start. Your system guru may do this for you. You can test `curvesls.sty` after substituting `\specials`, by replacing `curves` with `curvesls` in this file's `\documentstyle` command. This file looks the same  $\LaTeX$ ed with either `curves` or `curvesls`.

<sup>8</sup>Michael J. Wichura, *The  $\Pi\TeX$  Manual*, Chicago, 1987, available from  $\TeX$  Users' Group.

<sup>9</sup>Directory `/tex-archive/systems/msdos/emtex` at `ftp.shsu.edu` for example.

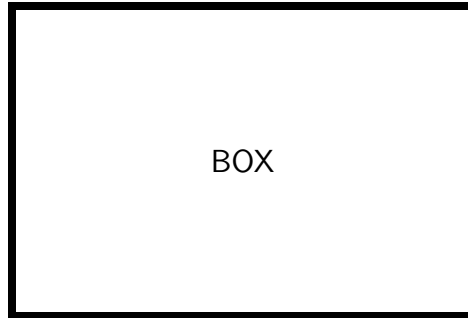


Figure 1: This is a box.

A drawing frequently uses auxiliary commands to size, place, label and caption it. The following commands draw the box in Figure 1 on page 5:

```

\begin{figure}
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(100,50) \large\sf
\linethickness{1mm}
\put(20,5){\framebox(60,40){BOX}}
\end{picture}
\end{center}
\caption{This is a box.}
\label{box}
\end{figure}


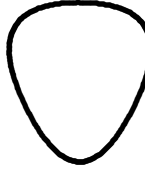

```

Lamport<sup>1</sup> explains these commands. This example is for those unfamiliar with the  $\LaTeX$  `picture` environment. The following examples avoid the `figure` environment but it is often essential.

### 3 Curves

The curve drawing commands, `\curve`, `\closecurve` and `\tagcurve`, draw parabola segments between coordinate points in the argument. The segments' tangents at these points are parallel to each other and to straight lines through the points either side. Segments at curve ends are from parabolas through the

last three points. `\closecurve` adds a parabola between end points to close the curve. `\tagcurve` omits the first and last segments drawing curves with end tangents specified. The following table shows these features.

Example	Curve
<code>\curve(0,0, 50,100, 100,0)</code>	
<code>\closecurve(0,0, 50,100, 100,0)</code>	
<code>\tagcurve(100,0, 0,0, 50,100, 100,0, 0,0)</code>	

Axial flow fans often use the RAF 6E aerofoil section. The section coordinates in the following macro come directly from aerodynamic tables<sup>10</sup>. The `\arc` commands draw the leading and trailing radii and the two coordinate `\curve` the flat chord.

```
\newcommand{\RAFsixE}{
  \scaleput(1.25,1.25){\arc(0,-1.25){-135}}
  \scaleput(0,0){\curve(0.366,2.133, 1.25,3.19, 2.5,4.42,
    5.0,6.10, 7.5,7.24, 10,8.09, 15,9.28, 20,9.90, 30,10.3,
    40,10.22, 50,9.80, 60,8.98, 70,7.70, 80,5.91, 90,3.79,
    95,2.58, 99.24,1.52)}
  \scaleput(99.24,0.76){\arc(0,-0.76){180}}
  \scaleput(0,0){\curve(1.25,0, 99.24,0)}
}
```

In a picture environment like:

```
\begin{picture}(100,20)
  \RAFsixE
\end{picture}
```

this macro draws:

<sup>10</sup>R.A. Wallis, *Axial Flow Fans*, Academic Press, 1961, p.335



The RAF 6E has a flat undersurface.

The drawing command `\bigcircle` works similarly to `\circle` except there is no `\circle*` equivalent. The following section scales it to an ellipse.

## 4 Scaling

The size of  $\text{\LaTeX}$  picture objects may be uniformly scaled by preceding them with:

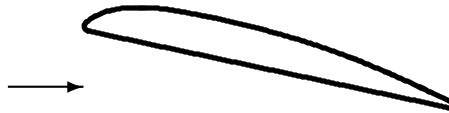
```
\setlength{\unitlength}{x\unitlength}
```

where  $x$  the desired scale factor is a decimal number.

The `\scaleputs` in `\RAFsixE` position the curves using the scale factors `\xscale`, `\xscaley`, `\yscale` and `\yscalex`. The drawing commands use the same scale factors. `curve` initially sets these scale factors to 1, 0, 1 and 0 respectively. `\renewcommand` may reset them to any decimal number. These factors can rotate pictures. The factors following rotate the RAF 6E through  $12^\circ$  clockwise about its (0,0) co-ordinate:

```
\renewcommand{\xscale}{0.9781}
\renewcommand{\xscaley}{0.2079}
\renewcommand{\yscale}{0.9781}
\renewcommand{\yscalex}{-0.2079}
\put(0,20){\RAFsixE}
```

This draws:



The RAF 6E has maximum lift at angles of attack over  $12^\circ$ .

Note that  $\cos 12^\circ \approx 0.9781$  and  $\sin 12^\circ \approx 0.2079$

Axonometric projection is another scaling application. Circles become ellipses and circular arcs become elliptical arcs. The commands drawing the ellipse and arc in the following washer are:

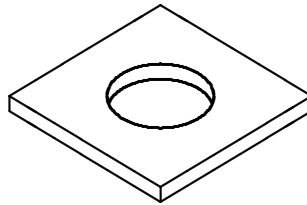
```
\put(20,5){
\renewcommand{\xscale}{1}
\renewcommand{\xscaley}{-1}
```

```

\renewcommand{\yscale}{0.6}
\renewcommand{\yscalex}{0.6}
\scaleput(10,10){\bigcircle{10}}
\put(0,-2){
  \scaleput(10,10){\arc(5,0){121}}
  \scaleput(10,10){\arc(5,0){-31}}
}
}

```

(20,5) are the drawing coordinates of the upper vertex of the washer closest to the reader. The angles for the `\arcs` were found by trial and error.



Square washers are sometimes preferred for soft materials.

## 5 Symbols

`curves` can also place symbols. `\curvesymbol` must first define the symbol as anything a `\put` or `\multiput` may draw. A negative symbol count between drawing command and coordinates *e.g.*, `\tagcurve[-3](0,100,...)` fixes the number of symbols per curve segment.

These commands draw flight times and successive positions in the following drawing:

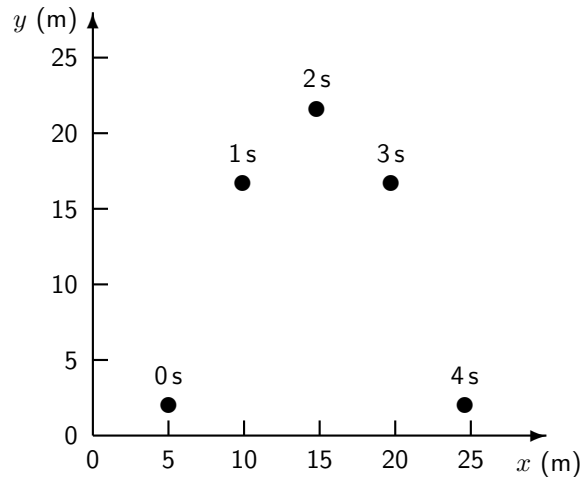
```

\newcounter{time}
\curvesymbol{\thetime\s\addtocounter{time}{1}}
\put(5,4){\curve[-2](0,0, 9.8,19.6, 19.6,0)}
\curvesymbol{\phantom{\circle*{1}}\circle*{1}}
\put(5,2){\curve[-2](0,0, 9.8,19.6, 19.6,0)}

```

where `\phantom` is a plain `TEX` command from the `TEXbook`<sup>2</sup>.





Successive positions of a sphere with initial position (5, 2) m, initial velocity (4.9, 9.8) m/s, and acceleration (0, -9.8) m/s<sup>2</sup>. The flight time is recorded above each sphere position.

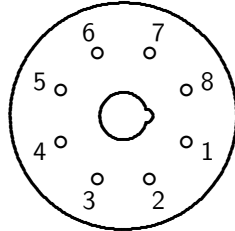
Fixed spacing of symbols at lengths other than the segment's requires more commands. Empty `\curvedashes`, empty `\curvesymbol` and negative symbol count stops drawing so a drawing command will calculate `\curvelength` only. `\curvesymbol` then resets the symbol and `\curvedashes` sets the spacing to its pattern length. If there are no symbols at the ends, `\overhang` pulls symbols along the curve. The last command with no symbol count draws the symbols.

`\arc` and `\bigcircle` use sixteen segments for a circle so if eight symbols are required the fixed spacing technique is necessary. The following commands draw the pin numbers on a relay base:

```

\newcounter{pin}
\curvedashes{}
\curvesymbol{}
\put(60,60){\arc[-1](40,0){-360}}
\setlength{\curvelength}{0.125\curvelength}
\curvedashes[\curvelength]{1}
\setlength{\overhang}{0.5\curvelength}
\curvesymbol{\addtocounter{pin}{1}\thepin}
\put(60,60){\arc(40,0){-360}}

```



The pin numbering of plug-in relays is clockwise from the spigot key when viewed from below.

If symbols and dash pattern exist and `\overhang` is 0pt, `curves` draw the first position blank. For equal spacing they draw the last position blank if rounding error causes the last pattern to be slightly short. If `\renewcommand` changes `\diskpitchstretch`, rounding error changes and the final symbol may reappear. To avoid fiddling with `\diskpitchstretch` for closed curves with symbols equally spaced, use an `\overhang` which is a fraction of a pattern length as in the previous example.

## 6 Dashes

`\curvedashes` must first define a dash pattern with length greater than 0pt. Many symbol and pattern combinations are possible. The fixed number and fixed spacing methods of symbol drawing described in Section 5 work with three methods of drawing dashes which are:

1. if there is no symbol count and no symbol, a dash pattern with its length reduced by `\csdiameter` is drawn between symbols spaces of width close to `\csdiameter` to give an overall spacing equal to the pattern length specified by the `\curvedashes` command;
2. if there is a symbol count but no symbol, the dash patterns drawn have their length equal to that defined by `\curvedashes` with `\csdiameter` gaps at symbol positions;
3. if there is a symbol count and a symbol, the dash patterns drawn have their length adjusted slightly so an integral number of patterns fit between symbol positions.

Dash pattern commands for centrelines<sup>11</sup> follow for the three techniques above in order:

---

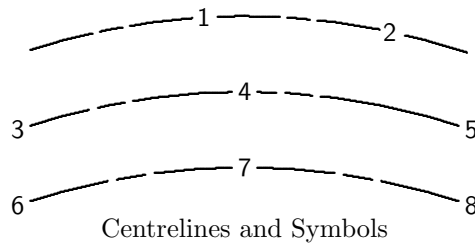
<sup>11</sup>R.N. Roth and I.A. van Haeringen, *The Australian Engineering Drawing Handbook, Part 1 Basic Principles and Techniques*, The Institution of Engineers, Australia, Canberra, 1986.

```

\linethickness{0.25mm}
\curvedashes[1.2mm]{0,8,1,3,1,8}
\settowidth{\csdiameter}{00}
\put(0,20){\curve(0,0, 30,5, 60,0)}
\put(0,10){\curve[1](0,0, 30,5, 60,0)}
\curvesymbol{\thepin\addtocounter{pin}{1}}
\setlength{\csdiameter}{2\csdiameter}
\put(0,0){\curve[1](0,0, 30,5, 60,0)}

```

The following figure shows the resulting dash patterns. The upper line has first position blank because the `\overhang` is 0pt. It has patterns shrunk to scale between symbol spaces *e.g.*, 1 to 2, and symbol space centres one pattern length apart. The middle line has patterns close to defined length but with the first dash part blanked by half of symbol space 3 and the second pattern broken in its first dash by symbol space 4. The lower line patterns are stretched between symbol spaces. Which pattern is appropriate depends on picture meaning and function.



## 7 Errors

Syntax errors like incorrect or missing punctuation while using `curves` will result in `TEX` or `LATEX` error messages. The `TEXbook`<sup>2</sup> and `LATEX manual`<sup>1</sup> explain the meaning and correction of these errors. The previous examples and Section 9 should make the correct syntax for `curves` commands clear.

`curves` will write a `LaTeX error.` message to the screen and log file if you supply an incorrect number of coordinates.

If four sequential points in a drawing command argument have the line through the first and third parallel to the line through the second and fourth:

- exactly or closely, `curves` knows it cannot draw a parabola tangent to two parallel lines, issues to the screen and log file:  
`LaTeX Warning: \curve straight from ...`  
and draws a straight line;
- or approximately, `curves` may draw an unexpected curve with no warning.

If four sequential points in a drawing command argument have the line through the first and second parallel to the line through the third and fourth:

- `curves` draws a parabola which may be nowhere near the curve.

If the four points were on a straight line, removing one or more points is a remedy. If they are not on a straight line, adding points may help. Specifying many points will give you a satisfactory curve with perhaps an annoying number of `\curve straight` warnings. After a `\curvewarnfalse`, `curves` still uses the straight lines but does not tell you.

Curvature changes sign on curves like  $y = \sin x$ . Specifying inflection points as `curve` coordinates will reduce error and specifying sufficient coordinates will then give satisfactory results. For discontinuous tangents splitting a curve into pieces is unavoidable. Splitting a curve into pieces with curvature the same sign can give satisfactory results with fewer coordinates. `\tagcurve` can prevent tangent discontinuities. If an inflexion point's exact location is unknown, try the midpoint of the straight line through the ends of its segment.

Curves appear rougher than horizontal and vertical lines. Picture digitization causes this not inaccuracy in  $\TeX$ , `curves.sty` or `curvesls.sty`. Setting `\diskpitchstretch` to a value less than one with `\renewcommand` may smooth an unusually rough curve.

Symbols and symbol spaces misaligned are usually due to rounding error. Adjusting `\diskpitchstretch` below one can reduce rounding error and increase alignment accuracy. This should be limited to the misaligned curve with `{ }`<sup>1</sup>.

The replacement `\bezier` does not give exactly the same results as the original in `bezier.sty` or in  $\LaTeX 2\epsilon$ . The difference is extremely small but if it is important to you comment out the five lines of code for `\bezier` and `\@bezier` near the start of `curves.sty` or `curvesls.sty`. You now have a `\bezier` which is slower and needs more memory but has only its original capabilities and gives only its original results.

Please email me examples of any errors not covered above. You may have found a bug in the code or documentation.

## 8 Pleas for the Future

`curves` will never work with plain  $\TeX$  and it will never be as powerful as `META-FONT` or importing encapsulated PostScript files. Suggestions or criticisms by email are welcome. Version 1.41 has benefitted greatly from previous help. The latest versions are first available at URL:

`ftp://ilm.mech.unsw.edu.au/pub/latex/macros/curves/`

The IP number is 149.171.228.99 but use your local CTAN mirror to reduce Internet load.

`curvesls.sty` uses much less TeX memory and is faster than `curves.sty`. More people could use `curvesls.sty` if emTeX or equivalent line drawing \specials were compiled into dvi drivers for viewing and printing. Ask your system people for this!

A .dvi file containing curves produced with `curvesls.sty` has many occurrences of the text strings `em:lineto`, `em:moveto` and `em:linewidth XXXXpt` placed by the emTeX \specials. These strings would have earlier occupied TeX memory. Extra space is also taken by the disks `curvesls` uses to cover the cracks between square line ends at a slight angle. Renaming these \specials to L, M and W XXXXpt respectively would save TeX memory. Even better, W XXXXpt could also calculate and store the bitmap of a disk which L would add to its line end to round it so `curvesls` need not add disks to cover cracks. A fourth \special, D, could place the disk at the current position which `curvesls` would use to round the beginning of a curve. If you write dvi drivers, please add these four proposed \specials.

## 9 Curves Summary

The commands following are for the picture environment in the L<sup>A</sup>T<sub>E</sub>X manual<sup>1</sup>.

### 9.1 Arguments of Commands

- <blank length> decimal number of <unit len> blank. Not negative.
- <character or symbol> is anything which a \put or \multiput may draw.
- <coordinates> are decimal numbers giving alternate *x* and *y* coordinates of the curve as multiples of \unitlength, comma separated.
- <[,dash...]> optional continuation of alternating dash and blank numbers of unit lengths, comma separated. Not negative. Allows decimal points.
- <diameter> is a decimal number giving the diameter in \unitlengths.
- <symbol count> is the number of symbols or patterns to be drawn, default 0.
- <unit len> unit length dimension *e.g.*, 2.5mm, 10pt, used in measuring blanks and dashes. Not negative. Default value is \unitlength.

### 9.2 Lengths used by Commands

- \csdiameter is the size of the space left for a symbol and can be increased or set with `\settowidth{\csdiameter}{<character or symbol>}`.
- \curvelength is the total length of the curve calculated before drawing by using Simpson's rule once between each pair of coordinate points.

`\overhang` length of as drawn dash pattern overlapping start of patterns.

### 9.3 Control Commands

`\curvewarntrue` turns warning of parabola replacement by straight lines on (default).

`\curvewarnfalse` turns warning of parabola replacement by straight lines off.

### 9.4 Parameter Setting Commands

`\curvesymbol{<character or symbol>}` sets `symbol` and `\csdiameter`.

`\curvedashes[<unit len>]{<blank length><[,dash...]>}` A drawing command before a `\curvedashes` or after one with an empty or zero length pattern will draw:

- if `<symbol count>` is zero or missing, a continuous curve;
- else if `<symbol count>` is positive, `<symbol count>-1` squares of line thickness size between and additional squares at coordinates or bezier end points;
- else if no `<character or symbol>` exists, nothing;
- else, `-<symbol count>-1` characters or symbols between coordinates and additional ones at coordinates or bezier end points.

After a `\curvedashes` command defining a pattern whose length exceeds zero, commands draw:

- if `<symbol count>` is zero or missing then at a spacing equal to the specified pattern length,
  - if no `<character or symbol>` exists, a dash pattern reduced in length by `\csdiameter` to fit between symbol spaces of `\csdiameter`,
  - else if `\overhang` is not 0pt, a `<character or symbol>` at all positions, else a `<character or symbol>` with the first position blank;
- else, `\csdiameter` wide symbol spaces, one at and `|\<symbol count>|-1` between coordinate points with dash pattern lengths,
  - if no `<character or symbol>` exists, exact but broken by the spaces,
  - else, adjusted to give a whole number of patterns between spaces.

`\diskpitchstretch` is initially 1 but `\renewcommand` can change it to a higher value like 5 to save memory in drafts of complex documents or a lower local value like 0.5 to smooth curve digitization.

`\linethickness{<len>}` sets line or dash thicknesses to `<len>` from 0.5pt up to 15pt (0.17mm to 5mm). `\thicklines` and `\thinlines` also set thickness.

`\xscale`, `\xscaley`, `\yscale` and `\yscalex` are scale factors initially set to 1, 0, 1 and 0 respectively which `\renewcommand` or `\def` can reset.

## 9.5 Curve Drawing Commands

Curves drawn consist of parabolic arcs between coordinate points with tangents at each point parallel to the straight line through adjacent points.

`\arc[<symbol count>](X1,Y1){<angle>}` draws a circular arc centred on current position, starting from (X1,Y1) and proceeding counterclockwise for <angle> degrees.

`\bezier{<symbol count>}(X1,Y1)(X2,Y2)(X3,Y3)` draws a curve through the end points (X1,Y1) and (X3,Y3) tangent to the straight lines joining each of them to (X2,Y2). Extended faster replacement for `bezier.sty` version.

`\bigcircle[<symbol count>]{<diameter>}` draws a circle of diameter equal to <diameter> times `\unitlength`.

`\closecurve[<symbol count>](<coordinates>)` draws a closed curve with continuous tangents at all points. At least 6 coordinates required.

`\curve[<symbol count>](<coordinates>)` draws a curve through the specified <coordinates>. For 4 coordinates this is a straight line.

`\scaleput(X1,Y1){<picture object>}` places a picture object in a position scaled by `\xscale`, `\xscaley`, `\yscale` and `\yscalex` for axonometric projection or rotations.

`\tagcurve[<symbol count>](<coordinates>)` draws a curve without its first and last segments but if only 6 coordinates draws the last segment only.