

The xypdf package

Daniel Müllner

1.4, dated 2010/05/13

Abstract

The `xypdf` package improves the output quality of the `Xy-pic` package when PDF documents are generated. It produces generic PDF code for graphical elements like lines, curves and circles instead of approximating these elements with glyphs in special fonts as the original `Xy-pic` package does. The `xypdf` package works with both $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ in the occurrences of $\mathrm{pdfT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{X}_{\mathrm{Y}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\varepsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ with `dvipdfm(x)` to generate PDF files. `xypdf` is being integrated and distributed together with `Xy-pic`, starting with `Xy-pic` version 3.8.

1 Introduction

The `Xy-pic` package is a utility for typesetting diagrams in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ documents. The authors of the `Xy-pic` package put much effort into the feature that most graphical elements are coded within the limited possibilities of the device independent file format (DVI). The diagrams can thus be generated with even the most basic $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ systems and displayed universally by all device drivers. For example, diagonal lines are composed of short dashes, which are glyphs in a special font. Since there are dashes in 127 discrete directions in the font `xydash10`, diagonal lines which do not match one of these slopes look slightly rugged when they are magnified.

For a better output quality in Postscript files, the authors of the `Xy-pic` package provided a Postscript backend for DVI-to-Postscript drivers. These extensions draw lines and curves by generic Postscript commands, thus trading a much better output quality against universality of the produced DVI files.

The development of the present package was based on `Xy-pic` version 3.7, from 1999, where there is no support for $\mathrm{pdfT}_{\mathrm{E}}\mathrm{X}$. In order to produce PDF files with high-quality `Xy-pic` diagrams, users had to use so far the Postscript file format as an intermediate step or embed the diagrams as external graphics. However, since many users directly generate PDF files from the $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ or DVI files (with bookmarks, hyperlinks and other PDF features), it is highly desirable to also have the possibility of directly generating `Xy-pic` diagrams with high-quality PDF graphics elements.

The present package `xypdf` adapts the output routines of the `Xy-pic` package to generate high-quality graphics for PDF output. It works with both $\mathrm{pdfT}_{\mathrm{E}}\mathrm{X}$ and the two-step compilation $\mathrm{T}_{\mathrm{E}}\mathrm{X} \rightarrow \mathrm{dvipdfm(x)}$ with an intermediate DVI file. Thus, it also works with $\mathrm{X}_{\mathrm{Y}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ since this program internally uses a modification of `dvipdfmx`. Note that some version of $\varepsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ is needed (which is anyway used by default in modern $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ installations). [Figure 1](#) compares the output quality of a small `Xy-pic` diagram.

The `xypdf` package is very similar to the Postscript backend to `Xy-pic`. It does not have (yet) all features of the Postscript backend (see [Section 4](#)) but is much more powerful in other respects, e.g. when drawing multiple curves. In general, it greatly improves graphics quality in most circumstances and otherwise leaves graphics elements as they are. Currently, the following features are implemented:

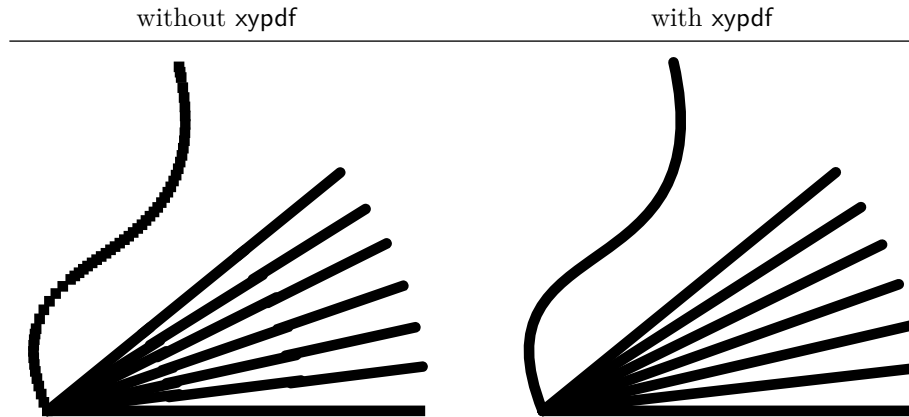
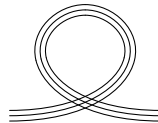


Figure 1: Comparison of $\text{\texttt{X}\texttt{Y-pic}}$ output, magnified 10 times.

- Both straight lines and curves (solid, dashed, dotted and squiggled) are drawn by generic PDF commands.
- $\text{\texttt{X}\texttt{Y-pic}}$ automatically draws the symbols of which lines and curves are composed at the very beginning and end of a segment. It then distributes the inner symbols evenly across the segment. Since the arc length of a Bézier curve is normally not proportional to its parameter, this is a nontrivial task in the case of curves. The $\text{\texttt{xypdf}}$ package handles this better than the original code. Compare the output in [Figure 2](#).
- As a highlight, $\text{\texttt{xypdf}}$ features a Bézier curve offset algorithm, producing high-quality curves with two or three parallel strokes.










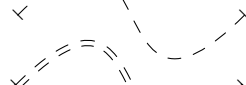

















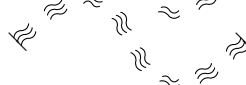
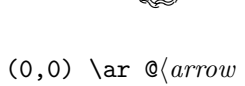
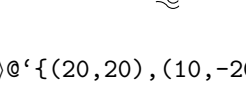

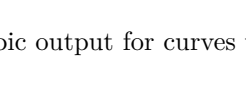
- The $\text{\texttt{\textbackslash cir}}$ object draws circles of arbitrary radius.

without xypdf	with xypdf	code
		$\text{\texttt{\textbackslash xy *}\text{\texttt{\textbackslash cir<16pt>\{}}}\text{\texttt{\textbackslash cir<19pt>\{}}}\text{\texttt{\textbackslash endxy}}$

- $\text{\texttt{xypdf}}$ supports the “rotate” extension of $\text{\texttt{X}\texttt{Y-pic}}$.

with xypdf	code
Test text	$\text{\texttt{\textbackslash xy *}\text{\texttt{[@!15]}\text{\texttt{\textbackslash hbox\{Test text\}}}\text{\texttt{\textbackslash endxy}}$
Test text	$\text{\texttt{\textbackslash xy *}\text{\texttt{[*1.5]}\text{\texttt{\textbackslash hbox\{Test text\}}}\text{\texttt{\textbackslash endxy}}$

- $\text{\texttt{xypdf}}$ supports the “frame” extension of $\text{\texttt{X}\texttt{Y-pic}}$. The only intended difference to the Postscript drivers is the appearance of the $\text{\texttt{\{**\}}}$ fill style. The postscript drivers fill the frame and stroke it with a line of thickness 0. The PDF driver stroke the region instead with a black line which is half as thick as the normal lines. Tip: If you want a colored boundary, overlay two frames.

without xypdf	with xypdf	$\langle arrow\ style \rangle$
		$\{ - \}$
		$2\{ - \}$
		$3\{ - \}$
		$\{ -- \}$
		$2\{ -- \}$
		$3\{ -- \}$
		$\{ \cdot \}$
		$2\{ \cdot \}$
		$3\{ \cdot \}$
		$\{ \sim \}$
		$2\{ \sim \}$
		$3\{ \sim \}$
		$\{ \sim\sim \}$
		$2\{ \sim\sim \}$
		$3\{ \sim\sim \}$

Code: `\xy (0,0) \ar @{\langle arrow\ style \rangle} @' {(20,20),(10,-20)} (30,0) \endxy`

Figure 2: Comparison of X_Y-pic output for curves with various line styles.

with xypdf code

Test text	<code>\xy ***[F**::lightgray]\hbox{Test text} \endxy</code>
Test text	<code>\xy ***[F-:red][F*:lightgray]\hbox{Test text} \endxy</code>

- xypdf supports the “color” extension of X_Y-pic. As described in the X_Y-pic Reference Manual, colors can be defined by the `\newcycolor` command, e. g.

```
\newcycolor{mygreen}{.5 0 1 .5 cmyk}
```

In addition, if the command `\color` is defined, e. g. if the `color`¹ or `xcolor`² package has been loaded, xypdf recognizes the named colors from these packages and uses the mechanisms from these packages to set colors in the output DVI or PDF file.

An example:

Orange Green

This was generated by `\usepackage{xcolor}` in the document preamble and the following code:

```
\definecolor{mygreen}{cmyk}{.5 0 1 .5}
\xymatrix{*[orange][F-:blue]\hbox{Orange}
& *[mygreen]\hbox{Green}}
```

When a named color has been defined by both `\newcycolor` and by a `color` package command like `\definecolor`, the X_Y-pic definition overrides the general one.

The X_Y-pic command `\UseCrayolaColors` defines a set of color names, as explained in the X_Y-pic Reference Manual. [Figure 3](#) lists these colors.

If you notice any unwanted behavior, please generate a minimal example and e-mail it to the author of this package. Current contact details are available at <http://www.math.uni-bonn.de/people/muellner>. Please report situations where the algorithms produce arithmetic overflows. Also, the code is not really optimized for speed but for accuracy, so feel free to report a significant slowdown of the compiling process for your thesis/paper/book.

2 Usage

Use `pdf` as an option to the X_Y-pic package, as in

```
\usepackage[pdf]{xy}
```

or

```
\usepackage{xy}
\xyoption{pdf}
```

for L^AT_EX and

```
\input xy
\xyoption{pdf}
```

¹<http://www.ctan.org/tex-archive/macros/latex/required/graphics/>

²<http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/>

GreenYellow:		Yellow:		Goldenrod:	
Dandelion:		Apricot:		Peach:	
Melon:		YellowOrange:		Orange:	
BurntOrange:		Bittersweet:		RedOrange:	
Mahogany:		Maroon:		BrickRed:	
Red:		OrangeRed:		RubineRed:	
WildStrawberry:		Salmon:		CarnationPink:	
Magenta:		VioletRed:		Rhodamine:	
Mulberry:		RedViolet:		Fuchsia:	
Lavender:		Thistle:		Orchid:	
DarkOrchid:		Purple:		Plum:	
Violet:		RoyalPurple:		BlueViolet:	
Periwinkle:		CadetBlue:		CornflowerBlue:	
MidnightBlue:		NavyBlue:		RoyalBlue:	
Blue:		Cerulean:		Cyan:	
ProcessBlue:		SkyBlue:		Turquoise:	
TealBlue:		Aquamarine:		BlueGreen:	
Emerald:		JungleGreen:		SeaGreen:	
Green:		ForestGreen:		PineGreen:	
LimeGreen:		YellowGreen:		SpringGreen:	
OliveGreen:		RawSienna:		Sepia:	
Brown:		Tan:		Gray:	
Black:		White:			

Figure 3: Additional color names provided by `\UseCrayolaColors`.

for plain $\text{T}_\text{E}\text{X}$. Do not use one of the other driver options to X_Y -pic like `dvips`, since combining two drivers will most likely result in mutilated diagrams.

The `xypdf` functionality can be switched off and on within the document by `\xypdfoff` and `\xypdfon`.

When you use plain $\text{T}_\text{E}\text{X}$, make sure that `xypdf` is called after any global changes to the math fonts.

3 Acknowledgements

Since the `xypdf` package extends X_Y -pic, some ideas are adopted from this package and its Postscript backend, and the author gratefully acknowledges the service which Kristoffer H. Rose and Ross Moore did to the mathematical community with their original package.

4 To do

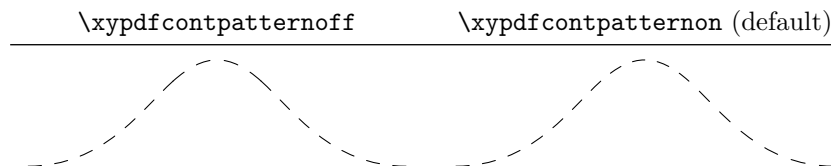
- Support for the “line” extension.

5 The fine print: curves with multiple segments

Since the dashes in Bézier segments are aligned to the boundary points, this would result in dashes of double length when a curve is composed of several Bézier segments, as shown in the upper left diagram. To avoid this, `xypdf` records the end point of each segment and adapts the dash pattern whenever the starting point of a segment coincides with the end

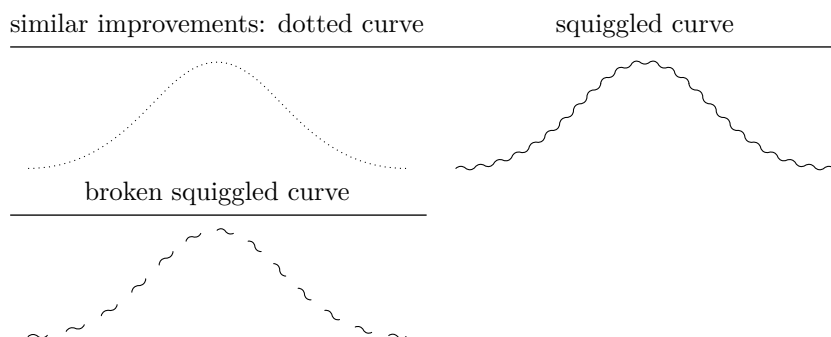
point of the previous one (see the upper right diagram). Analogous improvements apply to the “dotted” and “squiggled” line styles.

Since this mechanism does not exist in the original `Xy-pic`, it can be switched on and off by `\xypdfcontpatternon` and `\xypdfcontpatternoff`. By default, it is switched on.



code:

```
\xy (0,0);(50,0)
  **\crv{~**\dir{--} (10,0)&(20,15)&(30,15)&(40,0)}
\endxy
```



6 Troubleshooting

- `TEX` complains ! No room for a new `\dimen`.

Try to load `Xy-pic` with the `pdf` option as early as possible. `xypdf` assigns 20 new dimension registers which are released at the end of the initialization. Thus, it needs 20 free dimension registers but will effectively not occupy new dimension registers.

- I get the error message `pdfTEX version 1.40.0 or higher is needed for the xypdf package with PDF output`

You seem to use an old version of `pdfTEX`. If you cannot update your `TEX` system for some reason, you may still use the `xypdf` package in DVI mode and produce a PDF file via `dvipdfm(x)`. The pathway `TEX` \rightarrow `dvipdfm(x)` is preferable in many cases anyway since it usually produces much smaller PDF files.

- I get the error message `eTEX is needed for the xypdf package`.

In some `TEX` installations, the ε -`TEX` features are not enabled, although they most certainly can be in any reasonably modern `TEX` installation. The picture is heterogeneous, e.g. the author’s `MiKTEX` and `TEX Live 2009` have the ε -`TEX` features enabled without further ado, while another user reported the above error message in his `TEX Live 2009`. Here is what you can do:

You must rebuild the `TEX` and `LATEX` format files with ε -`TEX` enabled. If you are an expert, you may know how to do this anyway and may skip the following items. Otherwise, follow the instructions below for `TEX Live`. For other `TEX` distributions, please consult the respective documentation on how to build the format files.

1. Locate the file `fmtutil.cnf` (probably in `/texmf-var/web2c/`).
2. Look at the lines starting with `latex` and `pdflatex`. They probably end in `latex.ini` and `pdflatex.ini` *without* a star `*` before these last parameters. If there is a star, the problem is somewhere else.
3. Generate a new file `fmtutil-local.cnf` in `/texmf-local/web2c/` with the following content:

```

#!latex
latex pdftex <options> *latex.ini
#!pdflatex
pdflatex pdftex <options> *pdflatex.ini

```

Take the `<options>` from the corresponding lines in `fmtutil.cnf`. The important change is the star prefix to the last parameters. This tells \TeX to go into extended (ϵ - \TeX) mode.

4. Run `tlmgr generate fmtutil` to update the configuration file `fmtutil.cnf`.
5. Run `fmtutil-sys --all` to generate the \TeX format files.

7 Copyright, license and disclaimer

The copyright for the `xypdf` package is by its author, Daniel Müllner. Current contact details will be maintained at <http://www.math.uni-bonn.de/people/muellner>.

The `xypdf` package is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version. This license is available at <http://www.gnu.org/licenses/>.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

8 Implementation

This is the code for the file xypdf.tex. From version 1.4 on, it is loaded as the option pdf to \XeTeX -pic.

```
\xypdfversion \xyprovide defines \xypdfversion.
\xypdfdate    1 <{*basic>
               2 \ifx\xyloaded\undefined\input xy \fi
               3 \xyprovide{pdf}{PDF driver}{1.4}%
               4 {Daniel M\"ullner\newline}%
               5 {\url{http://www.math.uni-bonn.de/people/muellner}}{}
               6 \ifx\makeatletter\undefined\input miniltx \fi
               7 \newcommand*\xypdfdate{2010/05/13}
               8 \newdriver{%
               9 \xyaddsupport{pdf}\xP@pdf@on
              10 \xyaddsupport{color}\xP@color@on
              11 \xyaddsupport{curve}\xP@curve@on
              12 \xyaddsupport{frame}\xP@frame@on
              13 % \xyaddsupport{line}\xP@line@on
              14 \xyaddsupport{rotate}\xP@rotate@on
              15 }
              16 \xyaddunsupport{pdf}\xP@pdf@off
              17 \xyaddunsupport{color}\xP@color@off
              18 \xyaddunsupport{curve}\xP@curve@off
              19 \xyaddunsupport{frame}\xP@frame@off
              20 %\xyaddunsupport{line}\xP@line@off
              21 \xyaddunsupport{rotate}\xP@rotate@off
```

See the end of the file for the code that loads the other files xypdf-*.tex.

```
\xypdfon    Commands for switching the driver on and off.
\xypdfoff   22 \newcommand*\xypdfon{%
              23 \xP@pdf@on
              24 \xP@color@on
              25 \xP@curve@on
              26 \xP@frame@on
              27 \xP@line@on
              28 \xP@rotate@on
              29 }
              30 \newcommand*\xypdfoff{%
              31 \xP@pdf@off
              32 \xP@color@off
              33 \xP@curve@off
              34 \xP@frame@off
              35 \xP@line@off
              36 \xP@rotate@off
              37 }
```

Test for $\varepsilon\text{-TeX}$

```
38 \ifx\unexpanded\@undefined
39 \PackageError{xypdf}{ $\varepsilon\text{TeX}$  is needed for the xypdf package}{}
40 \fi
```

\xP@testpdfsave Test for \pdfsave, which was introduced in pdf \TeX version 1.40.0.

```
41 \newcommand*\xP@testpdfsave{%
42 \ifpdf
43 \ifx\pdfsave\@undefined
```



```

44      \PackageError{xypdf}{pdfTeX version 1.40.0 or higher is needed for the %
45      xypdf^^J%
46      package with PDF output}{}%
47      \fi
48      \fi
49      \let\xP@testpdfsave\@undefined
50  }

\xP@warning Check for \PackageWarning.
51  \ifx\PackageWarning\@undefined
52    \newcommand*\xP@warning[2]{\{%
53      \newlinechar‘^^J%
54      \@warning{Package #1 Warning: #2\@empty.}%
55    }}
56  \else
57    \newcommand*\xP@warning{\PackageWarning}
58  \fi

\xP@pdf@on The following initializations are necessary for each supported extension, otherwise the
\xP@pdf@off \xP@hook commands will not work.
59  \newcommand*\xP@pdf@on{}
60  \newcommand*\xP@pdf@off{}

\xP@color@on
\xP@color@off 61  \newcommand*\xP@color@on{}
62  \newcommand*\xP@color@off{}

\xP@curve@on
\xP@curve@off 63  \newcommand*\xP@curve@on{}
64  \newcommand*\xP@curve@off{}

\xP@frame@on
\xP@frame@off 65  \newcommand*\xP@frame@on{}
66  \newcommand*\xP@frame@off{}

\xP@line@on
\xP@line@off 67  \newcommand*\xP@line@on{}
68  \newcommand*\xP@line@off{}

\xP@rotate@on
\xP@rotate@off 69  \newcommand*\xP@rotate@on{}
70  \newcommand*\xP@rotate@off{}

\xP@hook Commands for switching the driver on and off.
71  \newcommand*\xP@hook[2]{%
72    \edef\next@{%
73      \let\expandafter\noexpand\csname xP@old@#2\endcsname
74      \expandafter\noexpand\csname#2\endcsname}%
75    \next@
76    \expandafter\edef\csname xP@#1@on\endcsname{%
77      \unexpanded\expandafter\expandafter\expandafter{\csname xP@#1@on\endcsname}%
78      \let\expandafter\noexpand\csname#2\endcsname
79      \expandafter\noexpand\csname xP@#2\endcsname
80    }%
81    \expandafter\edef\csname xP@#1@off\endcsname{%

```

```

82   \unexpanded\expandafter\expandafter\expandafter{\csname xP@#1@off\endcsname}%
83   \let\expandafter\noexpand\csname#2\endcsname
84   \expandafter\noexpand\csname xP@old@#2\endcsname
85 }%
86 }

```

`\xP@defpdfliteral` Two possibilities to insert literal PDF commands, one for pdftex and one for dvipdfm(x).

`\xP@literal` The command `\xP@cm` changes the current transformation matrix.

```

      \xP@cm      87 \newcommand*\xP@defpdfliteral{%
      \xP@setcolor 88 \ifpdf
      \xP@resetcolor 89 \newcommand*\xP@literal[1]{\pdfsave\pdfliteral{##1}\pdfrestore}
      \xP@resetcolor 90 \newcommand*\xP@cm[5]{%
      \xP@resetcolor 91 \pdfsave
      \xP@resetcolor 92 \pdfsetmatrix{##1 ##2 ##3 ##4}%
      \xP@resetcolor 93 ##5%
      \xP@resetcolor 94 \pdfrestore
      \xP@resetcolor 95 }

```

Mimick pdfTeX.def from the graphicx package.

```

96 \ifundefined{@pdfcolorstack}{%
97   \def\pdfcolorstack{\z@}%
98 }{%
99   \newcommand*\xP@setcolor[3]{%
100     \pdfcolorstack\pdfcolorstack push{##1 ##2 ##1 ##3}}
101   \newcommand*\xP@resetcolor{\pdfcolorstack\pdfcolorstack pop\relax}%
102 \else
103   \newcommand*\xP@literal{%
104     \xP@warning{xypdf}{%
105       The produced DVI file is NOT PORTABLE. Convert it with^^J%
106       dvipdfm(x) to the PDF format but do not expect the DVI file itself to be^^J%
107       displayed correctly\@gobble}%
108     \global\let\xP@literal\xP@literal@
109     \xP@literal
110   }
111   \newcommand*\xP@literal@[1]{\special{pdf:content ##1}}
112   \newcommand*\xP@cm[5]{%
113     \special{pdf:btrans matrix ##1 ##2 ##3 ##4 0 0}%
114     ##5%
115     \special{pdf:etrans}%
116   }
117   \newcommand*\xP@setcolor[3]{\special{pdf:bcolor[##1]}}
118   \newcommand*\xP@resetcolor{\special{pdf:ecolor}}%
119 \fi
120 \let\xP@defpdfliteral\@undefined
121 }

```

Rely on the ifpdf package to test for PDF output. The `\AtEndOfPackage` is necessary if xypdf is loaded as an option in `\usepackage[options]{xy}`. If it is called as a plain T_EX package, the commands below can be executed immediately.

```

122 \DN@{\@firstofone}
123 \DNii@{xy}
124 \ifx\@currname\nextii@
125   \ifx\AtEndOfPackage\@undefined
126   \else
127     \DN@{\AtEndOfPackage}%
128   \fi

```

```

129 \fi
130 \next@
131 {\RequirePackage{ifpdf}}%
132 \xP@testpdfsave
133 \xP@defpdfliteral}

\xP@digits Set the precision for dimension output according to pdfTeX's \pdfdecimaldigits. If this
number is not defined, use dvipdfm's default precision, which is two decimals.
134 \ifx\pdfdecimaldigits\undefined
135 \newcommand*\xP@digits{2}
136 \else
137 \@ifdefinable\xP@digits\relax
138 \xdef\xP@digits{\the\pdfdecimaldigits}
139 \ifnum\pdfdecimaldigits<2
140 \xP@warning{xypdf}{%
141 The precision in \string\pdfdecimaldigits\space is only \xP@digits\space
142 decimals.^^J%
143 It is recommended to set \string\pdfdecimaldigits\space to 2 or 3 for %
144 best output quality@gobble}
145 \fi
146 \fi

\xP@dim Conversion between TeX points (pt) and PDF/Postscript points (bp)
147 \newcommand*\xP@dim[1]{%
148 \expandafter\xP@removePT\the\dimexpr(#1)*800/803\relax\space}

\xP@precdim Precise conversion between TeX points (pt) and PDF/Postscript points (bp). No trunca-
tion.
149 \newcommand*\xP@precdim[1]{\xP@EARPT\dimexpr(#1)*800/803\relax\space}

\xP@EARPT
150 \newcommand*\xP@EARPT{\expandafter\removePT@the}

\xP@coord Coordinates: two dimensions
151 \newcommand*\xP@coord[1]{\xP@dim{#1}\xP@dim}

\xP@removePT The following two macros round and truncate a dimension to the desired number of decimal
digits.
152 \@ifdefinable\xP@removePT\relax
153 {\@makeother\p\@makeother\t\gdef\xP@removePT#1pt{\xP@removePT@#10000@}}

\xP@removePT@
154 \@ifdefinable\xP@removePT@\relax
155 \ifcase\xP@digits
0 decimals
156 \def\xP@removePT@#1.#2#3@{%
157 \ifnum#2<5
158 #1%
159 \else
160 \the\numexpr-\if-#1-\else-#1+\fi\@ne\relax
161 \fi
162 }
163 \or

```

1 decimal

```
164 \def\xP@removePT@#1#2.#3#4#5@{%
165   \ifnum#4<5
166     #1#2%
167     \if#30%
168       \else
169         .#3%
170       \fi
171     \else
172       \expandafter\xP@removePT
173       \the\dimexpr#1#2.#3pt+\if#1--\fi.12pt\relax
174     \fi
175   }
176 \or
```

2 decimals

```
177 \def\xP@removePT@#1#2.#3#4#5#6@{%
178   \ifnum#5<5
179     #1#2%
180     \if#40%
181       \if#30%
182       \else
183         .#3%
184       \fi
185     \else
186       .#3#4%
187     \fi
188   \else
189     \expandafter\xP@removePT
190     \the\dimexpr#1#2.#3#4pt+\if#1--\fi786sp\relax
191   \fi
192 }
193 \or
```

3 decimals

```
194 \def\xP@removePT@#1#2.#3#4#5#6#7@{%
195   \ifnum#6<5
196     #1#2%
197     \if#50%
198       \if#40%
199       \if#30%
200       \else
201         .#3%
202       \fi
203     \else
204       .#3#4%
205     \fi
206   \else
207     .#3#4#5%
208   \fi
209 \else
210   \expandafter\xP@removePT
211   \the\dimexpr#1#2.#3#4#5pt+\if#1--\fi79sp\relax
212 \fi
213 }
214 \or
```

4 decimals

```

215 \def\xP@removePT@#1#2.#3#4#5#6#7#8@{%
216   \ifnum#7<5
217     #1#2%
218     \if#60%
219       \if#50%
220         \if#40%
221           \if#30%
222             \else
223               .#3%
224             \fi
225           \else
226             .#3#4%
227           \fi
228         \else
229           .#3#4#5%
230         \fi
231       \else
232         .#3#4#5#6%
233       \fi
234     \else
235       \expandafter\xP@removePT
236       \the\dimexpr#1#2.#3#4#5#6pt+\if#1--\fi8sp\relax
237     \fi
238   }
239 \else
240   5 or more decimals: no truncation
241 \let\xP@dim\xP@precdim
242 \fi

```

`\xP@lw` Find out the default line width in the math fonts. This is done at the beginning of the document, when hopefully all potential changes to math fonts have taken place.

`\xP@preclw`

```

242 \AtBeginDocument{%
Initialize math fonts
243 {\setbox0\hbox{$ $}}%
244 \@ifdefinable\xP@lw\relax
245 \@ifdefinable\xP@preclw\relax
246 \edef\xP@preclw{\the\fontdimen8\textfont3}%
247 \edef\xP@lw{\xP@dim\xP@preclw}%
248 \PackageInfo{xypdf}{Line width: \xP@preclw}%
249 }

```

8.1 Straight lines

`\line@` Also change the code for `\dir{-}` as an object. Now these dashes are not drawn from the dash font any more but by generic PDF line commands.

```

250 \xP@hook{pdf}{\line@}
251 \newcommand*\xP@line@{%
252   \setboxz@h{%
253     \xP@setsolidpat
254     \xP@stroke{0 0 m \xP@coord{\cosDirection\xdashl@}{\sinDirection\xdashl@}l}%
255   }%
256   \U@c{\sinDirection\xdashl@
257   \D@c\z@
258   \ifdim\U@c<\z@
259     \multiply\U@c\m@ne

```

```

260 \xP@swapdim\U@c\D@c
261 \fi
262 \ht\z@\U@c
263 \dp\z@\D@c
264 \R@c\cosDirection\xydashl@
265 \L@c\z@
266 \ifdim\R@c<\z@
267 \multiply\R@c\m@ne
268 \xP@swapdim\L@c\R@c
269 \fi
270 \hskip\L@c\boxz@\hskip\R@c
271 \edef\tmp@{\egroup\U@c\the\U@c\D@c\the\D@c\L@c\the\L@c\R@c\the\R@c}%
272 \tmp@
273 \Edge@c={\rectangleEdge}%
274 \edef\Upness@{\ifdim\z@<\U@c1\else0\fi}%
275 \edef\Leftness@{\ifdim\z@<\L@c1\else0\fi}%
276 \def\Drop@@{\styledboxz@}\def\Connect@@{\solid@}%
277 }

\solid@ This is the hook for solid straight lines. Derived from \xyPSSolid@ in xyps.tex.
\xP@solid@ 278 \xP@hook{pdf}{solid@}
279 \newcommand*\xP@solid@{\straight@\xP@solidSpread}

\xP@solidSpread
280 \@ifdefinable\xP@solidSpread\relax
281 \def\xP@solidSpread#1\repeat@{\{%
Neglect zero-length lines.
282 \@tempswatrue
283 \ifdim\X@p=\X@c
284 \ifdim\Y@p=\Y@c
285 \@tempswafalse
286 \fi
287 \fi
288 \if@tempswa
289 \xP@setsolidpat
290 \xP@stroke{\xP@coord\X@p\Y@p m \xP@coord\X@c\Y@c l}%
291 \fi
292 }}

\xP@pattern
293 \newcommand*\xP@pattern{}

\xP@setsolidpat Pattern for solid lines
294 \newcommand*\xP@setsolidpat{%
295 \def\xP@pattern{1 J 1 j [] 0 d}%
296 \global\let\xP@lastpattern\xP@solidmacro
297 }

\xP@stroke
298 \newcommand*\xP@stroke[1]{\xP@literal{\xP@lw w \xP@pattern\space#1 S}}

\dash@ This is the hook for dashed straight lines. Derived from \xyPSdashed@ in xyps.tex.
\xP@dashed@ 299 \xP@hook{pdf}{dash@}
300 \newcommand\xP@dashed@{\line@def\Connect@@{\straight@\xP@dashedSpread}}

```

\xP@dashedSpread

```

301 \ifdefinable\xP@dashedSpread\relax
302 \def\xP@dashedSpread#1\repeat@{ {%
303   \xP@vecLen
Neglect zero-length lines.
304   \ifdim\@tempdimb>\z@
305     \xP@setdashpat
306     \xP@savec
307     \xP@stroke{\xP@coord\X@p\Y@p m \xP@coord\X@c\Y@c l}%
308   \fi
309 }}
```

\xP@setdashpat The formula for the dash length is the same as in the dashed operator in xypsdict.tex:

$$(\text{dash length}) = \frac{l}{2 \cdot \text{round}\left(\frac{l+d}{2d}\right) - 1},$$

where l is the length of the line and d is the minimal dash length.

The length l must be in \@tempdimb. The dash length is returned in \@tempdima.

```

310 \newcommand*\xP@setdashpat{%
311   \xP@testcont\xP@dashmacro
312   \ifxP@splinecont
Special pattern in case this line continues another dashed segment.
313     {\count@ \numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
314     \global\dimen@i
315     \ifnum\count@>\z@
316       \dimexpr\@tempdimb/\count@\relax
317     \else
318       \z@
319     \fi
320   }%
321   \@tempdima\dimen@i
322   \edef\xP@pattern{1 J 1 j [%
323     \ifdim\@tempdima>\z@
324       \xP@precdim\@tempdima]\xP@precdim\@tempdima
325     \else
326       ]0 %
327     \fi
328   d}%
329   \else
330     \@tempdima
331     \ifdim\@tempdimb>\xydashl@
332       \dimexpr\@tempdimb/(2*((\@tempdimb+\xydashl@)/(2*\xydashl@))-1)\relax
333     \else
334       \z@
335     \fi
336     \edef\xP@pattern{1 J 1 j [%
337       \ifdim\@tempdima>\z@\xP@precdim\@tempdima\fi
338     ]0 d}%
339   \fi
340   \global\let\xP@lastpattern\xP@dashmacro
341 }
```

\xP@setclashpat Dash pattern for closed paths. Offset is half of the dash length to avoid artifacts.

```

342 \newcommand*\xP@setcldashpat{%
343   {\count@\numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
344   \xdef\@gtempa{1 J 1 j [\ifnum\count@>\z@\xP@precdim{\@tempdimb/\count@}\fi}%
345   \ifnum\count@>\z@\xP@precdim{\@tempdimb*3/2/\count@}\else0 \fi d}%
346 }%
347 \edef\xP@pattern{\@gtempa}%
348 }

```

\point@ This is the hook for points. Derived from \xyPSPoint@ in xyps.tex.

```

\xP@point@ 349 \xP@hook{pdf}{point@}
350 \newcommand*\xP@point@{\xP@zerodot\egroup\Invisible@false
351   \Hidden@false\def\Leftness@{.5}\def\Upness@{.5}\ctipEdge@
352   \def\Drop@@{\stylenboxz}%
353   \def\Connect@@{\straight@\xP@dottedSpread}%
354 }

```

\xP@zerodot

```

355 \newcommand*\xP@zerodot{%
356   \hb@xt@\z@{\hss
357     \vbox to\z@{\vss\hrule\@width\xP@preclw\@height\xP@preclw\vss}%
358     \hss}%
359 }

```

\xP@dottedSpread

```

360 \@ifdefinable\xP@dottedSpread\relax
361 \def\xP@dottedSpread#1\repeat@{ {%
362   \xP@veclen
363   \ifdim\@tempdimb>\z@
364     \xP@setdottedpat
365     \xP@savvec
366     \xP@stroke{\xP@coor\X@p\Y@p m \xP@coor\X@c\Y@c l}%
367     \fi
368 }}

```

\xP@setdottedpat The formula for the distance between dots is the same as in the dotted operator in xypsdict.tex:

$$(\text{dot distance}) = \frac{l}{\text{round}\left(\frac{l}{2\text{pt}}\right) + 1},$$

where l is the length of the line.

The length l must be in \@tempdimb.

```

369 \newcommand*\xP@setdottedpat{%
370   \xP@testcont\xP@dotmacro
371   \ifxP@splinecont
372     \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
373     \edef\xP@pattern{%
374       0 J [%

```

Produce a dot pattern only when the segment is long enough.

```

375     \ifdim\@tempdima>\z@
376     \xP@precdim\xP@preclw\xP@precdim\@tempdima
377     \fi

```

Advance the offset very slightly by 1sp to really hide the first dot in the viewer. (This improves the display at least in the author's PDF-Xchange viewer.)

```

378     ]\xP@precdim{\xP@preclw+1sp}d}%

```



```

379 \else
380 \advance\@tempdimb-\xP@preclw
381 \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
382 \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
383 \edef\xP@pattern{%
384 0 J [%
Produce a dot pattern only when the segment is long enough.
385 \ifdim\@tempdima>\z@
386 \xP@precdim\xP@preclw\xP@precdim\@tempdima
387 \fi
388 ]0 d}%
389 \fi
390 \global\let\xP@lastpattern\xP@dotmacro
391 }

```

`\xP@setcldottedpat` Dotted pattern for closed paths. Offset is half of the dot distance to avoid artifacts.

```

392 \newcommand*\xP@setcldottedpat{%
393 \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
394 \edef\xP@pattern{%
395 0 J [%
396 \ifdim\@tempdima>\z@
397 \xP@precdim\xP@preclw\xP@precdim\@tempdima
398 \fi
399 ]\xP@precdim{\dimexpr\xP@preclw+\@tempdima/2\relax}d}%
400 }

```

In contrast to the Postscript drivers for X_y-pic, where some computations are left to the Postscript code, all arithmetic for the PDF output must be done by T_EX itself. With T_EX's rudimentary fixed-point arithmetic, it is still a pain to compute even the length of a line segment, but things have become considerably easier with ε -T_EX.

`\xP@abs` Absolute value

```

401 \newcommand*\xP@abs[1]{\ifdim#1<\z@\multiply#1\m@ne\fi}

```

`\xP@ifabsless`

```

402 \newcommand*\xP@ifabsless[2]{\ifpdfabsdim#1<#2}
403 \ifx\ifpdfabsdim\undefined
404 \renewcommand*\xP@ifabsless[2]{\ifdim\ifdim#1<\z@-\fi#1<\ifdim#2<\z@-\fi#2}
405 \@gobble\fi
406 \fi

```

`\xP@swapdim` Works unless parameter #2 is `\@tempdima`.

```

407 \newcommand*\xP@swapdim[2]{\@tempdima#1#1#2#2\@tempdima}

```

`\xP@swapnum` Works unless parameter #2 is `\@tempcnta`.

```

408 \newcommand*\xP@swapnum[2]{\@tempcnta#1#1#2#2\@tempcnta}

```

`\xP@min` Maximum of two lengths

```

409 \newcommand*\xP@min[2]{\ifdim#1<#2#1\else#2\fi}

```

`\xP@max` Maximum of two lengths

```

410 \newcommand*\xP@max[2]{\ifdim#1>#2#1\else#2\fi}

```

`\xP@Max` Assigns #1 the maximum of #1 and the absolute value of #2.

```

411 \newcommand*\xP@Max[2]{#1\ifdim#2<\z@\xP@max#1{-#2}\else\xP@max#1#2\fi}

```

`\xP@sqrt` Square root algorithm. The argument is in `\@tempdima`, and the start value for the iteration in `\@tempdimc`. The result goes into `\@tempdimb`.

```

412 \newcommand*\xP@sqrt{%
413   \loop
414     \@tempdimb\dimexpr(\@tempdimc+(\@tempdima*\p@/\@tempdimc))/2\relax

```

ε -TeX's `\unless` instead of `\else` since the plain TeX `\loop` cannot deal with `\else`.

```

415   \unless\ifdim\@tempdimc=\@tempdimb
iterate: (old approx.) := (new approx.)
416     \@tempdimc\@tempdimb\relax
417   \repeat
418 }

```

`\xP@veclen` Absolute length of the vector (`\d@X`, `\d@Y`). The result goes into the register `\@tempdimb`. Several L^AT_EX registers are used as temporary registers, so this function is called safely within a group.

(Maybe it is not necessary to scale the coordinates so much as it is done here, and a simpler code would be fine as well.)

```

419 \newcommand*\xP@veclen{%
420   \xP@veclen@
421   \global\dimen@i\@tempdimb
422   }\@tempdimb\dimen@i
423 }

```

`\xP@veclen@`

```

424 \newcommand*\xP@veclen@{%
425   \xP@abs\d@Y

```

1) Strictly vertical vector

```

426   \ifdim\d@X=\z@
427     \@tempdimb\d@Y
428   \else
429     \xP@abs\d@X

```

2) Strictly horizontal vector

```

430   \ifdim\d@Y=\z@
431     \@tempdimb\d@X
432   \else

```

3) Diagonal vector. 5931642sp = $\sqrt{\maxdimen}/2$. Test whether the components are small enough so that their sum of squares does not generate an arithmetic overflow.

```

433     \@tempswatrue
434     \ifdim\d@X>5931641sp\relax\@tempswafalse\fi
435     \ifdim\d@Y>5931641sp\relax\@tempswafalse\fi
436     \if@tempswa

```

3a) Small vector. `\count@` contains a scaling factor for a precise fixed-point arithmetic.

```

437     \count@\@ne
438     \loop
439       \@tempdima\dimexpr\d@X*\d@X/\p@+\d@Y*\d@Y/\p@\relax

```

If the coordinates are small enough, scale them up to improve precision.

```

440       \ifdim\@tempdima<4096pt
441         \@tempcnta\ifdim\@tempdima<1024pt\ifdim\@tempdima<256pt8\else4\fi%
442         \else\tw@\fi
443       \multiply\d@X\@tempcnta
444       \multiply\d@Y\@tempcnta

```

```

445         \multiply\count@\@tempcnta
446     \repeat
Starting value for the square root algorithm
447     \@tempdimc\dimexpr(\d@X+\d@Y)*3/4\relax
448     \xP@sqrt
Rescale
449     \@tempdimb\dimexpr\@tempdimb/\count@\relax
450     \else
451     \ifdim\d@X>83042982sp\relax\@tempwattrue\fi
452     \ifdim\d@Y>83042982sp\relax\@tempwattrue\fi
453     \if@tempswa
3b) Large vector. Scale the coordinates down to avoid an overflow. 11927552sp = 182pt
454     \@tempdima\dimexpr\d@X/182*\d@X/11927552+\d@Y/182*\d@Y/11927552\relax
455     \@tempdimc\dimexpr(\d@X+\d@Y)*3/728\relax
456     \xP@sqrt
457     \multiply\@tempdimb182\relax
458     \else
3c) Medium vector. Also scale the coordinates down. 12845056sp = 196pt = 142pt
459     \@tempdima\dimexpr\d@X*\d@X/12845056+\d@Y*\d@Y/12845056\relax
460     \@tempdimc\dimexpr(\d@X+\d@Y)*3/56\relax
461     \xP@sqrt
462     \multiply\@tempdimb14\relax
463     \fi
464     \fi
465     \fi
466     \fi
467 }

```

8.2 Squiggled lines

```

\squiggledSpread@ This is the hook for squiggled straight lines.
\xP@squiggledSpread@
468 \xP@hook{pdf}{\squiggledSpread@}
469 \@ifdefinable\xP@squiggledSpread@\relax
470 \def\xP@squiggledSpread@#1\repeat@{
471     \xP@veclen
Neglect zero-length lines.
472     \ifdim\@tempdimb>\z@
473     \edef\@tempa{\xP@coord\X@p\Y@p m }%
474     \toks@\expandafter{\@tempa}%
\@tempcnta = number of squiggles
475     \@tempcnta\numexpr\@tempdimb/\xybsql1@\relax
476     \ifnum\@tempcnta<\tw@\@tempcnta\tw@\fi
477     \@tempdima\dimexpr\d@X/\@tempcnta\relax
478     \@tempdimc\dimexpr\d@Y/\@tempcnta\relax
Reverse the direction of the little arcs, if the last squiggle from the previous segment makes
it necessary.
479     \xP@testcont\xP@oddsquigglemacro
480     \ifxP@splinecont
481     \def\xP@squigsign{-}%
482     \else
483     \let\xP@squigsign\@empty
484     \fi

```

```

485 \count@\z@
486 \loop

```

The fraction is the continuous fraction approximation for the best spline approximation to a quarter circle ($147546029/534618434 \approx \frac{1}{2} \cdot 0.55196760761152504532$).

```

487 \xP@append\toks@{%
488 \xP@coor{\X@p+\d@X*\count@/\@tempcnta+(\@tempdima
489 -\xP@squigsign\ifodd\count@-\fi\@tempdimc)*147546029/534618434}%
490 {\Y@p+\d@Y*\count@/\@tempcnta+(\@tempdimc
491 +\xP@squigsign\ifodd\count@-\fi\@tempdima)*147546029/534618434}%
492 }%
493 \advance\count@\@ne
494 \xP@append\toks@{%
495 \xP@coor{\X@p+\d@X*\count@/\@tempcnta-(\@tempdima
496 -\xP@squigsign\ifodd\count@-\fi\@tempdimc)*147546029/534618434}%
497 {\Y@p+\d@Y*\count@/\@tempcnta-(\@tempdimc
498 +\xP@squigsign\ifodd\count@-\fi\@tempdima)*147546029/534618434}%
499 \xP@coor{\X@p+\d@X*\count@/\@tempcnta}%
500 {\Y@p+\d@Y*\count@/\@tempcnta}%
501 c }%
502 \ifnum\count@<\@tempcnta
503 \repeat
504 \xP@setsolidpat

```

Record the direction of the last squiggle.

```

505 \global\expandafter\let\expandafter\xP@lastpattern
506 \ifodd\numexpr\count@\if\xP@squigsign-+1\fi\relax
507 \xP@oddsquigglemacro
508 \else
509 \xP@evensquigglemacro
510 \fi
511 \xP@savec
512 \xP@stroke{\the\toks@}%
513 \fi
514 }}

```

\xP@squigsign

```

515 \newcommand*\xP@squigsign{}

```

\xP@append

```

516 \newcommand*\xP@append[2]{\{
517 \edef\@tempa{\#1\the\#2}\}%
518 \expandafter\xP@tempa
519 }

```

8.3 Circles

\circhar@@ Replacement macro for the circle chars.

```

\xP@circhar@@ 520 \xP@hook{pdf}{\circhar@@}
521 \newcommand*\xP@circhar@@[1]{%
522 \expandafter\xP@circhar@@\ifcase#1 %

```

Bézier segments for 1/8 circle. Let

$$a := \sqrt{1/2} \approx .707106781,$$

$$b := \frac{8}{3}\sqrt{2} \cos(\pi/8) (1 - \cos(\pi/8)) \approx .2652164898,$$

$$c := \frac{1}{3}(-3 + 8 \cos(\pi/8) - 2 \cos^2(\pi/8)) \approx .8946431596,$$

$$d := \frac{1}{2}b(2 + 3 \cos(\pi/8) - \cos^2(\pi/8)) \approx .5195704027.$$

(We have $\cos(\pi/8) = \frac{1}{2}\sqrt{2 + \sqrt{2}}$.)

The fractions below are best possible rational approximations (obtained by continued fractions) to the following coordinates:

$(0, 0), (0, -b), (1 - c, -d), (1 - a, -a)$

```
523      00%
524      0{-173517671/654249180}%
525      {65307479/619869377}{-34221476/65864945}%
526      {225058681/768398401}{-543339720/768398401}%
527      \or
```

$(0, -a), (a - d, -c), (a - b, -1), (a, -1)$

```
528      0{-543339720/768398401}%
529      {181455824/967576667}{-554561898/619869377}%
530      {826676217/1870772527}{-1}%
531      {543339720/768398401}{-1}%
532      \or
```

$(0, -1), (b, -1), (d, -c), (a, -a)$

```
533      0{-1}%
534      {173517671/654249180}{-1}%
535      {34221476/65864945}{-554561898/619869377}%
536      {543339720/768398401}{-543339720/768398401}%
537      \or
```

$(0, -a), (c - a, -d), (1 - a, -b), (1 - a, 0)$

```
538      0{-543339720/768398401}%
539      {181455824/967576667}{-34221476/65864945}%
540      {225058681/768398401}{-173517671/654249180}%
541      {225058681/768398401}0%
542      \or
```

$(0, a), (c - a, d), (1 - a, b), (1 - a, 0)$

```
543      0{543339720/768398401}%
544      {181455824/967576667}{34221476/65864945}%
545      {225058681/768398401}{173517671/654249180}%
546      {225058681/768398401}0%
547      \or
```

$(0, 1), (b, 1), (d, c), (a, a)$

```
548      01%
549      {173517671/654249180}1%
550      {34221476/65864945}{554561898/619869377}%
551      {543339720/768398401}{543339720/768398401}%
552      \or
```

$(0, a), (a - d, c), (a - b, 1), (a, 1)$

```
553      0{543339720/768398401}%
554      {181455824/967576667}{554561898/619869377}%
555      {826676217/1870772527}1%
556      {543339720/768398401}1%
557      \or
```

$(0, 0), (0, b), (1 - c, d), (1 - a, a)$

```
558      00%
559      0{173517671/654249180}%

```

```

560 {65307479/619869377}{34221476/65864945}%
561 {225058681/768398401}{543339720/768398401}%
562 \fi}

```

`\xP@circhar@@@` Draw the arc of $1/8$ circle and use the same space as the chars from the circle font do.

```

563 \newcommand\xP@circhar@@@[8]{%
564 \xP@setsolidpat
565 \xP@stroke{\xP@coord{\R@*#1}{\R@*#2}m
566 \xP@coord{\R@*#3}{\R@*#4}\xP@coord{\R@*#5}{\R@*#6}%
567 \xP@coord{\R@*#7}{\R@*#8}c}%
568 \vrule width\z@ height\R@ depth\R@
569 \kern\dimexpr\R@*#7\relax
570 }

```

`\cirrestrict@@` Basically, `\cirrestrict@@` is turned into a no-op and does not change the radius.

```

\xP@cirrestrict@@ 571 \xP@hook{pdf}{\cirrestrict@@}
572 \newcommand*\xP@cirrestrict@@{\count@ \z@ \relax}

```

8.4 Rotation and scaling

```

573 </basic>
574 <*rotate>
575 \xycatcodes
576 \expandafter\let\csname xypdf-ro@loaded\endcsname\@empty

```

`\xyscale@@` Scale the box 0 to the factors in #1 and #2.

```

\xP@xyscale@@ 577 \xP@hook{rotate}{\xyscale@@}
578 \newcommand*\xP@xyscale@@[2]{%
579 \setboxz@h{%
580 \hskip\L@p
581 \hskip-\R@p
582 \lower\U@p\hbox{\xP@cm{#1}00{#2}%
583 {\raise\U@p\hb@xt@ \z@{\hskip-\L@p\boxz@ \hss}}}%
584 }%
585 }%
586 \global\let\xP@lastpattern\@empty
587 }

```

`\xyRotate@@` Rotation in the direction #1.

```

\xP@xyRotate@@ 588 \xP@hook{rotate}{\xyRotate@@}
589 \newcommand\xP@xyRotate@@{\xP@rotate@\xP@trigfromdir}

```

`\doSpecialRotate@@` Rotation by the angle in #1.

```

\xP@doSpecialRotate@@ 590 \xP@hook{rotate}{\doSpecialRotate@@}
591 \ifdefinable\xP@doSpecialRotate@@\relax
592 \def\xP@doSpecialRotate@@#1@@{\xP@rotate@\xP@trig{#1}}

```

`\xP@rotate@` Common code for both rotations: rotate the box 0.

```

593 \newcommand*\xP@rotate@[2]{%
594 \setboxz@h{%
595 #1{#2}%
596 \hskip\L@p
597 \hskip-\R@p
598 \lower\U@p\hbox{\xP@cm\cosDirection\sinDirection
599 {\if-\sinDirection\else-\sinDirection\fi}\cosDirection
600 {\raise\U@p\hb@xt@ \z@{\hskip-\L@p\boxz@ \hss}}}%

```

```

601    }%
602  }%
603  \global\let\xP@lastpattern\@empty
604 }

\xP@trig Calculate sine and cosine from the angle in #1.
605 \newcommand*\xP@trig[1]{%
606   \@tempdima\dimexpr#1pt\relax
  Translate the argument into the interval [0pt, 360pt].
607   \@tempdimb\@tempdima
  
$$23592960 = 360 \cdot 65536$$

608   \divide\@tempdimb23592960
609   \advance\@tempdima-23592960\@tempdimb
610   \ifdim\@tempdima<\z@\advance\@tempdima360pt\fi
611   \@tempdimb\@tempdima
  
$$5898240 = 90 \cdot 65536$$

612   \divide\@tempdimb5898240
  It's enough to know sin between  $0^\circ$  and  $90^\circ$ . The cos and the values in the other quadrants
  can be derived from that.
613   \ifcase\@tempdimb
614     \xP@sinpoly
615     \edef\sinDirection{\xP@EARPT\@tempdimb}%
616     \@tempdima\dimexpr90pt-\@tempdima\relax
617     \xP@sinpoly
618     \edef\cosDirection{\xP@EARPT\@tempdimb}%
619   \or
620     \@tempdima\dimexpr180pt-\@tempdima\relax
621     \xP@sinpoly
622     \edef\sinDirection{\xP@EARPT\@tempdimb}%
623     \@tempdima\dimexpr90pt-\@tempdima\relax
624     \xP@sinpoly
625     \edef\cosDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
626   \or
627     \@tempdima\dimexpr\@tempdima-180pt\relax
628     \xP@sinpoly
629     \edef\sinDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
630     \@tempdima\dimexpr90pt-\@tempdima\relax
631     \xP@sinpoly
632     \edef\cosDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
633   \or
634     \@tempdima\dimexpr360pt-\@tempdima\relax
635     \xP@sinpoly
636     \edef\sinDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
637     \@tempdima\dimexpr90pt-\@tempdima\relax
638     \xP@sinpoly
639     \edef\cosDirection{\xP@EARPT\@tempdimb}%
640   \else
641     \PackageError{xypdf}{Unexpected case in sin/cos calculation}%
642       {Feel free to contact the author of the xypdf package with a minimal %
643       example.}%
644   \fi
645 }

```

`\xP@sinpoly` Polynomial approximation to the sine in the interval [0pt,90pt]. The deviation should be $\pm 1\text{sp}$ maximal (but no guarantee). (3rd order, 4 subintervals, exact values for 0pt and 90pt)

```

646 \newcommand*\xP@sinpoly{%
647   \ifdim\@tempdima<49pt
648     \ifdim\@tempdima<27pt
649       \@tempdimb\dimexpr((\@tempdima*-529771058/16039085-1384933sp)%
650         * \@tempdima/268756075+10714164sp)*\@tempdima/613777813\relax
651     \else
652       \advance\@tempdima-27pt
653       \@tempdimb\dimexpr((\@tempdima*-743101305/20672414-238989613sp)%
654         * \@tempdima/80975565+42661556sp)*\@tempdima/622461739+2\p@)%
655         *157520747/693945047\relax
656     \fi
657   \else
658     \ifdim\@tempdima<70pt
659       \advance\@tempdima-49pt
660       \@tempdimb\dimexpr((\@tempdima*-348406699/107952940-55079229sp)%
661         * \@tempdima/866635628+408805sp)*\@tempdima/26926757+\p@)%
662         *135751711/179873976\relax
663     \else
664       \advance\@tempdima-70pt
665       \@tempdimb\dimexpr((\@tempdima*-1015850353/137849442-460519207sp)%
666         * \@tempdima/8742349+142263941sp)*\@tempdima/972432199+23\p@)%
667         *31253604/764969669\relax
668     \fi
669   \fi
670   \global\dimen@i\@tempdimb
671 } \@tempdimb\dimen@i
672 }

```

End of the section for Xy-pic’s “rotate” option. The macro `\xP@trigfromdir` below is also used for the `{-}` directional.

```

673 \xyendinput
674 </rotate>
675 <*basic>

```

`\xP@trigfromdir` Calculate sine and cosine from the direction number in #1.

```

676 \newcommand*\xP@trigfromdir[1]{%
677   \Direction#1\relax
678   \Direction mod 2048
679   \count@-\Direction
680   \advance\count@4096
681   \divide\count@2048
682   Assign the slope in the right way.
683   \ifcase\count@
684     \d@X\K@\p@
685     \d@Y\numexpr\Direction-3*\K@\relax\p@
686   \or
687     \d@X\numexpr\Direction-\K@\relax\p@
688     \d@Y-\K@\p@
689   \or
690     \d@X-\K@\p@
691     \d@Y\numexpr-\Direction-\K@\relax\p@
692   \or

```



```

691 \d@X\numexpr-\Direction-3*\K@\relax\p@
692 \d@Y\K@\p@
693 \else
694 \PackageError{xypdf}{Unexpected case in direction calculation}%
695 {Feel free to contact the author of the xypdf package with a minimal %
696 example.}%
697 \fi

Bring the pair (\d@X, \d@Y) to norm 1.

698 \xP@vecLen
699 \xdef\@gtempa{%
700 \def\noexpand\cosDirection{\xP@EARPT\dimexpr\d@X*\p@/\@tempdimb\relax}%
701 \def\noexpand\sinDirection{\xP@EARPT\dimexpr\d@Y*\p@/\@tempdimb\relax}%
702 }%
703 }\@gtempa
704 }

```

8.5 Temporary registers

\xP@newdimen Remove the \outer from \newdimen. (This applies for plain T_EX.)

```

705 \outer\def\@tempa{\alloc@1\dimen\dimendef\insc@unt}
706 \let\xP@newdimen\newdimen
707 \ifx\xP@newdimen\@tempa
708 \def\xP@newdimen{\alloc@1\dimen\dimendef\insc@unt}
709 \fi
710 \outer\def\@tempa#1{\count@=\escapechar\escapechar=-1
711 \expandafter\expandafter\expandafter
712 \def\@if#1{true}{\let#1=\iftrue}%
713 \expandafter\expandafter\expandafter
714 \def\@if#1{false}{\let#1=\iffalse}%
715 \@if#1{false}\escapechar=\count@}
716 \let\@tempa\relax

```

The next section is for the “curve” extension!

```

717 </basic>
718 <*curve>
719 \xycatcodes
720 \expandafter\let\csname xypdf-cu@loaded\endcsname\@empty

```

\xP@tempvar In order to save registers, xypdf shares X_Y-pic’s dimension and counter registers but uses different, more descriptive names. Every macro that uses these temporary variables must be safely encapsulated in a group so that the registers are not changed from the outside scope!

The xypdf package uses several sets of temporary variable names for different modules. Since it is important that these assignments do not overlap and that the variables are only used encapsulated within groups, the macros which use temporary variables are marked by colored bullets ●1, ●2, ●3, ●4, ●5, ●6, ●7 with one color for each set of variables.

The table in Figure 4 lists all variable assignments in these sets. It can be seen from the table which sets of variables can be used together. For example, set ●1 consisting of \xP@bigdim can be used together with all other temporary variables, while ●2 and ●4 must never be used together.

```

721 \newcommand*\xP@tempvar[2]{%
722 \ifdefinable#1\relax
723 \let#1#2%
724 }

```

Xy-pic var.	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
\quotPTK@	\xP@bigdim						
\L@p	\xP@parA	\xP@A		(\L@p)		(\L@p)	
\U@p	\xP@velA	\xP@B		(\U@p)		(\U@p)	
\R@p	\xP@parB	\xP@C		(\R@p)		(\R@p)	
\D@p	\xP@velB	\xP@D		(\D@p)		(\D@p)	
\X@origin	\xP@parC	\xP@E				\xP@temppar	
\Y@origin	\xP@velC	\xP@F				\xP@tempvel	
\X@xbase	\xP@parD	\xP@G				\xP@posX	
\Y@xbase	\xP@velD	\xP@H				\xP@posY	
\X@ybase	\xP@parE	\xP@I = \xP@a	\xP@a			\xP@oldpar	
\Y@ybase	\xP@velE	\xP@J = \xP@b	\xP@b			\xP@lastpar	
\X@min	\xP@lenA	\xP@K	\xP@c			\xP@tempvel@	
\Y@min	\xP@lenB	\xP@L	\xP@valA			\xP@parinc	
\X@max	\xP@partlen	\xP@fa	\xP@valB				
\Y@max	\xP@oldpartlen	\xP@fd	\xP@devA				
\almostz@	\xP@tolerance	\xP@tm	\xP@devB			\xP@squiglen	
\K@dXdY		\xP@xm	\xP@ti				
\K@dYdX		\xP@ym	\xP@tip				
new var. 1		\xP@off	(\xP@off)				
new var. 2		\xP@ta					
new var. 3		\xP@tb					
new var. 4		\xP@tc					
new var. 5		\xP@M					
new var. 6		\xP@oldobj					
new var. 7		\xP@Tax	\xP@sa				
new var. 8		\xP@Tay	\xP@sb				
new var. 9		\xP@Tdx	\xP@sc				
new var. 10		\xP@Tdy	\xP@Ab				
new var. 11		\xP@Tmx	\xP@AAb				
new var. 12		\xP@Tmy	\xP@Aba				
new var. 13		\xP@xa (\xP@xa)	\xP@Abb				
new var. 14		\xP@ya (\xP@ya)	\xP@Abc				
new var. 15		\xP@xb (\xP@xb)	\xP@AAba				
new var. 16		\xP@yb (\xP@yb)	\xP@AAbb				
new var. 17		\xP@xc (\xP@xc)	\xP@AAbc				
new var. 18		\xP@yc (\xP@yc)	\xP@dta				
new var. 19		\xP@xd (\xP@xd)	\xP@dtb				
new var. 20		\xP@yd (\xP@yd)	\xP@dtc				

Figure 4: Temporary dimension registers in xypdf.

`\xP@bigdim` ●1 A big constant less than $\frac{1}{3}\backslash\maxdimen \approx 5461\text{pt}$ and having many small prime factors.
725 `\xP@tempvar\xP@bigdim\quotPTK@`

`\xP@parA` ●2 Second set of temporary variables: for the arc length algorithm.
`\xP@velA` 726 `\xP@tempvar\xP@parA\L@p`
`\xP@parB` 727 `\xP@tempvar\xP@velA\U@p`
`\xP@velB` 728 `\xP@tempvar\xP@parB\R@p`
`\xP@parC` 729 `\xP@tempvar\xP@velB\D@p`
`\xP@velC` 730 `\xP@tempvar\xP@parC\X@origin`
`\xP@parD` 731 `\xP@tempvar\xP@velC\Y@origin`
`\xP@velD` 732 `\xP@tempvar\xP@parD\X@xbase`
`\xP@parE` 733 `\xP@tempvar\xP@velD\Y@xbase`
`\xP@velE` 734 `\xP@tempvar\xP@parE\X@ybase`
735 `\xP@tempvar\xP@velE\Y@ybase`
`\xP@lenA` 736 `\xP@tempvar\xP@lenA\X@min`
`\xP@lenB` 737 `\xP@tempvar\xP@lenB\Y@min`
`\xP@partlen` 738 `\xP@tempvar\xP@partlen\X@max`
`\xP@oldpartlen` 739 `\xP@tempvar\xP@oldpartlen\Y@max`
`\xP@tolerance` 740 `\xP@tempvar\xP@tolerance\almostz@`

`\xP@A` ●3 Third set of temporary registers: Bézier offset algorithm and solving linear equations.
`\xP@B` 741 `\xP@tempvar\xP@A\L@p`
`\xP@C` 742 `\xP@tempvar\xP@B\U@p`
`\xP@D` 743 `\xP@tempvar\xP@C\R@p`
`\xP@E` 744 `\xP@tempvar\xP@D\D@p`
`\xP@F` 745 `\xP@tempvar\xP@E\X@origin`
`\xP@G` 746 `\xP@tempvar\xP@F\Y@origin`
`\xP@H` 747 `\xP@tempvar\xP@G\X@xbase`
`\xP@I` 748 `\xP@tempvar\xP@H\Y@xbase`
`\xP@J` 749 `\xP@tempvar\xP@I\X@ybase`
`\xP@K` 750 `\xP@tempvar\xP@J\Y@ybase`
`\xP@L` 751 `\xP@tempvar\xP@K\X@min`
752 `\xP@tempvar\xP@L\Y@min`
`\xP@fa` 753 `\xP@tempvar\xP@fa\X@max`
`\xP@fd` 754 `\xP@tempvar\xP@fd\Y@max`
`\xP@tm` 755 `\xP@tempvar\xP@tm\almostz@`
`\xP@xm` 756 `\xP@tempvar\xP@xm\K@dXdY`
`\xP@ym` 757 `\xP@tempvar\xP@ym\K@dYdX`

`\xP@off` ●3 Alas, we need 20 more temporary registers. Hopefully, there are still free slots for
`\xP@ta` dimension registers. We take them for the temporary variables but release them afterwards
`\xP@tb` so that other packages can use them.
`\xP@tc` 758 `\@tempcnta\count11\relax`
`\xP@M` 759 `\xP@newdimen\xP@off`
`\xP@oldobj` 760 `\xP@newdimen\xP@ta`
`\xP@Tax` 761 `\xP@newdimen\xP@tb`
`\xP@Tay` 762 `\xP@newdimen\xP@tc`
`\xP@Tdx` 763 `\xP@newdimen\xP@M`
`\xP@Tdy` 764 `\xP@newdimen\xP@oldobj`
`\xP@Tmx` 765 `\xP@newdimen\xP@Tax`

`\xP@Tmy` ●4
`\xP@xa` 766 `\xP@newdimen\xP@Tay`
`\xP@ya` 767 `\xP@newdimen\xP@Tdx`
`\xP@xb` 768 `\xP@newdimen\xP@Tdy`
`\xP@yb` 769 `\xP@newdimen\xP@Tmx`
`\xP@xc`
`\xP@yc`
`\xP@xd`
`\xP@yd`

```

770 \xP@newdimen\xP@Tmy
771 \xP@newdimen\xP@xa
772 \xP@newdimen\xP@ya
773 \xP@newdimen\xP@xb
774 \xP@newdimen\xP@yb
775 \xP@newdimen\xP@xc
776 \xP@newdimen\xP@yc
777 \xP@newdimen\xP@xd
778 \xP@newdimen\xP@yd
779 \count11\@tempcnta

\xP@a •5 Fifth set of temporary variables: Parameters for drawing part of a spline segment.
\xP@b 780 \xP@tempvar\xP@a\X@ybase
\xP@c 781 \xP@tempvar\xP@b\Y@ybase
\xP@valA 782 \xP@tempvar\xP@c\X@min
\xP@valB 783 \xP@tempvar\xP@valA\Y@min
\xP@devA 784 \xP@tempvar\xP@valB\X@max
\xP@devB 785 \xP@tempvar\xP@devA\Y@max
\xP@ti 786 \xP@tempvar\xP@devB\almostz@
\xP@tip 787 \xP@tempvar\xP@ti\K@dXdY
788 \xP@tempvar\xP@tip\K@dYdX

\xP@sa •6 Sixth set of temporary variables: Solving a linear system approximately.
\xP@sb 789 \xP@tempvar\xP@sa\xP@Tax
\xP@sc 790 \xP@tempvar\xP@sb\xP@Tay
\xP@Ab 791 \xP@tempvar\xP@sc\xP@Tdx
\xP@AAb 792 \xP@tempvar\xP@Ab\xP@Tdy
\xP@Aba 793 \xP@tempvar\xP@AAb\xP@Tmx
\xP@Abb 794 \xP@tempvar\xP@Aba\xP@Tmy
\xP@Abc 795 \xP@tempvar\xP@Abb\xP@xa
796 \xP@tempvar\xP@Abc\xP@ya
797 \xP@tempvar\xP@AAba\xP@xb
798 \xP@tempvar\xP@AAbb\xP@yb
799 \xP@tempvar\xP@AAbc\xP@xc
800 \xP@tempvar\xP@dta\xP@yc
801 \xP@tempvar\xP@dtb\xP@xd
802 \xP@tempvar\xP@dtc\xP@yd

\xP@temppar •7 Seventh set of temporary registers: For multiple dotted splines.
\xP@tempvel 803 \xP@tempvar\xP@temppar\X@origin
\xP@posX 804 \xP@tempvar\xP@tempvel\Y@origin
\xP@posY 805 \xP@tempvar\xP@posX\X@xbase
\xP@oldpar 806 \xP@tempvar\xP@posY\Y@xbase
\xP@lastpar 807 \xP@tempvar\xP@oldpar\X@ybase
808 \xP@tempvar\xP@lastpar\Y@ybase
809 \xP@tempvar\xP@tempvel@X@min
810 \xP@tempvar\xP@parinc\Y@min
811 \xP@tempvar\xP@squiglen\almostz@

\xP@scaleone We also use temporary numerical registers for scaling factors in \xP@solvelinearsystem.
\xP@scaletwo 812 \xP@tempvar\xP@scaleone\K@
\xP@scaletthree 813 \xP@tempvar\xP@scaletwo\KK@
814 \xP@tempvar\xP@scaletthree\Direction

```

8.6 Bézier curves

`\splinesolid@` These are the hooks for single-stroke splines (solid, dashed and dotted).
`\splinedashed@` 815 `\xP@hook{curve}{splinesolid@}`
`\splinedotted@` 816 `\newcommand*\xP@splinesolid@{\xP@spline\xP@setsolidpat}`
817 `\xP@hook{curve}{splinedashed@}`
818 `\newcommand*\xP@splinedashed@{\xP@spline\xP@setdashpat}`
819 `\xP@hook{curve}{splinedotted@}`
820 `\newcommand*\xP@splinedotted@{\xP@spline\xP@setdottedpat}`

`\xP@spline` Output a spline segment. Parameter: Macro for the dash pattern generation.

821 `\newcommand*\xP@spline[1]{%`
822 `\readsplineparams@`

Neglect splines which are drawn “backwards”. Somehow Xy-pic draws curves forward and backward, but we need it to be drawn only once.

823 `\ifdim\dimen5<\dimen7`
824 `\xP@preparespline`

Neglect splines of length zero.

825 `\ifdim\@tempdimb>\z@`

Set the dash pattern.

826 `#1%`

Draw the spline.

827 `\xP@stroke{\xP@coor\X@p\Y@p m %`
828 `\xP@coor\L@c\U@c\xP@coor\R@c\D@c\xP@coor\X@c\Y@c c}%`

Record the end point for pattern continuation.

829 `\xP@savec`
830 `\fi`
831 `\fi`
832 `}`

`\xP@preparespline`

833 `\newcommand*\xP@preparespline{%`

If we have a quadratic Bézier segment, convert it to a cubic one.

834 `\ifx\splineinfo@\squineinfo@`
835 `\L@c\dimexpr(\X@p+2\A@)/3\relax`
836 `\U@c\dimexpr(\Y@p+2\B@)/3\relax`
837 `\R@c\dimexpr(\X@c+2\A@)/3\relax`
838 `\D@c\dimexpr(\Y@c+2\B@)/3\relax`
839 `\fi`

Cut the spline according to that start and end parameters in `\dimen5` and `\dimen7`.

840 `\xP@shavespline`

Determine the spline length (for the pattern generation; unnecessary for solid splines).

841 `\xP@bezierlength`
842 `}`

`\xP@inibigdim` •1 Initialize `\xP@bigdim` every time a macro that uses this register is called. See e.g. `\xP@shaveprec`.

843 `\newcommand*\xP@inibigdim{\xP@bigdim5040pt}`

`\xP@shavespline` Shave a cubic spline at both ends at the parameter values in `\dimen5` and `\dimen7`. For normal use, the parameters fulfill $0pt \leq \dimen5 < \dimen7 \leq 1pt$.

(Note that `\xP@bigdim` only occurs in the arguments to `\xP@shaveprec`, so this use is safe.)

```
844 \newcommand*\xP@shavespline{%
845   \xP@shaveprec{\dimen5*\xP@bigdim/\p@}{\dimen7*\xP@bigdim/\p@}%
846 }
```

`\xP@shaveprec` **•1** Shave a cubic spline at both ends at the parameter values in #1 and #2. For normal use, the parameters fulfill $0pt \leq \dimen1 < \dimen2 \leq \dimenbigdim$. The control points for the cubic Bézier curve are $(X@p, Y@p)$, $(L@c, U@c)$, $(R@c, D@c)$, $(X@c, Y@c)$. The `X`-pic registers `A@`, `B@`, `L@p`, `U@p`, `R@p`, `D@p`, `X@min` and `Y@min` are used as temporary registers, but safely encapsulated in a group.

```
847 \newcommand*\xP@shaveprec[2]{%
848   \xP@inibigdim
849   \A@{\dimexpr#1\relax
850   \B@{\dimexpr#2\relax

Shortcut in case the spline is not changed.

851   \@tempswatrue
852   \ifdim\A@=\z@ \ifdim\B@=\xP@bigdim \@tempswafalse \fi \fi
853   \if@tempswa
854     \L@p{\dimexpr\L@c-\X@p\relax
855     \U@p{\dimexpr\R@c-\L@p-\L@c\relax
856     \R@p{\dimexpr\X@c-3\R@c+3\L@c-\X@p\relax
857     \D@p{\dimexpr\U@c-\Y@p\relax
858     \X@min{\dimexpr\D@c-\D@p-\U@c\relax
859     \Y@min{\dimexpr\Y@c-3\D@c+3\U@c-\Y@p\relax
860     \xdef\@gtempa{%
861       \X@p\the\dimexpr\X@p+(3\L@p+(3\U@p+R@p*\A@/\xP@bigdim)%
862       *\A@/\xP@bigdim)*\A@/\xP@bigdim\relax
863       \Y@p\the\dimexpr\Y@p+(3\D@p+(3\X@min+\Y@min*\A@/\xP@bigdim)%
864       *\A@/\xP@bigdim)*\A@/\xP@bigdim\relax
865       \L@c\the\dimexpr\X@p+(2\A@+\B@)*\L@p/\xP@bigdim+((\A@+2\B@)%
866       *\U@p/\xP@bigdim+R@p*\A@/\xP@bigdim*\B@/\xP@bigdim)%
867       *\A@/\xP@bigdim\relax
868       \U@c\the\dimexpr\Y@p+(2\A@+\B@)*\D@p/\xP@bigdim+((\A@+2\B@)%
869       *\X@min/\xP@bigdim+\Y@min*\A@/\xP@bigdim*\B@/\xP@bigdim)%
870       *\A@/\xP@bigdim\relax
871       \R@c\the\dimexpr\X@p+(2\B@+\A@)*\L@p/\xP@bigdim+((\B@+2\A@)%
872       *\U@p/\xP@bigdim+R@p*\B@/\xP@bigdim*\A@/\xP@bigdim)%
873       *\B@/\xP@bigdim\relax
874       \D@c\the\dimexpr\Y@p+(2\B@+\A@)*\D@p/\xP@bigdim+((\B@+2\A@)%
875       *\X@min/\xP@bigdim+\Y@min*\B@/\xP@bigdim*\A@/\xP@bigdim)%
876       *\B@/\xP@bigdim\relax
877       \X@c\the\dimexpr\X@p+(3\L@p+(3\U@p+R@p*\B@/\xP@bigdim)%
878       *\B@/\xP@bigdim)*\B@/\xP@bigdim\relax
879       \Y@c\the\dimexpr\Y@p+(3\D@p+(3\X@min+\Y@min*\B@/\xP@bigdim)%
880       *\B@/\xP@bigdim)*\B@/\xP@bigdim\relax}%
881   \else
882     \global\let\@gtempa\relax
883   \fi
884 } \@gtempa
885 }
```

`\xP@bezierlength` **•1 •2** Compute the arc length of a cubic Bézier segment.

The following algorithm is used: The velocity for a partial segment is fitted at three points (A-C-E) by a quadratic function, and the arc length is approximated by the integral over this quadratic function.

Each interval is recursively divided in halves (A-B-C, C-D-E) as long as the result for the arc length changes more than the precision parameter `\xP@tolerance`. If the desired precision is reached, the arc length in the small interval is added to the total arc length, and the next interval is considered.

The result goes into `\@tempdimb`.

```
886 \newcommand*\xP@bezierlength{%
887   \xP@inibigdim
888   \@tempdimb\z@
889   \xP@parA\z@
890   \xP@velocity\z@\xP@velA
891   \xP@parC.5\xP@bigdim
892   \xP@velocity\xP@parC\xP@velC
893   \xP@velocity\xP@bigdim\xP@velE
```

Arc length (integral over the quadratic approximation)

```
894   \xP@oldpartlen\dimexpr(\xP@velA+4\xP@velC+\xP@velE)/6\relax
```

Tolerance parameter: It is set to 1/100000 of the approximate arc length, but at least 1sp.

```
895   \xP@tolerance\xP@max{1sp}{\dimexpr\xP@oldpartlen/100000\relax}%
```

Initiate the recursive algorithm with the interval [0, 1].

```
896   \xP@arclength\xP@parC\xP@velC\xP@bigdim\xP@velE\xP@oldpartlen
```

Pass the result to outside the group.

```
897   \global\dimen@i\@tempdimb
898 } \@tempdimb\dimen@i
899 }
```

`\xP@velocity` **•1** Compute the velocity at the point #1 on a cubic Bézier curve. Needs: Bézier control points `\X@p,...,\Y@c`. Parameter #2: dimension register for the result. Temporary: `\L@p, \U@p, \d@X, \d@Y`.

```
900 \newcommand*\xP@velocity[2]{%
901   \@tempdima\dimexpr#1\relax
902   \xP@tangent
903   \global\dimen@i\@tempdimb
904 }#2\dimen@i
905 }
```

`\xP@tangent` **•1**

```
906 \newcommand*\xP@tangent{%
907   \d@X3\xP@precbeziertan\X@p\L@c\R@c\X@c\@tempdima
908   \d@Y3\xP@precbeziertan\Y@p\U@c\D@c\Y@c\@tempdima
909   \xP@vecclen
910 }
```

`\xP@tangentvec` **•1** Tangent vector on a Bézier curve. Parameter #1: Parameter on the segment. Needs: Bézier parameters `\X@p,...,\Y@c`. Returns: vector in `(\d@X,\d@Y)`, norm in `\@tempdimb`.

```
911 \newcommand*\xP@tangentvec[1]{%
912   \@tempdima#1\relax
913   \xP@tangent
```

If the velocity is zero at some point, take the second derivative for the tangent vector.

```
914   \ifdim\@tempdimb=\z@
915     \L@p\dimexpr\X@c-\X@p+(\L@c-\R@c)*3\relax
```

```

916      \U@p\dimexpr\Y@c-\Y@p+(\U@c-\D@c)*3\relax
917      \d@X\dimexpr\L@p*\@tempdima/\xP@bigdim+(\X@p-2\L@c+\R@c)\relax
918      \d@Y\dimexpr\U@p*\@tempdima/\xP@bigdim+(\Y@p-2\U@c+\D@c)\relax
919      \xP@vecLen

```

Or even the third derivative.

```

920      \ifdim\@tempdimb=\z@
921      \d@X\L@p
922      \d@Y\U@p
923      \xP@vecLen
924      \ifdim\@tempdimb=\z@
925      \xP@warning{xyPDF}{Cannot determine a tangent vector to a curve}%
926      \@tempdimb\p@
927      \fi
928      \fi
929      \fi
930      \global\dimen@i\d@X
931      \global\dimen3\d@Y
932      \global\dimen5\@tempdimb
933      }%
934      \d@X\dimen@i
935      \d@Y\dimen3\relax
936      \@tempdimb\dimen5\relax
937      }

```

`\xP@arclength` ●2 The recursive step for the arc length computation.

Needs: `\xP@tolerance`, `\xP@parA`, `\xP@velA`. Parameter: #1 is the middle parameter, #2 the velocity at #1, #3 the third parameter, #4 the velocity at #3, #5 the approximate arc length in the interval from `\xP@parA` to #3.

```

938 \newcommand*\xP@arclength[5]{%
939   \xP@parE#3%
940   \xP@velE#4%
941   \xP@parC#1%
942   \xP@velC#2%
943   \xP@oldpartlen#5%

```

Compute two more pairs (parameter, velocity) at positions $\frac{1}{4}$ and $\frac{3}{4}$ of the interval.

```

944   \xP@parB\dimexpr(\xP@parC+\xP@parA)/2\relax
945   \xP@velocity\xP@parB\xP@velB
946   \xP@parD\dimexpr(\xP@parE+\xP@parC)/2\relax
947   \xP@velocity\xP@parD\xP@velD

```

Compute the approximations for the arc length on the two smaller parameter intervals (A-B-C) and (C-D-E).

```

948   \xP@lenA
949   \dimexpr(\xP@velA+4\xP@velB+\xP@velC)/6*(\xP@parC-\xP@parA)/\xP@bigdim\relax
950   \xP@lenB
951   \dimexpr(\xP@velC+4\xP@velD+\xP@velE)/6*(\xP@parE-\xP@parC)/\xP@bigdim\relax
952   \xP@partlen\dimexpr\xP@lenA+\xP@lenB\relax

```

Check whether the approximation for the arc length has changed more than the precision parameter. The code is a hack to compare the absolute value without occupying another dimension register.

```

953   {\@tempdima\dimexpr\xP@oldpartlen-\xP@partlen\relax
954   \expandafter}\ifdim\ifdim\@tempdima<\z@-\fi\@tempdima>\xP@tolerance

```

Yes? Subdivide the interval. The input queue serves as a LIFO stack here!

```

955   \edef\next@{%

```



```

956      \noexpand\xP@arclength\xP@parB\xP@velB\xP@parC\xP@velC\xP@lenA
957      \noexpand\xP@arclength{\the\xP@parD}{\the\xP@velD}{\the\xP@parE}%
958      {\the\xP@velE}{\the\xP@lenB}%
959    }%
960  \else
No? Proceed to the next parameter interval.
961    \xP@parA\xP@parE
962    \xP@velA\xP@velE
963    \advance\@tempdimb\xP@partlen
964    \DN@{}%
965  \fi
966  \next@
967 }

```

8.7 New improved curve styles

\@crv@ Extend the list of curve styles for which special routines exist. New curve styles: 3{.}, \xP@@crv@ {~}, 2{~}, 3{~}, {~~}, 2{~~}, 3{~~}

```

968 \xP@hook{curve}{@crv@}
969 \newcommand*\xP@@crv@[2]{\DN@{#1#2}%
970   \ifx\next@ \@empty \edef\next@{\crv@defaultshape}%
971   \ifx\bstartPLACE@ \@empty \xdef\crvSTYLE@{\crv@defaultshape}\fi
972   \else
973   \ifx\bstartPLACE@ \@empty \gdef\crvSTYLE@{#1#2}\fi
974   \fi
975   \ifx\next@ \@empty \crv@noobject \DN@{\crv@}{\xy@crvaddstack@}%
976   \else \def\tmp@{-}\ifx\next@\tmp@ \DN@{\crv@}{\xy@crvaddstack@}%
977   \else \def\tmp@{=}\ifx\next@\tmp@
978   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{=}}}%
979   \else \def\tmp@{2-}\ifx\next@\tmp@
980   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{2.}}}%
981   \else \def\tmp@{3-}\ifx\next@\tmp@
982   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{3.}}}%
983   \else \def\tmp@{--}\ifx\next@\tmp@
984   \DN@{\expandafter\crv@\crv@specialtemplate{--}}%
985   \else \def\tmp@{==}\ifx\next@\tmp@
986   \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{--}}}%
987   \else \def\tmp@{2--}\ifx\next@\tmp@
988   \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{--}}}%
989   \else \def\tmp@{3--}\ifx\next@\tmp@
990   \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{--}}}%
991   \else \def\tmp@{.}\ifx\next@\tmp@
992   \DN@{\expandafter\crv@\crv@specialtemplate{.}}%
993   \else \def\tmp@{:}\ifx\next@\tmp@
994   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{:}}}%
995   \else \def\tmp@{2.}\ifx\next@\tmp@
996   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{:}}}%
997   \else \def\tmp@{3.}\ifx\next@\tmp@
998   \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{.}}}%
999   \else \def\tmp@{~}\ifx\next@\tmp@
1000  \DN@{\expandafter\crv@\crv@normaltemplate{\dir{~}}}%
1001  \else \def\tmp@{2~}\ifx\next@\tmp@
1002  \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{~}}}%
1003  \else \def\tmp@{3~}\ifx\next@\tmp@
1004  \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{~}}}%

```

```

1005 \else\def\tmp@{~}\ifx\next@\tmp@
1006 \DN@{\expandafter\crv@\crv@normaltemplate{\dir{~}}}%
1007 \else\def\tmp@{2~}\ifx\next@\tmp@
1008 \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{~}}}%
1009 \else\def\tmp@{3~}\ifx\next@\tmp@
1010 \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{~}}}%
1011 \else\def\tmp@{.}\ifx\next@\tmp@
1012 \DN@{\expandafter\crv@\crv@specialtemplate@{.}}%
1013 \else
1014 \DN@{\expandafter\crv@\crv@othertemplate{\dir#1{#2}}}%
1015 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\next@

```

```

\sysplinespecialcases@ New: \dir3{.}, \dir2{--}, \dir3{--}, \dir{~}, \dir2{~}, \dir3{~}, \dir{~},
\XP@sysplinespecialcases@ \dir2{~}, \dir3{~}

```

```

1016 \XP@hook{curve}{\sysplinespecialcases@}
1017 \newcommand*\XP@sysplinespecialcases@{%
1018 \ifx\@empty\ycrvdrop@
1019 \ifx\@empty\ycrvconn@\DN@{\splinesolid@}%
1020 \else\DN@{ \dir{-}}\ifx\next@\ycrvconn@\DN@{\splinesolid@}%
1021 \else\DN@{ \dir2{-}}\ifx\next@\ycrvconn@\DN@{\splinedoubled@}%
1022 \else\DN@{ \dir{=}}\ifx\next@\ycrvconn@\DN@{\splineribboned@}%
1023 \else\DN@{ \dir{2.}}\ifx\next@\ycrvconn@\DN@{\splinedoubled@}%
1024 \else\DN@{ \dir3{-}}\ifx\next@\ycrvconn@\DN@{\splinetrebled@}%
1025 \else\DN@{ \dir{3.}}\ifx\next@\ycrvconn@\DN@{\splinetrebled@}%
1026 \else\DN@{ \dir{--}}\ifx\next@\ycrvconn@\DN@{\splinedashed@}%
1027 \else\DN@{ \dir{.}}\ifx\next@\ycrvconn@\DN@{\splinedotted@}%
1028 \else\DN@{ \dir{:}}\ifx\next@\ycrvconn@\DN@{\splinedbldotted@}%

```

The next line does not occur in X_Y-pic for an unknown reason. However, it seems reasonable to define the special pattern `\dir2{.}` in the same way as for straight lines.

```

1029 \else\DN@{ \dir2{.}}\ifx\next@\ycrvconn@\DN@{\splinedbldotted@}%
1030 \else\DN@{ \dir3{.}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrbldotted}%
1031 \else\DN@{ \dir2{--}}\ifx\next@\ycrvconn@\DN@{\XP@splinedbldashed}%
1032 \else\DN@{ \dir3{--}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrbldashed}%
1033 \else\DN@{ \dir{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinesquiggled}%
1034 \else\DN@{ \dir2{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinedblsquiggled}%
1035 \else\DN@{ \dir3{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrblsquiggled}%
1036 \else\DN@{ \dir{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokensquiggled}%
1037 \else\DN@{ \dir2{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokendblsquiggled}%
1038 \else\DN@{ \dir3{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokentrblsquiggled}%
1039 \else\ifdim\splinetol@>z@\else\splinedefaulttol@fi
1040 \DN@{\splineset@}\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
1041 \else
1042 \DN@{\splineset@}%
1043 \fi
1044 \ifInvisible@\DN@{}\fi
1045 \next@
1046 }

```

8.8 Multiple solid curves

```
\XP@splinedoubled@
```

```

1047 \XP@hook{curve}{\splinedoubled@}
1048 \newcommand*\XP@splinedoubled@{%
1049 \XP@checkspline\XP@splinemultsolid\XP@doublestroke}

```

```

\xP@splineribboned@
1050 \xP@hook{curve}{splineribboned@}
1051 \@ifdefinable\xP@splineribboned@\relax
1052 \let\xP@splineribboned@\xP@splinedoubled@

\xP@splinetrebled@
1053 \xP@hook{curve}{splinetrebled@}
1054 \newcommand*\xP@splinetrebled@{%
1055   \xP@checkspline\xP@splinemultsolid\xP@trblstroke}

\xP@doublestroke   Offset parameters for double lines and curves
1056 \newcommand*\xP@doublestroke{\xydashh@/2,-\xydashh@/2}

\xP@trblstroke     Offset parameters for treble lines and curves
1057 \newcommand*\xP@trblstroke{\xydashh@,\z@,-\xydashh@}

\xP@checkspline    Get and check spline parameters before the macro in #1 is executed.
1058 \newcommand*\xP@checkspline[1]{%
1059   \readsplineparams@
   Neglect splines which are drawn “backwards”. Somehow Xy-pic draws curves forward and
   backward, but we need it to be drawn only once.
1060   \let\next@\@gobble
1061   \ifdim\dimen5<\dimen7
1062     \xP@preparespline
   Neglect splines of zero length.
1063     \ifdim\@tempdimb>\z@
   If the path length is less than twice the line width, just draw a solid path.
1064       \ifdim\@tempdimb<2\dimexpr\xP@preclw\relax
1065         \let\next@\xP@splinemultsolid
1066       \else
1067         \let\next@#1%
1068       \fi
1069     \fi
1070   \fi
1071   \next@
1072 }

\xP@splinemultsolid •1
1073 \newcommand*\xP@splinemultsolid[1]{%
1074   \xP@inibigdim
1075   \@temptokena{}%
1076   \xP@setsolidpat
   The \@for loop does the multiple strokes. \@tempa records the respective offset distance.
1077   \@for\@tempa:={#1}\do{\xP@paintsolid\z@\xP@bigdim}%
1078   \xP@stroke{\the\@temptokena}%
1079 }

\xP@paintsolid •1 •5 Draw a solid spline in the parameter interval  $[#1, #2] \subseteq [0pt, \xP@bigdim]$  with a
certain offset. The offset distance is expected in \@tempa.
1080 \newcommand*\xP@paintsolid[2]{%

```

Record the original anchor points.

```

1081 \xP@savepts
1082 \xP@a#1\relax
1083 \xP@c#2\relax
1084 \xP@movetotrue
1085 \xP@paintsolid@
1086 \xdef\@gtempa{\the\@temptokena}%
1087 }%
1088 \@temptokena\expandafter{\@gtempa}%
1089 }

```

\xP@paintsolid@ ●1●5

```

1090 \newcommand*\xP@paintsolid@{%

```

These parameters record which part of the spline is currently being offset. They are varied as the spline may be subdivided for a precise offset curve.

```

1091 \xP@b\xP@c
Offset distance
1092 \xP@off\dimexpr\@tempa\relax
1093 \ifdim\xP@off=\z@
1094 \xP@shaveprec\xP@a\xP@c
1095 \else
1096 \loop

```

Restore the original anchor points.

```

1097 \xP@restorepts

```

Compute the approximate offset curve. Note that \xP@a and \xP@b contain the boundary parameters for the partial spline.

```

1098 \xP@offsetsegment
Test if the offset curve is good enough.
1099 \xP@testoffset
If not, shorten the parameter interval by 30%.
1100 \unless\ifxP@offsetok
1101 \xP@b\dimexpr\xP@a+(\xP@b-\xP@a)*7/10\relax
1102 \repeat
1103 \fi

```

Append the new segment to the path.

```

1104 \xP@append\@temptokena{\ifxP@moveto\xP@coor\X@p\Y@p m \fi
1105 \xP@coor\L@c\U@c\xP@coor\R@c\D@c\xP@coor\X@c\Y@c c }%
1106 \xP@movetofalse

```

Test if the end of the spline has been reached. If not, offset the rest of the curve.

```

1107 \ifdim\xP@b<\xP@c\relax
1108 \xP@a\xP@b
1109 \expandafter\xP@paintsolid@
1110 \fi
1111 }

```

\ifxP@moveto We need a PDF moveto operator only for the first partial segment. Additional segments connect seamlessly.

```

1112 \@ifdefinable\ifxP@moveto\relax
1113 \@ifdefinable\xP@movetotrue\relax
1114 \@ifdefinable\xP@movetofalse\relax
1115 \newif\ifxP@moveto

```

`\xP@savepts` •5 Save the anchor points to the second set of reserved variables.

```

1116 \newcommand*\xP@savepts{%
1117   \xP@xa\xP@p
1118   \xP@ya\Y@p
1119   \xP@xb\L@c
1120   \xP@yb\U@c
1121   \xP@xc\R@c
1122   \xP@yc\D@c
1123   \xP@xd\X@c
1124   \xP@yd\Y@c
1125 }

```

`\xP@restorepts` •5 Restore the anchor points from the second set of reserved variables.

```

1126 \newcommand*\xP@restorepts{%
1127   \X@p\xP@xa
1128   \Y@p\xP@ya
1129   \L@c\xP@xb
1130   \U@c\xP@yb
1131   \R@c\xP@xc
1132   \D@c\xP@yc
1133   \X@c\xP@xd
1134   \Y@c\xP@yd
1135 }

```

8.9 A Bézier curve offset algorithm

First, all control points are offset by the desired distance and in the direction of the normal vectors at the boundary points of the curve. We then adjust the distance of the inner two control points to the boundary control points along the tangents at the boundary points: $x_b = x_a + f_a T_{ax}$, $x_c = x_d + f_d T_{dx}$, and likewise for the y -coordinates. In nondegenerate cases, we have $T_{ax} = x_b - x_a$ and $T_{dx} = x_c - x_d$.

Let $P(a, b, c, d, t)$ denote the Bézier polynomial $a(1-t)^3 + 3bt(1-t)^2 + 3ct^2(1-t) + dt^3$. In order to determine the factors f_a and f_d , we set up a system of three equations.

- Two equations: The old point at parameter $\frac{1}{2}$ plus offset, (x_m, y_m) , is the new point at parameter t_m .

$$\begin{aligned}
 x_m &= P(x_a, x_a + f_a T_{ax}, x_d + f_d T_{dx}, x_d, t_m) \\
 y_m &= P(y_a, y_a + f_a T_{ay}, y_d + f_d T_{dy}, y_d, t_m)
 \end{aligned}$$

- Third equation: The old tangent at parameter $\frac{1}{2}$ is in the same direction as the new tangent at t_m .

$$\begin{aligned}
 &\frac{\partial}{\partial t_m} P(x_a, x_a + f_a T_{ax}, x_d + f_d T_{dx}, x_d, t_m) \cdot T_{my} \\
 &= \frac{\partial}{\partial t_m} P(y_a, y_a + f_a T_{ay}, y_d + f_d T_{dy}, y_d, t_m) \cdot T_{mx}
 \end{aligned}$$

Up to a scalar factor of $-3/4$, (T_{mx}, T_{my}) is the velocity vector to the original curve at parameter $\frac{1}{2}$. We have $T_{mx} = (X_a + X_b - X_c - X_d)/2$ (in the old coordinates!) and T_{my} analogously. The system above is a nonlinear system of three equations in three variables, which we solve by Newton's method. Let f_a , f_d , and t_m be approximate solutions, and denote by Δf_a , Δf_d , and Δt_m the increments to the next approximation. In the first order,

the three equations become:

$$\begin{aligned}
x_m &= P(x_a, x_b, x_c, x_d, t_m) + \Delta f_a \cdot T_{ax} \cdot 3t_m(1 - t_m)^2 + \Delta f_d \cdot T_{dx} \cdot 3t_m^2(1 - t_m) \\
&\quad + \Delta t_m \frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) \\
y_m &= P(y_a, y_b, y_c, y_d, t_m) + \Delta f_a \cdot T_{ay} \cdot 3t_m(1 - t_m)^2 + \Delta f_d \cdot T_{dy} \cdot 3t_m^2(1 - t_m) \\
&\quad + \Delta t_m \frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) \\
&\quad \left(\frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) + \Delta f_a \cdot T_{ax} \cdot 3(1 - 4t_m + 3t_m^2) + \Delta f_d \cdot T_{dx} \cdot 3(2t_m - 3t_m^2) \right. \\
&\quad \left. + \Delta t_m \cdot 6((x_a - 2x_b + x_c) + t_m(x_d - x_a + 3(x_b - x_c))) \right) \cdot T_{my} \\
&= \left(\frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) + \Delta f_a \cdot T_{ay} \cdot 3(1 - 4t_m + 3t_m^2) + \Delta f_d \cdot T_{dy} \cdot 3(2t_m - 3t_m^2) \right. \\
&\quad \left. + \Delta t_m \cdot 6((y_a - 2y_b + y_c) + t_m(y_d - y_a + 3(y_b - y_c))) \right) \cdot T_{mx}
\end{aligned}$$

Rewrite the equations so that they resemble the \TeX code.

$$\begin{aligned}
8P(x_a, x_b, x_c, x_d, t_m) - 8x_m &= -\Delta f_a \cdot 3T_{ax} \cdot 2t_m \cdot (2(1 - t_m))^2 \\
&\quad - \Delta f_d \cdot 3T_{dx} \cdot 4t_m^2 \cdot 2(1 - t_m) - \Delta t_m \cdot 8 \frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) \\
8P(y_a, y_b, y_c, y_d, t_m) - 8y_m &= -\Delta f_a \cdot 3T_{ay} \cdot 2t_m \cdot (2(1 - t_m))^2 \\
&\quad - \Delta f_d \cdot 3T_{dy} \cdot 4t_m^2 \cdot 2(1 - t_m) - \Delta t_m \cdot 8 \frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) \\
T_{mx} \cdot 8 \frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) - T_{my} \cdot 8 \frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) \\
&= -\Delta f_a \cdot (3T_{ay} \cdot 2T_{mx} - 3T_{ax} \cdot 2T_{my}) \cdot 2(1 - 3t_m) \cdot 2(1 - t_m) \\
&\quad - \Delta f_d \cdot (3T_{dy} \cdot 2T_{mx} - 3T_{dx} \cdot 2T_{my}) \cdot 2(2 - 3t_m) \cdot 2t_m \\
&\quad - \Delta t_m \cdot (((y_d - y_a + 3(y_b - y_c)) \cdot 2t_m + 2(y_a - 2y_b + y_c)) \cdot 3 \cdot 8T_{mx} \\
&\quad - ((x_d - x_a + 3(x_b - x_c)) \cdot 2t_m + 2(x_a - 2x_b + x_c)) \cdot 3 \cdot 8T_{my})
\end{aligned}$$

Substitute $2t_m = \tau_m$.

$$\begin{aligned}
8P(x_a, x_b, x_c, x_d, t_m) - 8x_m &= -\Delta f_a \cdot 3T_{ax} \cdot \tau_m(2 - \tau_m)^2 \\
&\quad - \Delta f_d \cdot 3T_{dx} \cdot \tau_m^2(2 - \tau_m) - \frac{1}{2}\Delta\tau_m \cdot 8 \frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) \\
8P(y_a, y_b, y_c, y_d, t_m) - 8y_m &= -\Delta f_a \cdot 3T_{ay} \cdot \tau_m \cdot (2 - \tau_m)^2 \\
&\quad - \Delta f_d \cdot 3T_{dy} \cdot \tau_m^2(2 - \tau_m) - \frac{1}{2}\Delta\tau_m \cdot 8 \frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) \\
T_{mx} \cdot 8 \frac{\partial}{\partial t_m} P(y_a, y_b, y_c, y_d, t_m) - T_{my} \cdot 8 \frac{\partial}{\partial t_m} P(x_a, x_b, x_c, x_d, t_m) \\
&= -\Delta f_a \cdot (3T_{ay} \cdot 2T_{mx} - 3T_{ax} \cdot 2T_{my}) \cdot (2 - 3\tau_m)(2 - \tau_m) \\
&\quad - \Delta f_d \cdot (3T_{dy} \cdot 2T_{mx} - 3T_{dx} \cdot 2T_{my}) \cdot (4 - 3\tau_m)\tau_m \\
&\quad - \Delta\tau_m \cdot (((y_d - y_a + 3(y_b - y_c)) \cdot \tau_m + 2(y_a - 2y_b + y_c)) \cdot 12T_{mx} \\
&\quad - ((x_d - x_a + 3(x_b - x_c)) \cdot \tau_m + 2(x_a - 2x_b + x_c)) \cdot 12T_{my})
\end{aligned}$$

The translation into \TeX dimensions:

- $f_a = \text{\xp@fa}$, $f_d = \text{\xp@fd}$
- $\tau_m = \text{\xp@tm}$
- $x_a = \text{\xp@xa}, \dots, x_d = \text{\xp@xd}, \dots, y_d = \text{\xp@yd}$

- $8P(x_1, x_2, x_3, x_4, \frac{1}{2}x_5) = \text{\xP@bezierpoly\#1\#2\#3\#4\#5}$
- $8x_m = \text{\xP@xm}$, $8y_m = \text{\xP@ym}$
- $3T_{ax} = \text{\xP@Tax}$, $3T_{dx} = \text{\xP@Tdx}$, $3T_{ay} = \text{\xP@Tay}$, $3T_{dy} = \text{\xP@Tdy}$
- $8\frac{\partial}{\partial x_5}P(x_1, x_2, x_3, x_4, \frac{1}{2}x_5) = \text{\xP@beziertan\#1\#2\#3\#4\#5}$

Temporary:

- $2 - \tau_m = \text{\xP@ta}$
- $\tau_m(2 - \tau_m) = \text{\xP@tb}$
- $T_{mx} = \text{\xP@Tmx}$, $T_{my} = \text{\xP@Tmy}$
- $2 - 3\tau_m = \text{\xP@tb}$
- $4 - 3\tau_m = \text{\xP@tc}$

Since the linear system above tends to be singular or ill-conditioned (think about the frequent case when all control points are nearly collinear!), the Gauss algorithm `\xP@solvelinearsystem` does not always return a valid solution. In these cases, the system is not solved exactly but approximated iteratively in `\xP@applinsys`.

```
\xP@tmx
\xP@tmy 1136 \ifdefinable\xP@tmx\relax
1137 \ifdefinable\xP@tmy\relax
```

```
\xP@Tmxy ●4
\xP@Tmyx 1138 \newcommand*\xP@Tmxy{* \xP@Tmx / \xP@Tmy}
1139 \newcommand*\xP@Tmyx{* \xP@Tmy / \xP@Tmx}
```

```
\xP@Tmzero
1140 \newcommand*\xP@Tmzero{* \z@}
```

`\xP@offsetsegment` ●1 ●3 ●4 Offset a cubic segment. The offset distance is given in `\xP@off`. The anchor points are given in `\X@p, ..., \Y@c`. The partial spline in the parameter interval $[\text{\xP@a}, \text{\xP@b}] \subseteq [0pt, \text{\xP@bigdim}]$ is offset. The new Bézier curve is returned in `\xP@xa, ..., \xP@yd`.

```
1141 \newcommand*\xP@offsetsegment{%
  New first anchor point and tangent vector at 0
1142 \xP@tangentvec\xP@a
1143 \xP@xa\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@a/8%
1144 +\d@Y*\xP@off/\@tempdimb\relax
1145 \xP@ya\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@a/8%
1146 -\d@X*\xP@off/\@tempdimb\relax
1147 \xP@scaleT
1148 \xP@Tax\d@X
1149 \xP@Tay\d@Y
1150 \xP@E\@tempdimb
  New last anchor point and tangent vector at 1
1151 \xP@tangentvec\xP@b
1152 \xP@xd\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@b/8%
1153 +\d@Y*\xP@off/\@tempdimb\relax
1154 \xP@yd\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@b/8%
1155 -\d@X*\xP@off/\@tempdimb\relax
1156 \xP@scaleT
```

```

1157 \xP@Tdx-\d@X
1158 \xP@Tdy-\d@Y
1159 \xP@F\@tempdimb

```

Scalar product of the tangent vectors

```

1160 \xP@M\z@
1161 \xP@Max\xP@M\xP@Tdx
1162 \xP@Max\xP@M\xP@Tdy
1163 \xP@L\dimexpr\xP@Tax*\xP@Tdx/\xP@M+\xP@Tay*\xP@Tdy/\xP@M\relax
1164 \xP@tm\dimexpr(\xP@a+\xP@b)/2\relax
1165 \ifdim\xP@L>\dimexpr\xP@E*\xP@F/\xP@M*49/50\relax

```

Trick to improve the offset algorithm near sharp bends and cusps: If the tangent vectors (T_{ax}, T_{ay}) and (T_{dx}, T_{dy}) point nearly in the same direction, we do not use the true tangent vector for (T_{mx}, T_{my}) at the middle point but a fake one. (The exact condition is that their normed scalar product is greater than $49/50$. For a straight line, the vectors would point in opposite directions.) The fake tangent vector is defined to be $(T_{ax} + T_{dx}, T_{ay} + T_{dy})$ rotated by $\pm 90^\circ$. Its direction is chosen such that the scalar product with $(X_d - X_a, Y_d - Y_a)$ is nonnegative. (Use $(X_c - X_b, Y_c - Y_b)$ in the degenerate case $(X_d - X_a, Y_d - Y_a) = (0, 0)$.)

Rationale: In the presence of a sharp bend or cusp, the offset algorithm will hardly meet the tip. Since the tangent/normal at the tip is needed for a good offset curve, we provide this artificially.

```

1166 \d@X-\dimexpr\xP@Tay+\xP@Tdy\relax
1167 \d@Y\dimexpr\xP@Tax+\xP@Tdx\relax
1168 \xP@vecLen
1169 \xP@A\dimexpr\X@c-\X@p\relax
1170 \xP@B\dimexpr\Y@c-\Y@p\relax
1171 \xP@M\z@
1172 \xP@Max\xP@M\xP@A
1173 \xP@Max\xP@M\xP@B
1174 \ifdim\xP@M=\z@
1175 \xP@A\dimexpr\R@c-\L@c\relax
1176 \xP@B\dimexpr\D@c-\U@c\relax
1177 \xP@Max\xP@M\xP@A
1178 \xP@Max\xP@M\xP@B
1179 \fi
1180 \xP@M\dimexpr\d@X*\xP@A/\xP@M+\d@Y*\xP@B/\xP@M\relax
1181 \ifdim\xP@M<\z@
1182 \multiply\d@X\m@ne
1183 \multiply\d@Y\m@ne
1184 \fi
1185 \else

```

Normal case: tangent vector at the middle point.

```

1186 \xP@tangentvec\xP@tm
1187 \fi

```

From here on, $\xP@a$ and $\xP@b$ will not be used any more, so these variables can be used under their other names $\xP@I$, $\xP@J$ for the linear systems below.

8 times (middle point plus offset)

```

1188 \xP@xm\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@tm%
1189 +8\d@Y*\xP@off/\@tempdimb\relax
1190 \xP@ym\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@tm%
1191 -8\d@X*\xP@off/\@tempdimb\relax

```

Tangent at middle point

```

1192 \xP@Tmx\d@X

```



```

1193 \xP@Tmy\d@Y
1194 \xP@ifabsless\xP@Tmy\xP@Tmx
1195 \let\xP@tmy\xP@Tmyx
1196 \let\xP@tmx\@empty
1197 \else
1198 \ifdim\xP@Tmy=\z@
1199 \let\xP@tmx\xP@Tmzero
1200 \let\xP@tmy\xP@Tmzero
1201 \else
1202 \let\xP@tmy\@empty
1203 \let\xP@tmx\xP@Tmxy
1204 \fi
1205 \fi

Initial guesses for the tangent vector scalings \xP@fa, \xP@fd and the near-middle position
\xP@tm

1206 \xP@fa\p@
1207 \xP@fd\p@
1208 \xP@tm\p@

The main loop for finding the offset curve

1209 \count@\z@
1210 \loop

Set the new control points up.

1211 \xP@offsetpoints
1212 \@tempswafalse

At most 10 iterations

1213 \ifnum10>\count@

Determine the quality of the approximation by an objective function.

1214 \xP@objfun\xP@oldobj
1215 \ifdim\xP@oldobj>\xP@maxobjfun\relax\@tempswatrue\fi
1216 \fi
1217 \if@tempswa
1218 \xP@offsetloop
1219 \repeat

Return the new anchor points.

1220 \xdef\@gtempa{\X@p\the\xP@xa\Y@p\the\xP@ya
1221 \L@c\the\xP@xb\U@c\the\xP@yb\R@c\the\xP@xc\D@c\the\xP@yc
1222 \X@c\the\xP@xd\Y@c\the\xP@yd\relax}%
1223 }%
1224 \@gtempa
1225 }

```

\xP@scaleT ●134 This macro contains another trick to improve the offset algorithm around sharp bends and cusps. It adjusts the length of the tangent/velocity vectors. Let $(\text{\texttt{\textbackslash d@X}}, \text{\texttt{\textbackslash d@Y}})$ be the velocity vector to the original curve at some point with velocity v_0 . The velocity at the same point, considered on a partial segment scales linearly with the length of the parameter interval. Hence, the velocity v_1 in the partial segment is $v_1 = v_0 \cdot (\text{\texttt{\textbackslash xP@b}} - \text{\texttt{\textbackslash xP@a}}) / \text{\texttt{\textbackslash xP@bigdim}}$. Additionally the offset curve goes with a radius of $r + \text{\texttt{\textbackslash xP@off}}$ around bends with radius r in the original curve. As an approximation to the velocity in the offset curve, we therefore scale the velocity vector in the end to the norm $v_1 + 2\pi \cdot |\text{\texttt{\textbackslash xP@off}}|$.

```

1226 \newcommand*\xP@scaleT{%
1227 \xP@B6.28\xP@off
1228 \xP@abs\xP@B

```

```

1229 \xP@C\dimexpr\d@X*\xP@B/\@tempdimb\relax
1230 \xP@D\dimexpr\d@Y*\xP@B/\@tempdimb\relax
1231 \xP@A\dimexpr\xP@b-\xP@a\relax
1232 \d@X\dimexpr\xP@C+\d@X*\xP@A/\xP@bigdim\relax
1233 \d@Y\dimexpr\xP@D+\d@Y*\xP@A/\xP@bigdim\relax

```

Also record the change to the norm of the vector.

```

1234 \@tempdimb\dimexpr\xP@B+\@tempdimb*\xP@A/\xP@bigdim\relax
1235 }

```

\xP@offsetloop ●1 ●3 ●4 The iteration in the offset loop: set up and solve (or approximate) the linear system.

```

1236 \newcommand*\xP@offsetloop{%
1237 \xP@C\dimexpr\xP@C/2\relax
1238 \xP@G\dimexpr\xP@G/2\relax

1st linear equation
1239 \xP@ta\dimexpr2\p@-\xP@tm\relax
1240 \xP@tb\dimexpr\xP@tm*\xP@ta/\p@\relax
1241 \xP@A\dimexpr\xP@Tax*\xP@tb/\p@*\xP@ta/\p@\relax
1242 \xP@B\dimexpr\xP@Tdx*\xP@tb/\p@*\xP@tm/\p@\relax

```

2nd linear equation

```

1243 \xP@E\dimexpr\xP@Tay*\xP@tb/\p@*\xP@ta/\p@\relax
1244 \xP@F\dimexpr\xP@Tdy*\xP@tb/\p@*\xP@tm/\p@\relax

```

3rd linear equation

```

1245 \xP@tb\dimexpr2\p@-3\xP@tm\relax
1246 \xP@tc\dimexpr\xP@tb+2\p@\relax
1247 \xP@I\dimexpr(2\xP@Tay\xP@tmx-2\xP@Tax\xP@tmy)*\xP@tb/\p@*\xP@ta/\p@\relax
1248 \xP@J\dimexpr(2\xP@Tdy\xP@tmx-2\xP@Tdx\xP@tmy)*\xP@tc/\p@*\xP@tm/\p@\relax
1249 \xP@K\dimexpr((\xP@yd-\xP@ya+(\xP@yb-\xP@yc)*3)
1250 *\xP@tm/\p@+(\xP@yc-2\xP@yb+\xP@ya)*2)*12\xP@tmx
1251 -((\xP@xd-\xP@xa+(\xP@xb-\xP@xc)*3)
1252 *\xP@tm/\p@+(\xP@xc-2\xP@xb+\xP@xa)*2)*12\xP@tmy\relax

```

Solve the system.

```

1253 \xP@solvelinearsystem
1254 \ifxP@validsol

```

Check whether the result is feasible and whether it actually improves the approximation.

```

1255 \xP@correctsol
1256 \ifdim\xP@ta=z@
1257 \ifdim\xP@tb=z@
1258 \ifdim\xP@tc=z@
1259 \xP@validsolfalse
1260 \fi\fi\fi
1261 \fi

```

If the exact solution is not valid, try to at least approximate a solution.

```

1262 \ifxP@validsol
1263 \else
1264 \xP@applinsys

```

This time, the solution is not checked but applied immediately.

```

1265 \advance\xP@fa-\xP@ta
1266 \advance\xP@fd-\xP@tb
1267 \advance\xP@tm-\xP@tc

```

The near-middle parameter on the curve must not lie outside the segment.

```

1268   \ifdim\xP@tm<\z@\xP@tm\z@\fi
1269   \ifdim\xP@tm>2\p@\xP@tm2\p@\fi
1270   \fi
1271   \advance\count@\@ne
1272 }

```

\xP@maxsol Heuristic: maximal solution so that no arithmetic overflow is produced.

```

1273 \newcommand*\xP@maxsol{3pt}

```

\xP@correctsol ●3 ●4 Check whether the solution is feasible and actually improves the objective function.

```

1274 \newcommand*\xP@correctsol{%
  If the solution is too big, scale all variables uniformly.

1275   \xP@M\z@
1276   \xP@Max\xP@M\xP@ta
1277   \xP@Max\xP@M\xP@tb
1278   \xP@Max\xP@M\xP@tc
1279   \ifdim\xP@M>\xP@maxsol
1280     \xP@ta\dimexpr\xP@maxsol*\xP@ta/\xP@M\relax
1281     \xP@tb\dimexpr\xP@maxsol*\xP@tb/\xP@M\relax
1282     \xP@tc\dimexpr\xP@maxsol*\xP@tc/\xP@M\relax
1283   \fi

```

Apply the solution. Save the old value of \xP@tm to be able to restore it.

```

1284   \advance\xP@fa-\xP@ta
1285   \advance\xP@fd-\xP@tb
1286   \xP@M\xP@tm
1287   \advance\xP@tm-\xP@tc

```

The near-middle parameter must lie on the segment.

```

1288   \ifdim\xP@tm<\z@\xP@tm\z@\fi
1289   \ifdim\xP@tm>2\p@\xP@tm2\p@\fi

```

Check whether the solution actually improves the objective function.

```

1290   {\xP@offsetpoints
1291     \xP@objfun\xP@M
1292   \expandafter}%

```

If not, restore the old values and declare the solution invalid.

```

1293   \ifdim\xP@M>\xP@oldobj
1294     \advance\xP@fa\xP@ta
1295     \advance\xP@fd\xP@tb
1296     \xP@tm\xP@M
1297     \xP@validsolfalse
1298   \fi
1299 }

```

\xP@objfun ●3 ●4 The objective function: sum of squares of the deviation in x - and y -direction and the angular deviation at the middle point. We also compute some terms which will be used in the linear system.

```

1300 \newcommand*\xP@objfun[1]{%
1301   \xP@D\dimexpr\xP@bezierpoly\xP@xa\xP@xb\xP@xc\xP@xd\xP@tm-\xP@xm\relax
1302   \xP@H\dimexpr\xP@bezierpoly\xP@ya\xP@yb\xP@yc\xP@yd\xP@tm-\xP@ym\relax
1303   \xP@C\xP@beziertan\xP@xa\xP@xb\xP@xc\xP@xd\xP@tm
1304   \xP@G\xP@beziertan\xP@ya\xP@yb\xP@yc\xP@yd\xP@tm
1305   \xP@L\dimexpr\xP@G\xP@tmx-\xP@C\xP@tmy\relax

```

If the deviation is too big, let the objective function be `\maxdimen`. Otherwise, compute the sum of squares.

```

1306 #1\z@
1307 \xP@Max#1\xP@D
1308 \xP@Max#1\xP@H
1309 \xP@Max#1\xP@L
1310 #1\ifdim#1>4843165sp
1311 \maxdimen
1312 \else
1313 \dimexpr\xP@D*\xP@D/\p@+\xP@H*\xP@H/\p@+\xP@L*\xP@L/\p@\relax
1314 \fi
1315 }

```

`\xP@offsetpoints` ●3 ●4 Compute the new control points from the factors `\xP@fa`, `\xP@fd`.

```

1316 \newcommand*\xP@offsetpoints{%
1317 \xP@xb\dimexpr\xP@xa+\xP@Tax*\xP@fa/196608\relax
1318 \xP@yb\dimexpr\xP@ya+\xP@Tay*\xP@fa/196608\relax
1319 \xP@xc\dimexpr\xP@xd+\xP@Tdx*\xP@fd/196608\relax
1320 \xP@yc\dimexpr\xP@yd+\xP@Tdy*\xP@fd/196608\relax
1321 }

```

`\xP@bezierpoly` Formula for the polynomial $8(\#1 \cdot (1-t)^3 + 3 \cdot \#2 \cdot t(1-t)^2 + 3 \cdot \#3 \cdot t^2(1-t) + \#4 \cdot t^3)$, $t = \frac{1}{2}\#5$.

```

1322 \newcommand*\xP@bezierpoly[5]{%
1323 \dimexpr(((\#4-\#1+(\#2-\#3)*3)*\#5/\p@+(\#1-2\#2+\#3)*6)*\#5/\p@+(\#2-\#1)*12)*\#5/\p@
1324 +\#1*8\relax
1325 }

```

`\xP@precbezierpoly` Formula for the polynomial $8(\#1 \cdot (1-t)^3 + 3 \cdot \#2 \cdot t(1-t)^2 + 3 \cdot \#3 \cdot t^2(1-t) + \#4 \cdot t^3)$, $t = \#5/\xP@bigdim$.

```

1326 \newcommand*\xP@precbezierpoly[5]{%
1327 \dimexpr(((\#4-\#1+(\#2-\#3)*3)*2*\#5/\xP@bigdim+(\#1-2\#2+\#3)*6)*2*\#5/\xP@bigdim
1328 +(\#2-\#1)*12)*2*\#5/\xP@bigdim+\#1*8\relax
1329 }

```

`\xP@beziertan` Formula for the polynomial

$$24(-\#1 \cdot (1-t)^2 + \#2 \cdot (3t^2 - 4t + 1) + \#3 \cdot (-3t^2 + 2t) + \#4 \cdot t^2), \quad t = \frac{1}{2}\#5.$$

Up to a scalar factor, this is the derivative of the third order Bézier polynomial above.

```

1330 \newcommand*\xP@beziertan[5]{%
1331 \dimexpr((\#4-\#1+(\#2-\#3)*3)*3*\#5/32768+(\#1-2\#2+\#3)*24)*\#5/\p@+(\#2-\#1)*24\relax
1332 }

```

`\xP@precbeziertan` Formula for the polynomial

$$(-\#1 \cdot (1-t)^2 + \#2 \cdot (3t^2 - 4t + 1) + \#3 \cdot (-3t^2 + 2t) + \#4 \cdot t^2), \quad t = \#5/\xP@bigdim.$$

This is $\frac{1}{3}$ times the derivative of the third order Bézier polynomial.

```

1333 \newcommand*\xP@precbeziertan[5]{%
1334 \dimexpr((\#4-\#1+(\#2-\#3)*3)*\#5/\xP@bigdim+(\#1-2\#2+\#3)*2)*\#5/\xP@bigdim
1335 +\#2-\#1\relax
1336 }

```

`\xP@solvelinearsystem` ●3 The macro `\xP@solvelinearsystem` solves a system of three linear equations by the Gauss algorithm. The coefficients and desired values are passed in the extended matrix

$$\left(\begin{array}{ccc|c} \xP@A & \xP@B & \xP@C & \xP@D \\ \xP@E & \xP@F & \xP@G & \xP@H \\ \xP@I & \xP@J & \xP@K & \xP@L \end{array} \right)$$

The solution is returned in the vector $(\xP@ta, \xP@tb, \xP@tc)$.

```

\xP@varone With column swapping in the Gauss algorithm, variable names might be changed. These
\xP@vartwo macros record the variables.
\xP@varthree 1337 \ifdefinable\xP@varone\relax
              1338 \ifdefinable\xP@vartwo\relax
              1339 \ifdefinable\xP@varthree\relax

\ifxP@validsol Records if a valid solution to the linear system is returned.
              1340 \ifdefinable\ifxP@validsol\relax
              1341 \ifdefinable\xP@validsoltrue\relax
              1342 \ifdefinable\xP@validsolfalse\relax
              1343 \newif\ifxP@validsol

              1344 \newcommand*\xP@solvelinearsystem{%
Scale the matrix so that the highest absolute value in each row and each column is  $\geq 2048$ pt
and  $< 4096$ pt.
              1345 \xP@scalerow\xP@A\xP@B\xP@C\xP@D
              1346 \xP@scalerow\xP@E\xP@F\xP@G\xP@H
              1347 \xP@scalerow\xP@I\xP@J\xP@K\xP@L
              1348 \xP@scalectol\xP@A\xP@E\xP@I\xP@scalectol
              1349 \xP@scalectol\xP@B\xP@F\xP@J\xP@scalectol
              1350 \xP@scalectol\xP@C\xP@G\xP@K\xP@scalectol
Record the initial variable-to-column assignment.
              1351 \let\xP@varone\xP@ta
              1352 \let\xP@vartwo\xP@tb
              1353 \let\xP@varthree\xP@tc
Find the pivot position. \xP@M is used temporarily.
              1354 \count@m@ne
              1355 \@tempcnta@m@ne
              1356 \xP@ifabsless\xP@A\xP@B\@tempcnta\z@\xP@M\xP@B
              1357 \else\xP@M\xP@A\fi
              1358 \xP@ifabsless\xP@M\xP@C\@tempcnta\@ne\xP@M\xP@C\fi
              1359 \xP@ifabsless\xP@M\xP@E\@tempcnta@m@ne\count@\z@\xP@M\xP@E\fi
              1360 \xP@ifabsless\xP@M\xP@F\@tempcnta\z@\count@\z@\xP@M\xP@F\fi
              1361 \xP@ifabsless\xP@M\xP@G\@tempcnta\@ne\count@\z@\xP@M\xP@G\fi
              1362 \xP@ifabsless\xP@M\xP@I\@tempcnta@m@ne\count@\@ne\xP@M\xP@I\fi
              1363 \xP@ifabsless\xP@M\xP@J\@tempcnta\z@\count@\@ne\xP@M\xP@J\fi
              1364 \xP@ifabsless\xP@M\xP@K\@tempcnta\@ne\count@\@ne\fi
Swap rows
              1365 \ifcase\count@
              1366 \xP@swapdim\xP@A\xP@E
              1367 \xP@swapdim\xP@B\xP@F
              1368 \xP@swapdim\xP@C\xP@G
              1369 \xP@swapdim\xP@D\xP@H
              1370 \or
              1371 \xP@swapdim\xP@A\xP@I

```

```

1372     \xP@swapdim\xP@B\xP@J
1373     \xP@swapdim\xP@C\xP@K
1374     \xP@swapdim\xP@D\xP@L
1375 \fi

Swap columns
1376 \ifcase\@tempcnta
1377     \xP@swapdim\xP@A\xP@B
1378     \xP@swapdim\xP@E\xP@F
1379     \xP@swapdim\xP@I\xP@J
1380     \let\xP@varone\xP@tb
1381     \let\xP@vartwo\xP@ta
1382     \xP@swapnum\xP@scaleone\xP@scaletwo
1383 \or
1384     \xP@swapdim\xP@A\xP@C
1385     \xP@swapdim\xP@E\xP@G
1386     \xP@swapdim\xP@I\xP@K
1387     \let\xP@varone\xP@tc
1388     \let\xP@varthree\xP@ta
1389     \xP@swapnum\xP@scaleone\xP@scaletthree
1390 \fi

First elimination
1391 \multiply\xP@E\m@ne
1392 \multiply\xP@I\m@ne

Absolute values below are < 8192pt.
1393 \ifdim\xP@A=\z@
1394 \else
1395     \advance\xP@F\dimexpr\xP@B*\xP@E/\xP@A\relax
1396     \advance\xP@G\dimexpr\xP@C*\xP@E/\xP@A\relax
1397     \advance\xP@H\dimexpr\xP@D*\xP@E/\xP@A\relax
1398     \advance\xP@J\dimexpr\xP@B*\xP@I/\xP@A\relax
1399     \advance\xP@K\dimexpr\xP@C*\xP@I/\xP@A\relax
1400     \advance\xP@L\dimexpr\xP@D*\xP@I/\xP@A\relax
1401 \fi

Find the second pivot element. \xP@M is used temporarily.
1402 \count@\m@ne
1403 \xP@ifabsless\xP@F\xP@G\@tempcnta\z@\xP@M\xP@G
1404     \else\@tempcnta\m@ne\xP@M\xP@F\fi
1405 \xP@ifabsless\xP@M\xP@J\@tempcnta\m@ne\count@\z@\xP@M\xP@J\fi
1406 \xP@ifabsless\xP@M\xP@K\@tempcnta\z@\count@\z@\fi

Swap rows
1407 \ifnum\count@=\z@
1408     \xP@swapdim\xP@F\xP@J
1409     \xP@swapdim\xP@G\xP@K
1410     \xP@swapdim\xP@H\xP@L
1411 \fi

Swap columns
1412 \ifnum\@tempcnta=\z@
1413     \xP@swapdim\xP@B\xP@C
1414     \xP@swapdim\xP@F\xP@G
1415     \xP@swapdim\xP@J\xP@K
1416     \let\@tempa\xP@varthree
1417     \let\xP@varthree\xP@vartwo
1418     \let\xP@vartwo\@tempa

```

```

1419 \xP@swapnum\xP@scaletwo\xP@scalethree
1420 \fi
    Second elimination. Absolute values are < 16384pt.
1421 \ifdim\xP@F=\z@
1422 \else
1423 \advance\xP@K\dimexpr-\xP@G*\xP@J/\xP@F\relax
1424 \advance\xP@L\dimexpr-\xP@H*\xP@J/\xP@F\relax
1425 \fi
    Compute the result from the upper triangular form. Since the matrix can be singular, we
    have to ensure in every step that no overflow occurs. In general, we do not allow any
    solution greater than 60pt.
1426 \xP@ifabsless{\dimexpr\xP@L/60\relax}{\dimexpr\xP@K/\xP@scalethree\relax}%
1427 \xP@validsoltrue
1428 \xP@varthree\dimexpr\xP@L*(\xP@scalethree*\p@)/\xP@K\relax
1429 \else
1430 \xP@validsolfalse
1431 \fi
1432 \xP@checkabs{\xP@H/8191}{\xP@F/\xP@scaletwo}%
1433 \xP@checkabs{\xP@G/\xP@scalethree/136}{\xP@F/\xP@scaletwo}%
1434 \ifxP@validsol
1435 \xP@vartwo\dimexpr\xP@H*(\xP@scaletwo*\p@)/\xP@F
1436 -\xP@varthree*\xP@scaletwo/\xP@scalethree*\xP@G/\xP@F\relax
1437 \xP@checkabs\xP@vartwo{60pt}%
1438 \fi
1439 \xP@checkabs{\xP@D/5461}{\xP@A/\xP@scaleone}%
1440 \xP@checkabs{\xP@B/\xP@scaletwo/91}{\xP@A/\xP@scaleone}%
1441 \xP@checkabs{\xP@C/\xP@scalethree/91}{\xP@A/\xP@scaleone}%
1442 \ifxP@validsol
1443 \xP@varone\dimexpr\xP@D*(\xP@scaleone*\p@)/\xP@A
1444 -\xP@vartwo*\xP@scaleone/\xP@scaletwo*\xP@B/\xP@A
1445 -\xP@varthree*\xP@scaleone/\xP@scalethree*\xP@C/\xP@A\relax
1446 \xP@checkabs\xP@varone{60pt}%
1447 \fi
    Return the result.
1448 \xdef\@gtempa{%
1449 \ifxP@validsol
1450 \xP@ta\the\xP@ta\relax
1451 \xP@tb\the\xP@tb\relax
1452 \xP@tc\the\xP@tc\relax
1453 \noexpand\xP@validsoltrue
1454 \else
1455 \noexpand\xP@validsolfalse
1456 \fi
1457 }%
1458 }\@gtempa
1459 }

```

\xP@scalerow ●3 Scale a row of the matrix to improve numerical precision. We scale by a power of two such that the maximal length is between 2048pt and 4096pt.

```

1460 \newcommand*\xP@scalerow[4]{%
1461 \xP@M\z@
1462 \xP@Max\xP@M#1%
1463 \xP@Max\xP@M#2%
1464 \xP@Max\xP@M#3%
1465 \xP@Max\xP@M#4%

```

$$134217727 = 2048 \cdot 65536 - 1$$

```

1466 \count@134217727
1467 \loop
1468 \divide\xP@M\tw@
1469 \ifdim\xP@M>\z@
1470 \divide\count@\tw@
1471 \repeat
1472 \advance\count@\@ne
1473 \multiply#1\count@
1474 \multiply#2\count@
1475 \multiply#3\count@
1476 \multiply#4\count@
1477 }

```

`\xP@scalecol` ●3 Scale a column of the matrix to improve numerical precision. The scaling factor has to be recorded for the solution assignment later.

```

1478 \newcommand*\xP@scalecol[4]{%
1479 \xP@M\z@
1480 \xP@Max\xP@M#1%
1481 \xP@Max\xP@M#2%
1482 \xP@Max\xP@M#3%
16777215 = 2048 · 8192 - 1
1483 #416777215
1484 \loop
1485 \divide\xP@M\tw@
1486 \ifdim\xP@M>\z@
1487 \divide#4\tw@
1488 \repeat
1489 \advance#4\@ne
1490 \multiply#1#4%
1491 \multiply#2#4%
1492 \multiply#3#4%
1493 }

```

`\xP@checkabs`

```

1494 \newcommand*\xP@checkabs[2]{%
1495 \xP@ifabsless{\dimexpr#1\relax}{\dimexpr#2\relax}\else\xP@validsolfalse\fi}

```

`\xP@applinsys` ●1 ●3 ●6 This is the second, alternative algorithm for Newton's method in the offset algorithm. Approximate a solution x for the linear system $Ax = b$ for a (3×3) -matrix A . The aim is to make the norm $\|Ax - b\|$ small with small values of $\|x\|$. The approach: Set $x = \lambda A^t b$ since the normed scalar product $\langle Ax, b \rangle / \|x\|$ is maximal in this case. The norm $\|Ax - b\|$ is then minimal for $\lambda = \|A^t b\|^2 / \|AA^t b\|^2$.

This approximation is performed between one and three times.

```

1496 \newcommand*\xP@applinsys{%

```

First iteration: approximate a solution and record the result.

```

1497 \xP@applinsys@
1498 \xP@ta\xP@dta
1499 \xP@tb\xP@dtb
1500 \xP@tc\xP@dtc

```

If the result is nonzero...

```

1501 \xP@checkapp
1502 \if@tempswa

```


...modify the objective function by the estimated change, approximate again,...

```
1503 \xP@modobj
1504 \xP@applinsys@
```

...and test for a nonzero result. If it is nonzero, repeat it a third time.

```
1505 \xP@checkapp
1506 \if@tempwa
1507 \xP@modsol
1508 \xP@modobj
1509 \xP@applinsys@
1510 \xP@modsol
1511 \fi
1512 \fi
```

Return the accumulated approximation from one to three iterations.

```
1513 \xdef\@gtempa{%
1514 \xP@ta\the\xP@ta\relax
1515 \xP@tb\the\xP@tb\relax
1516 \xP@tc\the\xP@tc\relax
1517 }}\@gtempa
1518 }
```

\xP@checkapp •6 Check whether the solution is nonzero.

```
1519 \newcommand*\xP@checkapp{%
1520 \@tempswatrue
1521 \ifdim\xP@dta=\z@
1522 \ifdim\xP@dtb=\z@
1523 \ifdim\xP@dtc=\z@
1524 \@tempswafalse
1525 \fi\fi\fi
1526 }
```

\xP@modobj •3 •6 Modify the objective function by the estimated difference, according to the first-order approximation.

```
1527 \newcommand*\xP@modobj{%
1528 \advance\xP@D
1529 \dimexpr-\xP@A*\xP@dta/\p@-\xP@B*\xP@dtb/\p@-\xP@C*\xP@dtc/\p@\relax
1530 \advance\xP@H
1531 \dimexpr-\xP@E*\xP@dta/\p@-\xP@F*\xP@dtb/\p@-\xP@G*\xP@dtc/\p@\relax
1532 \advance\xP@L
1533 \dimexpr-\xP@I*\xP@dta/\p@-\xP@J*\xP@dtb/\p@-\xP@K*\xP@dtc/\p@\relax
1534 }
```

\xP@modsol •3 •6 Modify the solution vector by the approximation.

```
1535 \newcommand*\xP@modsol{%
1536 \advance\xP@ta\xP@dta
1537 \advance\xP@tb\xP@dtb
1538 \advance\xP@tc\xP@dtc
1539 }
```

\xP@applinsys@ •1 •3 •6 The heart of the approximation routine.

```
1540 \newcommand*\xP@applinsys@{%
```

Determine scaling factors \xP@sa and \xP@sb to improve numerical precision.

```
1541 \xP@sa\z@
1542 \xP@Max\xP@sa\xP@A
1543 \xP@Max\xP@sa\xP@B
```

```

1544 \xP@Max\xP@sa\xP@C
1545 \xP@Max\xP@sa\xP@E
1546 \xP@Max\xP@sa\xP@F
1547 \xP@Max\xP@sa\xP@G
1548 \xP@Max\xP@sa\xP@I
1549 \xP@Max\xP@sa\xP@J
1550 \xP@Max\xP@sa\xP@K
1551 \xP@sa\ifdim\xP@sa<5460pt\thr@@\xP@sa\else\maxdimen\fi
1552 \xP@sb\z@
1553 \xP@Max\xP@sb\xP@D
1554 \xP@Max\xP@sb\xP@H
1555 \xP@Max\xP@sb\xP@L

```

Scale the vector b .

```

1556 \ifdim\xP@sb>\z@
1557 \xP@D\dimexpr\xP@D*\maxdimen/\xP@sb\relax
1558 \xP@H\dimexpr\xP@H*\maxdimen/\xP@sb\relax
1559 \xP@L\dimexpr\xP@L*\maxdimen/\xP@sb\relax
1560 \fi

```

Vector $A^t b$ (scaled)

```

1561 \xP@Aba\dimexpr\xP@A*\xP@D/\xP@sa+\xP@E*\xP@H/\xP@sa+\xP@I*\xP@L/\xP@sa\relax
1562 \xP@Abb\dimexpr\xP@B*\xP@D/\xP@sa+\xP@F*\xP@H/\xP@sa+\xP@J*\xP@L/\xP@sa\relax
1563 \xP@Abc\dimexpr\xP@C*\xP@D/\xP@sa+\xP@G*\xP@H/\xP@sa+\xP@K*\xP@L/\xP@sa\relax

```

Vector $AA^t b$ (scaled)

```

1564 \xP@AAba\dimexpr\xP@A*\xP@Aba/\xP@sa+\xP@B*\xP@Abb/\xP@sa
1565 +\xP@C*\xP@Abc/\xP@sa\relax
1566 \xP@AAbb\dimexpr\xP@E*\xP@Aba/\xP@sa+\xP@F*\xP@Abb/\xP@sa
1567 +\xP@G*\xP@Abc/\xP@sa\relax
1568 \xP@AAbc\dimexpr\xP@I*\xP@Aba/\xP@sa+\xP@J*\xP@Abb/\xP@sa
1569 +\xP@K*\xP@Abc/\xP@sa\relax

```

Another scaling factor.

```

1570 \xP@sc\z@
1571 \xP@Max\xP@sc\xP@Aba
1572 \xP@Max\xP@sc\xP@Abb
1573 \xP@Max\xP@sc\xP@Abc
1574 \xP@Max\xP@sc\xP@AAba
1575 \xP@Max\xP@sc\xP@AAbb
1576 \xP@Max\xP@sc\xP@AAbc

```

$\|A^t b\|^2$ and $\|AA^t b\|^2$

```

1577 \ifdim\xP@sc=\z@
1578 \xP@AAb\z@
1579 \else
1580 \xP@Ab\dimexpr\xP@Aba*\xP@bigdim/\xP@sc*\xP@Aba/\xP@sc
1581 +\xP@Abb*\xP@bigdim/\xP@sc*\xP@Abb/\xP@sc
1582 +\xP@Abc*\xP@bigdim/\xP@sc*\xP@Abc/\xP@sc
1583 \relax
1584 \xP@AAb\dimexpr\xP@AAba*\xP@bigdim/\xP@sc*\xP@AAba/\xP@sc
1585 +\xP@AAbb*\xP@bigdim/\xP@sc*\xP@AAbb/\xP@sc
1586 +\xP@AAbc*\xP@bigdim/\xP@sc*\xP@AAbc/\xP@sc
1587 \relax
1588 \fi

```

The approximation $x = \lambda A^t b$ with $\lambda = \|A^t b\|^2 / \|AA^t b\|^2$.

```

1589 \xdef\@gtempa{\%
1590 \ifdim\xP@AAb=\z@

```

```

1591      \xP@dt a\z@
1592      \xP@dt b\z@
1593      \xP@dt c\z@
1594      \else
1595      \xP@dt a\the\dimexpr\xP@Aba*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1596      \relax
1597      \xP@dt b\the\dimexpr\xP@Abb*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1598      \relax
1599      \xP@dt c\the\dimexpr\xP@Abc*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1600      \relax
1601      \fi
1602  }%
1603  }\@gtempa
1604 }

\ifxP@offsetok Switch whether the offset curve is enough
1605 \ifdefinable\ifxP@offsetok\relax
1606 \ifdefinable\xP@offsetoktrue\relax
1607 \ifdefinable\xP@offsetokfalse\relax
1608 \newif\ifxP@offsetok

\xP@maxdev Maximal deviation, measured at 19 points on the curve. The actual tolerance is 1/8 of
\xP@maxdev. With the current value 0.1pt, the tolerance is 0.0125pt, which is about 1/32
of the line width for the Computer Modern fonts.
1609 \newcommand*\xP@maxdev{.1pt}

\xP@maxobjfun Tolerance for the objective function. Recommended value is  $\frac{1}{2}(\xP@maxdev)^2$ .
1610 \newcommand*\xP@maxobjfun{.005pt}

\xP@testoffset •1 •5 Test procedure for the offset curve. It tests whether the Bézier curve defined
by the control points \X@p,...,\Y@c is a good approximation for the offset curve of the
partial curve defined by \xP@a,...,\xP@d in the parameter interval  $[\xP@a,\xP@b] \subseteq [0pt,\xP@bigdim]$ .
The parameter interval is uniformly divided by 20, and the deviation is measured at
the 19 inner positions. (Since the boundary points are offset exactly by the algorithm, they
do not need to be checked.)
For simplicity, the parameter interval for both curves is normalized to  $[0,1]$  in the
following explanations. Denote the original curve by  $c_1 : [0,1] \rightarrow \mathbb{R}^2$  and the offset curve
by  $c_2$ . The quality test is passed if the offset curve fulfills at each of the 19 test points
 $t_i \in \{\frac{1}{20}, \dots, \frac{19}{20}\}$  one of the following two conditions:


- Let  $v$  be the tangent vector  $c_1'(t_i)$ . For  $w := c_1(t_i) - c_2(t_i)$ , denote by  $w_{par}$  the
component parallel to  $v$  and by  $w_{orth}$  the component orthogonal to  $v$ . The test is
passed if  $|w_{par}| + |w_{orth} - \xP@off| \leq \frac{1}{8}\xP@maxdev$ . If  $\|v\|$  is very small so that
the direction cannot be determined precisely, the condition is  $\|c_1(t_i) - c_2(t_i)\| - |\xP@off| \leq \frac{1}{8}\xP@maxdev$ .
- Compute the normal line at  $t_i$  to the curve  $c_1$  and intersect it with  $c_2$ . The intersection
point is allowed to have a different parameter  $\hat{t}_i \in [t_i - 0.5, t_i + 0.5] \cap [0,1]$ . Then
let  $w := c_1(t_i) - c_2(\hat{t}_i)$  and test whether  $|w_{par}| + |w_{orth} - \xP@off| \leq \frac{1}{8}\xP@maxdev$ .
( $w_{par}$  is very small in this case and is nonzero only because of limited precision, in
particular since  $\hat{t}_i$  is determined with an error of  $\approx 2^{-17}$  ( $= \frac{1}{2}sp$ ).)


1611 \newcommand*\xP@testoffset{%
```

Default values for the return statement and the loop continuation.

```

1612 \gdef\xP@afteroffsetok{\xP@offsetoktrue}%
1613 \def\xP@offsetokif{\ifdim\xP@ti<1.85pt}%
1614 \xP@ti.1pt
1615 \loop
    \xP@tip =  $t_i$ , denormalized for  $c_1$ 
1616 \xP@tip\dimexpr\xP@a+(\xP@b-\xP@a)*\xP@ti/131072\relax
    Point on the original curve  $c_1$  (scaled by  $-8$ )
1617 \Lp\xP@precbezierpoly\xP@xa\xP@xb\xP@xc\xP@xd\xP@tip
1618 \Up\xP@precbezierpoly\xP@ya\xP@yb\xP@yc\xP@yd\xP@tip
     $8c_2(t_i) - 8c_1(t_i)$ 
1619 \xP@valA\dimexpr\xP@bezierpoly\Xp\Lc\Rc\Xc\xP@ti-\Lp\relax
1620 \xP@valB\dimexpr\xP@bezierpoly\Yp\Up\Dc\Yc\xP@ti-\Up\relax
     $v$ 
1621 \d@X3\xP@precbeziertan\xP@xa\xP@xb\xP@xc\xP@xd\xP@tip
1622 \d@Y3\xP@precbeziertan\xP@ya\xP@yb\xP@yc\xP@yd\xP@tip
1623 \xP@vecLen
    Decide if  $v$  is big enough (heuristically, may be changed in the future)
1624 \@tempdimc\dimexpr(\xP@b-\xP@a)*\@tempdimb/\xP@bigdim\relax
1625 \xP@abs\@tempdimc
1626 \ifdim.01pt<\@tempdimc
     $8w_{par}, 8w_{orth} - 8$  \xP@off,
1627 \xP@devA\dimexpr\xP@valA*\d@X/\@tempdimb+\xP@valB*\d@Y/\@tempdimb\relax
1628 \xP@devB\dimexpr\xP@valA*\d@Y/\@tempdimb-\xP@valB*\d@X/\@tempdimb-8\xP@off
1629 \relax
1630 \xP@abs\xP@devA
1631 \xP@abs\xP@devB
1632 \@tempdima\dimexpr\xP@devA+\xP@devB\relax
1633 \else
    If the velocity is zero, just pass the test.
1634 \ifdim\@tempdimc=\z@
1635 \@tempdima\z@
1636 \else
     $8\|c_1(t_i) - c_2(t_i)\|$ 
1637 {%
1638 \d@X\xP@valA
1639 \d@Y\xP@valB
1640 \xP@vecLen@
1641 \global\dimen@i\@tempdimb
1642 }\@tempdima\dimen@i
1643 \advance\@tempdima\ifdim\xP@off>\z@-\fi8\xP@off
1644 \xP@abs\@tempdima
1645 \fi
1646 \fi
    If the first condition is not fulfilled, test the second one.
1647 \ifdim\@tempdima>\xP@maxdev
     $-c_1(t_i)$ 
1648 \divide\Lp-8\relax
1649 \divide\Up-8\relax

```

Affine transformation of the offset curve: translate by $-c_1(t_i)$ and rotate so that the tangent v to $c_1(t_i)$ becomes the x -axis.

```

1650      {%
1651      \xP@transformcoor\X@p\Y@p
1652      \xP@transformcoor\L@c\U@c
1653      \xP@transformcoor\R@c\D@c
1654      \xP@transformcoor\X@c\Y@c

```

Find the parameter \tilde{t}_i and decide whether the approximation at \tilde{t}_i is good.

```

1655      \xP@findzero
1656      }%
1657      \fi
1658      \xP@offsetokif
1659      \advance\xP@ti.1pt
1660      \repeat
1661      \expandafter}\xP@afteroffsetok
1662 }

```

\xP@afteroffsetok

```

1663 \newcommand*\xP@afteroffsetok{}

```

\xP@offsetokif

```

1664 \newcommand*\xP@offsetokif{}

```

\xP@transformcoor •5 Affine coordinate transformation. First, translate the coordinates in (#1,#2) by the vector $-(\backslash L@p, \backslash U@p)$, then rotate by the angle between $v := (\backslash d@X, \backslash d@Y)$ and $(1,0)$. The register \@tempdimb must contain the length $\|v\|$.

```

1665 \newcommand*\xP@transformcoor[2]{%
1666   \advance#1\backslash L@p
1667   \advance#2\backslash U@p
1668   \@tempdima\dimexpr#1*\backslash d@X/\@tempdimb+#2*\backslash d@Y/\@tempdimb\relax
1669   #2\dimexpr#2*\backslash d@X/\@tempdimb-#1*\backslash d@Y/\@tempdimb\relax
1670   #1\@tempdima
1671 }

```

\xP@findzero •5 Find the parameter \tilde{t}_i by nested intervals/intermediate value theorem.

```

1672 \newcommand*\xP@findzero{%
1673   \xP@setleftvalue{.05}%
1674   \xP@setrightvalue{.05}%

```

Normalize: function value (x -coordinate) should be nonnegative at the upper end.

```

1675   \ifdim\xP@valB<\z@\xP@reversecoeff\fi

```

If the function value at the lower end is also positive, try a smaller parameter interval $t_i \pm \delta$ pt for $\delta \in \{.5, .35, .25, .2, .15, .1, .05\}$. Maybe we have different signs for the x -coordinate for the larger boundary parameters.

```

1676   \ifdim\xP@valA>\z@
1677     \@tempswatrue
1678     \for\@tempa:={.1,.15,.2,.25,.35,.5,1.1}\do{%
1679       \if@tempswa
1680         \xP@setleftvalue\@tempa
1681         \ifdim\xP@valA<\z@\@tempswafalse\fi
1682       \if@tempswa
1683         \xP@setrightvalue\@tempa
1684       \ifdim\xP@valB<\z@
1685         \@tempswafalse

```

```

1686         \xP@reversecoeff
1687     \fi
1688     \fi
1689     \fi
1690 }%

```

Last resort: Try the midpoint.

```

1691     \if@tempwa
1692         \L@p\xP@ti
1693         \xP@valA\xP@bezierpoly\X@p\L@c\R@c\X@c\L@p

```

If the midpoint leads to a negative value, we can proceed with a small interval. Otherwise, set both boundary points to the midpoint and effectively skip nested intervals.

```

1694         \ifdim\xP@valA<\z@

```

We had this before, so we know that the value is positive.

```

1695             \xP@setrightvalue{.05}%
1696         \else
1697             \U@p\L@p
1698             \xP@valB\xP@valA
1699         \fi
1700     \fi
1701 \fi

```

The actual nested interval algorithm

```

1702 \loop
1703 \ifnum\numexpr\U@p-\L@p\relax>\@ne
1704     \xP@ti\dimexpr(\L@p+\U@p)/2\relax
1705     \xP@devA\xP@bezierpoly\X@p\L@c\R@c\X@c\xP@ti
1706     \ifdim\xP@devA>\z@
1707         \U@p\xP@ti
1708         \xP@valB\xP@devA
1709     \else
1710         \L@p\xP@ti
1711         \xP@valA\xP@devA
1712     \fi
1713 \repeat

```

Take the left or right boundary point (only 1sp apart), depending on which one yields the smaller x -coordinate.

```

1714     \xP@ifabsless\xP@valB\xP@valA
1715     \L@p\U@p
1716     \xP@valA\xP@valB
1717 \fi

```

Compare the y -coordinate with $\xP@off$.

```

1718     \xP@valB\dimexpr\xP@bezierpoly\Y@p\U@c\D@c\Y@c\L@p+8\xP@off\relax
1719     \xP@abs\xP@valA
1720     \xP@abs\xP@valB
1721     \ifdim\dimexpr\xP@valA+\xP@valB\relax>\xP@maxdev\relax
1722         \xP@failed
1723     \fi
1724 }

```

$\xP@failed$ Break the loop for the t_i in $\xP@testoffset$. Set the return value to false.

```

1725 \newcommand*\xP@failed{%
1726     \global\let\xP@offsetokif\iffalse
1727     \gdef\xP@afteroffsetok{\xP@offsetokfalse}%
1728 }

```

`\xP@reversecoeff` Reverse the function for the nested interval algorithm.

```
1729 \newcommand*\xP@reversecoeff{%
1730   \multiply\xP@m@ne
1731   \multiply\L@c@m@ne
1732   \multiply\R@c@m@ne
1733   \multiply\X@c@m@ne
1734   \multiply\xP@valA@m@ne
1735   \multiply\xP@valB@m@ne
1736 }
```

`\xP@setleftvalue` •5

```
1737 \newcommand*\xP@setleftvalue[1]{%
1738   \L@p\dimexpr\xP@ti-#1\p@\relax
1739   \ifdim\L@p<-.1pt\L@p-.1pt\fi
1740   \xP@valA\xP@bezierpoly\X@p\L@c\R@c\X@c\L@p
1741 }
```

`\xP@setrightvalue` •5

```
1742 \newcommand*\xP@setrightvalue[1]{%
1743   \U@p\dimexpr\xP@ti+#1\p@\relax
1744   \ifdim\U@p>2.1\p@\U@p2.1\p@\fi
1745   \xP@valB\xP@bezierpoly\X@p\L@c\R@c\X@c\U@p
1746 }
```

8.10 Multiple dashed curves

`\xP@splinedbldashed`

```
1747 \newcommand*\xP@splinedbldashed{%
1748   \xP@checkspline\xP@splinemultdashed\xP@doublestroke}
```

`\xP@splinetrbldashed`

```
1749 \newcommand*\xP@splinetrbldashed{%
1750   \xP@checkspline\xP@splinemultdashed\xP@trblstroke}
```

`\xP@splinemultdashed`

```
1751 \newcommand*\xP@splinemultdashed[1]{%
Expected dash number. It is an even number if the spline is the continuation of the previous
one, otherwise (default case) an odd number.
1752   \xP@testcont\xP@dashmacro
1753   \@tempcnta
1754   \ifxP@splinecont
1755     \numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
1756   \else
1757     \numexpr2*((\@tempdimb+\xydashl@)/(2*\xydashl@))-1\relax
1758   \fi
1759   \ifnum\@tempcnta>\@ne
1760     \xP@splinemultdashed@#1%
1761   \else
```

One dash: paint a solid line. Less than one dash: Leave the segment out, just record the end point.

```
1762   \ifnum\@tempcnta=\@ne
1763     \xP@splinemultsolid#1
1764   \else
1765     \xP@savec
```

```

1766 \fi
1767 \fi
1768 \global\let\xP@lastpattern\xP@dashmacro
1769 }

```

`\xP@splinemultdashed@` ●1 ●7 Make a list of parameter pairs for the start and end point of a dash.

```

1770 \newcommand*\xP@splinemultdashed@[1]{%
1771 \xP@inibigdim
Dash length
1772 \@tempdima\dimexpr\@tempdimb/\@tempcnta\relax
1773 \xP@temppar\z@
1774 \toks@{}%
1775 \xP@savec
1776 \ifodd\@tempcnta
1777 \else
1778 \xP@slide
1779 \fi
1780 \@tempcnta\z@
1781 \loop
1782 \advance\@tempcnta\@ne
1783 \xP@append\toks@{\ifodd\@tempcnta\noexpand\xP@paintdash\fi
1784 \the\xP@temppar}}%
1785 \xP@oldpar\xP@temppar
1786 \xP@slide
1787 \ifdim\xP@temppar<\xP@bigdim
1788 \repeat

```

The last position is kept as a scaling factor so that the last dot can be drawn at exactly the parameter 1. Use the last or the next-to-last position, depending on the parity of segments.

```

1789 \xP@lastpar
1790 \ifodd\@tempcnta
1791 \xP@temppar
1792 \xP@append\toks@{\the\xP@temppar}}%
1793 \else
1794 \xP@oldpar
1795 \fi

```

Convert the list of parameters to a list of PDF tokens.

```

1796 \@temptokena{}%
1797 \xP@setsolidpat
1798 \global\let\xP@lastpattern\xP@dashmacro
1799 \@for\@tempa:={#1}\do{\the\toks@}%
1800 \xP@stroke{\the\@temptokena}%
1801 }}

```

`\xP@paintdash` ●1 ●7

```

1802 \newcommand*\xP@paintdash[2]{%
1803 \xP@paintsolid{\dimexpr#1*\xP@bigdim/\xP@lastpar\relax}%
1804 {\dimexpr#2*\xP@bigdim/\xP@lastpar\relax}%
1805 }

```

8.11 Multiple dotted curves

```

\splinedbldotted@
\xP@splinedbldotted@ 1806 \xP@hook{curve}{splinedbldotted@}
1807 \newcommand*\xP@splinedbldotted@{%

```



```

1808 \let\xP@normalmult\@ne
1809 \xP@checkspline\xP@splinemultdotted\xP@doublestroke}

\xP@splinetrbldotted

1810 \newcommand*\xP@splinetrbldotted{%
1811 \let\xP@normalmult\tw@
1812 \xP@checkspline\xP@splinemultdotted\xP@trblstroke}

\xP@mul tidottedpat Dotted lines with multiple strokes are drawn in a different way from single-stroked lines.
They are composed of many small, straight lines normal to the curve at every dot position.
Hence, the dot pattern for multiple curves has dots which are spaced by the normal distance
between strokes.

1813 \newcommand*\xP@mul tidottedpat{%
1814 \def\xP@pattern{0 J [\xP@coord\xP@preclw{\xydashh@-\xP@preclw}]0 d}%
1815 \global\let\xP@lastpattern\xP@dotmacro
1816 }

\xP@normalmult

1817 \@ifdefinable\xP@normalmult\relax

\xP@splinemultdotted ● 1 ● 7

1818 \newcommand\xP@splinemultdotted[1]{\%
1819 \xP@inibigdim
Make a list of dot positions on the spline segment.
1820 \xP@temppar\z@
1821 \xP@testcont\xP@dotmacro
1822 \ifxP@splinecont
Expected dot distance (see the formula in \xP@setdottedpat)
1823 \@tempdimc\dimexpr\@tempdimb/(\@tempdimb/131072+1)\relax
1824 \@tempdima\dimexpr\@tempdimc-\xP@preclw/2\relax
1825 \xP@slide
1826 \@tempdima\@tempdimc
1827 \else
1828 \@tempdima\dimexpr\xP@preclw/2\relax
1829 \xP@slide
Expected dot distance (see the formula in \xP@setdottedpat)
1830 \@tempdima\dimexpr\@tempdimb-\xP@preclw\relax
1831 \ifdim\@tempdima<\z@\@tempdima\z@\fi
1832 \@tempdima\dimexpr\@tempdima/(\@tempdima/131072+1)\relax
1833 \fi
1834 \xP@savvec
1835 \toks@{\}%
If the end of the segment is reached before the first dot position, leave the segment out.
1836 \ifdim\xP@temppar<\xP@bigdim
1837 \loop
1838 \xP@append\toks@{\noexpand\xP@paintdot{\the\xP@temppar}}%
1839 \xP@oldpar\xP@temppar
1840 \xP@slide
1841 \ifdim\xP@temppar<\xP@bigdim
1842 \repeat
1843 \xP@velocity\xP@bigdim\xP@tempvel

```

Test whether the last or the next-to-last dot is closer to `\xP@bigdim`. Measure from the end of the dot, hence the contribution of `\xP@preclw`. Also consider the case that the velocity at the end point is very small. In this case, always choose the next-to-last dot as the final one.

```

1844 \ifdim
1845 \ifdim\xP@preclw<\xP@tempvel
1846 \dimexpr2\xP@bigdim-\xP@oldpar-\xP@preclw*\xP@bigdim/\xP@tempvel\relax
1847 \else
1848 -\maxdimen
1849 \fi<\xP@temppar
1850 \xP@temppar\xP@oldpar
1851 \else
1852 \xP@append\toks@{\noexpand\xP@paintdot{\the\xP@temppar}}}%
1853 \fi
1854 \@tempdima\dimexpr\xP@preclw/2\relax
1855 \xP@slide
1856 \xP@lastpar\xP@temppar

```

Convert the list of parameters to a list of PDF tokens.

```

1857 \@temptokena{}%
1858 \the\toks@

```

Actually draw the points in the list.

```

1859 \xP@multidottedpat
1860 \xP@stroke{\the\@temptokena}%
1861 \else

```

Leave the segment out because it is too short.

```

1862 \global\let\xP@lastpattern\@empty
1863 \fi
1864 }}

```

`\xP@slide` ●1 ●7 Slide along the Bézier segment by `\@tempdima`. Needs: `Xy-pic` spline parameter, current position parameter `\xP@temppar`, total spline length `\@tempdimb`.

```

1865 \newcommand*\xP@slide{%
1866 \xP@slide@

```

Return the new spline parameter after sliding.

```

1867 \global\dimen@i\xP@temppar
1868 }\xP@temppar\dimen@i
1869 }

```

`\xP@slide@` ●1 ●7

```

1870 \newcommand*\xP@slide@{%

```

Compute the velocity at two points, the starting point and an estimate for the end point.

```

1871 \xP@velocity\xP@temppar\xP@tempvel

```

The first estimate for the parameter increment is based on the total spline length.

```

1872 \@tempdimc\dimexpr\xP@bigdim*\@tempdima/\@tempdimb\relax
1873 \count@z@
1874 \@tempwattrue

```

Improve the parameter increment iteratively.

```

1875 \loop

```

Velocity at the estimated end point.

```

1876 \xP@velocity{\xP@temppar+\@tempdimc}\xP@tempvel@

```

Prevent arithmetic overflow.

```

1877 \ifdim\dimexpr\@tempdima*4/13\relax>\xP@tempvel@
1878 \@tempswafalse
1879 \else

```

Difference to the old parameter increment. This is Newton's method, applied to the estimated spline length based on the velocities \xP@tempvel and \xP@tempvel@ at \xP@temppar and (\xP@temppar + \@tempdimc).

```

1880 \xP@parinc\dimexpr\@tempdima*\xP@bigdim/\xP@tempvel@
1881 -(\xP@tempvel+\xP@tempvel@)/2*\@tempdimc/\xP@tempvel@\relax
1882 \advance\@tempdimc\xP@parinc

```

If the estimated parameter increment is bigger than .12, increase the parameter by .1 and slide only partially. This increases the precision if the parameter increment is big.

```

1883 \ifdim\@tempdimc>.12\xP@bigdim
1884 \@tempswafalse
1885 \else

```

If the estimate is not improved, break the loop.

```

1886 \ifdim\xP@parinc=\z@
1887 \@tempswafalse
1888 \else

```

Also break the loop after 10 iterations.

```

1889 \ifnum\count@=9\relax
1890 \@tempswafalse
1891 \fi
1892 \fi
1893 \fi
1894 \fi
1895 \if@tempswa
1896 \advance\count@\@ne
1897 \repeat

```

Note that \if@tempswa is always false here.

If the parameter increment would be more than .1 and if the parameter is not too big already, increase the parameter by .1 and slide again.

```

1898 \ifdim\xP@temppar<5461pt
1899 \ifdim\@tempdimc>.1\xP@bigdim
1900 \@tempswatrue
1901 \fi
1902 \fi
1903 \if@tempswa
1904 {%
1905 \dimen5\xP@temppar
1906 \advance\xP@temppar.1\xP@bigdim

```

Cap the end parameter to prevent arithmetic overflows.

```

1907 \ifdim\xP@temppar>5461pt\xP@temppar5461pt\fi
1908 \dimen7\xP@temppar

```

Determine the exact distance of the partial slide.

```

1909 \xP@shaveprec{\dimen5}{\dimen7}%
1910 \xP@bezierlength
1911 \global\dimen@i\dimexpr\@tempdima-\@tempdimb\relax
1912 \global\dimen3\xP@temppar
1913}%
1914 \@tempdima\dimen@i
1915 \xP@temppar\dimen3\relax

```

Slide again.

```
1916 \expandafter\xP@slide@
1917 \else
```

Finish the slide and return the new parameter.

```
1918 \advance\xP@temppar\@tempdimc
1919 \fi
1920 }
```

`\xP@paintdot` ●1 ●7

```
1921 \newcommand*\xP@paintdot[1]{%
```

Scale the parameter with a correction factor

```
1922 \@tempdima\dimexpr#1*\xP@bigdim/\xP@lastpar\relax
```

Position at parameter value `\xP@temppar`

```
1923 \xP@tangent
1924 \xP@posX\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\@tempdima/8\relax
1925 \xP@posY\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\@tempdima/8\relax
```

Normal vector to the curve with length `\xydashh@`

```
1926 \@tempdima\dimexpr(\xydashh@+\xP@preclw/\xP@normalmult)/2\relax
1927 \L@p\dimexpr\d@Y*\@tempdima/\@tempdimb\relax
1928 \U@p\dimexpr-\d@X*\@tempdima/\@tempdimb\relax
```

Append two points on both sides of the curve to the list. (The “multidottedpat” pattern is made to draw points with distance `\xydashh@`.)

```
1929 \xP@append\@temptokena{\xP@coor{\xP@posX+\L@p*\xP@normalmult}}%
1930 {\xP@posY+\U@p*\xP@normalmult}m %
1931 \xP@coor{\xP@posX-\L@p*\xP@normalmult+\@ne}}%
1932 {\xP@posY-\U@p*\xP@normalmult+\@ne)}l }%
1933 }
```

8.12 Squiggled curves

`\xP@splinesquiggled`

```
1934 \newcommand*\xP@splinesquiggled{%
1935 \xP@checkspline\xP@splinesquiggled@z@}
```

`\xP@splinedblsquiggled`

```
1936 \newcommand*\xP@splinedblsquiggled{%
1937 \xP@checkspline\xP@splinesquiggled@\xP@doublestroke}
```

`\xP@splinetrbldsquiggled`

```
1938 \newcommand*\xP@splinetrbldsquiggled{%
1939 \xP@checkspline\xP@splinesquiggled@\xP@trblstroke}
```

`\xP@splinesquiggled@` ●1 ●7

```
1940 \newcommand*\xP@splinesquiggled@[1]{%
1941 \xP@inibigdim
```

Reverse the direction of the little arcs, if the last squiggle from the previous segment makes it necessary.

```
1942 \xP@testcont\xP@oddsquigglemacro
1943 \ifxP@splinecont
1944 \def\xP@squigsign{-}%
1945 \else
```

```

1946      \let\xP@squigsign\@empty
1947      \fi
1948      \xP@savvec
Expected squiggle length
1949      \@tempcnta=\numexpr\@tempdimb/\xybsqll\relax
1950      \ifnum\@tempcnta<\tw@\@tempcnta\tw@\fi
1951      \multiply\@tempcnta\tw@
1952      \@tempdima\dimexpr\@tempdimb/\@tempcnta\relax
1953      \xP@squiglen\@tempdima
Make a list of dot positions on the spline segment.
1954      \xP@temppar\z@
1955      \toks@{}%
1956      \@tempcnta\z@
1957      \loop
1958          \advance\@tempcnta\@ne
1959          \xP@append\toks@{\noexpand\xP@paintsquiggle{\the\xP@temppar}}%
1960          \xP@oldpar\xP@temppar
1961          \xP@slide
1962          \ifdim\xP@temppar<\xP@bigdim
1963              \repeat
The last position is kept as a scaling factor so that the last dot can be drawn at exactly
the parameter 1. Use the last or the next-to-last position, on the parity of the number of
positions.
1964      \xP@lastpar
1965      \ifodd\@tempcnta
1966          \xP@oldpar
1967          \advance\@tempcnta\m@ne
1968      \else
1969          \xP@temppar
1970          \xP@append\toks@{\noexpand\xP@paintsquiggle{\the\xP@temppar}}%
1971      \fi
Convert the list of parameters to a list of PDF tokens.
1972      \@temptokena{}%
1973      \xP@setsolidpat
Record the direction of the last squiggle.
1974      \global\expandafter\let\expandafter\xP@lastpattern
1975      \ifodd\numexpr\@tempcnta/2\if\xP@squigsign-+1\fi\relax
1976          \xP@oddsquigglemacro
1977      \else
1978          \xP@evensquigglemacro
1979      \fi
Draw the squiggles.
1980      \@for\@tempa:={#1}\do{%
1981          \let\xP@dosquiggle\xP@dosquiggle@
1982          \count@\z@
1983          \the\toks@
1984      }%
1985      \xP@stroke{\the\@temptokena}%
1986  }}
\xP@paintsquiggle  ●1 ●7
1987 \newcommand*\xP@paintsquiggle[1]{%
1988     \xP@squigglevectors{#1}%

```

```

1989 \xP@dosquiggle
1990 \ifnum\count@=\thr@@\relax\count@\z@\else\advance\count@\@ne\fi
1991 }

```

\xP@squigglevectors ●1 ●7

```

1992 \newcommand*\xP@squigglevectors[1]{%
  Scale the parameter with a correction factor
1993 \@tempdima\dimexpr#1*\xP@bigdim/\xP@lastpar\relax
  Position at parameter value \xP@temppar, offset for multiple curves.
1994 \xP@tangent
1995 \xP@posX\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\@tempdima/8%
1996 -\d@Y*(\@tempa)/\@tempdimb\relax
1997 \xP@posY\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\@tempdima/8%
1998 +\d@X*(\@tempa)/\@tempdimb\relax
  Tangent vector to the curve with correct length
1999 \L@p\dimexpr\d@X*\xP@squiglen/\@tempdimb\relax
2000 \U@p\dimexpr\d@Y*\xP@squiglen/\@tempdimb\relax
2001 \R@p\dimexpr\L@p*543339720/1311738121\relax
2002 \D@p\dimexpr\U@p*543339720/1311738121\relax
2003 \X@min\dimexpr\L@p*362911648/967576667\relax
2004 \Y@min\dimexpr\U@p*362911648/967576667\relax
2005 \X@max\dimexpr(\L@p+\xP@squigsign\U@p)*173517671/654249180\relax
2006 \Y@max\dimexpr(\L@p-\xP@squigsign\U@p)*173517671/654249180\relax
2007 }

```

\xP@dosquiggle ●7

```

2008 \ifdefinable\xP@dosquiggle@\relax

```

\xP@dosquiggle@ ●7

```

2009 \newcommand*\xP@dosquiggle@{%
2010 \edef\next@{\xP@coor{\xP@posX}{\xP@posY}m
2011 \xP@coor{\xP@posX+\Y@max}{\xP@posY+\xP@squigsign\X@max}%
2012 }%
2013 \let\xP@dosquiggle\xP@dosquiggle@@
2014 }

```

\xP@dosquiggle@@ ●7

```

2015 \newcommand*\xP@dosquiggle@@{%
2016 \xP@append\@temptokena{\next@\expandafter\xP@coor
2017 \ifcase\count@
2018 {\xP@posX-\Y@max}{\xP@posY-\xP@squigsign\X@max}%
2019 \xP@coor\xP@posX\xP@posY
2020 \or
2021 {\xP@posX-\xP@squigsign\D@p-\X@min}{\xP@posY+\xP@squigsign\R@p-\Y@min}%
2022 \xP@coor{\xP@posX-\xP@squigsign\D@p}{\xP@posY+\xP@squigsign\R@p}%
2023 \or
2024 {\xP@posX-\X@max}{\xP@posY+\xP@squigsign\Y@max}%
2025 \xP@coor\xP@posX\xP@posY
2026 \or
2027 {\xP@posX+\xP@squigsign\D@p-\X@min}{\xP@posY-\xP@squigsign\R@p-\Y@min}%
2028 \xP@coor{\xP@posX+\xP@squigsign\D@p}{\xP@posY-\xP@squigsign\R@p}%
2029 \fi c }%
2030 \edef\next@{\expandafter\xP@coor
2031 \ifcase\count@

```

```

2032      {\xP@posX+\Y@max}{\xP@posY+\xP@squigsign\X@max}%
2033      \or
2034      {\xP@posX-\xP@squigsign\D@p+\X@min}{\xP@posY+\xP@squigsign\R@p+\Y@min}%
2035      \or
2036      {\xP@posX+\X@max}{\xP@posY-\xP@squigsign\Y@max}%
2037      \or
2038      {\xP@posX+\xP@squigsign\D@p+\X@min}{\xP@posY-\xP@squigsign\R@p+\Y@min}%
2039      \fi
2040    }%
2041  }

```

\xP@splinebrokensquiggled

```

2042 \newcommand*\xP@splinebrokensquiggled{%
2043   \xP@checkspline\xP@splinebrokensquiggled@ \z@

```

\xP@splinebrokendsquiggled

```

2044 \newcommand*\xP@splinebrokendsquiggled{%
2045   \xP@checkspline\xP@splinebrokensquiggled@\xP@doublestroke}

```

\xP@splinebrokentrblesquiggled

```

2046 \newcommand*\xP@splinebrokentrblesquiggled{%
2047   \xP@checkspline\xP@splinebrokensquiggled@\xP@trblstroke}

```

\xP@splinebrokensquiggled@ ●1 ●7

```

2048 \newcommand*\xP@splinebrokensquiggled@[1]{%
2049   \xP@inibigdim

```

Start with a space if the last squiggle from the previous segment makes it necessary.

```

2050   \xP@testcont\xP@brokensquigglemacro
2051   \ifxP@splinecont
2052     \let\xP@squigsign \@firstoftwo
2053   \else
2054     \let\xP@squigsign \@secondoftwo
2055   \fi
2056   \xP@savvec

```

Expected squiggle length

```

2057   \@tempcnta\numexpr(\@tempdimb\xP@squigsign\{+2*\xybsql1@})%
2058   /(4*\xybsql1@)\relax
2059   \ifnum\@tempcnta<\@ne\@tempcnta\@ne\fi
2060   \@tempdima\dimexpr\@tempdimb/(8*\@tempcnta\xP@squigsign\{-4})\relax
2061   \xP@squiglen\@tempdima

```

Make a list of dot positions on the spline segment.

```

2062   \xP@temppar\z@
2063   \xP@squigsign{\xP@slide\xP@slide\xP@slide\xP@slide}\}%
2064   \count@\z@
2065   \toks@{\}%
2066   \loop
2067     \xP@append\toks@{\noexpand\xP@paintbrokensquiggle{\the\xP@temppar}}%
2068     \xP@slide
2069     \xP@append\toks@{\the\xP@temppar}%
2070     \xP@slide
2071     \xP@append\toks@{\the\xP@temppar}%
2072     \xP@slide
2073     \xP@append\toks@{\the\xP@temppar}%
2074     \xP@slide

```

```

2075      \xP@append\toks@{\the\xP@temppar}}%
2076      \xP@lastpar\xP@temppar
2077      \advance\count@\@ne
2078      \ifnum\count@<\@tempcnta
2079      \xP@slide
2080      \xP@slide
2081      \xP@slide
2082      \xP@slide
2083      \repeat

```

Convert the list of parameters to a list of PDF tokens.

```

2084      \@temptokena{%
2085      \xP@setsolidpat
2086      \global\let\xP@lastpattern\xP@brokensquigglesmacro

```

Draw the squiggles.

```

2087      \let\xP@squigsig\@empty
2088      \@for\@tempa:={#1}\do{\the\toks@}%
2089      \xP@stroke{\the\@temptokena}%
2090  }}

```

\xP@paintbrokensquiggle ●1●7

```

2091 \newcommand*\xP@paintbrokensquiggle[5]{%
2092   \xP@squigglevectors{#1}%
2093   \xP@append\@temptokena{%
2094     \xP@coor\xP@posX\xP@posY m %
2095     \xP@coor{\xP@posX+\Y@max}{\xP@posY+\X@max}%
2096   }%
2097   \xP@squigglevectors{#2}%
2098   \xP@append\@temptokena{%
2099     \xP@coor{\xP@posX-\D@p-\X@min}{\xP@posY+\R@p-\Y@min}%
2100     \xP@coor{\xP@posX-\D@p}{\xP@posY+\R@p}c %
2101     \xP@coor{\xP@posX-\D@p+\X@min}{\xP@posY+\R@p+\Y@min}%
2102   }%
2103   \xP@squigglevectors{#3}%
2104   \xP@append\@temptokena{%
2105     \xP@coor{\xP@posX-\X@max}{\xP@posY+\Y@max}%
2106     \xP@coor\xP@posX\xP@posY c %
2107     \xP@coor{\xP@posX+\X@max}{\xP@posY-\Y@max}%
2108   }%
2109   \xP@squigglevectors{#4}%
2110   \xP@append\@temptokena{%
2111     \xP@coor{\xP@posX+\D@p-\X@min}{\xP@posY-\R@p-\Y@min}%
2112     \xP@coor{\xP@posX+\D@p}{\xP@posY-\R@p}c %
2113     \xP@coor{\xP@posX+\D@p+\X@min}{\xP@posY-\R@p+\Y@min}%
2114   }%
2115   \xP@squigglevectors{#5}%
2116   \xP@append\@temptokena{%
2117     \xP@coor{\xP@posX-\Y@max}{\xP@posY-\X@max}%
2118     \xP@coor\xP@posX\xP@posY c %
2119   }%
2120 }

```

End of the section for Xy-pic's “curve” option.

```

2121 \xyendinput
2122 </curve>
2123 <*basic>

```


8.13 Spline continuation

The following code handles the spline continuation (see [Section 5](#)). We introduce global macros which store the last end point of a Bézier segment. If the next segment continues at exactly the same coordinates, the dash/dot/squiggle patterns recognize the continuation.

```

\XP@lastX
\XP@lastY 2124 \newcommand*\XP@lastX{}
\XP@lastpattern 2125 \newcommand*\XP@lastY{}
2126 \newcommand*\XP@lastpattern{}

\XP@solidmacro
\XP@dotmacro 2127 \newcommand*\XP@solidmacro{solid}
\XP@dashmacro 2128 \newcommand*\XP@dotmacro{dot}
\XP@evensquigglemacro 2129 \newcommand*\XP@dashmacro{dash}
\XP@oddsquigglemacro 2130 \newcommand*\XP@evensquigglemacro{evensquiggle}
\XP@brokensquigglemacro 2131 \newcommand*\XP@oddsquigglemacro{oddsquiggle}
2132 \newcommand*\XP@brokensquigglemacro{brokensquiggle}

\xyinside@ Reset the last position with every new diagram.
2133 \renewcommand*\xyinside@{%
2134   \global\let\XP@lastpattern\@empty
2135   \saveXyStyle@ \aftergroup\xycheck@end
2136   \setboxz\h\bgroup
2137   \plainxy@
2138   \X@c=\z@\Y@c=\z@\czeroEdge@
2139   \X@p=\z@\Y@p=\z@\U@p=\z@\D@p=\z@\L@p=\z@\R@p=\z@\Edge@p={\zeroEdge}%
2140   \X@min=\hsize\X@max=-\hsize\Y@min=\hsize\Y@max=-\hsize
2141   \mathsurround=\z@
2142   \expandafter\POS\everyxy@@
2143 }

\XP@savec Save the current end point
2144 \newcommand*\XP@savec{%
2145   \xdef\XP@lastX{\the\X@c}%
2146   \xdef\XP@lastY{\the\Y@c}%
2147 }

\ifxP@splinecont Switch: does the next line/spline continue at the end point of the last one?
2148 \@ifdefinable\ifxP@splinecont\relax
2149 \@ifdefinable\XP@splineconttrue\relax
2150 \@ifdefinable\XP@splinecontfalse\relax
2151 \newif\ifxP@splinecont

\XP@testcont Test for \ifxP@splinecont
2152 \newcommand*\XP@testcont[1]{%
2153   \XP@splinecontfalse
2154   \ifxP@cont
2155     \ifx\XP@lastpattern#1%
2156       \ifdim\XP@lastX=\X@p
2157         \ifdim\XP@lastY=\Y@p
2158           \XP@splineconttrue
2159         \fi
2160       \fi
2161     \fi
2162   \fi
2163 }
```

```

\ifxP@cont Switch: shall the spline hack be applied?
2164 \@ifdefinable\ifxP@cont\relax
2165 \@ifdefinable\xP@conttrue\relax
2166 \@ifdefinable\xP@contfalse\relax
2167 \newif\ifxP@cont

\xypdfcontpatternon
\xypdfcontpatternoff 2168 \newcommand*\xypdfcontpatternon{\xP@conttrue}
2169 \newcommand*\xypdfcontpatternoff{\xP@contfalse}
2170 \xP@conttrue

```

8.14 Color

```

2171 </basic>
2172 <*color>
2173 \xycatcodes
2174 \expandafter\let\csname xypdf-co@loaded\endcsname\@empty

\xP@colorname
  \xP@colA 2175 \@ifdefinable\xP@colorname\relax
  \xP@colB 2176 \@ifdefinable\xP@colA\relax
  \xP@colC 2177 \@ifdefinable\xP@colB\relax
  \xP@colD 2178 \@ifdefinable\xP@colC\relax
2179 \@ifdefinable\xP@colD\relax

\newxycolor
\xP@newxycolor 2180 \xP@hook{color}{\newxycolor}
2181 \newcommand*\xP@newxycolor[2]{%
2182   \def\xP@colorname{#1}%
2183   \xP@parsecolor#2 @%
2184 }

\xP@parsecolor
2185 \@ifdefinable\xP@parsecolor\relax
2186 \def\xP@parsecolor#1 #2 #3@{%
2187   \def\xP@colA{#1}%
2188   \def\xP@colB{#2}%
2189   \ifx\xP@colB\xP@gray
2190     \xP@newcolor\xP@colorname\xP@colA\xP@gray\newxycolor
2191   \else
2192     \xP@parsecolor@#3 @%
2193   \fi
2194 }

\xP@parsecolor@
2195 \@ifdefinable\xP@parsecolor@\relax
2196 \def\xP@parsecolor@#1 #2 #3 #4@{%
2197   \def\xP@colC{#1}%
2198   \def\xP@colD{#2}%
2199   \ifx\xP@colD\xP@rgb
2200     \xP@newcolor\xP@colorname{\xP@colA,\xP@colB,\xP@colC}\xP@rgb\newxycolor
2201   \else
2202     \def\@tempa{#3}%
2203     \ifx\@tempa\xP@cmyk
2204       \xP@newcolor\xP@colorname{\xP@colA,\xP@colB,\xP@colC,\xP@colD}{cmyk}%
2205       \newxycolor

```

```

2206     \else
2207     \PackageError{xypdf}{Syntax error in \string\newxycolor}{}%
2208 \fi
2209 }

\XP@gray
\XP@rgb 2210 \newcommand*\XP@gray{gray}
\XP@cmyk 2211 \newcommand*\XP@rgb{rgb}
2212 \newcommand*\XP@cmyk{cmyk}

\OBJECT@shape
\XP@OBJECT@shape 2213 \XP@hook{color}{OBJECT@shape}
2214 \newcommand*\XP@OBJECT@shape[1]{\DN@{shape [1]}}%
2215 \expandafter\let\expandafter\nextii@\csname\codeof\next@\endcsname
2216 \ifx\nextii@\relax\DN@{style [1]}}%
2217 \expandafter\let\expandafter\nextii@\csname\codeof\next@\endcsname
2218 \ifx\nextii@\relax\DN@{\XP@checkcolor{1}}}%
2219 \else\DN@{\nextii@\xyFN@OBJECT@}%
2220 \fi
2221 \else\expandafter\addtotoks@\expandafter{\nextii@}%
2222 \DN@{\xyFN@OBJECT@}%
2223 \fi \next@}

\XP@checkcolor
2224 \newcommand*\XP@checkcolor[1]{%
2225 \ifundefined{\string\color@\detokenize{1}}}%
2226 {\OBJECT@shapei[1]}}%
2227 {%
2228 \XP@append\toks@{\noexpand\XP@color{\detokenize{1}}}}%
2229 \xyFN@OBJECT@
2230 }%
2231 }

\XP@color
2232 \newcommand*\XP@color[1]{%
2233 \def\preStyle@{\styletoks@{\XP@protectedcolor#1}}%
2234 \let\postStyle@@\empty
2235 \modXYstyle@
2236 }

\XP@protectedcolor Somehow, \protect\color is not enough, so we use  $\varepsilon$ -TEX's way of \protected macro
definitions.
2237 \@ifdefinable\XP@protectedcolor\relax
2238 \protected\def\XP@protectedcolor{%
2239 \ifundefined{color}\XP@pdfcolor\color
2240 }

\XP@pdfcolor
2241 \@ifdefinable\XP@pdfcolor\relax
2242 \def\XP@pdfcolor[1]#2{%
2243 \edef\@tempa{#1}%
2244 \ifx\@tempa\XP@gray
2245 \DN@{\XP@graycolor{#2}}}%
2246 \else\ifx\@tempa\XP@rgb
2247 \DN@{\XP@rgbcolor#2@}%
2248 \else\ifx\@tempa\XP@cmyk

```

```

2249     \DN@{\xP@cmykcolor#2@}%
2250     \fi\fi\fi
2251     \aftergroup\xP@resetcolor
2252     \next@
2253 }%

\xP@graycolor
2254 \newcommand*\xP@graycolor[1]{\xP@setcolor{#1}gG}%

\xP@rgbcolor
2255 \@ifdefinable\xP@rgbcolor\relax
2256 \def\xP@rgbcolor#1,#2,#3@{\xP@setcolor{#1 #2 #3}{rg}{RG}}

\xP@cmykcolor
2257 \@ifdefinable\xP@cmykcolor\relax
2258 \def\xP@cmykcolor#1,#2,#3,#4@{\xP@setcolor{#1 #2 #3 #4}{kK}}

\xP@newcolor
2259 \newcommand*\xP@newcolor[4]{%
2260   \expandafter\let\expandafter\next@\csname shape [1]\endcsname
2261   \ifx\next@\relax
2262     \@ifundefined{string\color@#1}\relax
2263     {\xP@warning{xypdf}{The color ‘#1’ is overridden by %
2264       \string#4}}%
2265     \DN@{\newxystyle{#1}{\xP@unnamedcolor{#2}{#3}}}%
2266   \else
2267     \DN@{}%
2268   \fi
2269   \next@\relax}

\xP@unnamedcolor
2270 \newcommand*\xP@unnamedcolor[2]{\xP@color{[2]{#1}}}%

\xP@definecrayolacolor
2271 \newcommand\xP@definecrayolacolor[2]{%
2272   \xP@newcolor{#1}{#2}{cmyk}\UseCrayolaColors}%

\xP@installCrayolaColors
2273 \newcommand*\xP@installCrayolaColors{%
2274   \xP@definecrayolacolor{GreenYellow}{.15,0,.69,0}%
2275   \xP@definecrayolacolor{Yellow}{0,0,1,0}%
2276   \xP@definecrayolacolor{Goldenrod}{0,.1,.84,0}%
2277   \xP@definecrayolacolor{Dandelion}{0,.29,.84,0}%
2278   \xP@definecrayolacolor{Apricot}{0,.32,.52,0}%
2279   \xP@definecrayolacolor{Peach}{0,.5,.7,0}%
2280   \xP@definecrayolacolor{Melon}{0,.46,.5,0}%
2281   \xP@definecrayolacolor{YellowOrange}{0,.42,1,0}%
2282   \xP@definecrayolacolor{Orange}{0,.61,.87,0}%
2283   \xP@definecrayolacolor{BurntOrange}{0,.51,1,0}%
2284   \xP@definecrayolacolor{Bittersweet}{0,.75,1,.24}%
2285   \xP@definecrayolacolor{RedOrange}{0,.77,.87,0}%
2286   \xP@definecrayolacolor{Mahogany}{0,.85,.87,.35}%
2287   \xP@definecrayolacolor{Maroon}{0,.87,.68,.32}%
2288   \xP@definecrayolacolor{BrickRed}{0,.89,.94,.28}%
2289   \xP@definecrayolacolor{Red}{0,1,1,0}%
2290   \xP@definecrayolacolor{OrangeRed}{0,1,.5,0}%

```

```

2291 \xP@definecrayolacolor{RubineRed}{0,1,.13,0}%
2292 \xP@definecrayolacolor{WildStrawberry}{0,.96,.39,0}%
2293 \xP@definecrayolacolor{Salmon}{0,.53,.38,0}%
2294 \xP@definecrayolacolor{CarnationPink}{0,.63,0,0}%
2295 \xP@definecrayolacolor{Magenta}{0,1,0,0}%
2296 \xP@definecrayolacolor{VioletRed}{0,.81,0,0}%
2297 \xP@definecrayolacolor{Rhodamine}{0,.82,0,0}%
2298 \xP@definecrayolacolor{Mulberry}{.34,.9,0,.02}%
2299 \xP@definecrayolacolor{RedViolet}{.07,.9,0,.34}%
2300 \xP@definecrayolacolor{Fuchsia}{.47,.91,0,.08}%
2301 \xP@definecrayolacolor{Lavender}{0,.48,0,0}%
2302 \xP@definecrayolacolor{Thistle}{.12,.59,0,0}%
2303 \xP@definecrayolacolor{Orchid}{.32,.64,0,0}%
2304 \xP@definecrayolacolor{DarkOrchid}{.4,.8,.2,0}%
2305 \xP@definecrayolacolor{Purple}{.45,.86,0,0}%
2306 \xP@definecrayolacolor{Plum}{.5,1,0,0}%
2307 \xP@definecrayolacolor{Violet}{.79,.88,0,0}%
2308 \xP@definecrayolacolor{RoyalPurple}{.75,.9,0,0}%
2309 \xP@definecrayolacolor{BlueViolet}{.86,.91,0,.04}%
2310 \xP@definecrayolacolor{Periwinkle}{.57,.55,0,0}%
2311 \xP@definecrayolacolor{CadetBlue}{.62,.57,.23,0}%
2312 \xP@definecrayolacolor{CornflowerBlue}{.65,.13,0,0}%
2313 \xP@definecrayolacolor{MidnightBlue}{.98,.13,0,.43}%
2314 \xP@definecrayolacolor{NavyBlue}{.94,.54,0,0}%
2315 \xP@definecrayolacolor{RoyalBlue}{1,.5,0,0}%
2316 \xP@definecrayolacolor{Blue}{1,1,0,0}%
2317 \xP@definecrayolacolor{Cerulean}{.94,.11,0,0}%
2318 \xP@definecrayolacolor{Cyan}{1,0,0,0}%
2319 \xP@definecrayolacolor{ProcessBlue}{.96,0,0,0}%
2320 \xP@definecrayolacolor{SkyBlue}{.62,0,.12,0}%
2321 \xP@definecrayolacolor{Turquoise}{.85,0,.2,0}%
2322 \xP@definecrayolacolor{TealBlue}{.86,0,.34,.02}%
2323 \xP@definecrayolacolor{Aquamarine}{.82,0,.3,0}%
2324 \xP@definecrayolacolor{BlueGreen}{.85,0,.33,0}%
2325 \xP@definecrayolacolor{Emerald}{1,0,.5,0}%
2326 \xP@definecrayolacolor{JungleGreen}{.99,0,.52,0}%
2327 \xP@definecrayolacolor{SeaGreen}{.69,0,.5,0}%
2328 \xP@definecrayolacolor{Green}{1,0,1,0}%
2329 \xP@definecrayolacolor{ForestGreen}{.91,0,.88,.12}%
2330 \xP@definecrayolacolor{PineGreen}{.92,0,.59,.25}%
2331 \xP@definecrayolacolor{LimeGreen}{.5,0,1,0}%
2332 \xP@definecrayolacolor{YellowGreen}{.44,0,.74,0}%
2333 \xP@definecrayolacolor{SpringGreen}{.26,0,.76,0}%
2334 \xP@definecrayolacolor{OliveGreen}{.64,0,.95,.4}%
2335 \xP@definecrayolacolor{RawSienna}{0,.72,1,.45}%
2336 \xP@definecrayolacolor{Sepia}{0,.83,1,.7}%
2337 \xP@definecrayolacolor{Brown}{0,.81,1,.6}%
2338 \xP@definecrayolacolor{Tan}{.14,.42,.56,0}%
2339 \xP@definecrayolacolor{Gray}{0,0,0,.5}%
2340 \xP@definecrayolacolor{Black}{0,0,0,1}%
2341 \xP@definecrayolacolor{White}{0,0,0,0}%
2342 }
2343 \xywithoption{crayon}{%
2344 \xP@installCrayolaColors
2345 \renewcommand*\installCrayolaColors@{}%
2346 }

```

End of the section for Xy-pic's "color" option.

```
2347 \xyendinput
2348 \color
```

8.15 Frames

```
2349 \*frame)
2350 \xyrequire{curve}%
2351 \xycatcodes
```

\xypdf-fr@loaded

```
2352 \expandafter\let\csname xypdf-fr@loaded\endcsname\@empty
```

\xP@framedrop

```
2353 \newcommand*\xP@framedrop[1]{%
2354   \expandafter\frmDrop@\expandafter{%
2355     \expandafter\def\expandafter\prevEdge@@\expandafter{\prevEdge@@}%
2356     #1\frmradius@@}%
2357 }
```

\frm{-}

```
\xP@frm{-} 2358 \xP@hook{frame}{\frm{-}}
2359 \expandafter\newcommand\expandafter*\csname xP@frm{-}\endcsname{%
2360   \xP@framedrop\xP@solidframe
2361 }
```

\xP@solidframe

```
2362 \newcommand*\xP@solidframe[1]{%
2363   \R@#1\relax
2364   \xP@setsolidpat
2365   \let\xP@fillorstroke\xP@stroke
2366   \xP@frameifnotzero\xP@oval
2367 }
```

\xP@frameifnotzero

```
2368 \newcommand*\xP@frameifnotzero[1]{%
2369   \setboxz@h{%
2370     \hskip\X@c\raise\Y@c\hbox{%
2371       \DN@{\zeroEdge}%
2372       \ifx\next@\prevEdge@@
2373       \else
2374       #1%
2375       \fi
2376     }%
2377   }%
2378   \wd\z@\z@\ht\z@\z@\dp\z@\z@
2379   \boxz@
2380 }
```

\xP@oval

```
2381 \newcommand*\xP@oval{%
2382   \hskip-\L@c
2383   \lower\D@c\hbox{%
2384     \dimen@ \dimexpr\L@c+\R@c\relax
2385     \dimen@ii \dimexpr\U@c+\D@c\relax
2386     \R@\xP@min\R@{.5\dimen@}%
2387     \R@\xP@min\R@{.5\dimen@ii}%

```

Circumference. $696621973/405764219 \approx 8 - 2\pi$

```
2388 \tempdimb\dimexpr2\dimen@+2\dimen@ii-\R@*696621973/405764219\relax
2389 \ifdim\R@=\z@
```

Sharp corners: draw a rectangle.

```
2390 \xP@filllorstroke{0 0 \xP@coor\dimen@\dimen@ii re}%
2391 \else
```

Rounded corners.

```
2392 \def\@tempa{*119763188/267309217}%
2393 \xP@filllorstroke{%
2394 \xP@dim\R@0 m \xP@dim{\R@\@tempa}0 0 \xP@dim{\R@\@tempa}%
2395 0 \xP@dim\R@c %
2396 \ifdim2\R@=\dimen@ii\else
2397 0 \xP@dim{\dimen@ii-\R@}l %
2398 \fi
2399 0 \xP@dim{\dimen@ii-\R@\@tempa}\xP@coor{\R@\@tempa}\dimen@ii
2400 \xP@coor\R@\dimen@ii c %
2401 \ifdim2\R@=\dimen@\else
2402 \xP@coor{\dimen@-\R@}\dimen@ii l %
2403 \fi
2404 \xP@coor{\dimen@-\R@\@tempa}\dimen@ii
2405 \xP@coor\dimen@{\dimen@ii-\R@\@tempa}\xP@coor\dimen@{\dimen@ii-\R@} c %
2406 \ifdim2\R@=\dimen@ii\else
2407 \xP@coor\dimen@\R@ l %
2408 \fi
2409 \xP@coor\dimen@{\R@\@tempa}\xP@dim{\dimen@-\R@\@tempa}0 %
2410 \xP@dim{\dimen@-\R@}0 c h%
2411 }%
2412 \fi
2413 }%
2414 }
```

```
\frm[o]{-}
\xP@frm[o]{-} 2415 \xP@hook{frame}{frm[o]{-}}
2416 \expandafter\newcommand\expandafter*{csname xP@frm[o]{-}\endcsname{%
2417 \xP@framedrop{\xP@ellipseframe\xP@setsolidpat}%
2418 }
```

\xP@ellipseframe

```
2419 \newcommand*\xP@ellipseframe[2]{%
2420 \xP@getradii{#2}%
2421 \DN@{\zeroEdge}%
2422 \ifx\next@\prevEdge@@
2423 \else
2424 \def\xP@filllorstroke{#1\xP@stroke}%
2425 \setboxz@h{\hskip\X@c\raise\Y@c\hbox{\xP@framedellipse}}%
2426 \wd\z@=\z@ht\z@=\z@dp\z@=\z@
2427 \boxz@
2428 \fi
2429 }
```

```
\frm{.}
\xP@frm{.} 2430 \xP@hook{frame}{frm{.}}
2431 \expandafter\newcommand\expandafter*{csname xP@frm{.}\endcsname{%
2432 \xP@framedrop\xP@rectframedotted
2433 }
```

\xP@rectframedotted

```

2434 \newcommand*\xP@rectframedotted[1]{%
2435   \R@#1\relax
2436   \xP@frameifnotzero{%
2437     \ifdim\R@=\z@
2438       \xP@dottedrect
2439     \else
2440       \xP@dottedoval
2441     \fi
2442   }%
2443 }

```

\xP@dottedrect Make sure that there is a dot in every corner of the rectangle.

```

2444 \newcommand*\xP@dottedrect{%
2445   \hskip-\L@c
2446   \lower\D@c\hbox{%
2447     \dimen@ii\dimexpr\U@c+\D@c\relax
2448     \@tempdimc\dimexpr\xP@preclw/-2\relax
2449     \@tempdimb\dimexpr\L@c+\R@c+\xP@preclw\relax
2450     \xP@contfalse
2451     \xP@setdottedpat

```

Draw the horizontal lines $\frac{1}{2}$ whitespace longer to eliminate inaccuracies.

```

2452     \dimen@\dimexpr\@tempdimb+\@tempdima/2+\@tempdimc\relax
2453     \xP@stroke{\xP@dim\@tempdimc 0 \xP@dim\dimen@ 1 %
2454       \xP@coor\@tempdimc\dimen@ii m \xP@coor\dimen@\dimen@ii 1}%

```

Ensure pattern continuation.

```

2455     \let\xP@testcont\xP@alwaysconttrue
2456     \@tempdimb\dimen@ii
2457     \xP@setdottedpat
2458     \dimen@\dimexpr\L@c+\R@c\relax
2459     \advance\dimen@ii\dimexpr\@tempdimc-\@tempdima/2\relax
2460     \multiply\@tempdimc\m@ne

```

Draw the vertical lines $\frac{1}{2}$ whitespace longer.

```

2461     \xP@stroke{0 \xP@dim\@tempdimc m 0 \xP@dim\dimen@ii 1 %
2462       \xP@coor\dimen@\@tempdimc m \xP@coor\dimen@\dimen@ii 1}%
2463   }%
2464 }

```

\xP@dottedoval

```

2465 \newcommand*\xP@dottedoval{%
2466   \def\xP@fillorstroke{\xP@setcldottedpat\xP@stroke}%
2467   \xP@oval
2468 }

```

\frm[o]{.}

```

\xP@frm[o]{.} 2469 \xP@hook{frame}{frm[o]{.}}
2470 \expandafter\newcommand\expandafter*\csname xP@frm[o]{.}\endcsname{%
2471   \xP@framedrop{\xP@ellipseframe\xP@setcldottedpat}%
2472 }

```

\frm{-}

```

\xP@frm{-} 2473 \xP@hook{frame}{frm{-}}
2474 \expandafter\newcommand\expandafter*\csname xP@frm{-}\endcsname{%
2475   \xP@framedrop\xP@rectframedashed
2476 }

```



```

\xP@rectframedashed
2477 \newcommand*\xP@rectframedashed[1]{%
2478   \R@#1\relax
2479   \xP@frameifnotzero{%
2480     \ifdim\R@=\z@
2481       \xP@dashedrect
2482     \else
2483       \xP@dashedoval
2484     \fi
2485   }%
2486 }

\xP@dashedrect
2487 \newcommand*\xP@dashedrect{%
2488   \hskip-\L@c
2489   \lower\D@c\hbox{%
2490     \dimen@ \dimexpr\L@c+\R@c\relax
2491     \dimen@ii \dimexpr\U@c+\D@c\relax
2492     \@tempdimb\dimen@
2493     \xP@contfalse
2494     \xP@setdashpat
2495     \xP@stroke{0 0 m \xP@dim\dimen@0 1 %
2496       0 \xP@dim\dimen@ii m \xP@coor\dimen@\dimen@ii l}%
2497     \@tempdimb\dimen@ii
2498     \xP@setdashpat
2499     \xP@stroke{0 0 m 0 \xP@dim\dimen@ii 1 %
2500       \xP@dim\dimen@0 m \xP@coor\dimen@\dimen@ii l}%
2501   }%
2502 }

\xP@dashedoval
2503 \newcommand*\xP@dashedoval{%
2504   \def\xP@fillorstroke{\xP@setdashpat\xP@stroke}%
2505   \xP@oval
2506 }

\frm[o]{-}
\xP@frm[o]{-} 2507 \xP@hook{frame}{\frm[o]{--}}
2508 \expandafter\newcommand\expandafter*\csname xP@frm[o]{--}\endcsname{%
2509   \xP@framedrop{\xP@ellipseframe\xP@setdashpat}%
2510 }

\frm{,}
\xP@frm{,} 2511 \xP@hook{frame}{\frm{,}}
2512 \expandafter\newcommand\expandafter*\csname xP@frm{,}\endcsname{%
2513   \xP@framedrop\xP@frameshadow
2514 }

\xP@frameshadow
2515 \newcommand*\xP@frameshadow[1]{%
2516   \R@#1\relax
2517   \ifdim\R@=\z@\R@1.2pt\relax\fi
2518   \xP@frameifnotzero\xP@shadow
2519 }

```

\xP@shadow

```
2520 \newcommand*\xP@shadow{%
2521   \hskip\dimexpr\R@c+\R@/2\relax
2522   \lower\dimexpr\D@c+\R@/2\relax\hbox{%
2523     \def\xP@pattern{0 J 0 j [] 0 d}%
2524     \edef\xP@lw{\xP@dim\R@}%
2525     \xP@stroke{\xP@dim{\R@/2-\L@c-\R@c} 0 m 0 0 1 0 \xP@dim{\D@c+\U@c-\R@/2}1}%
2526   }%
2527 }
```

\frm{o-}

```
\xP@frm{o-} 2528 \xP@hook{frame}{\frm{o-}}
2529 \expandafter\newcommand\expandafter*\csname xP@frm{o-}\endcsname{%
2530   \xP@framedrop\xP@roundedrectframe
2531 }
```

\xP@roundedrectframe

```
2532 \newcommand*\xP@roundedrectframe[1]{%
2533   \R@#1\relax
2534   \ifdim\R@=\z@\R@\xydashl@ \relax \fi
2535   \xP@frameifnotzero\xP@roundedrectangle
2536 }
```

\xP@roundedrectangle

```
2537 \newcommand*\xP@roundedrectangle{%
2538   \dimen@ \dimexpr\L@c+\R@c\relax
2539   \dimen@ii \dimexpr\U@c+\D@c\relax
2540   \R@\xP@min\R@{.5\dimen@}%
2541   \R@\xP@min\R@{.5\dimen@ii}%
2542   \hskip-\L@c
2543   \lower\D@c\hbox{%
Rounded corners
2544     \@tempdimb \dimexpr2\dimen@+2\dimen@ii-\R@*696621973/405764219\relax
2545     \def\@tempa{*119763188/267309217}%
2546     \xP@setsolidpat
2547     \xP@stroke{%
2548       \xP@dim\R@ 0 m \xP@dim{\R@\@tempa} 0 0 \xP@dim{\R@\@tempa}%
2549       0 \xP@dim\R@ c %
2550       \ifdim2\R@=\dimen@ii\else
2551         0 \xP@dim{\dimen@ii-\R@}m %
2552       \fi
2553       0 \xP@dim{\dimen@ii-\R@\@tempa}\xP@coor{\R@\@tempa}\dimen@ii
2554       \xP@coor\R@\dimen@ii c %
2555       \ifdim2\R@=\dimen@\else
2556         \xP@coor{\dimen@-\R@}\dimen@ii m %
2557       \fi
2558       \xP@coor{\dimen@-\R@\@tempa}\dimen@ii
2559       \xP@coor\dimen@\{\dimen@ii-\R@\@tempa}\xP@coor\dimen@\{\dimen@ii-\R@} c %
2560       \ifdim2\R@=\dimen@ii\else
2561         \xP@coor\dimen@\R@ m %
2562       \fi
2563       \xP@coor\dimen@\{\R@\@tempa}\xP@dim{\dimen@-\R@\@tempa} 0 %
2564       \xP@dim{\dimen@-\R@} 0 c%
2565     }%
  }
```

Upper and lower horizontal dashes.

```

2566 \xP@contfalse
2567 \@tempdimb\dimexpr\L@c+\R@c-2\R@\relax
2568 \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
2569 \xP@setdashpat
2570 \ifdim\@tempdima>\z@
2571 \dimen@ \dimexpr\@tempdimb+\R@-\@tempdima/2\relax
2572 \dimen@ii \dimexpr\U@c+\D@c\relax
2573 \xP@stroke{%
2574 \xP@dim{\R@+\@tempdima}0 m \xP@dim\dimen@ 0 1 %
2575 \xP@coord{\R@+\@tempdima}\dimen@ii m \xP@coord\dimen@\dimen@ii 1%
2576 }%
2577 \fi

```

Left and right vertical dashes.

```

2578 \@tempdimb\dimexpr\U@c+\D@c-2\R@\relax
2579 \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
2580 \xP@setdashpat
2581 \ifdim\@tempdima>\z@
2582 \dimen@ \dimexpr\L@c+\R@c\relax
2583 \dimen@ii \dimexpr\@tempdimb+\R@-\@tempdima/2\relax
2584 \xP@stroke{%
2585 0 \xP@dim{\R@+\@tempdima}m 0 \xP@dim\dimen@ii 1 %
2586 \xP@coord\dimen@{\R@+\@tempdima}m \xP@coord\dimen@\dimen@ii 1%
2587 }%
2588 \fi
2589 }%
2590 }

```

```

\frm{=}
\xP@frm{=} 2591 \xP@hook{frame}{frm{=}}
2592 \expandafter\newcommand\expandafter*\csname xP@frm{=}\endcsname{%
2593 \xP@framedrop\xP@dsframe
2594 }

```

```

\xP@dsframe
2595 \newcommand*\xP@dsframe[1]{%
2596 \R@#1\relax
2597 \xP@frameifnotzero\xP@dsoval
2598 }

```

```

\xP@dsoval
2599 \newcommand*\xP@dsoval{%
2600 \dimen@ \dimexpr(\L@c+\R@c)/2\relax
2601 \ifdim\dimen@<\xydashh@\dimen@\xydashh@\fi
2602 \dimen@ii \dimexpr(\U@c+\D@c)/2\relax
2603 \ifdim\dimen@ii<\xydashh@\dimen@ii\xydashh@\fi
2604 \R@\xP@min\R@\dimen@
2605 \R@\xP@min\R@\dimen@ii
2606 \xP@setsolidpat
2607 \let\xP@fillorstroke\xP@stroke
2608 \xP@oval
2609 \hskip\L@c
2610 \advance\L@c-\xydashh@
2611 \advance\R@c-\xydashh@
2612 \advance\U@c-\xydashh@

```

```

2613 \advance\D@c-\xydashh@
2614 \advance\R@-\xydashh@
2615 \ifdim\R@<\z@\R@\z@\fi
2616 \xP@oval
2617 }

\frm[o]{=}
\xP@frm[o]{=} 2618 \xP@hook{frame}{frm[o]{=}}
2619 \expandafter\newcommand\expandafter*\csname xP@frm[o]{=}\endcsname{%
2620 \xP@framedrop\xP@dsellframe
2621 }

\frm{ee}
\xP@frm{ee} 2622 \xP@hook{frame}{frm{ee}}
2623 \expandafter\newcommand\expandafter*\csname xP@frm{ee}\endcsname{%
2624 \xP@framedrop\xP@dsellframe
2625 }

\xP@dsellframe
2626 \newcommand*\xP@dsellframe[1]{%
2627 \xP@getradii{#1}%
2628 \xP@frameifnotzero\xP@dsellipse
2629 }

\xP@temppath
2630 \@ifdefinable\xP@temppath\relax

\xP@dsellipse
2631 \newcommand*\xP@dsellipse{%
2632 \hskip\dimexpr(\R@c-\L@c)/2\relax
2633 \lower\dimexpr(\D@c-\U@c)/2\relax
2634 \hbox{%
2635 % Inner ellipse: true ellipse
2636 \advance\A@-\xydashh@
2637 \advance\B@-\xydashh@
2638 \ifdim\A@<\z@\A@\z@\fi
2639 \ifdim\B@<\z@\B@\z@\fi
2640 \def\xP@fillorstroke{\edef\xP@temppath}%
2641 \xP@ellipse@
2642 % Outer curve: offset ellipse
2643 \xP@inibigdim
2644 \let\@tempa\xydashh@
2645 \xP@offsetellipse
2646 \xP@setsolidpat
2647 \xP@stroke{\xP@temppath\space\the\@temptokena h}%
2648 }%
2649 }

\xP@offsetellipse TeX grouping: Not necessary, it's in an \hbox anyway.
2650 \newcommand*\xP@offsetellipse{%
2651 \xP@movetotruer
2652 \@temptokena{}%
2653 \xP@offsetelliptseg\A@\z@\A@{\B@*173517671/654249180}%
2654 {\A@*554561898/619869377}{\B@*34221476/65864945}%
2655 {\A@*543339720/768398401}{\B@*543339720/768398401}%
2656 \xP@offsetelliptseg{\A@*543339720/768398401}{\B@*543339720/768398401}%

```

```

2657     {\A@*34221476/65864945}{\B@*554561898/619869377}}%
2658     {\A@*173517671/654249180}\B@z@B@
2659     \xP@mirrorpath
2660 }

```

\xP@mirrorpath

```

2661 \newcommand*\xP@mirrorpath{%
2662   \edef\@tempa{\the\@temptokena\relax\space\space\space\space}%
2663   \let\@tempb\empty
2664   \let\@tempc\empty
2665   \expandafter\xP@mirrorpath@\@tempa
2666 }

```

\xP@mirrorpath@

```

2667 \@ifdefinable\xP@mirrorpath@\relax
2668 \def\xP@mirrorpath@#1 #2 #3 #4 #5 #6 #7 {%
2669   \ifx\relax#4%
2670     \xP@append\@temptokena{\@tempb\xP@minus#1 \xP@minus#2 #3 \@tempc h}%
2671   \else
2672     \edef\@tempb{\xP@minus#6 #7 \xP@minus#4 #5 \xP@minus#1 #2 c \@tempb%
2673       \if#3m\else\xP@minus#1 \xP@minus#2 #3 \fi%
2674       \xP@minus#4 \xP@minus#5 \xP@minus#6 \xP@minus#7 }%
2675     \edef\@tempc{#6 \xP@minus#7 #4 \xP@minus#5 #1 \xP@minus#2 c \@tempc}%
2676     \expandafter\xP@mirrorpath@
2677   \fi
2678 }

```

\xP@minus

```

2679 \@ifdefinable\xP@minus\relax
2680 \def\xP@minus#1 {\if-#1 \else\ifdim\dimexpr#1pt\relax=z@\else-\fi#1 \fi}

```

\xP@insertbefore

```

2681 \newcommand*\xP@insertbefore[2]{\{\edef\@tempa{#1{#2\the#1}}\expandafter}\@tempa}

```

\xP@offsetelliptseg

```

2682 \newcommand*\xP@offsetelliptseg[8]{%
2683   \X@p\dimexpr#1\relax
2684   \Y@p\dimexpr#2\relax
2685   \L@c\dimexpr#3\relax
2686   \U@c\dimexpr#4\relax
2687   \R@c\dimexpr#5\relax
2688   \D@c\dimexpr#6\relax
2689   \X@c\dimexpr#7\relax
2690   \Y@c\dimexpr#8\relax
2691   \xP@savepts
2692   \xP@a\z@
2693   \xP@c\xP@bigdim
2694   \xP@paintsolid@
2695 }

```

\xP@getradii

```

2696 \newcommand*\xP@getradii[1]{%
2697   \edef\@tempa{#1}%
2698   \expandafter\xP@getradii@\@tempa,\maxdimen,%
2699 }

```

```

\xP@getradii@
2700 \@ifdefinable\xP@getradii@\relax
2701 \def\xP@getradii@#1,#2,#3@{%
2702   \A@#1\relax
2703   \B@#2\relax
2704   \ifdim\B@=\maxdimen
2705     \A@\dimexpr(\L@c+\R@c)/2\relax
2706     \B@\dimexpr(\U@c+\D@c)/2\relax
2707   \fi
2708 }

\frm{o}
\xP@frm{o} 2709 \xP@hook{frame}{frm{o}}
2710 \expandafter\newcommand\expandafter*\csname xP@frm{o}\endcsname{%
2711   \xP@framedrop{\xP@circleframe\xP@setsolidpat}%
2712 }

\frm{-o}
\xP@frm{-o} 2713 \xP@hook{frame}{frm{-o}}
2714 \expandafter\newcommand\expandafter*\csname xP@frm{-o}\endcsname{%
2715   \xP@framedrop{\xP@circleframe\xP@setcldashpat}%
2716 }

\frm{.o}
\xP@frm{.o} 2717 \xP@hook{frame}{frm{.o}}
2718 \expandafter\newcommand\expandafter*\csname xP@frm{.o}\endcsname{%
2719   \xP@framedrop{\xP@circleframe\xP@setcldottedpat}%
2720 }

\xP@circleframe
2721 \newcommand*\xP@circleframe[2]{%
2722   \R@#2\relax
2723   \def\xP@fillorstroke{#1\xP@stroke}%
2724   \DN@{\zeroEdge}%
2725   \ifx\next@\prevEdge@@
2726     \ifdim\R@>\z@
2727       \xP@circleframe@
2728     \fi
2729   \else
2730     \ifdim\R@=\z@
2731       \A@\dimexpr(\L@c+\R@c)/2\relax
2732       \B@\dimexpr(\U@c+\D@c)/2\relax
2733       \R@\xP@max\A@\B@
2734     \fi
2735     \xP@circleframe@
2736   \fi
2737 }

\xP@circleframe@
2738 \newcommand*\xP@circleframe@{%
2739   \setboxz@h{\hskip\X@c\raise\Y@c\hbox{\xP@circle}}%
2740   \wd\z@=\z@\ht\z@=\z@\dp\z@=\z@
2741   \boxz@
2742 }

```

```

\frm{e}
\XP@frm{e} 2743 \XP@hook{frame}{frm{e}}
2744 \expandafter\newcommand\expandafter*\csname XP@frm{e}\endcsname{%
2745 \XP@framedrop{\XP@ellipseframe\XP@setsolidpat}%
2746 }

\frm{-e}
\XP@frm{-e} 2747 \XP@hook{frame}{frm{-e}}
2748 \expandafter\newcommand\expandafter*\csname XP@frm{-e}\endcsname{%
2749 \XP@framedrop{\XP@ellipseframe\XP@setcldashpat}%
2750 }

\frm{.e}
\XP@frm{.e} 2751 \XP@hook{frame}{frm{.e}}
2752 \expandafter\newcommand\expandafter*\csname XP@frm{.e}\endcsname{%
2753 \XP@framedrop{\XP@ellipseframe\XP@setcldottedpat}%
2754 }

\frm2{.e}
\XP@frm2{.e} 2755 \XP@hook{frame}{frm2{.e}}
2756 \expandafter\newcommand\expandafter*\csname XP@frm2{.e}\endcsname{%
2757 \XP@framedrop\XP@dsdottedellframe
2758 }

\XP@dsdottedellframe
2759 \newcommand*\XP@dsdottedellframe[1]{%
2760 \XP@getradii{#1}%
2761 \XP@frameifnotzero\XP@dsdottedellipse
2762 }

\XP@dsdottedellipse
2763 \newcommand*\XP@dsdottedellipse{%
2764 \hskip\dimexpr(\R@c-\L@c)/2\relax
2765 \lower\dimexpr(\D@c-\U@c)/2\relax
2766 \hbox{%
Intermediate ellipse: true ellipse
2767 \@tempdima.5\xydashhh@\relax
2768 \advance\A@-\@tempdima
2769 \advance\B@-\@tempdima
2770 \ifdim\A@<\@tempdima\A@\@tempdima\fi
2771 \ifdim\B@<\@tempdima\B@\@tempdima\fi
2772 \let\XP@normalmult\@ne
2773 \XP@specialellipse{\XP@splinemultdotted\XP@doublestroke}%
2774 }%
2775 }

\XP@specialellipse
2776 \newcommand*\XP@specialellipse[1]{%
2777 \def\@tempa{*147546029/267309217}%
2778 \Xp\A@
2779 \Yp\z@
2780 \Lc\A@
2781 \Uc\dimexpr\B@\@tempa\relax
2782 \Rc\dimexpr\A@\@tempa\relax
2783 \Dc\B@

```

```

2784 \X@c\z@
2785 \Y@c\B@
2786 \xP@bezierlength
2787 \let\xP@testcont\xP@alwaysconttrue
2788 #1%
2789 \X@p\z@
2790 \Y@p\B@
2791 \L@c-\R@c
2792 \D@c\U@c
2793 \U@c\B@
2794 \R@c-\A@
2795 \X@c-\A@
2796 \Y@c\z@
2797 #1%
2798 \X@p-\A@
2799 \Y@p\z@
2800 \R@c\L@c
2801 \L@c-\A@
2802 \U@c-\D@c
2803 \D@c-\B@
2804 \X@c\z@
2805 \Y@c-\B@
2806 #1%
2807 \X@p\z@
2808 \Y@p-\B@
2809 \L@c-\R@c
2810 \D@c\U@c
2811 \U@c-\B@
2812 \R@c\A@
2813 \X@c\A@
2814 \Y@c\z@
2815 #1%
2816 }

```

\xP@alwaysconttrue

```
2817 \newcommand*\xP@alwaysconttrue[1]{\xP@splineconttrue}
```

\frm2{-e}

```
\xP@frm2{-e} 2818 \xP@hook{frame}{frm2{-e}}
```

```
2819 \expandafter\newcommand\expandafter*\csname xP@frm2{-e}\endcsname{%
```

```
2820 \xP@framedrop\xP@dsdashedframe
```

```
2821 }
```

\xP@dsdashedframe

```
2822 \newcommand*\xP@dsdashedframe[1]{%
```

```
2823 \xP@getradii{#1}%
```

```
2824 \xP@frameifnotzero\xP@dsdashedellipse
```

```
2825 }
```

\xP@dsdashedellipse

```
2826 \newcommand*\xP@dsdashedellipse{%
```

```
2827 \hskip\dimexpr(\R@c-\L@c)/2\relax
```

```
2828 \lower\dimexpr(\D@c-\U@c)/2\relax
```

```
2829 \hbox{%
```

Inner ellipse: true ellipse


```

2830 \advance\A@-\xydashh@
2831 \advance\B@-\xydashh@
2832 \ifdim\A@<\z@\A@\z@\fi
2833 \ifdim\B@<\z@\B@\z@\fi
2834 \xP@specialellipse{\xP@splinemulldashed\xP@elldoublestroke}%
2835 }%
2836 }

\xP@elldoublestroke
2837 \newcommand*\xP@elldoublestroke{\z@,\xydashh@}

\xP@fill
2838 \newcommand*\xP@fill[1]{\xP@literal{#1 f}}

\xP@fillstroke
2839 \newcommand*\xP@fillstroke[1]{\xP@literal{\xP@dim{\xP@preclw/2}w 1 j 0 G #1 b}}

\xP@fillorstroke
2840 \newcommand*\xP@fillorstroke{

  \frm{*}
\xP@frm{*} 2841 \xP@hook{frame}{frm{*}}
2842 \expandafter\newcommand\expandafter* \csname xP@frm{*}\endcsname{%
2843 \xP@framedrop{\let\xP@fillorstroke\xP@fill\xP@framefill}%
2844 }

  \frm{**}
\xP@frm{**} 2845 \xP@hook{frame}{frm{**}}
2846 \expandafter\newcommand\expandafter* \csname xP@frm{**}\endcsname{%
2847 \xP@framedrop{\let\xP@fillorstroke\xP@fillstroke\xP@framefill}%
2848 }

\xP@framefill
2849 \newcommand*\xP@framefill[1]{%
2850 \R@#1\relax
2851 \xP@setsolidpat
2852 \setboxz@h{%
2853 \hskip\X@c\raise\Y@c\hbox{%
2854 \DN@{\rectangleEdge}%
2855 \ifx\next@\prevEdge@@
2856 \DN@{\xP@oval}%
2857 \else
2858 \DN@{\circleEdge}%
2859 \ifx\next@\prevEdge@@
2860 \ifdim\R@=\z@
2861 \DN@{\xP@filledellipse}%
2862 \else
2863 \DN@{\restR@max\xP@circle}%
2864 \fi
2865 \else
2866 \ifdim\R@=\z@
2867 \DN@{\xP@oval}%
2868 \else
2869 \DN@{\xP@circle}%
2870 \fi
2871 \fi

```

```

2872      \fi
2873      \next@
2874    }%
2875  }%
2876  \wd\z@\z@\ht\z@\z@\dp\z@\z@
2877  \boxz@
2878 }

\XP@circle
2879 \newcommand*\XP@circle{%
2880   \XP@ellipse\R@\R@
2881 }

\XP@filledellipse
2882 \newcommand*\XP@filledellipse{%
2883   \XP@ellipse{\dimexpr(\L@c+\R@c)/2\relax}{\dimexpr(\U@c+\D@c)/2\relax}%
2884 }

\XP@framedellipse
2885 \newcommand*\XP@framedellipse{%
2886   \XP@ellipse\A@\B@
2887 }

\XP@ellipse
2888 \newcommand*\XP@ellipse[2]{%
2889   \hskip\dimexpr(\R@c-\L@c)/2\relax
2890   \lower\dimexpr(\D@c-\U@c)/2\relax
2891   \hbox{%
2892     \A#1\relax
2893     \B#2\relax
2894     \XP@ellipse@
2895   }%
2896 }

\XP@ellipse@
2897 \newcommand*\XP@ellipse@{%
  Perimeter, second segment
2898   \X@p\dimexpr\A@*543339720/768398401\relax
2899   \Y@p\dimexpr\B@*543339720/768398401\relax
2900   \L@c\dimexpr\A@*34221476/65864945\relax
2901   \U@c\dimexpr\B@*554561898/619869377\relax
2902   \R@c\dimexpr\A@*173517671/654249180\relax
2903   \D@c\B@
2904   \X@c\z@
2905   \Y@c\B@
2906   \XP@bezierlength
2907   \@tempdima\@tempdimb
  Perimeter, first segment
2908   \X@p\A@
2909   \Y@p\z@
2910   \L@c\A@
2911   \U@c\dimexpr\B@*173517671/654249180\relax
2912   \R@c\dimexpr\A@*554561898/619869377\relax
2913   \D@c\dimexpr\B@*34221476/65864945\relax
2914   \X@c\dimexpr\A@*543339720/768398401\relax

```

```

2915 \Y@c\dimexpr\B@*543339720/768398401\relax
2916 \xP@bezierlength
2917 \@tempdimb4\dimexpr\@tempdima+\@tempdimb\relax
2918 \edef\@tempa{%
2919   \xP@dim\A@0 m
2920   \xP@coor\L@c\U@c
2921   \xP@coor\R@c\D@c
2922   \xP@coor\X@c\Y@c c %
2923   \xP@coor{\A@*34221476/65864945}{\B@*554561898/619869377}%
2924   \xP@coor{\A@*173517671/654249180}\B@
2925   0 \xP@dim\B@ c }%
2926 \@temptokena\expandafter{\@tempa}%
2927 \xP@mirrorpath
2928 \xP@fillorstroke{\the\@temptokena}%
2929 }

```

End of the section for Xy-pic's "frame" option.

```

2930 \xyendinput
2931 </frame>
2932 %
2933 % \subsection{Line styles}
2934 %
2935 % \begin{macrocode}
2936 <*line>
2937 \xycatcodes
2938 \expandafter\let\csname xypdf-li@loaded\endcsname\@empty
2939 % Dummy file.

```

End of the section for Xy-pic's "line" option.

```

2940 \xyendinput
2941 </line>
2942 <*basic>

```

Finish the package initialization. The \xywithoption commands are wrapped into \next@ so that they cannot change the catcodes for the next \xywithoption command.

```

2943 \let\@tempa\@undefined
2944 \let\@nextii\@undefined
2945 \DN@{%
2946   \xywithoption{color}{%
2947     \message{Xy-pic pdf driver: 'color' extension support}%
2948     \@ifundefined{xypdf-co@loaded}{\input xypdf-co\relax}{\message{not reloaded}}%
2949   }%
2950   \xywithoption{curve}{%
2951     \message{Xy-pic pdf driver: 'curve' extension support}%
2952     \@ifundefined{xypdf-cu@loaded}{\input xypdf-cu\relax}{\message{not reloaded}}%
2953   }%
2954   \xywithoption{frame}{%
2955     \message{Xy-pic pdf driver: 'frame' extension support}%
2956     \@ifundefined{xypdf-fr@loaded}{\input xypdf-fr\relax}{\message{not reloaded}}%
2957   }%
2958   \xywithoption{line}{%
2959     \message{Xy-pic pdf driver: 'line' extension support}%
2960     \@ifundefined{xypdf-li@loaded}{\input xypdf-li\relax}{\message{not reloaded}}%
2961   }%
2962   \xywithoption{rotate}{%
2963     \message{Xy-pic pdf driver: 'rotate' extension support}%
2964     \@ifundefined{xypdf-ro@loaded}{\input xypdf-ro\relax}{\message{not reloaded}}%

```

```

2965 }%
2966 }
2967 \next@
2968 \xyendinput
2969 \</basic>

```

9 Changelog

v1.0 2010/03/24

Initial version

v1.1 2010/03/30

- Added support for the $\text{\texttt{Xy-pic}}$ “rotate” extension.
- The parts of the style file dealing with $\text{\texttt{Xy-pic}}$ extensions (currently “curve” and “rotate”) are only executed when those extension were loaded.
- `xypdf` does not give an error message when used with $\text{\texttt{Xy-pic}}$ options which query the Postscript drivers (e. g. “all” or “color”).
- In DVI mode, a warning is issued that the DVI file is not portable, like $\text{\texttt{Xy-pic}}$ does when a Postscript driver is in use.

v1.2 2010/04/08

- Improved precision and numerical stability for the offset algorithm around cusps.
- Improved slide algorithm `\xP@slide@`
- Respect `\pdfdecimaldigits` when dimensions are written to the PDF file.
- Correct continuation for dashed/dotted/squiggled curves consisting of more than one segment.
- Code cleanup

v1.3 2010/04/12

- Bug fix: No “Extra `\fi`” if `\ifpdfabsdim` is not defined.
- Bug fix: Moved the code for the spline continuation out of the optional section for curves since it is also needed for straight lines.
- Check the version of pdf $\text{\texttt{TeX}}$ since `\pdfsave` is not defined prior to pdf $\text{\texttt{TeX}}$ 1.40.0.
- “**Troubleshooting**” paragraph for $\text{\texttt{TeX}}$ Live without the ε - $\text{\texttt{TeX}}$ features enabled.
- Generic PDF code for the `{-}` directional object.

v1.4 2010/05/13

- Support for both plain $\text{\texttt{TeX}}$ and $\text{\texttt{L\texttt{A}TeX}}$, reorganization of the code and splitting into several files. The $\text{\texttt{L\texttt{A}TeX}}$ style file `xypdf.sty` has been replaced by `xypdf.tex`, which is recognized by $\text{\texttt{Xy-pic}}$ as a driver.
- Integration into the $\text{\texttt{Xy-pic}}$ distribution.
- Support for the “color” and “frame” extensions.
- New supported curve style `{~~}` (broken squiggled curves).