

The xypdf package

Daniel Müllner

1.5, dated 2010/10/11

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 \TeX and \LaTeX in the occurrences of \pdf\TeX , \Xe\TeX and $\varepsilon\text{-}\text{\TeX}$ 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 \TeX and \LaTeX 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 \TeX 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 \pdf\TeX . 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 \TeX 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 PDF driver provided by the `xypdf` package adapts the output routines of the `Xy-pic` package to generate high-quality graphics for PDF output. It works with both \pdf\TeX and the two-step compilation $\text{\TeX} \rightarrow \text{\dvipdfm(x)}$ with an intermediate DVI file. Thus, it also works with \Xe\TeX since this program internally uses a modification of `dvipdfmx`. Note that some version of $\varepsilon\text{-}\text{\TeX}$ is needed (which is anyway used by default in modern \TeX installations). **Figure 1** compares the output quality of a small `Xy-pic` diagram.

The `xypdf` package is very similar to the Postscript drivers for `Xy-pic`. It does not have (yet) all of their features (see **Section 4**) but is much more powerful in other respects, e. g. when drawing multiple curves. In general, it greatly improves graphics quality. Currently, the following features are implemented:

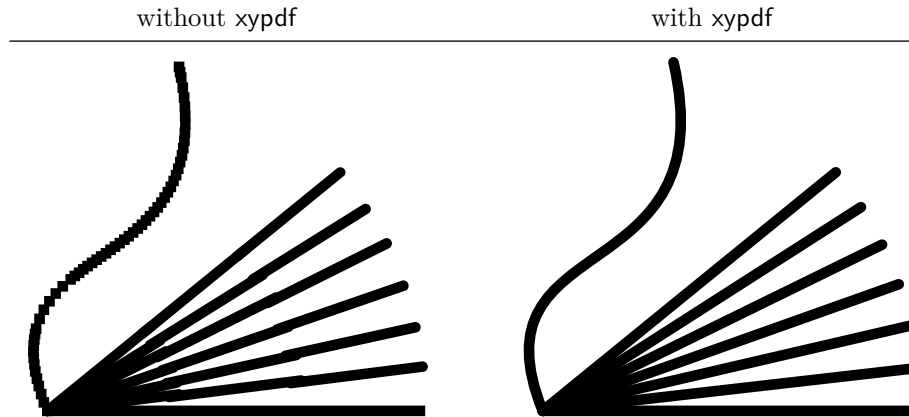
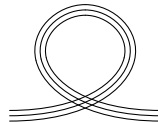
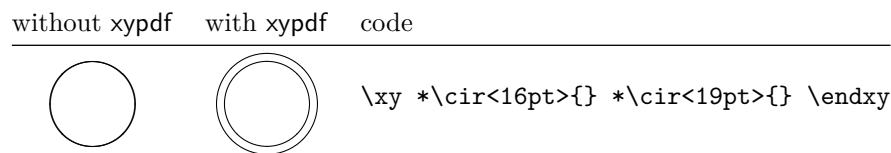


Figure 1: Comparison of X_Y-pic output, magnified 10 times.

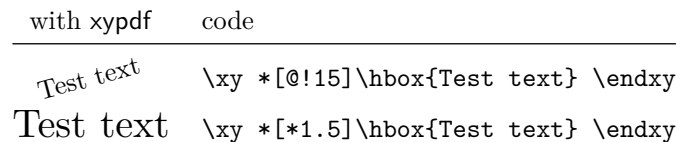
- Both straight lines and curves (solid, dashed, dotted and squiggled) are drawn by generic PDF commands.
- X_Y-pic automatically draws the symbols of which lines and curves are composed at the very beginning and end of a segment. It then distributes the inner symbols evenly across the segment. Since 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 xypdf package handles this better than the original code. Compare the output in [Figure 2](#).
- As a highlight, xypdf features a Bézier curve offset algorithm, producing high-quality curves with two or three parallel strokes.

































- The `\cir` object draws circles of arbitrary radius.



- xypdf supports the “rotate” extension of X_Y-pic.



- xypdf supports the “frame” extension of X_Y-pic. The only intended difference to the Postscript drivers is the appearance of the `{**}` fill style. The postscript drivers fill the frame and stroke it with a line of thickness 0. The PDF driver strokes 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}
\ymatrix{*[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 loaded 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

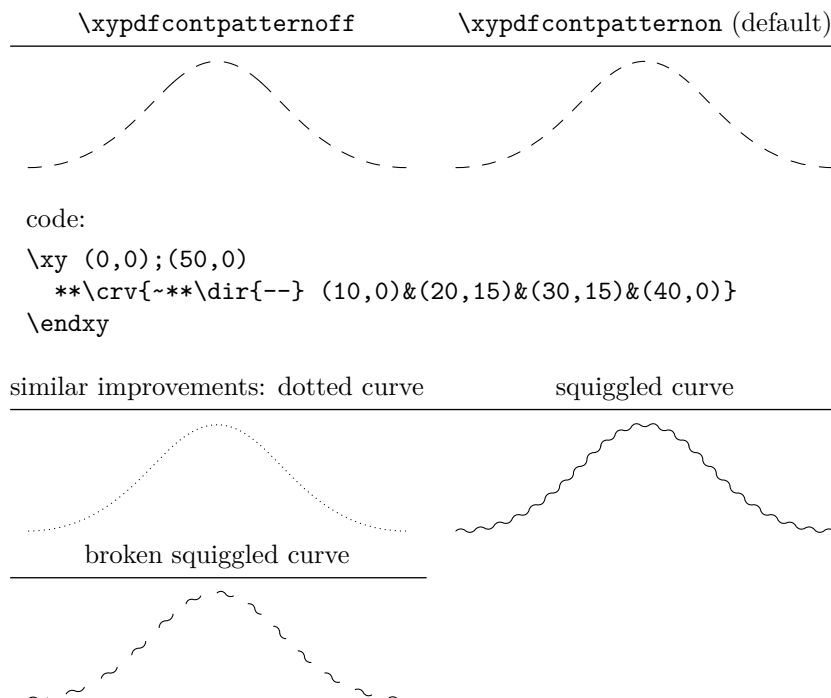
- Support for the “line” extension.

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.



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 pdfT_EX. If you cannot update your T_EX system for some reason, you may still use the `xypdf` package in DVI mode and produce a PDF file via `dvipdfm(x)`. The pathway T_EX → `dvipdfm(x)` is preferable in many cases anyway since it usually produces much smaller PDF files.

- I get the error message **eT_EX is needed for the xypdf package**.

In your T_EX installation, the ε -T_EX features are not enabled, although they most certainly can be in any reasonably modern T_EX installation. You must probably rebuild the T_EX and L^AT_EX format files with ε -T_EX enabled. Please consult the documentation of your T_EX distribution on how to rebuild the 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 $\text{\texttt{Xy-pic}}$.

```
\xpdfversion \xyprovide defines \xpdfversion.
\xpdfdate    1 <*basic>
              2 \ifx\xyloaded\undefined\input xy \fi
              3 \xyprovide{pdf}{PDF driver}{1.5}%
              4 {Daniel M\"ullner\newline}%
              5 {\url{http://www.math.uni-bonn.de/people/muellner}}{}
              6 \ifx\makeatletter\undefined\input miniltx \fi
              7 \newcommand*\xpdfdate{2010/10/11}
              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.

```
\xpdfon      Commands for switching the driver on and off.
\xpdfoff     22 \newcommand*\xpdfon{%
              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*\xpdfoff{%
              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
```

$\text{\texttt{\xP@testpdfsave}}$ Test for $\text{\texttt{\pdfsave}}$, which was introduced in pdf $\text{\texttt{TeX}}$ 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}
90 \newcommand*\xP@cm[5]{%
91   \pdfsave
92   \pdfsetmatrix{##1 ##2 ##3 ##4}%
93   ##5%
94   \pdfrestore
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 {\@makeoother\p\@makeoother\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@}%
277 \def\Connect@@{\solid@}%
278 }

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

\xP@solidSpread
281 \@ifdefinable\xP@solidSpread\relax
282 \def\xP@solidSpread#1\repeat@{ {%
Neglect zero-length lines.
283 \@tempswatrue
284 \ifdim\X@p=\X@c
285 \ifdim\Y@p=\Y@c
286 \@tempswafalse
287 \fi
288 \fi
289 \if@tempswa
290 \xP@setsolidpat
291 \xP@stroke{\xP@coor\X@p\Y@p m \xP@coor\X@c\Y@c l}%
292 \fi
293 }}

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

\xP@setsolidpat Pattern for solid lines
295 \newcommand*\xP@setsolidpat{%
296 \def\xP@pattern{1 J 1 j []0 d}%
297 \global\let\xP@lastpattern\xP@solidmacro
298 }

\xP@stroke
299 \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@ 300 \xP@hook{pdf}{dash@}
301 \newcommand\xP@dashed@{\line@\def\Connect@@{\straight@\xP@dashedSpread}}

```

\xP@dashedSpread

```

302 \@ifdefinable\xP@dashedSpread\relax
303 \def\xP@dashedSpread#1\repeat@{ {%
304   \xP@vecLen
Neglect zero-length lines.
305   \ifdim\@tempdimb>\z@
306     \xP@setdashpat
307     \xP@savec
308     \xP@stroke{\xP@coord\X@p\Y@p m \xP@coord\X@c\Y@c l}%
309   \fi
310 }}
```

\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`.

```

311 \newcommand*\xP@setdashpat{%
312   \xP@testcont\xP@dashmacro
313   \ifxP@splinecont
Special pattern in case this line continues another dashed segment.
314     {\count@ \numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
315     \global\dimen@i
316     \ifnum\count@>\z@
317       \dimexpr\@tempdimb/\count@\relax
318     \else
319       \z@
320     \fi
321   }%
322   \@tempdima\dimen@i
323   \edef\xP@pattern{1 J 1 j [%
324     \ifdim\@tempdima>\z@
325       \xP@precdim\@tempdima]\xP@precdim\@tempdima
326     \else
327       ]0 %
328     \fi
329   d}%
330 \else
331   \@tempdima
332   \ifdim\@tempdimb>\xydashl@
333     \dimexpr\@tempdimb/(2*((\@tempdimb+\xydashl@)/(2*\xydashl@))-1)\relax
334   \else
335     \z@
336   \fi
337   \edef\xP@pattern{1 J 1 j [%
338     \ifdim\@tempdima>\z@\xP@precdim\@tempdima\fi
339   ]0 d}%
340 \fi
341 \global\let\xP@lastpattern\xP@dashmacro
342 }
```

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

```

343 \newcommand*\xP@setcldashpat{%
344   {\count@\numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
345   \xdef\@gtempa{1 J 1 j [\ifnum\count@>\z@\xP@precdim{\@tempdimb/\count@}\fi}%
346   \ifnum\count@>\z@\xP@precdim{\@tempdimb*3/2/\count@}\else0 \fi d}%
347 }%
348 \edef\xP@pattern{\@gtempa}%
349 }

```

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

```

\xP@point@ 350 \xP@hook{pdf}{point@}
351 \newcommand*\xP@point@{\xP@zerodot\egroup\Invisible@false
352   \Hidden@false\def\Leftness@{.5}\def\Upness@{.5}\ctipEdge@
353   \def\Drop@@{\stylenboxz}%
354   \def\Connect@@{\straight@\xP@dottedSpread}%
355 }

```

\xP@zerodot

```

356 \newcommand*\xP@zerodot{%
357   \hb@xt@\z@{\hss
358     \vbox to\z@{\vss\hrule\@width\xP@preclw\@height\xP@preclw\vss}%
359   \hss}%
360 }

```

\xP@dottedSpread

```

361 \@ifdefinable\xP@dottedSpread\relax
362 \def\xP@dottedSpread#1\repeat@{ {%
363   \xP@veclen
364   \ifdim\@tempdimb>\z@
365     \xP@setdottedpat
366     \xP@savvec
367     \xP@stroke{\xP@coor\X@p\Y@p m \xP@coor\X@c\Y@c l}%
368     \fi
369 }}

```

\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.

```

370 \newcommand*\xP@setdottedpat{%
371   \xP@testcont\xP@dotmacro
372   \ifxP@splinecont
373     \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
374     \edef\xP@pattern{%
375       0 J [%

```

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

```

376     \ifdim\@tempdima>\z@
377       \xP@precdim\xP@preclw\xP@precdim\@tempdima
378       \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.)

```

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

```



```

380 \else
381   \advance\@tempdimb-\xP@preclw
382   \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
383   \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
384   \edef\xP@pattern{%
385     0 J [%
Produce a dot pattern only when the segment is long enough.
386     \ifdim\@tempdima>\z@
387       \xP@precdim\xP@preclw\xP@precdim\@tempdima
388       \fi
389     ]0 d}%
390 \fi
391 \global\let\xP@lastpattern\xP@dotmacro
392 }

```

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

```

393 \newcommand*\xP@setcldottedpat{%
394   \@tempdima\dimexpr\@tempdimb/(\@tempdimb/131072+1)-\xP@preclw\relax
395   \edef\xP@pattern{%
396     0 J [%
397     \ifdim\@tempdima>\z@
398       \xP@precdim\xP@preclw\xP@precdim\@tempdima
399       \fi
400     ]\xP@precdim{\dimexpr\xP@preclw+\@tempdima/2\relax}d}%
401 }

```

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

```

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

```

`\xP@ifabsless`

```

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

```

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

```

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

```

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

```

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

```

`\xP@min` Maximum of two lengths

```

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

```

`\xP@max` Maximum of two lengths

```

411 \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.

```

412 \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`.

```

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

```

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

```

416   \unless\ifdim\@tempdimc=\@tempdimb
iterate: (old approx.) := (new approx.)
417     \@tempdimc\@tempdimb\relax
418   \repeat
419 }

```

`\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.)

```

420 \newcommand*\xP@veclen{%
421   \xP@veclen@
422   \global\dimen@i\@tempdimb
423   }\@tempdimb\dimen@i
424 }

```

`\xP@veclen@`

```

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

```

1) Strictly vertical vector

```

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

```

2) Strictly horizontal vector

```

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

```

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

```

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

```

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

```

438     \count@\@ne
439     \loop
440       \@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.

```

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

```

```

446         \multiply\count@\@tempcnta
447         \repeat
Starting value for the square root algorithm
448         \@tempdimc\dimexpr(\d@X+\d@Y)*3/4\relax
449         \xP@sqrt
Rescale
450         \@tempdimb\dimexpr\@tempdimb/\count@\relax
451         \else
452         \ifdim\d@X>83042982sp\relax\@tempwattrue\fi
453         \ifdim\d@Y>83042982sp\relax\@tempwattrue\fi
454         \if@tempswa
3b) Large vector. Scale the coordinates down to avoid an overflow. 11927552sp = 182pt
455         \@tempdima\dimexpr\d@X/182*\d@X/11927552+\d@Y/182*\d@Y/11927552\relax
456         \@tempdimc\dimexpr(\d@X+\d@Y)*3/728\relax
457         \xP@sqrt
458         \multiply\@tempdimb182\relax
459         \else
3c) Medium vector. Also scale the coordinates down. 12845056sp = 196pt = 142pt
460         \@tempdima\dimexpr\d@X*\d@X/12845056+\d@Y*\d@Y/12845056\relax
461         \@tempdimc\dimexpr(\d@X+\d@Y)*3/56\relax
462         \xP@sqrt
463         \multiply\@tempdimb14\relax
464         \fi
465     \fi
466 \fi
467 \fi
468 }

```

8.2 Squiggled lines

```

\squiggledSpread@ This is the hook for squiggled straight lines.
\xP@squiggledSpread@
469 \xP@hook{pdf}{\squiggledSpread@}
470 \@ifdefinable\xP@squiggledSpread@\relax
471 \def\xP@squiggledSpread@#1\repeat@{
472     \xP@veclen

```

Neglect zero-length lines.

```

473 \ifdim\@tempdimb>\z@
474     \edef\@tempa{\xP@coord\X@p\Y@p m }%
475     \toks@\expandafter{\@tempa}%
\@tempcnta = number of squiggles
476 \@tempcnta\numexpr\@tempdimb/\xybsqll@\relax
477 \ifnum\@tempcnta<\tw@\@tempcnta\tw@\fi
478 \@tempdima\dimexpr\d@X/\@tempcnta\relax
479 \@tempdimc\dimexpr\d@Y/\@tempcnta\relax

```

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

```

480 \xP@testcont\xP@oddsquigglemacro
481 \ifxP@splinecont
482     \def\xP@squigsign{-}%
483 \else
484     \let\xP@squigsign\@empty
485 \fi

```

```

486 \count@\z@
487 \loop

```

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

```

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

```

Record the direction of the last squiggle.

```

506 \global\expandafter\let\expandafter\xP@lastpattern
507 \ifodd\numexpr\count@\if\xP@squigsign-+1\fi\relax
508 \xP@oddsquigglemacro
509 \else
510 \xP@evensquigglemacro
511 \fi
512 \xP@savec
513 \xP@stroke{\the\toks@}%
514 \fi
515 }}

```

\xP@squigsign

```

516 \newcommand*\xP@squigsign{}

```

\xP@append

```

517 \newcommand*\xP@append[2]{\{
518 \edef\@tempa{\#1\the\#2}\}%
519 \expandafter\xP@tempa
520 }

```

8.3 Circles

\circhar@@ Replacement macro for the circle chars.

```

\xP@circhar@@ 521 \xP@hook{pdf}{circhar@@}
522 \newcommand*\xP@circhar@@[1]{%
523 \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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
559      00%
560      0{173517671/654249180}%
```

```

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

```

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

```

564 \newcommand\xP@circhar@@@[8]{%
565 \xP@setsolidpat
566 \xP@stroke{\xP@coord{\R@*#1}{\R@*#2}m
567 \xP@coord{\R@*#3}{\R@*#4}\xP@coord{\R@*#5}{\R@*#6}%
568 \xP@coord{\R@*#7}{\R@*#8}c}%
569 \vrule width\z@ height\R@ depth\R@
570 \kern\dimexpr\R@*#7\relax
571 }

```

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

```

\xP@cirrestrict@@ 572 \xP@hook{pdf}{\cirrestrict@@}
573 \newcommand*\xP@cirrestrict@@{\count@ \z@ \relax}
574 \</basic>

```

8.4 Rotation and scaling

```

575 <*rotate>
576 \xycatcodes

```

`\xpdf-ro@loaded`

```

577 \expandafter\let\csname xpdf-ro@loaded\endcsname\@empty

```

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

```

\xP@xyscale@@ 578 \xP@hook{rotate}{\xyscale@@}
579 \newcommand*\xP@xyscale@@[2]{%
580 \setboxz@h{%
581 \hskip\L@p
582 \hskip-\R@p
583 \lower\U@p\hbox{\xP@cm{#1}00{#2}%
584 {\raise\U@p\hb@xt@ \z@{\hskip-\L@p\boxz@ \hss}}}%
585 }%
586 }%
587 \global\let\xP@lastpattern\@empty
588 }

```

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

```

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

```

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

```

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

```

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

```

594 \newcommand*\xP@rotate@[2]{%
595 \setboxz@h{%
596 #1{#2}%
597 \hskip\L@p
598 \hskip-\R@p
599 \lower\U@p\hbox{\xP@cm\cosDirection\sinDirection

```

```

600      {\if-\sinDirection\else-\sinDirection\fi}\cosDirection
601      {\raise\U@p\hb@xt@{z@{\hskip-\L@p\boxz@{hss}}}%
602      }%
603  }%
604  \global\let\xP@lastpattern\@empty
605 }

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

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

613   \divide\@tempdimb5898240
  It's enough to know sin between 0° and 90°. The cos and the values in the other quadrants
  can be derived from that.
614   \ifcase\@tempdimb
615     \xP@sinpoly
616     \edef\sinDirection{\xP@EARPT\@tempdimb}%
617     \@tempdima\dimexpr90pt-\@tempdima\relax
618     \xP@sinpoly
619     \edef\cosDirection{\xP@EARPT\@tempdimb}%
620   \or
621     \@tempdima\dimexpr180pt-\@tempdima\relax
622     \xP@sinpoly
623     \edef\sinDirection{\xP@EARPT\@tempdimb}%
624     \@tempdima\dimexpr90pt-\@tempdima\relax
625     \xP@sinpoly
626     \edef\cosDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
627   \or
628     \@tempdima\dimexpr\@tempdima-180pt\relax
629     \xP@sinpoly
630     \edef\sinDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
631     \@tempdima\dimexpr90pt-\@tempdima\relax
632     \xP@sinpoly
633     \edef\cosDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
634   \or
635     \@tempdima\dimexpr360pt-\@tempdima\relax
636     \xP@sinpoly
637     \edef\sinDirection{\xP@EARPT\dimexpr-\@tempdimb\relax}%
638     \@tempdima\dimexpr90pt-\@tempdima\relax
639     \xP@sinpoly
640     \edef\cosDirection{\xP@EARPT\@tempdimb}%
641   \else
642     \PackageError{xypdf}{Unexpected case in sin/cos calculation}%
643     {Feel free to contact the author of the xypdf package with a minimal %
644     example.}%
645   \fi
646 }

```

`\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)

```

647 \newcommand*\xP@sinpoly{%
648   \ifdim\@tempdima<49pt
649     \ifdim\@tempdima<27pt
650       \@tempdimb\dimexpr((\@tempdima*-529771058/16039085-1384933sp)%
651         * \@tempdima/268756075+10714164sp)*\@tempdima/613777813\relax
652     \else
653       \advance\@tempdima-27pt
654       \@tempdimb\dimexpr((\@tempdima*-743101305/20672414-238989613sp)%
655         * \@tempdima/80975565+42661556sp)*\@tempdima/622461739+2\p@)%
656       *157520747/693945047\relax
657     \fi
658   \else
659     \ifdim\@tempdima<70pt
660       \advance\@tempdima-49pt
661       \@tempdimb\dimexpr((\@tempdima*-348406699/107952940-55079229sp)%
662         * \@tempdima/866635628+408805sp)*\@tempdima/26926757+\p@)%
663       *135751711/179873976\relax
664     \else
665       \advance\@tempdima-70pt
666       \@tempdimb\dimexpr((\@tempdima*-1015850353/137849442-460519207sp)%
667         * \@tempdima/8742349+142263941sp)*\@tempdima/972432199+23\p@)%
668       *31253604/764969669\relax
669     \fi
670   \fi
671   \global\dimen@i\@tempdimb
672 } \@tempdimb\dimen@i
673 }

```

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

```

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

```

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

```

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

```



```

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

```

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

```

699 \xP@vecclen
700 \xdef\@gtempa{%
701 \def\noexpand\cosDirection{\xP@EARPT\dimexpr\d@X*\p@/\@tempdimb\relax}%
702 \def\noexpand\sinDirection{\xP@EARPT\dimexpr\d@Y*\p@/\@tempdimb\relax}%
703 }%
704 }\@gtempa
705 }

```

8.5 Temporary registers

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

```

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

```

The next section is for the “curve” extension!

```

718 </basic>
719 <*curve>
720 \xycatcodes

```

$\xypdf-cu@loaded$

```

721 \expandafter\let\csname xypdf-cu@loaded\endcsname\@empty

```

$\xP@tempvar$ In order to save registers, $xypdf$ shares $\Xy-pic$ ’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.

```

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

```

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.

```

724 \let#1#2%
725 }

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

\XP@parA ●2 Second set of temporary variables: for the arc length algorithm.
\XP@velA 727 \XP@tempvar\XP@parA\L@p
\XP@parB 728 \XP@tempvar\XP@velA\U@p
\XP@velB 729 \XP@tempvar\XP@parB\R@p
\XP@parC 730 \XP@tempvar\XP@velB\D@p
\XP@velC 731 \XP@tempvar\XP@parC\X@origin
\XP@parD 732 \XP@tempvar\XP@velC\Y@origin
\XP@velD 733 \XP@tempvar\XP@parD\X@xbase
\XP@parE 734 \XP@tempvar\XP@velD\Y@xbase
\XP@velE 735 \XP@tempvar\XP@parE\X@ybase
\XP@lenA 736 \XP@tempvar\XP@velE\Y@ybase
\XP@lenB 737 \XP@tempvar\XP@lenA\X@min
\XP@lenB 738 \XP@tempvar\XP@lenB\Y@min
\XP@partlen 739 \XP@tempvar\XP@partlen\X@max
\XP@oldpartlen 740 \XP@tempvar\XP@oldpartlen\Y@max
\XP@tolerance 741 \XP@tempvar\XP@tolerance\almostz@

\XP@A ●3 Third set of temporary registers: Bézier offset algorithm and solving linear equations.
\XP@B 742 \XP@tempvar\XP@A\L@p
\XP@C 743 \XP@tempvar\XP@B\U@p
\XP@D 744 \XP@tempvar\XP@C\R@p
\XP@E 745 \XP@tempvar\XP@D\D@p
\XP@F 746 \XP@tempvar\XP@E\X@origin
\XP@G 747 \XP@tempvar\XP@F\Y@origin
\XP@H 748 \XP@tempvar\XP@G\X@xbase
\XP@I 749 \XP@tempvar\XP@H\Y@xbase
\XP@J 750 \XP@tempvar\XP@I\X@ybase
\XP@K 751 \XP@tempvar\XP@J\Y@ybase
\XP@L 752 \XP@tempvar\XP@K\X@min
\XP@L 753 \XP@tempvar\XP@L\Y@min
\XP@fa 754 \XP@tempvar\XP@fa\X@max
\XP@fd 755 \XP@tempvar\XP@fd\Y@max
\XP@tm 756 \XP@tempvar\XP@tm\almostz@
\XP@xm 757 \XP@tempvar\XP@xm\K@dXdY
\XP@ym 758 \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 759 \@tempcnta\count11\relax
\XP@M 760 \XP@newdimen\XP@off
\XP@oldobj 761 \XP@newdimen\XP@ta
\XP@Tax 762 \XP@newdimen\XP@tb
\XP@Tay 763 \XP@newdimen\XP@tc
\XP@Tdx 764 \XP@newdimen\XP@M
\XP@Tdy 765 \XP@newdimen\XP@oldobj
\XP@Tmx 766 \XP@newdimen\XP@Tax
\XP@Tmy ●4
\XP@xa 767 \XP@newdimen\XP@Tay
\XP@ya
\XP@xb
\XP@yb
\XP@xc
\XP@yc
\XP@xd
\XP@yd

```

```

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

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

\xP@sa •6 Sixth set of temporary variables: Solving a linear system approximately.
\xP@sb 790 \xP@tempvar\xP@sa\xP@Tax
\xP@sc 791 \xP@tempvar\xP@sb\xP@Tay
\xP@Ab 792 \xP@tempvar\xP@sc\xP@Tdx
\xP@AAb 793 \xP@tempvar\xP@Ab\xP@Tdy
\xP@Aba 794 \xP@tempvar\xP@AAb\xP@Tmx
\xP@Abb 795 \xP@tempvar\xP@Aba\xP@Tmy
\xP@Abc 796 \xP@tempvar\xP@Abb\xP@xa
\xP@AAba 797 \xP@tempvar\xP@Abc\xP@ya
\xP@AAbb 798 \xP@tempvar\xP@AAba\xP@xb
\xP@AAbc 799 \xP@tempvar\xP@AAbb\xP@yb
800 \xP@tempvar\xP@AAbc\xP@xc
\xP@dta 801 \xP@tempvar\xP@dta\xP@yc
\xP@dtb 802 \xP@tempvar\xP@dtb\xP@xd
\xP@dtc 803 \xP@tempvar\xP@dtc\xP@yd

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

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

```

8.6 Bézier curves

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

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

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

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

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

Neglect splines of length zero.

826 `\ifdim\@tempdimb>\z@`

Set the dash pattern.

827 `#1%`

Draw the spline.

828 `\xP@stroke{\xP@coor\X@p\Y@p m %`
829 `\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.

830 `\xP@savec`
831 `\fi`
832 `\fi`
833 `}`

`\xP@preparespline`

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

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

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

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

841 `\xP@shavespline`

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

842 `\xP@bezierlength`
843 `}`

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

844 `\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.)

```
845 \newcommand*\xP@shavespline{%
846   \xP@shaveprec{\dimen5*\xP@bigdim/\p@}{\dimen7*\xP@bigdim/\p@}%
847 }
```

`\xP@shaveprec` **•1** Shave a cubic spline at both ends at the parameter values in #1 and #2. For normal use, the parameters fulfill $0pt \leq \dimen1 < \dimen2 \leq \xP@bigdim$. 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.

```
848 \newcommand*\xP@shaveprec[2]{%
849   \xP@inibigdim
850   \A@\dimexpr#1\relax
851   \B@\dimexpr#2\relax
852   \@tempswatrue
853   \ifdim\A@=\z@\ifdim\B@=\xP@bigdim\@tempswafalse\fi\fi
854   \if@tempswa
855     \L@\dimexpr\L@c-\X@p\relax
856     \U@\dimexpr\R@c-\L@p-\L@c\relax
857     \R@\dimexpr\X@c-3\R@c+3\L@c-\X@p\relax
858     \D@\dimexpr\U@c-\Y@p\relax
859     \X@min\dimexpr\D@c-\D@p-\U@c\relax
860     \Y@min\dimexpr\Y@c-3\D@c+3\U@c-\Y@p\relax
861     \xdef\@gtempa{%
862       \X@p\the\dimexpr\X@p+(3\L@p+(3\U@p+\R@p*\A@/\xP@bigdim)%
863         *\A@/\xP@bigdim)*\A@/\xP@bigdim\relax
864       \Y@p\the\dimexpr\Y@p+(3\D@p+(3\X@min+\Y@min*\A@/\xP@bigdim)%
865         *\A@/\xP@bigdim)*\A@/\xP@bigdim\relax
866       \L@c\the\dimexpr\X@p+(2\A@+\B@)*\L@p/\xP@bigdim+((\A@+2\B@)%
867         *\U@p/\xP@bigdim+\R@p*\A@/\xP@bigdim*\B@/\xP@bigdim)%
868         *\A@/\xP@bigdim\relax
869       \U@c\the\dimexpr\Y@p+(2\A@+\B@)*\D@p/\xP@bigdim+((\A@+2\B@)%
870         *\X@min/\xP@bigdim+\Y@min*\A@/\xP@bigdim*\B@/\xP@bigdim)%
871         *\A@/\xP@bigdim\relax
872       \R@c\the\dimexpr\X@p+(2\B@+\A@)*\L@p/\xP@bigdim+((\B@+2\A@)%
873         *\U@p/\xP@bigdim+\R@p*\B@/\xP@bigdim*\A@/\xP@bigdim)%
874         *\B@/\xP@bigdim\relax
875       \D@c\the\dimexpr\Y@p+(2\B@+\A@)*\D@p/\xP@bigdim+((\B@+2\A@)%
876         *\X@min/\xP@bigdim+\Y@min*\B@/\xP@bigdim*\A@/\xP@bigdim)%
877         *\B@/\xP@bigdim\relax
878       \X@c\the\dimexpr\X@p+(3\L@p+(3\U@p+\R@p*\B@/\xP@bigdim)%
879         *\B@/\xP@bigdim)*\B@/\xP@bigdim\relax
880       \Y@c\the\dimexpr\Y@p+(3\D@p+(3\X@min+\Y@min*\B@/\xP@bigdim)%
881         *\B@/\xP@bigdim)*\B@/\xP@bigdim\relax}%
882   \else
883     \global\let\@gtempa\relax
884   \fi
885 } \@gtempa
886 }
```

`\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`.

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

Arc length (integral over the quadratic approximation)

```
895   \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.

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

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

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

Pass the result to outside the group.

```
898   \global\dimen@i\@tempdimb
899 } \@tempdimb\dimen@i
900 }
```

`\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`.

```
901 \newcommand*\xP@velocity[2]{%
902   \@tempdima\dimexpr#1\relax
903   \xP@tangent
904   \global\dimen@i\@tempdimb
905 } #2\dimen@i
906 }
```

`\xP@tangent` **•1**

```
907 \newcommand*\xP@tangent{%
908   \d@X3\xP@precbeziertan\X@p\L@c\R@c\X@c\@tempdima
909   \d@Y3\xP@precbeziertan\Y@p\U@c\D@c\Y@c\@tempdima
910   \xP@vecclen
911 }
```

`\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`.

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

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

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

```

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

```

Or even the third derivative.

```

921      \ifdim\@tempdimb=\z@
922      \d@X\L@p
923      \d@Y\U@p
924      \xP@vecLen
925      \ifdim\@tempdimb=\z@
926      \xP@warning{xyPDF}{Cannot determine a tangent vector to a curve}%
927      \@tempdimb\p@
928      \fi
929      \fi
930      \fi
931      \global\dimen@i\d@X
932      \global\dimen3\d@Y
933      \global\dimen5\@tempdimb
934      }%
935      \d@X\dimen@i
936      \d@Y\dimen3\relax
937      \@tempdimb\dimen5\relax
938      }

```

`\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.

```

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

```

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

```

945   \xP@parB\dimexpr(\xP@parC+\xP@parA)/2\relax
946   \xP@velocity\xP@parB\xP@velB
947   \xP@parD\dimexpr(\xP@parE+\xP@parC)/2\relax
948   \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).

```

949   \xP@lenA
950   \dimexpr(\xP@velA+4\xP@velB+\xP@velC)/6*(\xP@parC-\xP@parA)/\xP@bigdim\relax
951   \xP@lenB
952   \dimexpr(\xP@velC+4\xP@velD+\xP@velE)/6*(\xP@parE-\xP@parC)/\xP@bigdim\relax
953   \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.

```

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

```

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

```

956   \edef\next@{%

```



```

957     \noexpand\xP@arclength\xP@parB\xP@velB\xP@parC\xP@velC\xP@lenA
958     \noexpand\xP@arclength{\the\xP@parD}{\the\xP@velD}{\the\xP@parE}%
959     {\the\xP@velE}{\the\xP@lenB}%
960   }%
961   \else
No? Proceed to the next parameter interval.
962     \xP@parA\xP@parE
963     \xP@velA\xP@velE
964     \advance\@tempdimb\xP@partlen
965     \DN@{}%
966   \fi
967   \next@
968 }

```

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{~~}

```

969 \xP@hook{curve}{@crv@}
970 \newcommand*\xP@@crv@[2]{\DN@{#1#2}%
971   \ifx\next@ \@empty \edef\next@{\crv@defaultshape}%
972   \ifx\bstartPLACE@ \@empty \xdef\crvSTYLE@{\crv@defaultshape}}\fi
973   \else
974   \ifx\bstartPLACE@ \@empty \gdef\crvSTYLE@{#1#2}}\fi
975   \fi
976   \ifx\next@ \@empty \crv@noobject \DN@{\crv@}{\xy@@crvaddstack@}%
977   \else \def\tmp@{-}\ifx\next@\tmp@ \DN@{\crv@}{\xy@@crvaddstack@}%
978   \else \def\tmp@{=}\ifx\next@\tmp@
979   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{=}}}%
980   \else \def\tmp@{2-}\ifx\next@\tmp@
981   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{2.}}}%
982   \else \def\tmp@{3-}\ifx\next@\tmp@
983   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{3.}}}%
984   \else \def\tmp@{--}\ifx\next@\tmp@
985   \DN@{\expandafter\crv@\crv@specialtemplate@{--}}%
986   \else \def\tmp@{==}\ifx\next@\tmp@
987   \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{--}}}%
988   \else \def\tmp@{2--}\ifx\next@\tmp@
989   \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{--}}}%
990   \else \def\tmp@{3--}\ifx\next@\tmp@
991   \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{--}}}%
992   \else \def\tmp@{.}\ifx\next@\tmp@
993   \DN@{\expandafter\crv@\crv@specialtemplate@{.}}%
994   \else \def\tmp@{:}\ifx\next@\tmp@
995   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{:}}}%
996   \else \def\tmp@{2.}\ifx\next@\tmp@
997   \DN@{\expandafter\crv@\crv@normaltemplate{\dir{:}}}%
998   \else \def\tmp@{3.}\ifx\next@\tmp@
999   \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{.}}}%
1000  \else \def\tmp@{~}\ifx\next@\tmp@
1001  \DN@{\expandafter\crv@\crv@normaltemplate{\dir{~}}}%
1002  \else \def\tmp@{2~}\ifx\next@\tmp@
1003  \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{~}}}%
1004  \else \def\tmp@{3~}\ifx\next@\tmp@
1005  \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{~}}}%

```

```

1006 \else\def\tmp@{~}\ifx\next@\tmp@
1007 \DN@{\expandafter\crv@\crv@normaltemplate{\dir{~}}}%
1008 \else\def\tmp@{2~}\ifx\next@\tmp@
1009 \DN@{\expandafter\crv@\crv@normaltemplate{\dir2{~}}}%
1010 \else\def\tmp@{3~}\ifx\next@\tmp@
1011 \DN@{\expandafter\crv@\crv@normaltemplate{\dir3{~}}}%
1012 \else\def\tmp@{.}\ifx\next@\tmp@
1013 \DN@{\expandafter\crv@\crv@specialtemplate@{.}}%
1014 \else
1015 \DN@{\expandafter\crv@\crv@othertemplate{\dir#1{#2}}}%
1016 \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{~}

```

```

1017 \XP@hook{curve}{\sysplinespecialcases@}
1018 \newcommand*\XP@sysplinespecialcases@{%
1019 \ifx\@empty\ycrvdrop@
1020 \ifx\@empty\ycrvconn@\DN@{\splinesolid@}%
1021 \else\DN@{ \dir{-}}\ifx\next@\ycrvconn@\DN@{\splinesolid@}%
1022 \else\DN@{ \dir2{-}}\ifx\next@\ycrvconn@\DN@{\splinedoubled@}%
1023 \else\DN@{ \dir{=}}\ifx\next@\ycrvconn@\DN@{\splineribboned@}%
1024 \else\DN@{ \dir{2.}}\ifx\next@\ycrvconn@\DN@{\splinedoubled@}%
1025 \else\DN@{ \dir3{-}}\ifx\next@\ycrvconn@\DN@{\splinetrebled@}%
1026 \else\DN@{ \dir{3.}}\ifx\next@\ycrvconn@\DN@{\splinetrebled@}%
1027 \else\DN@{ \dir{--}}\ifx\next@\ycrvconn@\DN@{\splinedashed@}%
1028 \else\DN@{ \dir{.}}\ifx\next@\ycrvconn@\DN@{\splinedotted@}%
1029 \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.

```

1030 \else\DN@{ \dir2{.}}\ifx\next@\ycrvconn@\DN@{\splinedbldotted@}%
1031 \else\DN@{ \dir3{.}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrbldotted@}%
1032 \else\DN@{ \dir2{--}}\ifx\next@\ycrvconn@\DN@{\XP@splinedbldashed@}%
1033 \else\DN@{ \dir3{--}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrbldashed@}%
1034 \else\DN@{ \dir{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinesquiggled@}%
1035 \else\DN@{ \dir2{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinedblsquiggled@}%
1036 \else\DN@{ \dir3{~}}\ifx\next@\ycrvconn@\DN@{\XP@splinetrblsquiggled@}%
1037 \else\DN@{ \dir{~~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokensquiggled@}%
1038 \else\DN@{ \dir2{~~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokendblsquiggled@}%
1039 \else\DN@{ \dir3{~~}}\ifx\next@\ycrvconn@\DN@{\XP@splinebrokentrblsquiggled@}%
1040 \else\ifdim\splinetol@>z@\else\splinedefaulttol@fi
1041 \DN@{\splineset@}\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
1042 \else
1043 \DN@{\splineset@}%
1044 \fi
1045 \ifInvisible@\DN@{}\fi
1046 \next@
1047 }

```

8.8 Multiple solid curves

```
\XP@splinedoubled@
```

```

1048 \XP@hook{curve}{\splinedoubled@}
1049 \newcommand*\XP@splinedoubled@{%
1050 \XP@checkspline\XP@splinemultsolid\XP@doublestroke}

```

```

\XP@splineribboned@
1051 \XP@hook{curve}{splineribboned@}
1052 \@ifdefinable\XP@splineribboned@\relax
1053 \let\XP@splineribboned@\XP@splinedoubled@

\XP@splinetrebled@
1054 \XP@hook{curve}{splinetrebled@}
1055 \newcommand*\XP@splinetrebled@{%
1056   \XP@checkspline\XP@splinemultsolid\XP@trblstroke}

\XP@doublestroke   Offset parameters for double lines and curves
1057 \newcommand*\XP@doublestroke{\xydashh@/2,-\xydashh@/2}

\XP@trblstroke     Offset parameters for treble lines and curves
1058 \newcommand*\XP@trblstroke{\xydashh@,\z@,-\xydashh@}

\XP@checkspline    Get and check spline parameters before the macro in #1 is executed.
1059 \newcommand*\XP@checkspline[1]{%
1060   \readsplineparams@
   Neglect splines which are drawn “backwards”. Somehow Xy-pic draws curves forward and
   backward, but we need it to be drawn only once.
1061   \let\next@\@gobble
1062   \ifdim\dimen5<\dimen7
1063     \XP@preparespline
   Neglect splines of zero length.
1064     \ifdim\@tempdimb>\z@
   If the path length is less than twice the line width, just draw a solid path.
1065       \ifdim\@tempdimb<2\dimexpr\XP@preclw\relax
1066         \let\next@\XP@splinemultsolid
1067       \else
1068         \let\next@#1%
1069       \fi
1070     \fi
1071   \fi
1072   \next@
1073 }

\XP@splinemultsolid •1
1074 \newcommand*\XP@splinemultsolid[1]{%
1075   \XP@inibigdim
1076   \@temptokena{}%
1077   \XP@setsolidpat
   The \@for loop does the multiple strokes. \@tempa records the respective offset distance.
1078   \@for\@tempa:={#1}\do{\XP@paintsolid\z@\XP@bigdim}%
1079   \XP@stroke{\the\@temptokena}%
1080 }}

\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.
1081 \newcommand*\XP@paintsolid[2]{%

```

Record the original anchor points.

```

1082 \xP@savepts
1083 \xP@a#1\relax
1084 \xP@c#2\relax
1085 \xP@movetotrue
1086 \xP@paintsolid@
1087 \xdef\@gtempa{\the\@temptokena}%
1088 }%
1089 \@temptokena\expandafter{\@gtempa}%
1090 }

```

\xP@paintsolid@ ●1 ●5

```

1091 \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.

```

1092 \xP@b\xP@c
Offset distance
1093 \xP@off\dimexpr\@tempa\relax
1094 \ifdim\xP@off=\z@
1095 \xP@shaveprec\xP@a\xP@c
1096 \else
1097 \loop

```

Restore the original anchor points.

```

1098 \xP@restorepts

```

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

```

1099 \xP@offsetsegment
Test if the offset curve is good enough.
1100 \xP@testoffset
If not, shorten the parameter interval by 30%.
1101 \unless\ifxP@offsetok
1102 \xP@b\dimexpr\xP@a+(\xP@b-\xP@a)*7/10\relax
1103 \repeat
1104 \fi

```

Append the new segment to the path.

```

1105 \xP@append\@temptokena{\ifxP@moveto\xP@coor\X@p\Y@p m \fi
1106 \xP@coor\L@c\U@c\xP@coor\R@c\D@c\xP@coor\X@c\Y@c c }%
1107 \xP@movetofalse

```

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

```

1108 \ifdim\xP@b<\xP@c\relax
1109 \xP@a\xP@b
1110 \expandafter\xP@paintsolid@
1111 \fi
1112 }

```

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

```

1113 \@ifdefinable\ifxP@moveto\relax
1114 \@ifdefinable\xP@movetotrue\relax
1115 \@ifdefinable\xP@movetofalse\relax
1116 \newif\ifxP@moveto

```

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

```

1117 \newcommand*\xP@savepts{%
1118   \xP@xa\xP@p
1119   \xP@ya\Y@p
1120   \xP@xb\L@c
1121   \xP@yb\U@c
1122   \xP@xc\R@c
1123   \xP@yc\D@c
1124   \xP@xd\X@c
1125   \xP@yd\Y@c
1126 }

```

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

```

1127 \newcommand*\xP@restorepts{%
1128   \X@p\xP@xa
1129   \Y@p\xP@ya
1130   \L@c\xP@xb
1131   \U@c\xP@yb
1132   \R@c\xP@xc
1133   \D@c\xP@yc
1134   \X@c\xP@xd
1135   \Y@c\xP@yd
1136 }

```

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:

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

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

Temporary:

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

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 1137 \ifdefinable\xP@tmx\relax
1138 \ifdefinable\xP@tmy\relax
```

```
\xP@Tmxy ●4
\xP@Tmyx 1139 \newcommand*\xP@Tmxy{* \xP@Tmx / \xP@Tmy}
1140 \newcommand*\xP@Tmyx{* \xP@Tmy / \xP@Tmx}
```

```
\xP@Tmzero
1141 \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, \dots, \Y@c`. The partial spline in the parameter interval $[\backslash\mathrm{xP@a}, \backslash\mathrm{xP@b}] \subseteq [0pt, \backslash\mathrm{xP@bigdim}]$ is offset. The new Bézier curve is returned in `\xP@xa, \dots, \xP@yd`.

```
1142 \newcommand*\xP@offsetsegment{%
New first anchor point and tangent vector at 0
1143 \xP@tangentvec\xP@a
1144 \xP@xa\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@a/8%
1145 +\d@Y*\xP@off/\@tempdimb\relax
1146 \xP@ya\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@a/8%
1147 -\d@X*\xP@off/\@tempdimb\relax
1148 \xP@scaleT
1149 \xP@Tax\d@X
1150 \xP@Tay\d@Y
1151 \xP@E\@tempdimb
New last anchor point and tangent vector at 1
1152 \xP@tangentvec\xP@b
1153 \xP@xd\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@b/8%
1154 +\d@Y*\xP@off/\@tempdimb\relax
1155 \xP@yd\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@b/8%
1156 -\d@X*\xP@off/\@tempdimb\relax
1157 \xP@scaleT
```

```

1158 \xP@Tdx-\d@X
1159 \xP@Tdy-\d@Y
1160 \xP@F\@tempdimb

```

Scalar product of the tangent vectors

```

1161 \xP@M\z@
1162 \xP@Max\xP@M\xP@Tdx
1163 \xP@Max\xP@M\xP@Tdy
1164 \xP@L\dimexpr\xP@Tax*\xP@Tdx/\xP@M+\xP@Tay*\xP@Tdy/\xP@M\relax
1165 \xP@tm\dimexpr(\xP@a+\xP@b)/2\relax
1166 \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.

```

1167 \d@X-\dimexpr\xP@Tay+\xP@Tdy\relax
1168 \d@Y\dimexpr\xP@Tax+\xP@Tdx\relax
1169 \xP@vecLen
1170 \xP@A\dimexpr\X@c-\X@p\relax
1171 \xP@B\dimexpr\Y@c-\Y@p\relax
1172 \xP@M\z@
1173 \xP@Max\xP@M\xP@A
1174 \xP@Max\xP@M\xP@B
1175 \ifdim\xP@M=\z@
1176 \xP@A\dimexpr\R@c-\L@c\relax
1177 \xP@B\dimexpr\D@c-\U@c\relax
1178 \xP@Max\xP@M\xP@A
1179 \xP@Max\xP@M\xP@B
1180 \fi
1181 \xP@M\dimexpr\d@X*\xP@A/\xP@M+\d@Y*\xP@B/\xP@M\relax
1182 \ifdim\xP@M<\z@
1183 \multiply\d@X\m@ne
1184 \multiply\d@Y\m@ne
1185 \fi
1186 \else

```

Normal case: tangent vector at the middle point.

```

1187 \xP@tangentvec\xP@tm
1188 \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)

```

1189 \xP@xm\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\xP@tm%
1190 +8\d@Y*\xP@off/\@tempdimb\relax
1191 \xP@ym\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\xP@tm%
1192 -8\d@X*\xP@off/\@tempdimb\relax

```

Tangent at middle point

```

1193 \xP@Tmx\d@X

```



```

1194 \xP@Tmy\d@Y
1195 \xP@ifabsless\xP@Tmy\xP@Tmx
1196 \let\xP@tmy\xP@Tmyx
1197 \let\xP@tmx\@empty
1198 \else
1199 \ifdim\xP@Tmy=\z@
1200 \let\xP@tmx\xP@Tmzero
1201 \let\xP@tmy\xP@Tmzero
1202 \else
1203 \let\xP@tmy\@empty
1204 \let\xP@tmx\xP@Tmxy
1205 \fi
1206 \fi

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

1207 \xP@fa\p@
1208 \xP@fd\p@
1209 \xP@tm\p@

The main loop for finding the offset curve

1210 \count@\z@
1211 \loop

Set the new control points up.

1212 \xP@offsetpoints
1213 \@tempswafalse

At most 10 iterations

1214 \ifnum10>\count@

Determine the quality of the approximation by an objective function.

1215 \xP@objfun\xP@oldobj
1216 \ifdim\xP@oldobj>\xP@maxobjfun\relax\@tempswatrue\fi
1217 \fi
1218 \if@tempswa
1219 \xP@offsetloop
1220 \repeat

Return the new anchor points.

1221 \xdef\@gtempa{\X@p\the\xP@xa\Y@p\the\xP@ya
1222 \L@c\the\xP@xb\U@c\the\xP@yb\R@c\the\xP@xc\D@c\the\xP@yc
1223 \X@c\the\xP@xd\Y@c\the\xP@yd\relax}%
1224 }%
1225 \@gtempa
1226 }

```

\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 $v_1 + 2\pi \cdot |\text{\texttt{\textbackslash xP@off}}|$.

```

1227 \newcommand*\xP@scaleT{%
1228 \xP@B6.28\xP@off
1229 \xP@abs\xP@B

```

```

1230 \xP@C\dimexpr\d@X*\xP@B/\@tempdimb\relax
1231 \xP@D\dimexpr\d@Y*\xP@B/\@tempdimb\relax
1232 \xP@A\dimexpr\xP@b-\xP@a\relax
1233 \d@X\dimexpr\xP@C+\d@X*\xP@A/\xP@bigdim\relax
1234 \d@Y\dimexpr\xP@D+\d@Y*\xP@A/\xP@bigdim\relax

```

Also record the change to the norm of the vector.

```

1235 \@tempdimb\dimexpr\xP@B+\@tempdimb*\xP@A/\xP@bigdim\relax
1236 }

```

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

```

1237 \newcommand*\xP@offsetloop{%
1238 \xP@C\dimexpr\xP@C/2\relax
1239 \xP@G\dimexpr\xP@G/2\relax

1st linear equation
1240 \xP@ta\dimexpr2\p@-\xP@tm\relax
1241 \xP@tb\dimexpr\xP@tm*\xP@ta/\p@\relax
1242 \xP@A\dimexpr\xP@Tax*\xP@tb/\p@*\xP@ta/\p@\relax
1243 \xP@B\dimexpr\xP@Tdx*\xP@tb/\p@*\xP@tm/\p@\relax

```

2nd linear equation

```

1244 \xP@E\dimexpr\xP@Tay*\xP@tb/\p@*\xP@ta/\p@\relax
1245 \xP@F\dimexpr\xP@Tdy*\xP@tb/\p@*\xP@tm/\p@\relax

```

3rd linear equation

```

1246 \xP@tb\dimexpr2\p@-3\xP@tm\relax
1247 \xP@tc\dimexpr\xP@tb+2\p@\relax
1248 \xP@I\dimexpr(2\xP@Tay\xP@tmx-2\xP@Tax\xP@tmy)*\xP@tb/\p@*\xP@ta/\p@\relax
1249 \xP@J\dimexpr(2\xP@Tdy\xP@tmx-2\xP@Tdx\xP@tmy)*\xP@tc/\p@*\xP@tm/\p@\relax
1250 \xP@K\dimexpr((\xP@yd-\xP@ya+(\xP@yb-\xP@yc)*3)
1251 * \xP@tm/\p@+(\xP@yc-2\xP@yb+\xP@ya)*2)*12\xP@tmx
1252 -((\xP@xd-\xP@xa+(\xP@xb-\xP@xc)*3)
1253 * \xP@tm/\p@+(\xP@xc-2\xP@xb+\xP@xa)*2)*12\xP@tmy\relax

```

Solve the system.

```

1254 \xP@solvelinearsystem
1255 \ifxP@validsol

```

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

```

1256 \xP@correctsol
1257 \ifdim\xP@ta=z@
1258 \ifdim\xP@tb=z@
1259 \ifdim\xP@tc=z@
1260 \xP@validsolfalse
1261 \fi\fi\fi
1262 \fi

```

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

```

1263 \ifxP@validsol
1264 \else
1265 \xP@applinsys

```

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

```

1266 \advance\xP@fa-\xP@ta
1267 \advance\xP@fd-\xP@tb
1268 \advance\xP@tm-\xP@tc

```

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

```

1269   \ifdim\xP@tm<\z@\xP@tm\z@\fi
1270   \ifdim\xP@tm>2\p@\xP@tm2\p@\fi
1271   \fi
1272   \advance\count@\@ne
1273 }

```

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

```

1274 \newcommand*\xP@maxsol{3pt}

```

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

```

1275 \newcommand*\xP@correctsol{%
  If the solution is too big, scale all variables uniformly.

1276   \xP@M\z@
1277   \xP@Max\xP@M\xP@ta
1278   \xP@Max\xP@M\xP@tb
1279   \xP@Max\xP@M\xP@tc
1280   \ifdim\xP@M>\xP@maxsol
1281     \xP@ta\dimexpr\xP@maxsol*\xP@ta/\xP@M\relax
1282     \xP@tb\dimexpr\xP@maxsol*\xP@tb/\xP@M\relax
1283     \xP@tc\dimexpr\xP@maxsol*\xP@tc/\xP@M\relax
1284   \fi

```

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

```

1285   \advance\xP@fa-\xP@ta
1286   \advance\xP@fd-\xP@tb
1287   \xP@M\xP@tm
1288   \advance\xP@tm-\xP@tc

```

The near-middle parameter must lie on the segment.

```

1289   \ifdim\xP@tm<\z@\xP@tm\z@\fi
1290   \ifdim\xP@tm>2\p@\xP@tm2\p@\fi

```

Check whether the solution actually improves the objective function.

```

1291   {\xP@offsetpoints
1292     \xP@objfun\xP@M
1293   \expandafter}%

```

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

```

1294   \ifdim\xP@M>\xP@oldobj
1295     \advance\xP@fa\xP@ta
1296     \advance\xP@fd\xP@tb
1297     \xP@tm\xP@M
1298     \xP@validsolfalse
1299   \fi
1300 }

```

`\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.

```

1301 \newcommand*\xP@objfun[1]{%
1302   \xP@D\dimexpr\xP@bezierpoly\xP@xa\xP@xb\xP@xc\xP@xd\xP@tm-\xP@xm\relax
1303   \xP@H\dimexpr\xP@bezierpoly\xP@ya\xP@yb\xP@yc\xP@yd\xP@tm-\xP@ym\relax
1304   \xP@C\xP@beziertan\xP@xa\xP@xb\xP@xc\xP@xd\xP@tm
1305   \xP@G\xP@beziertan\xP@ya\xP@yb\xP@yc\xP@yd\xP@tm
1306   \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.

```

1307 #1\z@
1308 \xP@Max#1\xP@D
1309 \xP@Max#1\xP@H
1310 \xP@Max#1\xP@L
1311 #1\ifdim#1>4843165sp
1312 \maxdimen
1313 \else
1314 \dimexpr\xP@D*\xP@D/\p@+\xP@H*\xP@H/\p@+\xP@L*\xP@L/\p@\relax
1315 \fi
1316 }

```

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

```

1317 \newcommand*\xP@offsetpoints{%
1318 \xP@xb\dimexpr\xP@xa+\xP@Tax*\xP@fa/196608\relax
1319 \xP@yb\dimexpr\xP@ya+\xP@Tay*\xP@fa/196608\relax
1320 \xP@xc\dimexpr\xP@xd+\xP@Tdx*\xP@fd/196608\relax
1321 \xP@yc\dimexpr\xP@yd+\xP@Tdy*\xP@fd/196608\relax
1322 }

```

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

```

1323 \newcommand*\xP@bezierpoly[5]{%
1324 \dimexpr(((\#4-\#1+(\#2-\#3)*3)*\#5/\p@+(\#1-2\#2+\#3)*6)*\#5/\p@+(\#2-\#1)*12)*\#5/\p@
1325 +\#1*8\relax
1326 }

```

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

```

1327 \newcommand*\xP@precbezierpoly[5]{%
1328 \dimexpr(((\#4-\#1+(\#2-\#3)*3)*2*\#5/\xP@bigdim+(\#1-2\#2+\#3)*6)*2*\#5/\xP@bigdim
1329 +(\#2-\#1)*12)*2*\#5/\xP@bigdim+\#1*8\relax
1330 }

```

`\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.

```

1331 \newcommand*\xP@beziertan[5]{%
1332 \dimexpr((\#4-\#1+(\#2-\#3)*3)*3*\#5/32768+(\#1-2\#2+\#3)*24)*\#5/\p@+(\#2-\#1)*24\relax
1333 }

```

`\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.

```

1334 \newcommand*\xP@precbeziertan[5]{%
1335 \dimexpr((\#4-\#1+(\#2-\#3)*3)*\#5/\xP@bigdim+(\#1-2\#2+\#3)*2)*\#5/\xP@bigdim
1336 +\#2-\#1\relax
1337 }

```

`\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 1338 \@ifdefinable\xP@varone\relax
              1339 \@ifdefinable\xP@vartwo\relax
              1340 \@ifdefinable\xP@varthree\relax

\ifxP@validsol  Records if a valid solution to the linear system is returned.
              1341 \@ifdefinable\ifxP@validsol\relax
              1342 \@ifdefinable\xP@validsoltrue\relax
              1343 \@ifdefinable\xP@validsolfalse\relax
              1344 \newif\ifxP@validsol

              1345 \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.
              1346 \xP@scalerow\xP@A\xP@B\xP@C\xP@D
              1347 \xP@scalerow\xP@E\xP@F\xP@G\xP@H
              1348 \xP@scalerow\xP@I\xP@J\xP@K\xP@L
              1349 \xP@scalectol\xP@A\xP@E\xP@I\xP@scalectol
              1350 \xP@scalectol\xP@B\xP@F\xP@J\xP@scalectol
              1351 \xP@scalectol\xP@C\xP@G\xP@K\xP@scalectol

Record the initial variable-to-column assignment.
              1352 \let\xP@varone\xP@ta
              1353 \let\xP@vartwo\xP@tb
              1354 \let\xP@varthree\xP@tc

Find the pivot position. \xP@M is used temporarily.
              1355 \count@m@ne
              1356 \@tempcnta@m@ne
              1357 \xP@ifabsless\xP@A\xP@B\@tempcnta\z@\xP@M\xP@B
              1358 \else\xP@M\xP@A\fi
              1359 \xP@ifabsless\xP@M\xP@C\@tempcnta\@ne\xP@M\xP@C\fi
              1360 \xP@ifabsless\xP@M\xP@E\@tempcnta@m@ne\count@\z@\xP@M\xP@E\fi
              1361 \xP@ifabsless\xP@M\xP@F\@tempcnta\z@\count@\z@\xP@M\xP@F\fi
              1362 \xP@ifabsless\xP@M\xP@G\@tempcnta\@ne\count@\z@\xP@M\xP@G\fi
              1363 \xP@ifabsless\xP@M\xP@I\@tempcnta@m@ne\count@\@ne\xP@M\xP@I\fi
              1364 \xP@ifabsless\xP@M\xP@J\@tempcnta\z@\count@\@ne\xP@M\xP@J\fi
              1365 \xP@ifabsless\xP@M\xP@K\@tempcnta\@ne\count@\@ne\fi

Swap rows
              1366 \ifcase\count@
              1367 \xP@swapdim\xP@A\xP@E
              1368 \xP@swapdim\xP@B\xP@F
              1369 \xP@swapdim\xP@C\xP@G
              1370 \xP@swapdim\xP@D\xP@H
              1371 \or
              1372 \xP@swapdim\xP@A\xP@I

```

```

1373     \xP@swapdim\xP@B\xP@J
1374     \xP@swapdim\xP@C\xP@K
1375     \xP@swapdim\xP@D\xP@L
1376 \fi

Swap columns
1377 \ifcase\@tempcnta
1378     \xP@swapdim\xP@A\xP@B
1379     \xP@swapdim\xP@E\xP@F
1380     \xP@swapdim\xP@I\xP@J
1381     \let\xP@varone\xP@tb
1382     \let\xP@vartwo\xP@ta
1383     \xP@swapnum\xP@scaleone\xP@scaletwo
1384 \or
1385     \xP@swapdim\xP@A\xP@C
1386     \xP@swapdim\xP@E\xP@G
1387     \xP@swapdim\xP@I\xP@K
1388     \let\xP@varone\xP@tc
1389     \let\xP@varthree\xP@ta
1390     \xP@swapnum\xP@scaleone\xP@scaletwo
1391 \fi

First elimination
1392 \multiply\xP@E\m@ne
1393 \multiply\xP@I\m@ne

Absolute values below are < 8192pt.
1394 \ifdim\xP@A=\z@
1395 \else
1396     \advance\xP@F\dimexpr\xP@B*\xP@E/\xP@A\relax
1397     \advance\xP@G\dimexpr\xP@C*\xP@E/\xP@A\relax
1398     \advance\xP@H\dimexpr\xP@D*\xP@E/\xP@A\relax
1399     \advance\xP@J\dimexpr\xP@B*\xP@I/\xP@A\relax
1400     \advance\xP@K\dimexpr\xP@C*\xP@I/\xP@A\relax
1401     \advance\xP@L\dimexpr\xP@D*\xP@I/\xP@A\relax
1402 \fi

Find the second pivot element. \xP@M is used temporarily.
1403 \count@\m@ne
1404 \xP@ifabsless\xP@F\xP@G\@tempcnta\z@\xP@M\xP@G
1405     \else\@tempcnta\m@ne\xP@M\xP@F\fi
1406 \xP@ifabsless\xP@M\xP@J\@tempcnta\m@ne\count@\z@\xP@M\xP@J\fi
1407 \xP@ifabsless\xP@M\xP@K\@tempcnta\z@\count@\z@\fi

Swap rows
1408 \ifnum\count@=\z@
1409     \xP@swapdim\xP@F\xP@J
1410     \xP@swapdim\xP@G\xP@K
1411     \xP@swapdim\xP@H\xP@L
1412 \fi

Swap columns
1413 \ifnum\@tempcnta=\z@
1414     \xP@swapdim\xP@B\xP@C
1415     \xP@swapdim\xP@F\xP@G
1416     \xP@swapdim\xP@J\xP@K
1417     \let\@tempa\xP@varthree
1418     \let\xP@varthree\xP@vartwo
1419     \let\xP@vartwo\@tempa

```

```

1420 \xP@swapnum\xP@scaletwo\xP@scalethree
1421 \fi
Second elimination. Absolute values are < 16384pt.
1422 \ifdim\xP@F=\z@
1423 \else
1424 \advance\xP@K\dimexpr-\xP@G*\xP@J/\xP@F\relax
1425 \advance\xP@L\dimexpr-\xP@H*\xP@J/\xP@F\relax
1426 \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.
1427 \xP@ifabsless{\dimexpr\xP@L/60\relax}{\dimexpr\xP@K/\xP@scalethree\relax}%
1428 \xP@validsoltrue
1429 \xP@varthree\dimexpr\xP@L*(\xP@scalethree*\p@)/\xP@K\relax
1430 \else
1431 \xP@validsolfalse
1432 \fi
1433 \xP@checkabs{\xP@H/8191}{\xP@F/\xP@scaletwo}%
1434 \xP@checkabs{\xP@G/\xP@scalethree/136}{\xP@F/\xP@scaletwo}%
1435 \ifxP@validsol
1436 \xP@vartwo\dimexpr\xP@H*(\xP@scaletwo*\p@)/\xP@F
1437 -\xP@varthree*\xP@scaletwo/\xP@scalethree*\xP@G/\xP@F\relax
1438 \xP@checkabs\xP@vartwo{60pt}%
1439 \fi
1440 \xP@checkabs{\xP@D/5461}{\xP@A/\xP@scaleone}%
1441 \xP@checkabs{\xP@B/\xP@scaletwo/91}{\xP@A/\xP@scaleone}%
1442 \xP@checkabs{\xP@C/\xP@scalethree/91}{\xP@A/\xP@scaleone}%
1443 \ifxP@validsol
1444 \xP@varone\dimexpr\xP@D*(\xP@scaleone*\p@)/\xP@A
1445 -\xP@vartwo*\xP@scaleone/\xP@scaletwo*\xP@B/\xP@A
1446 -\xP@varthree*\xP@scaleone/\xP@scalethree*\xP@C/\xP@A\relax
1447 \xP@checkabs\xP@varone{60pt}%
1448 \fi
Return the result.
1449 \xdef\@gtempa{%
1450 \ifxP@validsol
1451 \xP@ta\the\xP@ta\relax
1452 \xP@tb\the\xP@tb\relax
1453 \xP@tc\the\xP@tc\relax
1454 \noexpand\xP@validsoltrue
1455 \else
1456 \noexpand\xP@validsolfalse
1457 \fi
1458 }%
1459 }\@gtempa
1460 }

```

\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.

```

1461 \newcommand*\xP@scalerow[4]{%
1462 \xP@M\z@
1463 \xP@Max\xP@M#1%
1464 \xP@Max\xP@M#2%
1465 \xP@Max\xP@M#3%
1466 \xP@Max\xP@M#4%

```

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

```

1467 \count@134217727
1468 \loop
1469   \divide\xP@M\tw@
1470 \ifdim\xP@M>\z@
1471   \divide\count@\tw@
1472 \repeat
1473 \advance\count@\@ne
1474 \multiply#1\count@
1475 \multiply#2\count@
1476 \multiply#3\count@
1477 \multiply#4\count@
1478 }

```

`\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.

```

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

```

$$16777215 = 2048 \cdot 8192 - 1$$

`\xP@checkabs`

```

1495 \newcommand*\xP@checkabs[2]{%
1496   \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.

```

1497 \newcommand*\xP@applinsys{%

```

First iteration: approximate a solution and record the result.

```

1498 \xP@applinsys@
1499 \xP@ta\xP@dta
1500 \xP@tb\xP@dtb
1501 \xP@tc\xP@dtc

```

If the result is nonzero...

```

1502 \xP@checkapp
1503 \if@tempswa

```


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

```
1504 \xP@modobj
1505 \xP@applinsys@
```

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

```
1506 \xP@checkapp
1507 \if@tempwa
1508 \xP@modsol
1509 \xP@modobj
1510 \xP@applinsys@
1511 \xP@modsol
1512 \fi
1513 \fi
```

Return the accumulated approximation from one to three iterations.

```
1514 \xdef\@gtempa{%
1515 \xP@ta\the\xP@ta\relax
1516 \xP@tb\the\xP@tb\relax
1517 \xP@tc\the\xP@tc\relax
1518 }}\@gtempa
1519 }
```

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

```
1520 \newcommand*\xP@checkapp{%
1521 \@tempwattrue
1522 \ifdim\xP@dta=\z@
1523 \ifdim\xP@dtb=\z@
1524 \ifdim\xP@dtc=\z@
1525 \@tempwafalse
1526 \fi\fi\fi
1527 }
```

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

```
1528 \newcommand*\xP@modobj{%
1529 \advance\xP@D
1530 \dimexpr-\xP@A*\xP@dta/\p@-\xP@B*\xP@dtb/\p@-\xP@C*\xP@dtc/\p@\relax
1531 \advance\xP@H
1532 \dimexpr-\xP@E*\xP@dta/\p@-\xP@F*\xP@dtb/\p@-\xP@G*\xP@dtc/\p@\relax
1533 \advance\xP@L
1534 \dimexpr-\xP@I*\xP@dta/\p@-\xP@J*\xP@dtb/\p@-\xP@K*\xP@dtc/\p@\relax
1535 }
```

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

```
1536 \newcommand*\xP@modsol{%
1537 \advance\xP@ta\xP@dta
1538 \advance\xP@tb\xP@dtb
1539 \advance\xP@tc\xP@dtc
1540 }
```

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

```
1541 \newcommand*\xP@applinsys@{%
```

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

```
1542 \xP@sa\z@
1543 \xP@Max\xP@sa\xP@A
1544 \xP@Max\xP@sa\xP@B
```

```

1545 \xP@Max\xP@sa\xP@C
1546 \xP@Max\xP@sa\xP@E
1547 \xP@Max\xP@sa\xP@F
1548 \xP@Max\xP@sa\xP@G
1549 \xP@Max\xP@sa\xP@I
1550 \xP@Max\xP@sa\xP@J
1551 \xP@Max\xP@sa\xP@K
1552 \xP@sa\ifdim\xP@sa<5460pt\thr@@\xP@sa\else\maxdimen\fi
1553 \xP@sb\z@
1554 \xP@Max\xP@sb\xP@D
1555 \xP@Max\xP@sb\xP@H
1556 \xP@Max\xP@sb\xP@L

```

Scale the vector b .

```

1557 \ifdim\xP@sb>\z@
1558 \xP@D\dimexpr\xP@D*\maxdimen/\xP@sb\relax
1559 \xP@H\dimexpr\xP@H*\maxdimen/\xP@sb\relax
1560 \xP@L\dimexpr\xP@L*\maxdimen/\xP@sb\relax
1561 \fi

```

Vector $A^t b$ (scaled)

```

1562 \xP@Aba\dimexpr\xP@A*\xP@D/\xP@sa+\xP@E*\xP@H/\xP@sa+\xP@I*\xP@L/\xP@sa\relax
1563 \xP@Abb\dimexpr\xP@B*\xP@D/\xP@sa+\xP@F*\xP@H/\xP@sa+\xP@J*\xP@L/\xP@sa\relax
1564 \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)

```

1565 \xP@AAba\dimexpr\xP@A*\xP@Aba/\xP@sa+\xP@B*\xP@Abb/\xP@sa
1566 +\xP@C*\xP@Abc/\xP@sa\relax
1567 \xP@AAbb\dimexpr\xP@E*\xP@Aba/\xP@sa+\xP@F*\xP@Abb/\xP@sa
1568 +\xP@G*\xP@Abc/\xP@sa\relax
1569 \xP@AAbc\dimexpr\xP@I*\xP@Aba/\xP@sa+\xP@J*\xP@Abb/\xP@sa
1570 +\xP@K*\xP@Abc/\xP@sa\relax

```

Another scaling factor.

```

1571 \xP@sc\z@
1572 \xP@Max\xP@sc\xP@Aba
1573 \xP@Max\xP@sc\xP@Abb
1574 \xP@Max\xP@sc\xP@Abc
1575 \xP@Max\xP@sc\xP@AAba
1576 \xP@Max\xP@sc\xP@AAbb
1577 \xP@Max\xP@sc\xP@AAbc

```

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

```

1578 \ifdim\xP@sc=\z@
1579 \xP@AAb\z@
1580 \else
1581 \xP@Ab\dimexpr\xP@Aba*\xP@bigdim/\xP@sc*\xP@Aba/\xP@sc
1582 +\xP@Abb*\xP@bigdim/\xP@sc*\xP@Abb/\xP@sc
1583 +\xP@Abc*\xP@bigdim/\xP@sc*\xP@Abc/\xP@sc
1584 \relax
1585 \xP@AAb\dimexpr\xP@AAba*\xP@bigdim/\xP@sc*\xP@AAba/\xP@sc
1586 +\xP@AAbb*\xP@bigdim/\xP@sc*\xP@AAbb/\xP@sc
1587 +\xP@AAbc*\xP@bigdim/\xP@sc*\xP@AAbc/\xP@sc
1588 \relax
1589 \fi

```

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

```

1590 \xdef\@gtempa{\%
1591 \ifdim\xP@AAb=\z@

```

```

1592      \xP@dta\z@
1593      \xP@dtb\z@
1594      \xP@dtc\z@
1595      \else
1596      \xP@dta\the\dimexpr\xP@Aba*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1597      \relax
1598      \xP@dtb\the\dimexpr\xP@Abb*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1599      \relax
1600      \xP@dtc\the\dimexpr\xP@Abc*\xP@sb/\xP@sa*\p@/\xP@AAb*\xP@Ab/\maxdimen
1601      \relax
1602      \fi
1603      }%
1604      }\@gtempa
1605      }

\ifxP@offsetok Switch whether the offset curve is enough
1606 \ifdefinable\ifxP@offsetok\relax
1607 \ifdefinable\xP@offsetoktrue\relax
1608 \ifdefinable\xP@offsetokfalse\relax
1609 \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.
1610 \newcommand*\xP@maxdev{.1pt}

\xP@maxobjfun Tolerance for the objective function. Recommended value is  $\frac{1}{2}(\xP@maxdev)^2$ .
1611 \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  $\backslash X@p, \dots, \backslash Y@c$  is a good approximation for the offset curve of the
partial curve defined by  $\backslash xP@a, \dots, \backslash xP@y$  in the parameter interval  $[\backslash xP@a, \backslash xP@b] \subseteq [0pt, \backslash xP@bigdim]$ .

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  $|w_{par}| + |w_{orth} - \backslash xP@off| \leq \frac{1}{8}\backslash xP@maxdev$ . If  $\|v\|$  is very small so that
the direction cannot be determined precisely, the condition is  $\|c_1(t_i) - c_2(t_i)\| - \backslash xP@off \leq \frac{1}{8}\backslash xP@maxdev$ .
- 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  $|w_{par}| + |w_{orth} - \backslash xP@off| \leq \frac{1}{8}\backslash xP@maxdev$ .
( $w_{par}$  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  $\approx 2^{-17}$  ( $= \frac{1}{2}sp$ ).)


1612 \newcommand*\xP@testoffset{%
```

Default values for the return statement and the loop continuation.

```

1613 \gdef\xP@afteroffsetok{\xP@offsetoktrue}%
1614 \def\xP@offsetokif{\ifdim\xP@ti<1.85pt}%
1615 \xP@ti.1pt
1616 \loop
    \xP@tip =  $t_i$ , denormalized for  $c_1$ 
1617 \xP@tip\dimexpr\xP@a+(\xP@b-\xP@a)*\xP@ti/131072\relax
    Point on the original curve  $c_1$  (scaled by  $-8$ )
1618 \Lp\xP@precbezierpoly\xP@xa\xP@xb\xP@xc\xP@xd\xP@tip
1619 \Up\xP@precbezierpoly\xP@ya\xP@yb\xP@yc\xP@yd\xP@tip
     $8c_2(t_i) - 8c_1(t_i)$ 
1620 \xP@valA\dimexpr\xP@bezierpoly\Xp\Lc\Rc\Xc\xP@ti-\Lp\relax
1621 \xP@valB\dimexpr\xP@bezierpoly\Yp\Up\Uc\Uc\Yc\xP@ti-\Up\relax
     $v$ 
1622 \d@X3\xP@precbeziertan\xP@xa\xP@xb\xP@xc\xP@xd\xP@tip
1623 \d@Y3\xP@precbeziertan\xP@ya\xP@yb\xP@yc\xP@yd\xP@tip
1624 \xP@vecLen
    Decide if  $v$  is big enough (heuristically, may be changed in the future)
1625 \@tempdimc\dimexpr(\xP@b-\xP@a)*\@tempdimb/\xP@bigdim\relax
1626 \xP@abs\@tempdimc
1627 \ifdim.01pt<\@tempdimc
     $8w_{par}, 8w_{orth} - 8$ 
1628 \xP@devA\dimexpr\xP@valA*\d@X/\@tempdimb+\xP@valB*\d@Y/\@tempdimb\relax
1629 \xP@devB\dimexpr\xP@valA*\d@Y/\@tempdimb-\xP@valB*\d@X/\@tempdimb-8\xP@off
1630 \relax
1631 \xP@abs\xP@devA
1632 \xP@abs\xP@devB
1633 \@tempdima\dimexpr\xP@devA+\xP@devB\relax
1634 \else
    If the velocity is zero, just pass the test.
1635 \ifdim\@tempdimc=\z@
1636 \@tempdima\z@
1637 \else
     $8\|c_1(t_i) - c_2(t_i)\|$ 
1638 {\%
1639 \d@X\xP@valA
1640 \d@Y\xP@valB
1641 \xP@vecLen@
1642 \global\dimen@i\@tempdimb
1643 }\@tempdima\dimen@i
1644 \advance\@tempdima\ifdim\xP@off>\z@-\fi8\xP@off
1645 \xP@abs\@tempdima
1646 \fi
1647 \fi
    If the first condition is not fulfilled, test the second one.
1648 \ifdim\@tempdima>\xP@maxdev
     $-c_1(t_i)$ 
1649 \divide\Lp-8\relax
1650 \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.

```

1651      {%
1652      \xP@transformcoor\X@p\Y@p
1653      \xP@transformcoor\L@c\U@c
1654      \xP@transformcoor\R@c\D@c
1655      \xP@transformcoor\X@c\Y@c

```

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

```

1656      \xP@findzero
1657      }%
1658      \fi
1659      \xP@offsetokif
1660      \advance\xP@ti.1pt
1661      \repeat
1662      \expandafter}\xP@afteroffsetok
1663 }

```

\xP@afteroffsetok

```

1664 \newcommand*\xP@afteroffsetok{}

```

\xP@offsetokif

```

1665 \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 $\backslash @tempdimb$ must contain the length $\|v\|$.

```

1666 \newcommand*\xP@transformcoor[2]{%
1667   \advance#1\backslash L@p
1668   \advance#2\backslash U@p
1669   \@tempdima\dimexpr#1*\backslash d@X/\@tempdimb+#2*\backslash d@Y/\@tempdimb\relax
1670   #2\dimexpr#2*\backslash d@X/\@tempdimb-#1*\backslash d@Y/\@tempdimb\relax
1671   #1\@tempdima
1672 }

```

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

```

1673 \newcommand*\xP@findzero{%
1674   \xP@setleftvalue{.05}%
1675   \xP@setrightvalue{.05}%

```

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

```

1676   \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.

```

1677   \ifdim\xP@valA>\z@
1678     \@tempswatrue
1679     \@for\@tempa:={.1,.15,.2,.25,.35,.5,1.1}\do{%
1680       \if@tempswa
1681         \xP@setleftvalue\@tempa
1682         \ifdim\xP@valA<\z@\@tempswafalse\fi
1683       \if@tempswa
1684         \xP@setrightvalue\@tempa
1685       \ifdim\xP@valB<\z@
1686         \@tempswafalse

```

```

1687         \xP@reversecoeff
1688     \fi
1689     \fi
1690 \fi
1691 }%

```

Last resort: Try the midpoint.

```

1692 \if@tempwa
1693     \L@p\xP@ti
1694     \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.

```

1695     \ifdim\xP@valA<\z@

```

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

```

1696         \xP@setrightvalue{.05}%
1697     \else
1698         \U@p\L@p
1699         \xP@valB\xP@valA
1700     \fi
1701 \fi
1702 \fi

```

The actual nested interval algorithm

```

1703 \loop
1704 \ifnum\numexpr\U@p-\L@p\relax>\@ne
1705     \xP@ti\dimexpr(\L@p+\U@p)/2\relax
1706     \xP@devA\xP@bezierpoly\X@p\L@c\R@c\X@c\xP@ti
1707     \ifdim\xP@devA>\z@
1708         \U@p\xP@ti
1709         \xP@valB\xP@devA
1710     \else
1711         \L@p\xP@ti
1712         \xP@valA\xP@devA
1713     \fi
1714 \repeat

```

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

```

1715 \xP@ifabsless\xP@valB\xP@valA
1716     \L@p\U@p
1717     \xP@valA\xP@valB
1718 \fi

```

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

```

1719 \xP@valB\dimexpr\xP@bezierpoly\Y@p\U@c\D@c\Y@c\L@p+8\xP@off\relax
1720 \xP@abs\xP@valA
1721 \xP@abs\xP@valB
1722 \ifdim\dimexpr\xP@valA+\xP@valB\relax>\xP@maxdev\relax
1723     \xP@failed
1724 \fi
1725 }

```

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

```

1726 \newcommand*\xP@failed{%
1727     \global\let\xP@offsetokif\iffalse
1728     \gdef\xP@afteroffsetok{\xP@offsetokfalse}%
1729 }

```

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

```
1730 \newcommand*\xP@reversecoeff{%
1731   \multiply\xP@m@ne
1732   \multiply\L@c@m@ne
1733   \multiply\R@c@m@ne
1734   \multiply\X@c@m@ne
1735   \multiply\xP@valA@m@ne
1736   \multiply\xP@valB@m@ne
1737 }
```

`\xP@setleftvalue` •5

```
1738 \newcommand*\xP@setleftvalue[1]{%
1739   \L@p\dimexpr\xP@ti-#1\p@\relax
1740   \ifdim\L@p<-.1pt\L@p-.1pt\fi
1741   \xP@valA\xP@bezierpoly\X@p\L@c\R@c\X@c\L@p
1742 }
```

`\xP@setrightvalue` •5

```
1743 \newcommand*\xP@setrightvalue[1]{%
1744   \U@p\dimexpr\xP@ti+#1\p@\relax
1745   \ifdim\U@p>2.1\p@\U@p2.1\p@\fi
1746   \xP@valB\xP@bezierpoly\X@p\L@c\R@c\X@c\U@p
1747 }
```

8.10 Multiple dashed curves

`\xP@splinedbldashed`

```
1748 \newcommand*\xP@splinedbldashed{%
1749   \xP@checkspline\xP@splinemultdashed\xP@doublestroke}
```

`\xP@splinetrbldashed`

```
1750 \newcommand*\xP@splinetrbldashed{%
1751   \xP@checkspline\xP@splinemultdashed\xP@trblstroke}
```

`\xP@splinemultdashed`

```
1752 \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.

```
1753   \xP@testcont\xP@dashmacro
1754   \@tempcnta
1755   \ifxP@splinecont
1756     \numexpr2*((\@tempdimb-\xydashl@/3)/(2*\xydashl@))\relax
1757   \else
1758     \numexpr2*((\@tempdimb+\xydashl@)/(2*\xydashl@))-1\relax
1759   \fi
1760   \ifnum\@tempcnta>\@ne
1761     \xP@splinemultdashed@#1%
1762   \else
```

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

```
1763   \ifnum\@tempcnta=\@ne
1764     \xP@splinemultsolid#1
1765   \else
1766     \xP@savec
```

```

1767 \fi
1768 \fi
1769 \global\let\xP@lastpattern\xP@dashmacro
1770 }

```

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

```

1771 \newcommand*\xP@splinemultdashed@[1]{%
1772 \xP@inibigdim
Dash length
1773 \@tempdima\dimexpr\@tempdimb/\@tempcnta\relax
1774 \xP@temppar\z@
1775 \toks@{}%
1776 \xP@savec
1777 \ifodd\@tempcnta
1778 \else
1779 \xP@slide
1780 \fi
1781 \@tempcnta\z@
1782 \loop
1783 \advance\@tempcnta\@ne
1784 \xP@append\toks@{\ifodd\@tempcnta\noexpand\xP@paintdash\fi
1785 \the\xP@temppar}}%
1786 \xP@oldpar\xP@temppar
1787 \xP@slide
1788 \ifdim\xP@temppar<\xP@bigdim
1789 \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.

```

1790 \xP@lastpar
1791 \ifodd\@tempcnta
1792 \xP@temppar
1793 \xP@append\toks@{\the\xP@temppar}}%
1794 \else
1795 \xP@oldpar
1796 \fi

```

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

```

1797 \@temptokena{}%
1798 \xP@setsolidpat
1799 \global\let\xP@lastpattern\xP@dashmacro
1800 \@for\@tempa:={#1}\do{\the\toks@}%
1801 \xP@stroke{\the\@temptokena}%
1802 }}

```

`\xP@paintdash` ●1 ●7

```

1803 \newcommand*\xP@paintdash[2]{%
1804 \xP@paintsolid{\dimexpr#1*\xP@bigdim/\xP@lastpar\relax}%
1805 {\dimexpr#2*\xP@bigdim/\xP@lastpar\relax}%
1806 }

```

8.11 Multiple dotted curves

```

\splinedbldotted@
\xP@splinedbldotted@ 1807 \xP@hook{curve}{splinedbldotted@}
1808 \newcommand*\xP@splinedbldotted@{%

```



```

1809 \let\xP@normalmult\@ne
1810 \xP@checkspline\xP@splinemultdotted\xP@doublestroke}

\xP@splinetrbldotted

1811 \newcommand*\xP@splinetrbldotted{%
1812 \let\xP@normalmult\tw@
1813 \xP@checkspline\xP@splinemultdotted\xP@trblstroke}

\xP@multdottedpat 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.

1814 \newcommand*\xP@multdottedpat{%
1815 \def\xP@pattern{0 J [\xP@coord\xP@preclw{\xydashh@-\xP@preclw}]0 d}%
1816 \global\let\xP@lastpattern\xP@dotmacro
1817 }

\xP@normalmult

1818 \@ifdefinable\xP@normalmult\relax

\xP@splinemultdotted ●1●7

1819 \newcommand\xP@splinemultdotted[1]{%
1820 \xP@inibigdim
    Make a list of dot positions on the spline segment.
1821 \xP@temppar\z@
1822 \xP@testcont\xP@dotmacro
1823 \ifxP@splinecont
    Expected dot distance (see the formula in \xP@setdottedpat)
1824 \@tempdimc\dimexpr\@tempdimb/(\@tempdimb/131072+1)\relax
1825 \@tempdima\dimexpr\@tempdimc-\xP@preclw/2\relax
1826 \xP@slide
1827 \@tempdima\@tempdimc
1828 \else
1829 \@tempdima\dimexpr\xP@preclw/2\relax
1830 \xP@slide
    Expected dot distance (see the formula in \xP@setdottedpat)
1831 \@tempdima\dimexpr\@tempdimb-\xP@preclw\relax
1832 \ifdim\@tempdima<\z@\@tempdima\z@\fi
1833 \@tempdima\dimexpr\@tempdima/(\@tempdima/131072+1)\relax
1834 \fi
1835 \xP@savvec
1836 \toks@{}%
    If the end of the segment is reached before the first dot position, leave the segment out.
1837 \ifdim\xP@temppar<\xP@bigdim
1838 \loop
1839 \xP@append\toks@{\noexpand\xP@paintdot{\the\xP@temppar}}%
1840 \xP@oldpar\xP@temppar
1841 \xP@slide
1842 \ifdim\xP@temppar<\xP@bigdim
1843 \repeat
1844 \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.

```

1845 \ifdim
1846 \ifdim\xP@preclw<\xP@tempvel
1847 \dimexpr2\xP@bigdim-\xP@oldpar-\xP@preclw*\xP@bigdim/\xP@tempvel\relax
1848 \else
1849 -\maxdimen
1850 \fi<\xP@temppar
1851 \xP@temppar\xP@oldpar
1852 \else
1853 \xP@append\toks@{\noexpand\xP@paintdot{\the\xP@temppar}}}%
1854 \fi
1855 \@tempdima\dimexpr\xP@preclw/2\relax
1856 \xP@slide
1857 \xP@lastpar\xP@temppar

```

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

```

1858 \@temptokena{}%
1859 \the\toks@

```

Actually draw the points in the list.

```

1860 \xP@multidottedpat
1861 \xP@stroke{\the\@temptokena}%
1862 \else

```

Leave the segment out because it is too short.

```

1863 \global\let\xP@lastpattern\@empty
1864 \fi
1865 }}

```

`\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`.

```

1866 \newcommand*\xP@slide{%
1867 \xP@slide@

```

Return the new spline parameter after sliding.

```

1868 \global\dimen@i\xP@temppar
1869 }\xP@temppar\dimen@i
1870 }

```

`\xP@slide@` ●1 ●7

```

1871 \newcommand*\xP@slide@{%

```

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

```

1872 \xP@velocity\xP@temppar\xP@tempvel

```

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

```

1873 \@tempdimc\dimexpr\xP@bigdim*\@tempdima/\@tempdimb\relax
1874 \count@z@
1875 \@tempswatru

```

Improve the parameter increment iteratively.

```

1876 \loop

```

Velocity at the estimated end point.

```

1877 \xP@velocity{\xP@temppar+\@tempdimc}\xP@tempvel@

```

Prevent arithmetic overflow.

```

1878     \ifdim\dimexpr\@tempdima*4/13\relax>\xP@tempvel@
1879         \@tempwafalse
1880     \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).

```

1881         \xP@parinc\dimexpr\@tempdima*\xP@bigdim/\xP@tempvel@
1882         -(\xP@tempvel+\xP@tempvel@)/2*\@tempdimc/\xP@tempvel@\relax
1883         \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.

```

1884         \ifdim\@tempdimc>.12\xP@bigdim
1885             \@tempwafalse
1886         \else

```

If the estimate is not improved, break the loop.

```

1887             \ifdim\xP@parinc=\z@
1888                 \@tempwafalse
1889             \else

```

Also break the loop after 10 iterations.

```

1890                 \ifnum\count@=9\relax
1891                     \@tempwafalse
1892                 \fi
1893             \fi
1894         \fi
1895     \fi
1896     \if@tempswa
1897         \advance\count@\@ne
1898     \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.

```

1899     \ifdim\xP@temppar<5461pt
1900     \ifdim\@tempdimc>.1\xP@bigdim
1901         \@tempwatrue
1902     \fi
1903     \fi
1904     \if@tempswa
1905         {%
1906             \dimen5\xP@temppar
1907             \advance\xP@temppar.1\xP@bigdim

```

Cap the end parameter to prevent arithmetic overflows.

```

1908         \ifdim\xP@temppar>5461pt\xP@temppar5461pt\fi
1909         \dimen7\xP@temppar

```

Determine the exact distance of the partial slide.

```

1910         \xP@shaveprec{\dimen5}{\dimen7}%
1911         \xP@bezierlength
1912         \global\dimen@i\dimexpr\@tempdima-\@tempdimb\relax
1913         \global\dimen3\xP@temppar
1914     }%
1915     \@tempdima\dimen@i
1916     \xP@temppar\dimen3\relax

```

Slide again.

```
1917 \expandafter\xP@slide@
1918 \else
```

Finish the slide and return the new parameter.

```
1919 \advance\xP@temppar\@tempdimc
1920 \fi
1921 }
```

`\xP@paintdot` ●1 ●7

```
1922 \newcommand*\xP@paintdot[1]{%
```

Scale the parameter with a correction factor

```
1923 \@tempdima\dimexpr#1*\xP@bigdim/\xP@lastpar\relax
```

Position at parameter value `\xP@temppar`

```
1924 \xP@tangent
1925 \xP@posX\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\@tempdima/8\relax
1926 \xP@posY\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\@tempdima/8\relax
```

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

```
1927 \@tempdima\dimexpr(\xydashh@+\xP@preclw/\xP@normalmult)/2\relax
1928 \L@p\dimexpr\d@Y*\@tempdima/\@tempdimb\relax
1929 \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@`.)

```
1930 \xP@append\@temptokena{\xP@coor{\xP@posX+\L@p*\xP@normalmult}}%
1931 {\xP@posY+\U@p*\xP@normalmult}m %
1932 \xP@coor{\xP@posX-\L@p*(\xP@normalmult+\@ne)}}%
1933 {\xP@posY-\U@p*(\xP@normalmult+\@ne)}l }%
1934 }
```

8.12 Squiggled curves

`\xP@splinesquiggled`

```
1935 \newcommand*\xP@splinesquiggled{%
1936 \xP@checkspline\xP@splinesquiggled@z@}
```

`\xP@splinedblsquiggled`

```
1937 \newcommand*\xP@splinedblsquiggled{%
1938 \xP@checkspline\xP@splinesquiggled@\xP@doublestroke}
```

`\xP@splinetrbldsquiggled`

```
1939 \newcommand*\xP@splinetrbldsquiggled{%
1940 \xP@checkspline\xP@splinesquiggled@\xP@trblstroke}
```

`\xP@splinesquiggled@` ●1 ●7

```
1941 \newcommand*\xP@splinesquiggled@[1]{%
1942 \xP@inibigdim
```

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

```
1943 \xP@testcont\xP@oddsquigglemacro
1944 \ifxP@splinecont
1945 \def\xP@squigsign{-}%
1946 \else
```

```

1947      \let\xP@squigsign\@empty
1948      \fi
1949      \xP@savvec
Expected squiggle length
1950      \@tempcnta=\numexpr\@tempdimb/\xybsqll\relax
1951      \ifnum\@tempcnta<\tw@\@tempcnta\tw@\fi
1952      \multiply\@tempcnta\tw@
1953      \@tempdima\dimexpr\@tempdimb/\@tempcnta\relax
1954      \xP@squiglen\@tempdima
Make a list of dot positions on the spline segment.
1955      \xP@temppar\z@
1956      \toks@{}%
1957      \@tempcnta\z@
1958      \loop
1959      \advance\@tempcnta\@ne
1960      \xP@append\toks@{\noexpand\xP@paintsquiggle{\the\xP@temppar}}%
1961      \xP@oldpar\xP@temppar
1962      \xP@slide
1963      \ifdim\xP@temppar<\xP@bigdim
1964      \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.

```

1965      \xP@lastpar
1966      \ifodd\@tempcnta
1967      \xP@oldpar
1968      \advance\@tempcnta\m@ne
1969      \else
1970      \xP@temppar
1971      \xP@append\toks@{\noexpand\xP@paintsquiggle{\the\xP@temppar}}%
1972      \fi

```

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

```

1973      \@temptokena{}%
1974      \xP@setsolidpat

```

Record the direction of the last squiggle.

```

1975      \global\expandafter\let\expandafter\xP@lastpattern
1976      \ifodd\numexpr\@tempcnta/2\if\xP@squigsign-+1\fi\relax
1977      \xP@oddsquigglemacro
1978      \else
1979      \xP@evensquigglemacro
1980      \fi

```

Draw the squiggles.

```

1981      \@for\@tempa:={#1}\do{%
1982      \let\xP@dosquiggle\xP@dosquiggle@
1983      \count@\z@
1984      \the\toks@
1985      }%
1986      \xP@stroke{\the\@temptokena}%
1987  }}

```

\xP@paintsquiggle ●1 ●7

```

1988 \newcommand*\xP@paintsquiggle[1]{%
1989   \xP@squigglevectors{#1}%

```

```

1990 \xP@dosquiggle
1991 \ifnum\count@=\thr@@\relax\count@\z@\else\advance\count@\@ne\fi
1992 }

```

\xP@squigglevectors ●1 ●7

```

1993 \newcommand*\xP@squigglevectors[1]{%
  Scale the parameter with a correction factor
1994 \@tempdima\dimexpr#1*\xP@bigdim/\xP@lastpar\relax
  Position at parameter value \xP@temppar, offset for multiple curves.
1995 \xP@tangent
1996 \xP@posX\dimexpr\xP@precbezierpoly\X@p\L@c\R@c\X@c\@tempdima/8%
1997 -\d@Y*(\@tempa)/\@tempdimb\relax
1998 \xP@posY\dimexpr\xP@precbezierpoly\Y@p\U@c\D@c\Y@c\@tempdima/8%
1999 +\d@X*(\@tempa)/\@tempdimb\relax
  Tangent vector to the curve with correct length
2000 \L@p\dimexpr\d@X*\xP@squiglen/\@tempdimb\relax
2001 \U@p\dimexpr\d@Y*\xP@squiglen/\@tempdimb\relax
2002 \R@p\dimexpr\L@p*543339720/1311738121\relax
2003 \D@p\dimexpr\U@p*543339720/1311738121\relax
2004 \X@min\dimexpr\L@p*362911648/967576667\relax
2005 \Y@min\dimexpr\U@p*362911648/967576667\relax
2006 \X@max\dimexpr(\L@p+\xP@squigsign\U@p)*173517671/654249180\relax
2007 \Y@max\dimexpr(\L@p-\xP@squigsign\U@p)*173517671/654249180\relax
2008 }

```

\xP@dosquiggle ●7

```

2009 \@ifdefinable\xP@dosquiggle@\relax

```

\xP@dosquiggle@ ●7

```

2010 \newcommand*\xP@dosquiggle@{%
2011 \edef\next@{\xP@coor{\xP@posX}{\xP@posY}m
2012 \xP@coor{\xP@posX+\Y@max}{\xP@posY+\xP@squigsign\X@max}%
2013 }%
2014 \let\xP@dosquiggle\xP@dosquiggle@@
2015 }

```

\xP@dosquiggle@@ ●7

```

2016 \newcommand*\xP@dosquiggle@@{%
2017 \xP@append\@temptokena{\next@\expandafter\xP@coor
2018 \ifcase\count@
2019 {\xP@posX-\Y@max}{\xP@posY-\xP@squigsign\X@max}%
2020 \xP@coor\xP@posX\xP@posY
2021 \or
2022 {\xP@posX-\xP@squigsign\D@p-\X@min}{\xP@posY+\xP@squigsign\R@p-\Y@min}%
2023 \xP@coor{\xP@posX-\xP@squigsign\D@p}{\xP@posY+\xP@squigsign\R@p}%
2024 \or
2025 {\xP@posX-\X@max}{\xP@posY+\xP@squigsign\Y@max}%
2026 \xP@coor\xP@posX\xP@posY
2027 \or
2028 {\xP@posX+\xP@squigsign\D@p-\X@min}{\xP@posY-\xP@squigsign\R@p-\Y@min}%
2029 \xP@coor{\xP@posX+\xP@squigsign\D@p}{\xP@posY-\xP@squigsign\R@p}%
2030 \fi c }%
2031 \edef\next@{\expandafter\xP@coor
2032 \ifcase\count@

```

```

2033      {\xP@posX+\Y@max}{\xP@posY+\xP@squigsign\X@max}%
2034      \or
2035      {\xP@posX-\xP@squigsign\D@p+\X@min}{\xP@posY+\xP@squigsign\R@p+\Y@min}%
2036      \or
2037      {\xP@posX+\X@max}{\xP@posY-\xP@squigsign\Y@max}%
2038      \or
2039      {\xP@posX+\xP@squigsign\D@p+\X@min}{\xP@posY-\xP@squigsign\R@p+\Y@min}%
2040      \fi
2041    }%
2042  }

```

\xP@splinebrokensquiggled

```

2043 \newcommand*\xP@splinebrokensquiggled{%
2044   \xP@checkspline\xP@splinebrokensquiggled@{z@}

```

\xP@splinebrokendsquiggled

```

2045 \newcommand*\xP@splinebrokendsquiggled{%
2046   \xP@checkspline\xP@splinebrokensquiggled@\xP@doublestroke}

```

\xP@splinebrokentrbqsquiggled

```

2047 \newcommand*\xP@splinebrokentrbqsquiggled{%
2048   \xP@checkspline\xP@splinebrokensquiggled@\xP@trblstroke}

```

\xP@splinebrokensquiggled@ ●1 ●7

```

2049 \newcommand*\xP@splinebrokensquiggled@[1]{%
2050   \xP@inibigdim

```

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

```

2051   \xP@testcont\xP@brokensquigglemacro
2052   \ifxP@splinecont
2053     \let\xP@squigsign\@firstoftwo
2054   \else
2055     \let\xP@squigsign\@secondoftwo
2056   \fi
2057   \xP@savvec

```

Expected squiggle length

```

2058   \@tempcnta\numexpr(\@tempdimb\xP@squigsign\@{+2*\xybsql1@})%
2059   /(4*\xybsql1@)\relax
2060   \ifnum\@tempcnta<\@ne\@tempcnta\@ne\fi
2061   \@tempdima\dimexpr\@tempdimb/(8*\@tempcnta\xP@squigsign\@{-4})\relax
2062   \xP@squiglen\@tempdima

```

Make a list of dot positions on the spline segment.

```

2063   \xP@temppar{z@}
2064   \xP@squigsign{\xP@slide\xP@slide\xP@slide\xP@slide}{}%
2065   \count@\z@
2066   \toks@{}%
2067   \loop
2068     \xP@append\toks@{\noexpand\xP@paintbrokensquiggle{\the\xP@temppar}}%
2069     \xP@slide
2070     \xP@append\toks@{\the\xP@temppar}%
2071     \xP@slide
2072     \xP@append\toks@{\the\xP@temppar}%
2073     \xP@slide
2074     \xP@append\toks@{\the\xP@temppar}%
2075     \xP@slide

```

```

2076      \xP@append\toks@{\the\xP@temppar}}%
2077      \xP@lastpar\xP@temppar
2078      \advance\count@\@ne
2079      \ifnum\count@<\@tempcnta
2080      \xP@slide
2081      \xP@slide
2082      \xP@slide
2083      \xP@slide
2084      \repeat

```

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

```

2085      \@temptokena{%
2086      \xP@setsolidpat
2087      \global\let\xP@lastpattern\xP@brokensquigglemacro

```

Draw the squiggles.

```

2088      \let\xP@squigsig\@empty
2089      \@for\@tempa:={#1}\do{\the\toks@}%
2090      \xP@stroke{\the\@temptokena}%
2091  }}

```

\xP@paintbrokensquiggle ●1●7

```

2092 \newcommand*\xP@paintbrokensquiggle[5]{%
2093   \xP@squigglevectors{#1}%
2094   \xP@append\@temptokena{%
2095     \xP@coor\xP@posX\xP@posY m %
2096     \xP@coor{\xP@posX+\Y@max}{\xP@posY+\X@max}%
2097   }%
2098   \xP@squigglevectors{#2}%
2099   \xP@append\@temptokena{%
2100     \xP@coor{\xP@posX-\D@p-\X@min}{\xP@posY+\R@p-\Y@min}%
2101     \xP@coor{\xP@posX-\D@p}{\xP@posY+\R@p}c %
2102     \xP@coor{\xP@posX-\D@p+\X@min}{\xP@posY+\R@p+\Y@min}%
2103   }%
2104   \xP@squigglevectors{#3}%
2105   \xP@append\@temptokena{%
2106     \xP@coor{\xP@posX-\X@max}{\xP@posY+\Y@max}%
2107     \xP@coor\xP@posX\xP@posY c %
2108     \xP@coor{\xP@posX+\X@max}{\xP@posY-\Y@max}%
2109   }%
2110   \xP@squigglevectors{#4}%
2111   \xP@append\@temptokena{%
2112     \xP@coor{\xP@posX+\D@p-\X@min}{\xP@posY-\R@p-\Y@min}%
2113     \xP@coor{\xP@posX+\D@p}{\xP@posY-\R@p}c %
2114     \xP@coor{\xP@posX+\D@p+\X@min}{\xP@posY-\R@p+\Y@min}%
2115   }%
2116   \xP@squigglevectors{#5}%
2117   \xP@append\@temptokena{%
2118     \xP@coor{\xP@posX-\Y@max}{\xP@posY-\X@max}%
2119     \xP@coor\xP@posX\xP@posY c %
2120   }%
2121 }

```

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

```

2122 \xyendinput
2123 </curve>
2124 <*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 2125 \newcommand*\XP@lastX{}
\XP@lastpattern 2126 \newcommand*\XP@lastY{}
2127 \newcommand*\XP@lastpattern{}

\XP@solidmacro
\XP@dotmacro 2128 \newcommand*\XP@solidmacro{solid}
\XP@dashmacro 2129 \newcommand*\XP@dotmacro{dot}
\XP@evensquigglemacro 2130 \newcommand*\XP@dashmacro{dash}
\XP@oddsquigglemacro 2131 \newcommand*\XP@evensquigglemacro{evensquiggle}
\XP@brokensquigglemacro 2132 \newcommand*\XP@oddsquigglemacro{oddsquiggle}
2133 \newcommand*\XP@brokensquigglemacro{brokensquiggle}

\xyinside@ Reset the last position with every new diagram.
2134 \renewcommand*\xyinside@{%
2135   \global\let\XP@lastpattern\@empty
2136   \saveXyStyle@ \aftergroup\xycheck@end
2137   \setboxz\h\bgroup
2138   \plainxy@
2139   \X@c=\z@\Y@c=\z@\czeroEdge@
2140   \X@p=\z@\Y@p=\z@\U@p=\z@\D@p=\z@\L@p=\z@\R@p=\z@\Edge@p={\zeroEdge}%
2141   \X@min=\hsize\X@max=-\hsize\Y@min=\hsize\Y@max=-\hsize
2142   \mathsurround=\z@
2143   \expandafter\POS\everyxy@@
2144 }

\XP@savec Save the current end point
2145 \newcommand*\XP@savec{%
2146   \xdef\XP@lastX{\the\X@c}%
2147   \xdef\XP@lastY{\the\Y@c}%
2148 }

\ifxP@splinecont Switch: does the next line/spline continue at the end point of the last one?
2149 \@ifdefinable\ifxP@splinecont\relax
2150 \@ifdefinable\XP@splineconttrue\relax
2151 \@ifdefinable\XP@splinecontfalse\relax
2152 \newif\ifxP@splinecont

\XP@testcont Test for \ifxP@splinecont
2153 \newcommand*\XP@testcont[1]{%
2154   \XP@splinecontfalse
2155   \ifxP@cont
2156     \ifx\XP@lastpattern#1%
2157       \ifdim\XP@lastX=\X@p
2158         \ifdim\XP@lastY=\Y@p
2159           \XP@splineconttrue
2160         \fi
2161       \fi
2162     \fi
2163   \fi
2164 }
```

\ifxP@cont Switch: shall the spline hack be applied?

```
2165 \@ifdefinable\ifxP@cont\relax
2166 \@ifdefinable\xP@conttrue\relax
2167 \@ifdefinable\xP@contfalse\relax
2168 \newif\ifxP@cont
```

\xypdfcontpatternon

```
\xypdfcontpatternoff 2169 \newcommand*\xypdfcontpatternon{\xP@conttrue}
2170 \newcommand*\xypdfcontpatternoff{\xP@contfalse}
2171 \xP@conttrue
2172 \</basic>
```

8.14 Color

```
2173 \< *color>
2174 \xycatcodes
```

\xypdf-co@loaded

```
2175 \expandafter\let\csname xypdf-co@loaded\endcsname\@empty
```

\xP@colorname

```
\xP@colA 2176 \@ifdefinable\xP@colorname\relax
\xP@colB 2177 \@ifdefinable\xP@colA\relax
\xP@colC 2178 \@ifdefinable\xP@colB\relax
\xP@colD 2179 \@ifdefinable\xP@colC\relax
2180 \@ifdefinable\xP@colD\relax
```

\newxycolor

```
\xP@newxycolor 2181 \xP@hook{color}{\newxycolor}
2182 \newcommand*\xP@newxycolor[2]{%
2183   \def\xP@colorname{#1}%
2184   \xP@parsecolor#2 @%
2185 }
```

\xP@parsecolor

```
2186 \@ifdefinable\xP@parsecolor\relax
2187 \def\xP@parsecolor#1 #2 #3@{%
2188   \def\xP@colA{#1}%
2189   \def\xP@colB{#2}%
2190   \ifx\xP@colB\xP@gray
2191     \xP@newcolor\xP@colorname\xP@colA\xP@gray\newxycolor
2192   \else
2193     \xP@parsecolor@#3 @%
2194   \fi
2195 }
```

\xP@parsecolor@

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

```

2205     \xP@newcolor\xP@colorname{\xP@colA,\xP@colB,\xP@colC,\xP@colD}{cmyk}%
2206     \newxycolor
2207   \else
2208     \PackageError{xypdf}{Syntax error in \string\newxycolor}{}%
2209   \fi
2210 \fi
2211 }

\xP@gray
\xP@rgb 2212 \newcommand*\xP@gray{gray}
\xP@cmyk 2213 \newcommand*\xP@rgb{rgb}
2214 \newcommand*\xP@cmyk{cmyk}

\OBJECT@shape
\xP@OBJECT@shape 2215 \xP@hook{color}{OBJECT@shape}
2216 \newcommand*\xP@OBJECT@shape[1]{\DN@{shape [1]}}%
2217 \expandafter\let\expandafter\nextii@\csname\codeof\next@\endcsname
2218 \ifx\nextii@\relax\DN@{style [1]}%
2219   \expandafter\let\expandafter\nextii@\csname\codeof\next@\endcsname
2220   \ifx\nextii@\relax\DN@{\xP@checkcolor{1}}%
2221   \else\DN@{\nextii@\xyFN@OBJECT@}%
2222   \fi
2223 \else\expandafter\addtotoks@\expandafter{\nextii@}%
2224   \DN@{\xyFN@OBJECT@}%
2225 \fi \next@}

\xP@checkcolor
2226 \newcommand*\xP@checkcolor[1]{%
2227   \@ifundefined{\string\color@detokenize{1}}{%
2228     {\OBJECT@shapei[1]}%
2229     {%
2230       \xP@append\toks@{\noexpand\xP@color{\detokenize{1}}}%
2231       \xyFN@OBJECT@
2232     }%
2233 }

\xP@color
2234 \newcommand*\xP@color[1]{%
2235   \def\preStyle@{\addtostyletoks@{\bgroup\xP@protectedcolor#1}}%
2236   \def\postStyle@{\addtostyletoks@{\egroup}}%
2237   \modXYstyle@
2238 }

\xP@protectedcolor Somehow, \protect\color is not enough, so we use  $\varepsilon$ -TEX's way of \protected macro
definitions.
2239 \@ifdefinable\xP@protectedcolor\relax
2240 \protected\def\xP@protectedcolor{%
2241   \@ifundefined{color}\xP@pdfcolor\color
2242 }

\xP@pdfcolor
2243 \@ifdefinable\xP@pdfcolor\relax
2244 \def\xP@pdfcolor[#1]#2{%
2245   \edef\@tempa{#1}%
2246   \ifx\@tempa\xP@gray
2247     \DN@{\xP@graycolor{#2}}%

```

```

2248 \else\ifx\@tempa\xP@rgb
2249 \DN@{\xP@rgbcolor#2@}%
2250 \else\ifx\@tempa\xP@cmyk
2251 \DN@{\xP@cmykcolor#2@}%
2252 \fi\fi\fi
2253 \aftergroup\xP@resetcolor
2254 \next@
2255 }%

\xP@graycolor
2256 \newcommand*\xP@graycolor[1]{\xP@setcolor{#1}gG}%

\xP@rgbcolor
2257 \@ifdefinable\xP@rgbcolor\relax
2258 \def\xP@rgbcolor#1,#2,#3@{\xP@setcolor{#1 #2 #3}{rg}{RG}}

\xP@cmykcolor
2259 \@ifdefinable\xP@cmykcolor\relax
2260 \def\xP@cmykcolor#1,#2,#3,#4@{\xP@setcolor{#1 #2 #3 #4}kK}

\xP@newcolor
2261 \newcommand*\xP@newcolor[4]{%
2262 \expandafter\let\expandafter\next@\csname shape [1]\endcsname
2263 \ifx\next@\relax
2264 \ifundefined{string\color@#1}\relax
2265 {\xP@warning{xypdf}{The color ‘#1’ is overridden by %
2266 \string#4}}%
2267 \DN@{\newxystyle{#1}{\xP@unnamedcolor{#2}{#3}}}%
2268 \else
2269 \DN@{}%
2270 \fi
2271 \next@\relax}

\xP@unnamedcolor
2272 \newcommand*\xP@unnamedcolor[2]{\xP@color{[#2]{#1}}}%

\xP@definecrayolacolor
2273 \newcommand\xP@definecrayolacolor[2]{%
2274 \xP@newcolor{#1}{#2}{cmyk}\UseCrayolaColors}%

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

```

```

2290 \xP@definecrayolacolor{BrickRed}{0,.89,.94,.28}%
2291 \xP@definecrayolacolor{Red}{0,1,1,0}%
2292 \xP@definecrayolacolor{OrangeRed}{0,1,.5,0}%
2293 \xP@definecrayolacolor{RubineRed}{0,1,.13,0}%
2294 \xP@definecrayolacolor{WildStrawberry}{0,.96,.39,0}%
2295 \xP@definecrayolacolor{Salmon}{0,.53,.38,0}%
2296 \xP@definecrayolacolor{CarnationPink}{0,.63,0,0}%
2297 \xP@definecrayolacolor{Magenta}{0,1,0,0}%
2298 \xP@definecrayolacolor{VioletRed}{0,.81,0,0}%
2299 \xP@definecrayolacolor{Rhodamine}{0,.82,0,0}%
2300 \xP@definecrayolacolor{Mulberry}{.34,.9,0,.02}%
2301 \xP@definecrayolacolor{RedViolet}{.07,.9,0,.34}%
2302 \xP@definecrayolacolor{Fuchsia}{.47,.91,0,.08}%
2303 \xP@definecrayolacolor{Lavender}{0,.48,0,0}%
2304 \xP@definecrayolacolor{Thistle}{.12,.59,0,0}%
2305 \xP@definecrayolacolor{Orchid}{.32,.64,0,0}%
2306 \xP@definecrayolacolor{DarkOrchid}{.4,.8,.2,0}%
2307 \xP@definecrayolacolor{Purple}{.45,.86,0,0}%
2308 \xP@definecrayolacolor{Plum}{.5,1,0,0}%
2309 \xP@definecrayolacolor{Violet}{.79,.88,0,0}%
2310 \xP@definecrayolacolor{RoyalPurple}{.75,.9,0,0}%
2311 \xP@definecrayolacolor{BlueViolet}{.86,.91,0,.04}%
2312 \xP@definecrayolacolor{Periwinkle}{.57,.55,0,0}%
2313 \xP@definecrayolacolor{CadetBlue}{.62,.57,.23,0}%
2314 \xP@definecrayolacolor{CornflowerBlue}{.65,.13,0,0}%
2315 \xP@definecrayolacolor{MidnightBlue}{.98,.13,0,.43}%
2316 \xP@definecrayolacolor{NavyBlue}{.94,.54,0,0}%
2317 \xP@definecrayolacolor{RoyalBlue}{1,.5,0,0}%
2318 \xP@definecrayolacolor{Blue}{1,1,0,0}%
2319 \xP@definecrayolacolor{Cerulean}{.94,.11,0,0}%
2320 \xP@definecrayolacolor{Cyan}{1,0,0,0}%
2321 \xP@definecrayolacolor{ProcessBlue}{.96,0,0,0}%
2322 \xP@definecrayolacolor{SkyBlue}{.62,0,.12,0}%
2323 \xP@definecrayolacolor{Turquoise}{.85,0,.2,0}%
2324 \xP@definecrayolacolor{TealBlue}{.86,0,.34,.02}%
2325 \xP@definecrayolacolor{Aquamarine}{.82,0,.3,0}%
2326 \xP@definecrayolacolor{BlueGreen}{.85,0,.33,0}%
2327 \xP@definecrayolacolor{Emerald}{1,0,.5,0}%
2328 \xP@definecrayolacolor{JungleGreen}{.99,0,.52,0}%
2329 \xP@definecrayolacolor{SeaGreen}{.69,0,.5,0}%
2330 \xP@definecrayolacolor{Green}{1,0,1,0}%
2331 \xP@definecrayolacolor{ForestGreen}{.91,0,.88,.12}%
2332 \xP@definecrayolacolor{PineGreen}{.92,0,.59,.25}%
2333 \xP@definecrayolacolor{LimeGreen}{.5,0,1,0}%
2334 \xP@definecrayolacolor{YellowGreen}{.44,0,.74,0}%
2335 \xP@definecrayolacolor{SpringGreen}{.26,0,.76,0}%
2336 \xP@definecrayolacolor{OliveGreen}{.64,0,.95,.4}%
2337 \xP@definecrayolacolor{RawSienna}{0,.72,1,.45}%
2338 \xP@definecrayolacolor{Sepia}{0,.83,1,.7}%
2339 \xP@definecrayolacolor{Brown}{0,.81,1,.6}%
2340 \xP@definecrayolacolor{Tan}{.14,.42,.56,0}%
2341 \xP@definecrayolacolor{Gray}{0,0,0,.5}%
2342 \xP@definecrayolacolor{Black}{0,0,0,1}%
2343 \xP@definecrayolacolor{White}{0,0,0,0}%
2344 }
2345 \xywithoption{crayon}{%

```

```

2346 \xP@installCrayolaColors
2347 \renewcommand*\installCrayolaColors@{}%
2348 }

```

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

```

2349 \xyendinput
2350 \color

```

8.15 Frames

```

2351 \*frame)
2352 \xyrequire{curve}%
2353 \xycatcodes

```

\xypdf-fr@loaded

```

2354 \expandafter\let\csname xypdf-fr@loaded\endcsname\@empty

```

\xP@framedrop

```

2355 \newcommand*\xP@framedrop[1]{%
2356 \expandafter\frmDrop\expandafter{%
2357 \expandafter\def\expandafter\prevEdge@@\expandafter{\prevEdge@@}%
2358 #1\frmradius@@}%
2359 }

```

\frm{-}

```

\xP@frm{-} 2360 \xP@hook{frame}{\frm{-}}
2361 \expandafter\newcommand\expandafter*\csname xP@frm{-}\endcsname{%
2362 \xP@framedrop\xP@solidframe
2363 }

```

\xP@solidframe

```

2364 \newcommand*\xP@solidframe[1]{%
2365 \R@#1\relax
2366 \xP@setsolidpat
2367 \let\xP@fillorstroke\xP@stroke
2368 \xP@frameifnotzero\xP@oval
2369 }

```

\xP@frameifnotzero

```

2370 \newcommand*\xP@frameifnotzero[1]{%
2371 \setboxz@h{%
2372 \hskip\X@c\raise\Y@c\hbox{%
2373 \DN@{\zeroEdge}%
2374 \ifx\next@\prevEdge@@
2375 \else
2376 #1%
2377 \fi
2378 }%
2379 }%
2380 \wd\z@\z@\ht\z@\z@\dp\z@\z@
2381 \boxz@
2382 }

```

\xP@oval

```

2383 \newcommand*\xP@oval{%
2384 \hskip-\L@c

```

```

2385 \lower\D@c\hbox{%
2386 \dimen@\dimexpr\L@c+\R@c\relax
2387 \dimen@ii\dimexpr\U@c+\D@c\relax
2388 \R@\xP@min\R@{.5\dimen@}%
2389 \R@\xP@min\R@{.5\dimen@ii}%

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

2390 \@tempdimb\dimexpr2\dimen@+2\dimen@ii-\R@*696621973/405764219\relax
2391 \ifdim\R@=\z@

Sharp corners: draw a rectangle.

2392 \xP@fillorstroke{0 0 \xP@coor\dimen@\dimen@ii re}%
2393 \else

Rounded corners.

2394 \def\@tempa{*119763188/267309217}%
2395 \xP@fillorstroke{%
2396 \xP@dim\R@0 m \xP@dim{\R@\@tempa}0 0 \xP@dim{\R@\@tempa}%
2397 0 \xP@dim\R@c %
2398 \ifdim2\R@=\dimen@ii\else
2399 0 \xP@dim{\dimen@ii-\R@}1 %
2400 \fi
2401 0 \xP@dim{\dimen@ii-\R@\@tempa}\xP@coor{\R@\@tempa}\dimen@ii
2402 \xP@coor\R@\dimen@ii c %
2403 \ifdim2\R@=\dimen@\else
2404 \xP@coor{\dimen@-\R@}\dimen@ii l %
2405 \fi
2406 \xP@coor{\dimen@-\R@\@tempa}\dimen@ii
2407 \xP@coor\dimen@{\dimen@ii-\R@\@tempa}\xP@coor\dimen@{\dimen@ii-\R@} c %
2408 \ifdim2\R@=\dimen@ii\else
2409 \xP@coor\dimen@\R@ l %
2410 \fi
2411 \xP@coor\dimen@{\R@\@tempa}\xP@dim{\dimen@-\R@\@tempa}0 %
2412 \xP@dim{\dimen@-\R@}0 c h%
2413 }%
2414 \fi
2415 }%
2416 }

\frm[o]{-}
\xP@frm[o]{-} 2417 \xP@hook{frame}{\frm[o]{-}}
2418 \expandafter\newcommand\expandafter*\csname xP@frm[o]{-}\endcsname{%
2419 \xP@framedrop{\xP@ellipseframe\xP@setsolidpat}%
2420 }

\xP@ellipseframe
2421 \newcommand*\xP@ellipseframe[2]{%
2422 \xP@getradii{#2}%
2423 \DN@{\zeroEdge}%
2424 \ifx\next@\prevEdge@@
2425 \else
2426 \def\xP@fillorstroke{#1\xP@stroke}%
2427 \setboxz@h{\hskip\X@c\raise\Y@c\hbox{\xP@framedellipse}}%
2428 \wd\z@=\z@ht\z@=\z@dp\z@=\z@
2429 \boxz@
2430 \fi
2431 }

```

```

\frm{.}
\XP@frm{.} 2432 \XP@hook{frame}{frm{.}}
2433 \expandafter\newcommand\expandafter*\csname XP@frm{.}\endcsname{%
2434 \XP@framedrop\XP@rectframedotted
2435 }

\XP@rectframedotted
2436 \newcommand*\XP@rectframedotted[1]{%
2437 \R@#1\relax
2438 \XP@frameifnotzero{%
2439 \ifdim\R@=\z@
2440 \XP@dottedrect
2441 \else
2442 \XP@dottedoval
2443 \fi
2444 }%
2445 }

\XP@dottedrect Make sure that there is a dot in every corner of the rectangle.
2446 \newcommand*\XP@dottedrect{%
2447 \hskip-\L@c
2448 \lower\D@c\hbox{%
2449 \dimen@ii\dimexpr\U@c+\D@c\relax
2450 \@tempdimc\dimexpr\XP@preclw/-2\relax
2451 \@tempdimb\dimexpr\L@c+\R@c+\XP@preclw\relax
2452 \XP@contfalse
2453 \XP@setdottedpat
2454 \dimen@\dimexpr\@tempdimb+\@tempdima/2+\@tempdimc\relax
2455 \XP@stroke{\XP@dim\@tempdimc0 m \XP@dim\dimen@0 l %
2456 \XP@coor\@tempdimc\dimen@ii m \XP@coor\dimen@\dimen@ii l}%
2457 \let\XP@testcont\XP@alwayscontrue
2458 \@tempdimb\dimen@ii
2459 \XP@setdottedpat
2460 \dimen@\dimexpr\L@c+\R@c\relax
2461 \advance\dimen@ii\dimexpr\@tempdimc-\@tempdima/2\relax
2462 \multiply\@tempdimc\m@ne
2463 \draw the horizontal lines  $\frac{1}{2}$  whitespace longer to eliminate inaccuracies.
2464 \XP@stroke{0 \XP@dim\@tempdimc m 0 \XP@dim\dimen@ii l %
2465 \XP@coor\dimen@\@tempdimc m \XP@coor\dimen@\dimen@ii l}%
2466 }%
2467 }

\XP@dottedoval
2467 \newcommand*\XP@dottedoval{%
2468 \def\XP@fillorstroke{\XP@setcldottedpat\XP@stroke}%
2469 \XP@oval
2470 }

\frm[o]{.}
\XP@frm[o]{.} 2471 \XP@hook{frame}{frm[o]{.}}
2472 \expandafter\newcommand\expandafter*\csname XP@frm[o]{.}\endcsname{%
2473 \XP@framedrop{\XP@ellipseframe\XP@setcldottedpat}%
2474 }

```



```

\frm{-}
\XP@frm{-} 2475 \XP@hook{frame}{frm{--}}
2476 \expandafter\newcommand\expandafter*\csname XP@frm{--}\endcsname{%
2477 \XP@framedrop\XP@rectframedashed
2478 }

```

```

\XP@rectframedashed
2479 \newcommand*\XP@rectframedashed[1]{%
2480 \R@#1\relax
2481 \XP@frameifnotzero{%
2482 \ifdim\R@=\z@
2483 \XP@dashedrect
2484 \else
2485 \XP@dashedoval
2486 \fi
2487 }%
2488 }

```

```

\XP@dashedrect
2489 \newcommand*\XP@dashedrect{%
2490 \hskip-\L@c
2491 \lower\D@c\hbox{%
2492 \dimen@\dimexpr\L@c+\R@c\relax
2493 \dimen@ii\dimexpr\U@c+\D@c\relax
2494 \@tempdimb\dimen@
2495 \XP@contfalse
2496 \XP@setdashpat
2497 \XP@stroke{0 0 m \XP@dim\dimen@0 1 %
2498 0 \XP@dim\dimen@ii m \XP@coor\dimen@\dimen@ii l}%
2499 \@tempdimb\dimen@ii
2500 \XP@setdashpat
2501 \XP@stroke{0 0 m 0 \XP@dim\dimen@ii 1 %
2502 \XP@dim\dimen@0 m \XP@coor\dimen@\dimen@ii l}%
2503 }%
2504 }

```

```

\XP@dashedoval
2505 \newcommand*\XP@dashedoval{%
2506 \def\XP@fillorstroke{\XP@setdashpat\XP@stroke}%
2507 \XP@oval
2508 }

```

```

\frm[o]{-}
\XP@frm[o]{-} 2509 \XP@hook{frame}{frm[o]{--}}
2510 \expandafter\newcommand\expandafter*\csname XP@frm[o]{--}\endcsname{%
2511 \XP@framedrop{\XP@ellipseframe\XP@setdashpat}%
2512 }

```

```

\frm{,}
\XP@frm{,} 2513 \XP@hook{frame}{frm{,}}
2514 \expandafter\newcommand\expandafter*\csname XP@frm{,}\endcsname{%
2515 \XP@framedrop\XP@frameshadow
2516 }

```

```

\XP@frameshadow

```

```

2517 \newcommand*\xP@frameshadow[1]{%
2518   \R@#1\relax
2519   \ifdim\R@=\z@\R@1.2pt\relax\fi
2520   \xP@frameifnotzero\xP@shadow
2521 }

\xP@shadow
2522 \newcommand*\xP@shadow{%
2523   \hskip\dimexpr\R@c+\R@/2\relax
2524   \lower\dimexpr\D@c+\R@/2\relax\hbox{%
2525     \def\xP@pattern{0 J 0 j []0 d}%
2526     \edef\xP@lw{\xP@dim\R@}%
2527     \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}%
2528   }%
2529 }

\frm{o-}
\xP@frm{o-} 2530 \xP@hook{frame}{\frm{o-}}
2531 \expandafter\newcommand\expandafter*\csname xP@frm{o-}\endcsname{%
2532   \xP@framedrop\xP@roundedrectframe
2533 }

\xP@roundedrectframe
2534 \newcommand*\xP@roundedrectframe[1]{%
2535   \R@#1\relax
2536   \ifdim\R@=\z@\R@\xydashl@ \relax\fi
2537   \xP@frameifnotzero\xP@roundedrectangle
2538 }

\xP@roundedrectangle
2539 \newcommand*\xP@roundedrectangle{%
2540   \dimen@ \dimexpr\L@c+\R@c\relax
2541   \dimen@ii \dimexpr\U@c+\D@c\relax
2542   \R@\xP@min\R@{.5\dimen@}%
2543   \R@\xP@min\R@{.5\dimen@ii}%
2544   \hskip-\L@c
2545   \lower\D@c\hbox{%

Rounded corners
2546     \@tempdimb \dimexpr2\dimen@+2\dimen@ii-\R@*696621973/405764219\relax
2547     \def\@tempa{*119763188/267309217}%
2548     \xP@setsolidpat
2549     \xP@stroke{%
2550       \xP@dim\R@0 m \xP@dim{\R@\@tempa}0 0 \xP@dim{\R@\@tempa}%
2551       0 \xP@dim\R@c c %
2552       \ifdim2\R@=\dimen@ii\else
2553         0 \xP@dim{\dimen@ii-\R@}m %
2554       \fi
2555       0 \xP@dim{\dimen@ii-\R@\@tempa}\xP@coor{\R@\@tempa}\dimen@ii
2556       \xP@coor\R@\dimen@ii c %
2557       \ifdim2\R@=\dimen@ \else
2558         \xP@coor{\dimen@-\R@}\dimen@ii m %
2559       \fi
2560       \xP@coor{\dimen@-\R@\@tempa}\dimen@ii
2561       \xP@coor\dimen@\dimen@ii-\R@\@tempa\xP@coor\dimen@\dimen@ii-\R@ c %
2562       \ifdim2\R@=\dimen@ii\else

```

```

2563      \xP@coor\dimen@ \R@ m %
2564      \fi
2565      \xP@coor\dimen@{\R@ \tempa}\xP@dim{\dimen@-\R@ \tempa}0 %
2566      \xP@dim{\dimen@-\R@}0 c%
2567      }%

Upper and lower horizontal dashes.

2568      \xP@contfalse
2569      \@tempdimb\dimexpr\L@c+\R@c-2\R@\relax
2570      \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
2571      \xP@setdashpat
2572      \ifdim\@tempdima>\z@
2573          \dimen@\dimexpr\@tempdimb+\R@-\@tempdima/2\relax
2574          \dimen@ii\dimexpr\U@c+\D@c\relax
2575          \xP@stroke{%
2576              \xP@dim{\R@+\@tempdima}0 m \xP@dim\dimen@ 0 1 %
2577              \xP@coor{\R@+\@tempdima}\dimen@ii m \xP@coor\dimen@\dimen@ii 1%
2578          }%
2579      \fi

Left and right vertical dashes.

2580      \@tempdimb\dimexpr\U@c+\D@c-2\R@\relax
2581      \ifdim\@tempdimb<\z@\@tempdimb\z@\fi
2582      \xP@setdashpat
2583      \ifdim\@tempdima>\z@
2584          \dimen@\dimexpr\L@c+\R@c\relax
2585          \dimen@ii\dimexpr\@tempdimb+\R@-\@tempdima/2\relax
2586          \xP@stroke{%
2587              0 \xP@dim{\R@+\@tempdima}m 0 \xP@dim\dimen@ii 1 %
2588              \xP@coor\dimen@{\R@+\@tempdima}m \xP@coor\dimen@\dimen@ii 1%
2589          }%
2590      \fi
2591      }%
2592  }

\frm{=}
\xP@frm{=} 2593 \xP@hook{frame}{frm{=}}
2594 \expandafter\newcommand\expandafter*\csname xP@frm{=}\endcsname{%
2595     \xP@framedrop\xP@dsframe
2596 }

\xP@dsframe
2597 \newcommand*\xP@dsframe[1]{%
2598     \R@#1\relax
2599     \xP@frameifnotzero\xP@dsoval
2600 }

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

```

```

2610 \xP@oval
2611 \hskip\L@c
2612 \advance\L@c-\xydashh@
2613 \advance\R@c-\xydashh@
2614 \advance\U@c-\xydashh@
2615 \advance\D@c-\xydashh@
2616 \advance\R@c-\xydashh@
2617 \ifdim\R@c<\z@\R@\z@\fi
2618 \xP@oval
2619 }

\frm[o]{=}
\xP@frm[o]{=} 2620 \xP@hook{frame}{frm[o]{=}}
2621 \expandafter\newcommand\expandafter*\csname xP@frm[o]{=}\endcsname{%
2622 \xP@framedrop\xP@dsellframe
2623 }

\frm{ee}
\xP@frm{ee} 2624 \xP@hook{frame}{frm{ee}}
2625 \expandafter\newcommand\expandafter*\csname xP@frm{ee}\endcsname{%
2626 \xP@framedrop\xP@dsellframe
2627 }

\xP@dsellframe
2628 \newcommand*\xP@dsellframe[1]{%
2629 \xP@getradii{#1}%
2630 \xP@frameifnotzero\xP@dsellipse
2631 }

\xP@temppath
2632 \@ifdefinable\xP@temppath\relax

\xP@dsellipse
2633 \newcommand*\xP@dsellipse{%
2634 \hskip\dimexpr(\R@c-\L@c)/2\relax
2635 \lower\dimexpr(\D@c-\U@c)/2\relax
2636 \hbox{%
2637 % Inner ellipse: true ellipse
2638 \advance\A@-\xydashh@
2639 \advance\B@-\xydashh@
2640 \ifdim\A@<\z@\A@\z@\fi
2641 \ifdim\B@<\z@\B@\z@\fi
2642 \def\xP@fillorstroke{\edef\xP@temppath}%
2643 \xP@ellipse@
2644 % Outer curve: offset ellipse
2645 \xP@inibigdim
2646 \let\@tempa\xydashh@
2647 \xP@offsetellipse
2648 \xP@setsolidpat
2649 \xP@stroke{\xP@temppath\space\the\@temptokena}%
2650 }%
2651 }

\xP@offsetellipse TeX grouping: Not necessary, it's in an \hbox anyway.
2652 \newcommand*\xP@offsetellipse{%
2653 \xP@movetotru

```

```

2654 \@temptokena{%
2655 \xP@offsetelliptseg\A@z@A@{\B@*173517671/654249180}%
2656 {\A@*554561898/619869377}{\B@*34221476/65864945}%
2657 {\A@*543339720/768398401}{\B@*543339720/768398401}%
2658 \xP@offsetelliptseg{\A@*543339720/768398401}{\B@*543339720/768398401}%
2659 {\A@*34221476/65864945}{\B@*554561898/619869377}%
2660 {\A@*173517671/654249180}\B@z@B@
2661 \xP@mirrorpath
2662 }

```

\xP@mirrorpath

```

2663 \newcommand*\xP@mirrorpath{%
2664 \edef\@tempa{\the\@temptokena\relax\space\space\space\space}%
2665 \let\@tempb\empty
2666 \let\@tempc\empty
2667 \expandafter\xP@mirrorpath@\@tempa
2668 }

```

\xP@mirrorpath@

```

2669 \@ifdefinable\xP@mirrorpath@\relax
2670 \def\xP@mirrorpath@#1 #2 #3 #4 #5 #6 #7 {%
2671 \ifx\relax#4%
2672 \xP@append\@temptokena{\@tempb\xP@minus#1 \xP@minus#2 #3 \@tempc h}%
2673 \else
2674 \edef\@tempb{\xP@minus#6 #7 \xP@minus#4 #5 \xP@minus#1 #2 c \@tempb%
2675 \if#3m\else\xP@minus#1 \xP@minus#2 #3 \fi%
2676 \xP@minus#4 \xP@minus#5 \xP@minus#6 \xP@minus#7 }%
2677 \edef\@tempc{#6 \xP@minus#7 #4 \xP@minus#5 #1 \xP@minus#2 c \@tempc}%
2678 \expandafter\xP@mirrorpath@
2679 \fi
2680 }

```

\xP@minus

```

2681 \@ifdefinable\xP@minus\relax
2682 \def\xP@minus#1 {\if-#1 \else\ifdim\dimexpr#1pt\relax=z@\else-\fi#1 \fi}

```

\xP@insertbefore

```

2683 \newcommand*\xP@insertbefore[2]{\edef\@tempa{#1{#2\the#1}}\expandafter}\@tempa}

```

\xP@offsetelliptseg

```

2684 \newcommand*\xP@offsetelliptseg[8]{%
2685 \X@p\dimexpr#1\relax
2686 \Y@p\dimexpr#2\relax
2687 \L@c\dimexpr#3\relax
2688 \U@c\dimexpr#4\relax
2689 \R@c\dimexpr#5\relax
2690 \D@c\dimexpr#6\relax
2691 \X@c\dimexpr#7\relax
2692 \Y@c\dimexpr#8\relax
2693 \xP@savepts
2694 \xP@a\z@
2695 \xP@c\xP@bigdim
2696 \xP@paintsolid@
2697 }

```

```

\xP@getradii
2698 \newcommand*\xP@getradii[1]{%
2699 \edef\@tempa{#1}%
2700 \expandafter\xP@getradii@\@tempa,\maxdimen,%
2701 }

\xP@getradii@
2702 \@ifdefinable\xP@getradii@\relax
2703 \def\xP@getradii@#1,#2,#3{%
2704 \A@#1\relax
2705 \B@#2\relax
2706 \ifdim\B@=\maxdimen
2707 \A@\dimexpr(\L@c+\R@c)/2\relax
2708 \B@\dimexpr(\U@c+\D@c)/2\relax
2709 \fi
2710 }

\frm{o}
\xP@frm{o} 2711 \xP@hook{frame}{frm{o}}
2712 \expandafter\newcommand\expandafter*\csname xP@frm{o}\endcsname{%
2713 \xP@framedrop{\xP@circleframe\xP@setsolidpat}%
2714 }

\frm{-o}
\xP@frm{-o} 2715 \xP@hook{frame}{frm{-o}}
2716 \expandafter\newcommand\expandafter*\csname xP@frm{-o}\endcsname{%
2717 \xP@framedrop{\xP@circleframe\xP@setcldashpat}%
2718 }

\frm{.o}
\xP@frm{.o} 2719 \xP@hook{frame}{frm{.o}}
2720 \expandafter\newcommand\expandafter*\csname xP@frm{.o}\endcsname{%
2721 \xP@framedrop{\xP@circleframe\xP@setcldottedpat}%
2722 }

\xP@circleframe
2723 \newcommand*\xP@circleframe[2]{%
2724 \R@#2\relax
2725 \def\xP@fillorstroke{#1\xP@stroke}%
2726 \DN@{\zeroEdge}%
2727 \ifx\next@\prevEdge@@
2728 \ifdim\R@>\z@
2729 \xP@circleframe@
2730 \fi
2731 \else
2732 \ifdim\R@=\z@
2733 \A@\dimexpr(\L@c+\R@c)/2\relax
2734 \B@\dimexpr(\U@c+\D@c)/2\relax
2735 \R@\xP@max\A@\B@
2736 \fi
2737 \xP@circleframe@
2738 \fi
2739 }

\xP@circleframe@

```

```

2740 \newcommand*\xP@circleframe@{%
2741   \setboxz@h{\hskip\X@c\raise\Y@c\hbox{\xP@circle}}}%
2742   \wd\z@\z@\ht\z@\z@\dp\z@\z@
2743   \boxz@
2744 }

\frm{e}
\xP@frm{e} 2745 \xP@hook{frame}{frm{e}}
2746 \expandafter\newcommand\expandafter*\csname xP@frm{e}\endcsname{%
2747   \xP@framedrop{\xP@ellipseframe\xP@setsolidpat}%
2748 }

\frm{-e}
\xP@frm{-e} 2749 \xP@hook{frame}{frm{-e}}
2750 \expandafter\newcommand\expandafter*\csname xP@frm{-e}\endcsname{%
2751   \xP@framedrop{\xP@ellipseframe\xP@setcldashpat}%
2752 }

\frm{.e}
\xP@frm{.e} 2753 \xP@hook{frame}{frm{.e}}
2754 \expandafter\newcommand\expandafter*\csname xP@frm{.e}\endcsname{%
2755   \xP@framedrop{\xP@ellipseframe\xP@setcldottedpat}%
2756 }

\frm2{.e}
\xP@frm2{.e} 2757 \xP@hook{frame}{frm2{.e}}
2758 \expandafter\newcommand\expandafter*\csname xP@frm2{.e}\endcsname{%
2759   \xP@framedrop\xP@dottedellframe
2760 }

\xP@dottedellframe
2761 \newcommand*\xP@dottedellframe[1]{%
2762   \xP@getradii{#1}%
2763   \xP@frameifnotzero\xP@dottedellipse
2764 }

\xP@dottedellipse
2765 \newcommand*\xP@dottedellipse{%
2766   \hskip\dimexpr(\R@c-\L@c)/2\relax
2767   \lower\dimexpr(\D@c-\U@c)/2\relax
2768   \hbox{%
Intermediate ellipse: true ellipse
2769     \@tempdima.5\xydashh@\relax
2770     \advance\A@-\@tempdima
2771     \advance\B@-\@tempdima
2772     \ifdim\A@<\@tempdima\A@\@tempdima\fi
2773     \ifdim\B@<\@tempdima\B@\@tempdima\fi
2774     \let\xP@normalmult\@ne
2775     \xP@specialellipse{\xP@splinemultdotted\xP@doublestroke}%
2776   }%
2777 }

\xP@specialellipse
2778 \newcommand*\xP@specialellipse[1]{%
2779   \def\@tempa{*147546029/267309217}%

```

```

2780 \X@p\A@
2781 \Y@p\z@
2782 \L@c\A@
2783 \U@c\dimexpr\B@\@tempa\relax
2784 \R@c\dimexpr\A@\@tempa\relax
2785 \D@c\B@
2786 \X@c\z@
2787 \Y@c\B@
2788 \xP@bezierlength
2789 \let\xP@testcont\xP@alwaysconttrue
2790 #1%
2791 \X@p\z@
2792 \Y@p\B@
2793 \L@c-\R@c
2794 \D@c\U@c
2795 \U@c\B@
2796 \R@c-\A@
2797 \X@c-\A@
2798 \Y@c\z@
2799 #1%
2800 \X@p-\A@
2801 \Y@p\z@
2802 \R@c\L@c
2803 \L@c-\A@
2804 \U@c-\D@c
2805 \D@c-\B@
2806 \X@c\z@
2807 \Y@c-\B@
2808 #1%
2809 \X@p\z@
2810 \Y@p-\B@
2811 \L@c-\R@c
2812 \D@c\U@c
2813 \U@c-\B@
2814 \R@c\A@
2815 \X@c\A@
2816 \Y@c\z@
2817 #1%
2818 }

```

\xP@alwaysconttrue

```

2819 \newcommand*\xP@alwaysconttrue[1]{\xP@splineconttrue}

```

\frm2{-e}

\xP@frm2{-e}

```

2820 \xP@hook{frame}{frm2{-e}}
2821 \expandafter\newcommand\expandafter*\csname xP@frm2{-e}\endcsname{%
2822 \xP@framedrop\xP@dsdashedframe
2823 }

```

\xP@dsdashedframe

```

2824 \newcommand*\xP@dsdashedframe[1]{%
2825 \xP@getradii{#1}%
2826 \xP@frameifnotzero\xP@dsdashedellipse
2827 }

```

\xP@dsdashedellipse


```

2828 \newcommand*\xP@dsdashedellipse{%
2829   \hskip\dimexpr(\R@c-\L@c)/2\relax
2830   \lower\dimexpr(\D@c-\U@c)/2\relax
2831   \hbox{%
      Inner ellipse: true ellipse
2832     \advance\A@-\xydashh@
2833     \advance\B@-\xydashh@
2834     \ifdim\A@<\z@\A@\z@\fi
2835     \ifdim\B@<\z@\B@\z@\fi
2836     \xP@specialellipse{\xP@splinemultdashed\xP@elldoublestroke}%
2837   }%
2838 }

\xP@elldoublestroke
2839 \newcommand*\xP@elldoublestroke{\z@,\xydashh@}

\xP@fill
2840 \newcommand*\xP@fill[1]{\xP@literal{#1 f}}

\xP@fillstroke
2841 \newcommand*\xP@fillstroke[1]{\xP@literal{\xP@dim{\xP@preclw/2}w 1 j 0 G #1 b}}

\xP@fillorstroke
2842 \newcommand*\xP@fillorstroke{}

  \frm{*}
\xP@frm{*} 2843 \xP@hook{frame}{frm{*}}
2844 \expandafter\newcommand\expandafter*\csname xP@frm{*}\endcsname{%
2845   \xP@framedrop{\let\xP@fillorstroke\xP@fill\xP@framefill}%
2846 }

  \frm{**}
\xP@frm{**} 2847 \xP@hook{frame}{frm{**}}
2848 \expandafter\newcommand\expandafter*\csname xP@frm{**}\endcsname{%
2849   \xP@framedrop{\let\xP@fillorstroke\xP@fillstroke\xP@framefill}%
2850 }

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

```

```

2868         \ifdim\R@=\z@
2869         \DN@{\xP@oval}%
2870         \else
2871         \DN@{\xP@circle}%
2872         \fi
2873     \fi
2874 \fi
2875     \next@
2876 }%
2877 }%
2878 \wd\z@\z@\ht\z@\z@\dp\z@\z@
2879 \boxz@
2880 }

\xP@circle
2881 \newcommand*\xP@circle{%
2882 \xP@ellipse\R@\R@
2883 }

\xP@filledellipse
2884 \newcommand*\xP@filledellipse{%
2885 \xP@ellipse{\dimexpr(\L@+\R@)/2\relax}{\dimexpr(\U@+\D@)/2\relax}%
2886 }

\xP@framedellipse
2887 \newcommand*\xP@framedellipse{%
2888 \xP@ellipse\A@\B@
2889 }

\xP@ellipse
2890 \newcommand*\xP@ellipse[2]{%
2891 \hskip\dimexpr(\R@-\L@)/2\relax
2892 \lower\dimexpr(\D@-\U@)/2\relax
2893 \hbox{%
2894 \A@#1\relax
2895 \B@#2\relax
2896 \xP@ellipse@
2897 }%
2898 }

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

```

Perimeter, first segment

```

2910 \X@p\A@
2911 \Y@p\z@
2912 \L@c\A@
2913 \U@c\dimexpr\B@*173517671/654249180\relax
2914 \R@c\dimexpr\A@*554561898/619869377\relax
2915 \D@c\dimexpr\B@*34221476/65864945\relax
2916 \X@c\dimexpr\A@*543339720/768398401\relax
2917 \Y@c\dimexpr\B@*543339720/768398401\relax
2918 \xP@bezierlength
2919 \@tempdimb4\dimexpr\@tempdima+\@tempdimb\relax
2920 \edef\@tempa{%
2921   \xP@dim\A@0 m
2922   \xP@coor\L@c\U@c
2923   \xP@coor\R@c\D@c
2924   \xP@coor\X@c\Y@c c %
2925   \xP@coor{\A@*34221476/65864945}\B@*554561898/619869377}%
2926   \xP@coor{\A@*173517671/654249180}\B@
2927   0 \xP@dim\B@ c }%
2928 \@temptokena\expandafter{\@tempa}%
2929 \xP@mirrorpath
2930 \xP@fillorstroke{\the\@temptokena}%
2931 }

```

End of the section for Xy-pic’s “frame” option.

```

2932 \xyendinput
2933 </frame>

```

8.16 Line styles

```

2934 <*line>
2935 \xycatcodes
2936 \expandafter\let\csname xypdf-li@loaded\endcsname\@empty
2937 % Dummy file.

```

End of the section for Xy-pic’s “line” option.

```

2938 \xyendinput
2939 </line>
2940 <*basic>

```

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

```

2941 \let\@tempa\@undefined
2942 \let\nextii\@undefined
2943 \DN@{%
2944   \xywithoption{color}{%
2945     \message{Xy-pic pdf driver: ‘color’ extension support}%
2946     \@ifundefined{xypdf-co@loaded}{\input xypdf-co\relax}{\message{not reloaded}}%
2947   }%
2948   \xywithoption{curve}{%
2949     \message{Xy-pic pdf driver: ‘curve’ extension support}%
2950     \@ifundefined{xypdf-cu@loaded}{\input xypdf-cu\relax}{\message{not reloaded}}%
2951   }%
2952   \xywithoption{frame}{%
2953     \message{Xy-pic pdf driver: ‘frame’ extension support}%
2954     \@ifundefined{xypdf-fr@loaded}{\input xypdf-fr\relax}{\message{not reloaded}}%
2955   }%

```

```

2956 \xywithoption{line}{%
2957 \message{Xy-pic pdf driver: 'line' extension support}%
2958 \@ifundefined{xy-pdf-li@loaded}{\input xy-pdf-li\relax}{\message{not reloaded}}%
2959 }%
2960 \xywithoption{rotate}{%
2961 \message{Xy-pic pdf driver: 'rotate' extension support}%
2962 \@ifundefined{xy-pdf-ro@loaded}{\input xy-pdf-ro\relax}{\message{not reloaded}}%
2963 }%
2964 }
2965 \next@
2966 \xyendinput
2967 </basic>

```

9 Changelog

v1.0 2010/03/24

Initial version

v1.1 2010/03/30

- Added support for the Xy-pic “rotate” extension.
- The parts of the style file dealing with 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 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 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 pdfTeX since \pdfsave is not defined prior to pdfTeX 1.40.0.
- “**Troubleshooting**” paragraph for TeX Live without the ε -TeX features enabled.
- Generic PDF code for the {-} directional object.

v1.4 2010/05/13

- Support for both plain TeX and L^ATeX, reorganization of the code and splitting into several files. The L^ATeX style file xypdf.sty has been replaced by xypdf.tex, which is recognized by Xy-pic as a driver.

- Integration into the $\text{\texttt{Xy-pic}}$ distribution.
- Support for the “color” and “frame” extensions.
- New supported curve style $\{\sim\sim\}$ (broken squiggled curves).

v1.5 2010/10/11

- Bug fix: Colored curves.
- Bug fix: PDF syntax in double elliptical frames.