

L^AT_EX 'package' cubezier

Samenvatting Het L^AT_EX-pakket *cubezier* tekent met de functie `\cubezier` derdegraads Bézierlijnen, desgewenst als stippellijn, door PostScript, of, met aangepaste Béziersubintervalparameter, als tweedegraads Bézierlijn. `\cubezier` is een alternatief voor `\cbezier`, van het pakket *bez123*.

Gebruiksaanwijzing *Inlezen* van het pakket *cubezier* na de documentclass aanhef:

```
\usepackage{cubezier}
```

Aanroep van de functie `\cubezier` voor de derdegraads kromme tussen beginpunt (x_0, y_0) en eindpunt (x_1, y_1) , met controlepunt (x_{t0}, y_{t0}) op de raaklijn aan de kromme in het beginpunt en controlepunt (x_{t1}, y_{t1}) op de raaklijn in het eindpunt:

```
\cubezier (x0,y0)(xt0,yt0)(xt1,yt1)(x1,y1)
```

Optionele aanroep als stippellijn met een gegeven aantal N punten:

```
\cubezier[N](x0,y0)(xt0,yt0)(xt1,yt1)(x1,y1)
```

Parameters zijn (met hier de voorgekozen (verstek)waarden):

```
\gdef\Beziersubintervals{3}% classic Bezier
\cubezierPSfalse % no PostScript special
\gdef\qbeziermax{500}% LaTeX maximum for number of pixels
```

waarbij de tweede een conditie is en de laatste gedefinieerd in L^AT_EX.

Postscript tekenen aanzetten met

```
\cubezierPStrue % PostScript
```

Gebruiksvoorbeelden Zelfde voorbeelden als in de documentatie van het L^AT_EX-pakket *bez123* van Peter Wilson ('The bez123 and multiply packages', 2-2-2009, texmf-dist/doc/latex/bez123).

```
\begin{picture}(70,60)(-30,0)
\linethickness{0.5pt}%
\put (0,30){\cubezier [30](0,0)(30,0)(0,30)(0,0)}%
\put (40,30){\cubezier [30](0,0)(10,30)(20,0)(30,30)}%
\put (0,0){\cubezier (0,0)(30,0)(0,30)(10,10)}%
\put (40,0){\cubezier (0,0)(30,30)(10,30)(20,0)}%
\end{picture}
```



Voorbeeld gebruik subintervalparameter Het concept subintervallen behoort bij de numerieke afgeleide van de parameter kromme. De numerieke eerste afgeleide van de Bézierkromme in het beginpunt is:

$$(x_{t0}-x_0, y_{t0}-y_0)/\tau \quad 1/\tau=\text{\Beziersubintervals}$$

en evenzo voor het eindpunt. Merk op, dat 'aantal' (Beziersubintervals) niet geheel hoeft te zijn, omdat τ iedere waarde kan hebben.

Met 3 subintervallen krijgen we de 'gewone' kubische Bézier `\cbezier`, met controlepunten op $1/3$ en $2/3$, zoals we reeds zagen bij gebruiksvoorbeelden gebaseerd op de standaard subintervalwaarde 3.

Met 2 subintervallen en samenvallende controlepunten, in het snijpunt van de raaklijnen op de symmetrielijns halverwege, degenerereert de functie naar kwadratische `\qbezier`:

```
\def\Beziersubintervals{2}\cbezier (0,30)(30,30)(30,30)(30,0)
\put (40,0){\cbezier (0,0)(21.21320,21.21320)(21.21320,21.21320)(42.42641,0)}%
```



Rechtvaardiging voor de functie `\cbezier` is de mogelijkheid van 4 subintervallen. Met 4 subintervallen krijgen we een herhaalde kwadratische Bézier, die nagenoeg een vierdegraads Bézier is, waardoor de controlepunten praktisch op de raaklijn aan de kromme in het middenpunt ligt. Dit illustreren we aan de cirkel. Om een eerstekwadrantkwartcirkel, met straal 30, te verkrijgen wordt het eerste controlepunt bepaald op $y = 30 \tan(\pi/8) = 30(\sqrt{2} - 1) = 12.421641$, en het tweede vergelijkbaar:

```
\def\Beziersubintervals{4}\cbezier (30,0)(30,12.42641)(12.42641,30)(0,30)
\def\Beziersubintervals{3}\cbezier [60](30,0)(30,12.42641)(12.42641,30)(0,30)
\def\Beziersubintervals{2}\cbezier [30](30,0)(30,12.42641)(12.42641,30)(0,30)
```



Ter vergelijking zijn twee andere intervalwaarden gegeven als stippellijn. Merk op, dat het eerste controlepunt van deze kwartcirkel bij de klassieke subintervalwaarde 3 een factor $4/3$ hoger moet komen (dus op hoogte $4/3(\sqrt{2} - 1)$ bij de eenheids cirkel) voor hetzelfde resultaat.

```
\def\Beziersubintervals{4}\cbezier (30,0)(30,12.42641)(12.42641,30)(0,30)
\def\Beziersubintervals{3}\cbezier [60](30,0)(30,16.56854)(16.56854,30)(0,30)
```



PS PostScript tekent een veel mooiere lijn dan TeX, maar de 'texinstallatie' moet dit ondersteunen. In mijn geval gebruikte ik L^AT_EX uitvoer naar *.dvi, te bezien met viewer dviout die PS ondersteunt. Daarna *.dvi omzetten naar *.pdf via dvi2pdf om het PS resultaat te zien.

```
\cubezierPStrue % PostScript
\def\Beziersubintervals{4}\cbezier (30,0)(30,12.42641)(12.42641,30)(0,30)
\def\Beziersubintervals{3}\cbezier [60](30,0)(30,12.42641)(12.42641,30)(0,30)
\def\Beziersubintervals{2}\cbezier [30](30,0)(30,12.42641)(12.42641,30)(0,30)
```



Helaas tekent `cubezier` iedere Bézierlijn apart in een nieuw pad, dat niet wordt toegevoegd aan het bestaande pad. Een handige gebruiker kan de `cubezierspecial` aanpassen teneinde sluitende paden te verkrijgen die gevuld kunnen worden.