How to Generate LaTeX Picture Environments Using OpenOffice.org 3 Draw

Herbert Möller*

Abstract. The powerful free drawing software `OpenOffice.org 3 Draw` (abbreviated `OOoDraw` in the following) and a Perl filter program are used to conveniently create even complicated figures with the LaTeX `picture` environment. Parsing PostScript files exported by `OOoDraw`, the filter program `OOopict.pl` generates LaTeX code ready for use. Along with other advantages `OOoDraw` has its own PostScript translator, which does not depend on the operating systems supported by `OpenOffice.org`, namely Linux, Mac OS, Unix, and Windows. Therefore, `OOopict.pl` can be applied to PostScript files of `OOoDraw` on all these operating systems. Moreover, all curves constructed with `OOoDraw` are represented in the PostScript files as cubic Bézier curves which can be directly used by the package `pict2e` [2] to produce vector graphics with LaTeX.

1. Introduction

Three years ago a new method for the creation of figures with the LaTeX `picture` environment was published in the Comprehensive TeX Archive Network (CTAN) [3]. It is called “GaPFilL” (*Graphics as PostScript Filtered for LaTeX*) because it consists of the following four steps:

(1) Generating the figures with a drawing program or geometry software;
(2) “Printing” or exporting the construction as a PostScript file;
(3) Applying a filter program to the PostScript text;
(4) Copying the resulting code or parts of it into a LaTeX document.

Since steps (1) and (2) and the packages used in the LaTeX document may vary, it is necessary to apply different filter programs. Until now the only two filter programs `CABebez.pl` and `CABpict.pl` in the CTAN have been developed for the Macintosh version of the geometry software `Cabri-géomètre`$^{\text{TM}}$ `II`. The first one requires the macro package `ebezier` [1], the second additionally uses the new package `pict2e` [2].

The website *Mathkompass* [4] contains two filter programs `CDTex` and `CDebez.pl` for use in conjunction with the drawing software `ClarisDraw`. They were not published in the CTAN because `ClarisDraw` was abandoned when the office program `AppleWorks` was released. The first filter is an AppleScript program written in 2002 mainly for tracing already existing figures with the layer technique of `ClarisDraw`. The filter program `CDebez.pl` is based on `CABebez.pl`, but profits from the fact that `ClarisDraw` has its own PostScript translator which represents cubic Bézier curves directly usable with the package `ebezier`.

The documentation GaPDoc1.pdf in [3] contains the description of the GaPFilL method by means of the two prototypes `CABebez.pl` and `CABpict.pl`. In the next two sections we will concentrate on those features of `OOoDraw` that are useful for the generation of LaTeX figures. Then the application of the perl system with Mac OS and Windows will be described. The last section contains some explanations of the characteristics of the filter program `OOopict.pl` different from those of `CABpict.pl`.

2. The Drawing Program

`OOoDraw` is part of the open-source office software suite `OpenOffice.org 3` which can be downloaded free of charge (`http://www.openoffice.org`, 424 MB). Compared with `ClarisDraw` it is even more powerful. Here only some hints are given which are difficult to find in the online help.

Customizing Toolbars

If `OOoDraw` is mainly used for generating LaTeX figures, the entries ‘Drawing’, ‘Edit Points’, ‘Line and Filling’, ‘Options’, ‘Standard’, and ‘Color bar’ should be selected in the submenu ‘Toolbars’ of the ‘View’ menu. The index item ‘customizing; toolbars’ describes how to add or change buttons in the toolbars. The ‘Drawing’ bar should visibly contain the buttons ‘Select’, ‘Line’, ‘Square, Unfilled’, ‘Rectangle, Unfilled’, ‘Circle, Unfilled’, ‘Ellipse, Unfilled’, ‘Circle Arc’, ‘Arc’, ‘Polygon’, ‘Dimension Line’, ‘Curve’, ‘Close Bézier’, ‘Points’, and ‘Position and Size’. The filter program will not process filled objects or arrows.

Drawing Curves

‘Drawing’ is the most helpful tool because `OOoDraw` and the package `pict2e` permit to draw arbitrary curves as vector graphics with LaTeX. The help entries ‘curves; drawing’ and ‘curves; editing’ describe how to handle these cubic Bézier curves. The filter program `OOopict.pl` recognises lines and curves by the respective line endings in the PostScript file. **Therefore, it is not possible to continue a curve segment with a line.**

Since the packages `ebezier` and `pict2e` don’t offer dotted cubic Bézier curves, the filter `OOopict.pl` approximates these curves by four quadratic Bézier curves which can be dotted. `OOopict.pl` generates quadratic Bézier curves from blue or green polygons consisting of two segments. These curves which represent parabolas are not visible in the slides of `OOoDraw`. The fact that each quadratic Bézier curve is a special cubic one, may be used to produce visible parabolas (see FIGURE 1).

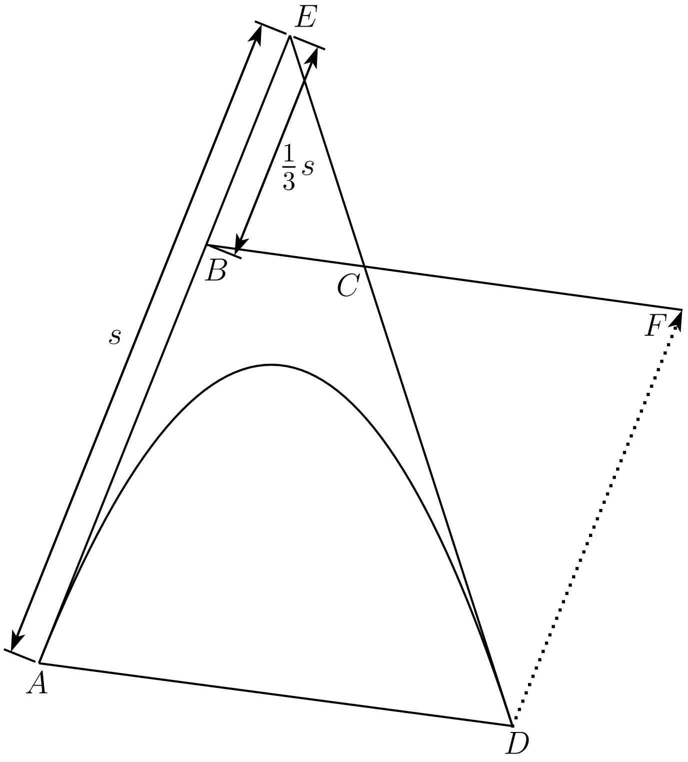

FIGURE 1. Visualization of quadratic Bézier curves

The quadratic Bézier curve generated by the polygon AED is exactly the same as the cubic Bézier curve with the control points A, B, C, and D, where $|BE| = \frac{1}{3}\,|AE|$, and $ADFB$ is a parallelogram. The length of segments can be determined with the tool ‘Dimension Line’ of `OOoDraw`.

Some useful drawing aids will be mentioned in the following explanation of how to close curves or polygons. The direct way is to double-click the starting point and to observe the number of points shown on the left-hand side of the ‘Status Bar’ at the bottom of the window. In case it proves difficult to snap an object point, you should enlarge the ‘Snap range’ in the Menu ‘Grid’ of the `OOoDraw` ‘Preferences’, or zoom in with the sliding control on the right-hand side of the Status Bar. The simplest method is presented by the tool ‘Close Bézier’ which closes curves and polygons with a line. As explained above, in the case of a curve this line must be converted to a curve segment. This can be done with the tools ‘Smooth Transition’ or ‘Symmetric Transition’ on the ‘Edit Points Bar’.

Layers and PostScript

Most drawing programs support layers for the arrangement of elements on a page. The objects on each layer can be hidden from view, excluded from printing, or locked. For the GaPFilL method the most important application is copying pictures by tracing. `OOoDraw` provides three predefined layers: Layout, Controls, and Dimension Lines. The Layout layer is the default workspace. Therefore, when opening a file, its contents is inserted on this layer. Since `OOoDraw` accepts the jpg-format, even photos taken by a digital camera may be traced.

For the traced version you have to add at least one layer by choosing ‘Insert - Layer’ or by right-clicking the layer tab area at the bottom. At first, tracing should be done with colours that are easy to distinguish from the original. Before saving, colours must be changed according to the information they have to carry described in the next section.

Since the PostScript translator takes into account the contents of all layers, even if they are invisible or locked, all but the added layers must be emptied before exporting to an eps-file via the menu ‘`File/Export...`’. In the window ‘`EPS Export Options`’ which opens when you click on ‘`Save`’, you have to select No Preview, Level 2, Color, and No Compression.

3. The Use of Colours

With the aid of colours additional information is conveyed to the filter program. `OOoDraw` offers several colour lists. In the following we use twelve of the first fourteen colours in the list ‘standard.soc’. All colours of the list are shown in the

'Color Bar'. Moving the cursor over any of the coloured squares, the name of the colour appears. Since most of the names of the same colours in the list 'html.soc' are more customary, they are also used internally in the filter program. These names are added in parentheses to the "standard names" in the table at the end of this section.

There are three methods to colour a line or curve. A colour may be preselected by clicking on the corresponding area in the menu 'Line color' of the toolbar 'Line and Filling'. The colour of a selected line or curve can be changed by right-clicking on the area of the new colour in the 'Color Bar'. Finally, dragging a colour of the 'Color Bar' to an arbitrary line or curve changes its colour if it is different.

In the following an "object" may be a line, a polygon, a circle, an arc of a circle, an ellipse, an arc of an ellipse, or a cubic Bézier curve.

- *Unbroken objects* are drawn with the default colour black.
- *Dotted objects* are obtained by using the colour light green.
- *Unbroken arrows* must be drawn as a turquoise line. A blue line generates an *arrow with a dotted shank.*
- *Unbroken quadratic Bézier curves* are entered as light blue polygons with two segments. A *dotted parabola* results from green polygons.
- *Text or formula positions* are defined by using lines coloured in light cyan. For the bounding box to be calculated correctly, the marking line should approximately be the diagonal of the smallest rectangle enclosing the text or formula beginning at the lower left corner. In the LaTeX output a serial number will be written at the position of the starting point.
- The remaining five colours are used for *filling, hatching* or *dotting* areas bordered by *polygons.* For this purpose the filter program splits the polygon area into triangles which have the starting point of the polygon as a common corner. To avoid overlapping, the given area must be devided into polygons in such a way that for each polygon the triangles arising from connecting the starting point with the other corners have no more than one side in common with the other triangles (see [3]).

  Light red *polygons* will be *filled* without boundary. For *hatching with visible boundary lines* the colour red must be chosen for the polygon; *without boundary lines* the colour brown. If the polygon is magenta-coloured, its area will be *dotted with visible boundary lines*; using the colour $\boxed{\text{light magenta}}$ an area can be *dotted without boundary lines*. The previous assignments of colours are summarized in the following table.

- All *drawing elements* coloured in $\boxed{\text{other colours}}$ may serve as *drawing aids* because they are ignored by the filter program.

Type	Colour
unbroken object	black (black)
dotted object	light green (lime)
unbroken arrow	turquoise (teal)
dotted arrow	blue (navy)
unbroken quad. Bézier curve	light blue (blue)
dotted quad. Bézier curve	green (green)
text marker	light cyan (cyan)
filled without border	light red (red)
hatched with border	red (maroon)
hatched without border	brown (olive)
dotted with border	magenta (purple)
dotted without border	light magenta (magenta)

4. APPLICATION OF THE PERL SYSTEM

To apply the filter program `OOopict.pl`, you need the Perl system and a text editor able to run a script or a command line while a text file is open in its primary window.

Mac OS X

The operating system provides the native Perl system. As a suitable source editor `TextWrangler 2.3` is available free of charge (`http://www.barebones.com/products/textwrangler`). It is a general purpose text editor developed as a by-product of the professional HTML and text editor `BBEdit`$^{\text{TM}}$ `9` of Bare Bones Software, Inc. `TextWrangler` works directly with the Perl system as a Unix scripting environment. Therefore, it is accessed via the menu '`#!`' which gets its "name" from the beginning of the first line in each Unix script.

`OOopict.pl` must be placed in the subdirectory 'Unix Filters' which is opened by the command 'Open Filters Folder' in the submenu 'Unix Filters' of the menu '`#!`'. With a `OOoDraw` PostScript file open in the primary window, you only have to select `OOopict.pl` from the submenu 'Unix Filters'. As if by magic, the PostScript text will then be replaced by the corresponding LaTeX program.

Windows and other Operating Systems

The complete, ready-to-install Perl distribution `ActivePerl` for Windows, Linux, Mac OS X, Solaris, AIX and HP-UX is available for free from ActiveState Software, Inc. (`http://www.activestate.com/activeperl`). This company also offers the free and open source editor `Komodo Edit 5.1` (`http://www.activestate.com/komodo_edit`) based on the professional `Komodo`™ `IDE`.

After `ActivePerl` is installed, `OOpict.pl` can be saved, for example, in the directory '`C:\Perl\eg\`'. Having opened an `OOoDraw` PostScript file for the first time and typed 'ctrl-A', you can select 'Run Command' in the menu 'Tools'. In the command line you will have to enter '`C:\Perl\eg\OOpict.pl`' (or the path of the directory you have chosen for `OOpict.pl`). Before running the command, you must activate the checkbox 'Pass selection as input'. If you also activate the third checkbox, the command will be stored in the toolbox from where you can run it by double-clicking its name.

5. The filter program `OOpict.pl`

The Perl filter program `OOpict.pl` has got 701 lines not including comment lines beginning with '#' and the lines containing closing braces '}'. The following line numbers don't belong to the program itself.

```
 1 #!/usr/bin/perl -w
 2 # OOpict.pl
 3 # (c) Copyright 2009 H. Moeller (mollerh@math.uni-muenster.de).
 4 # Version 1.3 for OpenOffice.org 3 Draw with Perl 5.8,
     and with the LaTeX-package 'pict2e' from 2008
     or with 'pict2e' (2003) and 'ebezier'.
 ...
14 # If your package pict2e is from 2008 or later (else $pictnew = 0):
15 $pictnew = 1;
```

Since there are two versions of `pict2e`, the flag `$pictnew` of line 15 is used to switch from the environment of `CABpict.pl` to the new one which doesn't need the package `ebezier`, because the 2008 version of `pict2e` can handle dotted quadratic Bézier curves. Furthermore, the absolute value of the bound of the slope components is increased from 1000 to 16383 which, in `OOpict.pl`, is also incorporated into the flag `$pictnew`.

Usually the file `pict2e.sty` is located in a subdirectory ‘`.../tex/latex/`’. If you don’t find it in order to look for the version date, you can run a test with a LaTeX program containing `\usepackage[pdftex]{pict2e}` in the preamble and a line

```
\put(0,0){\line(1001,1000){10}}
```

in the document. If you have the old version you will get an error message.

If you know where the file `pict2e.sty` is located, you can replace it downloading the directory `pict2e` from [2], running (or opening) the file `pict2e.dtx` (or `pict2e.ins`) with LaTeX , and moving the resulting `.sty` file to the desired location.

```
46 @lines = <>;
47 do {
48   $_ = $lines[$i++];
49   if ((/ l  /o) or (/ ct /o) or (/ p[cs ][\se]/o)) {
50     s/\d+ lw \d+ lj //go;
51     s/ m //go;
52     if (/ c /o) {
53       s/(\d\.\d+) (\d\.\d+) (\d\.\d+) c //o;
54       $c=$1.$2.$3." ";
55     }
56     $t="";
57     $l="";
58     $tflag=0;
59     $fflag=0;
60     while ($_ !~ (/p[cs ][\se]/o)) {
 ...
88     if ($tflag and $fflag) {
89       $curve[++$#curve] = $c.$t;
90     }
91     elsif ($fflag) {
92       $polygon[++$#polygon] = $c.$l;
93     }
94   }
95 }
96 until $i == $#lines;
```

There are three difficulties in filtering the text lines for LaTeX. i) If successive elements (lines, polygons, curves) have the same colour, only the first one carries the colour data (lines 52 to 54). ii) Long data lines are broken by the PostScript translator before getting to the characterizing ending (‘`ps`’, ‘`pc`’ or ‘`ef`’). Therefore, the successive lines have to be assembled in the strings ‘`$l`’ and ‘`$t`’ according to the data separators, namely '`l`' ('`lineto`') for lines and '`ct`' ('`curveto`') for curves. iii) Whether an object is to be filled, can only be recognized by the ending '`ef`'. Here, the flag '`$fflag`' is used to decide which strings have to be saved in the arrays '`@curve`' or '`@polygon`' (lines 88 to 92).

Since `OOopict.pl` only needs two arrays for collecting the data strings (instead of five in `CABpict.pl`), there are at most 130 lines for calling the appropriate subprocedures (less than half as many as in `CABpict.pl`). About 120 lines and the subprocedure '`arc`' could be eliminated, because the PostScript translator of `OOoDraw` represents all curves as cubic Bézier curves.

Some of the remaining eight subprocedures differ from those in `CABpict.pl`. The subprocedure '`best`' has been corrected, because in the earlier version the fact that the greatest integer of Perl is given by $2^{31} - 1$ was overlooked. Furthermore, an entry for the upper bound of the approximating numerator and denominator has been introduced.

In the subprocedure '`lin`' dotted lines are now generated with the '`\multiput`' command of LaTeX. Depending on the flag `$pictnew` there are two ways to produce dotted quadratic Bézier curves with the subprocedure '`qbez`' and dotted cubic Bézier curves with '`cbez`'. As mentioned above, the latter is done by approximating cubic Bézier curves with four quadratic Bézier curves, but the result is only satisfying if the cubic Bézier curve doesn't contain points of inflection or self-intersection. This can always be achieved by cutting the cubic Bézier curve into suitable segments with `OOoDraw`.

REFERENCES

[1] Bachmaier, Gerhard A.: Using `ebezier`. Package in CTAN:`/macros/latex/contrib/ebezier`, 2002.

[2] Gäßlein, Hubert, Niepraschk, Rolf and Tkadlec, Josef: The `pict2e`-package. Package in CTAN:`/macros/latex/contrib/pict2e/pict2e.dtx`, 2008.

[3] Möller, Herbert: How to generate LaTeX `picture` environments using the GaPFilL method. CTAN:`/graphics/gapfill/GaPDoc1.pdf`, 2006.

[4] Möller, Herbert: *Mathkompass* (`http://wwwmath.uni-muenster.de/u/mollerh`).