Drawing partitions of a simply connected plane domain

Le Phuong Quan
(Cantho University)
`lpquan@ctu.edu.vn`

June 22, 2014

This drawing procedure takes shape from the macros `\psline`, `\pspolygon`, `\multido` and `\parametricplot` in PSTricks[1]. The TEX arithmetic gives the pointwise parametrization of curves.

1. How to make a boundary using Bézier curves

We can make the closed boundary of a plane domain by joining Bézier curves. This can produce various boundaries. We consider here two Bézier curves of degree 5 whose vector functions are

$$\mathbf{r}_a(t) = \sum_{i=0}^{5} \binom{5}{i} t^i (1-t)^{5-i} \mathbf{a}_i, \quad \mathbf{r}_b(t) = \sum_{i=0}^{5} \binom{5}{i} t^i (1-t)^{5-i} \mathbf{b}_i,$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are the position vectors of control points M_i, N_i of the two curves, respectively, $i = 0, \ldots, 5$. Here we choose: $M_0 = (0,0)$, $M_1 = (0,1.5)$, $M_2 = (1.5,1)$, $M_3 = (3,2)$, $M_4 = (5,1.5)$, $M_5 = (5,0)$, and $N_0 = (5,0)$, $N_1 = (5,-1.5)$, $N_2 = (3.5,-1)$, $N_3 = (2,-2)$, $N_4 = (0,-1.5)$, $N_5 = (0,0)$. The two curves and their control points are shown in Figure 1 and the picture in it gives us an obvious explanation of how to combine two Bézier curves to make a closed and smooth boundary. To change the shape of the boundary, we take one couple of multipliers for the second coordinates of M_2, M_3 and another one for those of N_2, N_3. To draw the curves $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$ we

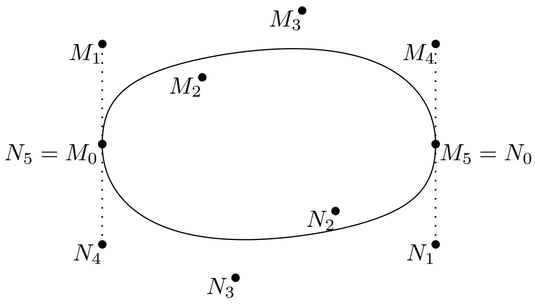

Figure 1: A Bézier-spline boundary.

---

[1] PSTricks is the original work of Timothy Van Zandt (email address: `tvz@econ.insead.fr`). It is currently edited by Herbert Voß (`hvoss@tug.org`).

need their parametrization $(X_a(t), Y_a(t))$ and $(X_b(t), Y_b(t))$, $t \in [0,1]$. From their control points and the given couple of multipliers α_M, β_M, we derive

$$\begin{aligned} X_a(t) &= 15t^2(1-t)^3 + 30t^3(1-t)^2 + 25t^4(1-t) + 5t^5, \\ Y_a(t) &= 7.5t(1-t)^4 + 10\alpha_M t^2(1-t)^3 + 20\beta_M t^3(1-t)^2 + 7.5t^4(1-t). \end{aligned}$$

Similarly, we have

$$\begin{aligned} X_b(t) &= 5(1-t)^5 + 25t(1-t)^4 + 35t^2(1-t)^3 + 20t^3(1-t)^2, \\ Y_b(t) &= -7.5t(1-t)^4 - 10\alpha_N t^2(1-t)^3 - 20\beta_N t^3(1-t)^2 - 7.5t^4(1-t), \end{aligned}$$

where α_N, β_N are multipliers. Then, these curves will be depicted by the macro `\parametricplot` with the corresponding declarations in the `algebraic` form:

```
\def\XoneFive{15*t^2*(1-t)^3+30*t^3*(1-t)^2+25*t^4*(1-t)+5*t^5}
\def\YoneFive#1#2%
{7.5*t*(1-t)^4+(#1)*10*t^2*(1-t)^3+(#2)*20*t^3*(1-t)^2+7.5*t^4*(1-t)}
\def\XtwoFive{5*(1-t)^5+25*t*(1-t)^4+35*t^2*(1-t)^3+20*t^3*(1-t)^2}
\def\YtwoFive#1#2%
{(-7.5)*t*(1-t)^4+(#1)*(-10)*t^2*(1-t)^3+(#2)*(-20)*t^3*(1-t)^2+(-7.5)*t^4*(1-t)}
```

where `#1`, `#2` stand for α_M and β_M in the definition of `\YoneFive` and for α_N and β_N in that of `\YtwoFive`, respectively. In Figure 2, the boundaries are drawn by using together the commands

```
\parametricplot[algebraic,plotpoints=200,linewidth=0.5pt]{0}{1}
{\XoneFive|\YoneFive{alphaM}{betaM}}
\parametricplot[algebraic,plotpoints=200,linewidth=0.5pt]{0}{1}
{\XtwoFive|\YtwoFive{alphaN}{betaN}}
```

where, `alphaM`, `betaM`, `alphaN` and `betaN` are chosen values of α_M, β_M, α_N and β_N, respectively.

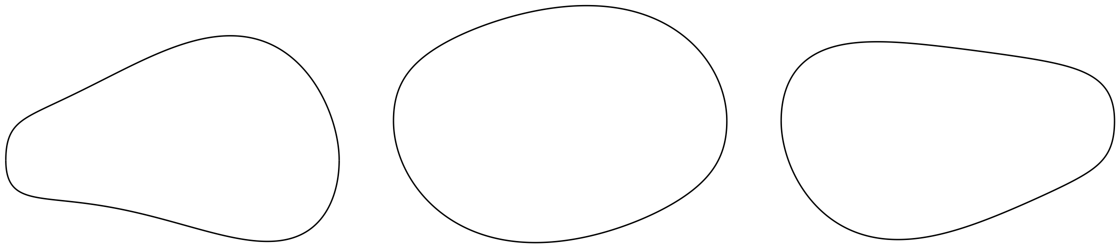

Figure 2: From left to right, corresponding to the couple of values: $\alpha_M = -0.5$, $\beta_M = 2.1$, $\alpha_N = 2.4$, $\beta_N = -0.4$; $\alpha_M = 1.1$, $\beta_M = 1.4$, $\alpha_N = 1.2$, $\beta_N = 1.5$; $\alpha_M = 1.7$, $\beta_M = 0.2$, $\alpha_N = 0.1$, $\beta_N = 1.8$

2. How to draw a partition of a plane domain

To draw a partition of a plane domain whose boundary is made by the way that has just been described, we need the two procedures `\NetDrawOne` and `\NetDrawTwo`, corresponding to $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$. Their calling sequences take three arguments in order: one for c (the number of cells) and the others for α_M and β_M or α_N and β_N. Both of `\NetDrawOne` and `\NetDrawTwo` are used here to find cells in a given grid that have points in common with $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$, and to color those cells. To obtain the definition of these procedures, we first state the problem that we are considering here.

A grid of rectangular cells will be put on a rectangle R containing a domain D whose boundary consist of $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$, including the given values of c, α_M, β_M, α_N and β_N. Then, the steps for drawing a partition of D can be listed as follows:

- Coloring cells that have points in common with the boundary of D. This is the result of calling `\NetDrawOne{`c`}{`α_M`}{`β_M`}` and `\NetDrawTwo{`c`}{`α_N`}{`β_N`}` together.
- Drawing the grid on R, which has been chosen as $R = \{(x, y)\colon -1 \leq x \leq 6, \ -3 \leq y \leq 3\}$.
- Drawing the two curves $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$.

In practice, a partition of D can be given by a calling sequence that has the following structure inside the `\begin{pspicture}` … `\end{pspicture}` environment:

<table>
<tr><th>Commands</th><th>Result</th></tr>
<tr><td>\NetDrawOne{c}{α_M}{β_M}\NetDrawTwo{c}{α_N}{β_N}</td><td>Coloring cells</td></tr>
<tr><td>\multido{\nx=-1.00+\decimal\Xsize}{$c+1$}
\psline[linewidth=0.2pt](\nx,-3)(\nx,3)
\multido{\ny=-3.00+\decimal\Ysize}{$c+1$}
\psline[linewidth=0.2pt](-1,\ny)(6,\ny)</td><td>Drawing the grid on R</td></tr>
<tr><td>\parametricplot[algebraic,plotpoints=200,linecolor=white,
linewidth=0.5pt]{0}{1}{\XoneBFive|\YoneBFive{α_M}{β_M}}
\parametricplot[algebraic,plotpoints=200,linecolor=white,
linewidth=0.5pt]{0}{1}{\XtwoFive|\YtwoFive{α_N}{β_N}}</td><td>Drawing $\mathbf{r}_a(t)$, $\mathbf{r}_b(t)$</td></tr>
</table>

In the following, beside the definitions of `\XoneFive`, `\YoneFive`, `\XtwoFive` and `\YtwoFive`, we list all the remaining macros by their control sequences only and the results derived from running them. They form the whole drawing package, and of course they are also put in the preamble or in a single TEX file to be loaded when running the package.

- `\def\xch`: Setting `\catcode'\p=12`, `\catcode'\t=12`.
- `\def\ych`: Setting `\catcode'\p=11`, `\catcode'\t=11`.
- `\def\decimal#1`: Getting the numeric value of `#1` without unit.
- `\def\FACTORIAL#1`: The factorial function. Ex: `\FACTORIAL{3}` $= 6$.
- `\def\BINOMIAL#1#2`: Binomial coefficients. Ex: `\BINOMIAL{5}{2}` $= 10$.
- `\def\BERNSTEIN#1#2#3`: Bernstein's functions. Ex: `\BERNSTEIN{3}{2}{t}` $= \binom{5}{3} t^3 (1-t)^2$.
- `\def\XoneBC#1`, `\def\XtwoBC#1`: The values of $X_a(t)$, $X_b(t)$. Ex: `\XoneBC{0.5}` $= X_a(0.5)$.
- `\def\YoneBC#1#2#3`, `\def\YtwoBC#1#2#3`: The value of $Y_a(t)$ with given values of α_M, β_M, and the value of $Y_b(t)$ with given values of α_N, β_N. Ex: `\YoneBC{0.5}{1.8}{1.2}` $= Y_a(0.5)$, with $\alpha_M = 1.8$, $\beta_M = 1.2$.

We recall here the main idea for this drawing procedure. We have already the expressions of $\mathbf{r}_a(t)$ and $\mathbf{r}_b(t)$ to draw these Bézier curves by the macro `\parametricplot`. But, we need to have in hand their points' coordinates to determine cells of a grid containing points in common with these curves. Fortunately, we may use the TEX arithmetic to design expressions of $X_a(t)$, $Y_a(t)$, $X_b(t)$ and $Y_b(t)$, hence we can evaluate their values at each given $t \in [0,1]$. For instance, the macro `\def\XoneBC#1` is destined for evaluating the value of $X_a(t)$, and since it is a linear combination of Bernstein's functions, we need binomial coefficients. We do not take directly numeric values of those coefficients, and we have had the macro `\def\BINOMIAL#1#2` do that instead. It might be possible to use this macro in other problems later. In Appendix A, we list the replacement texts of the above macros and all the local and global variables they need.

From the listed macros, `\NetDrawOne` may now have its definition as

```
==================================================================================
\def\NetDrawOne#1#2#3{%
\newdimen\Xsize
\newdimen\Ysize
\newdimen\tempx
\newdimen\tempy
\Xsize=7pt \divide\Xsize by #1
\Ysize=6pt \divide\Ysize by #1
\parametricplot[algebraic,fillstyle=solid,fillcolor=yellow!80,plotpoints=200,
linewidth=0.5pt]{0}{1}{\XoneFive|\YoneFive{#2}{#3}}
\multido{\nz=0.00+0.005}{200}{\XoneBC{\nz}\YoneBC{\nz}{#2}{#3}
\multido{\nx=-1.00+\decimal\Xsize}{#1}
{\tempx=\nx pt\multido{\ny=-3.00+\decimal\Ysize}{#1}
{\tempy=\ny pt
     \ifdim\YBST<\tempy\relax\else\advance\tempy by \Ysize
      \ifdim\YBST>\tempy\relax\else
       \ifdim\XBST<\tempx\relax\else\advance\tempx by \Xsize
        \ifdim\XBST>\tempx\relax\else\pspolygon[fillstyle=solid,fillcolor=blue!70,
linecolor=black,linewidth=0.2pt](\nx,\ny)(\decimal\tempx,\ny)
(\decimal\tempx,\decimal\tempy)(\nx,\decimal\tempy)(\nx,\ny)\relax\fi\fi\fi\fi}}}}
==================================================================================
```

To obtain the definition of `\NetDrawTwo`, we just replace by `\XtwoFive`, `\Ytwofive`, `\XtwoBC` and `\YtwoBC` for `\XoneFive`, `\YoneFive`, `\XoneBC` and `\YoneBC` in that of `\NetDrawaOne`, respectively.

Because of its inevitable shortcoming, TEX often gives approximate results of calculations on dimensions. In the following table, let us see a slight difference between results obtained from the same expression (by declaration of operations) in Maple and in the TEX arithmetic:

t	0.12	0.33	0.54	0.74	0.97
`\XoneBC{`t`}`	0.19081	1.1921	2.6316	4.02428	4.98177
Maple	0.19203	1.1935	2.6328	4.02479	4.98266

Actually, calculations in determining if a point of the two curves belongs a cell almost give the desired result.

Before taking some examples, we give an explanation about what the main algorithm in the procedure `\NetDrawOne` (or `\NetDrawTwo`) is. Let us take an approximate sequence of points for the

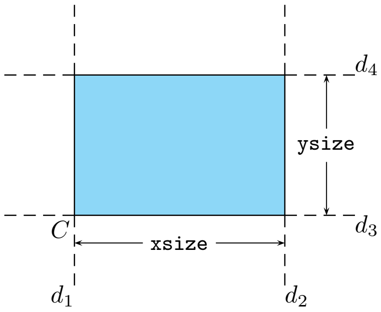

Figure 3: A typical cell.

curve $\mathbf{r}_a(t)$, say P_i(\XoneBC{t_i}, \YoneBC{t_i}{α_M}{β_M}), $i = 1, \ldots, 200$. Then, each P_i is examined whether to be in a cell that has the reference point $C = (x, y)$ by the instruction that: do nothing if P_i is

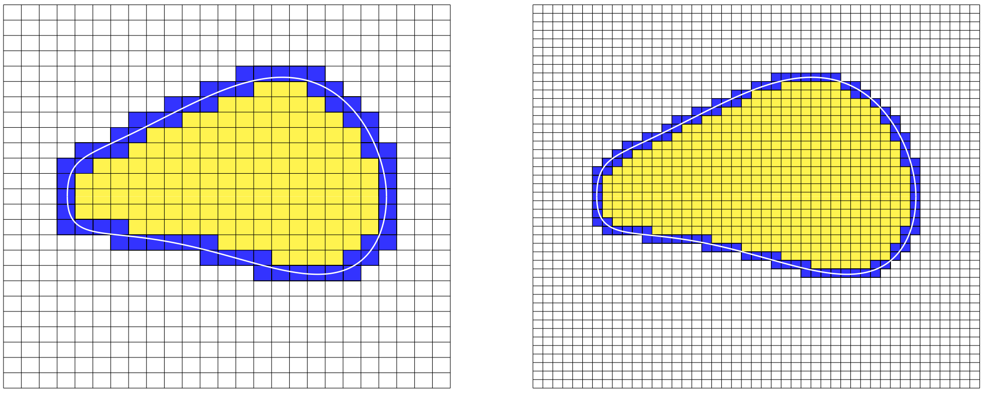

Figure 4: From left to right: $c = 25$ and $c = 45$.

- below d_3, or
- above d_4, or
- on the left side of d_1, or
- on the right side of d_2;

otherwise, color the cell. This algorithm can be expressed in the form of

```
\ifdim\YBST<y \relax\else\advance y by ysize
  \ifdim\YBST>y \relax\else
    \ifdim\XBST<x \relax\else\advance x by xsize
      \ifdim\XBST>x \relax\else\pspolygon[fillstyle=solid,...]
```

where `\XBST` and `\YBST` hold the values of \XoneBC{t_i} and \YoneBC{t_i}{α_M}{β_M}, respectively. This structure is almost the same as that in the procedure `\NetDrawOne` (or `\NetDrawTwo`), but the variables here are declared according to the gloss in Figure 3.

Finally, let us take two examples where we just give values of c for the caption of figures. In Figure 4, we take the same couples $\alpha_M = 1.7$, $\beta_M = 0.2$, $\alpha_N = 0.1$ and $\beta_N = 1.8$ for its two pictures. In Figure 5, we take the same couples $\alpha_M = -0.5$, $\beta_M = 2.1$, $\alpha_N = 2.4$ and $\beta_N = -0.4$ for its three pictures.

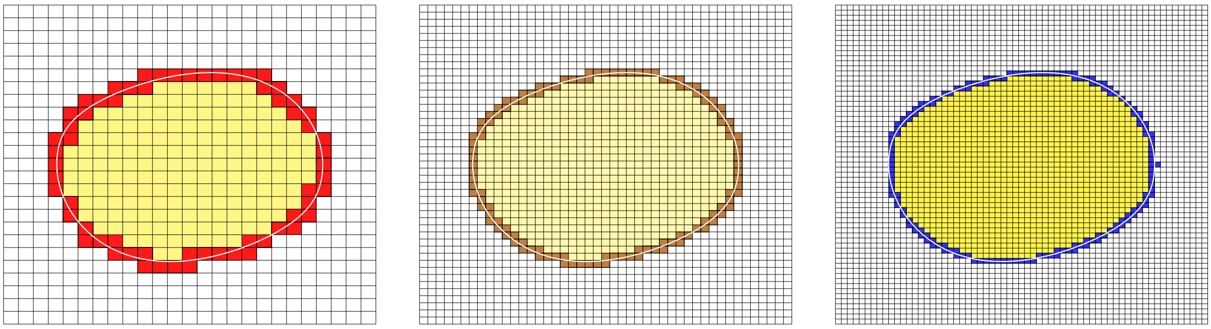

Figure 5: From left to right: $c = 25$, $c = 45$ and $c = 63$.

A. Appendix: detailed definitions

The following is the list of all the detailed definitions needed for the procedure, except those of `\XoneFive`, `\YoneFive`, `\XtwoFive`, `\YtwoFive`, `\NetDrawOne` and `\NetDrawTwo`.

```
----------------------------------------------------------------------------------
\def\xch{\catcode`\p=12 \catcode`\t=12}
\def\ych{\catcode`\p=11 \catcode`\t=11}\xch
\def\dec#1pt{#1}\ych
\def\decimal#1{\expandafter\dec \the#1}
\newcount\Fa\newcount\Fct\newcount\tempA

\def\Factor{\ifnum\Fa=1\relax\else\advance\Fa by -1\multiply\Fct by \Fa\Factor\fi}
\def\FACTORIAL#1{\Fa=#1 \ifnum\Fa=0 \Fct=1\relax\else\Fct=\Fa
\Factor\fi\global\tempA=\Fct}
----------------------------------------------------------------------------------
\newcount\BINOM
\newcount\temp
\newcount\tmp

\def\BINOMIAL#1#2{%
\temp=#1\advance\temp by -#2
\FACTORIAL{#1}
\tmp=\tempA
\FACTORIAL{\temp}
\temp=\tempA
\divide\tmp by \temp
\FACTORIAL{#2}
\temp=\tempA
\divide\tmp by \temp\global\BINOM=\tmp}
----------------------------------------------------------------------------------
\newdimen\Xa\newdimen\Yb\newcount\kc
\newdimen\BSTemp

\def\xmult#1{\ifnum\kc<#1\advance\kc by 1\Yb=\decimal\Xa\Yb\xmult{#1}
\else\relax\fi}
```

```
\def\BERNSTEIN#1#2#3{\Xa=#3pt\kc=0\Yb=1pt\xmult{#1}\kc=0\Xa=-\Xa\advance\Xa
by 1pt\xmult{#2}\BINOMIAL{5}{#1}\global\BSTemp=\BINOM\Yb}
---------------------------------------------------------------------------------
\newdimen\Xrf
\newdimen\Yrf
\newdimen\XoneBST\newdimen\YoneBST
\newdimen\XtwoBST\newdimen\YtwoBST
\newdimen\XBST\newdimen\YBST

\def\XoneBC#1{%
\BERNSTEIN{2}{3}{#1}
\Xrf=1.5pt
\XoneBST=\decimal\BSTemp\Xrf
\BERNSTEIN{3}{2}{#1}
\Xrf=3pt
\advance\XoneBST by \decimal\BSTemp\Xrf
\BERNSTEIN{4}{1}{#1}
\Xrf=5pt
\advance\XoneBST by \decimal\BSTemp\Xrf
\BERNSTEIN{5}{0}{#1}
\Xrf=5pt
\advance\XoneBST by \decimal\BSTemp\Xrf
\global\XBST=\XoneBST}
---------------------------------------------------------------------------------
\def\XtwoBC#1{%
\BERNSTEIN{0}{5}{#1}
\Xrf=5pt
\XtwoBST=\decimal\BSTemp\Xrf
\BERNSTEIN{1}{4}{#1}
\Xrf=5pt
\advance\XtwoBST by \decimal\BSTemp\Xrf
\BERNSTEIN{2}{3}{#1}
\Xrf=3.5pt
\advance\XtwoBST by \decimal\BSTemp\Xrf
\BERNSTEIN{3}{2}{#1}
\Xrf=2pt
\advance\XtwoBST by \decimal\BSTemp\Xrf
\global\XBST=\XtwoBST}
---------------------------------------------------------------------------------
\def\YoneBC#1#2#3{%
\BERNSTEIN{1}{4}{#1}
\Yrf=1.5pt
\YoneBST=\decimal\BSTemp\Yrf
\BERNSTEIN{2}{3}{#1}
\Yrf=1pt
\Yrf=#2\Yrf
\advance\YoneBST by \decimal\BSTemp\Yrf
\BERNSTEIN{3}{2}{#1}
\Yrf=2pt
\Yrf=#3\Yrf
\advance\YoneBST by \decimal\BSTemp\Yrf
\BERNSTEIN{4}{1}{#1}
\Yrf=1.5pt
\advance\YoneBST by \decimal\BSTemp\Yrf
\global\YBST=\YoneBST}
-------------------------------------------------------------------------------------
\def\YtwoBC#1#2#3{%
\BERNSTEIN{1}{4}{#1}
\Yrf=-1.5pt
\YtwoBST=\decimal\BSTemp\Yrf
\BERNSTEIN{2}{3}{#1}
\Yrf=-1pt
\Yrf=#2\Yrf
\advance\YtwoBST by \decimal\BSTemp\Yrf
\BERNSTEIN{3}{2}{#1}
\Yrf=-2pt
\Yrf=#3\Yrf
\advance\YtwoBST by \decimal\BSTemp\Yrf
\BERNSTEIN{4}{1}{#1}
\Yrf=-1.5pt
\advance\YtwoBST by \decimal\BSTemp\Yrf
\global\YBST=\YtwoBST}
-------------------------------------------------------------------------------------
```

References

[1] Dominique Rodriguez, Michael Sharpe & Herbert Voß. *`pstricks-add`: Additional Macros for PSTricks*. Version 3.60, http://ctan.org/tex-archive/graphics/pstricks/contrib, 2013

[2] Eitan M. Gurari. *Writing With TEX*, McGraw-Hill, Inc., 1994, ISBN 0-07-025207-6

[3] Helmut Kopka & Patrick W. Daly. *Guide to LATEX*. Addison-Wesley, Fourth Edition, 2004, ISBN 0321173856

[4] Timothy Van Zandt. *User's Guide*. Version 1.5, http://ctan.org/tex-archive/graphics/pstricks/base, 2007