ISSN: 2037-6367

MatematicaMente

Pubblicazione mensile della sezione veronese della MATHESIS – Società Italiana di Scienze Matematiche e Fisiche – Fondata nel 1895 – Autorizzazione del Tribunale di Verona n. 1360 del 15 – 03 – 1999 – I diritti d'autore sono riservati. Direttore: Luciano Corso - Redazione: Alberto Burato, Elisabetta Capotosto, Carlo Marchiori, Giovanna Tessari – Via IV Novembre, 11/b – 37126 Verona – tel e fax (045) 8344785 – 338 6416432 – e-mail: lcorso@iol.it – Stampa in proprio - Numero 172 – Pubblicato il 5 – 10 – 2012

On Bézier and Subdivision Curves

Len Bos [*]

[*Continued from number* 171]

2.2 **Piecewise Quadratic Curves**

These are more interesting and also more applicable curves. Figure 4 (taken from the Internet) shows an example of designing the Greek letter π using such curves. The control point bookkeeping goes as follows.

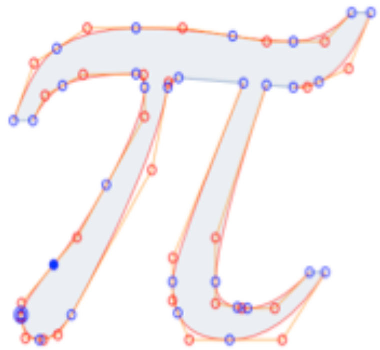

Figure 4: A Piecewise Quadratic Version of π

We begin with a quadratic segment with control points $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2 \in \mathbf{R}^2$: To this we wish to attach a second segment, but so that the resulting composite curve is at least continuous. This means that the *first* point of the second segment should be the same as the *last* point of the first segment, i.e., the first control point of the second segment should equal $\mathbf{b}_2$. In other words, for the second segment we take control points $\mathbf{b}_2$, $\mathbf{b}_3$, $\mathbf{b}_4$. Continuing in this way, we can make a continuous composite piece-wise quadratic curve with n segments from the control points $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_{2n}$ where the ith segment is generated by $\mathbf{b}_{2i}$; $\mathbf{b}_{2i+1}$; $\mathbf{b}_{2i+2}$; $i = 0, \ldots, (n-1)$. Using the formula (2) for a quadratic segment, we can write a formula for this continuous curve, setting, as before, $t_i = t - i$ for $t \in [i, i+1]$:

$$\mathbf{s}(t) = (1-t_i)^2\,\mathbf{b}_{2i} + 2t_i(1-t_i)\mathbf{b}_{2i+1} + t_i^2\mathbf{b}_{2i+2}, \quad i = 0, 1, \cdots, (n-1).$$

Actually we can do better – we can make the curve also have a continuous tangent! How to do this? The first two segments are generated by $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_2$, $\mathbf{b}_3$, $\mathbf{b}_4$. They meet at $\mathbf{b}_2$. Up above we calculated the tangent to a quadratic curve ($\mathbf{Z}_t$) at the "tail" of the curve ($t = 0$) and at the "head" (t = 1). We want the tangent at the head of the first segment to equal the tangent at the tail of the second, i.e. that

$$2(\mathbf{b}_2 - \mathbf{b}_1) = 2(\mathbf{b}_3 - \mathbf{b}_2)$$

which holds if and only if

$$\mathbf{b}_2 = \frac{\mathbf{b}_1 + \mathbf{b}_3}{2}.$$

If we apply this condition for all adjoining segments we will end up with a curve with continuous tangent, provided

$$\mathbf{b}_{2i} = \frac{\mathbf{b}_{2i-1} + \mathbf{b}_{2i+1}}{2}, \quad i = 1, 2, \cdots, (n-1) \quad . \qquad (4)$$

Note that there is no condition at $\mathbf{b}_0$ and $\mathbf{b}_{2n}$ as these are the endpoints of the composite curve and there are no segments to be attached there.

If we insert the midpoints formula (4) into the formula for $\mathbf{s}(t)$ we get

$$\mathbf{s}(t) = (1-t_i)^2\,\frac{\mathbf{b}_{2i-1} + \mathbf{b}_{2i+1}}{2} + 2t_i(1-t_i)\mathbf{b}_{2i+1} + t_i^2\,\frac{\mathbf{b}_{2i+1} + \mathbf{b}_{2i+3}}{2}.$$

for $i \le t \le i + 1$. Notice that this now depends only on the *odd* control points. To emphasize this, let's change notation and set $\mathbf{d}_i := \mathbf{b}_{2i+3}$ (this may appear at first glance a bit of a strange choice, but we'll see that it simplifies the final formula!). Then (after some algebra) we have

$$\mathbf{s}(t) = (1-t_i)^2\,\mathbf{d}_{i-2} + \left\{\frac{(1-t_i)^2}{2} + 2t_i(1-t_i) + \frac{t_i^2}{2}\right\}\mathbf{d}_{i-1} + \frac{t_i^2}{2}\mathbf{d}_i,$$

again for $i \le t \le i + 1$.

Just as in the linear case we can write $\mathbf{s}(t)$ using certain basis functions. Indeed (with a little effort) we can see that

$$s(t) = \sum_i d_i Q_i(t)$$

where

$$Q_i(t) = \begin{cases} \dfrac{t_i^2}{2} & t \in [i, i+1] \\ \dfrac{1 + 2t_{i+1} - 2t_{i+1}^2}{2} & t \in [i+1, i+2] \\ \dfrac{(1-t_{i+2})^2}{2} & t \in [i+2, i+3] \\ 0 & \textit{otherwise} \end{cases} \qquad (5)$$

Here we have been deliberately a bit vague. The sum is formally over all integers, but we only evaluate it for $t \in [0; n]$. Also, technically speaking, we have to add some "artificial" d_i at the beginning and end to handle the boundary effects. Putting in all the precise details obscures a bit the form of the equation, and so we have left them out. The basis functions $Q_i(t)$ are quite interesting. Figure 5 gives a plot of a couple of them. They are

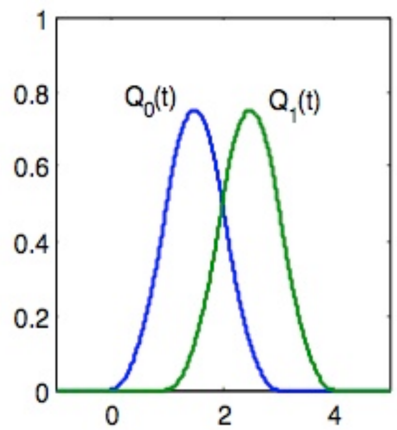

Figure 5: Plots of $Q_0(t)$ and $Q_1(t)$

what is known as quadratic B-splines – they are made up of quadratic pieces and are zero outside the interval $[i; i+3]$. It is easy to check that the other $Q_i(t)$ are translates of $Q_0(t)$, indeed, $Q_i(t) = Q_0(t-i)$, $i = 0; \ldots, (n-2)$. Moreover there is a 2-scale relation:

$$Q_0(t)=\frac{1}{4}\sum_{k=0}^{3}\binom{3}{k}Q_0(2t-k)\,. \qquad (6)$$

Just as for the linear case, this 2-scale relation can be used to calculate the values of $Q_0(t)$ at the half-integers, quarter-integers and so on (and hence all the $Q_0(t)$; and hence the curve $\mathbf{s}(t)$), supposing that we know the values of $Q_0(t)$ for t an integer. But there is another, interesting thing that we can do. Let's write the 2-scale relation abstractly as

$$Q_0(t)=\sum_k a_k Q_0(2t-k)$$

(where the coeficients ak are given by (6)). Then, we may calculate

$$\begin{aligned}\mathbf{s}(t)&=\sum_i \mathbf{d}_i Q_i(t)\\ &=\sum_i \mathbf{d}_i Q_0(t-i)\\ &=\sum_i \mathbf{d}_i\left\{\sum_k a_k Q_0(2(t-i)-k)\right\}\\ &=\sum_{i'}\left\{\sum_{k'} a_{i'-2k'}\mathbf{d}_{k'}\right\}Q_0(2t-i')\end{aligned}$$

after "changing variables" $i' = 2i+k$; $k' = i$. If we think of

$$\mathbf{d}_i^{(1)}=\sum_k a_{i-2k}\mathbf{d}_k$$

as *new* control points, then we have

$$\mathbf{s}(t)=\sum_i \mathbf{d}_i^{(1)} Q_0(2t-i)$$

which is an expression for $\mathbf{s}(t)$ in terms of the basis functions $Q_0(2t-i)$, i.e., piecewise quadratic but with breaks at the half-integers. In other words we have an expression for exactly the same $\mathbf{s}(t)$ considered as a curve with breaks at the half-integers. This is of course possible because any curve with breaks at the integers is automatically also a curve with breaks at the half-integers (it's just that the breaks at the actual half-integers aren't "active"). But now the interesting thing happens. If we plot the polygon of the control points, the one for the $\mathbf{d}_i^{(1)}$ is closer to the curve than the original; see Figure 6.

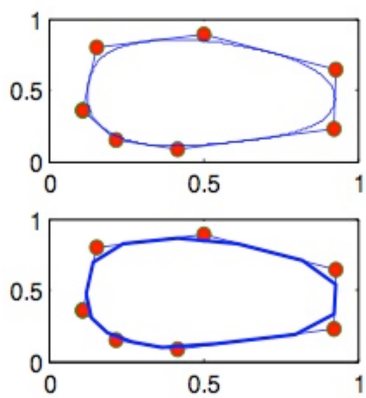

Figure 6: Refined Polygons

In fact, if we repeat this procedure, going from $\mathbf{d}_i$ (a "rough" description of he curves) to $\mathbf{d}_i^{(1)}$ to $\mathbf{d}_i^{(2)}$ etc., using the *refinement* equation

$$\mathbf{d}_i^{(j+1)}=\sum_k a_{i-2k}\,\mathbf{d}_k^{(j)} \qquad (7)$$

the " refined" polygons get closer and closer, and eventually converges, to the curve $\mathbf{s}(t)$.
This gives an algorithm to compute the curve starting from the control points

$$\mathbf{d}_i^{(0)}=\mathbf{d}_i$$

and not using any other information. Figure 6 shows an example of this. The top figure is the original control polygon with the final curve, while the bottom shows the original polygon with the first refinement. It works remarkably well! There's also some interesting geometry to this scheme that helps explain why it works. First of all note that

$$\mathbf{d}_{2i}^{(j+1)}=\sum_k a_{2i-2k}\mathbf{d}_k^{(j)}$$

and

$$\mathbf{d}_{2i+1}^{(j+1)}=\sum_k a_{2i+1-2k}\mathbf{d}_k^{(j)}$$

so that for the refinement of the even control points $\mathbf{d}_{2i}^{(j+1)}$ we use only the even indexed a_k and for the odd control points $\mathbf{d}_{2i+1}^{(j+1)}$ we use only the odd indexed a_k: Specifically, the rule (7) can be written as

$$\mathbf{d}_{2i}^{(j+1)}=\frac{1}{4}\mathbf{d}_i^{(j)}+\frac{3}{4}\mathbf{d}_{i-1}^{(j)}\ ,$$

$$\mathbf{d}_{2i+1}^{(j+1)}=\frac{3}{4}\mathbf{d}_i^{(j)}+\frac{1}{4}\mathbf{d}_{i-1}^{(j)}\ .$$

This means that the edge $\overrightarrow{\mathbf{d}_{i-1}^{(j)}\mathbf{d}_i^{(j)}}$ gets replaced by two new control points at the 1/4 and 3/4 points. Connecting the new control points has the effect of "cutting of the corner" at $\mathbf{d}_i^{(j)}$.
The lower graph in Figure 6 gives an illustration. In fact the scheme is also called Chaikin's Corner Cutting Scheme.
[*to be continued in number* 173]

[*] Professore Ordinario di Analisi Numerica, Università degli Studi di Verona. E-mail: leonardpeter.bos@univr.it

Content and Language Integrated Learning (CLIL)

Con i decreti attuativi della Riforma della Scuola Secondaria di secondo grado del 2010, la metodologia *Content and Language Integrated Learning* (CLIL) si afferma in modo definitivo. L'ITIS Marconi di Verona, come altre scuole d'eccellenza in Italia, da anni ne riconosce il valore didattico e promuove esperienze specifiche in ambito tecnico-scientifico. In particolare, il progetto *Mathematics for Computer Science* attivato nel 2009 con la collaborazione del prof. Leonard Bos, madrelingua inglese, contribuisce realmente allo sviluppo e al potenziamento delle conoscenze e riceve il costante gradimento degli studenti coinvolti. La possibilità di imparare concetti di una specifica disciplina in un contesto didattico non convenzionale, più stimolante, e l'opportunità di migliorare al contempo le competenze linguistiche sono infatti tra i punti di forza di un percorso formativo CLIL, un'opportunità a cui finalmente la scuola italiana non può più rinunciare. (*A.B.*)