Nonuniform splines

1. Method

Nonuniform cubic splines of the "not-a-knot" kind, where at each end the two outermost intervals have the same cubic curve.

2. Quantities given and to be found

Given:

Data points $\mathbf{S}^{(0)}, \ldots, \mathbf{S}^{(n)}$ in $\mathbf{R}^2$, at least some small minimum distance apart.

To be found:

Bézier curves $\mathbf{Q}^{(1)}(t), \ldots, \mathbf{Q}^{(n)}(t)$.

Here $\mathbf{Q}^{(i)}(t)$ is parameterized on $0 \leq t \leq \Delta t_i$, where $\Delta t_i = \|\mathbf{S}^{(i)} - \mathbf{S}^{(i-1)}\|$ and $\mathbf{Q}^{(i)}(0) = \mathbf{S}^{(i-1)}$, $\mathbf{Q}^{(i)}(\Delta t_i) = \mathbf{S}^{(i)}$.

3. Auxiliary quantities

Distances $\Delta t_1, \ldots, \Delta t_n$ between knots, taken to be the same as the Euclidean distances between the $\mathbf{S}^{(i)}$.

Unknown vectors $\mathbf{v}^{(0)}, \ldots, \mathbf{v}^{(n)}$, where $\mathbf{v}^{(i)}$ is the velocity at $\mathbf{S}^{(i)}$.

Vectors $\Delta\mathbf{S}^{(1)}, \ldots, \Delta\mathbf{S}^{(n)}$, representing the segments between the $\mathbf{S}^{(i)}$.

Unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(n)}$ along the segments.

Vectors $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(n)}$ and $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}$ that are scaled versions of the unit vectors.

Control points for Bézier curves $\mathbf{P}^{(1)}(t), \ldots, \mathbf{P}^{(n)}(t)$ parameterized on $0 \leq t \leq 1$. Then $\mathbf{Q}^{(i)}(t) = \mathbf{P}^{(i)}(t/\Delta t_i)$.

4. The algorithm

For $i = 1, \ldots, n$:

$\Delta\mathbf{S}^{(i)} = \mathbf{S}^{(i)} - \mathbf{S}^{(i-1)}$

$\Delta t_i = \|\Delta\mathbf{S}^{(i)}\| = \sqrt{(S_1^{(i)})^2 + (S_2^{(i)})^2}$

$\mathbf{u}^{(i)} = \frac{1}{\Delta t_i}\Delta\mathbf{S}^{(i)}$

$\mathbf{w}^{(i)} = \frac{1}{\Delta t_i}\mathbf{u}^{(i)}$

$\mathbf{z}^{(i)} = \frac{1}{\Delta t_i}\mathbf{w}^{(i)}$.

Solve the $(n+1)\times(n+1)$ set of equations $MV = B$, where rows and columns are indexed starting from 0 and

V is an $(n+1)\times 2$ matrix of unknown vectors with rows $\mathbf{v}^{(0)},\ldots,\mathbf{v}^{(n)}$,

B is an $(n+1)\times 2$ matrix with rows $\mathbf{B}^{(0)},\ldots,\mathbf{B}^{(n)}$, where

$\mathbf{B}^{(0)} = 2\left(\mathbf{z}^{(2)} - \mathbf{z}^{(1)}\right)$

$\mathbf{B}^{(i)} = 3\left(\mathbf{w}^{(i+1)} + \mathbf{w}^{(i)}\right)$ for $i = 1,\ldots,n-1$

$\mathbf{B}^{(n)} = 2\left(\mathbf{z}^{(n)} - \mathbf{z}^{(n-1)}\right)$.

$$M = \begin{pmatrix} -\frac{1}{\Delta t_1^2} & \left(\frac{1}{\Delta t_2^2} - \frac{1}{\Delta t_1^2}\right) & \frac{1}{\Delta t_2^2} & 0 & 0 & \ldots & 0 \\ \frac{1}{\Delta t_1} & 2\left(\frac{1}{\Delta t_1} + \frac{1}{\Delta t_2}\right) & \frac{1}{\Delta t_2} & 0 & 0 & \ldots & 0 \\ 0 & \frac{1}{\Delta t_2} & 2\left(\frac{1}{\Delta t_2} + \frac{1}{\Delta t_3}\right) & \frac{1}{\Delta t_3} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & \frac{1}{\Delta t_{n-1}} & 2\left(\frac{1}{\Delta t_{n-1}} + \frac{1}{\Delta t_n}\right) & \frac{1}{\Delta t_n} \\ 0 & \ldots & 0 & 0 & -\frac{1}{\Delta t_{n-1}^2} & \left(\frac{1}{\Delta t_n^2} - \frac{1}{\Delta t_{n-1}^2}\right) & \frac{1}{\Delta t_n^2} \end{pmatrix}.$$

In the extended matrix $(M|B)$, by subtracting $\frac{1}{\Delta t_2}$ times the second row from the first and adding $\frac{1}{\Delta t_{n-1}}$ times the next-to-last row to the last, the system can be made tridiagonal and solved using standard methods.

The control points for $\mathbf{P}^{(i)}(t)$ $(i = 1,\ldots,n)$ are $\mathbf{P}_0^{(i)}, \mathbf{P}_1^{(i)}, \mathbf{P}_2^{(i)}, \mathbf{P}_3^{(i)}$, where

$\mathbf{P}_0^{(i)} = \mathbf{S}^{(i-1)}$,
$\mathbf{P}_1^{(i)} = \mathbf{S}^{(i-1)} + \frac{\Delta t_i}{3}\mathbf{v}^{(i-1)}$
$\mathbf{P}_2^{(i)} = \mathbf{S}^{(i)} - \frac{\Delta t_i}{3}\mathbf{v}^{(i)}$
$\mathbf{P}_3^{(i)} = \mathbf{S}^{(i)}$.

A routine can then evaluate $\mathbf{P}^{(i)}(t)$ for any t with $0 \le t \le 1$.

Finally, as mentioned $\mathbf{Q}^{(i)}(t) = \mathbf{P}^{(i)}(\frac{t}{\Delta t_i})$ for $0 \le t \le \Delta t_i$.

5. Uniform plotting of nonuniform splines

Let $t_0 = 0$, $t_1 = \Delta t_1$, $t_2 = t_1 + \Delta t_1$, ... ,$t_n = t_{n-1} + \Delta t_n$. The spline curve determined by $\mathbf{Q}^{(1)}(t), \ldots, \mathbf{Q}^{(n)}(t)$ together is defined mathematically by

$$Q(t) = \begin{cases} \mathbf{Q}^{(1)}(t) & \text{if} \quad 0 \le t < t_1 \\ \mathbf{Q}^{(2)}(t - t_1) & \text{if} \quad t_1 \le t < t_2 \\ \cdots & \cdots \quad \cdots \\ \mathbf{Q}^{(n)}(t - t_{n-1}) & \text{if} \quad t_{n-1} \le t \le t_n \end{cases}$$

It may be desirable to plot the curve using equally spaced parameter values, rather than let points bunch up. Suppose that about K points per unit are desired. (If coordinates are measured in inches, K might be 10 or 20; if in pixels, K might be 0.1 or 0.2.) Let N be an integer obtained by rounding Kt_n. Let $dt = t_n/N$. Then the points to plot are

$Q(0), Q(dt), Q(2dt), Q(3dt), \ldots, Q(Ndt)$.

Rather than evaluate $Q(t)$ using `if` statements, it is possible to keep a running t value that starts at 0 and is decremented by the current Δt_i if it exceeds Δt_i. More specifically:

```
t = 0;
i = 1;
newpt = S[0];
for (j=0; j<N; j++)
{
    t += dt;
    while (t >= deltat[i])
    {
        t -= deltat[i];
        i++;
        // just to be sure, check that i <= n
    }
    lastpt = newpt;
    newpt = Q(i,t);
    // draw line segment from lastpt to newpt
}
```

If $\Delta t_i < dt$ for some i, it is possible that segment i should be skipped over, which this routine will do.