Project 2

Kerman Munitxa, Júlia Gumà, Markel Pisano **markel.p@digipen.edu**
kerman.m@digipen.edu j.gum@digipen.edu
540002615 540002517 540000117

1. Description of the problem

Given $n+1$ points, construct a curve passing through all of them. The output curve mantains the twice differentiability in their extreme points, as Cubic Splines shall do.

2. Mathematical explanation of numerical methods

(mat300 slides, a) (mat300 slides, b)

Given $n+1$ points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ with $x_i \neq x_j$ for $i \neq j$ find a polynomial p satisfying $p(t_i) = P_i$ for $i = 0, 1, \dots, n$.

In such problems we only restrict the polynomial to pass through certain points. We may want as well to restrict the slope of the polynomial at those points, to make it increasing or decreasing at certain point, or to have maximum or minimum values at those points. For doing so we have to impose conditions to the derivatives of p.

2.1 Hermite interpolant polynomial

Let $(x\_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ with $x_i \neq x_j$ for $i \neq j$. Let $z_i \in$ for $i = 0, 1, \dots, n$. A Hermite interpolant polynomial p satisfies:

- $p(t_i) = P_i$ for $i = 0, 1, \dots, n$.
- $p'(t_i) = z_i$ for some i.

2.2 Osculating polynomials

We can extend Hermite interpolation to a higher order of derivatives.

Let $P_{00}(t_0) = (x_{00}, y_{00}), P_{10}(t_1) = (x_{10}, y_{10}), \dots, P_{n0}(t_n) = (x_{n0}, y_{n0})$ with distinct t-parameter. Let $P_{ij} \in$ for $i = 0, 1, \dots, n$ and $j = 0, \dots, r_i$. An osculating interpolant polynomial p satisfies:

- $p^j(t_i) = P_{ij}$; $i = 0, 1, \dots, n$; $j = 0, \dots, r_i$

2.3 The cubic spline interpolation problem

Given $n+1$ points $(x_0, y_0), \ldots, (x_n, y_n)$ with $x_i < x_{i+1}$ for $i = 0, \ldots, n-1$, cubic spline interpolation consists of finding an interpolant polynomial $p \in P^n_{3,2}[t_0, \ldots, t_n]$ through the given points.

As we have $n+1$ constraints and $dim(P^n_{3,2}[t_0, \ldots, t_n]) = 4n - 3(n-1) = n+3$ we need to impose two more conditions to have a unique solution:

$p^{''}(t_0) = 0$ and $p^{''}(t_n) = 0$.

3. Code implementation

First of all an array for t at range $[0, n]$ is created. Then a matrix of polynomials of the form $P_i = a_0 + a_1 t_i + a_2 t_i^2 + a_3 t_i^3 + a_4 (t_i - t_1)^3_+ + \cdots + a_{n+2}(t_i - t_{i-1})^3_+$, being $i = 0, \ldots, n$ the row number. We fill the last columns with the corresponding value at t of each axis (line 28).

After this we add other two rows to the matrix corresponding to the second derivative for the extreme points equaling to 0 on the augmented part (line 55).

Then we apply *Reduced Row Echelon Form* to extract the values of a for xyz (line 72).

After this we substitute a on the polynomial coefficients and using a mesh of t values $\in [0, n]$ with k *outputnodes* nodes. This gives points in the curve to be plotted (line 86).

4. Examples

Lets take for instance the set of points $S = P_0, P_1, P_2$, being $P_0 = (0, 1)$, $P_1 = (1, 2)$, $P_2 = (0, 3)$.

Our set of t are $T = 0, 1, 2$.

Our basis will be $B = 1, t, t^2, t^3, (t-1)^3_+$

We get the augmented matrix of polynomials:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & | & 0 & | & 1 \\ 1 & 1 & 1 & 1 & 0 & | & 1 & | & 2 \\ 1 & 2 & 4 & 8 & 1 & | & 0 & | & 3 \\ 0 & 0 & 2 & 0 & 0 & | & 0 & | & 0 \\ 0 & 0 & 2 & 12 & 6 & | & 0 & | & 0 \end{bmatrix} \tag{1}$$

Now we apply *RREF* to it and we get the coefficients for X and Y from the last two columns.

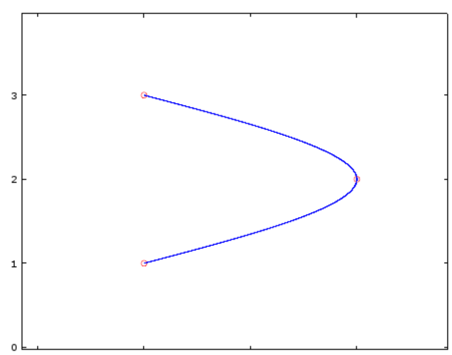
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & | & 0 & | & 1 \\ 0 & 1 & 0 & 0 & 0 & | & \frac{3}{2} & | & 1 \\ 0 & 0 & 1 & 0 & 0 & | & 0 & | & 0 \\ 0 & 0 & 0 & 1 & 0 & | & -\frac{1}{2} & | & 0 \\ 0 & 0 & 0 & 0 & 1 & | & 1 & | & 0 \end{bmatrix} \tag{2}$$

Now we substitute this coefficients in the polynomial

$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 (t-1)^3_+$

giving

$P_x(t) = \frac{3}{2}t - \frac{1}{2}t^3 + (t-1)^3$ and $P_y(t) = 1 + t$.



(i) time = 0.0323131s

5. Observations

Cubic Splines are useful because changing points changes very little the resultant curve, not like interpolation methods such as *Newton method*. The curves are smooth because are twice differentiable. If you have a lot of points, like we will have a big number to the power of three, it may be too big and start losing precision.

References

mat300 slides. Osculating polynomials. https://distance.digipen.edu/2020-spring/pluginfile.php/68872/mod_resource/content/1/lecture7.pdf, a.

mat300 slides. Cubic spline. https://distance.digipen.edu/2020-spring/pluginfile.php/72175/mod_resource/content/1/lecture9.pdf, b.