Fall 2018 CIS 515

Fundamentals of Linear Algebra and Optimization
Jean Gallier

Project 3

November 14, 2018; Due November 28, 2018

The purpose of this project is to solve a curve interpolation problem using cubic splines. This type of problem arises frequently in computer graphics and in robotics (path planning).

Recall from Project 1 that polynomial curves of degree $\leq m$ can be defined in terms of control points and the Bézier polynomials. Cubic Bézier curves are often used because they are cheap to implement and give more flexibility than quadratic Bézier curves.

A *cubic Bézier curve* $C(t)$ (in $\mathbb{R}^2$ or $\mathbb{R}^3$) is specified by a list of four *control points* (b_0, b_2, b_2, b_3) and is given parametrically by the equation

$$C(t) = (1-t)^3\, b_0 + 3(1-t)^2 t\, b_1 + 3(1-t)t^2\, b_2 + t^3\, b_3.$$

Clearly, $C(0) = b_0$, $C(1) = b_3$, and for $t \in [0,1]$, the point $C(t)$ belongs to the convex hull of the control points b_0, b_1, b_2, b_3.

Interpolation problems require finding curves passing through some given data points and possibly satisfying some extra constraints.

It is known that Lagrange interpolation is not very satisfactory when N is greater than 5, since Lagrange interpolants tend to oscillate in an undesirable manner. Thus, we turn to Bézier spline curves.

A *Bézier spline curve* F is a curve which is made up of curve segments which are Bézier curves, say $C_1, \ldots, C_m$ $(m \geq 2)$. We will assume that F defined on $[0, m]$, so that for $i = 1, \ldots, m$,

$$F(t) = C_i(t - i + 1), \quad i - 1 \leq t \leq i.$$

Typically, some smoothness is required between any two junction points, that is, between any two points $C_i(1)$ and $C_{i+1}(0)$, for $i = 1, \ldots, m-1$. We require that $C_i(1) = C_{i+1}(0)$ (C^0-*continuity*), and typically that the derivatives of C_i at 1 and of C_{i+1} at 0 agree up to second order derivatives. This is called C^2-*continuity*, and it ensures that the tangents agree as well as the curvatures.

There are a number of interpolation problems, and we consider one of the most common problems which can be stated as follows:

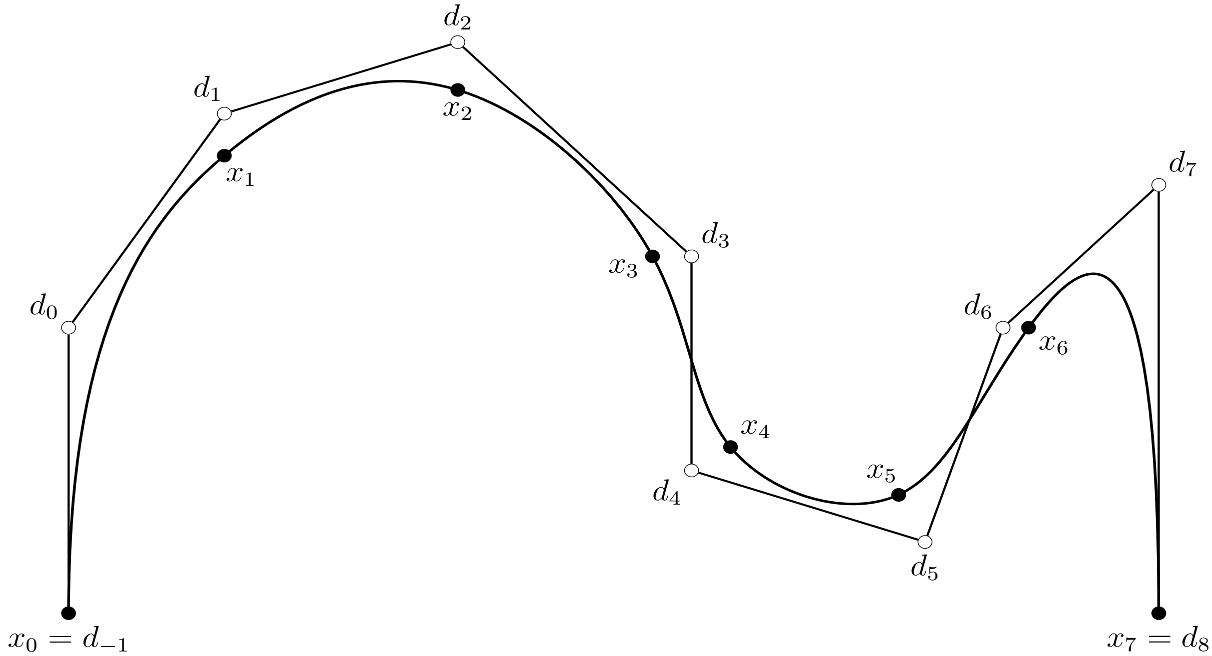

Figure 1: A C^2 cubic interpolation spline curve passing through the points $x_0, x_1,\ x_2, x_3,$ $x_4, x_5,\ x_6, x_7$

Problem: Given $N+1$ data points $x_0, \ldots, x_N$, find a C^2 cubic spline curve F, such that $F(i) = x_i$, for all i, $0 \leq i \leq N$ $(N \geq 2)$.

A way to solve this problem is to find $N+3$ auxiliary points $d_{-1}, \ldots, d_{N+1}$ called *de Boor control points* from which N Bézier curves can be found. Actually,

$$d_{-1} = x_0 \quad \text{and} \quad d_{N+1} = x_N$$

so we only need to find $N+1$ points $d_0, \ldots, d_N$.

It turns out that the C^2-continuity constraints on the N Bézier curves yield only $N-1$ equations, so d_0 and d_N can be chosen arbitrarily. In practice, d_0 and d_N are chosen according to various "end conditions," such as prescribed velocities at x_0 and x_N. For the time being, we will assume that d_0 and d_N are given.

Figure 1 illustrates an interpolation problem involving $N+1 = 7+1 = 8$ data points. The control points d_0 and d_7 were chosen arbitrarily.

It can be shown that $d_1, \ldots, d_{N-1}$ are given by the linear system

$$\begin{pmatrix} \frac{7}{2} & 1 & & & \\ 1 & 4 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & 1 & 4 & 1 \\ & & & 1 & \frac{7}{2} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix} = \begin{pmatrix} 6x_1 - \frac{3}{2}d_0 \\ 6x_2 \\ \vdots \\ 6x_{N-2} \\ 6x_{N-1} - \frac{3}{2}d_N \end{pmatrix}.$$

The derivation of the above system assumes that $N \geq 4$. If $N = 3$, this system reduces to

$$\begin{pmatrix} \frac{7}{2} & 1 \\ 1 & \frac{7}{2} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 6x_1 - \frac{3}{2}d_0 \\ 6x_2 - \frac{3}{2}d_3 \end{pmatrix}$$

When $N = 2$, it can be shown that d_1 is given by

$$d_1 = 2x_1 - \frac{1}{2}d_0 - \frac{1}{2}d_2.$$

Observe that when $N \geq 3$, the above matrix is strictly diagonally dominant, so it is invertible. Actually, the above system needs to be solved for the x-coordinates and for the y-coordinates of the d_is (and also for the z-coordinates, if the points are in $\mathbb{R}^3$). Once the above system is solved, the Bézier cubics $C_1, \ldots, C_N$ are determined as follows (we assume $N \geq 2$): For $2 \leq i \leq N-1$, the control points $(b_0^i, b_1^i, b_2^i, b_3^i)$ of C_i are given by

$$\begin{aligned} b_0^i &= x_{i-1} \\ b_1^i &= \frac{2}{3}d_{i-1} + \frac{1}{3}d_i \\ b_2^i &= \frac{1}{3}d_{i-1} + \frac{2}{3}d_i \\ b_3^i &= x_i. \end{aligned}$$

The control points $(b_0^1, b_1^1, b_2^1, b_3^1)$ of C_1 are given by

$$\begin{aligned} b_0^1 &= x_0 \\ b_1^1 &= d_0 \\ b_2^1 &= \frac{1}{2}d_0 + \frac{1}{2}d_1 \\ b_3^1 &= x_1, \end{aligned}$$

and the control points $(b_0^N, b_1^N, b_2^N, b_3^N)$ of C_N are given by

$$\begin{aligned} b_0^N &= x_{N-1} \\ b_1^N &= \frac{1}{2}d_{N-1} + \frac{1}{2}d_N \\ b_2^N &= d_N \\ b_3^N &= x_N. \end{aligned}$$

Prove that the tangent vectors m_0 at x_0 and m_N at x_N are given by

$$\begin{aligned} m_0 &= 3(d_0 - x_0) \\ m_N &= 3(x_N - d_N). \end{aligned}$$

End Conditions.

One method to determine the points d_0 and d_N is the *natural end condition*, which consists in setting the second derivatives at x_0 and at x_N to be zero, that is,

$$C_1''(0) = 0, \quad C_N''(1) = 0.$$

(1) Prove that the second derivative at b_0 of a Bézier cubic specified by the control points (b_0, b_1, b_2, b_3) is

$$6(b_0 - 2b_1 + b_2),$$

and the second derivative at b_3 is

$$6(b_1 - 2b_2 + b_3).$$

Prove that our system becomes

$$\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix} = \begin{pmatrix} 6x_1 - x_0 \\ 6x_2 \\ \vdots \\ 6x_{N-2} \\ 6x_{N-1} - x_N \end{pmatrix},$$

where d_0, d_N are given by

$$\begin{aligned} d_0 &= \frac{2}{3}x_0 + \frac{1}{3}d_1 \\ d_N &= \frac{1}{3}d_{N-1} + \frac{2}{3}x_N. \end{aligned}$$

Note that d_0 is on the line segment (x_0, d_1) (1/3 of the way from x_0) and d_N is on the line segment (d_{N-1}, x_N) (1/3 of the way from x_N).

In the above derivation we assumed that $N \geq 4$. If $N = 3$, show that this system reduces to

$$\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 6x_1 - x_0 \\ 6x_2 - x_3 \end{pmatrix}.$$

Another method to determine the points d_0 and d_N is the *quadratic end condition*, which consists in requiring that the second derivatives at x_0 and at x_1 agree, and similarly for the second derivatives at x_{N-1} and x_N; this means that

$$C_1''(0) = C_1''(1) \quad \text{and} \quad C_N''(0) = C_N''(1).$$

(2) Prove that our system becomes

$$\begin{pmatrix} 5 & 1 & & & 0 \\ 1 & 4 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & 1 & 4 & 1 \\ & & & 1 & 5 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix} = \begin{pmatrix} 7x_1 - x_0 \\ 6x_2 \\ \vdots \\ 6x_{N-2} \\ 7x_{N-1} - x_N \end{pmatrix},$$

where d_0, d_N are given by

$$\begin{aligned} d_0 &= d_1 + \frac{2}{3}x_0 - \frac{2}{3}x_1 \\ d_N &= d_{N-1} + \frac{2}{3}x_N - \frac{2}{3}x_{N-1}. \end{aligned}$$

Geometrically, the vector $d_0 - d_1$ is equal to $\frac{2}{3}(x_0 - x_1)$ and similarly the vector $d_N - d_{N-1}$ is equal to $\frac{2}{3}(x_N - x_{N-1})$; in particular, the line segments (d_0, d_1) and (x_0, x_1) are parallel, and so are (d_N, d_{N-1}) and (x_N, x_{N-1}).

In the above derivation we assumed that $N \geq 4$. If $N = 3$, show that this system reduces to

$$\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 7x_1 - x_0 \\ 7x_2 - x_3 \end{pmatrix}.$$

Yet another method known as *Bessel end condition* is to require the first derivative $C_1'(0)$ at x_0 to be equal to the first derivative of the unique parabola passing through x_0, x_1, x_2, for $t = 0, 1, 2$, and to require the first derivative $C_N'(1)$ at x_N to be equal to the first derivative of the unique parabola passing through x_{N-2}, x_{N-1}, x_N, for $t = 0, 1, 2$.

A parabola passing through x_0 and x_2 as above is given by

$$C(t) = \frac{(2-t)^2}{4}x_0 + \frac{(2-t)t}{2}b_1 + \frac{t^2}{4}x_2,$$

so to require that $C(1) = x_1$ means that

$$x_1 = \frac{1}{4}x_0 + \frac{1}{2}b_1 + \frac{1}{4}x_2,$$

which yields

$$b_1 = -\frac{1}{2}x_0 + 2x_1 - \frac{1}{2}x_2.$$

(3) Show that

$$m_0 = C'(0) = b_1 - x_0 = -\frac{3}{2}x_0 + 2x_1 - \frac{1}{2}x_2,$$

and using the fact that $d_0 = x_0 + \frac{1}{3}m_0$, show that

$$d_0 = \frac{1}{2}x_0 + \frac{1}{2}\left(x_1 + \frac{1}{3}(x_1 - x_2)\right).$$

Geometrically, d_0 is the midpoint of the line segment from x_0 to a point obtained by extrapolation from x_1 and x_2 $(x_1 + \frac{1}{3}(x_1 - x_2))$.

Similarly, for the parabola interpolating x_{N-2}, x_{N-1} and x_N, we get

$$b_N = -\frac{1}{2}x_{N-2} + 2x_{N-1} - \frac{1}{2}x_N,$$

and show that

$$m_N = C'(2) = x_N - b_N = \frac{1}{2}x_{N-2} - 2x_{N-1} + \frac{3}{2}x_N,$$

and using the fact that $d_N = x_N - \frac{1}{3}m_N$, show that

$$d_N = \frac{1}{2}\left(x_{N-1} + \frac{1}{3}(x_{N-1} - x_{N-2})\right) + \frac{1}{2}x_N.$$

Geometrically, d_N is the midpoint of the line segment from x_N to a point obtained by extrapolation from x_{N-1} and x_{N-2} $(x_{N-1} + \frac{1}{3}(x_{N-1} - x_{N-2}))$. Note that the above derivation is correct for $N \geq 2$.

Finally, there is the *not a knot end condition*, which consists in forcing the first two Bézier segments C_1 and C_2 to belong to the same cubic curve, and similarly for the last two Bézier segments C_{N-1} and C_N. This amounts to require that $C_1'''(1) = C_2'''(0)$ and $C_{N-1}'''(1) = C_N'''(0)$.

(4) Prove that the third derivative at b_0 and at b_3 of a Bézier cubic specified by the control points (b_0, b_1, b_2, b_3) is

$$6(-b_0 + 3b_1 - 3b_2 + b_3).$$

Prove that if $N = 3$, then

$$\begin{aligned}
d_0 &= \frac{7}{18}x_0 + \frac{8}{9}x_1 + \frac{7}{18}x_2 - \frac{2}{3}d_2 \\
d_1 &= -\frac{1}{6}x_0 + \frac{4}{3}x_1 - \frac{1}{6}x_2 \\
d_2 &= -\frac{1}{6}x_1 + \frac{4}{3}x_2 - \frac{1}{6}x_3 \\
d_3 &= \frac{7}{18}x_1 + \frac{8}{9}x_2 + \frac{7}{18}x_3 - \frac{2}{3}d_1
\end{aligned}$$

are already computed in terms of $x_0, \ldots, x_3$, and there is no need to solve any linear system.

Prove that if $N = 4$, then

$$\begin{aligned}
d_1 &= -\frac{1}{6}x_0 + \frac{4}{3}x_1 - \frac{1}{6}x_2 \\
d_3 &= -\frac{1}{6}x_2 + \frac{4}{3}x_3 - \frac{1}{6}x_4 \\
d_2 &= \frac{3}{2}x_2 - \frac{1}{4}d_1 - \frac{1}{4}d_3 \\
d_0 &= \frac{7}{18}x_0 + \frac{8}{9}x_1 + \frac{7}{18}x_2 - \frac{2}{3}d_2 \\
d_4 &= \frac{7}{18}x_2 + \frac{8}{9}x_3 + \frac{7}{18}x_4 - \frac{2}{3}d_2.
\end{aligned}$$

Prove that if $N \geq 5$, our linear system becomes the $(N-3) \times (N-3)$ system

$$\begin{pmatrix} 4 & 1 & 0 & & 0 \\ 1 & 4 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & 1 & 4 & 1 \\ 0 & & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} d_2 \\ d_3 \\ \vdots \\ d_{N-3} \\ d_{N-2} \end{pmatrix} = \begin{pmatrix} 6x_2 + \frac{1}{6}x_0 - \frac{4}{3}x_1 + \frac{1}{6}x_2 \\ 6x_3 \\ \vdots \\ 6x_{N-3} \\ 6x_{N-2} + \frac{1}{6}x_{N-2} - \frac{4}{3}x_{N-1} + \frac{1}{6}x_N \end{pmatrix},$$

and d_0, d_1, d_{N-1}, d_N are given by

$$\begin{aligned}
d_0 &= \frac{7}{18}x_0 + \frac{8}{9}x_1 + \frac{7}{18}x_2 - \frac{2}{3}d_2 \\
d_1 &= -\frac{1}{6}x_0 + \frac{4}{3}x_1 - \frac{1}{6}x_2 \\
d_{N-1} &= -\frac{1}{6}x_{N-2} + \frac{4}{3}x_{N-1} - \frac{1}{6}x_N \\
d_N &= \frac{7}{18}x_{N-2} + \frac{8}{9}x_{N-1} + \frac{7}{18}x_N - \frac{2}{3}d_{N-2}.
\end{aligned}$$

If $N = 5$, this system reduces to

$$\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 6x_2 + \frac{1}{6}x_0 - \frac{4}{3}x_1 + \frac{1}{6}x_2 \\ 6x_3 + \frac{1}{6}x_3 - \frac{4}{3}x_4 + \frac{1}{6}x_5 \end{pmatrix}.$$

(5) Implement the Gaussian elimination method with partial pivoting as well as the method for solving a triangular system by back-substitution.

Use your program to solve several instances of the interpolation problem. Verify that no pivoting is needed.

(6) Implement the LU-factorization method for tridiagonal matrices and test it on the same interpolation problems as in (5).

Do you notice any improvement over Gaussian elimination (running time, numerical precision)?

(7) After computing $d_1, \ldots, d_{N-1}$, use your program from Project 1 to compute the control points for the Bézier curves $C_1, \ldots C_N$ and to plot these Bézier segments (for $t \in [0, 1]$) to visualize the interpolating spline. Experiment with the choice of end conditions.

TOTAL: 300 points.