Project 3

Danica Fine and Armon Shariati

November 12, 2015

PROBLEM 0

We wish to prove that the tangent vectors m_0 at x_0 and m_N at x_N are given by

$$m_0 = 3(d_0 - x_0)$$
$$m_N = 3(x_N - d_N).$$

Since the curves are given by $C(t)$ and the set of control points, $b_0, \ldots, b_3$, we compute the derivative of $C(t)$ and evaluate at 0 and 1 to get m_0 and m_n, respectively.

$$\begin{aligned} C(t) &= (1-t)^3 b_0 + 3(1-t)^2 t b_1 + 3(1-t)t^2 b_2 + t^3 b_3 \\ C'(t) &= -3(1-t)^2 b_0 - 6(1-t)t b_1 + 3(1-t)^2 b_1 - 3t^2 b_2 + 6(1-t)t b_2 + 3t^2 b_3 \\ &= 3(1-t)^2(b_1 - b_0) + 6t(1-t)(b_2 - b_1) + 3t^2(b_3 - b_2) \end{aligned}$$

So for m_0, we get

$$\begin{aligned} C'(0) &= 3(1)^2(b_1 - b_0) + 6(0)(1-t)(b_2 - b_1) + 3(0)^2(b_3 - b_2) \\ &= 3(b_1 - b_0) \\ m_0 &= 3(d_0 - x_0), \end{aligned}$$

where the final substitution is made from equations given to us in the project instructions for the first curve. Similarly, for m_N we acquire

$$\begin{aligned} C'(1) &= 3(0)^2(b_1 - b_0) + 6(1)(0)(b_2 - b_1) + 3(1)^2(b_3 - b_2) \\ &= 3(b_3 - b_2) \\ m_0 &= 3(x_N - d_N), \end{aligned}$$

where the substitutions are provided to us for the final curve, C_N.

PROBLEM 1

From $C'(t)$ above, we compute $C''(t)$

$$\begin{aligned} C''(t) &= -6(1-t)^2(b_1 - b_0) + 6((1-t) - t)(b_2 - b_1) + 6t(b_3 - b_2) \\ &= -6(1-t)b_1 + 6(1-t)b_0 + 6(1-t)(b_2 - b_1) - 6t(b_2 - b_1) + 6tb_3 - 6tb_2. \end{aligned}$$

For C_1, we evaluate $C''(t)$ for $t=0$ to get

$$\begin{aligned}
C''(0) &= -6(0)b_1 + 6(0)b_0 + 6(0)(b_2-b_1) - 6(1)(b_2-b_1) + 6(1)b_3 - 6(1)b_2 \\
&= -6b_2 + 6b_1 + 6b_3 - 6b_2 \\
&= 6(b_1 - 2b_2 + b_3),
\end{aligned}$$

and for C_N, we evaluate the expression for $t=1$, which gives us

$$\begin{aligned}
C''(1) &= -6(1)b_1 + 6(1)b_0 + 6(1)(b_2-b_1) - 6(0)(b_2-b_1) + 6(0)b_3 - 6(0)b_2 \\
&= -6b_1 + 6b_0 + 6b_2 - 6b_1 \\
&= 6(b_0 - 2b_1 + b_2).
\end{aligned}$$

Given the linear system

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

and

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

we can substitute d_0 and d_N into the following equations

$$\begin{aligned}
\frac{7}{2}d_1 + d_2 &= 6x_1 - \frac{3}{2}d_0 \\
d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_{N-1} - \frac{3}{2}d_N.
\end{aligned}$$

As a result,

$$\begin{aligned}
\frac{7}{2}d_1 + d_2 &= 6x_1 - \frac{3}{2}\left(\frac{2}{3}x_0 + \frac{1}{3}d_1\right) \\
\frac{7}{2}d_1 + d_2 &= 6x_1 - x_0 - \frac{1}{2}d_1 \\
4d_1 + d_2 &= 6x_1 - x_0
\end{aligned}
\qquad
\begin{aligned}
d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_1 - \frac{3}{2}\left(\frac{1}{3}d_{N-1} + \frac{2}{3}x_N\right) \\
d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_1 - \frac{1}{2}d_{N-1} - x_N \\
d_{N-2} + 4d_{N-1} &= 6x_1 - x_N.
\end{aligned}$$

Therefore, the original system becomes

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

PROBLEM 2

Given the linear system

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

and

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

we can substitute d_0 and d_N into the following equations

$$\begin{aligned} \frac{7}{2}d_1 + d_2 &= 6x_1 - \frac{3}{2}d_0 \\ d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_{N-1} - \frac{3}{2}d_N. \end{aligned}$$

As a result,

$$\begin{aligned} \frac{7}{2}d_1 + d_2 &= 6x_1 - \frac{3}{2}\left(d_1 + \frac{2}{3}x_0 - \frac{2}{3}x_1\right) \\ \frac{7}{2}d_1 + d_2 &= 6x_1 - \frac{3}{2}d_1 - x_0 + x_1 \\ 5d_1 + d_2 &= 7x_1 - x_0 \end{aligned}$$

$$\begin{aligned} d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_1 - \frac{3}{2}\left(d_{N-1} + \frac{2}{3}x_N - \frac{2}{3}x_{N-1}\right) \\ d_{N-2} + \frac{7}{2}d_{N-1} &= 6x_1 - \frac{3}{2}d_{N-1} - x_N + x_{N-1} \\ d_{N-2} + 5d_{N-1} &= 7x_{N-1} - x_N \end{aligned}$$

Therefore, the original system becomes

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

PROBLEM 3

We want to show that

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

By definition, m_i is the vector tangent to the point x_i. We know, then, that $m_0 = C'(0)$. Thus, we may proceed as follows:

$$\begin{aligned}
C(t) &= \frac{(2-t)^2}{4}x_0 + \frac{(2-t)t}{2}b_1 + \frac{t^2}{4}x_2 \\
C'(t) &= \frac{(2t-4)}{4}x_0 - \frac{t}{2}b_1 + \frac{(2-t)}{2}b_1 + \frac{t}{2}x_2 \\
C'(0) &= \frac{(2(0)-4)}{4}x_0 - \frac{(0)}{2}b_1 + \frac{(2-(0))}{2}b_1 + \frac{(0)}{2}x_2 \\
&= -x_0 - 0 + b_1 + 0 \\
&= b_1 - x_0.
\end{aligned}$$

From a previous result, we know that $b_1 = -\frac{1}{2}x_0 + 2x_1 - \frac{1}{2}x_2$. Substituting this expression in for b_1 of our computed value of $C'(0)$, we acquire

$$\begin{aligned}
m_0 &= b_1 - x_0 \\
&= (-\frac{1}{2}x_0 + 2x_1 - \frac{1}{2}x_2) - x_0 \\
&= -\frac{3}{2}x_0 + 2x_1 - \frac{1}{2}x_2,
\end{aligned}$$

as desired.
Given that $d_0 = x_0 + \frac{1}{3}m_0$, we can expand using the above result,

$$\begin{aligned}
d_0 &= x_0 + \frac{1}{3}m_0 \\
&= x_0 + \frac{1}{3}(-\frac{3}{2}x_0 + 2x_1 - \frac{1}{2}x_2) \\
&= x_0 - \frac{1}{2}x_0 + \frac{2}{3}x_1 - \frac{1}{6}x_2 \\
&= \frac{1}{2}x_0 + \frac{1}{2}(\frac{4}{3}x_1 - \frac{1}{3}x_2) \\
&= \frac{1}{2}x_0 + \frac{1}{2}(x_1 + \frac{1}{3}x_1 + \frac{1}{3}x_2) \\
&= \frac{1}{2}x_0 + \frac{1}{2}(x_1 + \frac{1}{3}(x_1 - x_2))
\end{aligned}$$

and we arrive at the desired expression.
Now we wish to show that

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

Similar to the above, we proceed as follows:

$$\begin{aligned}
C(t) &= \frac{(2-t)^2}{4}x_{N-1} + \frac{(2-t)t}{2}b_N + \frac{t^2}{4}x_N \\
C'(t) &= \frac{(2t-4)}{4}x_{N-1} - \frac{t}{2}b_N + \frac{(2-t)}{2}b_N + \frac{t}{2}x_N \\
C'(2) &= \frac{(2(2)-4)}{4}x_{N-1} - \frac{(2)}{2}b_N + \frac{(2-(2))}{2}b_N + \frac{(2)}{2}x_N \\
&= \frac{0}{4}x_{N-1} - \frac{2}{2}b_N + \frac{0}{2}b_N + \frac{2}{2}x_N \\
&= 0 - b_1 + 0 + x_N \\
&= x_N - b_N.
\end{aligned}$$

We are given that $b_N = -\frac{1}{2}x_{N-2} + 2x_{N-1} - \frac{1}{2}x_N$. Substituting this expression in for b_1 of our computed value of $C'(2)$, we acquire

$$\begin{aligned}
m_N &= x_N - b_N \\
&= x_N - (-\frac{1}{2}x_{N-2} + 2x_{N-1} - \frac{1}{2}x_N) \\
&= \frac{1}{2}x_{N-2} - 2x_{N-1} + \frac{3}{2}x_N,
\end{aligned}$$

as desired.
Given that $d_N = x_N - \frac{1}{3}m_N$, we can expand using the above result,

$$\begin{aligned}
d_N &= x_N - \frac{1}{3}m_N \\
&= x_N - \frac{1}{3}(\frac{1}{2}x_{N-2} - 2x_{N-1} + \frac{3}{2}x_N) \\
&= x_N - \frac{1}{6}x_{N-2} + \frac{2}{3}x_{N-1} - \frac{1}{2}x_N \\
&= \frac{1}{2}(\frac{4}{3}x_{N-1} - \frac{1}{3}x_{N-2}) + \frac{1}{2}x_N \\
&= \frac{1}{2}(x_{N-1} + \frac{1}{3}(x_{N-1} - x_{N-2})) + \frac{1}{2}x_N,
\end{aligned}$$

and we arrive at the desired expression.

PROBLEM 4

First, we want to show that the third derivation at b_0 and b_3 of a Bezier cubic specified by the control points (b_0, b_1, b_2, b_3) is $6(-b_0 + 3b_1 - 3b_1 + b_3)$. We begin with the function $C(t)$.

$$\begin{aligned}
C(t) &= (1-t)^3 b_0 + 3(1-t)^2 b_1 + 3(1-t)t^2 b_2 + t^3 b_3 \\
C'(t) &= 3(1-t)^2(-1)b_0 + 3[2(1-t)(-1)t + (1-t)^2]b_1 + 3[(-1)t^2 + (2)(1-t)t]b_2 + 3t^2 b_3 \\
&= -3(1-t)^2 b_0 - 6(1-t)tb_1 + 3(1-t)^2 b_1 - 3t^2 b_2 + 6(1-t)tb_2 + 3t^2 b_3 \\
C''(t) &= 3(2)(1-t)b_0 - 6[(-1)t + (1-t)]b_1 - 3(2)(1-t)b_1 - 3(2)tb_2 + 6[-t + (1-t)]b_2 + 6tb_3 \\
&= 6(1-t)b_0 + 6tb_1 - 6(1-t)b_1 - 6(1-t)b_1 - 6tb_2 - 6tb_2 + 6(1-t)b_2 + 6tb_3 \\
C'''(t) &= 6(-1)b_0 + 6b_1 - 6(-1)b_1 - 6(-1)b_1 - 6b_2 - 6b_2 + 6(-1)b_2 + 6b_2 \\
&= 6(-b_0 + b_1 + b_1 + b_1 - b_2 - b_2 - b_2 + b_3) \\
&= 6(-b_0 + 3b_1 - 3b_2 + b_3)
\end{aligned}$$

Now, we wish to show that when $N = 3$, there is no need to solve a linear system, and that the points d_0, d_1, d_2, d_3 are given in terms of x_0, x_1, x_2, x_3.
It is given that $C_1'''(1) = C_2'''(0)$. So we have

$$\begin{aligned}
6(-b_0^1 + 3b_1^1 - 3b_2^1 + b_3^1) &= 6(-b_0^2 + 3b_1^2 - 3b_2^2 + b_3^2) \\
-b_0^1 + 3b_1^1 - 3b_2^1 + b_3^1 &= -b_0^2 + 3b_1^2 - 3b_2^2 + b_3^2.
\end{aligned}$$

Using the set of equations from page 3 of the project instructions, we can substitute in for the b_i^j and solve for d_0.

$$\begin{aligned}
-(x_0)+3(d_0)-3(\frac{1}{2}d_0+\frac{1}{2}d_1)+x_1 &= -(x_1)+3(\frac{2}{3}d_1+\frac{1}{3}d_2)-3(\frac{1}{3}d_1+\frac{2}{3}d_2)+x_2\\
-x_0+\frac{3}{2}d_0-\frac{3}{2}d_1+x_1 &= -x_1+d_1-d_2+x_2\\
\frac{3}{2}d_0 &= x_0-2x_1+x_2+\frac{3}{2}d_1+d_1-d_2\\
d_0 &= \frac{2}{3}x_0-\frac{4}{3}x_1+\frac{2}{3}x_2+\frac{5}{3}d_1-\frac{2}{3}d_2
\end{aligned}$$

From the original linear system, we obtain

$$\begin{aligned}
\frac{7}{2}d_1+d_2 &= 6x_1-\frac{3}{2}d_0\\
\frac{7}{2}d_1 &= 6x_1-\frac{3}{2}d_0-d_2\\
d_1 &= (\frac{2}{7})6x_1-(\frac{2}{7})\frac{3}{2}d_0-(\frac{2}{7})d_2\\
&= \frac{12}{7}x_1-\frac{3}{7}d_0-\frac{2}{7}d_2\\
\frac{5}{3}d_1 &= (\frac{5}{3})\frac{12}{7}x_1-(\frac{5}{3})\frac{3}{7}d_0-(\frac{5}{3})\frac{2}{7}d_2\\
\frac{5}{3}d_1 &= \frac{20}{7}x_1-\frac{5}{7}d_0-\frac{10}{21}d_2
\end{aligned}$$

and can substitute this into the above equation for d_0. This yields

$$\begin{aligned}
d_0 &= \frac{2}{3}x_0-\frac{4}{3}x_1+\frac{2}{3}x_2+(\frac{20}{7}x_1-\frac{5}{7}d_0-\frac{10}{21}d_2)-\frac{2}{3}d_2\\
\frac{12}{7}d_0 &= \frac{2}{3}x_0-\frac{32}{21}x_1+\frac{2}{3}x_2-\frac{8}{7}d_2\\
d_0 &= (\frac{7}{12})\frac{2}{3}x_0-(\frac{7}{12})\frac{32}{21}x_1+(\frac{7}{12})\frac{2}{3}x_2-(\frac{7}{12})\frac{8}{7}d_2\\
&= \frac{7}{18}x_0+\frac{8}{9}x_1+\frac{7}{18}x_2-\frac{2}{3}d_2,
\end{aligned}$$

which is the desired equation of d_0. This equation can then be substituted into the one we acquired from the original linear system

$$\begin{aligned}
d_1 &= \frac{12}{7}x_1-\frac{3}{7}d_0-\frac{2}{7}d_2\\
&= \frac{12}{7}x_1-\frac{3}{7}(\frac{7}{18}x_0+\frac{8}{9}x_1+\frac{7}{18}x_2-\frac{2}{3}d_2)-\frac{2}{7}d_2\\
&= \frac{12}{7}x_1-\frac{1}{6}x_0-\frac{8}{21}x_1-\frac{1}{6}x_2+\frac{2}{7}d_2-\frac{2}{7}d_2\\
&= -\frac{1}{6}x_0+\frac{4}{3}x_1-\frac{1}{6}x_2
\end{aligned}$$

which gives us the desired equation for d_1. One may note that the process for acquiring d_2 and d_3 is symmetric (Prof. Gallier said that it was alright to point this out and not show the work), and we arrive at the following system:

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

Since d_1 and d_2 are computed entirely from the x_i, and d_0 and d_3 depend on the x_i, d_1, and d_2, it is obvious that we need not solve any linear system when $N = 3$.

Using a similar process, we seek to find the system of equations for when $N = 4$. To begin, we note that the equations which were used to find d_0 and d_N in the previous part of the problem are still valid for this curve because the endpoints will be computed the same way. Similarly, the second and second to last points are calculated in the same manner. Thus, we know that

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

and we have only to conclude that

$$d_2 = \frac{3}{2}x_2 - \frac{1}{4}d_1 - \frac{1}{4}d_3.$$

To do so, we refer back to the linear system on page 4 of the project instructions. From this, we can obtain

$$\begin{aligned}
d_1 + 4d_2 + d_3 &= 6x_2 \\
4d_2 &= 6x_2 - d_1 - d_3 \\
d_2 &= \frac{3}{2}x_2 - \frac{1}{4}d_1 - \frac{1}{4}d_3,
\end{aligned}$$

as required.

When $N \geq 5$, the process for calculating d_0, d_1, d_{N-1}, and d_N is the same as the above, so we have

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

The remaining values $d_2, \ldots, d_{N-2}$ are derived from the original linear system given on page 3 of the project instructions – the same equation we used to derive d_2 in the above part. Instead of deriving d_2 directly, we reorganize the expression to obtain

$$\begin{aligned}
d_1 + 4d_2 + d_3 &= 6x_2 \\
4d_2 + d_3 &= 6x_2 - d_1 \\
&= 6x_2 - \left(-\frac{1}{6}x_0 + \frac{4}{3}x_1 - \frac{1}{6}x_2\right) \\
&= 6x_2 + \frac{1}{6}x_0 - \frac{4}{3}x_1 + \frac{1}{6}x_2,
\end{aligned}$$

which gives us the correct solution to the linear system. The process is symmetric for d_{N-2}. The remaining entries are derived directly from the original system, as we will obtain

$$d_i + 4d_{i+1} + d_{i+2} = 6x_{i+1}$$

for $2 \le i \le (N-2)$.
Because the system is represented by an $(N-3)\times(N-3)$ matrix, when $N=5$, the given linear system is reduced to the 2×2 system formed by the first and last columns of the original:

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

PROBLEM 5

Executing `project_3.m` launches a graphical interface where the user can click a series of points which will interpolate a b-spline curve. The first curve is drawn after at least 5 points are clicked and redrawn for every subsequent point. With every point added, the tridiagonal system $A * d = x$ expands by 1 and must be solved again. The program solves the system at each iteration with a call to `solvetri.m`. This function solves the tridiagonal system using both Gaussian elimination and LU decomposition, and outputs the time each process takes to the console.

It is indeed true that Gaussian elimination does not require any pivoting when solving interpolation problems.

PROBLEM 6

Executing `project_3.m` launches a graphical interface where the user can click a series of points which will interpolate a b-spline curve. The first curve is drawn after at least 5 points are clicked and redrawn for every subsequent point. With every point added, the tridiagonal system $A * d = x$ expands by 1 and must be solved again. The program solves the system at each iteration with a call to `solvetri.m`. This function solves the tridiagonal system using both Gaussian elimination and LU decomposition, and outputs the time each process takes to the console.

When the dimension of A is small, the system is solved faster using Gaussian elimination than using LU decomposition. However, as the system grows larger the complexity of Gaussian elimination causes it to take much longer than LU decomposition. The user can verify this by clicking more points to add to the system and observing the increasing computation time of Gaussian elimination.

Using alternative tridiagonal systems with large diagonal entries, we can also see that LU factorization is more numerically stable than Gaussian elimination.

PROBLEM 7

Executing `project_3.m` launches a graphical interface where the user can click a series of points which will interpolate a b-spline curve. The first curve is drawn after at least 5 points are clicked and redrawn for every subsequent point. With every point added, the tridiagonal system $A * d = x$ expands by 1 and must be solved again. The program solves the system at each iteration with a call to `solvetri.m`. This function solves the tridiagonal system using both Gaussian elimination and LU decomposition, and outputs the time each process takes to the console.

The main function `project_3.m` calls the function `curveinterp.m` to build and solve the tridiagonal system with various end conditions. Modifying the `econd` parameter in `project_3.m`, changes which end condition `curveinterp.m` will use. This parameter can be set to `natural`, `quadratic`, `bessel`, and `knot`. The user will see the effects of each choice of the parameter as the curve is drawn.