Fitting a Natural Spline to Samples of the Form $(t, \mathbf{f}(t))$

David Eberly
Geometric Tools, LLC
http://www.geometrictools.com/

Created: January 5, 2008
Last Modified: February 14, 2008

Contents

1 Introduction

Consider a function $\mathbf{f} : D \to \mathbb{R}^m$, whose domain is $D = [t_{\min}, t_{\max}]$ and whose range consists of m-tuples with $m \geq 1$. The only information we have about the function is a set of samples, $\{(t_i, \mathbf{f}_i)\}_{i=0}^n$, where $\mathbf{f}_i = \mathbf{f}(t_i)$ and $t_{\min} \leq t_0 < t_1 < \cdots < t_{n-1} < t_n \leq t_{\max}$. We want to fit the function with a piecewise cubic polynomial spline

$$S(t) = \left\{ \begin{array}{ll} S_0(t), & t \in [t_0, t_1] \\ S_1(t), & t \in [t_1, t_2] \\ \vdots & \\ S_{n-2}(t), & t \in [t_{n-2}, t_{n-1}] \\ S_{n-1}(t), & t \in [t_{n-1}, t_n] \end{array} \right\} \tag{1}$$

where

$$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3 \tag{2}$$

for $0 \leq i \leq n - 1$. Moreover, we want $S(t)$ to be a C^2 function; that is, $S(t)$, $S'(t)$, and $S''(t)$ must be continuous functions of t on the interval $[t_0, t_n]$. Each of $S_i(t)$ is a C^2 function on the open interval (t_i, t_{i+1}), so the only points of concern to be C^2 everywhere are the sample points. We have n polynomials to determine, each having 4 unknown coefficients. Thus, we have $4n$ unknowns. Imposing the C^2 condition leads to linear constraints on the unknowns.

We need the polynomials, their first derivatives, and their second derivatives to match at the sample points t_1 through t_{n-1}. Specifically,

$$S_i(t_{i+1}) = S_{i+1}(t_i), \;\; S_i'(t_{i+1}) = S_{i+1}'(t_i), \;\; S_i''(t_{i+1}) = S_{i+1}''(t_i), \;\; 0 \leq i \leq n - 2 \tag{3}$$

This is a set of $3(n - 1)$ constraints on the unknowns. We also need the polynomials to pass through the sample points, so

$$S_i(t_i) = f_i, \;\; 0 \leq i \leq n - 1, \;\; \text{and} \;\; S_{n-1}(t_n) = f_n \tag{4}$$

This is a set of $n + 1$ constraints on the unknowns. Equations (3) and (4) represent $4n - 2$ constraints on the $4n$ unknowns. We have two degrees of freedom. Three standard choices are the following.

1.1 Constraints for Free Splines

We may allow the endpoints to have whatever slope necessary by specifying the second derivatives at those points to be zero:

$$S_0''(t_0) = 0, \;\; S_{n-1}''(t_n) = 0 \tag{5}$$

The resulting curve is said to be a *free spline*. These two constraints and the previously mentioned ones give us $4n$ linear equations in $4n$ unknowns; that is, the coefficients of the cubic polynomials are obtained by solving a linear system of equations. It turns out that the system is tridiagonal, so a specialized linear solver may be used that is $O(n)$. General linear solvers are $O(n^3)$, so the specialized solver saves computational time for a large number of sample points.

1.2 Constraints for Clamped Splines

We may clamp the endpoints in the sense that we specify the first derivatives at those points:

$$S_0'(t_0) = \sigma_0, \quad S_{n-1}'(t_n) = \sigma_1 \tag{6}$$

for user-specified constants σ_0 and σ_1. The resulting curve is said to be a *clamped spline*. These two constraints and the previously mentioned ones give us $4n$ linear equations in $4n$ unknowns. As for the case of free splines, the linear system is tridiagonal and may be solved by a specialized linear solver.

1.3 Constraints for Periodic Splines

We may require that the first and last samples have the same function values, namely, that $\mathbf{f}_n = \mathbf{f}_0$. Moreover, the C^2 condition requires us to impose the derivative equality constraints

$$S_0'(t_0) = S_{n-1}'(t_n), \quad S_0''(t_0) = S_{n-1}''(t_n) \tag{7}$$

The resulting curve is said to be a *periodic spline*. These two constraints and the previously mentioned ones produce a system of $4n$ linear equations in $4n$ unknowns, but this system is not tridiagonal. However, the system may be solved by a specialized linear solver that is $O(n)$.

2 Constructing Free Splines

This section describes how to set up and solve the tridiagonal linear system of equations for the polynomial coefficients when the endpoints are required to have zero second-order derivatives.

The ratio of differences of function values and time values occurs often in the derivation, so let us first define

$$q_i = \frac{f_{i+1} - f_i}{\Delta_i} \tag{8}$$

Define $\Delta_i = t_{i+1} - t_i$ for all relevant i. The conditions $S_i(t_{i+1}) = S_{i+1}(t_i)$ and $S_i(t_i) = f_i$ are equivalent to

$$\Delta_i b_i + \Delta_i^2 c_i + \Delta_i^3 d_i = f_{i+1} - f_i \tag{9}$$

The conditions $S_i'(t_{i+1}) = S_{i+1}'(t_i)$ are equivalent to

$$2\Delta_i c_i + 3\Delta_i^2 d_i = b_{i+1} - b_i \tag{10}$$

The conditions $S_i''(t_{i+1}) = S_{i+1}''(t_i)$ are equivalent to

$$3\Delta_i d_i = c_{i+1} - c_i \tag{11}$$

Substitute Equation (11) into Equations (9) and (10) to obtain

$$\Delta_i b_i + \Delta_i^2 \left(\frac{c_{i+1} + 2c_i}{3} \right) = f_{i+1} - f_i \tag{12}$$

and

$$\Delta_i \left(c_{i+1} + c_i \right) = b_{i+1} - b_i \tag{13}$$

Increment the index in Equation (12) to obtain

$$\Delta_{i+1} b_{i+1} + \Delta_{i+1}^2 \left(\frac{c_{i+2} + 2c_{i+1}}{3} \right) = f_{i+2} - f_{i+1} \tag{14}$$

Compute Δ_i times Equation (14) and subtract from it Δ_{i+1} times Equation (12) to obtain

$$\begin{aligned} &\Delta_i \Delta_{i+1} (b_{i+1} - b_i) + \Delta_i \Delta_{i+1}^2 \left(\tfrac{c_{i+2}+2c_{i+1}}{3} \right) - \Delta_{i+1} \Delta_i^2 \left(\tfrac{c_{i+1}+2c_i}{3} \right) \\ &= \Delta_i (f_{i+2} - f_{i+1}) - \Delta_{i+1} (f_{i+1} - f_i) \end{aligned} \tag{15}$$

Substitute Equation (13) into Equation (15), collect terms, and divide by $\Delta_i \Delta_{i+1}$ to obtain

$$\Delta_{i+1} c_{i+2} + 2(\Delta_{i+1} + \Delta_i) c_{i+1} + \Delta_i c_i = 3(q_{i+1} - q_i) \tag{16}$$

which is defined for $0 \le i \le n-3$. We know that the constraint $S_0''(t_0) = 0$ in Equation (5) implies $c_0 = 0$. The constraint $S_{n-1}''(t_n)$ in Equation (5) implies $\Delta_{n-1} c_{n-1} + 3\Delta_{n-1}^2 d_{n-1} = 0$. If we define $c_n = 0$, this constraint fits into the construction here and Equation (16) also holds for $i = n-2$. Essentially, we are thinking of the spline curve having one additional segment $S_n(t) = f_n + b_n(t - t_n)$, a line segment, so that $c_n = d_n = 0$.

Equation (16) and boundary conditions $c_0 = c_n = 0$ may be solved by setting up a linear system

$$M\mathbf{c} = \mathbf{p} \tag{17}$$

where the $(n-1) \times (n-1)$ matrix M is tridiagonal, symmetric, and diagonally dominant:

$$M = \begin{bmatrix} 2(\Delta_0 + \Delta_1) & \Delta_1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \Delta_1 & 2(\Delta_1 + \Delta_2) & \Delta_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta_2 & 2(\Delta_2 + \Delta_3) & \Delta_3 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \Delta_{n-4} & 2(\Delta_{n-4} + \Delta_{n-3}) & \Delta_{n-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \Delta_{n-3} & 2(\Delta_{n-3} + \Delta_{n-2}) & \Delta_{n-2} \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \Delta_{n-2} & 2(\Delta_{n-2} + \Delta_{n-1}) \end{bmatrix} \tag{18}$$

The $(n-1) \times 1$ column vector $\mathbf{c}$ has components c_1 through c_{n-1} and the $(n-1) \times 1$ column vector $\mathbf{p}$ has components p_1 through p_{n-1} where

$$p_i = 3(q_i - q_{i-1}) \tag{19}$$

The matrix M and vector $\mathbf{p}$ may be computed and then used as input to a linear system solver that is specialized for tridiagonal systems. The solver computes c_1 through c_{n-1}, and we already know that $c_0 = 0$. The d-terms are computed from Equation (11),

$$d_i = \frac{c_{i+1} - c_i}{3\Delta_i} \tag{20}$$

and the b-terms are computed from Equation (12),

$$b_i = q_i - \Delta_i \left(\frac{c_{i+1} + 2c_i}{3} \right) \tag{21}$$

The a-terms are already known,

$$a_i = f_i \tag{22}$$

3 Constructing Clamped Splines

This section describes how to set up and solve the tridiagonal linear system of equations for the polynomial coefficients when the endpoints are required to have user-specified first-order derivatives.

Multiply Equation (9) by $3/\Delta_i$ and subtract from it Equation (10) to obtain

$$3b_i + \Delta_i c_i = 3q_i - (b_{i+1} - b_i) \tag{23}$$

Increment the index in Equation (23) to obtain

$$3b_{i+1} + \Delta_{i+1} c_{i+1} = 3q_{i+1} - (b_{i+2} - b_{i+1}) \tag{24}$$

Multiply Equation (23) by Δ_{i+1}, multiply Equation (24) by Δ_i and add the two together to obtain

$$3(\Delta_{i+1} b_i + \Delta_i b_{i+1}) + \Delta_{i+1}\Delta_i(c_{i+1} + c_i) = 3(\Delta_{i+1} q_i + \Delta_i q_{i+1}) - \Delta_{i+1}(b_{i+1} - b_i) - \Delta_i(b_{i+2} - b_{i+1}) \tag{25}$$

In the left-hand side of Equation (25), substitute Equation (13) to eliminate the c-terms. Some algebra simplifies this to

$$\Delta_i b_{i+1} + 2(\Delta_i + \Delta_{i+1}) b_{i+1} + \Delta_{i+1} b_i = 3(\Delta_{i+1} q_i + \Delta_i q_{i+1}) \tag{26}$$

which is defined for $0 \le i \le n-3$. We know that the constraint $S_0'(t_0) = \sigma_0$ in Equation (6) implies $b_0 = \sigma_0$. The constraint $S_{n-1}'(t_n) = \sigma_1$ in Equation (6) implies $\Delta_{n-1} b_{n-1} + 2\Delta_{n-1}^2 c_{n-1} + 3\Delta_{n-1}^3 d_{n-1} = \sigma_1$. If we define $b_n = \sigma_1$, this constraint fits into the construction here and Equation (26) also holds for $i = n-2$. Essentially, we are thinking of the spline curve having one additional segment $S_n(t) = f_n + b_n(t - t_n) + c_n(t - t_n)^2$, a parabolic arc, so that $2c_n = 2\Delta_{n-1} c_{n-1} + 6\Delta_{n-1}^2 d_{n-1}$ and $d_n = 0$.

Equation (26) and boundary conditions $b_0 = \sigma_0$ and $b_n = \sigma_1$ may be solved by setting up a linear system

$$N\mathbf{b} = \mathbf{r} \tag{27}$$

where the $(n-1) \times (n-1)$ matrix N is tridiagonal, symmetric, and diagonally dominant:

$$N = \begin{bmatrix}
2(\Delta_0 + \Delta_1) & \Delta_0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
\Delta_2 & 2(\Delta_1 + \Delta_2) & \Delta_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & \Delta_3 & 2(\Delta_2 + \Delta_3) & \Delta_2 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & \Delta_{n-3} & 2(\Delta_{n-4} + \Delta_{n-3}) & \Delta_{n-4} & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \Delta_{n-2} & 2(\Delta_{n-3} + \Delta_{n-2}) & \Delta_{n-3} \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \Delta_{n-1} & 2(\Delta_{n-2} + \Delta_{n-1})
\end{bmatrix} \tag{28}$$

The $(n-1) \times 1$ column vector $\mathbf{b}$ has components b_1 through b_{n-1} and the $(n-1) \times 1$ column vector $\mathbf{r}$ has components r_1 through r_{n-1} where

$$\begin{aligned}
r_1 &= 3(\Delta_1 q_0 + \Delta_0 q_1) - \Delta_1 \sigma_0 \\
r_i &= 3(\Delta_i q_{i-1} + \Delta_{i-1} q_i), \quad 2 \le i \le n-2 \\
r_{n-1} &= 3(\Delta_{n-1} q_{n-2} + \Delta_{n-2} q_{n-1}) - \Delta_{n-2} \sigma_1
\end{aligned} \tag{29}$$

The matrix N and vector $\mathbf{r}$ may be computed and then used as input to a linear system solver that is specialized for tridiagonal systems. The solver computes b_1 through b_{n-1}. The c-terms are computed from Equation (23),

$$c_i = \frac{3q_i - b_{i+1} - 2b_i}{\Delta_i} \tag{30}$$

and the d-terms are computed from Equation (10),

$$d_i = \frac{b_{i+1} - b_i - 2\Delta_i c_i}{3\Delta_i^2} \tag{31}$$

The a-terms are already known,

$$a_i = f_i \tag{32}$$

4 Constructing Periodic Splines

This section describes how to set up and solve the linear system of equations for the polynomial coefficients when the endpoints are required to have the same function value, same first-order derivative, and same second-order derivative.

4.1 Three Samples

Consider the simplest case of three samples: $(t_0, \mathbf{f}_0)$, $(t_1, \mathbf{f}_1)$, and $(t_2, \mathbf{f}_0)$. The two cubic polynomials are $S_0(t) = a_0 + b_0(t - t_0) + c_0(t - t_0)^2 + d_0(t - t_0)^3$ and $S_1(t) = a_1 + b_1(t - t_1) + c_1(t - t_1)^2 + d_1(t - t_1)^3$. The eight constraints are

$$\begin{aligned}
&\mathbf{f}_0 = S_0(t_0) = S_1(t_2) && \mathbf{f}_1 = S_0(t_1) = S_1(t_1) \\
&S_0'(t_0) = S_1'(t_2) && S_0'(t_1) = S_1'(t_1) \\
&S_0''(t_0) = S_1''(t_2) && S_0''(t_1) = S_1''(t_1)
\end{aligned} \tag{33}$$

Define $\Delta_0 = t_1 - t_0$ and $\Delta_1 = t_2 - t_1$. In terms of the polynomial coefficients, the constraints are

$$\begin{aligned}
&a_0 + b_0\Delta_0 + c_0\Delta_0^2 + d_0\Delta_0^3 = a_1 = f_1 && a_1 + b_1\Delta_1 + c_1\Delta_1^2 + d_1\Delta_1^3 = a_0 = f_0 \\
&b_0 + 2c_0\Delta_0 + 3d_0\Delta_0^2 = b_1 && b_1 + 2c_1\Delta_1 + 3d_1\Delta_1^2 = b_0 \\
&2c_0 + 6d_0\Delta_0 = 2c_1 && 2c_1 + 6d_1\Delta_1 = 2c_0
\end{aligned} \tag{34}$$

The linear system of equations is

$$\left[\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \Delta_0 & \Delta_0^2 & \Delta_0^3 & -1 & 0 & 0 & 0 \\
0 & 1 & 2\Delta_0 & 3\Delta_0^2 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 3\Delta_0 & 0 & 0 & -1 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 1 & \Delta_1 & \Delta_1^2 & \Delta_1^3 \\
0 & -1 & 0 & 0 & 0 & 1 & 2\Delta_1 & 3\Delta_1^2 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & 3\Delta_1
\end{array}\right]
\left[\begin{array}{c}
a_0 \\ b_0 \\ c_0 \\ d_0 \\ \hline a_1 \\ b_1 \\ c_1 \\ d_1
\end{array}\right]
=
\left[\begin{array}{c}
f_0 \\ 0 \\ 0 \\ 0 \\ \hline f_1 \\ 0 \\ 0 \\ 0
\end{array}\right] \tag{35}$$

In block matrix form,

$$\left[\begin{array}{c|c} M_0 & -L \\ \hline -L & M_1 \end{array}\right] \left[\begin{array}{c} \mathbf{K}_0 \\ \hline \mathbf{K}_1 \end{array}\right] = \left[\begin{array}{c} f_0\mathbf{U} \\ \hline f_1\mathbf{U} \end{array}\right] \tag{36}$$

where

$$M_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \Delta_i & \Delta_i^2 & \Delta_i^3 \\ 0 & 1 & 2\Delta_i & 3\Delta_i^2 \\ 0 & 0 & 1 & 3\Delta_0 \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{K}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{37}$$

Notice that M_i is invertible.

The linear system may be solved by row-reducing the augmented block matrix as shown next. The matrix I is the 4×4 identity matrix.

$$\begin{aligned}
\left[\begin{array}{cc|c} M_0 & -L & f_0\mathbf{U} \\ -L & M_1 & f_1\mathbf{U} \end{array}\right] &\sim \left[\begin{array}{cc|c} I & -M_0^{-1}L & f_0M_0^{-1}\mathbf{U} \\ -L & M_1 & f_1\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{cc|c} I & -M_0^{-1}L & f_0M_0^{-1}\mathbf{U} \\ 0 & M_1 - LM_0^{-1}L & (f_1I + f_0LM_0^{-1})\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{cc|c} I & -M_0^{-1}L & f_0M_0^{-1}\mathbf{U} \\ 0 & I & (M_1 - LM_0^{-1}L)^{-1}(f_1I + f_0LM_0^{-1})\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{cc|c} I & 0 & [f_0M_0^{-1} + M_0^{-1}L(M_1 - LM_0^{-1}L)^{-1}(f_1I + f_0LM_0^{-1})]\mathbf{U} \\ 0 & I & (M_1 - LM_0^{-1}L)^{-1}(f_1I + f_0LM_0^{-1})\mathbf{U} \end{array}\right] \\
&= \left[\begin{array}{cc|c} I & 0 & \mathbf{K}_0 \\ 0 & I & \mathbf{K}_1 \end{array}\right]
\end{aligned} \tag{38}$$

where

$$\mathbf{K}_1 = (M_1 - LM_0^{-1}L)^{-1}(f_1I + f_0LM_0^{-1}), \quad \mathbf{K}_0 = f_0M_0^{-1}\mathbf{U} + M_0^{-1}L\mathbf{K}_1 \tag{39}$$

4.2 Four Samples

Similar to the case of three samples, the constraints may be formulated as a block-matrix system,

$$\left[\begin{array}{c|c|c} M_0 & -L & 0 \\ \hline 0 & M_1 & -L \\ \hline -L & 0 & M_2 \end{array}\right] \left[\begin{array}{c} \mathbf{K}_0 \\ \hline \mathbf{K}_1 \\ \hline \mathbf{K}_2 \end{array}\right] = \left[\begin{array}{c} f_0\mathbf{U} \\ \hline f_1\mathbf{U} \\ \hline f_2\mathbf{U} \end{array}\right] \tag{40}$$

The linear system may be solved by row-reducing the augmented block matrix,

$$\begin{aligned}
\left[\begin{array}{ccc|c} M_0 & -L & 0 & f_0\mathbf{U} \\ 0 & M_1 & -L & f_1\mathbf{U} \\ -L & 0 & M_2 & f_2\mathbf{U} \end{array}\right] &\sim \left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & f_1M_1^{-1}\mathbf{U} \\ -L & 0 & M_2 & f_2\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & f_1M_1^{-1}\mathbf{U} \\ 0 & -LM_0^{-1}L & M_2 & (f_2I + f_0LM_0^{-1})\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & f_1M_1^{-1}\mathbf{U} \\ 0 & 0 & M_2 - LM_0^{-1}LM_1^{-1}L & [f_2I + LM_0^{-1}(f_0I + f_1LM_1^{-1})]\mathbf{U} \end{array}\right] \\
&\sim \left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & f_1M_1^{-1}\mathbf{U} \\ 0 & 0 & I & \mathbf{K}_2 \end{array}\right]
\end{aligned} \tag{41}$$

where

$$\mathbf{K}_2 = (M_2 - LM_0^{-1}LM_1^{-1}L)^{-1}[f_2I + LM_0^{-1}(f_0I + f_1LM_1^{-1})]\mathbf{U} \tag{42}$$

Back substitution may now be used to compute $\mathbf{K}_0$ and $\mathbf{K}_1$. The first back substitution is

$$\left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & f_1M_1^{-1}\mathbf{U} \\ 0 & 0 & I & \mathbf{K}_2 \end{array}\right] \sim \left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & 0 & \mathbf{K}_1 \\ 0 & 0 & I & \mathbf{K}_2 \end{array}\right] \tag{43}$$

where

$$\mathbf{K}_1 = f_1M_1^{-1}\mathbf{U} + M_1^{-1}L\mathbf{K}_2 \tag{44}$$

The second back substitution is

$$\left[\begin{array}{ccc|c} I & -M_0^{-1}L & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & 0 & \mathbf{K}_1 \\ 0 & 0 & I & \mathbf{K}_2 \end{array}\right] \sim \left[\begin{array}{ccc|c} I & 0 & 0 & \mathbf{K}_0 \\ 0 & I & 0 & \mathbf{K}_1 \\ 0 & 0 & I & \mathbf{K}_2 \end{array}\right] \tag{45}$$

where

$$\mathbf{K}_0 = f_0M_0^{-1}\mathbf{U} + M_0^{-1}L\mathbf{K}_1 \tag{46}$$

Equations (46), (44), and (42) are the polynomial coefficients for the spline.

4.3 The General Case

In general, the constraints may be formulated as a block-matrix system,

$$\begin{bmatrix} M_0 & -L & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & M_1 & -L & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & M_{n-2} & -L \\ -L & 0 & 0 & 0 & \cdots & 0 & 0 & M_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{K}_0 \\ \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_{n-1} \end{bmatrix} = \begin{bmatrix} f_0\mathbf{U} \\ f_1\mathbf{U} \\ \vdots \\ f_{n-1}\mathbf{U} \end{bmatrix} \tag{47}$$

whose solution is obtained by row reducing the augmented block matrix

$$\left[\begin{array}{cccccccc|c} M_0 & -L & 0 & 0 & \cdots & 0 & 0 & 0 & f_0\mathbf{U} \\ 0 & M_1 & -L & 0 & \cdots & 0 & 0 & 0 & f_1\mathbf{U} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & M_{n-2} & -L & f_{n-2}\mathbf{U} \\ -L & 0 & 0 & 0 & \cdots & 0 & 0 & M_{n-1} & f_{n-1}\mathbf{U} \end{array}\right] \tag{48}$$

Inverting the M_i matrices in the first $n-1$ block rows leads to

$$\left[\begin{array}{cccccccc|c} I & -M_0^{-1}L & 0 & 0 & \cdots & 0 & 0 & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & 0 & \cdots & 0 & 0 & 0 & f_1M_1^{-1}\mathbf{U} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & I & -M_{n-2}^{-1}L & f_{n-2}M_{n-2}^{-1}\mathbf{U} \\ -L & 0 & 0 & 0 & \cdots & 0 & 0 & M_{n-1} & f_{n-1}\mathbf{U} \end{array}\right] \tag{49}$$

Forward elimination by the first $n-1$ block rows leads to

$$\left[\begin{array}{cccccccc|c} I & -M_0^{-1}L & 0 & 0 & \cdots & 0 & 0 & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & 0 & \cdots & 0 & 0 & 0 & f_1M_1^{-1}\mathbf{U} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & I & -M_{n-2}^{-1}L & f_{n-2}M_{n-2}^{-1}\mathbf{U} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & P & \mathbf{Q} \end{array}\right] \tag{50}$$

where

$$P = M_{n-1} - \left(LM_0^{-1}LM_1^{-1}\cdots LM_{n-2}^{-1}\right)L \tag{51}$$

and

$$\begin{aligned} \mathbf{Q} &= \left[f_{n-1}I + f_0LM_0^{-1} + f_1LM_0^{-1}LM_1^{-1} + \cdots + f_{n-2}\left(LM_0^{-1}LM_1^{-1}\cdots LM_{n-2}^{-1}\right)\right]\mathbf{U} \\ &= f_{n-1}\mathbf{U} + LM_0^{-1}\left[f_0\mathbf{U} + LM_1^{-1}\left(f_1\mathbf{U} + \cdots\right)\right] \end{aligned} \tag{52}$$

The last expression shows that $\mathbf{Q}$ may be computed efficiently in a nested manner.

Inverting P in the last row of the reduced augmented block matrix,

$$\left[\begin{array}{cccccccc|c} I & -M_0^{-1}L & 0 & 0 & \cdots & 0 & 0 & 0 & f_0M_0^{-1}\mathbf{U} \\ 0 & I & -M_1^{-1}L & 0 & \cdots & 0 & 0 & 0 & f_1M_1^{-1}\mathbf{U} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & I & -M_{n-2}^{-1}L & f_{n-2}M_{n-2}^{-1}\mathbf{U} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I & \mathbf{K}_{n-1} \end{array}\right] \tag{53}$$

where

$$\mathbf{K}_{n-1} = P^{-1}\mathbf{Q} \tag{54}$$

Back substitution may be applied $n-1$ times to solve for the other coefficients. Generally, these are

$$\mathbf{K}_i = f_iM_i^{-1}\mathbf{U} + M_i^{-1}L\mathbf{K}_{i+1}, \;\; n-2 \geq i \geq 0 \tag{55}$$

In the calculations, we need the inverses of the M_i matrices. These are

$$M_i^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-3}{\Delta_i} & \frac{3}{\Delta_i} & -2 & \Delta_i \\ \frac{3}{\Delta_i^2} & \frac{-3}{\Delta_i^2} & \frac{3}{\Delta_i} & -2 \\ \frac{-1}{\Delta_i^3} & \frac{1}{\Delta_i^3} & \frac{-1}{\Delta_i^2} & \frac{1}{\Delta_i} \end{bmatrix} \tag{56}$$

5 Implementations

See the files

```
GeometricTools/WildMagic4/LibFoundation/Curves/Wm4NaturalSpline1.{h,cpp}
```

for an implementation. A direct implementation of the method discussed here for periodic splines showed that the algorithm is ill-conditioned. However, a standard solver for linear systems was numerically stable. I have a TODO in the source code to look into implementing a numerically stable row-reduction method that still has the $O(n)$ asymptotic behavior.