

Algorithm for Constructing Piecewise Quintic Monotone Interpolating Splines

Thomas C.H. Lux

September 9, 2019

Forward

When provided data that has no assigned first and second derivative values, the derivative data is filled by a linear fit of neighboring data points. End points are set to be the slope between the end and its nearest neighbor.

The method for finding the transition point of a boolean function on a line is the Golden Section search. This will be referred to in pseudo code as `line_search(g, a, b)` where $a, b \in S$ for S closed under convex combination, $g : S \rightarrow \{0, 1\}$ is a boolean function, and $g(b) = 1$. If $g(a) = 1$ then a is returned, otherwise the smallest $c \in [0, 1]$ such that $g(a(1 - c) + cb) = 1$ is returned.

After assigning function values and derivative values, an interpolating function is constructed from a quintic B-spline basis.

1 Verifying Monotonicity of a Quintic Polynomial

Let f be a quintic polynomial over a closed interval $[x_0, x_1] \subset \mathbb{R}$. Now f is uniquely defined by the evaluation tuples $(x_0, f(x_0), f'(x_0), f''(x_0))$ and $(x_1, f(x_1), f'(x_1), f''(x_1))$. Assume without loss of generality that $f(x_0) < f(x_1)$, where the case of monotonic decreasing f would consider the negated the function values. The following algorithm will determine whether or not f is monotone increasing on the interval $[x_0, x_1]$.

Algorithm 1a: `is_monotone`

```
0: if ( $f'(x_0) = 0$  or  $f'(x_1) = 0$ ) return is_monotone_simplified
1: if ( $f'(x_0) < 0$  or  $f'(x_1) < 0$ ) return FALSE
```

This can be seen clearly from the fact that f is analytic; there will exist some nonempty interval about x_0 or x_1 for which f' is negative.

```
2:  $A = f'(x_0) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$ 
3:  $B = f'(x_1) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$ 
```

The variables A and B correspond directly to the theoretical foundation for positive quartic polynomials established in [3], first defined after equation 18.

$$\begin{aligned}
8: \gamma_0 &= 4 \frac{f'(x_0)}{f'(x_1)} (B/A)^{3/4} \\
9: \gamma_1 &= \frac{x_1 - x_0}{f'(x_1)} (B/A)^{3/4} \\
4: \alpha_0 &= 4(B/A)^{1/4} \\
5: \alpha_1 &= -\frac{x_1 - x_0}{f'(x_1)} (B/A)^{1/4} \\
6: \beta_0 &= 30 - \frac{12(f'(x_0) + f'(x_1))(x_1 - x_0)}{(f(x_1) - f(x_0))\sqrt{A}\sqrt{B}} \\
7: \beta_1 &= \frac{-3(x_1 - x_0)^2}{2(f(x_1) - f(x_0))\sqrt{A}\sqrt{B}}
\end{aligned}$$

The γ , α , and β terms with subscripts 0 and 1 are algebraic reductions of the simplified conditions for satisfying Theorem 2 in [3] (equation 16). These terms with subscripts 0 and 1 give the computation of α , β , and γ the form seen in lines 10-12 below.

$$\begin{aligned}
10: \gamma &= \gamma_0 + \gamma_1 f''(x_0) \\
11: \alpha &= \alpha_0 + \alpha_1 f''(x_1) \\
12: \beta &= \beta_0 + \beta_1 (f''(x_0) - f''(x_1)) \\
13: \text{if } (\beta \leq 6) \text{ then return } \alpha &> -(\beta + 2)/2 \\
14: \text{else return } \gamma &> -2\sqrt{\beta - 2}
\end{aligned}$$

Given the same initial conditions there are special circumstances which allow for the usage of simpler monotonicity conditions. In this case, consider when the quintic function has either $f'(x_0) = 0$ or $f'(x_1) = 0$. This reduces the problem of verifying monotonicity to one of cubics established by [2].

Algorithm 1b: `is_monotone_simplified`

$$\begin{aligned}
0: \alpha &= 30 - \frac{(x_1 - x_0)(14f'(x_0) + 16f'(x_1) - (f''(x_1) - f''(x_0))(x_1 - x_0))}{2(f(x_1) - f(x_0))} \\
1: \beta &= 30 - \frac{(x_1 - x_0)(2f'(x_0) + 24f'(x_1) - (f''(x_0) + 3f''(x_1))(x_1 - x_0))}{2(f(x_1) - f(x_0))} \\
2: \gamma &= \frac{(x_1 - x_0)(7f'(x_0) + f''(x_0)(x_1 - x_0))}{f(x_1) - f(x_0)} \\
3: \delta &= \frac{f'(x_0)(x_1 - x_0)}{f(x_1) - f(x_0)}
\end{aligned}$$

The variables above are algebraic expansions of the coefficients for the cubic derivative function in [2].

$$\begin{aligned}
4: \text{if } (\min(\alpha, \delta) < 0) \text{ return FALSE} \\
5: \text{else if } (\beta < \alpha - 2\sqrt{\alpha\delta}) \text{ return FALSE} \\
6: \text{else if } (\gamma < \delta - 2\sqrt{\alpha\delta}) \text{ return FALSE} \\
7: \text{else return TRUE}
\end{aligned}$$

Next the modification of a quintic spline to enforce monotonicity will be discussed.

2 Enforcing Monotonicity of a Quintic Polynomial

Algorithm 2a: `make_monotone`

```

0: if  $(f(x_1) - f(x_0) = 0)$  return  $f'(x_0) = f'(x_1) = f''(x_0) = f''(x_1) = 0$ 
1:  $f'(x_0) = \text{median}(0, f'(x_0), 14 \frac{f(x_1) - f(x_0)}{x_1 - x_0})$ 
2:  $f'(x_1) = \text{median}(0, f'(x_1), 14 \frac{f(x_1) - f(x_0)}{x_1 - x_0})$ 

```

This selection of values for $f'(x_0)$ and $f'(x_1)$ is suggested by [3] (originally from [1]), and quickly enforces upper and lower bounds on derivative values to ensure quintic monotonicity is obtainable.

```

3:  $A = f'(x_0) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$ 
4:  $B = f'(x_1) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$ 
5: if  $AB \leq 0$  return make_monotone_simplified
6: if  $(\max(A, B) > 6)$ 
    $f'(x_0) = 6f'(x_0) / \max(A, B)$ 
    $f'(x_1) = 6f'(x_1) / \max(A, B)$ 

```

This box bound ensures that (A, B) remains within a viable region of monotonicity (satisfying Theorem 4, seen in Fig. 6 of [3]).

```

7:  $\hat{f}''(x_0) = -\sqrt{A}(7\sqrt{A} + 3\sqrt{B}) \frac{f(x_1) - f(x_0)}{(x_1 - x_0)^2}$ 
    $\hat{f}''(x_1) = \sqrt{B}(3\sqrt{A} + 7\sqrt{B}) \frac{f(x_1) - f(x_0)}{(x_1 - x_0)^2}$ 

```

This selection of values of f'' is guaranteed to satisfy Theorem 4 from [3] and is chosen because it is (reasonably) the average of the two endpoints of the interval of monotonicity for second derivative values.

```

8:  $\eta = (f''(x_0), f''(x_1))$ 
    $\eta_0 = (\hat{f}''(x_0), \hat{f}''(x_1))$ 
    $f''(x_0), f''(x_1) = \text{line\_search}(\text{is\_monotone}, \eta, \eta_0)$ 

```

Similar to the simplified check for monotonicity, when the derivative value at one endpoint of the interval is 0, a simplified set of steps can be taken to enforce monotonicity.

Algorithm 2b: `make_monotone_simplified`

```

0:  $f''(x_0) = \max\left(f''(x_0), \frac{-6f'(x_0)}{x_1 - x_0}\right)$ 

```

Considering the α , γ , β , and δ defined in [2], this first step enforces $\gamma > \delta$. It is already guaranteed that $\delta = \frac{f'(x_0)(x_1-x_0)}{f(x_1)-f(x_0)} > 0$. Only two conditions remain to guarantee monotonicity.

$$1: f''(x_1) = \max\left(f''(x_1), f''(x_0) + \frac{14f'(x_0)+16f'(x_1)+60(f(x_1)-f(x_0))/(x_1-x_0)}{x_1-x_0}\right)$$

Now it is guaranteed that $\alpha \geq 0$.

$$2: f''(x_1) = \max\left(f''(x_1), 6f'(x_0) - 4f'(x_1) - f''(x_0)\right)$$

Lastly, this guarantees that $\beta \geq \alpha$. All conditions are met to satisfy proposition 2 of [2] and ensure monotonicity.

Notice that all above algorithms (assuming a fixed level of precision is desired) have $\mathcal{O}(1)$ runtime. Only a finite number of operations are needed for monotonicity verification. A line search is performed for enforcement, however that search requires a fixed number of steps to achieve any predetermined relative precision on the line.

3 Constructing a Piecewise Quintic Monotone Spline

Finally, the construction of a full piecewise quintic spline is outlined using the above algorithms. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a function in \mathcal{C}^2 . Proceed given evaluation tuples $(x_i, f(x_i), f'(x_i), f''(x_i))$ for $i = 0, \dots, N$ such that $x_i < x_{i+1}$ and (without loss of generality) $f(x_i) \leq f(x_{i+1})$ for $i = 1, \dots, N-1$.

Algorithm 3: monotone.spline

0: for $i = 0, \dots, N-1$

1: if (not is_monotone($i, i+1$)) make_monotone($i, i+1$)

In the shorthand notation above, i and $i+1$ refer to the associated tuples of the form $(x_i, f(x_i), f'(x_i), f''(x_i))$. This notation will be used through the remainder of this algorithm.

2: for $j = i-1, \dots, 0$

3: if (not is_monotone($j, j+1$)) make_monotone($j, j+1$)

4: else break

The above ‘for’ loop will be referred to as the cascade fix. If the adjustment of second derivative values causes the previous interval to become nonmonotone, then the left-hand second derivative value must be updated. This may (abnormally) require adjustments across all previously visited intervals.

5: while (not is_monotone($i, i+1$))

6: $f'(x_i) = f'(x_i)/k$

```

7:    $f'(x_{i+1}) = f'(x_{i+1})/k$ 
8:   make_monotone( $i, i + 1$ )

```

In the case that the two corrections to neighboring intervals contradict, the first derivative values of the active interval are decreased to enlarge the overlap of regions I and II (from [3]) of the two intervals.

```

9:   for  $j = i - 1, \dots, 0$ 
10:     if (not is_monotone( $j, j + 1$ )) make_monotone( $j, j + 1$ )
11:     else break

```

Finally an additional ‘cascade fix’ is performed to ensure that all previous intervals are still monotone after shrinking the derivative values of the current interval.

It is mentioned in [3] that for sufficiently small $f'(x_i)$ and $f'(x_{i+1})$ the admissible solution interval of second derivative values becomes arbitrarily large. It can also be seen that decreasing the assigned derivative on right-hand side of an interval always allows the achievement of monotonicity because shrinking $f'(x_1)$ results in γ growing faster than $\sqrt{\beta}$ for *algorithm 1a*, for *algorithm 1b* β and γ will grow faster α and δ respectively. In application, one needs only to pick some $k > 1$ to ensure successful termination.

Given the potential for successive cascade fixes, the runtime of *algorithm 3* is $\mathcal{O}(N^2)$. In practice this has been observed to be incredibly unlikely and difficult to produce. While single cascade fixes can occur from hand crafted examples, the generously rounded values selected to ensure monotonicity frequently prevent successive cascade fixes.

References

- [1] Huynh, H.T.: Accurate monotone cubic interpolation. SIAM Journal on Numerical Analysis **30**(1), 57–100 (1993)
- [2] Schmidt, J.W., Hess, W.: Positivity of cubic polynomials on intervals and positive spline interpolation. BIT Numerical Mathematics **28**(2), 340–352 (1988)
- [3] Ulrich, G., Watson, L.T.: Positivity conditions for quartic polynomials. SIAM Journal on Scientific Computing **15**(3), 528–544 (1994)