CubicSpline

Sunday, November 5, 2017 8:40 AM

on the way to finding g_n-s a matrix was constructed

$$
\begin{array}{c}
1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N
\end{array}
\left[\begin{array}{cccccc}
 & & ? & & & \\
\Delta_2 & 2(\Delta_1+\Delta_2) & \Delta_1 & 0 & 0 & \cdots \\
0 & \Delta_3 & 2(\Delta_2+\Delta_3) & \Delta_2 & 0 & \cdots \\
\vdots & & \ddots & & & \\
\cdots & & \Delta_{N-1} & 2(\Delta_{N-2}+\Delta_{N-1}) & \Delta_{N-2} & \\
 & & ? & & &
\end{array}\right]
\left\{\begin{array}{c} ? \\ \\ \vec{g} \\ \\ ? \end{array}\right\}
=
\left\{\begin{array}{c}
? \\
\frac{3\Delta_2}{\Delta_1}(f_1-f_2)+\frac{3\Delta_1}{\Delta_2}(f_3-f_1) \\
\vdots \\
\frac{3\Delta_{N-1}}{\Delta_{N-2}}(f_{N-2}-f_{N-1})+\frac{3\Delta_{N-2}}{\Delta_{N-1}}(f_N-f_{N-1}) \\
?
\end{array}\right\}
$$

this matrix however, is missing some information for rows 1, and N. we can use eq'n (5) to find these given "free" Boundary conditions

let $f''_{cs}(x_1) = f''_{cs}(x_N) = 0$

observe also that $t(x_1) = \dfrac{x_1 - x_1}{x_2 - x_1} = 0$

and $t(x_N) = \dfrac{x_N - x_{N-1}}{x_N - x_{N-1}} = 1$

then equation 5 Becomes

$$x_1:\ 0 = f_1(\underbrace{\cancel{12t}-6}_{-6}) + f_2(\underbrace{6-\cancel{12t}}_{6}) + \Delta_1\left(g_1(\underbrace{\cancel{6t}-4}_{-4}) + g_2(\underbrace{\cancel{6t}-2}_{-2})\right)$$

$$\rightarrow \boxed{-4\Delta_1 g_1 - 2\Delta_1 g_2 = 6f_1 - 6f_2}$$

$$x_{N-1}:\ 0 = f_{N-1}(\underbrace{\cancel{12t-6}}_{6}) + f_N(\underbrace{\cancel{6-12t}}_{-6}) + \Delta_{N-1}\left(g_{N-1}(\underbrace{\cancel{6t-4}}_{2}) + g_N(\underbrace{\cancel{6t-2}}_{4})\right)$$

$$\rightarrow \boxed{2\Delta_{N-1} g_{N-1} + 4\Delta_{N-1} g_N = 6f_N - 6f_{N-1}}$$

we can now fill in the first and last

rows of our matrix, and solve the linear system to obtain $\vec{g}$.

then having found $\vec{g}$, we can construct $\hat{f}cs_n$ for $n \in [1, N]$

Interpolating Polynomials:

I did not get the same polynomial using m=20. This confused me and frustrated me, and I even tried generating a plot via the same algorithm in Matlab. This failed also. However, I did notice some things about the coefficient matrix which could lead to problems

The matrix **X** was highly ill conditioned. Foe values less than 1, as the columns grew, the value would get smaller, and smaller, eventually, likely, approaching the region of roundoff error or subnormal numbers. For numbers greater than 1, as the columns grew the values would get larger and larger, likely approaching a region of truncation error or overflow error. When doing the arithmetic to solve the system, these errors could have been magnified. When solving the system in Matlab, Matlab warned me that the results could be erroneous, and they appear to be.

Nonetheless, I think that BOTH the Lagrangian and m=20 fits are terrible. They provide little to no use for interpolation since they are so oscillatory. I think the fit at m=15 is the best of the 4 generated.