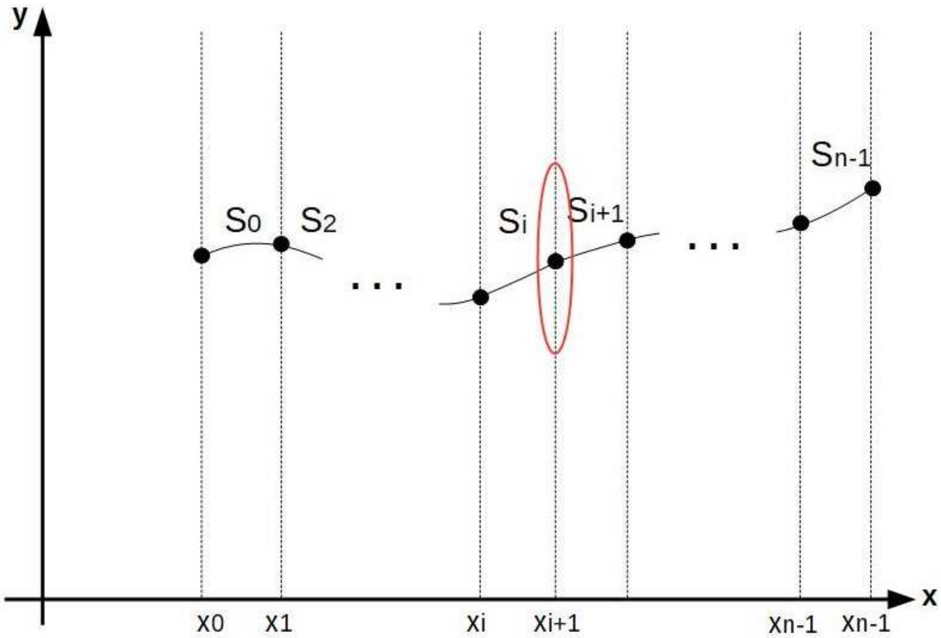
CS323 LECTURE NOTES - LECTURE 14

1 Cubic Splines

In the cases of polynomial interpolation that we have talked about (Lagrange, Neville, divided differences) the polynomial will go through each one of the given points, but it some times oscillates too much for intermediate points. How could we find a better interpolation, which is smoother and still goes through each one of the given points? The answer cannot be another degree n polynomial that goes through the given $n+1$ points, since that polynomial is unique. We are left with trying a lower degree polynomial.

The only way to use lower degree polynomials that go through each one of the $n+1$ points is by reducing the number of points, which takes us to a piecewise polynomial. We could, for example use a first-degree polynomial going through points x_0 and x_1, then another polynomial going through points x_1 and x_2, and so on. The problem with this approach is the lack of smoothness in the unions. We could try degree two polynomials, and ask some boundary conditions to obtain a smooth transition from one polynomial to another. In practice the idea that works best (looks smoother) is to use 3rd-degree polynomials. This method is called *cubic splines.*

1.1 Conditions



1. There will be a 3rd-degree polynomial S_i for each interval $[x_i, x_{i+1}]$. Recall that the given points go from x_0 to x_n, so we will have n polynomials $S_0(x), \ldots, S_{n-1}(x)$

2. The two polynomials $S_i(x)$ and $S_{i+1}(x)$ must agree on point (x_{i+1}, y_{i+1}) (nodal point). In the figure it shown with an oval.

3. The (first) derivative of both polynomials that share the end of each interval must agree.

4. If we count the number of equations so far, they will be less than the number of unknowns, therefore we will add the extra condition that the second derivative of both polynomials that share the end of each interval must agree.

5. If there is no given external condition outside of the interval $[x_0, x_n]$, we will require the second derivative of $S_0(x)$ in x_0 to be equal to zero.

We can write each 3rd-degree polynomial as:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Our objective is to find a_i, b_i, c_i, and d_i, $i = 0, \ldots, n-1$. Which suggest that we must solve a system of equations.

We will need the first and second derivatives of each polynomials that share the point (x_i, y_i), (x_{i+1}, y_{i+1}):

$$\begin{aligned}
S_{i+1}(x) &= a_{i+1} + b_{i+1}(x - x_{i+1}) + c_{i+1}(x - x_{i+1})^2 + d_{i+1}(x - x_{i+1})^3 \\
S'_i(x) &= b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \\
S'_{i+1}(x) &= b_{i+1} + 2c_{i+1}(x - x_{i+1}) + 3d_{i+1}(x - x_{i+1})^2 \\
S''_i(x) &= 2c_i + 6d_i(x - x_i) \\
S''_{i+1}(x) &= 2c_{i+1} + 6d_{i+1}(x - x_{i+1})
\end{aligned}$$

Now we formally write each one of the conditions:
Conditions:

1. $S_i(x_i) = f(x_i)\ \forall i = 0, \ldots, n-1$
2. $S_{i+1}(x_{i+1}) = S_i(x_{i+1})\ \forall i = 0, \ldots, n-1$
3. $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})\ \forall i = 0, \ldots, n-1$
4. $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})\ \forall i = 0, \ldots, n-1$
5. $S''_0(x_0) = 0$

We can define h_i as the width of each interval:

$$h_i = x_{i+1} - x_i \quad \forall i = 0, \ldots, n-1$$

If we substitute in each one of the equations we get:

1. $a_i = f(x_i)$
2. $a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3$ **(5)**
3. $b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2$ **(6)**
4. $c_{i+1} = c_i + 3d_i h_i$ **(7)**
5. $c_0 = 0$

solving for d_i in (7)

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i) \quad \textbf{(11)}$$

now substitute d_i in (5) and solve for b_i

$$b_i = \frac{a_{i+1} - a_i}{h_i} - \frac{2c_i + c_{i+1}}{3}h_i \quad \textbf{(10)}$$

we can compute b_{i+1} using b_i from the previous step

$$b_{i+1} = \frac{a_{i+2} - a_{i+1}}{h_{i+1}} - \frac{2c_{i+1} + c_{i+2}}{3}h_{i+1}$$

substitute b_i and b_{i+1} in (6)
after doing some algebra we get:

$$h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2} = 3\left(\frac{a_{i+2} - a_{i+1}}{h_{i+1}} - \frac{a_{i+1} - a_i}{h_i}\right) \qquad (\mathbf{8})$$

Notice that for each $i = 0 \ldots n-1$ we have an equation. But if we substitute $i = n-1$ in (8) we will need c_{n+1}, which does not exist. So we can only use those equations for $i = 0, \ldots, n-3$, but then we only have $n-2$ equations, and n unknowns: $c_0, \ldots, c_{n-1}$. We can add $c_0 = 0$ obtained from condition 5. We still need one more equation.

We can use a dummy polynomial $S_n(x)$, with the conditions: $S_n(x_n) = x_n$, and $S_n''(x_n) = 0$, from which we get that $c_n = 0$.

We can now write down the complete system of equations:

$$\begin{array}{llllll}
c_0 & & & & & = 0 \\
h_0 & +2(h_0+h_1)c_1 & +h_1c_2 & & & = \frac{3}{h_1}(a_2-a_1) \\
& & & & & \quad -\frac{3}{h_0}(a_1-a_0) \\
& h_1c_1 & +2(h_1+h_2)c_2 & +h_2c_3 & & = \frac{3}{h_2}(a_3-a_2) \\
& & & & & \quad -\frac{3}{h_1}(a_2-a_1) \\
& & h_2c_2 & +2(h_2+h_3)c_3 & +h_3c_4 & = \frac{3}{h_3}(a_4-a_3) \\
& & & & & \quad -\frac{3}{h_2}(a_3-a_2) \\
& & & & \vdots & \vdots \\
& & & & c_n & = 0
\end{array}$$

Notice that $a_i = y_i$, so we know all the a_i's, and the h_i are just the widths of the intervals. Therefore the only unknowns are the c_i's.

We can write the system in matrix form **(9)**:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & \dots & 0 \\
h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & \dots & 0 \\
0 & h_1 & 2(h_1+h_2) & h_2 & 0 & \dots & 0 \\
0 & 0 & h_2 & 2(h_2+h_3) & h_3 & \dots & 0 \\
0 & 0 & 0 & h_3 & 2(h_3+h_4) & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & 1
\end{pmatrix}
\begin{pmatrix}
c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ \vdots \\ c_n
\end{pmatrix}$$

$$= \begin{pmatrix}
0 \\
\frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\
\frac{3}{h_2}(a_3-a_2) - \frac{3}{h_1}(a_2-a_1) \\
\frac{3}{h_3}(a_4-a_3) - \frac{3}{h_2}(a_3-a_2) \\
\vdots \\
\vdots \\
0
\end{pmatrix}$$

1.2 Algorithm

Now we can describe the algorithm to find $S_0(x)$,...,$S_n(x)$.

- Use (1) to find a_i $\forall i = 0, \dots, n$

- Solve the system of equations (9) to obtain c_i $\forall i = 0, \ldots, n$
- Substitute in (10) to obtain b_i, $\forall i = 0, \ldots, n-1$. Notice that to compute b_{n-1} it will be required to use $c_n = 0$ y $a_n = y_n$
- Finally, we substitute in (11) to obtain d_i $\forall i = 0, \ldots, n$

1.3 Example

Suppose that we want to find cubic splines that go through the points:

x	**y**
1	5
2	6
3	6.5
4	5.5
5	5.5
6	7

In the case of equidistant spacing (constant h), we have that the system of equations that we need to solve to find the c_i's is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 3(5-2(6)+6.5) \\ 3(6-2(6.5)+5.5) \\ 3(6.5-2(5.5)+5.5) \\ 3(5.5-2(5.5)+7) \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1.5 \\ -4.5 \\ 3 \\ 4.5 \\ 0 \end{pmatrix}$$

Solving this system we get:

$$\begin{aligned} c_1 &= 0 \\ c_2 &= -0.043062 \\ c_3 &= -1.327751 \\ c_4 &= 0.854067 \\ c_5 &= 0.911483 \\ c_6 &= 0 \end{aligned}$$

The following table shows the values of b and d obtained from equations (11) and (10):

x	y	a	b	c	d
1	5	5	1.014354	0	-0.014354
2	6	6	0.971291667	-0.043062	-0.42822967
3	6.5	6.5	-0.39952167	-1.327751	0.727272667
4	5.5	5.5	-0.87320567	0.854067	0.019138667
5	5.5	5.5	0.892344667	0.911483	-0.30382767
6	7	7		0	

Finally, the polynomials are:

$$
\begin{aligned}
S_0(x) &= 5 + 1.014354(x-1) - 0.014354(x-1)^3 \\
S_1(x) &= 6 + 0.971291667(x-2) - 0.043062(x-2)^2 - 0.42822967(x-2)^3 \\
S_2(x) &= 6.5 - 0.39952167(x-3) - 1.327751(x-3)^2 + 0.727272667(x-3)^3 \\
S_3(x) &= 5.5 - 0.87320567(x-4) + 0.854067(x-4)^2 + 0.019138667(x-4)^3 \\
S_4(x) &= 5.5 + 0.89234467(x-5) + 0.911483(x-5)^2 - 0.30382767(x-5)^3
\end{aligned}
$$