Chapter 2

Piecewise polynomial interpolation

In Section 1.6.1, and in Lab 1, we learned that it is not a good idea to interpolate functions by a high-order polynomials at equally spaced points. However, it transpires that it is possible to obtain a very good approximations using a very simple method. The trick is to use a *spline*: a *piecewise polynomial interpolating function*.

We'll consider three important example of splines:

1. *linear splines*
2. *(natural) cubic splines*.
3. *Hermite* piecewise cubics.

In this section, we always have N equally spaced points: let $h = (b-a)/N$, then

$$a = x_0, \quad b = x_N \quad \text{and} \quad x_i = x_0 + ih.$$

Often these are referred to as *knots points* (or simply as *knots*), and denote the set of knot points by $\omega^N := \{x_i\}_{i=0}^N$.

For more details about splines, have a look at [SM03, Chap. 11], and [S98, Lectures 10 and 11].

2.1 Linear Interpolating Splines

We first study the *piecewise linear interpolant*, also called a *linear spline*. We will see that they have important properties, including

(a) they are easy to construct and analyse;

(b) the bound on the error decreases as the number of interpolation points increases;

(c) the error we get using a linear spline is no more than twice the error using the best possible (piecewise linear) approximation; amd

(d) of all the interpolants to f at a given set of points, the linear spline is the one with the smallest 1st derivative.

2.1.1 Construction

Definition 2.1.1. Let f be a function that is continuous on $[a, b]$. The *linear spline interpolant* to f is the continuous function l such that

(i) $l(x_i) = f(x_i)$ for each $i = 0, 1, \dots, N$,

(ii) l is a linear function l_i on each interval $[x_{i-1}, x_i]$.

It is easy to write down a formula for the l_i, based on Lagrange polynomials. Set $h = (b-a)/N$. Then, for $x \in [x_{i-1}, x_i]$

$$l_i(x) = f(x_{i-1})\frac{x_i - x}{h} + f(x_i)\frac{x - x_{i-1}}{h}. \qquad (2.1)$$

Example 2.1.2. Write down the linear spline interpolant to $f(x) = e^x$ at the knot points $\{-1, 0, 1\}$.

<u>Take notes</u>:

2.1.2 Analysis

We know that if p_N is the polynomial of degree N that interpolates f at N equally spaced points, it does **not** follow that $p_N \to f$ as $n \to \infty$. But as we will see, the piecewise linear interpolant to f converges to f, albeit slowly.

This is verified in the following theorem, which is a direct consequence of Theorem 1.3.3.

Theorem 2.1.3. *Suppose that* f, f' *and* f'' *are all continuous and defined on the interval* $[a, b]$. *Let* l *be the linear spline interpolant to* f *on the* $N+1$ *equally spaced points* $a = x_0 < x_1 \cdots < x_N = b$ *with* $h = x_i - x_{i-1} = (b-a)/N$. *Then*

$$\|f - l\|_\infty \leqslant \frac{h^2}{8}\|f''\|_\infty,$$

(Here, as usual, $\|g\|_\infty$ is defined as $\max_{a\leqslant x\leqslant b} |g(x)|$.)

Take notes:

It now follows directly from Theorem 2.1.3 that

$$\lim_{n\to\infty} \|f - l\|_\infty = 0.$$

Example 2.1.4. Figure 2.1 shows linear spline interpolations of Runge's example:

$$f(x) = \frac{1}{1+x^2} \text{ on } [-5, 5].$$

These diagrams appear to support our assertion that the error tends to zero as $n \to \infty$.

Example 2.1.5. Suppose you are interpolating $f(x) = e^x$ on N equally spaced intervals between $x_0 = -1$ and $x_N = 1$. What value of N would you have to take to ensure that the maximum error is less than 10^{-2}?

Take notes:

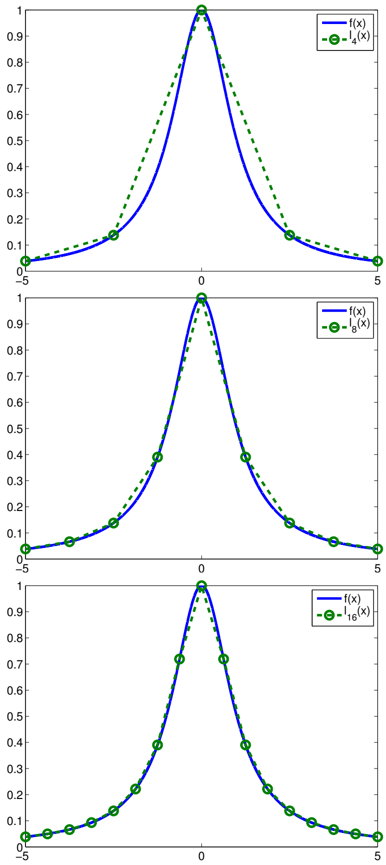

Fig. 2.1: Linear spline interpolants to $1/(1 + x^2)$ on $[-5, 5]$. Contrast with Figure 1.4

2.1.3 Best approximation

For the next part of the analysis it will help to think of piecewise linear interpolation as an *operator*. Then we can compare the linear spline to all the other piecewise linear approximations.

First, observe that one can define an infinite number of piecewise linear functions on a given set of knot points ω^N. We'll call the set of these functions $\mathcal{L}$.

Definition 2.1.6. For a fixed set of knot points ω^N, let L be the operator that maps the continuous function f to its linear spline interpolant $l \in \mathcal{L}$.

Now suppose that $g \in \mathcal{L}$. Then $L(g) = g$. That is L is a *projection*: $L(L(f)) = L(f)$.

It is not hard to see that one could find a different function $\hat{l} \in \mathcal{L}$ that is a better approximation of f in sense that

$$\max_{x_0\leqslant x\leqslant x_n} |f(x) - \hat{l}(x)| < \max_{x_0\leqslant x\leqslant x_n} |f(x) - l(x)|.$$

However, l is very easy to find, and the associated error is no worse than twice $\|f(x) - \hat{l}(x)\|_\infty$.

Theorem 2.1.7 (Stewart's "Afternotes goes to grad school",

Lecture 10). *Let* $l = L(f)$. *For all* $\hat{l} \in \mathcal{L}$,

$$\|f - l\|_\infty \leqslant 2\|f - \hat{l}\|_\infty.$$

(That L is a projection is key to the proof.)

Take notes:

2.1.4 Minimum Energy

The final interesting property of l that we will study is called the *minimum energy property*.

Definition 2.1.8. Let u be a function that is continuous and defined on the interval $[a, b]$ except, maybe, at the (countable set) ω^N of knot points[1] Then the 2-norm of u is

$$\|u\|_{2,[a,b]} := \Big(\int_a^b u^2(x)dx\Big)^{1/2}.$$

Usually we just write this as $\|u\|_2$.

Let H^1 be the set of all functions u that are continuous on $[a, b]$ and have $\|u'\|_2 < \infty$. Note that $l' \in H^1$, even though we have not properly defined l' at the mesh points ω^N.

Theorem 2.1.9 (Süli and Mayers, Thm. 11.2). *Let* w *be* any *function in* H^1 *that interpolates the function* f *at the points in* ω^N. *Let* l *we the linear spline interpolant of* f. *Then*

$$\|l'\|_2 \leqslant \|w'\|_2.$$

Take notes:

[1] More precisely, we should say "everywhere, except on a set of measure zero". What that means is very important to the areas of measure theory and functional analysis. However, since some of you are not taking those courses until next year, we'll air-brush over the exact definition.

2.1.5 Exercises

Exercise 2.1. Page 28 of the Department of Education's old Mathematics Tables ("The *Log Tables*") reports that $\ln(1) = 0$, $\ln(1.5) = 0.4055$ and $\ln(2) = 0.6931$.

(i) Write down the linear spline l that interpolates $f(x) = \ln(x)$ at the points $x_0 = 1$, $x_1 = 1.5$ and $x_2 = 2$.

(ii) Use this to estimate $\ln(x)$ at $x = 1.2$. How does this compare to the value in the tables? (0.1823)

(iii) Give an estimate for the maximum error:

$$\max_{1 \leqslant x \leqslant 2} |f(x) - l(x)|.$$

(iv) What value of n would you choose to ensure that $|f(x) - l(x)| \leqslant 0.001$ for all $x \in [1, 2]$.

Exercise 2.2. As an alternative to (2.1), one can define the linear spline interpolant to a function is as a linear combination of a set of piecewise linear basis functions $\{\psi_i\}_{i=0}^N$:

$$\psi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

(i) Write down a formula for the $\psi_i(x)$;

(ii) derive a formula for $l(x)$ in terms of the ψ_i.

This exercise is useful: we'll be using these basis functions (called "hat" functions) in the final section of the course.

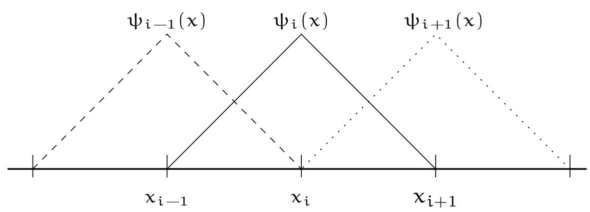

Fig. 2.2: Some hat functions

2.2 Cubic Splines

The Cubic Spline[2] is perhaps the most useful and popular interpolating function used in numerical analysis, automotive and aeronautical engineering, computer and film animation, digital photography, financial modelling, and as many other areas as there are human endeavours where continuous processes are modelled by discrete ones.

One could argue that the most fundamental shortcoming of the linear spline interpolant to f (see, for example, Figure 2.1) is that they are not "smooth": that is, although $l_i(x_i) = l_{i+1}(x_i)$, it is **not** generally the case that $l_i'(x_i) = l_{i+1}'(x_i)$. Also, the interpolating function cannot capture the "curvature" of f. (If you think about this last statement, you'll see that it can be expressed as $l''(x) = 0$ for all $x \in [a, b]$.)

Cubic splines try to balance the simplicity of the linear spline approach — by using a low-order polynomial on each interval — with the desire for curvature and more smoothness – by using cubics, and by forcing adjacent ones, and their first and second derivatives, to agree at knot points.

Definition 2.2.1. Let f be a function that is continuous on $[a, b]$. The *cubic spline interpolant* to f is the continuous function S such that

(i) for $i = 1, \ldots, N$, on each interval $[x_{i-1}, x_i]$ let $S(x) = s_i(x)$, where each of the s_i is a cubic polynomial.

(ii) $s_i(x_{i-1}) = f(x_{i-1})$ for $i = 1, \ldots, N$,

(iii) $s_i(x_i) = f(x_i)$ for $i = 1, \ldots, N$,

(iv) $s_i'(x_i) = s_{i+1}'(x_i)$ for $i = 1, \ldots, N-1$,

(v) $s_i''(x_i) = s_{i+1}''(x_i)$ for $i = 1, \ldots, N-1$.

So, we have defined the cubic spline S as a function that interpolates f at $N+1$ points, has continuous first and second derivatives on $[x_0, x_N]$ and is a cubic polynomial on each of the n intervals $[x_{i-1}, x_i]$. That is, it is *piecewise cubic*. We know that one can write a cubic as

$$p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$

Thus it takes 4 terms to uniquely define a single cubic. To uniquely determine N cubics requires $4N$ terms. They can be found by solving $4N$ (linearly independent) equations. But an inspection of (ii)–(iv) above reveals only $4N - 2$ equations.

[2] It is generally accepted that (mathematical) splines were first described by Isaac Schoenberg (1903-1990) during WW2. However their physical realisation had been used in the ship-building and aircraft industries prior to that.

The "missing" equations can be chosen in a number of ways:

(i) by setting $S''(x_0) = 0$ and $S''(x_N) = 0$. This is called a *natural* spline, and is the approach we'll take.

(ii) by setting $S'(x_0) = 0$ and $S'(x_N) = 0$. This is called a *clamped* spline.

(iii) set $S'(x_0) = S'(x_N)$ and $S''(x_0) = S''(x_N)$. This is the *periodic* spline and is used for interpolating, say, trigonometric functions.

(iv) only use $N - 2$ components of the spline: $s_2, \ldots, s_{N-1}$. But extend to two end ones so that $s_2(x_0) = f(x_0)$ and $s_{N-1} = f(x_N)$. This is called the *not-a-knot* condition.

In Figure 2.3 we show the natural cubic spline interpolants to $f(x) = 1/(1 + x^2)$ (with $a = -5$ and $b = 5$ as usual) on $N = 4$, 8 and 16 intervals. Compared with Figure 2.1, we seem to get significantly better approximation. Moreover, the rate at which the error decreases with respect to h is much faster.

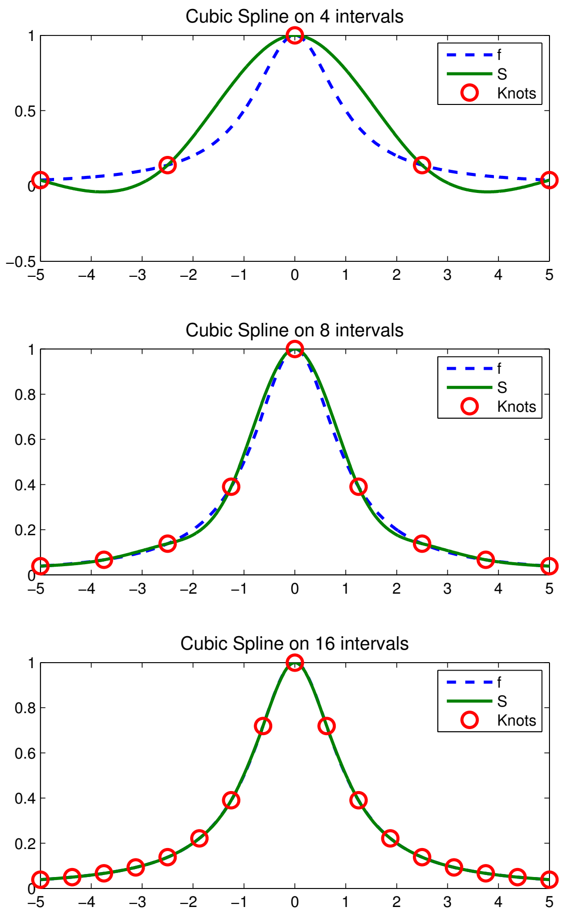

Fig. 2.3: Cubic spline interpolants to $1/(1 + x^2)$. Compare with Figure 2.1

2.2.1 Constructing Cubic Splines

Some notation:

- $h = x_i - x_{i-1} = (x_N - x_0)/N$ for all i.
- $f_i := f(x_i)$.

To construct the spline, first observe that if S is piecewise cubic, then S' is piecewise quadratic, and S'' is piecewise linear.

(i) Let $\sigma_i = S''(x_i)$ for $i = 0, \ldots, N$.

Take notes:

(ii) Integrate twice:

Take notes:

We get

$$s_i(x) = \alpha_i(x - x_{i-1}) + \beta_i(x_i - x) + \frac{\sigma_{i-1}}{6h}(x_i - x)^3 + \frac{\sigma_i}{6h}(x - x_{i-1})^3, \quad (2.2)$$

for $x \in [x_{i-1}, x_i]$.

(iii) The α_i and β_i arose as the constants of integration. To find them:

Take notes:

and so, for $i = 1, 2, \ldots, N$

$$\alpha_i = \frac{f_i}{h} - \frac{h}{6}\sigma_i, \quad \beta_i = \frac{f_{i-1}}{h} - \frac{h}{6}\sigma_{i-1}. \quad (2.3)$$

Notice that this gives $\beta_i = \alpha_{i-1}$.

(iv) So, now we "just" need the equations for σ_i. Two of these come from the fact that this is a "natural" cubic spline, with $S''(x_0) = 0$ and $S''(x_N) = 0$, therefore $\sigma_0 = 0$ and $\sigma_N = 0$.

The remaining $N - 1$ equations come from the fact that S' is continuous at $x_1, x_2, \ldots, x_{N-1}$. So we set that

$$s_i(x_i) = s_{i+1}(x_i).$$

After some work (see Exercise 2.3), we can show this means that the system is:

$$\sigma_0 = 0, \quad (2.4a)$$

$$\frac{1}{6}(\sigma_{i-1} + 4\sigma_i + \sigma_{i+1}) = \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}) \quad (2.4b)$$

for $i = 1, \ldots, N-1$,

$$\sigma_N = 0. \quad (2.4c)$$

2.2.2 A cubic spline example

Example 2.2.2. Find the natural cubic spline interpolant to f at the points $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ and $x_3 = 3$ where $f_0 = 0, f_1 = 2, f_2 = 1$ and $f_3 = 0$.

[We won't do all the details in class, so here they are]

Solution:

From (2.2) we see we are looking for $S(x)$=

$$\begin{aligned} &\alpha_1 x + \beta_1(1 - x) + \frac{\sigma_0}{6h}(1 - x)^3 + \frac{\sigma_1}{6h}x^3, x \in [0, 1],\\ &\alpha_2(x - 1) + \beta_2(2 - x) + \frac{\sigma_1}{6h}(2 - x)^3 + \frac{\sigma_2}{6h}(x - 1)^3, x \in [1, 2],\\ &\alpha_3(x - x_2) + \beta_3(3 - x) + \frac{\sigma_2}{6h}(3 - x)^3 + \frac{\sigma_3}{6h}(x - 2)^3, x \in [2, 3]. \end{aligned}$$

We first solve for the σ_i using (2.4):

$$\frac{1}{6}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -18 \\ 0 \\ 0 \end{pmatrix}$$

This gives

$$\sigma_0 = 0, \quad \sigma_1 = -\frac{24}{5}, \quad \sigma_2 = \frac{6}{5}, \quad \sigma_3 = 0.$$

Now use (2.3) to get

$$\alpha_1 = \frac{14}{5}, \quad \alpha_2 = \frac{4}{5}, \quad \alpha_3 = 0.$$

and

$$\beta_1 = 0 \quad \beta_2 = \frac{14}{5}, \quad \beta_3 = \frac{4}{5}.$$

The answer is

$$S(x) = \begin{cases} \frac{14}{5}x - \frac{4}{5}x^3, & x \in [0,1], \\ \frac{4}{5}(x-1) + \frac{14}{5}(2-x) - \\ \qquad \frac{4}{5}(2-x)^3 + \frac{1}{5}(x-1)^3, & x \in [1,2], \\ \frac{4}{5}(3-x) + \frac{1}{5}(3-x)^3. & x \in [2,3]. \end{cases}$$

This is shown in Figure 2.4.

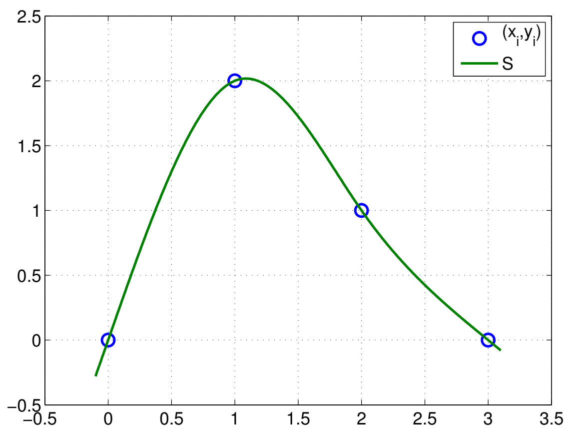

Fig. 2.4: Spline interpolant from Example 2.2.2

2.2.3 Error Estimates

We state the following error estimates without proof. You don't need to know these, or be able to prove them. But you should be able to use them if required.

Theorem 2.2.3. *If $f \in C^4[a,b]$ and S is its cubic spline interpolant on $N+1$ equally spaced points, then*

$$\|f - S\|_\infty \leqslant \frac{5}{384} M_4 h^4,$$
$$\|f' - S'\|_\infty \leqslant \frac{1}{24} M_4 h^3,$$
$$\|f'' - S''\|_\infty \leqslant \frac{3}{8} M_4 h^2,$$

where $M_4 = \|f^{(iv)}\|_\infty$.

Example 2.2.4. Give an upper bound on the error for the cubic spline interpolant to $f = e^x$ on the interval $[-1, 1]$ with $N = 10$ mesh points.

<u>`Take notes`</u>:

Example 2.2.5. What is the smallest value of N that you must take to ensure that, if interpolating $f(x) = e^x$ on N equally sized intervals, the error is less that 10^{-8}? How does this compare with a linear spline interpolation (see Example 2.1.5)?

<u>`Take notes`</u>:

Recall Theorem 2.1.9 for linear splines. There is an analogous result for natural cubic splines.

Theorem 2.2.6. *Let u be any function that interpolates f at ω^N, and is such that $u \in H^2(x_0, x_N)$. Then*

$$\|S''\|_2 \leqslant \|u''\|_2.$$

The proof is not hard - it is analogous to the proof of Theorem 2.1.9.

2.2.4 Exercises

Exercise 2.3. When deducing the system of equations for the natural cubic spline, we showed how to construct the formulation given in (2.2) and the relationship between σ_i, α_i and β_i in (2.3). Now carefully show to deduce the system (2.4).

Exercise 2.4. (For students who did MA385). Write the equations in (2.4) as a linear system $A\boldsymbol{\sigma} = \mathbf{b}$, where A is an $n \times n$ matrix. Show that A is nonsingular, and hence that the system as a unique solution.

Exercise 2.5. Find the natural cubic spline interpolant to $f(x) = \sin(\pi x/2)$ at the nodes $\{x_i\}_{i=0}^3 = \{0, 1, 2, 3\}$.

Calculate value of the interpolant at $x = 2.5$. What is the error at this point?

Exercise 2.6. Take $f(x) = \ln(x)$, $x_0 = 1$, $x_N = 2$. What value of N would you have to take to ensure that $|\ln(x) - S(x)| \leqslant 10^{-4}$ for all $x \in [1, 2]$?

2.3 Piecewise Hermite Interpolation

In this section we introduce another type of cubic spline. It is not as smooth as the natural spline of the precious section—only it and its first derivative are continuous on $[x_0, x_N]$—and it requires that we know $f'(x_i)$. But its easier to construct (don't have to solve a linear system) and analyse than that natural spline.

As before, for short-hand, we write $f(x_i)$ as f_i and $f'(x_i)$ as f'_i.

And, as ever, we'll simplify the analysis by taking the points to be equally spaced: $x_i - x_{i-1} = h$ for each i).

2.3.1 PCHIP

Definition 2.3.1. Given a set of interpolation points $x_0 < x_1 < \cdots < x_N$, the *Piecewise Cubic Hermite Spline Interpolant* (*PCHIP*), S, to the function f, satisfies

(i) $S \in C^1[x_0, x_N]$,

(ii) $S(x_i) = f(x_i)$ and $S'(x_i) = f'(x_i)$ for $i = 0, 1, \dots, N$.

(iii) On each interval $[x_{i-1}, x_i]$, S is a cubic polynomial.

We can construct these splines as follows. On each interval $[x_{i-1}, x_i]$, let S be the cubic polynomial S_i given by

$$S_i(x) = c_0 + c_1(x - x_{i-1}) + c_2(x - x_{i-1})^2 + c_3(x - x_{i-1})^3, \quad (2.5)$$

Then we can show... Take notes:

This gives that

$$\begin{aligned} c_0 &= f_{i-1}, \\ c_1 &= f'_{i-1}, \\ c_2 &= \frac{3}{h^2}(f_i - f_{i-1}) - \frac{1}{h}(f'_i + 2f'_{i-1}), \\ c_3 &= \frac{1}{h^2}(f'_i + f'_{i-1}) - \frac{2}{h^3}(f_i - f_{i-1}). \end{aligned}$$

Figure 2.5 shows some PCHIP interpolants to $f(x) = 1/(1+x^2)$ on the interval $[-5, 5]$. Compare with Figures 2.1 and 2.3.

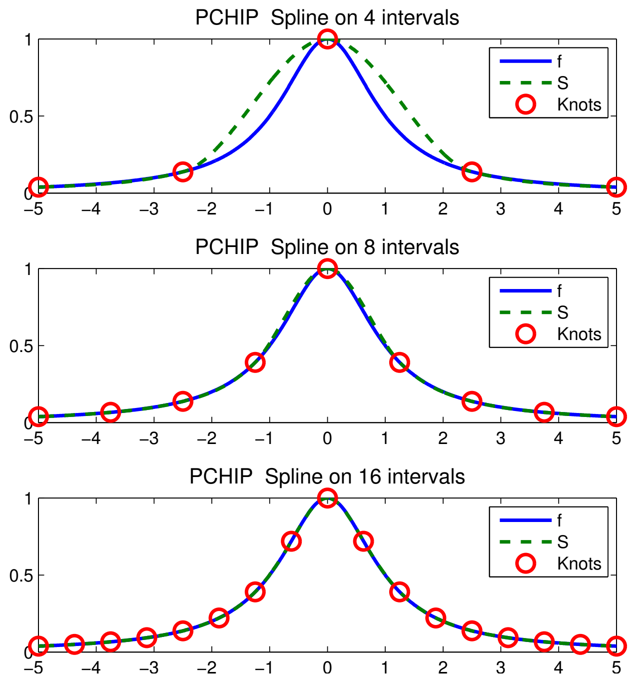

Fig. 2.5: PCHIP interpolants to $1/(1+x^2)$. Compare with Figure 2.3

We now want to prove an error estimate. The norm we use is

$$\|f - S\|_\infty := \max_{x_0 \leqslant x \leqslant x_N} |f(x) - S(x)|$$

Theorem 2.3.2. *Let $f \in C^4[x_0, x_N]$ and let S be the Hermite Cubic Spline interpolant to it at the N equally spaced points $a = x_0 < x_1 < \cdots < x_N = b$. Then*

$$\|f - S\|_\infty \leqslant \frac{h^4}{384}\|f^{(iv)}\|_\infty.$$

Take notes:

2.3.2 Estimating derivatives

You can ignore this section. It is included for completeness. It should make sense to anyone who took MA385, given that it relies only on Taylor series.

As we defined the PCHIP interpolant one needs to know f' in order to be able to construct it. However, you might notice that the MATLAB function `>> pchip` only requires f, not f'. How does it do this, one might wonder?

It turns out that there are important variants on the PHCIP scheme that don't involve knowing f'_0, f'_1, ..., f'_N, but instead uses *approximations* for f'. There are several approaches for deriving these estimates, including

(i) Geometrically. Recall, all we are really looking for is the slope of the tangent to f at $x = x_i$.

(ii) By finding a polynomial that interpolates f and differentiating that.

(iii) Taylor's Theorem.

For more details see [S96, Lecture 24]. Or look back at your notes for MA385, Lecture 14.

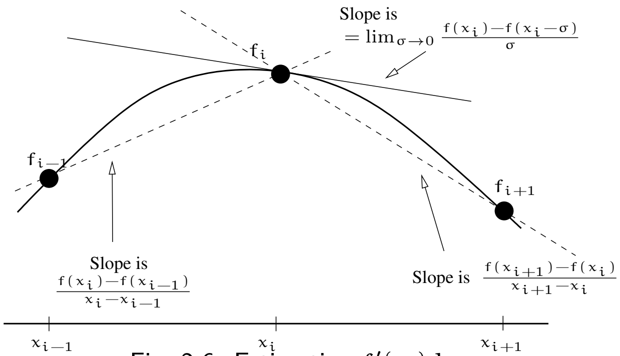

Fig. 2.6: Estimating $f'(x_i)dx$

As suggested by Figure 2.6 one could take

$$f'(x_i) \approx \frac{f_i - f_{i-1}}{x_i - x_{i-1}} = \frac{1}{h}(f_i - f_{i-1})$$

Or

$$f'(x_i) \approx \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{1}{h}(f_{i+1} - f_i).$$

Since one seems to over estimate $f'(x_i)$, and the other seems to under-estimate, we could take the average:

$$f'(x_i) \approx \frac{-f_{i-1} + f_{i+1}}{x_{i+1} - x_{i-1}} = \frac{1}{2h}(-f_{i-1} + f_i)$$

These approximations can also be easily derived from Taylor's Theorem:

$$f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \frac{(x-a)^3}{3!}f'''(a) + \cdots + \frac{(x-a)^n}{n!}f^{(n)}(a) + \frac{(x-a)^{n+1}}{(n+1)!}f^{(n+1)}(\eta),$$

for some $\eta \in (a, b)$. Using this we can get

$$f_{i+1} = f_i + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(\tau_1),$$
$$f_{i-1} = f_i - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(\tau_2),$$

for $\tau_1 \in (x_i, x_{i+1})$ and $\tau_2 \in (x_{i-1}, x_i)$. Now subtract the 1st equation from the second to get the formula and error estimate.

2.3.3 Exercises

Exercise 2.7. Recall Exercise 2.5. Calculate the value to the PCHIP interpolant to $f(x) = \sin(\pi x/2)$ at the nodes $\{x_i\}_{i=0}^3 = \{0, 1, 2, 3\}$ at the point $x = 2.5$. What is the error at this point?

Exercise 2.8. Let $f(x) = \ln(x)$. Let l and S be the piecewise linear and Hermite cubic spline interpolants (respectively) to f on $n+1$ equally spaced points $1 = x_0 < x_1 < \cdots < x_N = 2$. What value of n would you have to take to ensure that

(i) $\max_{1\leqslant x\leqslant 2} |f(x) - l(x)| \leqslant 10^{-4}$?

(ii) $\max_{1\leqslant x\leqslant 2} |f(x) - S(x)| \leqslant 10^{-4}$?

Also, using the code you developed in Lab 2, compare these theoretical results with the value needed on practice.

Exercise 2.9. There are ways of constructing the PCHIP, other than (2.5). For example, let $s = x - x_{k-1}$, then

$$S(x) = \frac{h^3 - 3hs^2 + 2s^3}{h^3}f_{k-1} + \frac{3hs^2 - 2s^3}{h^3}f_k + \frac{s(s-h)^2}{h^2}f'_{k-1} + \frac{s^2(s-h)}{h^2}f'_k$$

Show that this is the same as the PCHIP.