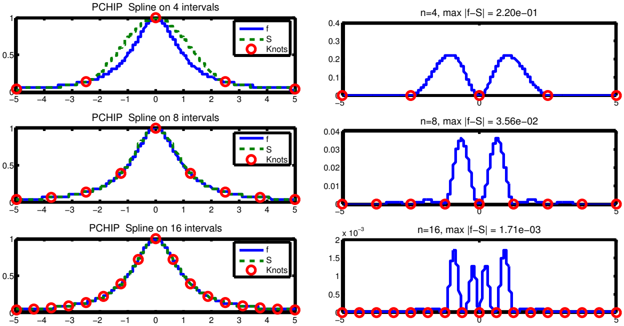
§2 Piecewise Polynomial Interpolation

§2.3 The PCHIP Interpolant

MA378/531 – Numerical Analysis II ("NA2")

February 2017

started
15/2/17



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.

So.
(i) not as smooth as natural spline
(ii) need f'_i

But
(iii) Easy to construct
(iv) Easy to analyse.

(Also (v) it is convex)

Definition (Piecewise Cubic Hermite Spline Interpolant)

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) $S \in C^1[x_0, x_N]$, so S and S' are continuous.

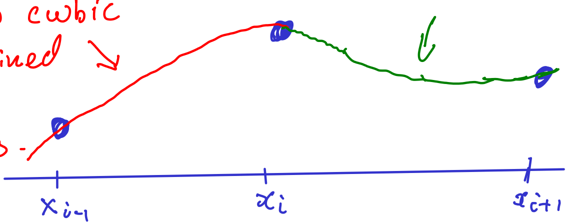
(ii) $S(x_i) = f(x_i)$ and $S'(x_i) = f'(x_i)$ for $i = 0, 1, \ldots, N$.

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

interpolates f & f'

S_{i+1}

$S_i(x)$ is cubic, so determined by $\underline{4}$ conditions.



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

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

Then we can show... how to derive formulae for c_0, c_1, c_2, c_3.

c_0 & c_1 are easy:

Since $S_i(x_{i-1}) = f_{i-1}$, we get that

$$S_i(x_{i-1}) = c_0 + c_1 (\underbrace{x_{i-1} - x_{i-1}}_{0}) + c_2(\underbrace{x_{i-1} - x_{i-1}}_{0})^2 + c_3(\underbrace{x_{i-1} - x_{i-1}}_{0.})^3$$
$$= f_{i-1}$$

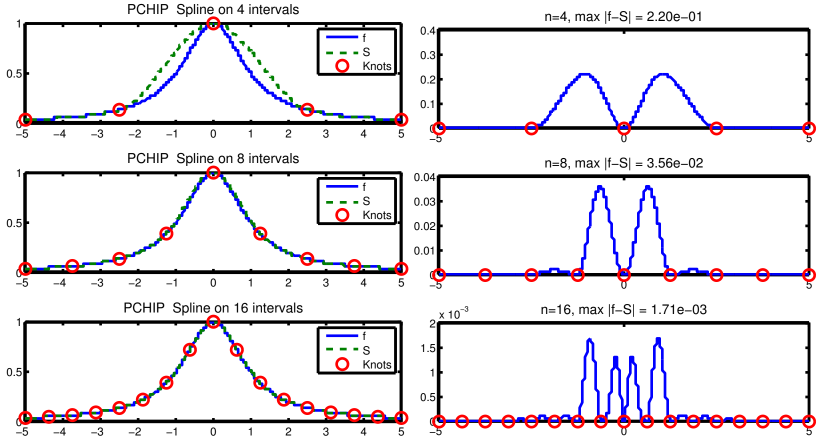
Thus $c_0 = f_{i-1}$.

For c_1, use $S_i'(x_{i-1}) = f_{i-1}'$ (see board).

This gives that

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

The figure below shows some PCHIP interpolants to $f(x) = 1/(1+x^2)$ on the interval $[-5, 5]$.



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

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

Theorem

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

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

Proof comes directly from the analysis of the Hermite interpolant in Section 1:

$$\|f - S\|_\infty = \max_{i=1,\dots,N} \max_{x_{i-1} \le x \le x_i} |f(x) - S_i(x)|.$$

(The remainder of these slides differ from the "book" version of the notes on the NA2 website).

To finish with this section, we'll discuss

- This it is possible to compute the PCHIP interpolant without knowing f';
- Further extensions and ideas;
- How this section relates to the rest of the module.

You can ignore this section. Details are included in Section 2.3.2 of the "book" version of the notes. They are only included for completeness. They 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, the MATLAB function `pchip` that constructs these interpolants 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'. These are obtained using simple expression involving Taylor Series, e.g.,

$$f'(x_i) = \frac{1}{h}(f_i - f_{i-1}) + \mathcal{O}(h), \text{ or } f'(x_i) = \frac{1}{2h}(-f_{i-1} + f_i) + \mathcal{O}(h^2).$$

There are lots of great texts out there on spline interpolation. One of the most famous is Carl de Boor's *Practical Guide to Splines*. He made fundamental contributions to the idea of "B-splines", initiated when working in General Motors in the 1960's.

The idea actually dates back to the 19th century and the work on Nikolai Lobachevsky, who was (unfairly) made (in)famous by Tom Lehrer:

> *Plagiarize,*
> *Let no one else's work evade your eyes,*
> *Remember why the good Lord made your eyes,*
> *So don't shade your eyes,*
> *But plagiarize, plagiarize, plagiarize...*
> *Only be sure always to call it please, "research".*

Extensions to the ideas we have seen here include

- *Functional data analysis* in statistics, where one tries to compute spline (for example) approximations to data with noise. Instead of interpolating all the data, you try to balance the error at the node points with the magnitude of the second derivatives.
- Non-uniform rational basis spline (NURBS) used in computer graphics.

Our next topics in NA2 are:

1. **Approximation of definite integrals**. We shall see that the classic methods of the Trapezium Rule and Simpson's rule are just based on piecewise polynomial interpolation.

Also, Gaussian quatrature, based on Hermite interpolants, but with $f'(x_i)=0$.

2. ***Solution of differential equations***

"finite Elements".

Finished 15/2/17