MA378 Lab 2: Piecewise Polynomial Interpolation

Goal: investigate the accuracy and convergence of piecewise linear and cubic spline interpolants.

. . . .

1 Outline

In class, we did a few examples of writing out piecewise polynomial interpolants of functions. However, in practice they are always implemented by computer.

In this lab, we will look at how this is done in MATLAB.

Moreover, you will investigate how the numerical results compare with the theory. This has two purposes.

(i) To verify that the theory is correct. This is of particular interest for the error estimate given for cubic splines since they were presented without proof.

(ii) (More important) to verify that the theory is *sharp*. For example, we might prove that the error for piecewise linear interpolation is bounded by Ch^2, for some constant C, but this does not exclude the possibility that in fact it can be bounded by Ch^3. However this is easily verified experimentally.

2 Some Spline Basics

Let $\{x_i\}_{i=0}^N := x_0 < x_1 < \cdots < x_N$ be a set of interpolation points. The *piecewise linear interpolant*, l, of a function f at those points is defined as follows

(a) l is continuous on $[x_0, x_N]$,

(b) l is linear (i.e., a polynomial of degree 1), denoted l_i, on each subinterval $[x_{i-1}, x_i]$, for $i = 1, \dots, N$.

(c) $l(x_i) = f(x_i)$ on each $i = 0, \dots, N$.

Revise your notes from Section 2.1 to see how to write down a formula for l_i, the restriction of l to the interval $[x_{i-1}, x_i]$.

Recall Theorem 2.1.3: that

$$\|f - l\|_\infty \le \frac{h^2}{8}\|f''\|_\infty, \tag{1}$$

where $h = x_i - x_{i-1}$, and the norm $\|\cdot\|_\infty$ means

$$\|u\|_\infty := \max_{x_0 \le x \le x_N} |u(x)|.$$

3 Piecewise Linears

Download the files `lspine.m` and `lab2.m`.

The function `lspline()` is called with syntax:

```
L = lspline(x,f,X);
```

and sets L to be value of the piecewise linear interpolant to $\{(x_i, f_i)\}_{i=0}^n$ evaluated at the point X. Read the code and make sure you understand how it works. (By the way, it is written in a rather inefficient manner. This is just so that it is relatively easy to understand.)

The Matlab script in the file `lab2.m` plots

$$f(x) = \sin(\pi x),$$

against its piecewise linear interpolant for $N = 4$, $x_0 = 0$, $x_N = 1$, $h = (b-a)/N$ and $\omega^N = \{x_i\}_{i=0}^N$ where $x_i = x_0 + ih$. It also computes the error.

By taking $N = 1, 2, 4, 8, 16, \dots$ verify that, for this example, the error bounded is proportional to Ch^2, as suggested by (1). That is, we will suppose that there is a constant, C, such that

$$\|f - l\|_{\infty,[0,1]} \le Ch^2.$$

Use your code to give an estimate for C.

Also, use that value of C to estimate what value of N you must take to ensure that

$$\|f - l\|_{\infty,[0,1]} \le 10^{-3}.$$

Check that this works in practice.

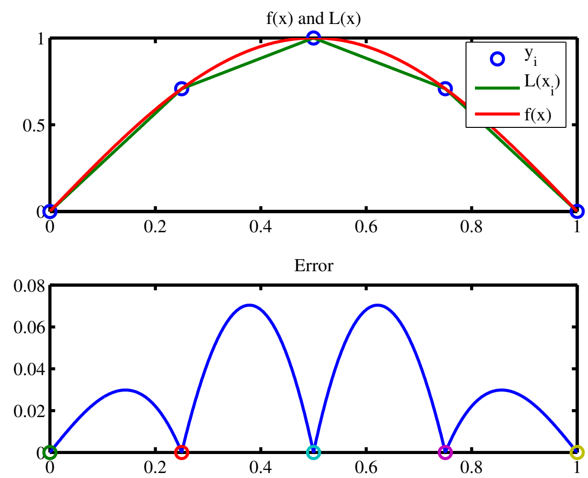

Figure 1: The piecewise linear interpolant to $\sin(\pi x)$ on $[0,1]$ with $n = 4$.

4 Cubic splines

In the file `lab2.m` if you change the line

```
L(i) = lspline(x,y, X(i));
```

to

```
S(i) = spline(x,y, X(i));
```

then a built-in MATLAB function will be used to compute the natural cubic spline interpolant to f. You also need to change `L` to `S` elsewhere in the code, as appropriate.

Try this, and try to verify that, as stated in Theorem 2.2.3, the error is proportional to h^4. That is, if we suppose that there is a constant, C, such that

$$\|f - S\|_{\infty,[0,1]} \leq Ch^4,$$

use your code to give an estimate for C. Again, use this value of C to estimate what value of n you must take to ensure that

$$\|f - S\|_{\infty,[0,1]} \leq 10^{-3}.$$

5 PCHIP

In Section 2.3 of NA2 we studied a simpler cubic spline interpolation called *piecewise cubic hermite interpolating polynomial (PCHIP)*. On each interval it $[x_{i-1}, x_i]$ it gives the cubic polynomial

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

where

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

Note that

$$S(x_{i-1}) = f(x_{i-1}), \quad S(x_i) = f(x_i),$$

and

$$S'(x_{i-1}) = f'(x_{i-1}), \quad S'(x_i) = f'(x_i).$$

Like piecewise linear interpolation, it is "local".

Write a function similar to `lspline.m` that computes the PCHIP at a given point:

```
function S = pchip_interp(x,f(x),df(x),X)
```

Note that it takes as one of its arguments the vector `df(x)` that contains the values of the derivative of f at the knot points.

Verify that the error for this method is proportional to h^4.

(There is also a built-in "`pchip`" function in MATLAB, but don't use it: its accuracy is proportional only to h^2).

....

At the end of the lab, upload the following to the "`Lab 2`" of the Blackboard module:

1. Your estimate for C for the linear spline.
2. Your estimate for C for the natural cubic spline.
3. Your code to compute the PCHIP interpolant.
4. Your estimate for C for computing PCHIP.

Give your `pchip` files a sensible name, and which includes your name or ID number. Both your name and ID number should appear as comments at the start of the file, along with a brief description of what the program does, and how. The estimates for C can be recorded as comments.

Deadline: Monday 27 Feb (but you should be able to complete this before leaving the lab).