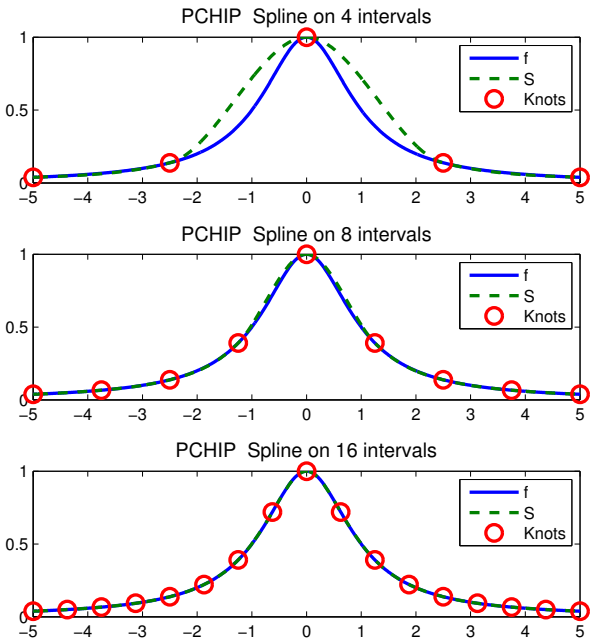
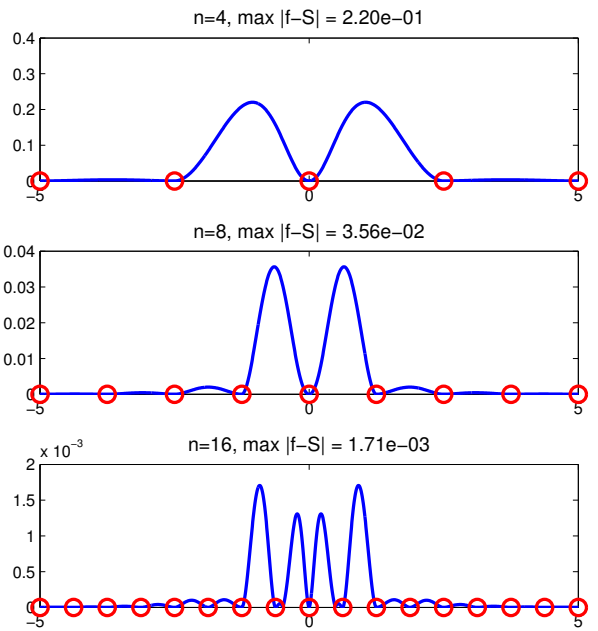
MA378 Chapter 2: Splines

§2.3 The PCHIP Interpolant

Dr Niall Madden

Start: 20 February 2026 (W05.2)





3.0 This section:

1 Introduction
- An Application

2 PCHIP

3 Error Estimate

4 Wrap-up
- Estimating derivatives
- Further reading
- Next, in NA2

5 Exercises

3.1 Introduction

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.

PCHIPs are harder to construct than linear splines.
And need information on f', unlike natural splines. And they are less smooth.
However, they are superior to them in ways too:

1. They enjoy the same max-norm stability property as linear splines, meaning they don't suffer from spurious oscillations;
2. The are smoother than linear splines, and have curvature.

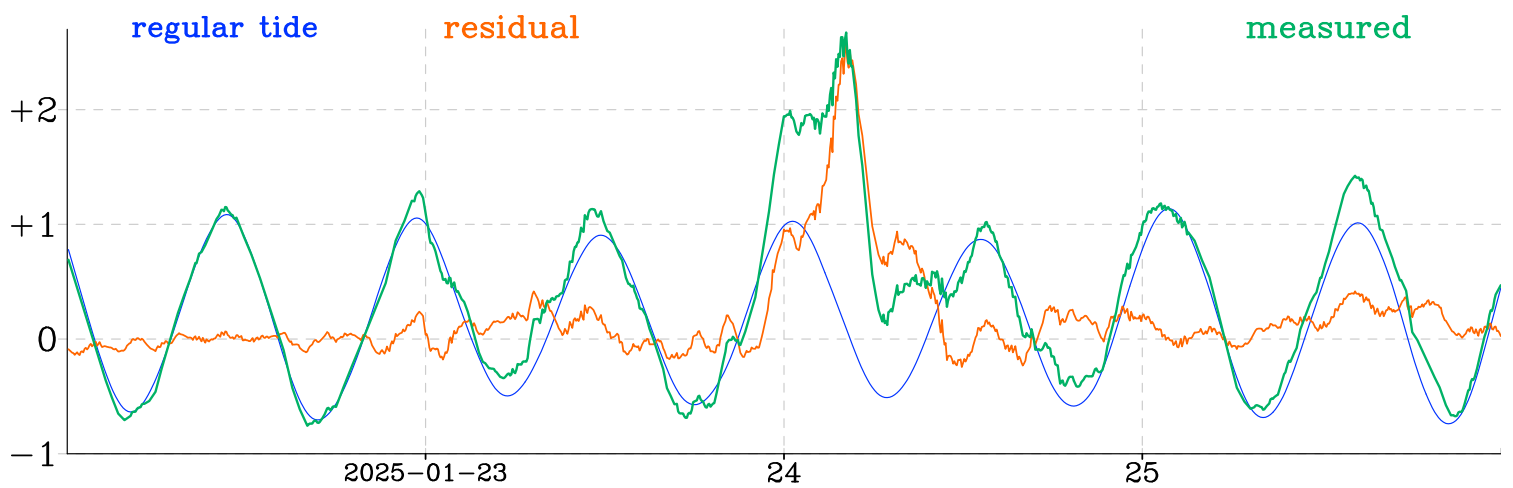
PCHIP splines have lots of applications. Here is one that I was involved recently...

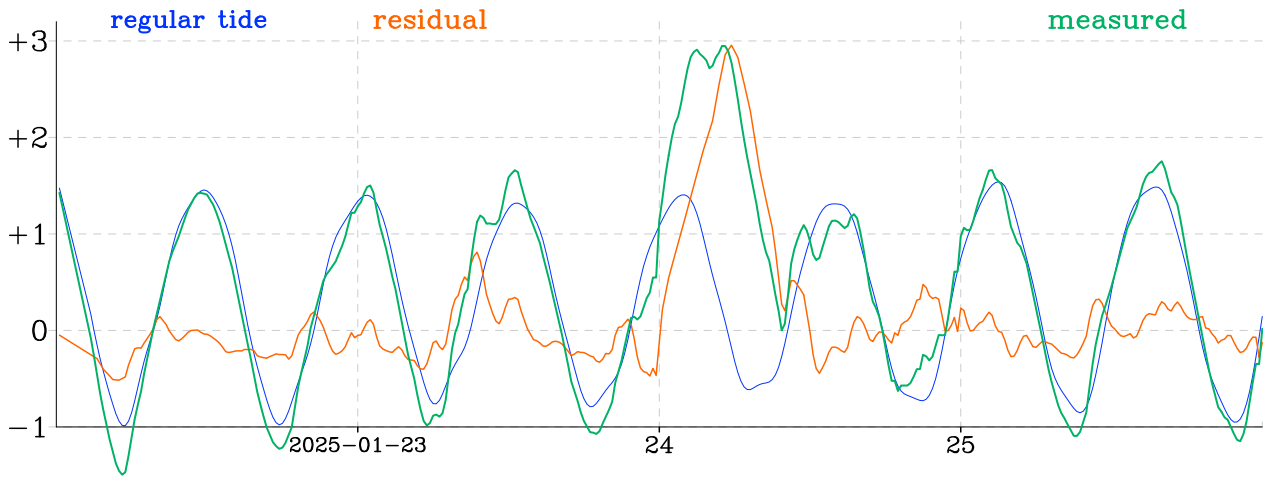
How to estimate a tidal storm surge.

Storm Éowyn was one of the strongest ever to hit Ireland. It also cause a record tidal surge in Galway and Limerick.

- Flooding was minimal because the storm hit the west coast on a low and eddbing tide.
- We can still estimate the size of the surge.

Tides are **very** regular and predicable. If we have a good predication, then the difference between it and the observed water height is caused by weather.
I had the good fortune to work with the amazing Indie Olbert and Sasha Shchepetkin on a project where we did that.
One of the key steps is using **PCHIPs** to interpolate the data.
(Next slides are just images, explained in class. No, they will not be on the exam).


regular tide
residual
measured
+2
+1
0
−1
2025−01−23
24
25


regular tide
residual
measured
+3
+2
+1
0
−1
2025−01−23
24
25

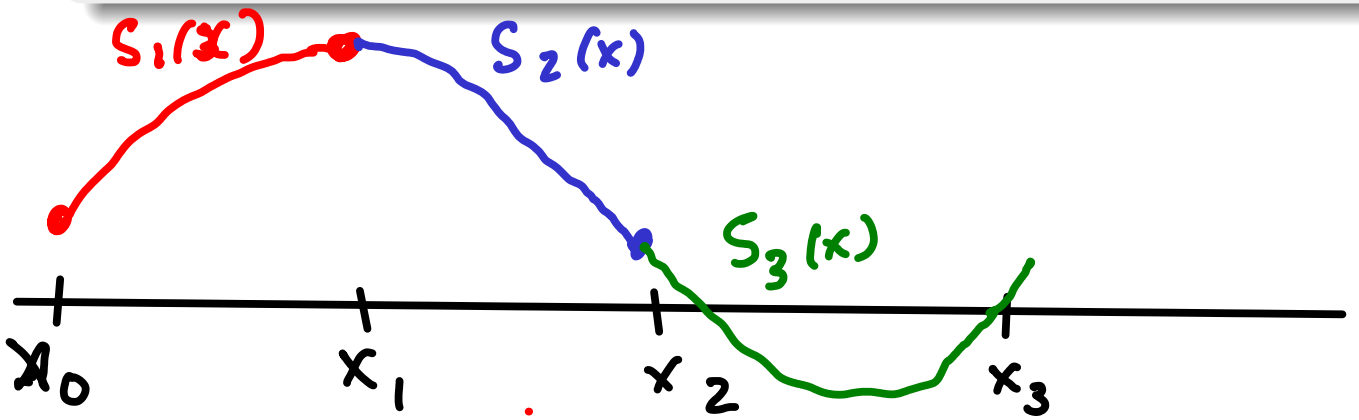
Definition 3.1 (Piecewise Cubic Hermite Spline Interpolant)

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) Both S and S' are continuous on $[x_0, x_N]$.

(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.



3.2 PCHIP

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 and c_3.

Note: Each S_i has to satisfy 4 conditions:

$S_i(x_{i-1}) = f_{i-1}$

$S_i(x_i) = f_i$

$S_i'(x_{i-1}) = f'_{i-1}$

$S_i'(x_i) = f'_i$

$f_i := f(x_i)$

$f'_i := f'(x_i)$

3.2 PCHIP

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 and c_3.

First use that $S_i(x_{i-1}) = f_{i-1}$

So $S_i(x_{i-1}) = c_0 + c_1(x_{i-1} - x_{i-1}) + c_2(x_{i-1} - x_{i-1})^2 + c_3(x_{i-1} - x_{i-1})^3 = f_{i-1}$

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

3.2 PCHIP

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 and c_3.

Next, Note that

$$S_i'(x) = c_1 + 2c_2(x - x_{i-1}) + 3c_3(x - x_{i-1})^2$$

$$\Rightarrow S_i'(x_{i-1}) = c_1. \quad \text{Since } S_i'(x_{i-1}) = f'_{i-1}$$

we get $\boxed{c_1 = f'_{i-1}}$.

3.2 PCHIP

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 and c_3.

Next we use that $S_i(x_i) = f_i$

and $x_i - x_{i-1} = h$

Then $S_i(x_i) = c_0 + c_1 h + c_2 h^2 + c_3 h^3$

$\Rightarrow$ $f_{i-1} + h f'_{i-1} + c_2 h^2 + c_3 h^3 = f_i$

3.2 PCHIP

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 and c_3.

Similarly we get from $S_i'(x_i) = f_i'$

thus

$$S_i'(x) = c_1 + 2c_2(x - x_{i-1}) + 3c_3(x - x_{i-1})^2 = f_i'$$

$$\Rightarrow f_{i-1}' + 2c_2 h + 3c_3 h^2 = f_i'$$

3.2 PCHIP

3.2 PCHIP

This gives that

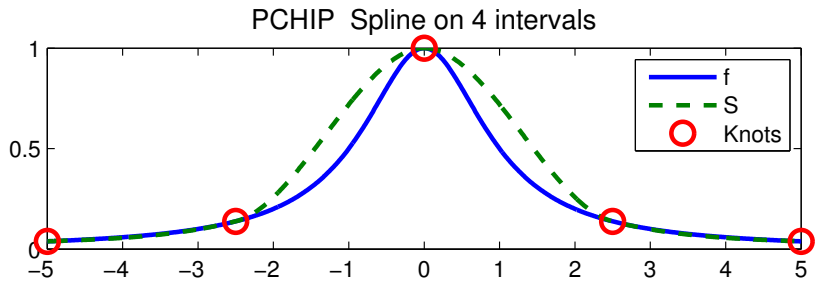
$$c_0 = f_{i-1}, \qquad c_1 = f'_{i-1},$$
$$c_2 = \frac{3}{h^2}(f_i - f_{i-1}) - \frac{1}{h}(f'_i + 2f'_{i-1}),$$
$$c_3 = \frac{1}{h^2}(f'_i + f'_{i-1}) - \frac{2}{h^3}(f_i - f_{i-1}).$$

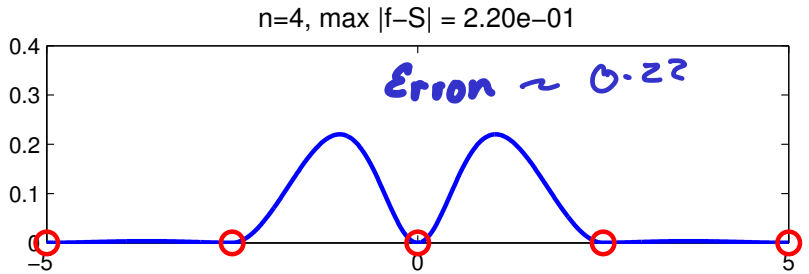
Exer: check that the equations on the previous slide give this.

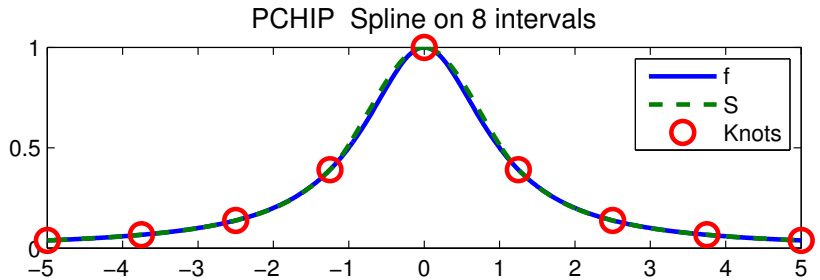
Also: the S_i are locally defined.

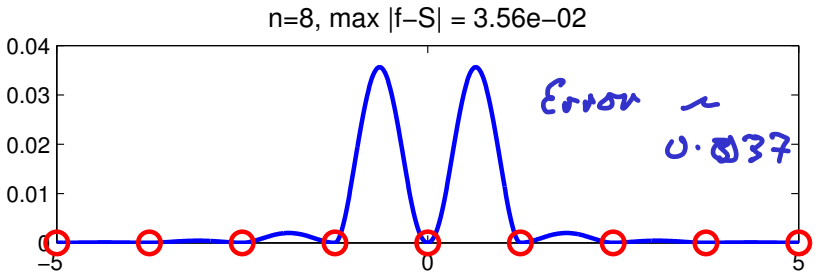
3.2 PCHIP

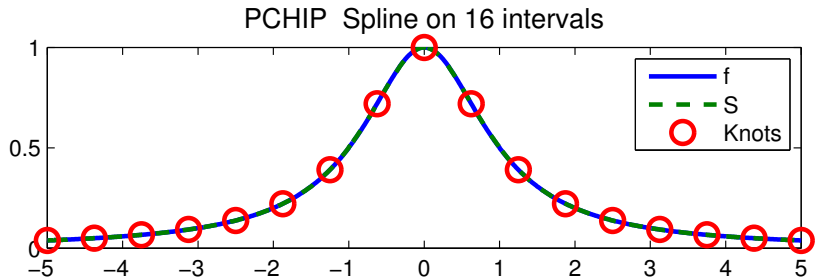
The figure below shows some PCHIP interpolants to $f(x) = 1/(1+x^2)$ on the interval $[-5, 5]$.

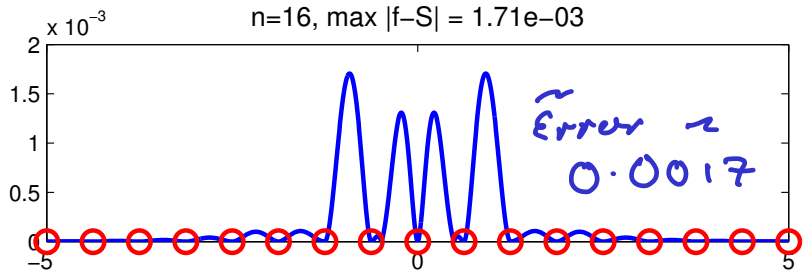












3.3 Error Estimate

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 3.2

Let $f \in C^4[x_0, x_N]$ and let S be the **Hermite Cubic Spline** *interpolant to f at the $N+1$ 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: On each subinterval, S_i is just the usual Hermite interpolant with $n = 1$. So

$$|f(x) - S_i(x)| \le \frac{\|f^{(4)}\|_\infty}{4!} \max \left((x - x_{i-1})(x - x_i) \right)^2$$

3.3 Error Estimate

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 3.2

Let $f \in C^4[x_0, x_N]$ and let S be the **Hermite Cubic Spline** *interpolant to f at the $N+1$ equally spaced points $a = x_0 < x_1 < \cdots < x_N = b$. Then*

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

we know that

$$\max_{x_{i-1} \le x \le x_i} (x - x_i)(x - x_{i-1}) = \frac{h^2}{4}$$

so

$$\max \left((x - x_i)(x - x_{i-1})\right)^2 = \frac{h^4}{16}.$$

3.4 Wrap-up

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. It is only 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, most libraries constructs these interpolants only requires f, not f'. Examples: `pchip` the MATLAB or `PchipInterpolator()` in *`Python/scipy`*. *How do they do this?*

It turns out that there are important variants on the PCHIP 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.,

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

The reality is a little more complicated. For example, one could approximate

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

or

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

In practice one takes an (harmonic) mean of these two values, if they have the same sign, and zero otherwise.

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.

2. **Solution of differential equations**

Finished here

3.5 Exercises

Exercise 3.1

Recall Exercise 2.3. 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 3.2

Let $f(x) = \ln(x) - x^4$. 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) $\max_{1 \le x \le 2} |f(x) - l(x)| \le 10^{-4}$?

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

Exercise 3.3

There are ways of constructing the PCHIP, other than that shown in (1) of Section 2.2. 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.