Computing derivatives by Polynomial fitting

Another way to approximate derivatives is to fit a polynomial to nearby points and differentiate that.

If we fit a line to \( \frac{x-h}{f(x-h)} \), we clearly get the central approximation

\[
P_2(x) = f(x_0) + f[x_0, x+h] (x-x_0)
\]

and

\[
f'(x) \approx P_2'(x) = f[x_0, x+h]
\]

and

\[
f'(x_0) \approx P_2'(x_0) = f[x_0, x_0+h] = \frac{f(x_0+h) - f(x_0)}{h}
\]
Now suppose we interpolate on

\[ x_0 \quad x_1 \quad x_2 \]
\[ x-h \quad x \quad x+h \]

If we use \( x_0, x_2 \) alone, we get

\[
P_1(x) = f(x_0) + f[x_0, x_2](x-x_0)
\]

\[
P'_1(x) = f[x_0, x_2] = \frac{f(x+h)-f(x-h)}{2h}
\]

Now if we interpolate all 3 nodes, we get

\[
P_2(x) = f(x_0) + f[x_0, x_1](x-x_0) + f[x_0, x_1, x_2](x-x_1)(x-x_0)
\]

\[
P'_2(x) = f[x_0, x_1] + f[x_0, x_1, x_2](2x-x_0-x_1).
\]

we note that

\[
P'_2(x_2) = f[x_0, x_1] = \frac{f(x+h)-f(x-h)}{2h}
\]

which is another way of seeing this formula is better than the first one.
So how do we estimate the error in these formulae?

If we let \( \omega(x) = \prod_{i=0}^{n} (x-x_i) \), then we recall that our exact formula

\[
f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \omega(x)
\]

involves an \((n+1)\)st derivative at some point in \((x_0, x_n)\). We can differentiate to get

\[
f'(x) - p'_n(x) = \frac{1}{(n+1)!} f^{(n+2)}(\xi) \omega'(x)
\]

\[+ \frac{1}{(n+1)!} \omega(x) \frac{d}{dx} \left[ f^{(n+1)}(\xi) \right].
\]

After all, \( \xi \) depends on \( x \), so this term is not always zero.
It is pretty clear that this formula can only be of help at a node, where we get

\[ f'(x_i) = p'_n(x_i) + \frac{1}{(n+1)!} f^{(n+1)}(\xi) \omega'(x_i). \]

For example, for the two node approximation, we get

\[ f'(x_0) = p'_1(x_0) + \frac{1}{2} f''(\xi) \]

\[ = f[x_0, x_2] + \frac{1}{2} f''(\xi) \]

which is the Taylor theorem error bound from before.
A useful case of this is

\[
\begin{array}{cccccc}
x_2 & x_0 & x_1 & x_3 \\
\hline
x-2h & x-h & x & x+h & x+2h
\end{array}
\]

We fit

\[
P_3(x) = f[x_0] + f[x_0, x_1](x-x_0) \\
+ f[x_0, x_1, x_2](x-x_0)(x-x_2) \\
+ f[x_0, x_1, x_2, x_3](x-x_0)(x-x_1)(x-x_2).
\]

So

\[
P_3'(x) = f[x_0, x_1] + f[x_0, x_1, x_2](2x-x_0-x_0) \\
+ f[x_0, x_1, x_2, x_3](x-x_0)(x-x_0) + (x-x_0)(x-x_2) \\
+ (x-x_1)(x-x_2)
\]

If we start actually plugging stuff in, at \( x \)

we get

\[
= f[x_0, x_1] + f[x_0, x_1, x_2](h-h) \\
+ f[x_0, x_1, x_2, x_3](h(-h) + h(2h) + (-h)(2h))
\]
\[ f'(x) \approx \frac{1}{2h} [f(x+h) - f(x-h)] - \frac{1}{12h} [f(x+2h) - 2f(x+h) - 2f(x-h) + f(x-2h)] \]

This is a 3rd degree polynomial approx, so the error is \( O(h^4) \). How does this compare to the corresponding Richardson extrapolation \( O(h^4) \) formula? (derivative_and_difference

\[ \]
What about second derivatives?

It's worth noting that we can extract second derivatives as well. Recall

\[ f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2} + \frac{f'''(x)h^3}{6} + \ldots \]

\[ f(x-h) = f(x) - f'(x)h + \frac{f''(x)h^2}{2} - \frac{f'''(x)h^3}{6} + \ldots \]

If we add, we get

\[ \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} = f''(x) + \frac{f'''(x)h^2}{12} + \ldots \]

Of course we have to check this numerically (note that we could use Richardson extrapolation to help here, too).