MAT360 Section Summary: 4.1 (part a)

Numerical Differentiation

1. Summary

In this section we use various schemes for approximating derivatives, using discrete points, starting from the first-order divided difference approximation

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}$$

This is a **two-point** formula, for the approximation, relying on points x_0 and $x_0 + h$. If we can use two points, can't we use three to get even better approximations? Of course we can!

2. **Definitions**

- Given
$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}$$
forward-difference formula: $h > 0$; **backward-difference formula**: $h < 0$.

- **centered-difference formula**: works out to the average of the forward- and backward-difference formulas:
$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}$$

3. **Theorems/Formulas**

But what error are we making in that approximation? Well, if f is twice differentiable, then this approximation will fall out of the Lagrange interpolating polynomial and its error term. Consider two points x_0 and x_1, and the linear Lagrange interpolating polynomial. Define $h = x_1 - x_0$. Then

$$f(x) = P_{0,1}(x) + f''(\xi(x))\frac{(x-x_0)(x-x_1)}{2!}$$

Then

$$f'(x) = P'_{0,1}(x) + \frac{d}{dx}f''(\xi(x))\frac{(x-x_0)(x-x_1)}{2!} + f''(\xi(x))\frac{d}{dx}\left[\frac{(x-x_0)(x-x_1)}{2!}\right]$$

The Newton form of the interpolating polynomial (which is equivalent to the Lagrange interpolating polynomial, remember!) gives us the derivative as the divided difference, and then we have to use the product rule to produce the mess with the rest:

$$f'(x) = \frac{f(x_0+h) - f(x_0)}{h} + \frac{d}{dx}f''(\xi(x))\frac{(x-x_0)(x-x_1)}{2!} + f''(\xi(x))\frac{2(x-x_0)-h}{2}$$

When $x = x_0$ we get some nice simplification: we have that

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - f''(\xi(x_0))\frac{h}{2}$$

so, in general, the forward (or backward) difference approximations have errors that satisfy

$$|f'(x_0) - \frac{f(x_0+h) - f(x_0)}{h}| \leq \frac{M|h|}{2}$$

where $M > 0$ is a bound on the size of the second derivative on the interval $[x_0, x_1]$.

4. Properties/Tricks/Hints/Etc.

We can get the error bound for the centered-difference formula using Taylor series quite easily, provided f is thrice-differentiable:

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{3!}f'''(\xi(x_0))$$

and

$$f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{3!}f'''(\phi(x_0))$$

Then the centered-difference formula yields

$$\frac{f(x_0+h) - f(x_0-h)}{2h} = f'(x_0) + \frac{h^2}{2 \cdot 3!}(f'''(\xi(x_0)) + f'''(\phi(x_0)))$$

and, provided f''' is continuous, we can find a $\psi(x_0) \in [x_0 - h, x_0 + h]$ such that

$$f'''(\psi(x_0)) = \frac{(f'''(\xi(x_0)) + f'''(\phi(x_0)))}{2}$$

so that

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} - f^{(3)}(\psi(x_0))\frac{h^2}{6}$$

Marvellous! Don't you love that Taylor formula?

Round-off versus Step-size – Fight of the Century!

Even though we pretend that we are calculating "real" values, we're making errors (truncation, round-off). Consider the case of the forward-difference formula. In this case, we compute with errors:

$$\tilde{f}(x_0) = f(x_0) - e(x_0)$$

and

$$\tilde{f}(x_0+h) = f(x_0+h) - e(x_0+h)$$

where e represents round-off error, and $h > 0$.

Now the absolute error E in our derivative calculation will be made up of two parts:

$$E = \left| f'(x_0) - \tfrac{\tilde{f}(x_0+h) - \tilde{f}(x_0)}{h} \right|$$

$$= \left| f'(x_0) - \tfrac{f(x_0+h) - e(x_0+h) - f(x_0) + e(x_0)}{h} \right|$$

$$= \left| f'(x_0) - f'(x_0) + \tfrac{h}{2}f''(\xi) + \tfrac{e(x_0+h) - e(x_0)}{h} \right|$$

so

$$E = \left| \frac{h}{2} f''(\xi) + \frac{e(x_0 + h) - e(x_0)}{h} \right| \leq \frac{Mh}{2} + \frac{2\epsilon}{h}$$

where $M > 0$ is a bound on the second derivative on the interval of interest, and $\epsilon > 0$ is a bound on the size of a truncation or round-off error.

The upshot: we can't just make h as small as we like, and expect approximations to get better and better: round-off error will ultimately kill us. We need to balance the round-off against small h – and we can even guess what value of h is appropriate, given a particular function (and its second derivative), and the size of round-off errors on your particular machine.

Example: #24, p. 178