Lecture 4: Derivates

Numerical derivatives

- Simpler than numerical integration, in a way.
- Computing errors is usually a doozey though.
- Based on Taylor series approximations.

1. Forward difference approximation: $f'(x) \approx \dfrac{f(x+h) - f(x)}{h}$,
2. Backward difference approximation: $f'(x) \approx \dfrac{f(x) - f(x-h)}{h}$,

Estimation of approximation error

Use Taylor series to find error in these approximations:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + h.o.t.$$

Isolate for $f'(x)$:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) + h.o.t.$$

⇒ error is 1st-order in h (same is true for backward difference method).

Central differences

- Using Taylor series to find sneaky improvements to finite difference (FD) schemes.
- Example: central FD method:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

- Notice it still only involves subtracting 2 points, it's just that the location of the 2 points is different.

- Error:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + h.o.t.$$
$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + h.o.t.$$

- Subtract:

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3}f'''(x) + h.o.t.$$

- Isolate for $f'(x)$ and add:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \boxed{\frac{h^2}{3}f'''(x)} + h.o.t.$$

Slide Type Sub-Slide

- So we see that this formula is accurate to 2nd order in h.
- Can get higher-order methods by including more points (see table 5.1 on page 196).
- Might have to do different things near the boundaries
- Partial derivatives: similar techniques
- Higher order derivatives (e.g. f''): also similar techniques.

Slide Type Sub-Slide

Roundoff error

- Let's take another look at this formula:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + h.o.t.$$

- What happens when we consider roundoff error? Recall that subtracting numbers that are close to each other in value, and yield an answer close to 0, is dangerous!
- Each of the terms $f(x+h)$ and $f(x)$ have error $\sim C|f(x)|$. Their difference will have approximate error $2Cf(x)$ ("worst case" error).
- So in fact there are two sources of error and this leads to (eqn. (5.91) in book):

$$\epsilon = \underbrace{\frac{2C|f(x)|}{h}}_{\text{round-off error}} + \underbrace{\frac{1}{2}h|f''(x)| + h.o.t.}_{\text{approximation error}}$$

Slide Type Sub-Slide

- There's a limit to the improvement you can obtain by going to finer resolution
- The precision expected on differentiation is orders of magnitude less than that of other operations we have discussed.

Slide Type

Interpolation

Slide Type

Linear

Slide Type

Assume the function follows a straight line from $f(a)$ to $f(b)$. Estimate $f(x)$ for $a < x < b$.

Slide Type

$f(x) \approx y + z\ ,\ y = m(x-a)$

Therefore:

$f(x) \approx \frac{f(b)-f(a)}{b-a}(x-a) + f(a)$

$= \frac{(b-x)f(a)+(x-a)f(b)}{b-a}$ fundamental formula of linear interpolation

- Can also use this to extrapolate the function to points $x > b$ or $x < a$
- But the further you go, the less likely it is that the extrapolation will be accurate

Slide Type

Error

Slide Type

- Leading-order term in approximation error: $(a-x)(b-x)f''(x)$
  - Vanishes as x approaches a or b
  - Largest in the middle of the interval, assuming $f''(x)$ varies slowly
  - Worst-case error is $O(h^2)$

Slide Type

- Rounding error not a big problem, since fundamental formula involves sum (not difference) of function values at two closely spaced points

Slide Type

More complicated interpolations

Slide Type

- Lagrange interpolation methods: use quadratics or higher polynomials
- To use polynomial of degree n, we need to know the value of the function at n+1 or more points
- For large n, using (n-1)th order polynomial doesn't work very well
  - because very high order polynomials tend to have a lot of wiggles, therefore can deviate badly in the intervals between points
  - better to use many quadratics or cubics on smaller sets of adjacent points
    - but slope changes abruptly at the join-points between polynomials, resulting in uneven interpolation
  - even better to use splines: use the derivatives of the polynomials at the join-points, so the interpolation has smooth slope everywhere
    - cubic spline is most commonly used