2. Numerical differentiation.

Approximate a derivative of a given function.

Approximate a derivative of a function defined by discrete data at the discrete points.

Formulas for numerical differentiation can be derived from a derivative of the (Lagrange form of) interpolating polynomial.

.

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

$$p_n(x) = \sum_{i=0}^{n} L_{n,i}(x)\, f_i \ , \qquad L_{n,i}(x) := \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Exc 2-0) Derive the form of finite difference formula for the first derivative, starting from a) Lagrange form, and b) Newton form.

ex) The second derivative f ''(x) using 3 data points, x_{-1}, x_0, x_1.

.

$$f(x) = p_2(x) + \frac{f^{(3)}(\xi)}{3!}(x - x_{-1})(x - x_0)(x - x_1)$$

$$p_2(x) = \frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} f_{-1} + \frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} f_0 + \frac{(x - x_{-1})(x - x_0)}{(x_1 - x_{-1})(x_1 - x_0)} f_1$$

$$\begin{aligned} f''(x) = p_2''(x) \;&+\; \frac{f^{(3)}(\xi)}{3!}(3x - x_{-1} - x_0 - x_1) && \approx O(\Delta x) \\ &+\; 2\frac{d}{dx}\left[\frac{f^{(3)}(\xi)}{3!}\right]\frac{d}{dx}\left[(x - x_{-1})(x - x_0)(x - x_1)\right] && \approx O(\Delta x^2) \\ &+\; \frac{d^2}{dx^2}\left[\frac{f^{(3)}(\xi)}{3!}\right](x - x_{-1})(x - x_0)(x - x_1) && \approx O(\Delta x^3) \end{aligned}$$

$$p_2''(x) = \frac{2}{(x_{-1} - x_0)(x_{-1} - x_1)} f_{-1} + \frac{2}{(x_0 - x_{-1})(x_0 - x_1)} f_0 + \frac{2}{(x_1 - x_{-1})(x_1 - x_0)} f_1$$

For $x = x_0$, and equally spaced x_i, $O(\Delta x)$ error cancel out.

Exc 2-1) Derive the following finite difference formulas for the first derivative $f'(x)$ with equally spaced abscissas.

a) Second order backward, forward, and centered formula.

b) Third order difference formula..

c) Fourth order centered difference formula..

Exc 2-2) Evaluate the error of finite difference formula for the second derivative $f''(x)$ which uses equally spaced 5 data points .

Exc 2-3) Compute numerical derivative of e^x using 2nd and 4th order centered formulas at x=1 with equally spaced abscissas. Decreasing the spacing Δx, as 10^{-n}, check the fractional error converges as expected $O(\Delta x^2)$, and $O(\Delta x^4)$. (Make a plot.) What happens if n varies from 1 to 32 ?

- Richardson extrapolation: A procedure to obtain higher order approximation

Notation: (the same) order $O(\Delta x^k)$; small (higher) order $o(\Delta x^k)$

$f(\Delta x) = O(\Delta x^k)$ stands for

$\exists L > 0$, a constant, such that $\left|\frac{f(\Delta x)}{\Delta x^k}\right| \leq L$ for sufficiently small Δx.

$f(\Delta x) = o(\Delta x^k)$ is used for $\lim_{\Delta x \to 0} \left|\frac{f(\Delta x)}{\Delta x^k}\right| = 0.$

Now, suppose f(x) is approximated by $f_{\Delta x}$ at $x = x_0$, and its order is p-th order, we may write, with a constant K_1

$$f_{\Delta x} = f(x_0) + K_1 \Delta x^p + o(\Delta x^p)$$

If we decresed the size of interval Δx to Δx / b, (b>1 const), we have

$$f_{\Delta x/b} = f(x_0) + K_1 \Delta x^p / b^p + o(\Delta x^p)$$

Substituting K_1 from one to the other, we have higher order approximation

$$\frac{b^p f_{\Delta x/b} - f_{\Delta x}}{b^p - 1} = f(x_0) + o(\Delta x^p)$$

(Commonly b = 2 is chosen. (recommended))

- To apply Richardson extrapolation, the order of approximation formula should be known.
- If this order is know towards higher ones, one can repeatedly use the extrapolation to have higher order approximation.

$$D_{\Delta x} = D(x_0) + K_1 \Delta x^2 + K_2 \Delta x^4 + o(\Delta x^4)$$

$$D_{\Delta x/2} = D(x_0) + K_1 \frac{\Delta x^2}{4} + K_2 \frac{\Delta x^4}{16} + o(\Delta x^4)$$

$$D^{(1)}_{\Delta x} := \frac{4D_{\Delta x/2} - D_{\Delta x}}{3} = D(x_0) + K_2' \Delta x^4 + o(\Delta x^4)$$

$$D^{(1)}_{\Delta x/2} := \frac{4D_{\Delta x/4} - D_{\Delta x/2}}{3} = D(x_0) + K_2' \frac{\Delta x^4}{16} + o(\Delta x^4)$$

$$D^{(2)}_{\Delta x} := \frac{16 D^{(1)}_{\Delta x/2} - D^{(1)}_{\Delta x}}{15} = D(x_0) + o(\Delta x^4)$$

Exc 2-4) a) For the first order forward difference approximaion, apply Richardson extrapolation to O(Δx^3).

b) Write a code to apply this to calculate a derivative of f(x)=log x at x = 2. Choose a step size Δx = 0.1, and make it 1/2 at each level of extrapolation.