To see why: Say r is a root w/ multiplicity m of $f(x)$:

$$f(r) = f'(r) = \cdots = f^{(m-1)}(r) = 0, \quad f^{(m)}(r) \neq 0$$

(from the defin.)

Consequently:

- $f(r+\epsilon_k) = \cancel{f(r)} + \cancel{f'(r)\epsilon_k} + \cancel{\frac{f''(r)}{2!}\epsilon_k^2} + \cdots + \cancel{\frac{f^{(m-1)}(r)}{(m-1)!}\epsilon_k^{m-1}} + \frac{f^{(m)}(r)}{m!}\epsilon_k^m + O(\epsilon_k^{m+1})$

$$= \frac{f^{(m)}(r)}{m!}\epsilon_k^m + \frac{f^{(m+1)}(r)}{(m+1)!}\epsilon_k^{m+1} + O(\epsilon_k^{m+2})$$

- $f'(r+\epsilon_k) = \cancel{f'(r)} + \cancel{f''(r)\epsilon_k} + \cdots + \cancel{\frac{f^{(m-1)}(r)}{(m-2)!}\epsilon_k^{m-2}} + \frac{f^{(m)}(r)}{(m-1)!}\epsilon_k^{m-1} + O(\epsilon_k^m)$

$$= \frac{f^{(m)}(r)}{(m-1)!}\epsilon_k^{m-1} + \frac{f^{(m+1)}(r)}{m!}\epsilon_k^m + O(\epsilon_k^{m+1})$$

So the Newton's iteration formula (in terms of ϵ_k's) is now written as:

$$\epsilon_{k+1} = \epsilon_k - \frac{m f(r+\epsilon_k)}{f'(r+\epsilon_k)}$$

$$= \epsilon_k - \frac{m\left[\frac{f^{(m)}(r)}{m!}\epsilon_k^m + \frac{f^{(m+1)}(r)}{(m+1)!}\epsilon_k^{m+1} + O(\epsilon_k^{m+2})\right]}{\frac{f^{(m)}(r)}{(m-1)!}\epsilon_k^{m-1} + \frac{f^{(m+1)}(r)}{m!}\epsilon_k^{m} + O(\epsilon_k^{m+1})}$$

confirm (fill in the details)

$$\frac{m}{m}\epsilon_k = \epsilon_k$$

$$= \epsilon_k - \frac{1}{m}\epsilon_k + O(\epsilon_k^2) = \frac{m-1}{m}\epsilon_k + O(\epsilon_k^2)$$

$$\therefore \frac{\epsilon_{k+1}}{\epsilon_k^1} \to \frac{m-1}{m} \in (0,1)$$

(linear convergence)

$$\epsilon_{k+1} = * \epsilon_k^2 + O(\epsilon_k^3)$$

$$\therefore \frac{\epsilon_{k+1}}{\epsilon_k^2} \to *$$

as $k \to \infty$

(quad. conv.)

Calculating nth Roots

Question. Let n be a positive integer. Use Newton's method to produce a quadratically convergent method for calculating the nth root of a positive number a. Prove quadratic convergence.

Given $n \in \mathbb{N}$, $a > 0$, calculate $\sqrt[n]{a}$ using Newton.

Note that $\sqrt[n]{a}$ is a root of

$$x^n = a$$

So, it is a root of

$$f(x) = x^n - a$$

$$\boxed{x_{k+1} = \frac{n-1}{n} x_k + \frac{a}{n x_k^{n-1}}}$$

e.g. $(n=2)$

$$x_{k+1} = \frac{1}{2} x_k + \frac{a}{2 x_k}$$

Newton

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^n - a}{n x_k^{n-1}} = \frac{n-1}{n} x_k + \frac{a}{n x_k^{n-1}}$$

Predicting Next Error

$x(x^2-4) = x(x-2)(x+2)$

Question. Let $f(x) = x^3 - 4x$. =

(a) • The function $f(x)$ has a root at $r = 2$. If the error $\epsilon_k = x_k - r$ after four steps of Newton's method is $\epsilon_4 = 10^{-6}$, estimate ϵ_5.

(b) • Do the same to the root $r = 0$. (Exercise)

Recall: Newton's method

$$\boxed{\epsilon_{k+1} = \frac{f'(r)}{2f''(r)} \epsilon_k^2 + O(\epsilon_k^3)}$$

(a) $r = 2$,

$$\begin{cases} f'(x) = 3x^2 - 4 \\ f''(x) = 6x \end{cases} \Rightarrow \begin{cases} f'(2) = 8 \\ f''(2) = 12 \end{cases}$$

So $\underline{\text{at } r=2}$

$$\epsilon_{k+1} = \frac{8}{24} \epsilon_k^2 + O(\epsilon_k^3)$$

$k=4$, $\epsilon_k = 10^{-6}$

$$\epsilon_5 = \frac{1}{3} 10^{-12} + O(10^{-18}) \approx \boxed{3.33 \times 10^{-11}}$$

Secant Method

Assume that iterates $x_1, x_2, \ldots$ generated by the secant method converges to a root r and $f'(r) \neq 0$. Let $\epsilon_k = x_k - r$.

Exercise.[1] Show that

1. The error ϵ_k satisfies the approximate equation

$$\left|\epsilon_{k+1}\right| \approx \left|\frac{f''(r)}{2f'(r)}\right| \left|\epsilon_k\right| \left|\epsilon_{k-1}\right|.$$

2. If in addition $\lim_{k\to\infty} \left|\epsilon_{k+1}\right| / \left|\epsilon_k\right|^{\alpha}$ exists and is nonzero for some $\alpha > 0$, then

$$\left|\epsilon_{k+1}\right| \approx \left|\frac{f''(r)}{2f'(r)}\right|^{\alpha-1} \left|\epsilon_k\right|^{\alpha}, \quad \text{where } \alpha = \frac{1+\sqrt{5}}{2}.$$

[1] This exercise is from Lecture 22.

<u>Hints</u> Error analysis for Secant Method.

- $x_k = r + \epsilon_k$. $\quad (\epsilon_k \to 0$ as $k \to \infty)$
- Taylor expansion
- big-O notation for simplification

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \approx f'(x_k)$$

Recall: iter. form. for secant method

$$x_{k+1} = x_k - \frac{(x_k - x_{k-1}) f(x_k)}{f(x_k) - f(x_{k-1})}$$

Lec ~~29~~ 30: Problem Solving Session 3

- MATLAB
  - interp1 (PL, cubic spline)
  - spline

- 2-D splines

Exercise with Piecewise Interpolation

B-splines

- piecewise linear (simplest : "connect the dots")
  * hat functions
- piecewise cubic (most common ; smooth outcome)
  * Hermite cubic interpolation
  * cubic spline.
    - not-a-knot B.C. (MATLAB's default)
    - clamped B.C. ("spline")

Derivation of Hermite Cubic Interpolation

Given (x_i, y_i, σ_i), for $i = 1, 2, \cdots, n$, find the piecewise cubic polynomial

$$\mathbb{P}(x) = \begin{cases} p_1(x), & [x_1, x_2) \\ p_2(x), & [x_2, x_3) \\ \vdots & \vdots \\ p_{n-1}(x), & [x_{n-1}, x_n] \end{cases}$$

where $p_i(x) = c_{i,1} + c_{i,2}(x-x_i) + c_{i,3}(x-x_i)^2 + c_{i,4}(x-x_i)^3$

satisfying

$$\mathbb{P}(x_i) = y_i \text{ and } \mathbb{P}'(x_i) = \sigma_i \text{ for all } i = 1, 2, \cdots, n.$$

$\boxed{\text{Strategy}}$: For each $i = 1, 2, \cdots, n-1$, determine

$$p_i(x) = c_{i,1} + c_{i,2}(x-x_i) + c_{i,3}(x-x_i)^2 + c_{i,4}(x-x_i)^3$$

such that

$$\begin{cases} ① : p_i(x_i) = y_i \\ ② : p_i'(x_i) = \sigma_i \end{cases} \qquad \begin{cases} ③ : p_i(x_{i+1}) = y_{i+1} \\ ④ : p_i'(x_{i+1}) = \sigma_{i+1} \end{cases}$$

$\boxed{\begin{array}{l} 4 \text{ eqns} \\ 4 \text{ unknowns} \end{array}}$

$\boxed{\text{Solution}}$ Easy to see from ① and ② that

$$\boxed{c_{i,1} = y_i, \qquad c_{i,2} = \sigma_i,}$$

Eqns ③ and ④ are written out as

$$\begin{cases} ③' : \quad y_i + \sigma_i \Delta x_i + c_{i,3} (\Delta x_i)^2 + c_{i,4} (\Delta x_i)^3 = y_{i+1} \\ ④' : \quad \sigma_i + 2 c_{i,3} \Delta x_i + 3 c_{i,4} (\Delta x_i)^2 = \sigma_{i+1} \end{cases}$$

where $\Delta x_i = x_{i+1} - x_i$. (Analogously, we use $\Delta y_i = y_{i+1} - y_i$ and define $y[x_i, x_{i+1}] = \Delta y_i / \Delta x_i$ in what follows.)

Rewrite:

③′:
$$c_{i,3} + \Delta x_i \, c_{i,4} = \frac{y_{i+1} - y_i - \sigma_i \Delta x_i}{(\Delta x_i)^2}$$

$$= \frac{\Delta y_i / \Delta x_i - \sigma_i}{\Delta x_i}$$

$$= \frac{y[x_i, x_{i+1}] - \sigma_i}{\Delta x_i}$$

④′:
$$c_{i,3} + \frac{3}{2}\Delta x_i \, c_{i,4} = \frac{\sigma_{i+1} - \sigma_i}{2\Delta x_i}$$

First, solve for $C_{i,4}$ by eliminating $C_{i,3}$:

$$④' - ③' : \quad \frac{1}{2}\Delta x_i \, C_{i,4} = \frac{(\sigma_{i+1} - \sigma_i) - 2(y[x_i, x_{i+1}] - \sigma_i)}{2\Delta x_i}$$

$$= \frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{2\Delta x_i}$$

$$\therefore \quad \boxed{C_{i,4} = \frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{(\Delta x_i)^2}}$$

Plugging this back into 4':

$$c_{i,3} = \frac{\sigma_{i+1} - \sigma_i}{2\Delta x_i} - \frac{3}{2}\cancel{\Delta x_i}\,\frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{(\Delta x_i)^{\cancel{2}}}$$

$$= \frac{(\sigma_{i+1} - \sigma_i) - 3(\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}])}{2\Delta x_i}$$

$$= \frac{6y[x_i, x_{i+1}] - 4\sigma_i - 2\sigma_{i+1}}{2\Delta x_i}$$

$$\therefore \boxed{c_{i,3} = \frac{3y[x_i, x_{i+1}] - 2\sigma_i - \sigma_{i+1}}{\Delta x_i}}$$

Cubic Spline by Hand

(Ignore)

- **FNC** 5.3.1

Cubic Spline with Periodic Boundary Conditions

- **FNC** 5.3.7
- Also see the appendix to Lecture 26 slides for various other boundary conditions.

- Try working out Exam 2 problem (fitting heartbeat data) using splines.

Error Analysis

(4^{th}-order accuracy) log-log graph

- **FNC** 5.3.4

- See the live script accompanying Lecture 26.