Solution of Nonlinear Equations

Roots of non-linear Equations:

$$f(x) = 0$$

f may be a function belonging to any class: algebraic, trigonometric, hyperbolic, polynomials, logarithmic, exponential, etc.

We will learn the following groups of methods:

- ✓ **Bracketing methods:** Bisection, Regula-Falsi
- ✓ **Open methods:** Fixed point, Newton-Raphson, Secant, Muller
- ✓ **Special methods for polynomials:** Bairstow

Background assumed (MTH 101): intermediate value theorem; nested interval theorem; Cauchy sequence and convergence; Taylor's and Maclaurin's series; etc.

Bracketing Methods

- ✓ Intermediate value theorem: Let f be continuous on $[a, b]$ and let $f(a) < s < f(b)$, then there exists at least one x such that $a < x < b$ and $f(x) = s$.
  - ✓ Bracketing methods are application of this theorem with $s = 0$
- ✓ Nested interval theorem: For each n, let $I_n = [a_n, b_n]$ be a sequence of (non-empty) bounded intervals of real numbers such that $I_1 \supset I_2 \supset \cdots I_n \supset I_{n+1} \supset \cdots$ and $\lim_{n\to\infty}(a_n - b_n) = 0$, then $\bigcap_{n=1}^{\infty} I_n$ contains only one point.
  - ✓ This guarantees the convergence of the bracketing methods to the root.
- ✓ In bracketing methods, a sequence of nested interval is generated such that each interval follows the *intermediate value theorem* with $s = 0$. Then the method converges to the root by the one point specified by the *nested interval theorem*. Methods only differ in ways to generate the nested intervals.

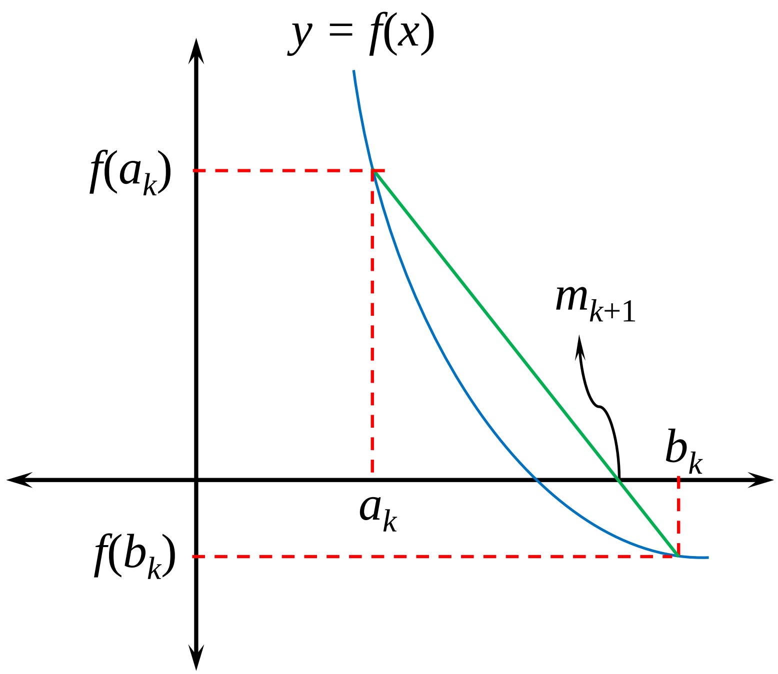
Bisection Method

- ✓ Principle: Choose an initial interval based on intermediate value theorem and halve the interval at each iteration step to generate the nested intervals.
- ✓ Initialize: Choose a_0 and b_0 such that, $f(a_0)f(b_0) < 0$. This is done by trial and error.
- ✓ Iteration step k:
  - ✓ Compute mid-point $m_{k+1} = (a_k + b_k)/2$ and functional value $f(m_{k+1})$
  - ✓ If $f(m_{k+1}) = 0$, m_{k+1} is the root. (It's your lucky day!)
  - ✓ If $f(a_k)f(m_{k+1}) < 0$: $a_{k+1} = a_k$ and $b_{k+1} = m_{k+1}$; else, $a_{k+1} = m_{k+1}$ and $b_{k+1} = b_k$
  - ✓ After n iterations: size of the interval $d_n = (b_n - a_n) = 2^{-n}(b_0 - a_0)$, stop if $d_n \le \varepsilon$
  - ✓ Estimate the root ($x = \alpha$ say!) as: $\alpha = m_{n+1} \pm 2^{-(n+1)}(b_0 - a_0)$

Regula-Falsi or Method of False Position

- ✓ Principle: In place of the mid point, the function is assumed to be linear within the interval and the root of the linear function is chosen.
- ✓ Initialize: Choose a_0 and b_0 such that, $f(a_0)f(b_0) < 0$. This is done by trial and error.
- ✓ Iteration step k:
  - ✓ A straight line passing through two points $(a_k, f(a_k))$ and $(b_k, f(b_k))$ is given by: $\frac{x-a_k}{f(x)-f(a_k)} = \frac{b_k-a_k}{f(b_k)-f(a_k)}$
  - ✓ Root of this equation at $f(x) = 0$ is: $x = m_{k+1} = a_k - \frac{b_k-a_k}{f(b_k)-f(a_k)} f(a_k)$
  - ✓ If $f(m_{k+1}) = 0$, m_{k+1} is the root. (It's your lucky day!)
  - ✓ If $f(a_k)f(m_{k+1}) < 0$: $a_{k+1} = a_k$ and $b_{k+1} = m_{k+1}$; else, $a_{k+1} = m_{k+1}$ and $b_{k+1} = b_k$
  - ✓ After n iterations: size of the interval $d_n = (b_n - a_n)$, stop if $d_n \leq \varepsilon$
  - ✓ Estimate the root ($x = \alpha$ say!) as: $\alpha = a_n - \frac{b_n-a_n}{f(b_n)-f(a_n)} f(a_n)$

Regula-Falsi or Method of False Position



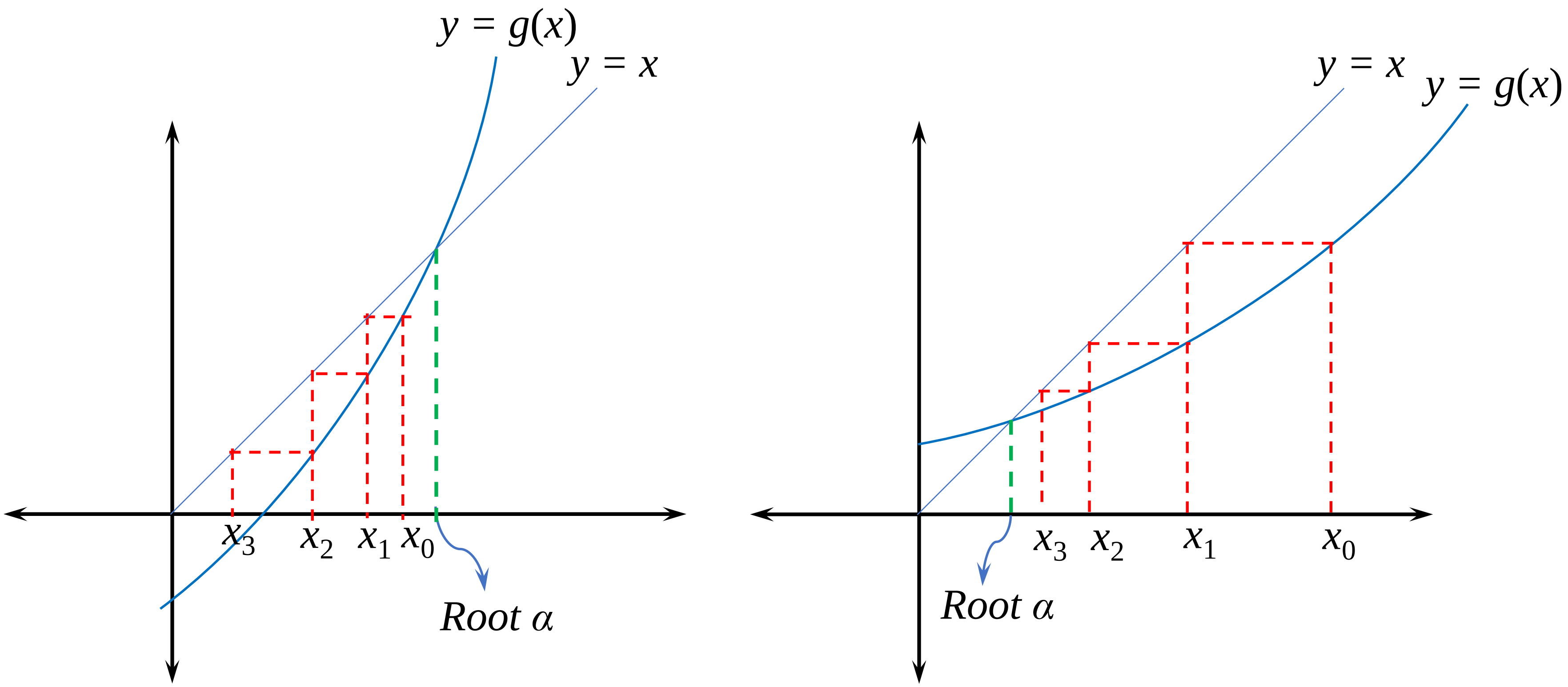
Open Methods: Fixed Point

- ✓ Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$
  - ✓ Re-arrange the function: $f(x) = 0$ to $x = g(x)$
  - ✓ Iteration: $x_{k+1} = g(x_k)$
  - ✓ Stopping criteria: $\left|\frac{x_{k+1}-x_k}{x_k}\right| \le \varepsilon$
- ✓ Convergence: after n iterations,
  - ✓ At the root: $\alpha = g(\alpha)$ or $\alpha - x_{n+1} = g(\alpha) - g(x_n)$
  - ✓ Mean Value Theorem: $\frac{g(\alpha)-g(x_n)}{\alpha - x_n} = g'(\xi)$ for some $\xi \in (\alpha, x_n)$

$(\alpha - x_{n+1}) = g'(\xi)(\alpha - x_n)$ or $e_{n+1} = g'(\xi)\, e_n$ or $\frac{|e_{n+1}|}{|e_n|} = |g'(\xi)|$

  - ✓ Condition for convergence: $\left| g'(\xi) \right| < 1$
  - ✓ As $x_n \to \alpha$, $\frac{|e_{n+1}|}{|e_n|} = |g'(\alpha)| = \text{constant}$

Open Methods: Fixed Point



Open Methods: Newton-Raphson

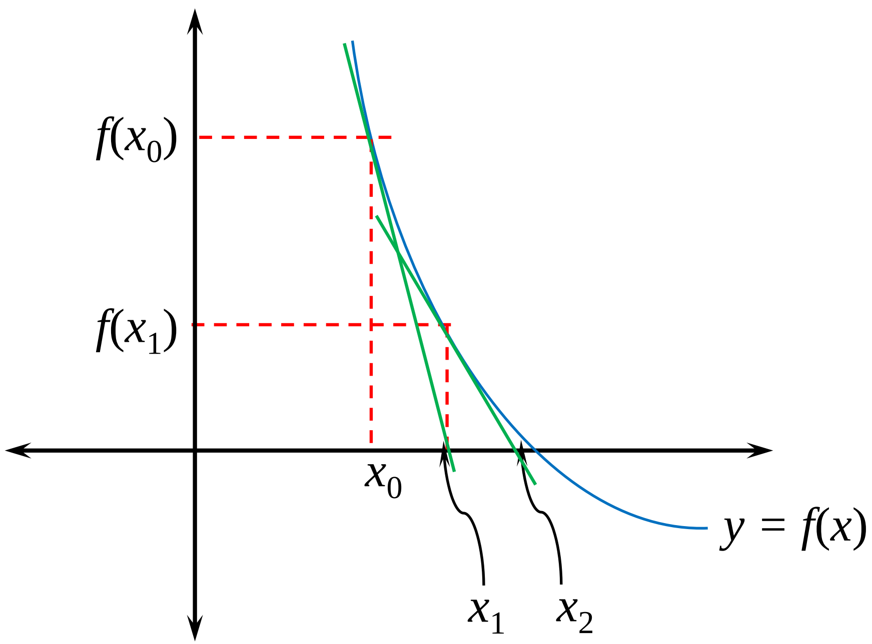
- Principle: Approximate the function as a straight line having same slope as the original function at the point of iteration.
- Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$
  - Iteration Step k: Taylor's Theorem

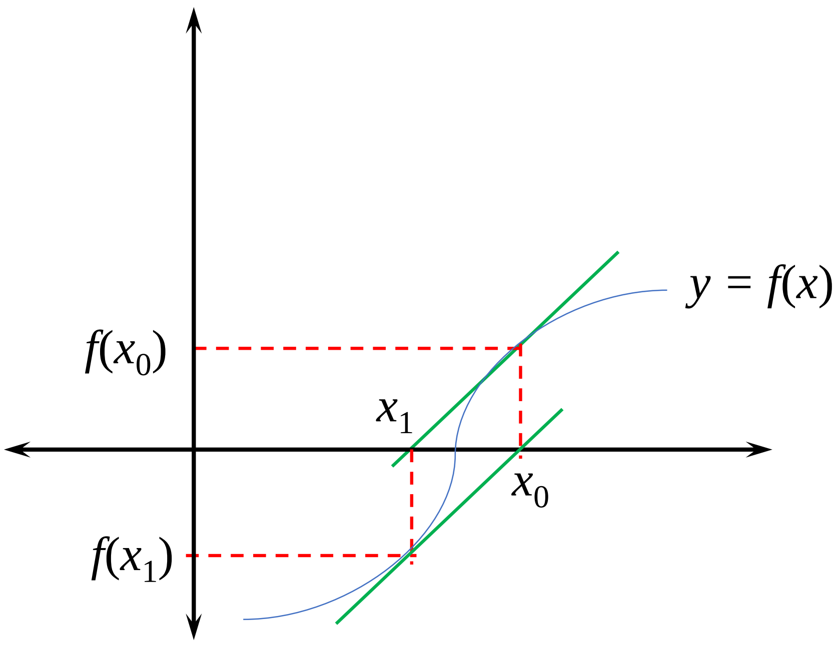
$$f(x_{k+1}) = f(x_k) + (x_{k+1} - x_k)\, f'(x_k) + \frac{1}{2!}(x_{k+1} - x_k)^2\, f''(x_k) + \cdots + \frac{1}{n!}(x_{k+1} - x_k)^n\, f^n(x_k) + \frac{1}{(n+1)!}(x_{k+1} - x_k)^{n+1}\, f^{n+1}(\xi)$$

for some $\xi \in (x_k, x_{k+1})$

  - Assumptions: Neglect 2nd and higher order terms and assume that the root is arrived at the $(k+1)^{\text{th}}$ iteration, i.e., $f(x_{k+1}) = 0$
  - Iteration Formula: $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$
  - Stopping criteria: $\left|\dfrac{x_{k+1} - x_k}{x_k}\right| \leq \varepsilon$

Open Methods: Newton-Raphson





Newton-Raphson method may get stuck!

Open Methods: Newton-Raphson

✓ Convergence: Taylor's Theorem after *n* iterations,

$$0 = f(\alpha) = f(x_n) + (\alpha - x_n)\, f'(x_n) + \tfrac{1}{2!}(\alpha - x_n)^2\, f''(\xi)$$

for some $\xi \in (x_n, \alpha)$

✓ Re-arrange: $\dfrac{f(x_n)}{f'(x_n)} + (\alpha - x_n) = -\tfrac{1}{2!}(\alpha - x_n)^2\, \dfrac{f''(\xi)}{f'(x_n)}$

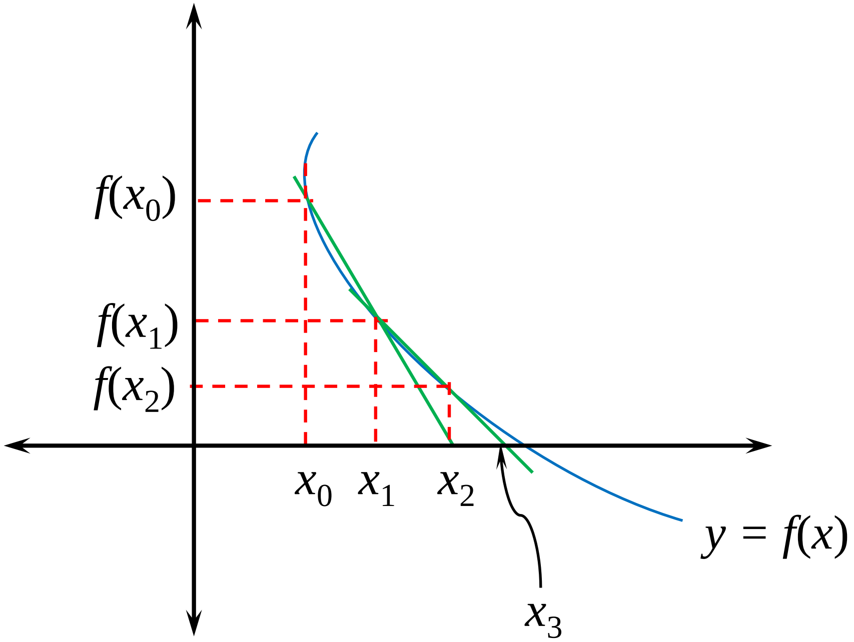
or $(\alpha - x_{n+1}) = -\tfrac{1}{2!}(\alpha - x_n)^2\, \dfrac{f''(\xi)}{f'(x_n)}$ or $e_{n+1} = -\tfrac{1}{2!}{e_n}^2\, \dfrac{f''(\xi)}{f'(x_n)}$

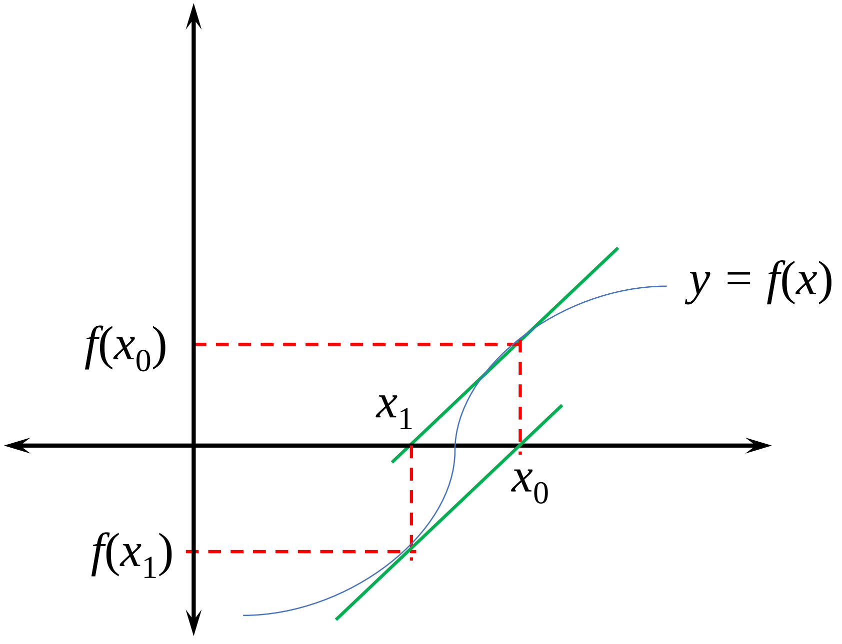
✓ As $x_n \to \alpha$, $\dfrac{|e_{n+1}|}{|{e_n}^2|} = \left| -\tfrac{1}{2}\, \dfrac{f''(\alpha)}{f'(\alpha)} \right| =$ constant

Open Methods: Secant

- ✓ Principle: Use a difference approximation for the slope or derivative in the Newton-Raphson method. This is equivalent to approximating the tangent with a secant.
- ✓ Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$
  - ✓ Initialize: choose two points x_0 and x_1 and evaluate $f(x_0)$ and $f(x_1)$
  - ✓ Approximation: $f'(x_k) \approx \frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$, replace in Newton-Raphson
  - ✓ Iteration Formula: $x_{k+1} = x_k - f(x_k)\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}$
  - ✓ Stopping criteria: $\left|\frac{x_{k+1}-x_k}{x_k}\right| \leq \varepsilon$

Open Methods: Secant





Open Methods: Secant

- ✓ Convergence: Newton's polynomial,

$$0 = f(\alpha) = f(x_n) + (\alpha - x_n) f[x_n, x_{n-1}] + \tfrac{1}{2}(\alpha - x_n)\,(\alpha - x_{n-1})\, f''(\xi)$$

for some $\xi \in (x_{n-1}, x_n, \alpha)$

Divided difference: $f[x_n, x_{n-1}] = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

- ✓ Re-arrange:

$$\frac{f(x_n)}{f[x_{n-1}, x_n]} + (\alpha - x_n) = (\alpha - x_{n+1}) = -\tfrac{1}{2}(\alpha - x_n)\,(\alpha - x_{n-1})\,\frac{f''(\xi)}{f[x_{n-1}, x_n]}$$

$$e_{n+1} = -\frac{1}{2}\, e_n\, e_{n-1}\, \frac{f''(\xi)}{f'(\xi')}, \quad \xi' \in (x_{n-1}, x_n) \;\; \text{[using mean value theorem]}$$

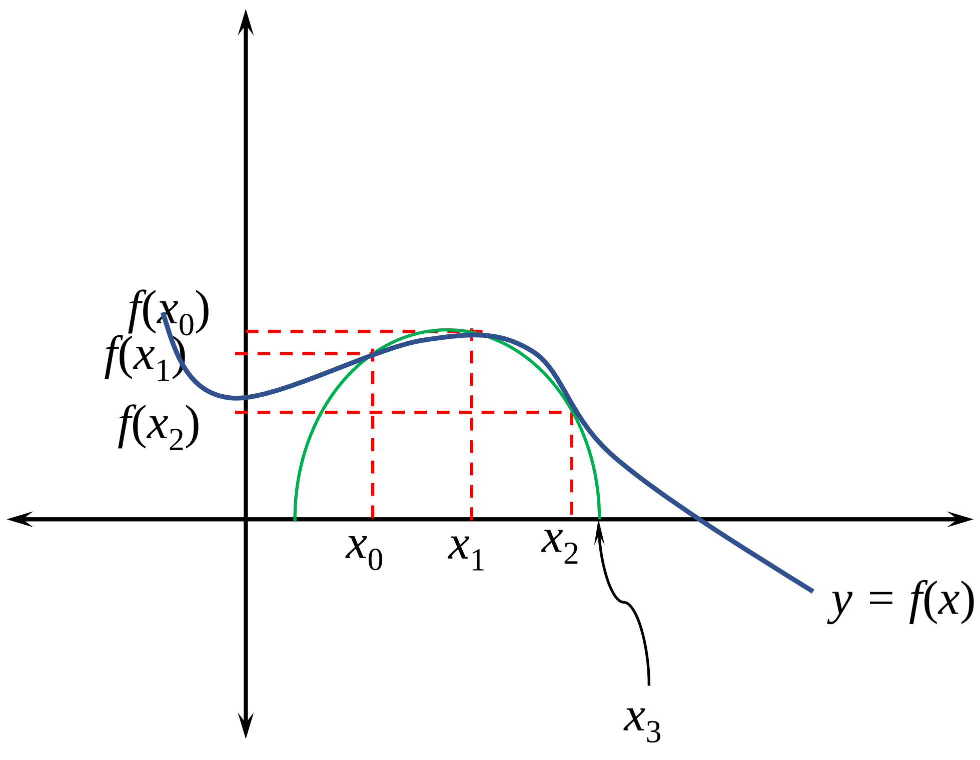
- ✓ As $x_n \to \alpha$, $\frac{|e_{n+1}|}{|e_n||e_{n-1}|} = \left| -\frac{1}{2}\,\frac{f''(\alpha)}{f'(\alpha)} \right|$ = constant

Open Methods: Secant

- ✓ Derivative of the Newton-Raphson method is evaluated numerically using difference approximation.
- ✓ Numerical methods for estimation of derivative of a function will be covered in detail later.
- ✓ Rest of the method is same.

Open Methods: Muller

- ✓ Principle: fit a quadratic polynomial through three points to approximate the function.
- ✓ Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$



Open Methods: Muller

- Initialize: choose three points x_0, x_1, x_2 and evaluate $f(x_0)$, $f(x_1)$, $f(x_2)$. Denote $f(x_k) = f_k$
  - Fit a quadratic polynomial through three points x_k, x_{k-1}, x_{k-2} as:

$$p(x) = a(x - x_k)^2 + b(x - x_k) + c$$

- Equations:

$$p(x_k) = f_k = c$$

$$p(x_{k-1}) = f_{k-1} = a(x_{k-1} - x_k)^2 + b(x_{k-1} - x_k) + f_k$$

$$p(x_{k-2}) = f_{k-2} = a(x_{k-2} - x_k)^2 + b(x_{k-2} - x_k) + f_k$$

- Define Divided Differences:
  - 1st Divided Difference: $f[x_k, x_{k-1}] = \frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$
  - 2nd Divided Difference: $f[x_k, x_{k-1}, x_{k-2}] = \frac{f[x_k,x_{k-1}]-f[x_{k-1},x_{k-2}]}{x_k-x_{k-2}}$

Open Methods: Muller

$p(x_k) = f_k = c$

$p(x_{k\text{-}1}) = f_{k\text{-}1} = a(x_{k\text{-}1} - x_k)^2 + b(x_{k\text{-}1} - x_k) + f_k$

$p(x_{k\text{-}2}) = f_{k\text{-}2} = a(x_{k\text{-}2} - x_k)^2 + b(x_{k\text{-}2} - x_k) + f_k$

- ✓ Express as:

$$\frac{f_k - f_{k-1}}{x_k - x_{k-1}} = f[x_k, x_{k-1}] = a(x_{k-1} - x_k) + b$$

$$\frac{f_k - f_{k-2}}{x_k - x_{k-2}} = f[x_k, x_{k-2}] = a(x_{k-2} - x_k) + b$$

- ✓ Solutions:

$$a = \frac{f[x_k, x_{k-1}] - f[x_k, x_{k-2}]}{x_{k-1} - x_{k-2}}$$

$$b = f[x_k, x_{k-2}] + a(x_k - x_{k-2})$$

Properties of Divided Differences

2nd Divided Difference: $f[x_k, x_{k-1}, x_{k-2}] = \frac{f[x_k, x_{k-1}] - f[x_{k-1}, x_{k-2}]}{x_k - x_{k-2}}$

$$a = \frac{f[x_k, x_{k-1}] - f[x_k, x_{k-2}]}{x_{k-1} - x_{k-2}}$$

$$= \frac{\frac{f_k - f_{k-1}}{x_k - x_{k-1}} - \frac{f_k - f_{k-2}}{x_k - x_{k-2}}}{x_{k-1} - x_{k-2}}$$

$$+x_{k-1}f_{k-1} - x_{k-1}f_{k-1}$$

$$= \frac{\cancel{x_k f_k} - x_k f_{k-1} - x_{k-2}f_k + x_{k-2}f_{k-1} - \cancel{x_k f_k} + x_k f_{k-2} + x_{k-1}f_k - x_{k-1}f_{k-2}}{(x_k - x_{k-1})(x_{k-1} - x_{k-2})(x_k - x_{k-2})}$$

$$= \frac{(x_{k-1} - x_{k-2})f_k - (x_{k-1} - x_{k-2})f_{k-1} - (x_k - x_{k-1})f_{k-1} + (x_k - x_{k-1})f_{k-2}}{(x_k - x_{k-1})(x_{k-1} - x_{k-2})(x_k - x_{k-2})}$$

$$= \frac{\frac{f_k - f_{k-1}}{x_k - x_{k-1}} - \frac{f_{k-1} - f_{k-2}}{x_{k-1} - x_{k-2}}}{x_k - x_{k-2}} = \frac{f[x_k, x_{k-1}] - f[x_{k-1}, x_{k-2}]}{x_k - x_{k-2}} = f[x_k, x_{k-1}, x_{k-2}]$$

Note: Properties of Divided Differences

✓ 1st Divided Difference:

$$f[x_k, x_{k-1}] = \frac{f_k - f_{k-1}}{x_k - x_{k-1}} = \frac{f_{k-1} - f_k}{x_{k-1} - x_k} = f[x_{k-1}, x_k]$$

✓ 2nd Divided Difference:

$$f[x_k, x_{k-1}, x_{k-2}] = \frac{f[x_k, x_{k-1}] - f[x_{k-1}, x_{k-2}]}{x_k - x_{k-2}} = \frac{f[x_k, x_{k-1}] - f[x_k, x_{k-2}]}{x_{k-1} - x_{k-2}} = \frac{f[x_k, x_{k-2}] - f[x_{k-1}, x_{k-2}]}{x_k - x_{k-1}}$$

$$f[x_k, x_{k-1}, x_{k-2}] = f[x_{k-1}, x_k, x_{k-2}] = f[x_{k-2}, x_{k-1}, x_k] = f[x_k, x_{k-2}, x_{k-1}]$$

We shall use these properties for the *Theory of Approximation*!

Open Methods: Muller

- ✓ Principle: fit a quadratic polynomial through three points to approximate the function.
- ✓ Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$
  - ✓ Initialize: choose three points x_0, x_1, x_2 and evaluate $f(x_0)$, $f(x_1)$, $f(x_2)$
  - ✓ Fit a quadratic polynomial through three points x_k, x_{k-1}, x_{k-2} as:

$$p(x) = a(x - x_k)^2 + b(x - x_k) + c$$

  - ✓ Constants: $c = f(x_k)$; $b = f[x_k, x_{k-1}] + (x_k - x_{k-1})f[x_k, x_{k-1}, x_{k-2}]$; $a = f[x_k, x_{k-1}, x_{k-2}]$
  - ✓ Iteration step k: $x_{k+1} = x_k - \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$ or $x_{k+1} = x_k - \frac{2c}{b \pm \sqrt{b^2 - 4ac}}$
  - ✓ Stopping criteria: $\left|\frac{x_{k+1} - x_k}{x_k}\right| \leq \varepsilon$

Open Methods: Muller

✓ Convergence: Analysis is similar to Secant method, using Newton's Polynomial.

$$e_{n+1} = -\frac{1}{6} e_n \, e_{n-1} \, e_{n-2} \frac{f'''(\xi)}{f'(\xi')},$$

$$\xi \in (x_{n-2}, x_{n-1}, x_n, \alpha), \qquad \xi' \in (x_{n-1}, x_n)$$

as $x_n \to \alpha$, $\dfrac{|e_{n+1}|}{|e_n||e_{n-1}||e_{n-2}|} = \left| -\dfrac{1}{6} \dfrac{f'''(\alpha)}{f'(\alpha)} \right| = \text{constant}$

Order of Convergence

- ✓ Definition: Let $\{x_0, x_1, x_2, x_3 \ \ldots\}$ be a sequence which converges to α. Define $e_n = x_n - \alpha$. If there exists a number p and a constant $C \neq 0$ such that,

$$\lim_{n \to \propto} \frac{|e_{n+1}|}{|e_n|^p} = C$$

Then, p is called the order of convergence of the sequence and C is the asymptotic error constant.

- ✓ Fixed Point: $\frac{|e_{n+1}|}{|e_n|} = |g'(\alpha)| = C$, 1st Order.
- ✓ Newton Raphson: $\frac{|e_{n+1}|}{|e_n|^2} = \left|-\frac{1}{2}\frac{f''(\alpha)}{f'(\alpha)}\right| = C$, 2nd Order
- ✓ Secant: $\frac{|e_{n+1}|}{|e_n||e_{n-1}|} = \left|-\frac{1}{2}\frac{f''(\alpha)}{f'(\alpha)}\right| = C$, mixed order, ≈ 1.6
- ✓ Muller: $\frac{|e_{n+1}|}{|e_n||e_{n-1}||e_{n-2}|} = \left|-\frac{1}{6}\frac{f'''(\alpha)}{f'(\alpha)}\right| = C$, mixed order, ≈ 1.84

Polynomial Methods: Single Root

$$p_n(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

If we divide by a factor (x - r) such that, $r = \alpha$ is a root of the polynomial, we will get an exact polynomial of order (n - 1).

$$q_{n-1}(x) = \sum_{k=0}^{n-1} b_{k+1} x^k = b_1 + b_2 x + b_3 x^2 + \cdots + b_n x^{n-1}$$

If $r \neq \alpha$, dividing by a factor (x - r) will have a remainder b_0.

Polynomial Methods: Single Root

$$p_n(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

$$= (x - r) q_{n-1}(x) + b_0 = (x - r) \sum_{k=0}^{n-1} b_{k+1} x^k + b_0$$

$$= b_0 + b_1(x - r) + b_2 x(x - r) + b_3 x^2(x - r) + \cdots + b_{n-2} x^{n-3}(x - r)$$
$$+ b_{n-1} x^{n-2}(x - r) + b_n x^{n-1}(x - r)$$

$$= (b_0 - r b_1) + x(b_1 - r b_2) + x^2(b_2 - r b_3) + \cdots + x^{n-2}(b_{n-2} - r b_{n-1})$$
$$+ x^{n-1}(b_{n-1} - r b_n) + b_n x^n$$

$$b_n = a_n; \;\; b_i = a_i + r b_{i+1}; \quad i = \; (n-1), (n-2), \cdots 2, 1, 0$$

Polynomial Methods: Single Root

b_0 is a function of $r \longrightarrow b_0(r)$, at $r = \alpha$, $b_0(r) = 0$

Problem: $f(x) = 0$, find a root $x = \alpha$ such that $f(\alpha) = 0$

Problem: $b_0(r) = 0$, find a root $r = \alpha$ such that $b_0(\alpha) = 0$

Apply Newton-Raphson:

Iteration Formula for Step k:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{or} \quad r_{k+1} = r_k - \frac{b_0(r_k)}{b_0'(r_k)}$$

$$b_0 = a_0 + rb_1 \longrightarrow b_0'(r) = b_1 \longrightarrow r_{k+1} = r_k - \frac{b_0(r_k)}{b_1(r_k)}$$

Assume a value of r, estimate b_0 and b_1, compute new r.

Continue until b_0 becomes zero. (with acceptable relative error)

Polynomial Methods: Bairstow

$$p_n(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Let us divide by a factor ($x^2 - rx - s$). If the factor is exact, the resulting polynomial will be of order ($n - 2$). Two roots of the polynomial can be estimated simultaneously as the roots of the quadratic factor. For the complex roots, they will be the complex conjugates.

$$q_{n-2}(x) = \sum_{k=0}^{n-2} b_{k+2} x^k = b_2 + b_3 x + b_4 x^2 + \cdots + b_n x^{n-2}$$

If the factor ($x^2 - rx - s$) is not exact, there will be two remainder terms, one function of x and another constant.

Let us express the remainder term as $b_1(x - r) + b_0$. This form instead of the standard $b_1 x + b_0$ is chosen to device a convenient iteration formula!

Polynomial Methods: Bairstow

$$p_n(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$
$$= (x^2 - rx - s) q_{n-2}(x) + b_1(x - r) + b_0$$

$$= (x^2 - rx - s) \sum_{k=0}^{n-2} b_{k+2} x^k + b_1(x - r) + b_0$$
$$= b_0 + b_1(x - r) + b_2(x^2 - rx - s) + b_3 x(x^2 - rx - s) + \cdots$$
$$+ b_{n-2} x^{n-4}(x^2 - rx - s) + b_{n-1} x^{n-3}(x^2 - rx - s) + b_n x^{n-2}(x^2 - rx - s)$$

$$= (b_0 - rb_1 - sb_2) + x(b_1 - rb_2 - sb_3) + x^2(b_2 - rb_3 - sb_4) + \cdots$$
$$+ x^{n-2}(b_{n-2} - rb_{n-1} - sb_n) + x^{n-1}(b_{n-1} - rb_n) + b_n x^n$$

$$b_n = a_n;\ b_{n-1} = a_{n-1} + rb_n\ ;\ b_i = a_i + rb_{i+1} + sb_{i+2}; \quad i = (n-2), \cdots 2, 1, 0$$

Polynomial Methods: Bairstow

b_0 and b_1 are functions of r and s $\rightarrow$ $b_0(r, s)$ and $b_1(r, s)$

Expand in Taylor's series: Apply 2-d Newton-Raphson

$$0 = b_0(r + \Delta r, s + \Delta s) = b_0 + \frac{\partial b_0}{\partial r}\Delta r + \frac{\partial b_0}{\partial s}\Delta s + HOT$$

$$0 = b_1(r + \Delta r, s + \Delta s) = b_1 + \frac{\partial b_1}{\partial r}\Delta r + \frac{\partial b_1}{\partial s}\Delta s + HOT$$

$$\begin{bmatrix} \frac{\partial b_0}{\partial r} & \frac{\partial b_0}{\partial s} \\ \frac{\partial b_1}{\partial r} & \frac{\partial b_1}{\partial s} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta s \end{bmatrix} = \begin{bmatrix} -b_0 \\ -b_1 \end{bmatrix}$$

Need to evaluate: $\frac{\partial b_0}{\partial r}$, $\frac{\partial b_0}{\partial s}$, $\frac{\partial b_1}{\partial r}$ and $\frac{\partial b_1}{\partial s}$

Polynomial Methods: Bairstow

Partial differentials with respect to *r*:

$$b_n = a_n \ \rightarrow \frac{\partial b_n}{\partial r} = 0; \qquad b_{n-1} = a_{n-1} + rb_n \rightarrow \frac{\partial b_{n-1}}{\partial r} = b_n = c_n$$

$$b_{n-2} = a_{n-2} + rb_{n-1} + sb_n \rightarrow \frac{\partial b_{n-2}}{\partial r} = b_{n-1} + r\frac{\partial b_{n-1}}{\partial r} + s\frac{\partial b_n}{\partial r}$$

$$= b_{n-1} + rc_n = c_{n-1}$$

$$b_{n-3} = a_{n-3} + rb_{n-2} + sb_{n-1} \ \rightarrow \ \frac{\partial b_{n-3}}{\partial r} = b_{n-2} + r\frac{\partial b_{n-2}}{\partial r} + s\frac{\partial b_{n-1}}{\partial r}$$

$$= b_{n-2} + rc_{n-1} + sc_n = c_{n-2}$$

$$c_n = b_n; \ c_{n-1} = b_{n-1} + rc_n \ ; \ c_i = b_i + rc_{i+1} + sc_{i+2}; \quad i = (n-2), \cdots 2, 1, 0$$

$$\frac{\partial b_i}{\partial r} = c_{i+1}; \qquad i = (n-1), \cdots 2{,}1{,}0$$

Polynomial Methods: Bairstow

Partial differentials with respect to *s*:

$$b_n = a_n \;\rightarrow\; \frac{\partial b_n}{\partial s} = 0;\; b_{n-1} = a_{n-1} + rb_n \;\rightarrow\; \frac{\partial b_{n-1}}{\partial s} = 0$$

$$b_{n-2} = a_{n-2} + rb_{n-1} + sb_n \rightarrow \frac{\partial b_{n-2}}{\partial s} = b_n + r\frac{\partial b_{n-1}}{\partial s} + s\frac{\partial b_n}{\partial s} = b_n = c_n$$

$$b_{n-3} = a_{n-3} + rb_{n-2} + sb_{n-1} \;\rightarrow\; \frac{\partial b_{n-3}}{\partial s} = b_{n-1} + r\frac{\partial b_{n-2}}{\partial s} + s\frac{\partial b_{n-1}}{\partial s}$$

$$= b_{n-1} + rc_n = c_{n-1}$$

$$b_{n-4} = a_{n-4} + rb_{n-3} + sb_{n-2} \;\rightarrow\; \frac{\partial b_{n-4}}{\partial s} = b_{n-2} + r\frac{\partial b_{n-3}}{\partial s} + s\frac{\partial b_{n-2}}{\partial s}$$

$$= b_{n-2} + rc_{n-1} + sb_n = c_{n-2}$$

$$c_n = b_n;\; c_{n-1} = b_{n-1} + rc_n\,;\; c_i = b_i + rc_{i+1} + sc_{i+2};\quad i = (n-2), \cdots 2, 1, 0$$

$$\frac{\partial b_i}{\partial s} = c_{i+2};\quad i = (n-2), \cdots 2,1,0$$

Polynomial Methods: Bairstow

$$\frac{\partial b_i}{\partial r} = c_{i+1};\ i = (n-1), \cdots 2,1,0 \ \text{ and } \ \frac{\partial b_i}{\partial s} = c_{i+2};\ i = (n-2), \cdots 2,1,0$$

$$\frac{\partial b_0}{\partial r} = c_1; \quad \frac{\partial b_1}{\partial r} = c_2; \quad \frac{\partial b_0}{\partial s} = c_2 \ \text{ and } \ \frac{\partial b_1}{\partial s} = c_3$$

$$\begin{bmatrix} \frac{\partial b_0}{\partial r} & \frac{\partial b_0}{\partial s} \\ \frac{\partial b_1}{\partial r} & \frac{\partial b_1}{\partial s} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta s \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta s \end{bmatrix} = \begin{bmatrix} -b_0 \\ -b_1 \end{bmatrix}$$

For any given polynomial, we know $\{a_0, a_1, \ldots a_n\}$. Assume r and s.
Compute $\{b_0, b_1, \ldots b_n\}$ and $\{c_0, c_1, \ldots c_n\}$. Compute Δr and Δs.

Polynomial Methods: Bairstow Algorithm

- ✓ Step 1: input $a_0, a_1, \ldots a_n$ and initialize r and s.
- ✓ Step 2: compute $b_0, b_1, \ldots b_n$

$$b_n = a_n; b_{n-1} = a_{n-1} + rb_n \ ; \ b_i = a_i + rb_{i+1} + sb_{i+2}; \ i = (n-2), \cdots 2, 1, 0$$

- ✓ Step 3: compute $c_0, c_1, \ldots c_n$

$$c_n = b_n; \ c_{n-1} = b_{n-1} + rc_n \ ; \ c_i = b_i + rc_{i+1} + sc_{i+2}; \ i = (n-2), \cdots 2, 1, 0$$

- ✓ Step 4: compute Δr and Δs from $\begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta s \end{bmatrix} = \begin{bmatrix} -b_0 \\ -b_1 \end{bmatrix}$
- ✓ Step 5: compute $r_{new} = r + \Delta r$, $s_{new} = s + \Delta s$
- ✓ Step 6: check for convergence, $\left|\frac{r_{new}-r}{r}\right|, \left|\frac{s_{new}-s}{s}\right| \leq \varepsilon$ and $b_0, b_1 \leq \varepsilon'$
- ✓ Step 7: Stop if all convergence checks are satisfied. Else, set $r = r_{new}$, $s = s_{new}$ and go to step 2.

Multiple Roots

- ✓ Definition: A root α of the equation $f(x) = 0$ is said to have a multiplicity of q if,

$$g(x) = \frac{f(x)}{(x-\alpha)^q} \qquad 0 \neq g(\alpha) < \propto$$

when, $q > 1$, the order of convergence are no longer valid.

- ✓ Solution: Suppose a function $f(x)$ is q-times continuously differentiable in the neighbourhood of a root α of multiplicity q,

$$f(x) = \frac{1}{q!}(x-\alpha)^q f^q(\xi) \text{ and } f'(x) = \frac{1}{(q-1)!}(x-\alpha)^{q-1} f^q(\xi')$$

$$\text{where } \xi, \xi' \in (x, \alpha)$$

$$\text{Define } u(x) = \frac{f(x)}{f'(x)} = \frac{1}{q}(x-\alpha)\frac{f^q(\xi)}{f^q(\xi')}$$

$$\lim_{x \to \alpha} \frac{u(x)}{(x-\alpha)} = \frac{1}{q}$$

Therefore, α is a root of $f(x)$ of multiplicity q but is a simple root of $u(x)$!