Lecture 03 | Roots of a single-variable equations.

4) Secant method: convergence
5) Bisection method
6) Regula Falsi method.

$$x_{n+1} = x_n - f(x_n)\frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Tailor exp. near $x = \xi$, where $f(\xi) = 0$ (no fixed point)

$$x_{n+1} = x_n - (x_n - x_{n-1})\left\{\frac{\overset{=0}{f(\xi)} + f'(\xi)(x_n-\xi) + \frac{1}{2}f''(\xi)(x_n-\xi)^2}{\overset{=0}{f(\xi)} + f'(x_n-\xi) + \frac{1}{2}f''(\xi)(x_{n-1}-\xi)^2 - \overset{=0}{f(\xi)} - f'(x_{n-1}-\xi) - f''(x_{n-1}-\xi)^2}\right\}$$

$$x_{n+1} - x_n = -(x_n - x_{n-1})\left\{\frac{(x_n-\xi)\frac{1}{f'(\xi)}\left(1+\frac{f''(\xi)}{2f'(\xi)}(x_n-\xi)\right)}{f'(x_n - \cancel{\xi} - x_{n-1} + \cancel{\xi}) + \frac{1}{2}f''(\xi)\left[(x_n-\xi)^2 - (x_{n-1}-\xi)^2\right]}\right\}$$

$$= -\cancel{(x_n - x_{n-1})}\left\{\frac{(x_n-\xi)\cancel{\frac{1}{f'(\xi)}}\left(1+\frac{f''(\xi)}{2f'(\xi)}(x_n-\xi)\right)}{\cancel{\frac{1}{f'(\xi)}}\left(\cancel{(x_n-x_{n-1})} + \frac{1}{2}\frac{f''(\xi)}{f'(\xi)}\frac{\left((x_n^2-x_{n-1}^2) - 2\xi(x_n-x_{n-1})\right)}{(x_n-x_{n-1})}\right)}\right\}$$

$$= -\left\{\frac{(x_n-\xi)\left(1+\frac{f''(\xi)}{2f'(\xi)}(x_n-\xi)\right)}{\left(1+\frac{1}{2}\frac{f''(\xi)}{f'(\xi)}\left((x_n+x_{n-1})-2\xi\right)\right)}\right\} =$$

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

$$= \left\{\begin{array}{l}\frac{1+x}{1+x+y} \approx 1 - \frac{1}{1+x}y \approx \\ \approx 1-y \\ x \ll 1\end{array}\right\} \approx -(x_n-\xi)\left[1-\frac{f''(\xi)}{2f'(\xi)}(x_{n-1}-\xi)\right] \Rightarrow$$

$$\Rightarrow x_{n+1} - \cancel{x_n} - \cancel{x_n} + \xi + \frac{f''(\xi)}{2f'(\xi)}(x_n-\xi)(x_{n-1}-\xi)$$

$$\boxed{(x_{n+1}-\xi) = \frac{f''(\xi)}{2f'(\xi)}(x_n-\xi)(x_{n-1}-\xi)}$$

$$(x_{n+1}-\xi)=\alpha(x_n-\xi)(x_{n-1}-\xi)$$ put in

We are looking for relation in the form: $\boxed{(x_{n+1}-\xi)=\alpha^{a}(x_n-\xi)^{p}}$

$$(x_n-\xi)=\alpha^{a}(x_{n-1}-\xi)^{p}$$

$$(x_n-\xi)^{\frac{1}{p}}=\alpha^{\frac{a}{p}}(x_{n-1}-\xi)$$

$$(x_{n-1}-\xi)=(x_n-\xi)^{\frac{1}{p}}\alpha^{-\frac{a}{p}} \Rightarrow$$

$$(x_{n+1}-\xi)=\alpha^{a}(x_n-\xi)^{p}=\alpha(x_n-\xi)(x_n-\xi)^{\frac{1}{p}}\alpha^{-\frac{a}{p}}$$

$$\alpha^{a}(x_n-\xi)^{p}=\alpha^{1-\frac{a}{p}}(x_n-\xi)^{1+\frac{1}{p}} \rightarrow$$

$$\begin{cases} a=1-\frac{a}{p} & ap=p-a \Rightarrow a(p+1)=p \quad \boxed{a=\frac{p}{p+1}} \\ p=1+\frac{1}{p} & \Rightarrow p^2-p-1=0 \end{cases}$$

$$\boxed{p>0} \Rightarrow p=\frac{1\pm\sqrt{1+4}}{2}=\frac{1\pm\sqrt{5}}{2} \Rightarrow \boxed{p=\frac{\sqrt{5}+1}{2}}$$

$$\boxed{p\approx 1.62}$$

$$\boxed{(x_{n+1}-\xi)=\left[\frac{f''(\xi)}{2f'(\xi)}\right]^{a}(x_n-\xi)^{(\sqrt{5}+1)/2}}$$

Convergence is superlinear

!! Secant method converges slower than Newton's method.

Two-points methods: Bracketing – minimization of the interval I', where zero ξ lies.

Bisection method:

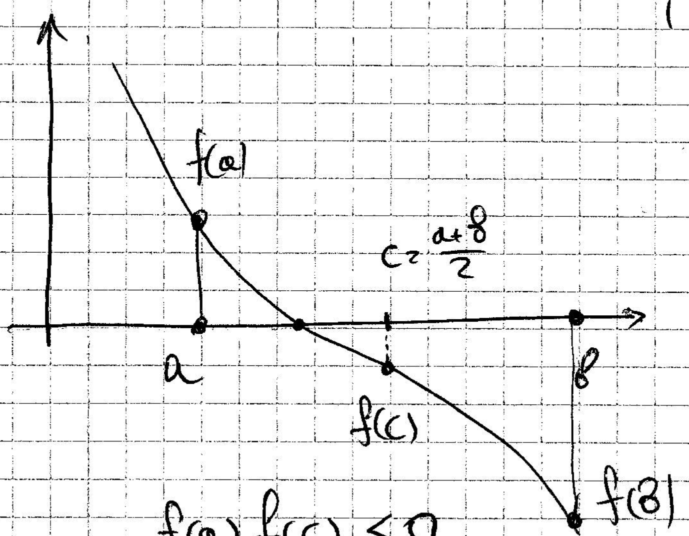
if $f(x)\in[a,b]=I$, and (*) $\boxed{f(a)\cdot f(b)<0} \Rightarrow$ by mean value theorem $\exists\ \xi\in I$, for which $f(\xi)=0$

(*) → means $f(x)$ has different sign at I boundary

Algorithm:

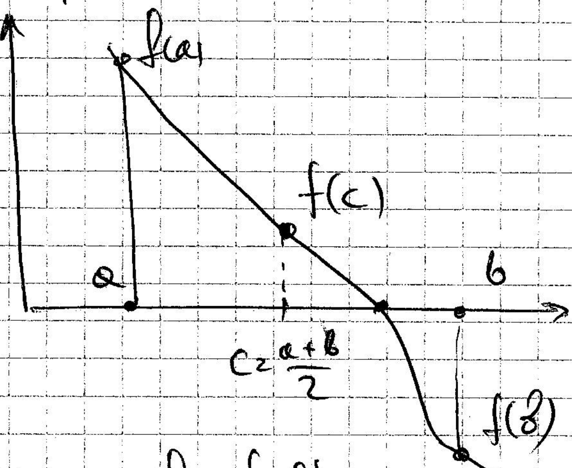
1) $f(x)$
2) $f(a)\cdot f(b) < 0$
3) $c = \frac{1}{2}(a+b) = a + \left(\frac{b-a}{2}\right)$ → better to add ~~small~~ correction to ~~the~~ a previous approximation

4) Check if $f(a)f(c) < 0$ repeat (3) with $b=c$ | Check if $f(c)f(b) < 0$ repeat (3) with $a=c$



$f(a)f(c) < 0$

$\Rightarrow b = c$



$f(c)f(b) < 0$

$\Rightarrow a = c$

5) if $|f(c)| \leq \delta$ stop.

Convergence

We have sequence of bisections:

$[a_0, b_0], [a_1, b_1] \ldots [a_n, b_n]$ || every next interval is a half of the previous one.

$\forall\, n > 0$; $b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n) \Rightarrow$

$(b_1 - a_1) = \frac{1}{2}(b_0 - a_0)$; $(b_2 - a_2) = \frac{1}{2}(b_1 - a_1) = \frac{1}{2}\cdot\frac{1}{2}(b_0 - a_0) \Rightarrow$

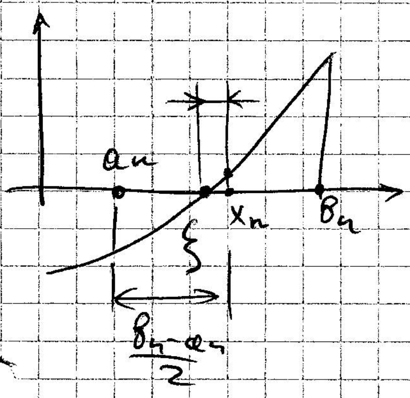
$$\boxed{(b_n - a_n) = 2^{-n}(b_0 - a_0)} \quad (1)$$

for convergence $|x_{n+1}-\xi| \le C\,|x_n-\xi|^p$

p — order of convergence

$$x_n = \frac{a_n+b_n}{2};$$

$$|x_n-\xi| \le \frac{b_n-a_n}{2},$$ the absolute value of the n-th approximation is smaller or equal to the half of the last interval, as $\xi \in [a_n, x_n]$.



$$|x_n-\xi| \le \frac{1}{2}2^{-n}(b_0-a_0) = 2^{-(n+1)}(b_0-a_0)$$

Then $|(x_{n+1}-\xi)| \le 2^{-(n+2)}(b_0-a_0)$

$$\frac{|x_{n+1}-\xi|}{|x_n-\xi|} \le \frac{2^{-(n+2)}}{2^{-(n+1)}}\frac{\cancel{b_0-a_0}}{\cancel{b_0-a_0}} = \frac{1}{2} \Rightarrow$$

$$\boxed{|x_{n+1}-\xi| \le \frac{1}{2}|x_n-\xi|}$$

$C = \frac{1}{2} < 1$

$p = 1$ – linear convergence.

Number of steps estimate.

δ is relative error we are looking for.

$$|x_{n+1}-\xi| \le 2^{-(n+1)}(b_0-a_0) < \delta$$

$$2^{-(n+1)} < \delta/(b_0-a_0) \Leftrightarrow \left(\frac{(b_0-a_0)}{\delta}\right) < 2^{(n+1)}$$

$$\log_2\left(\frac{b_0-a_0}{\delta}\right) < (n+1)$$

$$n > \left[\frac{\log\left(\frac{b_0-a_0}{\delta}\right)}{\log 2} - 1\right] \qquad n > \frac{\log\left(\frac{b_0-a_0}{\delta}\right)}{\log 2}$$

Experimental determination of convergence order p

$$|x_{n+1}-\xi| \le C\,|x_n-\xi|^p$$

$$\log_{C|x_n-\xi|}(x_{n+1}-\xi) \le p$$

$$\frac{\log|x_{n+1}-\xi|}{\log(C|x_n-\xi|)} \le p_n$$

$$\boxed{p_n \approx \frac{\log|x_{n+1}-\xi|}{\log|x_n-\xi|}} \qquad p=\lim_{n\to\infty} p_n$$

Algorithm:

1) to find ξ with a given ~~error~~ accuracy δ.
2) the calculate sequence $\{p_n\}$ till p_n converge to some p'
3) Compare with analytical p.

Newton's method: $\boxed{p=2}$
Secant method: $\boxed{p=\frac{\sqrt{5}+1}{2}\approx 1.618}$
Bisection —"— : $\boxed{p=1}$
Regula Falsi —"— : superlinear $\boxed{2>p>1}$

Multiple roots

~~Roots of polynomials~~
~~1) Horner method~~
~~2) Muller method~~

DF: The root ξ of a function $f(x)\in C[a,b]$ is called a j-fold (multiple of jth order) root for $j\in\mathbb{N}$, if $\boxed{f(x)=(x-\xi)^j g(x)}$ with a continuous $g(x)$ for which $g(\xi)\neq 0$.

Theorem: if $\xi \in I$ is j-fold root of f, and f is sufficiently often continuosly differentiable on I, then ξ is a simple root of the function $g(x)$ defined by

$$\boxed{u(x) = \frac{f(x)}{f'(x)}}$$

○

$$f(x) = (x-\xi)^j g(x) \Rightarrow$$

$$u(x) = \frac{f(x)}{f'(x)} = \frac{g(x)(x-\xi)^j}{j(x-\xi)^{j-1} g(x) + (x-\xi)^j g'(x)} =$$

$$= \frac{(x-\xi)^j g(x)}{(x-\xi)^{j-1}(j g(x) + (x-\xi) g'(x))} = \frac{(x-\xi) g(x)}{(j g(x) + (x-\xi) g'(x))}$$

$$g(\xi) \neq 0; \quad u(\xi) = \frac{(\xi-\xi) g(\xi)}{(j \cdot g(\xi) + (\xi-\xi) g'(\xi))} =$$

$$= \frac{0}{j \cdot g(\xi) + 0} = 0 \Rightarrow \xi \text{ simple root.}$$

$$\boxed{u(x) = (x-\xi)\tilde{u}(x)}$$

□

Then in iteration methods one chose $u(x) = \frac{f(x)}{f'(x)}$ as a fixed point function.

Ex. Newton's method.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$u(x) = \frac{f(x)}{f'(x)} \Rightarrow \qquad x_{n+1} = x_n - \frac{u(x_n)}{u'(x_n)} =$$

$$= x_n - \frac{\cancel{f}(x_n)}{\cancel{f'}(x_n)} \frac{(\cancel{f'}(x_n))^2}{(\cancel{f'}(x_n))^2 - \cancel{f}(x_n)\cancel{f''}(x_n)} = x_n - \frac{f(x_n) f'(x_n)}{(f'(x_n))^2 - f(x_n) f''(x_n)}$$

- iteration rule for multiple roots.
- Convergence rate $p = 2$

Roots of Polynomials.

1) One is looking for all roots of polynomials of order n. 2) To use iterative methods starting points should be find for ~~every~~ each root close to it to ensure convergence

1) Müller method gives an approximate solution for all roots of polynomials.

2) One need an effective and fast method to evaluate polynomials and it derivatives at a given points (Horner method.)

Algorithm:

1) for a given polynomials $P_n(x)$
2) Muller method gives starting points for each root.
3) Roots are searched iteractively using Newton method
4) Polynomials and its derivative are calculated using Horner method.

---

1) Horner method. - is used to calculate values of a polynomial and its derivatives at a given x_0.

Theorem: $P_n(x) = a_n^0 x^n + a_{n-1}^0 x^{n-1} \dots + a_1^0 x + a_0^0$; $a_i^0 \in \mathbb{R}$

Then for $x = x_0$ =>

$$P_n(x_0) = \left(a_n^0 x_0^n + a_{n-1}^0 x_0^{n-1} \cdots a_1^0 x_0 + a_0^0\right) =$$

$$= \left(a_n^0 x_0^{n-1} + a_{n-1}^0 x^{n-2} \cdots a_1^0\right) x_0 + a_0^0 =$$

$$= \left(\left(a_n^0 x_0^{n-2} + a_{n-1}^0 x^{n-3} + \dots a_2^0\right) x_0 + a_1^0\right) x_0 + a_0^0 =$$

$$= \left(\dots\left(\left(a_n^0 x_0 + a_{n-1}^0\right) x_0 + a_{n-2}^0\right) x_0 + \dots a_1^0\right) x_0 + a_0^0$$

Let $\begin{cases} a_k^1 = a_{k+1}^1 x_0 + a_k^0, & k = n-1, n-2 \dots, 0 \\ a_n^1 = a_n^0 \end{cases}$

Then
(1) $\boxed{P_n(x_0) = a_0^1}$

$\square$ $a_k^0 = a_k^1 - a_{k+1}^1 x_0$

$P_n(x_0) = (\dots((a_n^0 x_0 + a_{n-1}^0)x_0 + a_{n-2}^0)x_0 + \dots a_1^0)x_0 + a_0^0$

$P_n(x_0) = (\dots(\cancel{a_n^1 x_0} + a_{n-1}^1 - \cancel{a_n^1 x_0})x_0 + a_{n-2}^0)x_0 \dots a_0^0$

$= (\dots(a_{n-1}^1 x_0 + a_{n-2}^0)x_0 + \dots a_0^0$

$= (\dots(\cancel{a_{n-1}^1 x_0} + a_{n-2}^1 - \cancel{a_{n-1}^1 x_0})x_0 + a_0^0$

$= a_1^1 x_0 + a_0^0 \equiv a_0^1 \Rightarrow \boxed{P_n(x_0) = a_0^1}$

$\square$

(2) if $Q(x) = a_n^1 x^{n-1} + a_{n-1}^1 x^{n-2} + \dots + a_1^1 \Rightarrow$

$P(x) = (x - x_0)Q(x) + a_0^1 = (x - x_0)Q(x) + P_n(x_0)$

$\square$

$Q(x)(x - x_0) = a_n^1 (x - x_0)x^{n-1} + a_{n-1}^1 (x - x_0)x^{n-2} + \dots a_1^1 =$

$= a_n^0 x^n - a_n^0 x_0 x^{n-1} + a_{n-1}^1 x^{n-1} - a_{n-1}^1 x_0 x^{n-2} + \dots a_1^1 =$

$= a_n^0 x^n - \cancel{a_n^0 x_0 x^{n-1}} + \cancel{a_n^0 x_0 x^{n-1}} + a_{n-1}^0 x^{n-1} - a_n^0 x_0^2 x^{n-2} - a_{n-1}^0 x_0 x^{n-2} +$

$\cancel{a_{n-1}^1 = a_n^1 x_0 + a_{n-1}^1}$

$+ a_1^1 = \qquad a_{n-1}^1 = a_n^1 x_0 + a_{n-1}^0 = a_n^0 x_0 + a_{n-1}^0$

$= \{ a_1^1 = a_2^1 x_0 + a_1^0 \} = (a_n^0 x^n + a_{n-1}^0 x^{n-1} + \dots a_0^0) -$

$- (a_n^0 x_0^n + \dots a_0^0) = P_n(X) - P_n(X_0) \Rightarrow$

$\boxed{P_n(x) = (x - x_0)Q(x) + P_n(x_0)}$

$\square$

$$\boxed{P_n(x) = (x-x_0)P_{n-1}(x) + P_n(x_0)}$$; here $P_{n-1}(x) \equiv Q(x)$

deflation

$$P_n'(x) = P_{n-1}'(x)(x-x_0) + P_{n-1}(x) \Rightarrow$$

1) $\boxed{P_n'(x_0) = P_{n-1}(x_0)}$ 2) if x_0 is zero of $P_n(x) \Rightarrow$

2) $\boxed{P_n(x_0) = (x-x_0)P_{n-1}(x)}$ if x_0 root of $P_n(x)$

<u>Algorithm:</u>

1) $P_n(x)$; n, $\{a_i^0\}_{i=0}^n$; x_0

2) evaluate $\tilde{X}_0 = P(x_0)$; $\tilde{Y}_0 = P'(x_0)$

1) $\tilde{X}_0 = a_n^0$; compute a_n^1 for P_n $\Leftrightarrow$ a_n^1

$\tilde{Y}_0 = a_n^0$; compute a_{n-1}^1 for P_n'

2) loop $j = n-1, 1$

$$\tilde{X}_0 = x_0\tilde{X}_0 + a_j^0 \quad \Leftrightarrow \quad \begin{cases} a_j^1 = a_{j+1}^1 x_0 + a_j^0 \\ a_j^2 = a_{j+1}^2 x_0 + a_j^1 = a_{j+1}^2 x_0 + a_{j+1}^1 x + a_j^0 \end{cases}$$

$$\tilde{Y}_0 = x_0\tilde{Y}_0 + \tilde{x}_0 \quad \Leftrightarrow$$

3) $\tilde{Y}_0 = x_0\tilde{Y}_0 + a_n^0$

4) $\Rightarrow \{\tilde{X}_0, \tilde{Y}_0\}$

$\tilde{X}_0 = a_0^1 = P_n(x_0)$

<u>Horner's scheme</u> for $\begin{cases} P_n(x_0) \\ P_n'(x_0) \end{cases}$

~~operations~~

Consequence: $P_{n-1}(x) = (x-x_0)P_{n-2}(x) + P_{n-1}(x_0)$; so if x_0 root of P_{n-1} $\Rightarrow$

$$P_{n-1}(x) = (x-x_0)P_{n-2}(x) \Longrightarrow$$

1) $P_n(x) = (x-x_1)(x-x_2)P_{n-2}(x)$, where x_1, x_2 roots of $P_n(x)$;

x_1 root of $P_{n-1}(x)$
x_2 root of $P_{n-2}(x)$; $\Rightarrow P_n(x) = (x-x_1)(x-x_2)\cdots(x-x_n)$

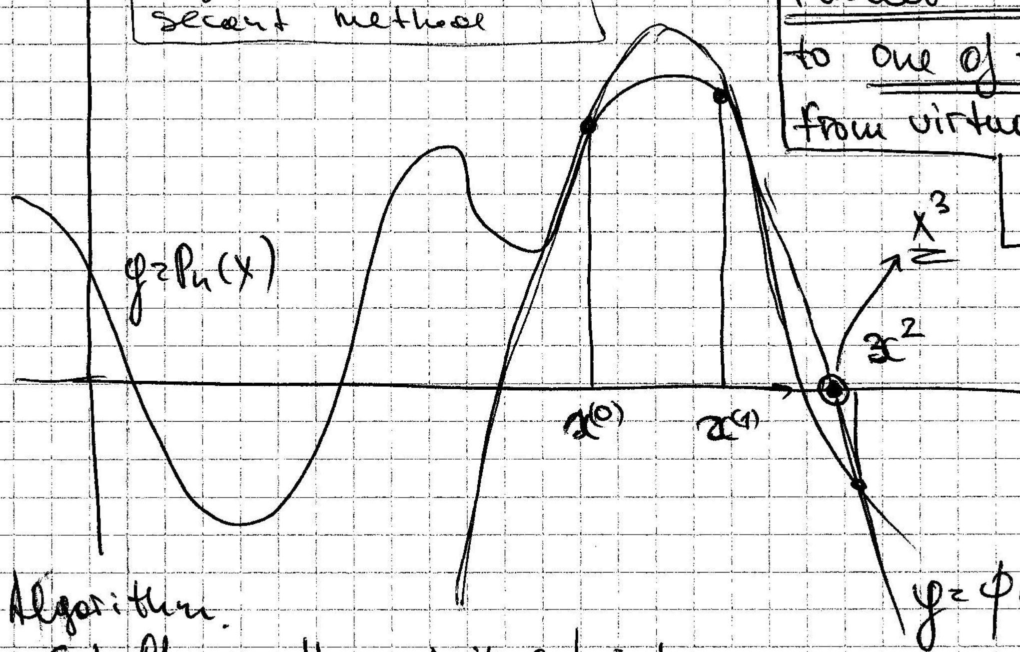
2) $\boxed{\frac{1}{2}P_n''(x_0) = \cancel{\tfrac{1}{2}} P_{n-2}(x_0)} \Rightarrow \boxed{P_n^{(k)}(x_0) = \cancel{\tfrac{1}{k}}\, k!\, P_{n-k}(x_0)}$

<u>Comment:</u>

Coefficients in $P_{n-1}(x)$; $P_{n-2}(x)$ etc. depends on x_0 root of $P_n(x)$, ~~because~~ as $a_j^n = a_{j+1}^n \underline{x_0} + a_j^{n-1} \Rightarrow$ error in x_1 propagates in coeff $a_j^1 \Rightarrow$ error of finding x_2 ~~grows~~ $\uparrow$.

2) <u>Method ~~of~~ to find all roots: Müller method.</u>

↳ generalization of secant method



<u>Müller iteration</u> converges to <u>one of the roots</u> starting from virtually <u>any</u> initial points (x_0, x_1, x_2)

!! Separating this root for the polynomial using Horner Method one can go for other roots.

Algorithm.

1) Choose three initial points
2) find parabola going through them.
3) find intersections with x-axis
4) define $x^{(2)}$ closest to root of $P_n(x)$

- Secant is <u>linear</u> interpolation method;
- Müller — " — <u>parabolic</u> — " — ;

1) $\{x_k, f(x_k)\}$, $k = j-2, j-1, j$

quadric interpolation of polynomial $\varphi(x_k) = f(x_k)$

2) $x_{j+1} = x_j + h_j q_{j+1}$; with $q_{j+1} = \dfrac{-2C_j}{B_j \pm \sqrt{B_j^2 - 4A_jC_j}}$, ↑ sign is chosen so that x_{j+1} is closest to ξ

with $h_j = (x_j - x_{j-1})$; $q_j = \dfrac{h_j}{h_{j-1}}$

$$\begin{cases} A_j = q_j f_j - q_j(1+q_j) f_{j-1} + q_j^2 f_{j-2} \\ B_j = (2q_j+1) f_j - (1+q_j)^2 f_{j-1} - q_j^2 f_{j-2} \\ C_j = (1+q_j) f_j \end{cases}$$

1) Typical initial values are $\boxed{x_0 = -1;\ x_1 = 1;\ x_2 = 0}$

2) instead of evaluating $f(x_i) = P_n(x_i)$ Müller has suggested to truncate P_n to quadric polynomial

$P_2(x) = a_0 + a_1 x + a_2 x^2 \Rightarrow$

$f_0 = P_2(x_0) = a_0 - a_1 + a_2$

$f_1 = P_2(x_1) = a_0 + a_1 + a_2$

$f_2 = P_2(x_2) = P_n(x_2) = a_0$

2) Break-off criterion, as usual

$|x_{n+1} - x_n| < \delta |x_n|$.

3) Convergence rate $p \simeq 1.84$ - simple zero

$p \simeq 1.23$ - multiple zero.

All roots: for $P_n(x) = \sum_{i=0}^{n} a_i x^i$

~~1) from initial guess x_1~~

1) to find x_1 → smallest root using Müller iteration

2) to find coefficients for $P_{n-1}(x)$ using Horner scheme

$|x_1| \leq |x_2| \leq \dots \leq |x_n|$

3) ~~Newton~~ Newton's or other method to improve $\{x_i\}$.