Interpolating polynomial Theorem

Suppose g is a function with $n+1$ derivatives and p is the interpolating polynomial of degree n through $(y_i, g(y_i))$ for $i = 0, 1, \dots, n$. Then

$$g(y) = p(y) + (y-y_0)(y-y_1)\cdots(y-y_n)\,\frac{g^{(n+1)}(c)}{(n+1)!}$$

where c is some value between $\min(y, y_0, y_1, \dots, y_n)$ and $\max(y, y_0, y_1, \dots, y_n)$.

Given a function f we trying to solve $f(x) = 0$

There points

$$(f(x_0), x_0),\ (f(x_1), x_2), \dots, (f(x_n), x_n)$$

lie on the graph of the inverse function of f. Created a polynomial p_1 of degree n (or less) such that

$$p_1(f(x_0)) = x_0,\ p_1(f(x_1)) = x_1, \dots\ p_1(f(x_n)) = x_n$$

Then evaluated $\boxed{x_{n+1} = p_1(0)}$. This is supposed to be a better approx of the root. Now consider the polynomial p_2 such that

$$p_2(f(x_1)) = x_1, \dots\ p_2(f(x_n)) = x_n,\ p_2(f(x_{n+1})) = x_{n+1}$$

and define $x_{n+2} = p_2(0)$

Define error $\varepsilon_k = x_k - r$ where r is the root we are looking for...

and let $g(y) = f^{-1}(y)$ near the root..

* if we know $g(y)$ we're done... but we don't.

Implicit differentiation to find derivatives of g.

$$f(\underbrace{g(y)}_{x}) = f(f^{-1}(y)) = y$$

$$y = f(g(y))$$

$$1 = \frac{dy}{dy} = \frac{d}{dy} f(g(y)) = f'(g(y))\, g'(y)$$

$$g'(y) = \frac{1}{f'(g(y))} = \frac{1}{f'(x)} \quad \text{where} \quad x = g(y) = f^{-1}(y).$$

$r = g(0)$ ← by definition $f(r)=0$

How does ε_{n+1} depend on the previous errors?

polynomial through the previous $n+1$ points.

$$\varepsilon_{k+1} = x_{k+1} - r = p_{k-n,n+1}(0) - g(0) = -(0-y_{k-n})(0-y_{k-n+1})\cdots(0-y_k)\frac{g^{(n+1)}(c)}{(n+1)!}$$

$$= (-1)^n y_{k-n} y_{k-n+1} \cdots y_k \frac{g^{(n+1)}(c)}{(n+1)!}$$

$$= (-1)^n f(x_{k-n}) f(x_{k-n+1}) \cdots f(x_k) \frac{g^{(n+1)}(c)}{(n+1)!}$$

for some c between $\min(0, f(x_{k-n}), f(x_{k-n+1}), \cdots, f(x_k))$ and $\max(0, f(x_{k-n}), f(x_{k-n+1}), \cdots, f(x_k))$

By the mean-value theorem

$$f(x_{k-n}) = f(x_{k-n}) - f(r) = f'(\xi_0)(x_{k-n} - r) = f'(\xi_0)\,\varepsilon_{k-n}$$

$$f(x_{k-n+1}) = f(x_{k-n+1}) - f(r) = f'(\xi_1)(x_{k-n+1} - r) = f'(\xi_1)\,\varepsilon_{k-n+1}$$

$$\vdots$$

$$f(x_k) = f(x_k) - f(r) = f'(\xi_n)(x_k - r) = f'(\xi_n)\,\varepsilon_k$$

for some ξ_n between r and x_n

$$\varepsilon_{k+1} = \underbrace{(-1)^n f'(\xi_{k-n}) f'(\xi_{k-n+1}) \cdots f'(\xi_k) \frac{g^{(n+1)}(c)}{(n+1)!}}_{\text{assume this is bounded..}} \underbrace{\varepsilon_{k-n}\, \varepsilon_{k-n+1} \cdots \varepsilon_k}$$

next error in term of the previous errors

If $x_k \to r$ then $\xi_k \to r$

$$(-1)^n f'(\xi_{k-n}) f'(\xi_{k-n+1}) \cdots f'(\xi_k) \frac{g^{(n+1)}(c)}{(n+1)!} \to C$$

Therefore, when the x_k's are near the root there is a larger constant M such that

$$\left| f'(\xi_{k-n}) f'(\xi_{k-n+1}) \cdots f'(\xi_k) \frac{g^{(n+1)}(c)}{(n+1)!} \right| \leq M$$

Thus

$$\varepsilon_{k+1} \leq M \underbrace{\varepsilon_{k-n}\, \varepsilon_{k-n+1} \cdots \varepsilon_k}_{n+1 \text{ terms}}$$

is a bound on the error.

What about when $n=2$ and we have 2^{nd} degree parabolas that approximate the inverse of f. Then

$$\varepsilon_{k+1} \leq M\, \varepsilon_{k-2}\, \varepsilon_{k-1}\, \varepsilon_k$$

Good bound...

In the limit

$$\boxed{\frac{\varepsilon_{k+1}}{\varepsilon_{k-2}\, \varepsilon_{k-1}\, \varepsilon_k} \to C \text{ as } k \to \infty.}$$

Compare with quadratic convergence of Newton's method.

Newton: $\frac{\varepsilon_{k+1}}{\varepsilon_k^2} \to C_N = \frac{1}{2}\frac{f''(r)}{f'(r)}$ as $k \to \infty$

What would the power α be if

If $\alpha > 2$ converging faster per iteration than Newton, otherwise slower...

$$\frac{\varepsilon_{k+1}}{\varepsilon_k^\alpha} \to C_2$$

constant for inverse interpolation of parabolas

we don't care what this constant is, but it need a name.

$$\boxed{\varepsilon_{k+1} \approx C_2 \varepsilon_k^\alpha} \to \boxed{\varepsilon_k \approx C_2 \varepsilon_{k-1}^\alpha}$$

substitute

$$\varepsilon_{k+1} \approx C\, \varepsilon_{k-2}\, \varepsilon_{k-1}\, \varepsilon_k$$

substitute

$$C_2 \varepsilon_k^\alpha \approx C\, \varepsilon_{k-2}\, \varepsilon_{k-1}\, \varepsilon_k$$

$$C_2 (C_2 \varepsilon_{k-1}^\alpha)^\alpha \approx C\, \varepsilon_{k-2}\, \varepsilon_{k-1}\, C_2 \varepsilon_{k-1}^\alpha$$

substitute

$$\varepsilon_{k-1} \approx C_2 \varepsilon_{k-2}^\alpha$$

$$C_2 (C_2 (C_2 \varepsilon_{k-2}^\alpha)^\alpha)^\alpha \approx C\, \varepsilon_{k-2}^1\, C_2 \varepsilon_{k-2}^\alpha\, C_2 (C_2 \varepsilon_{k-2}^\alpha)^\alpha$$

Same subscript and goes to zero.
The only way the $\approx$ could be equality in the limit is for the exponents to balance

Therefore ... $\alpha^3 = 1 + \alpha + \alpha^2$ this is cubic equation and the order of the method is the largest real root.