From last time solving $f(x)=0$ by fixed point iteration

$$x_{n+1} = g(x_n) \quad \text{where} \quad g(x) = x + c f(x)$$

and x_0 is an initial approximation of the root.

Let r be the root so $r = g(r)$ and $f(r) = 0$.

Now

$$|\varepsilon_{n+1}| = |g'(\xi_n)|\,|\varepsilon_n|$$

(by mean value theorem)

and if $x_n \to r$ then $\xi_n \to r$ and the optimal choice for c is so that

$$g'(r) = 0.$$

That means $g'(x) = 1 + c f'(x)$

$$g'(r) = 1 + c f'(r) = 0$$

so $c = -1/f'(r)$ is optimal.

Newton's method: Idea choose the value for c based on the best approximation of r iteratively.

where Newton's method comes from...

Instead of

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

use

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

best approx so far of r.

Newton's method...

More detailed analysis of Newton's method led to

$$\boxed{|\varepsilon_{n+1}| = \frac{\varepsilon_n^2}{2}\left|\frac{f''(\eta_n)}{f'(x_n)}\right|}$$

need hypothesis to ensure this is not zero.

Let M be a bound so that

interval near the root that contains all the x_n's and η_n's.

$$\frac{1}{2}\left|\frac{f''(\eta_n)}{f'(x_n)}\right| \le \frac{1}{2}\,\frac{\max\{|f''(x)| : x\in[r-\delta,r+\delta]\}}{\min\{|f'(x)| : x\in[r-\delta,r+\delta]\}} = M$$

make sure this is not zero...

Suppose $f'(r)\neq 0$ ... equivalently r is a root of multiplicity 1.

further choose δ smaller if needed so $f'(x)$ is not zero on the whole interval $[r-\delta, r+\delta]$.

Thus

$$|\varepsilon_{n+1}| \le M|\varepsilon_n|^2$$

Suppose $|\varepsilon_n| \le 5\cdot 10^{-(k+1)}$ so that x_n is accurate to the k^{th} decimal place.

$r = 1.345 \rightarrow 1.3$

$x_n = 1.399 \rightarrow 1.4$

$.054 = 5.4\times 10^{-2}$

Then

$$|\varepsilon_{n+1}| \leq M|\varepsilon_n|^2 = M\left[5 \cdot 10^{-(k+1)}\right]^2 = \underbrace{25M}\, 10^{-2k-2}$$

Let β be such that $10^\beta = 5M$ or $\beta = \dfrac{\log 25M}{\log 10}$

$$|\varepsilon_{n+1}| \leq 10^\beta \cdot 5 \cdot 10^{-2k-2} = 5 \cdot 10^{-(\underbrace{2k+1-\beta}+1)}$$

The next approx is accurate to the $(2k+1-\beta)^{th}$ decimal place.

If k is large then $2k+1-\beta$ is about double k

$$\lim_{k\to\infty} \frac{2k+1-\beta}{k} = 2$$

so the number of digits correct in the approximation (approximately) double as $k \to \infty$.

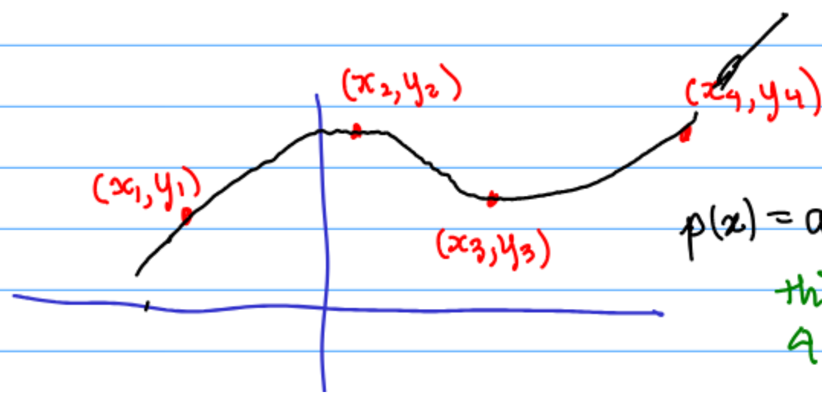
This is called superlinear convergence "quadratic" in this case...

Idea of inverse interpolation to find the root r such that $f(r)=0$.

(Vandermonde matrix)

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix} \begin{bmatrix} d \\ c \\ b \\ a \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Solve system of linear equation to get polynomial



$p(x) = ax^3 + bx^2 + cx + d$

third degree has 4 paraters.

Think of the polynomial being an approximation of the function that provided the points...

Large deviations from the data...

Sometimes the approximation doesn't look so good...

```
julia> V=[float(t[i])^j for
10x4 Matrix{Float64}:
```

Idea of inverse interpolation to find the root r such that $f(r)=0$.

Given the points $(x_0, f(x_0)), (x_1, f(x_1)) \ldots (x_n, f(x_n))$ these can be used to find a polynomial p passing through the points... Approximate f by p.

$$f(x) \approx p(x)$$

Instead of approximating f, approximate its inverse.

$$(f(x_0), x_0), (f(x_1), x_1) \ldots (f(x_n), x_n)$$ points on the graph of the inverse function

Construct the polynomial $p(y)$ that passes through these points...

$$f^{-1}(y) \approx p(y)$$

Now an approximation for the root $r = f^{-1}(0)$ can be obtained as $r = p(0)$.

Iterative algorithm. Given the points

$$(f(x_0), x_0), (f(x_1), x_1) \ldots (f(x_n), x_n)$$

define $x_{n+1} = p_n(0)$ polynomial through these points and now consider the points

$$(f(x_1), x_1) \ldots (f(x_n), x_n), (f(x_{n+1}), x_{n+1})$$ these points

Then let $x_{n+2} = p_{n+1}(0)$ where p_{n+1} is polynomial through

$$(f(x_2), x_2) \ldots, (f(x_{n+1}), x_{n+1}), (f(x_{n+2}), x_{n+2})$$

Continue ...

if this is 0 then I've found the root...

note all these y values need to be distinct in order to even find the polynomial passing through them.

Ideally as the x_n's get closer to the root the polynomial becomes a more accurate approximation of the inverse function near the root ... so

$x_{n+m+1} = p_{n+m}(0)$ is even closer to the root.

Two things left

① analyse the errors in the polynomial interpolations...

② try it out...