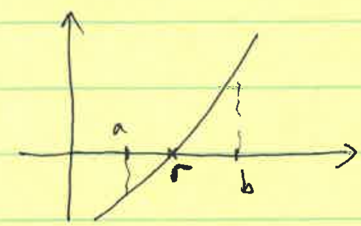
Solution of nonlinear equations of one variable

A root or a zero of a function f is a number x_* such that $f(x_*)=0$.

§3.1 The bisection method

Exploits the fact that if f is continuous on $[a,b]$ and $f(a)f(b) \leq 0$, then f must have at least one root in $[a,b]$ (This follows from the intermediate value theorem).



Idea 1) suppose, for simplicity that f has exactly one root r in (a,b).

2) Assume $f(a)\,f(b) < 0$

3) Let $c = \frac{a+b}{2}$ midpoint of $[a,b]$

$$f(a)\,f(c) \begin{cases} =0 & \Rightarrow c \text{ is root} \rightarrow \boxed{\text{STOP}} \\ <0 & \Rightarrow \text{root} \in [a,c] \qquad b \leftarrow c \quad \text{Go to } 2) \\ >0 & \Rightarrow \text{root} \in [c,b] \qquad a \leftarrow c \quad \text{Go to } 2) \end{cases}$$

Ex. $f(x) = x^3 - 3x + 1$, on $[0,1]$

n	a_n	sign $f(a_n)$	b_n	sign $f(b_n)$	c_n	$f(c_n)$	$b_n - a_n$
0	0	+	1	−	0.5	−0.375	1
1	0	+	0.5	−	0.25	+0.266	½
2	0.25	+	0.5	−	0.375	-7.23×10^{-2}	2^{-2}
3	0.25	+	0.375	−	0.3125	9.30×10^{-2}	2^{-3}
4	0.3125	+	0.375	−	0.34375	9.37×10^{-3}	2^{-4}
19					0.3472967	-9.54×10^{-7}	2^{-19}

```
procedure Bisection 2 (f, a, b, n, n-max, ε)
integer n, n-max
real a, b, c, fa, fb, fc, error
fa ← f(a)
fb ← f(b)
if sign(fa) = sign(fb) then
    output a, b, fa, fb
    output "function has same signs at a and b" => { f may not have a root in [a,b]. algorithm cannot proceed.
    return
endif
error ← b - a
for n=0 to n-max do
    error ← error/2
    c ← a + error
    fc ← f(c)
    output n, c, fc, error
    if |error| < ε or fc sufficiently small
        output "convergence"
        return
    endif
    if sign(fa) ≠ sign(fc) then
        b ← c
        fb ← fc
    else
        a ← c
        fa ← fc
    endif
end for
end procedure Bisection 2
```

Ex. How many steps of the bisection algorithm are needed to compute a root of f to full machine precision on the marc-32 ?

$(16)_{10} = 1.00000\text{-}0.0 \times 2^4$

$(17)_{10} = 1.00010\text{-}0 \times 2^4$

We already know $\underline{\underline{5}}$ bits of the answer $(1.000\overset{\text{must be zero}}{\overset{\downarrow}{0}}0 \text{ - - - -})_2$

That leaves 19 bits to determine. Thus 19 or 20 steps should be enough.

Ex. How many steps are needed on an interval of the form $[2^m, 2^{m+1}]$?

Convergence analysis

Bisection method Theorem: If the bisection algorithm is applied to a continuous fn f on $[a,b]$, where $f(a)f(b)<0$, then after n steps, an approximate root will have been computed with an error at most $\frac{b-a}{2^{n+1}}$, $\Leftarrow |r-c_n| \le \frac{b-a}{2^{n+1}}$

Note this estimate is <u>function independent</u>

<u>Ex.</u> How many steps of the bisection algorithm are needed to compute a root of f (any f) to within 10^{-7} with $a=0$, $b=2$

$$|r-c_n| \le \frac{2-0}{2^{n+1}} \le 10^{-7} \Longleftrightarrow 2^{n+1} \ge 2\times 10^{7}$$

$$(n+1)\log 2 \ge \log(2\times 10^{7}) \Longleftrightarrow n \ge \frac{\log(2\times 10^{7})}{\log 2} - 1,$$

$$24.25 \quad -1$$

$$\Rightarrow \boxed{n \ge 24}$$

{ Each step gives one more (at least) correct binary bit

The fixed point problem.

~~Given a function~~ $g: R \to R$, A fixed point problem for g takes the form: Find p such that

$$p = g(p).$$

Fixed point problems are very important in their own right. Indeed, many existence, uniqueness problems ~~are~~ solved by casting them as fixed point problems, e.g. the solution of differential equations. Here, our interest in them is limited to their relationship to root finding problems. In fact, either problem may be cast, in many different ways actually, in the form of the other.

Ex. Consider the root finding problem for $f(x) = x^4 + 2x^2 - x - 3$. A root of f, say p, can be a fixed point of $g(x) = x^4 + 2x^2 - 3$.

Other possibilities for g are

$$g_2(x) = \left(\frac{x + 3 - x^4}{2}\right)^{1/2} ; \quad g_3(x) = \left(3 + x - 2x^2\right)^{1/4} ; \; - - - -$$

As far as existence and uniqueness are concerned, we have the following

Theorem Suppose g is continuous on $[a,b]$.

(a) Suppose g maps $[a,b]$ into $[a,b]$, i.e. $a \le g(x) \le b \;\; \forall x \in [a,b]$. Then g has a fixed point in $[a,b]$.

(b) Suppose in addition that g' exists on (a,b)

and $|g'(x)| \le k < 1$, for some k. Then the fixed point is unique.

proof (a) If $g(a) = a$ or $g(b) = b$, we are done. So assume that $a < g(x) < b$, $\forall\, x \in [a,b]$. Consider the function $f(x) = g(x) - x$. Now

$$f(a) = g(a) - a > 0 \text{ and } f(b) = g(b) - b < 0.$$

Since f is also continuous, it must have a root p in (a,b) by the intermediate value theorem. Obviously p is a fixed pt. of g.

(b) Suppose there are two distinct fixed points p_1 and p_2 in $[a,b]$, i.e. $p_1 = g(p_1)$ and $p_2 = g(p_2)$. Assume without loss of generality that $p_1 < p_2$. By the Mean Value Theorem, there exists $c \in (p_1, p_2)$ such that

$$p_2 - p_1 = g(p_2) - g(p_1) \overset{\Downarrow}{=} g'(c)(p_2 - p_1).$$

Thus

$$|p_2 - p_1| = |g'(c)|\,|p_2 - p_1| \le k\,|p_2 - p_1| < |p_2 - p_1|$$

which is a contradiction. $\blacksquare$

Fixed point iteration takes the form

$$\begin{cases} p_{n+1} = g(p_n), & n = 0, 1, \ldots \\ p_0 \text{ given} \end{cases}$$

Theorem Under the conditions of the preceding theorem, for any $p_0 \in [a,b]$, the fixed point iteration generates a sequence $\{p_n\}_{n \ge 0}$ that is well-defined and converges to the unique fixed point p. Moreover

$$|p - p_n| \le k^n \max\{p_0 - a,\ b - p_0\}.$$

<u>proof.</u> First note that if $p_n = p$ for some n, then the algorithm has terminated. So assume that $p_n \neq p$ $\forall n$. Now, by the Mean Value Theorem

$$p - p_{n+1} = g(p) - g(p_n) = g'(c_n)(p - p_n),$$

for some c_n between p and p_n. In particular $c_n \in (a, b)$. We have

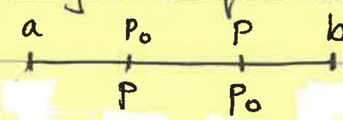
$$|p - p_{n+1}| = |g'(c_n)|\,|p - p_n| \leq k\,|p - p_n|, \quad \forall n$$

This implies

$$|p - p_n| \leq k^n |p - p_0| \quad \forall n,$$

and since $k < 1$, shows convergence of the sequence.

Finally, as the sketch on the right shows, it is the case that

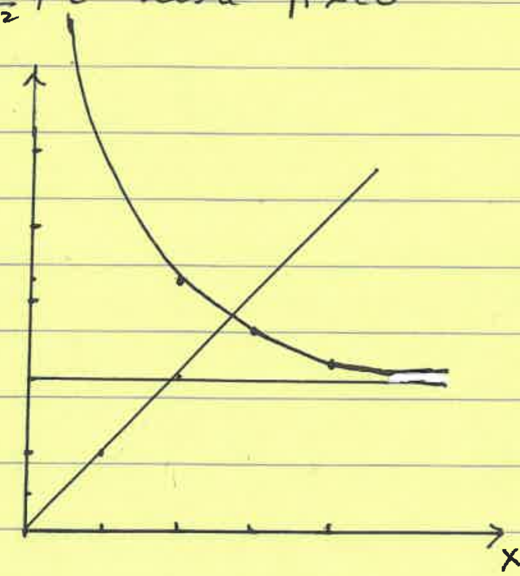


$$|p - p_0| \leq p_0 - a \quad \text{if} \quad a < p < p_0 < b$$

or

$$|p - p_0| \leq b - p_0 \quad \text{if} \quad a < p_0 < p < b. \qquad \blacksquare$$

<u>Ex.</u> Show that the function $g(x) = \frac{5}{x^2} + 2$ has a fixed point in the interval $[2, 4]$.



First, note that g is decreasing so it is enough to check values at endpoints

$g(2) = 3.25 \in [2, 4]$

$g(4) = 2.3125 \in [2, 4]$

Hence g maps $[2,4]$ into itself, so it has a fixed point in $[2,4]$.

As for convergence of the fixed point, we have $g'(x) = -10x^{-3}$. We see that $\max\limits_{2 \leq x \leq 4} |g'(x)| = \frac{10}{8} = 1.25 > 1$

Thus, in order to guarantee that the fixed point iteration will converge, we need to work with a different interval. We try $[a,b]=[2.2,4]$

$$g(2.2)=3.03306 \in [2.2,4]$$
$$g(4)=2.3125 \in [2.2,4]$$

$$k \equiv \max_{2.2\le x\le 4} |g'(x)| = \frac{10}{(2.2)^3} = .93914 < 1.$$

As far as the convergence rate of the fixed point iteration is concerned, we shall see that it depends on the value of $g'(p)$.

<u>Lemma</u> Suppose $p_{n+1}=g(p_n)$ and $\lim_{n\to\infty} p_n = p$, the fixed point of g. Then

(i) If $g'(p)\neq 0$, then the convergence is linear with $\lambda = |g'(p)|$.

(ii) If $g'(p)=0$, then the convergence is quadratic

<u>proof</u> (i) $p_{n+1}-p = g(p_n)-g(p) = g'(c_n)(p_n-p)$, for some c_n between p and p_n. We have

$$\frac{|p_{n+1}-p|}{|p_n-p|} = |g'(c_n)| \Rightarrow \lim_{n\to\infty}\frac{|p_{n+1}-p|}{|p_n-p|} = \lim_{n\to\infty}|g'(c_n)|$$
$$= |g'(p)|.$$

(ii) As above, but using Taylor's thm., since $g'(p)=0$

$$p_{n+1}-p = g(p_n)-g(p) = \cancel{g(p)} + \overset{0}{\cancel{g'(p)(p_n-p)}} + \frac{g''(c_n)}{2}(p_n-p)^2 - \cancel{g(p)}$$

$$\Rightarrow \quad p_{n+1}-p = \frac{g''(c_n)}{2}(p_n-p)^2.$$

As $n\to\infty$, $c_n \to p \Rightarrow |p_{n+1}-p| \le K|p_n-p|^2$ with $K \approx \frac{|g''(p)|}{2}$.

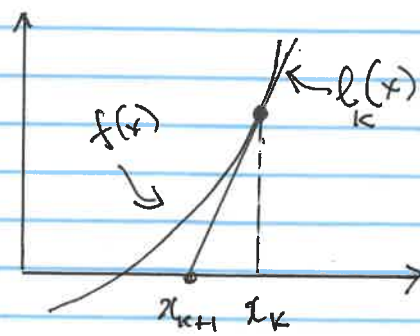
Newton's Method

Newton's (or the Newton–Raphson) Method is one of the most powerful and well-known numerical methods for solving a root-finding problem.

Idea: Suppose an approximation x_k to the root of f is known. We wish to construct a new and hopefully better approximation x_{k+1}.

Step 1: Approximate f at x_k by the tangent line to the graph of f:

$$\ell_k(x) = f(x_k) + f'(x_k)(x - x_k)$$



Step 2: If $\ell_k(x)$ is a good approximation to $f(x)$, then it makes sense to conclude that the root of $\ell_k(x)$ will be a good approximation to the root of f. So, we define x_{k+1} by

$$\ell_k(x_{k+1}) = 0 \Rightarrow \boxed{f'(x_k)(x_{k+1} - x_k) + f(x_k) = 0}$$

We can rewrite this as $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$

Newton's Algorithm.

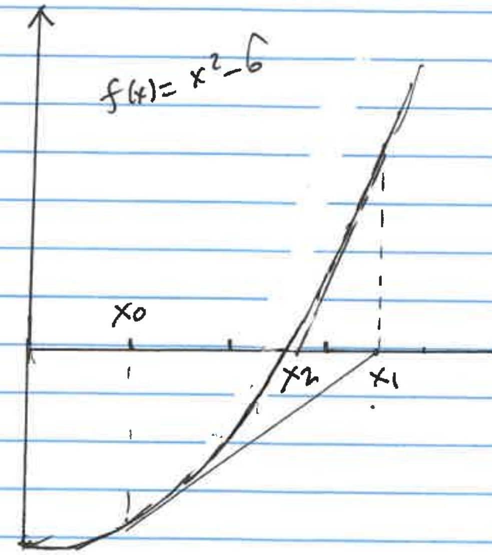
$$\begin{cases} x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \ldots \\ x_0 \text{ given.} \end{cases}$$

Remark: Newton's method is a fixed point iteration for the function $g(x) = x - \dfrac{f(x)}{f'(x)}$.

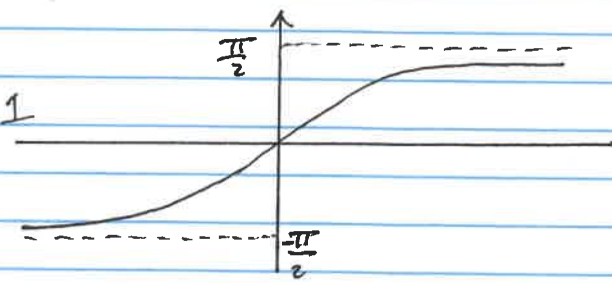
Ex. let $f(x) = x^2 - 6$ and $x_0 = 1$. perform 5 iterations of Newton's method

k	x_k
0	1
1	3.5
2	2.60714285714
3	2.45425636008
4	2.44949437161
5	2.44948974279

$$x_{k+1} = x_k - \frac{x_k^2 - 6}{2x_k} = \frac{x_k^2 + 6}{2x_k}$$



Ex. let $f(x) = \tan^{-1} x$.
Implement Newton's method with $x_0 = 5$ and $x_0 = 1$



$$x_{k+1} = x_k - \frac{\tan^{-1} x_k}{\frac{1}{1+x_k^2}}$$

$$= x_k - (1 + x_k^2) \tan^{-1} x_k.$$

k	x_k
0	5
1	-3.0708
2	1.4214×10^3
3	-3.1707×10^6
4	1.57925×10^{13}
$\downarrow$	$\downarrow$ $\pm\infty$

K	x_k
0	1
1	-5.7079632679489
2	$1.16859903998 \times 10^{-1}$
3	$-1.061022117047 \times 10^{-3}$
4	$7.963096044106 \times 10^{-10}$
5	0 to 16 digits

This example shows that even when f has a root, $r=0$ in this case, Newton's method is not guaranteed to converge.

<u>Theorem</u> Let $f \in C^2[a,b]$ i.e. f, f' and f'' exist on $[a,b]$ and are continuous. Suppose $p \in [a,b]$ is such that

(i) $f(p)=0$
(ii) $f'(p) \neq 0$.

Then, there exists $\delta > 0$ such that for any $p_0 \in [p-\delta, p+\delta]$

(a) Newton's method starting with p_0 generates a well-defined sequence $\{p_n\}$ which converges to p

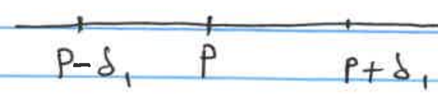
(b) the convergence is <u>quadratic</u>, i.e. there is a constant $M>0$ such that

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = L = \frac{f''(p)}{2f'(p)}, \qquad e_n = |p - p_n|.$$

Before starting the proof, what this says is that if f is sufficiently smooth <u>and</u> you start sufficiently close to the root, then Newton's method will generate a sequence which will converge to the root and do so at a quadratic rate.

<u>proof</u>

<u>Claim 1</u> There exists a $\delta_1 > 0$ such that p is the only root of f in the interval $(p-\delta_1, p+\delta_1)$.



Indeed By Taylor's Thm for $x \in (a,b)$

$$f(x) = \cancelto{0}{f(p)} + f'(p)(x-p) + \frac{f''(\xi)}{2}(x-p)^2, \ \xi \text{ between } x \text{ and } p$$

Now, the negation of the claim means that we can find a sequence $x_n \neq p$ with $x_n \to x$ and $f(x_n) = 0 \ \forall n$.
Also, since f'' is continuous on $[a,b]$, it is bounded there i.e. $\exists K > 0$ such that (by the extreme value Thm.)

$$|f''(x)| \leq K \quad \forall x \in [a,b]$$

Now for any such n $0 = f(x_n) = f'(p)(x_n - p) + \frac{f''(\xi_n)}{2}(x_n-p)^2$

which implies $|f'(p)| = \left|\frac{f''(\xi_n)}{2}(x_n-p)\right| \le K|x_n - p|$.

Note that $|f'(p)|$ has a fixed and non zero value, whereas we can make $|x_n - p|$ as small as we want. This leads to a contradiction and proves the claim.

<u>claim 2</u> There exists $\delta_2 > 0$ such that

$|f'(x)| \ge \frac{|f'(p)|}{2}$ $\forall x \in (p-\delta_2, p+\delta_2)$. In particular, f' is non zero on $(p-\delta_2, p+\delta_2)$. Now to show this

$$f'(x) = f'(p) + f''(\xi)(x-p), \quad \xi \text{ between } x \text{ and } p.$$

Let $\delta_2 = \frac{|f'(p)|}{2K}$. we have

$$|f'(x)| \ge \Big|\,|f'(p)| - |f''(\xi)(x-p)|\,\Big|.$$

Now for $x \in (p-\delta_2, p+\delta_2)$, $|x-p| \le \delta_2$. Hence

$$|f''(\xi)(x-p)| \le K\delta_2 = \frac{|f'(p)|}{2K}K = |f'(p)|/2.$$

Hence $|f'(x)| \ge |f'(p)| - \frac{|f'(p)|}{2} = |f'(p)|/2$. ✓

Now let $\boxed{M = \frac{K}{|f'(p)|}}$ and $\boxed{\delta = \min\{\delta_1, \delta_2, \frac{1}{2M}\}}$.

To prove the convergence of Newton's method for any $p_0 \in (p-\delta, p+\delta)$, assume as an induction hypothesis that Newton's method generates a sequence $p_0, \dots, p_n$ all in $(p-\delta, p+\delta)$. we will show that p_{n+1} is also well-defined and belongs to $(p-\delta, p+\delta)$. we have

$$0 = f(p) = f(p_n) + f'(p_n)(p-p_n) + \frac{f''(\xi_n)}{2}(p-p_n)^2$$

$$= \underbrace{f(p_n) + f'(p_n)(p_{n+1}-p_n)}_{=0} + f'(p_n)(p-p_{n+1}) + \frac{f''(\xi_n)}{2}(p-p_n)^2$$

Note that $f(p_n) + f'(p_n)(p_{n+1} - p_n) = 0$. (This is Newton's formula). Also, ξ_n is between p and p_n and hence belongs to $(p-\delta, p+\delta)$.

$$f'(p_n)(p - p_{n+1}) = -\frac{f''(\xi_n)}{2}(p-p_n)^2$$

$$\Rightarrow \quad |p - p_{n+1}| = \frac{|f''(\xi_n)|}{2|f'(p_n)|}(p-p_n)^2 \qquad \circledast$$

by Claim 2

$$\leq \frac{K}{2 \cdot \frac{|f'(p)|}{2}}(p-p_n)^2 = \overbrace{\frac{K}{|f'(p)|}}^{M}(p-p_n)^2$$

$$= \left(M\,|p-p_n|\right)|p-p_n|.$$

Now $|p-p_n| \leq \delta \leq \frac{1}{2M} \Rightarrow M|p-p_n| \leq 1/2$.

This shows that

$$|p - p_{n+1}| \leq \frac{1}{2}|p-p_n| \leq \delta/2$$

This closes the induction argument and proves convergence. The quadratic rate is clear from $\circledast$ since $\xi_n, p_n \to p$. ∎

<u>Remark</u> Note that Newton's method is indeed a fixed point iteration with

$$p_{n+1} = g(p_n) \quad \text{where } g(x) = x - \frac{f(x)}{f'(x)}.$$

What happens if we don't have the condition $f'(p) \neq 0$? In this case, Newton's method may still converge if $f' \neq 0$ in a neighborhood of p. But we will lose the quadratic convergence rate.

Ex. Apply Newton's method to $f(x) = x^2$

$$x_{n+1} = x_n - \frac{x_n^2}{2x_n} = \frac{x_n}{2}$$

n	x_n
0	1
1	$1/2$
2	$1/4$
3	$1/8$
4	$1/16$
⋮	

Defn. We say that p is a root of multiplicity $m \geq 1$ of f if

$$0 = f(p) = f'(p) = \cdots = f^{(m-1)}(p), \text{ but } f^{(m)}(p) \neq 0.$$

What we have seen above is that Newton's method may lose the quadratic convergence rate if the root has multiplicity greater than 1.

If the multiplicity of the root is known, then a simple modification of Newton's method can restore the quadratic convergence rate!

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

Ex. The function $f(x) = e^x - x - 1$ has a root of multiplicity 2 at $x = 0$

n	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$	$x_{n+1} = x_n - 2\frac{f(x_n)}{f'(x_n)}$
0	$1.$	1
1	$5.819767068 \times 10^{-1}$	1.63953×10^{-1}
2	$3.1905504091 \times 10^{-1}$	4.4781×10^{-3}
3	$1.6799617288 \times 10^{-1}$	3.342×10^{-6}
4	$8.6348873 74 \times 10^{-2}$	1.086×10^{-11}
5	$4.37957036 \times 10^{-2}$	$1.0864531 \times 10^{-11}$

If the multiplicity m is unknown, then we can apply Newton's method to the function

$$\mu(x) = \frac{f(x)}{f'(x)}.$$

Indeed, suppose p is a root of multiplicity m of f. Then we can write

$$f(x) = (x-p)^m q(x) \quad \text{with} \quad q(p) \neq 0.$$

Now

$$\mu(x) = \frac{(x-p)^m q(x)}{m(x-p)^{m-1} q(x) + (x-p)^m q'(x)}.$$

$$= (x-p)\frac{q(x)}{m\,q(x) + (x-p)\,q'(x)} \equiv (x-p)\,\rho(x)$$

Clearly p is a root of μ. On the other hand

$\rho(p) = \frac{q(p)}{m\,q(p)} = \frac{1}{m} \neq$ since $q(p) \neq 0$. Thus p is a root of multiplicity 1 of μ.

$$\Rightarrow \quad x_{n+1} = x_n - \frac{\mu(x_n)}{\mu'(x_n)}.$$

The secant method

In case $f'(x)$ is not available or is difficult to evaluate, we can replace $f'(x_n)$ in Newton's formula by the difference quotient

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Secant Method

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

Note starts at 1

$$= x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, \ldots$$

For the secant method two starting values x_0, x_1 are needed. Also, we assume that $x_{n-1} \neq x_n$, otherwise the method has converged.

Convergence is slower than Newton's method.

It can be shown that

$$|x_{n+1} - r| \le c\,|x_n - r|^{\alpha}, \quad \alpha = \frac{1}{2}(1+\sqrt{5}) \approx 1.62.$$

vs. $|x_{n+1} - r| \le c\,|x_n - r|^2$ for Newton's method.

So it is slower than quadratic, but faster than superlinear.

Accelerating convergence: Aitken's Δ^2 method

Newton's method is quadratically convergent, under certain conditions and this is considered as "fast".
However, linear convergence rate can be very slow especially if $\lambda \approx 1$.

Given a sequence which converges, there are ways to construct a new sequence (derived) which has the potential of faster convergence.

Suppose $\{p_n\}_{n=0}^{\infty}$ converges linearly to p, i.e.

$$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|} = \lambda \quad , 0<\lambda<1.$$

Motivated by this, it is reasonable to assume that

$$\left|\frac{p_{n+1}-p}{p_n-p}\right| \approx \left|\frac{p_{n+2}-p}{p_{n+1}-p}\right| \quad \forall n \Leftarrow \text{both ratios} \to \lambda$$

Further assume that the signs agree

$$\Rightarrow \frac{p_{n+1}-p}{p_n-p} \approx \frac{p_{n+2}-p}{p_{n+1}-p}.$$

Solving for p, after some algebra, we get

$$p \approx \frac{p_{n+2}\,p_n - p_{n+1}^2}{p_{n+2}-2p_{n+1}+p_n}.$$

we may leave this as is, but we prefer to rewrite it as follows:

$$p \approx \frac{p_{n+2}\,p_n - \cancel{p_{n+1}^2} - 2p_n p_{n+1} + p_n^2 + 2p_n p_{n+1} - p_n^2 - p_{n+1}^2}{p_{n+2}-2p_{n+1}+p_n}$$

$$p = p_n - \frac{(p_{n+1}-p_n)^2}{p_{n+2}-2p_{n+1}+p_n}$$

This motivates setting $$\boxed{\hat{P}_n = P_n - \frac{(P_{n+1}-P_n)^2}{P_{n+2}-2P_{n+1}+P_n}, \quad n=0,1,\dots}$$

Forward differences.

Given a sequence $\{P_n\}_{n=0}^{\infty}$, we define the forward difference operator Δ by

$$\boxed{\Delta P_n = P_{n+1} - P_n, \quad n=0,1,\dots}$$

we can define powers of Δ as follows

$$\Delta^2 P_n = \Delta(\Delta P_n) = \Delta(P_{n+1}-P_n) = \Delta P_{n+1} - \Delta P_n$$
$$= P_{n+2} - P_{n+1} - (P_{n+1} - P_n)$$
$$= P_{n+2} - 2P_{n+1} + P_n$$

Note that Aitken's formula can be expressed in terms of Δ, Δ^2 as follows

$$\boxed{\hat{P}_n = P_n - \frac{(\Delta P_n)^2}{\Delta^2 P_n}, \quad n=0,1,\dots}$$

Ex. Apply Aitken's Δ^2-method to the fixed point iteration

$$\begin{cases} P_{n+1} = e^{-P_n} & n=0,1,\dots \\ P_0 = 1 \end{cases}$$

```
p[0] = 1.0000000000000000e+00
p[1] = 3.6787944117144234e-01
p[2] = 6.9220062755534639e-01   ptilde[0] = 5.8222609699562299e-01
p[3] = 5.0047350056363681e-01   ptilde[1] = 5.7170576752725210e-01
p[4] = 6.0624353508559736e-01   ptilde[2] = 5.6863880586446602e-01
p[5] = 5.4539578597502702e-01   ptilde[3] = 5.6761699484663541e-01
p[6] = 5.7961233550337887e-01   ptilde[4] = 5.6729675248863387e-01
p[7] = 5.6011546136108914e-01   ptilde[5] = 5.6719242788720647e-01
p[8] = 5.7114311508017701e-01   ptilde[6] = 5.6715913383400073e-01
p[9] = 5.6487934739104950e-01   ptilde[7] = 5.6714837922695837e-01
p[10] = 5.6842872502906072e-01  ptilde[8] = 5.6714492852985132e-01
p[11] = 5.6641473314688328e-01  ptilde[9] = 5.6714381707466432e-01
p[12] = 5.6755663732828343e-01  ptilde[10] = 5.6714345985563219e-01
p[13] = 5.6690891192149528e-01  ptilde[11] = 5.6714334490436946e-01
p[14] = 5.6727623217556955e-01  ptilde[12] = 5.6714330793949274e-01
p[15] = 5.6706789839078841e-01  ptilde[13] = 5.6714329604797764e-01
p[16] = 5.6718605009935696e-01  ptilde[14] = 5.6714329222336545e-01
p[17] = 5.6711904005721491e-01  ptilde[15] = 5.6714329099311633e-01
p[18] = 5.6715704400129751e-01  ptilde[16] = 5.6714329059741530e-01
p[19] = 5.6713549020627840e-01  ptilde[17] = 5.6714329047013545e-01
```

It is clear that the new sequence $\hat{p}_n$ is converging faster as indicated by the agreements between digits in successive iterations.

<u>Theorem</u> Suppose $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges linearly to the limit p. Sequence $\{\hat{p}_n\}$ converges faster to p in the sense that

$$\lim_{n\to\infty} \frac{\hat{p}_n - p}{p_n - p} = 0.$$

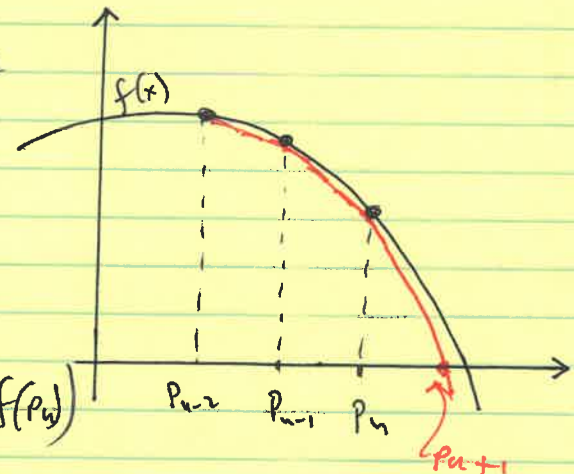
<u>Steffensen's Method</u> Is a structured application of Aitken's Δ^2 method to a sequence generated by the fixed point iteration.

$$\left.\begin{array}{l} p_0^{(0)} = p_0 \text{ Given} \\ \downarrow \\ p_1^{(0)} = g(p_0) \\ \downarrow \\ p_2^{(0)} = g(p_1) \end{array}\right\} \Rightarrow \left.\begin{array}{l} \hat{p}_0^{(1)} = \text{Aitken}\{p_0^{(0)}, p_1^{(0)}, p_2^{(0)}\} \\ \downarrow \\ p_1^{(1)} = g(p_0^{(1)}) \\ \downarrow \\ p_2^{(1)} = g(p_1^{(1)}) \end{array}\right\} \Rightarrow p_0^{(2)} = \text{Aitken}\{p_0^{(1)}, p_1^{(1)}, p\ldots$$

Müller's method: An extension of the secant method.

Recall that the secant method uses the last two approximations p_{n-1} and p_n to the root to construct p_{n+1}, ~~this~~ which is defined as the root of the line or affine function passing through $(p_{n-1}, f(p_{n-1}))$ and $(p_n, f(p_n))$.

Müller's method uses the last 3 approximations p_{n-2}, p_{n-1}, p_n to construct p_{n+1}.



This is done by constructing the unique (in general) quadratic polynomial $q(x)$ which passes through the 3 points (assumed distinct)

$(p_{n-2}, f(p_{n-2})), (p_{n-1}, f(p_{n-1})), (p_n, f(p_n))$

and letting p_{n+1} be the root of $q(x)$.

It takes some algebra, but $q(x)$ can be constructed explicitely. Its roots (note the difference with secant) are given by

$$P_3 = P_2 - \frac{2c}{b \pm \sqrt{b^2-4ac}} = P_2 - \frac{2c}{b + \operatorname{sign}(b)\sqrt{b^2-4ac}}$$

where $q(x) = a(x-p_2)^2 + b(x-p_2) + c$.

In Müller's method we choose the sign of the radical to agree with that of b. This maximizes the denominator resulting in the choice of the root of q which is the closest to P_2.

Roots of polynomials.

Theorem (Fundamental Theorem of Algebra) If $p(x)$ is a polynomial of degree $n \geq 1$ with real or complex coefficients, then $p(x)=0$ has at least one (possibly complex) root.

Remark A polynomial with real coefficients may have complex roots.

Corollary A polynomial of degree $n \geq 1$ has exactly n roots (counting multiplicities). In particular we can write

$$P(x) = c(x-r_1)^{m_1} \cdots (x-r_s)^{m_s}$$

where $r_1, \ldots, r_s$ are the distinct roots of P, m_i is the multiplicity of root r_i and c is a constant.

Horner's method provides an efficient way for evaluating a polynomial at a given number. It also turns out that much more information can be gleaned from an application of this method/algorithm.

Theorem let $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ and let x_0 be given.
Define $b_n = a_n$

$$b_k = a_k + b_{k+1} x_0 \quad \text{for } k = n-1, \ldots, 1, 0.$$

Then

$$P(x) = (x-x_0) Q(x) + b_0$$

where

$$Q(x) \equiv b_n x^{n-1} + \cdots + b_2 x + b_1$$

Proof.
From the definition of $Q(x)$

$$(x-x_0)Q(x)+b_0 = (x-x_0)(b_n x^{n-1} + \cdots + b_2 x + b_1) + b_0$$

$$= b_n x^n + b_{n-1}x^{n-1} + \cdots + b_2 x^2 + b_1 x$$
$$- b_n x_0 x^{n-1} \quad - b_3 x_0 x^2 - b_2 x_0 x - b_1 x_0 + b_0$$

$$= b_n x^n + (b_{n-1} - b_n x_0)x^{n-1} + \cdots + (b_2 - b_3 x_0)x^2 + (b_1 - b_2 x_0)x$$
$$+ (b_0 - b_1 x_0)$$

$$= b_n x^n + \sum_{k=0}^{n-1} (b_k - b_{k+1}x_0)x^k .$$

By definition, $b_n = a_n$ and

$$b_k = a_k + b_{k+1}x_0 \Rightarrow b_k - b_{k+1}x_0 = a_k \qquad k = 0, \ldots, n-1$$

$$\Rightarrow \qquad = \sum_{k=0}^{n} a_k x^k . \qquad \blacksquare$$

Remark From $P(x) = (x-x_0)Q(x) + b_0$

it follows readily that $\boxed{P(x_0) = b_0}$.

There is more information embodied in this relation. Indeed, it provides an algorithm for "dividing" the polynomial $P(x)$ by the factor $x - x_0$ yielding the "quotient" $Q(x)$ and remainder b_0.

Also, differentiating, we get

$$P'(x) = Q(x) + (x-x_0)Q'(x)$$

$$\Rightarrow \boxed{P'(x_0) = Q(x_0)}$$

Thus $Q(x_0)$ is the value of the derivative of $P(x)$ at x_0. This can be useful if Newton's method is applied to find a root of $P(x)$.

Ex. $p(x) = 2x^4 - 3x^2 + 3x - 4$, $x_0 = -2$

	a_4	a_3	a_2	a_1	a_0
	2	0	-3	3	-4
-2	2	-4	5	-7	$\boxed{10}$
	b_4	b_3	b_2	b_1	b_0

$$\Rightarrow 2x^4 - 3x^2 + 3x - 4 = (x+2)(2x^3 - 4x^2 + 5x - 7) + 10.$$

Deflation This is yet another application of Horner's method. It can be used to find the remaining roots of a polynomial once a good approximation of one root has been found.

To begin, suppose the roots of $p_n(x)$ are denoted by $r_1, \ldots, r_n$. If r_1 is known, then

$$p_n(x) = (x - r_1)\, Q(x) + \overbrace{p_n(r_1)}^{= b_0 = 0},$$

hence $Q(x)$ has $r_2, \ldots, r_n$ as roots.

More generally, suppose a good approximation $\hat{r}_1$ to r_1 has been calculated. Then

$$p_n(x) = (x - \hat{r}_1)\, p_{n-1}(x) + p_n(\hat{r}_1).$$

If $\hat{r}_1 \approx r_1$ $\overset{\Rightarrow\, p_n(\hat{r}_1) \approx 0}{,}$ then we write $p_n(x) \approx (x - \hat{r}_1)\, p_{n-1}(x)$

and argue that $p_{n-1}(x)$ has roots $\hat{r}_2, \ldots, \hat{r}_n$ which are good approximations to $r_2, \ldots, r_n$.

Effectively, we have eliminated root r_1 (assuming it is simple). This is referred to as deflation.

We can next apply a root finding method to find a root of $p_{n-1}(x)$, say r_2, deflate it, etc. This way, all the roots of $p_n(x)$ can be approximated.

However, it should be noted that due to its very nature, errors tend to accumulate, since we only have the approximate equality

$$P_m(x) \approx (x - \hat{r}_{n-m+1}) P_{m-1}(x).$$

Hence, once approximations $\hat{r}_1, \dots, \hat{r}_n$ have been obtain, it is a good idea to <u>refine</u> them using the original polynomial $P_n(x)$ and Newton's or some other method.

<u>Ex.</u> consider the polynomial $P_3(x) = 20x^3 - 30x^2 + 12x - 1$.

Suppose we know that $\hat{r}_1 = .499$ is a good approximation to one of the 3 roots of $P_3(x)$.

$$\begin{array}{c|cccc} & 20 & -30 & 12 & -1 \\ \hline .499 & 20 & -20.02 & 2.01002 & .00299998 \end{array}$$

This shows that

$$20x^3 - 30x^2 + 12x - 1 \approx (x - .499)\underbrace{(20x^2 - 20.02x + 2.01002)}_{P_2(x)}$$

Now we can apply some root finding process to approximate a second root from $P_2(x)$ and deflate etc.

But since $P_2(x)$ is quadratic, we use directly the quadratic formula:

$$r_2, r_3 \approx \frac{20.02 \pm \sqrt{(20.02)^2 - 4(20)(2.01002)}}{40}$$

$$\nearrow .113202 6336$$
$$\searrow .887797366$$

$$\Rightarrow \hat{r}_1 \cong .499, \quad r_2 \cong .113202 6336, \quad r_3 \approx .887797366$$

We now refine these approximations by using them as starting values for Newton's method using the original polynomial $P_3(x)$.

n	r_1	r_2	r_3
0	4.98999999999999999e-01	1.13202633600000005e-01	8.87797366000000032e-01
1	5.00000013333599780e-01	1.12700690437272877e-01	8.87299296227791023e-01
2	5.00000000000000333e-01	1.12701665375577020e-01	8.87298334624323393e-01
3	4.99999999999999722e-01	1.12701665379258298e-01	8.87298334620741702e-01
4	5.00000000000000333e-01	1.12701665379258312e-01	8.87298334620742035e-01
5	4.99999999999999722e-01	1.12701665379258312e-01	8.87298334620742035e-01

Roots of polynomials via eigenvalues

Finding all the roots of a polynomial can be a daunting task say if we use Newton's method coupled with deflation.

On the other hand, given any polynomial, we can construct a square matrix whose characteristic polynomial is precisely the polynomial that was given. Hence a variety of methods for finding eigenvalues may be used.

Given the monic polynomial

$$P(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$$

The $n \times n$ matrix

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & 0 \cdots & 0 & -c_2 \\ & & \ddots & & \vdots \\ & & & 1 & -c_{n-1} \end{bmatrix}$$

is called the Frobenius companion matrix to $P(x)$.

Ex show that the characteristic polynomial of C is equal to $P(x)$, i.e.

$$\det(\lambda I - C) = P(x).$$

For $p(x) = 20x^3 - 30x^2 + 12x - 1$. we first write p as monic

$$\Rightarrow \quad x^3 - \frac{3}{2}x^2 + \frac{3}{5}x - \frac{1}{20}$$

$$\Rightarrow \quad C = \begin{bmatrix} 0 & 0 & 1/20 \\ 1 & 0 & -3/5 \\ 0 & 1 & 3/2 \end{bmatrix}.$$

Using the Matlab function eig(C) we find the eigenvalues to be

$.4\bar{9}$, $.112701665379259$, $.887298334620741$

Starting and stopping criteria for iterative methods.

An *iterative method*, e.g. Newton, Secant etc., generate a sequence of approximations $\{x_n\}_{n\geq 0}$ which more often than not are infinite. The x_n's could be real or complex numbers, vectors or even matrices.

Two important issues present themselves immediately:

1) How to choose the starting/initial x_0 in an "appropriate" way. A poor choice of x_0 may lead to slow convergence or even divergence.

2) When to stop the iteration? Here of course the goal is to stop the iteration when a "sufficiently" accurate approximation x_n has been obtained.

Let us say from the onset that there are no satisfactorily general answers to either of these concerns. Such issues must be dealt with within the context of the specific problem at hand.

Concerning the stopping criterion, a commonly used practice is to stop the iteration when $\|x_n - x_{n-1}\| < \epsilon$ where ϵ is a user specified tolerance/error level. While simple and often effective, this approach can lead to unsatisfactory results as the following example shows

Ex. let $x_n = (.999999)^n$, $n \geq 0$.

Obviously $\lim_{n\to\infty} x_n = 0$. Also, $|x_{11} - x_{10}| \approx 10^{-6}$ whereas you need $n \geq 13{,}815{,}504$ in order to have $|x_n - 0| \leq 10^{-6}$.

The problem here was that the convergence of the sequence $\{x_n\}$ was only linear and λ was very close to 1. On the other hand we can prove that in case a sequence converges "fast" then the above criterion can be effective.

Lemma Suppose the sequence $\{x_n\}_{n\geq 0}$ converges superlinearly to x, i.e.

$$\lim_{n\to\infty} \frac{|x_{n+1} - x|}{|x_n - x|} = 0.$$

Then

$$\lim_{n\to\infty} \frac{|x_{n+1} - x_n|}{|x_n - x|} = 1.$$

proof. let $s_n = x_{n+1} - x_n$ and $e_n = x_n - x$.

$$\lim_{n\to\infty} \left| \frac{|s_n|}{|e_n|} - 1 \right| = \lim_{n\to\infty} \left| \frac{|s_n| - |e_n|}{|e_n|} \right|$$

$$\leq \lim_{n\to\infty} \left| \frac{|s_n + e_n|}{e_n} \right| \quad \text{triangle inequality}$$

$$= \lim_{n\to\infty} \left| \frac{e_{n+1}}{e_n} \right| = 0. \quad \blacksquare$$

This shows that for n large, $|x_{n+1} - x_n| \approx |x - x_n|$ so the stopping criterion $|x_{n+1} - x| \leq \epsilon$ becomes reliable.