Actually:
$$f(x) = x^2 \underbrace{\frac{e^x - x - 1}{x^2}}_{g(x)}$$

where
$$\lim_{x\to 0} g(x) \underset{\text{l'hopital}}{=} \lim_{x\to 0} \frac{e^x - 1}{2x} = \lim_{x\to 0} \frac{e^x}{2} = \frac{1}{2} \neq 0$$

limit can be evaluated using Taylor's theorem too

See book for numerical results.

How to fix this?

let $\mu(x) = \dfrac{f(x)}{f'(x)}$. If x_* is a zero of multiplicity q of f then:
$$f(x) = (x - x_*)^q g(x)$$

$$\Rightarrow \mu(x) = \frac{(x-x_*)^q g(x)}{q(x-x_*)^{q-1} g(x) + (x-x_*)^q g'(x)}$$
$$= (x - x_*) \frac{g(x)}{q g(x) + (x-x_*) g'(x)}$$

We also have $\mu(x_*) = 0$.

However: $g(x_*) \neq 0$ so:
$$\lim_{x \to x_*} \frac{g(x)}{q g(x) + (x-x_*) g'(x)} = \frac{1}{q} \neq 0$$

therefore $\mu(x)$ has a zero of order 1 (simple) at x_* and we can use Newton's method on $\mu(x)$.

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

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

In principle one should get quadratically convergent iterates regardless of multiplicity of root with this method. The only extra cost is that iterates are more difficult to compute and require $f''(x_n)$.

§2.5 Accelerating Convergence: Aitken's method and Steffensen's method.

Consider a sequence $p_n \to p$ for which:

$$(*) \qquad p_{n+1} - p = (c + \delta_n)(p_n - p) \quad \text{where } \delta_n \to 0 \text{ as } n \to \infty. \text{ and } |c| < 1.$$

then:

$$\frac{p_{n+1} - p}{p_n - p} = c + \delta_n \approx c$$

$$\frac{p_{n+2} - p}{p_{n+1} - p} = c + \delta_{n+1} \approx c.$$

$$\Rightarrow \quad (p_{n+1} - p)^2 \approx (p_{n+2} - p)(p_n - p)$$

$$p_{n+1}^2 - 2p\,p_{n+1} + p^2 \approx p_{n+2}\,p_n - (p_n + p_{n+2})p + p^2$$

$$(p_{n+2} - 2p_{n+1} + p_n)\,p \approx p_{n+2}\,p_n - p_{n+1}^2$$

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

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

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

Notation:

Forward difference: $\Delta p_n = p_{n+1} - p_n$

$$\Delta^k p_n = \Delta(\Delta^{k-1} p_n)$$

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

We can define a new iteration $\hat{p}_n$ as follows:

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

This iteration is called Aitken's Δ^2 method or simply Aitken's acceleration

Theorem (Aitken's acceleration) Assuming p_n satisfies cdt° (*)

the new sequence $\hat{p}_n$ converges faster to p than p_n in the sense that:

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

Proof: Let $h_n = p_n - p$ then:

$$\hat{p}_n = h_n + p - \frac{(\Delta h_n)^2}{\Delta^2 h_n}$$

$\Delta p_n = p_{n+1} - p_n = h_{n+1} - h_n = \Delta h_n$

etc...

also:

$h_{n+1} = (c+\delta_n) h_n$

$h_{n+2} = (c+\delta_{n+1})(c+\delta_n) h_n$

we have:

$$\Delta h_n = (c+\delta_n - 1) h_n$$

$$\Delta^2 h_n = (c+\delta_{n+1} - 1) h_{n+1} - (c+\delta_n - 1) h_n$$

$$= (c+\delta_{n+1} - 1)(c+\delta_n) h_n - (c+\delta_n - 1) h_n$$

Thus.

$$\hat{p}_n - p = \frac{h_n \Delta^2 h_n - (\Delta h_n)^2}{\Delta^2 h_n}$$

$$= \frac{h_n^2 (\Delta^2 h_n / h_n) - (c+\delta_n - 1)^2 h_n^2}{h_n (\Delta^2 h_n / h_n)}$$

$$= h_n \frac{(\Delta^2 h_n / h_n) - (c+\delta_n - 1)^2}{(\Delta^2 h_n / h_n)}$$

now: $$\frac{\Delta^2 h_n}{h_n} = (c+\delta_{n+1}-1)(c+\delta_n) - (c+\delta_n - 1)$$
$$\to (c-1)c - (c-1) = (c-1)^2 \neq 0.$$
as $n \to \infty$

therefore:
$$\frac{\hat{p}_n - p}{h_n} = \frac{\hat{p}_n - p}{p_n - p} = -\frac{(\Delta^2 h_n / h_n) - (c+\delta_n-1)^2}{\Delta^2 h_n / h_n}$$

$$\left.\begin{aligned}&\text{numerator} \to 0\\ &\text{denominator} \to (c-1)^2 \neq 0\end{aligned}\right\} \Rightarrow \frac{\hat{p}_n - p}{p_n - p} \to 0.$$

QED.

Aitken's method can be used to improve the linear convergence rate of fixed point iteration to quadratic. Here is the idea:

<u>Aitken</u>

$$\begin{aligned}
&p_0\\
&p_1 = F(p_0)\\
&p_2 = F(p_1)\\
&p_3 = F(p_2)\\
&p_4 = F(p_3)
\end{aligned}$$

$\left.\begin{aligned}&p_0\\&p_1\\&p_2\end{aligned}\right\}\hat{p}_0 \qquad \left.\begin{aligned}&p_1\\&p_2\\&p_3\end{aligned}\right\}\hat{p}_1 \qquad \left.\begin{aligned}&p_2\\&p_3\\&p_4\end{aligned}\right\}\hat{p}_2$

<u>Steffensen</u>

$$\begin{aligned}
&p_0^{(0)}\\
&p_1^{(0)} = F(p_0^{(0)})\\
&p_2^{(0)} = F(p_1^{(0)})\\
&p_0^{(1)} = \text{Aitken}(p_0^{(0)}, p_1^{(0)}, p_2^{(0)})\\
&p_1^{(1)} = F(p_0^{(1)})\\
&p_2^{(1)} = F(p_1^{(1)})\\
&p_0^{(2)} = \text{Aitken}(p_0^{(1)}, p_1^{(1)}, p_2^{(1)})\\
&\vdots
\end{aligned}$$

Aitken estimate of sol is used in fixed point iteration.

§2.6 Computing roots of a polynomial

We start with some background on polynomials:

A polynomial of degree n has the form:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0$$

with $a_n \neq 0$.

Theorem Fundamental theorem of algebra (Gauss)

Every non-constant polynomial has at least one (possibly complex) root.

⚠ FTA does not guarantee a polynomial will have real roots:

$$p(z) = z^2 + 1$$

Remainder Theorem

We can write for $p(z)$ of degree n:

$$p(z) = (z-c)\,q(z) + r$$

(this is a generalization of Euclidian division to polynomials)

$q(z)$ = quotient, degree $n-1$

r = remainder $\in \mathbb{C}$.

Factor theorem: c is a root of $p(z)$ iff $r=0$ i.e.

$$p(z) = (z-c)\,q(z)$$

Then we can show using an induction argument that:

Theorem A polynomial of degree n has exactly n roots in the complex plane (each root is counted as many times as its multiplicity) proof: By FTA $p(z) = (z - r_1)\, q_1(z)$

By reapplying FTA to $q_1(z)$ it has at least one root r_2 -

$$\Rightarrow \quad p(z) = (z - r_1)(z - r_2)\, q_2(z)$$

$$\vdots$$

$$p(z) = (z - r_1)(z - r_2) \cdots (z - r_n) \underbrace{q_n}_{\in \mathbb{C}} \qquad (*)$$

roots may be repeated.

multiplicity = # of times a given root is repeated as a factor of $p(z)$.

It is clear from $(*)$ that the only way a polynomial of degree n has more than n roots is for $p \equiv 0$. thus we get the important corollary:

Corollary: Let $p(z)$ and $q(z)$ be two polynomials of degree $\leq n$
If $z_1, z_2, \ldots, z_k$ are $k > n$ distinct numbers for which:

$$p(z_i) = q(z_i) \quad \text{then} \quad p(z) = q(z) \ \forall z \in \mathbb{C}.$$

proof: $r(z) = p(z) - q(z)$ has degree $\leq n$

but $r(z)$ has $k > n$ distinct roots:

$$r(z_i) = p(z_i) - q(z_i) = 0$$

$$\Rightarrow r(z) \equiv 0 \qquad \Rightarrow p(z) = q(z).$$

Horner's Algorithm :

Efficient way of computing $q(z)$ an $p(z_0)$ s.t.

$$p(z) = (z - z_0)\, q(z) + p(z_0) \qquad (*)$$

⇝ can be thought as an specialization of polynomial division to linear factors. $z - z_0$.

⇝ can be used to implement efficiently Newton's method to find polynomial roots.

Idea: Let $p(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_2 z^2 + a_1 z + a_0$ be a polynomial of degree n.

Then $q(z)$ is of degree $n-1$ and can be written:

$$q(z) = b_{n-1} z^{n-1} + b_{n-2} z^{n-2} + \dots + b_2 z^2 + b_1 z + b_0$$

Equating powers of z in $(*)$ we get:

z^n coeff: $\quad a_n = b_{n-1}$

z^{n-1} coeff: $\quad a_{n-1} = b_{n-2} - z_0 b_{n-1}$

z^{n-2} coeff: $\quad a_{n-2} = b_{n-3} - z_0 b_{n-2}$

$\vdots \qquad \vdots$

z coeff: $\quad a_1 = b_0 - z_0 b_1$

z^0 coeff: $\quad a_0 = p(z_0) - z_0 b_0$

Thus we can solve for $b_{n-1}, b_{n-2}, \dots, b_1, b_0, p(z_0) \equiv b_{-1}$ in this order in the previous equations.

Horner's algorithm:

inputs: n, $a_0, a_1, \dots, a_n$ and z_0.

$b_{n-1} = a_n$

for $k = n-1:-1:0$

$\quad b_{k-1} = a_k + z_0 b_k$

where we used the convention $b_{-1} \equiv p(z_0)$.

This algorithm can be carried out by hand as follows.

"synthetic" division

$$\begin{array}{c|ccccc} & a_n & a_{n-1} & a_{n-2} & \cdots & a_0 \\ z_0 & & z_0 b_{n-1} & z_0 b_{n-2} & \cdots & z_0 b_0 \\ \hline & b_{n-1} & b_{n-2} & b_{n-3} & \cdots & \boxed{b_{-1}} \end{array}$$

<u>Example</u>: Use Horner's algo to evaluate $p(3)$ where:

$$p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & \boxed{19} \end{array}$$

thus $p(z) = (z-3)(z^3 - 3z^2 + 12z + 21) + 19$

and $p(3) = 19$.

As an added bonus one can compute derivative since:

$$p(z) = (z - z_0)\, q(z) + p(z_0)$$

$$p'(z) = (z - z_0)\, q'(z) + q(z)$$

$\Rightarrow p'(z_0) = q(z_0)$ which can be evaluated again using Horner's method.

- <u>Horner's method</u> can be used for <u>deflation</u>, i.e. removing a linear factor from a polynomial. For example:

$$p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2 \quad \text{(as before)}$$
$$z_0 = 2$$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 2 & & 2 & -4 & 6 & 2 \\ \hline & 1 & -2 & 3 & 1 & \boxed{0} \end{array}$$

$$\Rightarrow p(z) = (z-2)\underbrace{(z^3 - 2z^2 + 3z + 1)}_{q(z)}$$

then the remaining roots of $p(z)$ are the roots of $q(z) = \dfrac{p(z)}{z - z_0}$

- <u>Horner's method</u> can also be used to find <u>Taylor expansion</u> of a polynomial about a point:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_0$$
$$= c_n (z - z_0)^n + c_{n-1}(z - z_0)^{n-1} + \dots + c_0$$

Taylor's theorem implies: $c_k = \dfrac{p^{(k)}(z_0)}{k!}$

but this can be error prone to compute!

<u>Idea</u>: $p(z_0) = c_0 \rightsquigarrow$ <u>apply Horner</u> to $p(z)$ at z_0 to obtain c_0

we also get the polynomial:

$$q(z) = \frac{p(z) - p(z_0)}{z - z_0} = c_n (z - z_0)^{n-1} + c_{n-1}(z - z_0)^{n-2} + \dots + c_1$$

$\Rightarrow$ <u>apply Horner</u> again to compute $c_1 = q(z_0)$

repeat until all c_k are found.

Example (*) Find Taylor expansion of $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$ about the point $z_0 = 3$.

We apply Horner's method successively: "complete" Horner's method

		1	-4	7	-5	-2
$k=0$	3		3	-3	12	21
		1	-1	4	7	$\boxed{19}$
$k=1$	3		3	6	30	
		1	2	10	$\boxed{37}$	
$k=2$	3		3	15		
		1	5	$\boxed{25}$		
$k=3$	3		3			
		$\boxed{1}$	$\boxed{8}$			

Thus: $p(z) = 19 + 37(z-3) + 25(z-3)^2 + 8(z-3)^3 + (z-3)^4$

Complete Horner Method

inputs: n, $a_0, a_1, \dots, a_n$ (overwritten by result c_k), z_0

for $k = 0 .. n-1$

  for $j = n-1 : -1 : k$

    $a_j = a_j + z_0 a_{j+1}$

Now how do we use Horner's method to find roots of a polynomial?
We have an efficient way of computing $p(z_0)$ and $p'(z_0)$ for any z_0.

$$z_{k+1} = z_k - \frac{p(z_k)}{p'(z_k)}$$

All we need for Newton's method are first two rows of example (*).
We can write these in pseudocode as follows:

```
[α, β] = function Horner(n, a0, a1, .., an, z0)
    α = an
    β = 0
    for k = n-1 : -1 : 0
        β = α + z0 β
        α = ak + z0 α
    return α, β
    /* note: α = p(z0) and β = p'(z0) */
```

Here is how this algorithm would run for $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$
$(n=4)$

before loop

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & & & & \\ \hline & 1_{\alpha} & & & & \end{array}$$

$\beta = 0.$

$k = 3$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & & & \\ \hline & 1 & -1_{\alpha} & & & \\ 3 & & & & & \\ \hline & 1_{\beta} & & & & \end{array}$$

$k = 2$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & & \\ \hline & 1 & -1 & 4_{\alpha} & & \\ 3 & & 3 & & & \\ \hline & 1 & 2_{\beta} & & & \end{array}$$

$k = 1$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & \\ \hline & 1 & -1 & 4 & 7_{\alpha} & \\ 3 & & 3 & 6 & & \\ \hline & 1 & 2 & 10_{\beta} & & \end{array}$$

$k = 0$

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & 19_{\alpha} \\ 3 & & 3 & 6 & 30 & \\ \hline & 1 & 2 & 10 & 37_{\beta} & \end{array}$$

$\alpha = p(z_0)$

$\beta = p'(z_0)$

Thus Newton's iteration would be:

z_0 = given ; $a_n, a_{n-1}, \ldots, a_0$ = poly. coeff.

for $j = 1$: Max Iter

$[\alpha, \beta] = \text{horner}(n, a_0, a_1, \ldots, a_n, z_0)$

$z_1 = z_0 - \alpha/\beta$

if $|z_1 - z_0| < \varepsilon$ stop

$z_0 = z_1$

Once one has a root to within tolerance ⇝ deflate i.e. work with

$$q(z) = \frac{p(z) - p(z_0)}{(z - z_0)}$$ instead of original polynomial and repeat . .

However errors can accumulate so the "roots" obtained by successive deflation could not be as accurate as we want them. This can be remediated by using roots as initial guess for Newton's method on the original polynomial $p(z)$.

Another potential problem with using Newton's method or Secant method to find roots of polynomials is that if initial guess is real and poly. has real coeff then all successive iterates remain real and one cannot converge to a complex root.

Example (from HW2):

Newton's method diverges on $p(x) = x^2 + 1$, regardless of initial guess.

possible solution: take complex initial guess & use complex arithmetic (Could be more expensive: one complex mult is roughly 6 real operations)

Theorem on real quadratic factor:

If p is a polynomial with real coeff. and if w is a non-real root of p then $\overline{w}$ is also a root and $(z-\overline{w})(z-w)$ is a real quadratic factor of p.

Proof: $p(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0$

$$p(w) = a_n w^n + a_{n-1} w^{n-1} + \dots + a_1 w + a_0 = 0$$

Take conjugates on both sides. Recall that for some complex

$$z = x_1 + i x_2 \, , \quad \begin{aligned} x_1 &= \mathrm{Re}\, z \in \mathbb{R} \\ x_2 &= \mathrm{Im}\, z \in \mathbb{R} \end{aligned}$$

the conjugate is:

$$\overline{z} = x_1 - i x_2$$

We have properties: $\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2}$

$$\overline{z_1 z_2} = \overline{z_1}\,\overline{z_2} .$$

thus:

$$0 = \overline{p(w)} = \overline{a_n}\,\overline{w}^n + \overline{a_{n-1}}\,\overline{w}^{n-1} + \dots + \overline{a_1}\,\overline{w}^1 + \overline{a_0}$$

coeff of p are real $\longrightarrow$ $= a_n \overline{w}^n + a_{n-1} \overline{w}^{n-1} + \dots + a_1 \overline{w}^1 + a_0$

$$= p(\overline{w})$$

since $w \neq \overline{w}$ p must contain quadratic factor:

$$(z-w)(z-\overline{w}) = z^2 - wz - \overline{w}z + w\overline{w}$$

$$= z^2 - 2z\,\mathrm{Re}(w) + |w|^2 \quad \square .$$

There is a method called Bairstow's method that exploits this theorem to find roots of poly w/ real coeff by just using real arithmetic. For ref see §3.5 in Kincaid & Cheney.