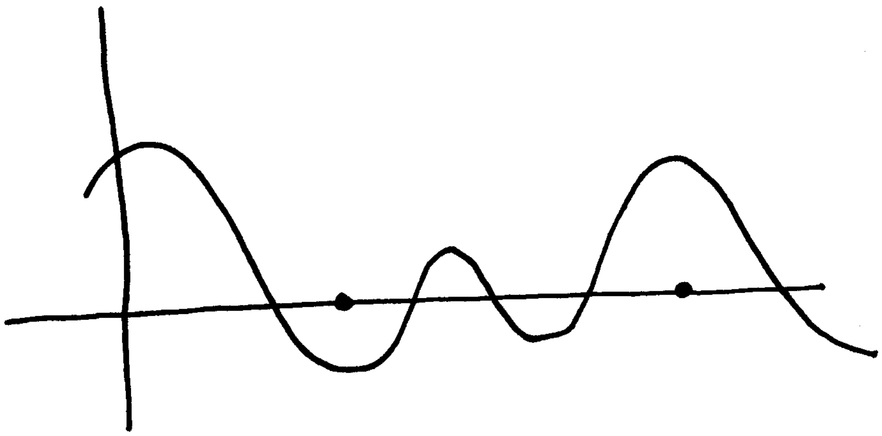
3.1. Finding Roots of Functions.

We are interested in solving equations of the form $f(x)=0$ for x.



Idea 1. Given an interval $[a,b]$ where sign $f(a) \neq$ sign $f(b)$, we compute $f(c)$ where $c = \frac{a+b}{2}$ and restrict our attention to either $[a,c]$ or $[c,b]$ depending on sign $f(c)$.

This is called the bisection method. ⁑

<Mathematica demonstration>

After n steps, the error in the computed position of the root is at most $\frac{b-a}{2^{n+1}}$.

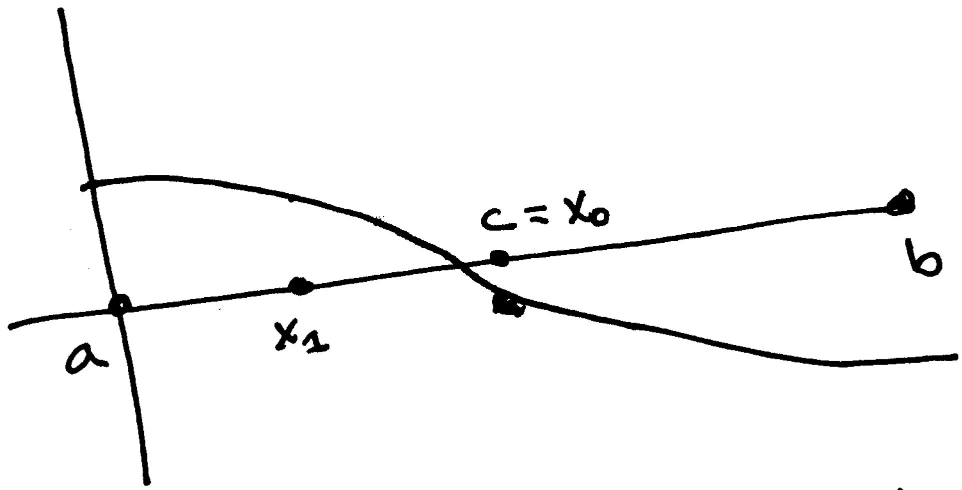
Definition. If $\{x_n\} \to x$, then the sequence has <u>linear</u> convergence if $\exists\, C \in [0,1)$ so that

$$|x_{n+1} - x| \le C\,|x_n - x|$$

Lemma. If $\{x_n\} \to x$ linearly with constant C, then $|x_{n+1} - x| \le A C^n$, where $A = |x_1 - x|$.

Question. Does the bisection method converge linearly? (We take the sequence to be the sequence of midpoints, and x to be whatever root the method converges to.)

Consider

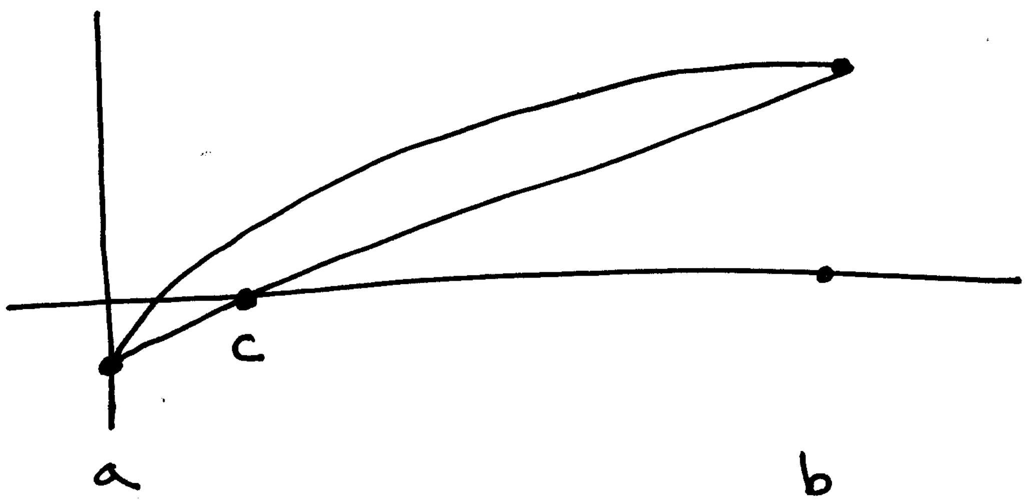


On step 1, we replace our intial guess of $a+b/2$ by $\dfrac{a+\frac{a+b}{2}}{2} = \frac{3}{4}a + \frac{1}{4}b =: x_1$.
But x_0 was actually closer to the root than x_1!

We see that bisection is <u>not</u> guaranteed to improve at each step.

On the other hand, bisection does give the conclusion of the Lemma, which is almost equally useful in practice.

We can improve the bisection method by changing our choice of midpoint.



The "false position" method guesses that the function is well approximated by the secant line through $(a, f(a))$ and $(b, f(b))$ and chooses the guess for the new zero accordingly.

This can have linear (and even superlinear) convergence if the details are handled right... we will return to this method soon!

What if we can calculate a derivative of our function?

Estimating where the tangent line crosses the x-axis is the basis for Newton's Method.

Doing the algebra establishes that

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

<newton_method.nb demonstration>

We saw that the number of correct digits increases exponentially. To be more precise, we will prove

Definition. We say $\{x_n\} \to x$ quadratically if $|x_{n+1} - x| \leq c|x_n - x|^2$ for some c.

Observe that if x_n has K correct decimal digits, then $|x_n - x| < 10^{-K}$, so $|x_{n+1} - x| < c(10^{-K})^2 = c\,10^{-2K}$, and the number of correct digits has ~~almost~~ approximately doubled (depending on c).

Newton's Method Theorem. If f, f', f'' are continuous in a neighborhood of a root r of f, and $f'(r) \neq 0$, there is a neighborhood N_δ of r of radius δ so that if $x_0 \in N_\delta$ then all $x_n \in N_\delta$ and

$$|r - x_{n+1}| \leq c(\delta)\,|r - x_n|^2$$

for some c depending on f and δ (given below).

Proof. Let $e_n = r - x_n$. We know

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

$$= (r - x_n) + \frac{f(x_n)}{f'(x_n)}$$

$$= e_n + \frac{f(x_n)}{f'(x_n)}$$

$$= \frac{e_n f'(x_n) + f(x_n)}{f'(x_n)}.$$

Now let's Taylor expand f around x_n. We know

$$0 = f(r) = f(x_n + e_n) = f(x_n) + e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n)$$

where $\xi_n \in [x_n, r]$. This means

$$e_{n+1} = -\frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} e_n^2,$$

which is almost what we want.

Observe that we can define a function

$$c(\delta) = \frac{1}{2} \frac{\max_{N_\delta} |f''(x)|}{\min_{N_\delta} |f'(x)|}$$

which is finite for small enough δ. In fact, we can choose δ small enough that

$$\delta \, c(\delta) < 1$$

since as $\delta \to 0$, $c(\delta) \to \frac{f''(r)}{f'(r)}$. Now all we have to observe is that if $x_n \in N_\delta$,

$$\left| \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} \right| \leq c(\delta)$$

since, x_n, ξ_n are in N_δ. In this case

$$|e_{n+1}| \leq c(\delta) e_n^2 \leq \delta \, c(\delta) e_n < e_n,$$

so x_{n+1} is in N_δ as well. So if we choose x_0 in N_δ, all subsequent x_n are also in N_δ.

We now only have to show that $\{x_n\} \to r$.

Observe

$$|e_n| < \delta c(\delta)\, |e_{n-1}| < \ldots < (\delta c(\delta))^{n+1} e_0.$$

Since $\delta c(\delta) < 1$, this means $\{e_n\} \to 0$, as desired.

A very cool extension of Newton's method is commonly used to solve systems of nonlinear equations.

Idea: Given the system

$$f_1(x_1, \ldots, x_N) = 0$$

$$\vdots$$

$$f_N(x_1, \ldots, x_N) = 0$$

we can think of this as a map

$$F : \mathbb{R}^n \to \mathbb{R}^n, \text{ given by } F(x_1, \ldots, x_n) = (f_1, \ldots, f_n).$$

We then write down Newton's method, recalling that ~~the~~

$$F(\vec{x}+\vec{h}) \approx F(\vec{x}) + DF_x(\vec{h}) = L(\vec{h})$$

where

$$DF_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Now the linear approximation $L(\vec{h})$ vanishes when $\vec{h}$ solves the linear system

$$(DF_x)\,\vec{h} = -F(\vec{x}).$$

So we let $\vec{h}_n$ be the solution to

$$(DF_{x_n})\,\vec{h} = -F(\vec{x}_n)$$

and set

$$\vec{x}_{n+1} = \vec{x}_n + \vec{h}_n.$$
$$= \vec{x}_n - (DF_x)^{-1} F(\vec{x}_n).$$

Notice that this only works when DF_x is a nonsingular matrix, just as 1-d Newton's method works only for $f' \neq 0$.

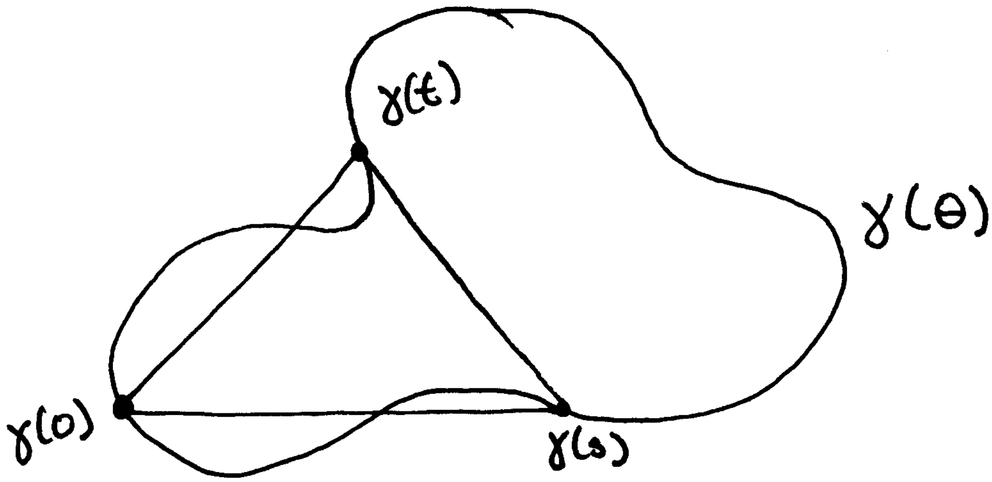
Example. <computer>

One of the interesting features of Newton's method is how it chooses a final point to converge to.

<demonstration with solving $z^3 = 1$ in complex plane>

Example. (Real-world)

Suppose we have a plane curve



so that $\gamma(0)$ = the origin. We are interested in finding s,t on the curve so that

$\gamma(0), \gamma(s), \gamma(t)$ forms an equilateral triangle

with Newton's method. We need to write this in the form of a pair of equations for s and t, preferably with the Jacobian of our system nonsingular at the solution.

Let

$$f_1 = |\gamma(s)|^2 - |\gamma(t)|^2$$

$$f_2 = |\gamma(s)|^2 - |\gamma(s) - \gamma(t)|^2.$$

We see the Jacobian is

$$\frac{\partial f_1}{\partial t} = -\frac{d}{dt}\langle \gamma(t), \gamma(t)\rangle = -2\langle \gamma'(t), \gamma(t)\rangle.$$

$$\frac{\partial f_1}{\partial s} = \frac{d}{ds}\langle \gamma(s), \gamma(s)\rangle = 2\langle \gamma'(s), \gamma(s)\rangle.$$

$$\frac{\partial f_2}{\partial s} = 2\langle \gamma'(s), \gamma(s)\rangle - 2\langle \gamma'(s), \gamma(s) - \gamma(t)\rangle$$
$$= 2\langle \gamma'(s), \gamma(t)\rangle$$

$$\frac{\partial f_2}{\partial t} = +2\langle \gamma'(t), \gamma(s) - \gamma(t)\rangle.$$

or

$$DF_{(s,t)} = 2\begin{bmatrix} \langle \gamma'(s), \gamma(s)\rangle & -\langle \gamma'(t), \gamma(t)\rangle \\ \\ \langle \gamma'(s), \gamma(t)\rangle & \langle \gamma'(t), \gamma(s) - \gamma(t)\rangle \end{bmatrix}$$

Plugging into <u>Mathematica</u>, we get...

<triangle_on_curve.nb>

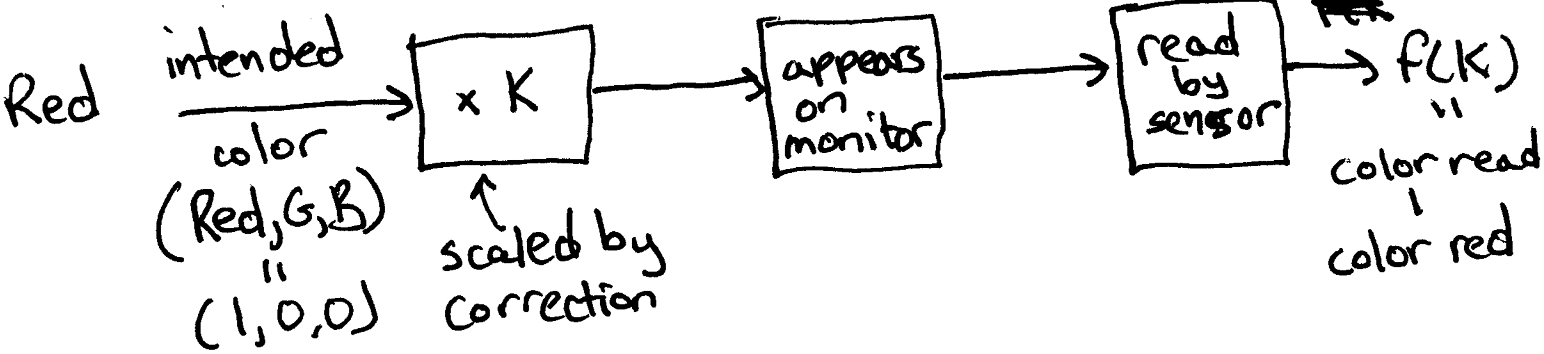
That was cool!

Rootfinding without derivatives.

Example. Color Correction.

A difficult problem is making sure that computer monitors display "true" color representations.

We usually fix this with a sensor attached to the monitor



This procedure takes   seconds.

Imagine designing software to choose K.

We don't know f'(K) or f(K) explicitly.

Example 2. Intake for jet engine.

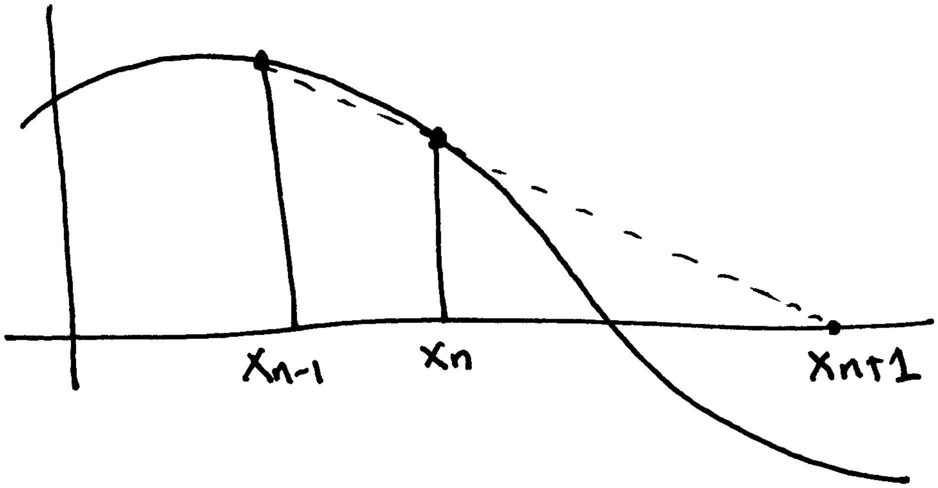
It's known that noise produced by an engine on a cruise missile is affected by the angle θ of an air intake ramp.

The noise is extracted from a numerical simulation that takes about an hour to run.

Solving $\text{Noise}(\theta) = \text{desired noise}$ requires a numerical method, hopefully a fast one.

Again, $\text{Noise}'(\theta)$ is not available.

The secant method



We don't know $f'(x_n)$, but we can approximate it, we hope, with the slope

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

so we compute

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

or

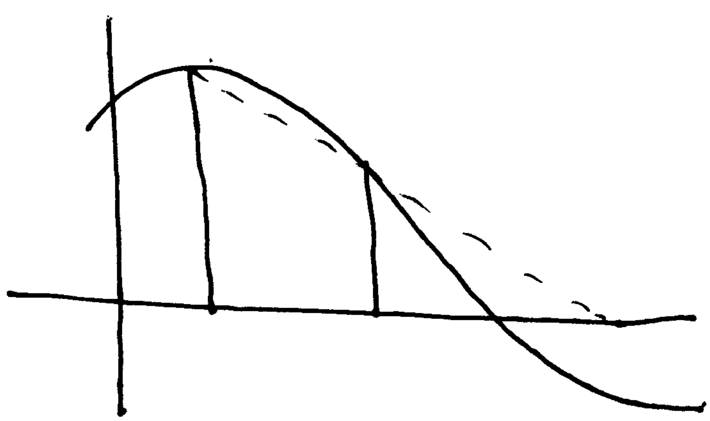
$$-f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} = x_{n+1} - x_n$$

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

Keep in mind that this is going to get dicey as $f(x_n), f(x_{n-1})$ and x_n, x_{n-1} approach each other. So we should stop iterating when

$$\frac{|f(x_n) - f(x_{n-1})|}{|f(x_n)|} < \text{some relatively large number like } 10^{-6}$$

Convergence of the secant method.



It turns out to be the case that

$$e_{n+1} = -\frac{1}{2}\left(\frac{f''(\xi_n)}{f'(\gamma_n)}\right) e_n e_{n-1} \approx -\frac{1}{2}\left(\frac{f''(r)}{f'(r)}\right) e_n e_{n-1}$$

~~We see that~~ Suppose we can get

$$|e_{n+1}| \leq C|e_n|^{\alpha}$$

We want to solve for α, assuming there is a c with

$$|e_{n+1}| \leq c|e_n||e_{n-1}|.$$

If $c|e_0|, c|e_1| < D$, we see

$$c|e_2| \leq c|e_1|\, c|e_0| \leq D^2$$
$$c|e_3| \leq c|e_2|\, c|e_1| \leq D^2 D^1 = D^3$$
$$c|e_4| \leq c|e_3|\, c|e_2| \leq D^3 D^2 = D^5$$

In general,

$$c|e_n| \leq D^{\lambda_{n+1}}$$

where

$$\lambda_0 = 1, \quad \lambda_1 = 0, \qquad \lambda_n = \lambda_{n-1} + \lambda_{n-2}.$$

These numbers are the Fibonacci numbers (!).
In particular,

$$\lambda_n = \frac{1}{\sqrt{5}}\left(\gamma^n - \beta^n\right)$$

where $\gamma = \frac{1}{2}(1+\sqrt{5})$, $\beta = \frac{1}{2}(1-\sqrt{5})$.

~~To solve for α, we observe that in general,~~

We now know for <u>any</u> α.

$$
\begin{aligned}
|e_{n+1}| &\lesssim c\,|e_n|\,|e_{n-1}| \\
&= c\,|e_n|^{\alpha}\,|e_n|^{1-\alpha}\,|e_{n-1}| \\
&\approx c\,|e_n|^{\alpha}\left(c^{-1}D^{\lambda_{n+1}}\right)^{1-\alpha}\left(c^{-1}D^{\lambda_n}\right) \\
&= |e_n|^{\alpha}\,c^{1-(1-\alpha)-1}\,D^{(1-\alpha)\lambda_{n+1}+\lambda_n} \\
&= |e_n|^{\alpha}\,c^{\alpha-1}\,D^{-\alpha\lambda_{n+1}+\lambda_{n+1}+\lambda_n} \\
&= |e_n|^{\alpha}\,c^{\alpha-1}\,D^{\lambda_{n+2}-\alpha\lambda_{n+1}}.
\end{aligned}
$$

So we need to choose α so that $\lambda_{n+2}-\alpha\lambda_{n+1}\to 0$ as $n\to\infty$. Luckily, we know for $\alpha=\frac{1}{2}(1+\sqrt{5})$, we have

$$
\begin{aligned}
\lambda_{n+2}-\alpha\lambda_{n+1} &= \frac{1}{\sqrt{5}}\left(\alpha^{n+2}-\beta^{n+2}\right)-\frac{1}{\sqrt{5}}\left(\alpha^{n+2}-\alpha\beta^{n+1}\right) \\
&= \frac{1}{\sqrt{5}}\left(\alpha\beta^{n+1}-\beta^{n+2}\right) \\
&= \frac{1}{\sqrt{5}}(\alpha-\beta)\,\beta^{n+1}
\end{aligned}
$$

where

$$\beta = \frac{1}{2}(1-\sqrt{5}) < 1,$$

so we see that $\alpha = \frac{1}{2}(1+\sqrt{5})$ works.

We comment that this means we have convergence faster than a linear method, but slower than a quadratic method.